Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ciencias de la Computación e Informática

Revisión de temas avanzados para la carrera de Bachillerato en Computación con Énfasis en Ingeniería de Software: Un conjunto de estudios empíricos

Patricia Agüero Flores A60070
Elizabeth Gamboa Bermúdez B22649
Cruz Maricel Monge Guzmán B34367
Mauricio Pandolfi González B14879

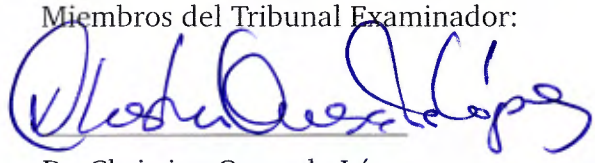
Memoria de Seminario de Graduación

Ciudad Universitaria Rodrigo Facio
San Pedro, San José, Costa Rica

2020

Este proyecto de graduación ha sido aceptado por el Tribunal Examinador como requisito parcial para optar al grado académico de Licenciatura en Computación e Informática.

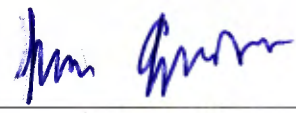
Miembros del Tribunal Examinador:



Dr. Christian Quesada López
Director TFG

Dra. Alexandra Martínez Porras
Miembro del Comité Asesor

Dr. Marcelo Jenkins Coronas
Miembro del Comité Asesor



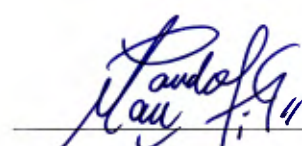
Dr. José Guevara Coto
Profesor Invitado

Dr. Luis Guerrero Blanco
Presidente

Patricia Agüero Flores
Sustentante, Carné A60070

Elizabeth Gamboa Bermúdez
Sustentante, Carné B22649

Cruz Maricel Monge Guzmán
Sustentante, Carné B34367



Mauricio Pandolfi González
Sustentante, Carné B14879

Dedicatoria

Dedicatoria de Patricia

Dedico esta investigación a Dios, por permitirme su desarrollo y conclusión. A mi papá por ser el pilar que me apoya en mis proyectos y quien me brindó la oportunidad de iniciar este largo proceso desde la Telesecundaria, el Bachillerato por Madurez y todos estos años en la Universidad. A mi adorada mamita que me enseñó a esforzarme y luchar por mis metas, ella en este momento me acompaña desde el cielo y la llevo siempre en mi corazón. A mis hermanas: Rosalba, Andrea y Saray que se desvelaron y me ayudaron en el proceso. A mi hermano Juan Francisco y mi cuñada Suyen, quienes me apoyaron y aconsejaron. A mi ahijada Lucía que me motiva a seguir adelante.

Dedicatoria de Elizabeth

Dedico este trabajo primeramente a Dios, por darme la perseverancia y a quién me hizo continuar con el trabajo. A mi familia, mi mamá Emérita, mi papá Luis y a mis hermanos: Laura, Luis y Gladys, por el constante apoyo y palabras de aliento que me brindaron para seguir adelante y poder sacar este proyecto.

Dedicatoria de Cruz Maricel

Dedico este trabajo principalmente a Dios, por darme la bendición de haber llegado hasta este momento tan trascendental en mi carrera profesional. A mis papás por ser el pilar de apoyo más importante e incondicional durante mi vida. A mis hermanas, Katherine y Keily, por escucharme y estar dispuestas a ayudarme cada vez que lo necesitaba. A mi tía Cynthia por el apoyo y consejos que siempre me ha dado. A mi familia en general y amigos que entendieron el sacrificio y esfuerzo que implicó este proyecto para mí.

Dedicatoria de Mauricio

A mi mamá, quien hizo su tesis conmigo en el vientre. Pasar por esta carrera, la otra, y por esta Memoria, se dio porque siempre tuve ese apoyo incondicional de parte de ella. Cumplir mis sueños siempre ha sido su prioridad. Me enseñó sobre la excelencia y cómo hacer las cosas desde el corazón. Estaré eternamente agradecido, por tanto amor entregado.

A mi papá, porque mi éxito académico es, a gran escala, por lo que me inculcó cuando hacíamos los carteles para la escuela: "las cosas se hacen bien, o mejor no se hacen". Mientras elaboraba este trabajo luchamos juntos, en las horas de hospital y las citas médicas aprendí muchísimo de la vida. Por eso, el hecho que yo le dedique este trabajo no puede ser más simbólico. Mi papá es la persona más inteligente que he conocido, pero durante el trabajo en este documento, me demostró que también es la más fuerte.

Papi y mami, ustedes me acercaron por primera vez a una computadora, me dieron educación, y gracias a eso ahora voy a ser licenciado. Quiero aclararles que no solo les dedico este logro, sino también todos los de mi vida. Mis éxitos, en cualquier área, siempre son, en primer lugar, por ustedes. Los amo.

Entonces, a mis héroes y ejemplos: Leonardo Pandolfi Lizano y Giselle González Arce.

Prefacio

El presente documento reúne un conjunto de investigaciones empíricas del área de Ingeniería de *Software*, las cuales se desarrollaron para optar por el grado de Licenciatura en Ciencias de la Computación e Informática de la Universidad de Costa Rica.

Las investigaciones se realizaron en el contexto del Seminario "Revisión de Temas Avanzados para la Carrera de Bachillerato en Computación con Énfasis en Ingeniería de *Software*: un conjunto de estudios empíricos", cuyo objetivo es generar conocimiento en áreas innovadoras de la Ingeniería de *Software* y ofrecer material de referencia actual y basado en la literatura, para ser utilizados en la carrera de Bachillerato en Computación que imparte la Escuela de Ciencias de la Computación e Informática de la Universidad de Costa Rica.

Esta memoria da a conocer cuatro temas relacionados con herramientas de pruebas de *software* para evaluar la accesibilidad y seguridad Web, y a técnicas de minería de datos y aprendizaje automático aplicadas a los contextos de la segmentación de clientes y la clasificación de noticias Web. La implementación de estos temas se hizo siguiendo los procedimientos de una metodología para mapeos sistemáticos de literatura, los cuales consistieron en la definición de un objetivo y preguntas de investigación a partir de una problemática. Posteriormente, se desarrolló un proceso de búsqueda automatizada de estudios en bases de datos digitales de artículos científicos, se realizó una selección de dichos estudios mediante criterios de inclusión y exclusión, se aplicaron reglas de evaluación de calidad según la relevancia de cada uno para la investigación. Por último, se llevó a cabo la extracción de datos y su análisis con el fin de dar respuesta al problema planteado.

Esta memoria constituye un esfuerzo por obtener una visualización sobre temas

relevantes que han sido trabajados en el área de la Ingeniería de *Software*. La elección de estos se hizo mediante una combinación entre el ámbito académico y la experiencia e interés profesional en el campo en que se desarrolla cada uno de los investigadores.

Agradecimientos

Agradecimientos de Patricia

Una gratitud sincera a mis compañeras y compañero de grupo, por el apoyo, la entrega responsable, la constancia, el respeto y, además, por saber comprender mis fortalezas y debilidades, por soportar los malos ratos de discusiones productivas para poder sacar adelante este complicado proceso.

A los profesores Christian, Alexandra y Marcelo, por la sapiencia para saber dirigir este Seminario y compartir sus conocimientos y experiencias, lo cual dio más valor a este aprendizaje, no sólo para mi vida profesional, sino también personal.

A todas aquellas personas, quienes aportaron un granito de arena, de una u otra forma, para que este proyecto fuese una realidad.

Agradecimientos de Elizabeth

Agradezco a mis amigos por el apoyo que siempre me han dado. A Maricel, por ser parte importante a lo largo de la carrera y de los proyectos que hemos llevado juntas. A mis compañeros de tesis y a los profesores tutores por la constante ayuda y los consejos brindados durante el desarrollo de este trabajo de investigación. Y a las demás personas que fueron partícipes en este proyecto de mi vida.

Agradecimientos de Cruz Maricel

A mis compañeros durante este proceso: Patricia, Elizabeth y Mauricio, sin el apoyo, esfuerzo y dedicación que pusieron no hubiera sido posible permanecer durante todo el avance del trabajo.

A mis profesores tutores por la guía, correcciones y consejos que nos brindaron

durante el desarrollo de esta investigación.

A todas esas personas importantes que tuvieron que escuchar mis quejas, aguantar mi humor cuando estaba frustrada, ayudarme a encontrar motivación cuando ya no la tenía, de verdad, una y mil gracias.

Agradecimientos de Mauricio

Gracias a los que estuvieron y van a estar siempre: mi familia. Gracias a Leo, por ser mi ejemplo en temas de computación y estudio; y a Chelo, por serlo en cuanto a disciplina; mis dos queridos hermanos, porque siempre encuentro en ustedes un fuerte apoyo. A mis compañeras de equipo Mari, Eli y Patri, quienes por este trabajo se convirtieron en mis amigas. Fueron pilares fundamentales en todo el proceso. A mis tías, por el apoyo de siempre. Tía Mary: recibirme cuando lo necesité significó para mí mucho más de lo que puedo expresar. Tía Marlen: gracias por el gesto invaluable de ayudarnos con la corrección de este documento. A mi abuelito Gonzalo y abue Luz, quienes me acompañaron desde el cielo. Gracias a Rebe, mi compañía estrella, por escucharme, comprenderme y darme amor en los momentos justos. En general, a todos los que comprendieron mis malos humores y mis cancelaciones de planes: muchas gracias.

Índice general

Índice de figuras

Índice de cuadros

Lista de Acrónimos

AJAX	*Asynchronous JavaScript And XML*.
API	*Application Programming Interface* (Interfaz de programación de aplicaciones).
ATUSA	*Automatic testing of AJAX user interface* (Interfaz de usuario de pruebas automática de AJAX).
CSRF	*Cross-site request forgery* (Falsificación de petición en sitios cruzados).
DAST	*Dinamyc Application security testing* (Pruebas de seguridad de aplicaciones dinámicas).
DNS	*Denial of services* (Negación de servicios).
FN	*False negatives* (Falsos negativos).
FP	*False positives* (Falsos positivos).
GQM	*Goal Question Metric* (Meta pregunta métrica).
HCI	*Human-Computer Interaction* (Interacción humano-computador).
IAAT	*Injection Aware Application Testing* (Pruebas de aplicaciones de prevención de inyección).
IDS	*Signature based intrusion detection systems* (Sistemas de detección de intrusos).
IEC	*International Electrotechnical Commission* (Comisión Electrotécnica Internacional).

IMAATT	*Integrated MultiAgent Testing tool* (Herramienta de pruebas multiagente integrada).
IS	Ingeniería de *software*.
ISE	Ingeniería de *software* experimental.
ISO	*International Standard* (Estándar internacional).
ISTA	*Integration and system test Automation* (Sistema de pruebas automatizadas y de integración).
JWAST	*Java Web Application Security Tester* (Pruebas de seguridad en aplicaciones Web de Java).
LDAP	*Lightweight Directory Access Protocol* (Protocolo ligero de acceso a directorios).
PBST	*Pattern Based Security Testing tool* (Herramienta de pruebas de seguridad basada en patrones).
PICO	*Population Intervention Comparison Outcome* (Población-Intenvención-Comparación-Salida).
PROSIC	Programa de la Sociedad de la Información y el Conocimiento.
RBVT	*Risk-based vulnerabilities testing* (Pruebas de vulnerabilidad basada en riesgos).
SEES	*Software security evaluation system* (Sistema de evaluación de seguridad de Software).
SOAP	*Simple Object Access Protocol* (protocolo de acceso simple a objetos).
SQL	*Structured Query Languaje* (lenguaje de consulta estructurada).
SQLIDT	*SQL Injection Vulnerability Detection Tool* (Herramienta de detección de vulnerabilidades de inyección SQL).
SQLMI	*SOLve and Mutation-based test generation for XML Injection* (Generación de pruebas para inyección XML basadas en mutación).
SQTL	*Scalable Quality and Testing Lab* (Laboratorio de Calidad Escalable y Pruebas).

TN	True negatives (Verdaderos negativos).
TP	True positives (Verdaderos positivos).
W3C	*World Wide Web Consortium* (Consorcio WWW).
WAI	*Web Accessibility Initiative* (Iniciativa de Accesibilidad Web).
WCAG	*Web Content Accessibility Guidelines* (Directrices de Accesibilidad del Contenido Web).
WS	*Web services* (Servicio Web).
WSVTS	*Web service vulnerability testing system* (Sistema de pruebas de vulnerabilidad en servicios Web).
XML	*EXtensible Markup Languaje* (Lenguaje de marcado extensible).
XMLI	*EXtensible Markup Languaje Injection* (Lenguaje de marcado extensible de inyección).
XPath	*XML Path Language* (Lenguaje de ruta de XML).
XSS	*Cross-site scripting* (Secuencia de comandos en sitios cruzados)

Resumen de la memoria

El presente documento comprende cuatro mapeos sistemáticos de literatura, donde los temas estudiados corresponden a herramientas de pruebas de *software* para evaluar la accesibilidad y seguridad Web, y las técnicas de minería de datos y aprendizaje automático aplicadas a los contextos de la segmentación de clientes y la clasificación de noticias Web.

Para realizar cada uno de los mapeos, se utilizaron los lineamientos establecidos por Petersen, Vakkalanka y Kuzniarz [1] y Kitchenham y Charters [2] para el diseño, conducción y análisis de estudios secundarios en la Ingeniería del *Software*. La formulación de los objetivos y delimitación del alcance de los estudios se realizó con el modelo GQM (objetivo, preguntas, métricas, por sus siglas en inglés) [3]. A partir del objetivo se establecieron las preguntas de investigación que guiaron la identificación de los artículos de control para apoyar el proceso de definición del protocolo. A partir de las preguntas de investigación y los artículos de control se implementaron los modelos PICO (población, intervención, comparación, salidas, por sus siglas en inglés) [4], que ayudaron a establecer las cadenas de búsqueda automática utilizadas en las bases de datos digitales de artículos científicos. Por medio de la extracción de estudios de las bases de datos Scopus, IEEE Xplore y Web of Science se hizo la eliminación de duplicados y se incluyeron o excluyeron los estudios según los criterios definidos para cada investigación con base en el título, el resumen y las palabras clave. Para cada conjunto de estudios se realizó una evaluación de calidad de acuerdo con criterios definidos para determinar su relevancia en la investigación. Los estudios seleccionados en cada tema fueron leídos a profundidad para extraer y clasificar los resultados reportados en los formularios de extracción. Finalmente, se realizó el análisis y síntesis de resultados para responder a las preguntas planteadas. Los análisis

consistieron en la clasificación y agrupamiento de los datos por categorías, presentando dicha información en tablas y gráficos que muestran las tendencias del análisis que se realiza en las distintas áreas de estudio. Para cada tema se desarrolló un reporte técnico que se detalla en todos los capítulos de esta memoria, y un artículo científico que fue publicado en una conferencia científica.

A continuación, se describe brevemente cada uno de los mapeos realizados. El mapeo sobre herramientas de evaluación de accesibilidad Web tiene como objetivo identificar las herramientas para pruebas, los criterios de accesibilidad que se evalúan de acuerdo con el estándar WCAG y los desafíos reportados por los investigadores. El mapeo identificó 38 herramientas para evaluar la accesibilidad Web. Con la aplicación de estas herramientas se logró determinar los criterios que se reportan con más frecuencia como incumplidos, estos son: 1.1.1 Contenido no textual (23), 2.4.4 Propósito de los enlaces (en contexto) (15), 1.3.1 Información y relaciones (13), 3.1.1 Idioma de la página (12). Asimismo, los 5 desafíos técnicos más reportados fueron: la disponibilidad del personal de desarrollo Web capacitado en accesibilidad (4), la necesidad de evaluadores expertos en accesibilidad Web (3), la consideración de los criterios de accesibilidad durante la definición de los aspectos del diseño de sitios Web (3), la utilización de una herramienta de evaluación única como fuente de información (2) el lenguaje variable de los sitios Web (1). Los 3 desafíos sobre regulaciones de Gobierno y reglamentación son: los responsables de las políticas deben desarrollar y promover marcos legales y normativas para abordar los problemas de accesibilidad Web (7), se debe hacer que la accesibilidad de los sitios Web gubernamentales sea un requisito obligatorio (4), se deben fortalecer y compartir las políticas de accesibilidad Web de cada país, así como aplicar mejores leyes y fomentar prácticas que hagan que los sitios Web sean más accesibles (3). Los resultados muestran la necesidad de estudios adicionales que aborden de forma detallada, cuáles son las capacidades de las herramientas para evaluar la accesibilidad Web, así como, analizar qué se puede hacer automáticamente y qué solo por humanos y la mejor forma de combinar estrategias (herramientas y humanos) para mejorar la efectividad de las evaluaciones que se realizan.

La investigación sobre las herramientas que evalúan la seguridad de las aplicaciones Web tiene como objetivo identificar y conocer las herramientas que han sido utilizadas para detectar vulnerabilidades por medio de pruebas automatizadas. El mapeo

identificó 66 herramientas que evalúan la tendencia de vulnerabilidades de las aplicaciones Web utilizadas entre el 2006 y el 2019. Las herramientas se clasificaron según los tipos de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web. La categoría de pruebas para detectar vulnerabilidades más comunes fue la de *Input Validation Testing (4.8)* con 55 herramientas, seguido de las pruebas de *Configuration and Deployment Management Testing (4.3)*, *Session Management Testing (4.7)*, y *Client Side Testing (4.12)* con 15 herramientas utilizadas cada una. Los tipos de pruebas más reportados fueron los de la categoría *Input Validation Testing (4.8)*. En este caso *SQL Injection (4.8.5)* con 40 herramientas, *Cross-Site Scripting (4.8.2)* con 30 herramientas, y *Testing for HTTP Incoming Requests (4.8.17)* con 19 herramientas utilizadas. Los resultados muestran que existe una gran variedad de herramientas que evalúan la mayor cantidad de las tendencias de vulnerabilidades definidas por OWASP. Aunque se identificó otras más que realizan distintos tipos de pruebas de seguridad, solo pocas se encuentran en la lista de herramientas recomendadas por OWASP, lo que denota la necesidad de contar con evaluaciones empíricas de las existentes en el área.

El mapeo sobre técnicas de minería de datos y aprendizaje automático en el contexto de segmentación de clientes bancarios, tiene como objetivo caracterizar la literatura según las técnicas, las herramientas, los conjuntos de datos y las métricas de evaluación. Este mapeo logró identificar y clasificar 55 técnicas de minería de datos y aprendizaje automático en nueve paradigmas, como los más reportados aparecen *decision tree* y *linear predictors*. También se constató que 22 herramientas soportan la implementación de las técnicas, estas poseen diferentes características como: el tipo de licenciamiento, las restricciones para implementar y configurar las técnicas, entre otros. Con respecto a los conjuntos de datos, se registran 31 repositorios entre públicos y privados. El repositorio *UCI Machine Learning* de la Universidad de California fue el más utilizado. De la mayoría de estudios se extrajo la cantidad de registros, el enlace para consulta y sus atributos determinantes, entre los que destacan: la edad, el trabajo, el género, la temporalidad y las características crediticias del cliente. En el caso de las métricas de evaluación, la mayoría de estudios reportan las métricas estándar. La más aplicada fue *accuracy*, pues se presentó en el 78 % de los casos analizados. Los resultados demuestran que existen diversas técnicas de minería de datos y aprendizaje automático que se han aplicado al problema de segmentación de clientes, con

tendencias claras sobre los elementos evaluados.

El análisis de aprendizaje automático y minería de datos sobre noticias Web tiene como objetivo caracterizar las técnicas utilizadas en este contexto, los conjuntos de datos que los estudios emplearon y las métricas para evaluarlas. El mapeo incluyó 51 estudios en total y mostró que las técnicas de los paradigmas *clustering*, *support vector machines* y *generative models* fueron las más frecuentes, con 13, 10 y 10 apariciones respectivamente. La métrica más aplicada para evaluar fue *F-measure* con 25 registros. Los conjuntos de datos tuvieron orígenes desde extracciones propias de la Web (con 33 reportes) y conjuntos de datos de terceros (con 25 reportes). Aunque los estudios no mostraron un solo resultado en la especificación de las técnicas utilizadas, los conjuntos de datos y las métricas, estas describen distintos escenarios para la aplicación de las técnicas en la academia, la industria y la investigación.

Este trabajo realizó aportes desde tres áreas principales. En el ámbito profesional, los mapeos sistemáticos que identificaron técnicas y herramientas permitieron ampliar criterios técnicos para considerar su aplicabilidad en la industria. En el área de la investigación, el trabajo evidenció la utilidad de la aplicación de mapeos sistemáticos sobre metodologías empíricas, para considerar los estudios en la literatura relacionados a un tema específico, con el fin de recopilar sus resultados para análisis posterior. Finalmente, desde el ámbito académico, la información brindada sirve de insumo para identificar áreas de interés que permitan innovar en la carrera de Ciencias de la Computación e Informática.

Palabras clave

Accesibilidad, WCAG, pruebas automatizadas, seguridad, aplicaciones Web, minería de datos, aprendizaje automático, segmentación de clientes bancarios, clasificación de noticias Web, mapeo sistemático de literatura, estudios secundarios, estudios empíricos, Ingeniería de *Software* Empírica, Ingeniería de *Software*.

Introducción de la memoria

Bourque y Fairley [5] definen Ingeniería de *Software* (IS) como la aplicación de un acercamiento sistemático, disciplinado y cuantificable en el desarrollo, la operación y el mantenimiento del *software*.

La investigación aplicada a la IS conforma la Ingeniería de *Software* Experimental (ISE). La ISE busca que la IS sea un proceso más enfocado en lo científico y plantea cómo los estudios empíricos tienen un rol en la solución de ese objetivo. Para ello, se enfoca en estudiar los experimentos sobre sistemas de *software*, recoge sus datos y desarrolla nuevas teorías a partir del análisis de sus resultados. Así, logra caracterizar, evaluar y comparar los productos, procesos y recursos de la IS [6].

La experimentación es esencial para la evolución de la IS y ha permitido que la transferencia de las tecnologías a la industria mejore los procesos que en esta se utilizan. Además, la experimentación da evidencia científica sobre los resultados al adoptar nuevos paradigmas, procesos y tecnologías en el campo de la IS [6].

Considerando la importancia expuesta sobre los estudios empíricos, este documento presenta cuatro mapeos sistemáticos de literatura, los cuales son un medio para identificar, evaluar e interpretar toda la investigación disponible para preguntas en particular o áreas de interés [2]. Es importante porque permite sintetizar el trabajo existente, mediante un estudio exhaustivo y justo. Considera que la metodología sea cuidadosa y permite evaluar la integridad de la estrategia de búsqueda de la investigación, dándole así valor práctico y metódico [2].

Los mapeos sistemáticos de literatura evalúan e interpretan los estudios disponibles y relevantes para una pregunta de investigación particular. También, incentivan la indagación en temas relacionados con desarrollo y mejoras del *software*. Este tipo de investigaciones ha permitido que la IS se haya ampliado a otras disciplinas como

las que se estudian en este documento: minería de datos, aprendizaje automático, herramientas para evaluación de accesibilidad Web y seguridad de aplicaciones Web. No obstante, los resultados que se obtienen al utilizar esta metodología se limitan a lo que se reporta en los estudios incorporados para el análisis.

El trabajo realizado tuvo como objetivo generar conocimiento en temas avanzados de la Ingeniería de *Software* y ofrecer materiales actualizados que puedan ser utilizados en la carrera de Bachillerato en Computación e Informática con Énfasis en Ingeniería de *Software* que imparte la Escuela de Ciencias de la Computación e Informática de la Universidad de Costa Rica, mediante la aplicación de metodologías empíricas.

El capítulo 1 presenta un mapeo de literatura sobre herramientas de evaluación de accesibilidad Web con el fin de identificar las herramientas, criterios y desafíos en dicho contexto. La Web se ha convertido en un recurso esencial para distintos aspectos de la vida en sociedad, por ello es necesario que esta sea accesible para brindar igualdad de oportunidades a todas las personas que utilizan la Web. La accesibilidad puede ayudar a personas mayores con capacidades cambiantes debido al envejecimiento, con discapacidades temporales o con limitaciones situacionales. Este capítulo está basado en la revisión y análisis de 50 estudios primarios cuyas publicaciones identificadas se realizan entre los años 2004-2019. A partir del capítulo de la memoria se desarrolló un artículo científico que fue enviado y aceptado en las IV Jornadas Costarricenses de Investigación en Computación e Informática JoCICI 2019, celebradas en Universidad Estatal a Distancia (UNED) durante el 19 y 20 de agosto del 2019. El artículo titulado, Tools for the evaluation of web accessibility: A systematic literature mapping, fue publicado en la IEEE Explore.

El capítulo 2 presenta un mapeo sistemático sobre las herramientas utilizadas que evalúan la seguridad de las aplicaciones Web; estas detectan vulnerabilidades y amenazas externas hacia la página Web, con el fin de que su administrador aumente la seguridad. Es importante que el administrador asegure la seguridad del sitio Web y de la información de sus usuarios. Por ello, se presenta un mapeo de las herramientas con el objetivo de habilitar mecanismos para mejorar la confiabilidad y la seguridad de las aplicaciones Web. Este capítulo comprende la revisión y el análisis de 63 estudios publicados entre los años 2006-2019. A partir del desarrollo de este capítulo, se

generó un artículo científico el cual fue enviado y aceptado en *Iberoamerican Conference on Software Engineering* que se desarrollará el 16-20 de Noviembre, en Curitiba, Brasil. El artículo fue publicado en Scopus Proceedings of the XXIII Ibero-American Conference on Software Engineering.

El capítulo 3 presenta un mapeo sistemático de literatura sobre técnicas de minería de datos y aprendizaje automático en el contexto de segmentación de clientes bancarios. La minería de datos y el aprendizaje automático pertenecen a un amplio rango de tecnologías que buscan aplicar técnicas a conjuntos de datos. Esto con el fin de analizarlos y extraer información útil para solventar problemas en diferentes áreas. En el caso del sector bancario, la necesidad de conocer las características de cada cliente implica una ventaja empresarial porque pueden segmentarlos y ofrecerles productos y servicios cada vez más personalizados. El fin de este estudio es caracterizar la literatura según las técnicas, las herramientas, los conjuntos de datos y las métricas de evaluación. Este capítulo está basado en la revisión y análisis de 87 estudios primarios publicados en el período del 2005-2019. A partir del capítulo de la memoria se desarrolló un artículo científico que fue enviado y aceptado en la Intelligent Systems Conference (IntelliSys) que se desarrollará el 3-4 de setiembre del 2020 en Amsterdam, Holanda. El artículo fue publicado en el Springer series “Advances in Intelligent Systems and Computing” e indexado en ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar y Springerlink.

El capítulo 4 expone un mapeo sistemático sobre aprendizaje automático y minería de datos de noticias Web, con el propósito de caracterizar las técnicas utilizadas, así como los conjuntos de datos y las métricas que los estudios emplearon. En un momento en que las noticias se consumen virtualmente, hacer un análisis temático a partir de su clasificación puede ayudar a comprender conductas y descubrir patrones sobre el contexto social y político de la sociedad. Las técnicas de minería de datos hacen posible la detección de dichos patrones estructurales sobre este tipo de documentos, consideran además su avanzada complejidad y creciente volumen. Este capítulo contiene la revisión y análisis de 51 estudios primarios entre los años 2000-2019 que identificaron técnicas de minería de datos y aprendizaje automático en este contexto.

La estructura de cada capítulo presenta las siguientes secciones: Resumen, In-

troducción, Marco teórico, Trabajo relacionado, Metodología, Análisis de resultados, Discusión, Lecciones aprendidas, Conclusiones y Trabajo futuro y Anexos.

Capítulo 1

Herramientas para la evaluación de la accesibilidad Web: un mapeo sistemático de literatura

Patricia Agüero Flores

1.1. Resumen

Contexto: En los últimos años, se han propuesto diferentes herramientas para automatizar la evaluación de los criterios de accesibilidad a los contenidos Web por parte del Consorcio World Wide Web (W3C). Estas herramientas pueden verificar que un sitio Web cumpla con los estándares de accesibilidad Web como WCAG, pero los resultados de la evaluación pueden depender de la herramienta utilizada. **Objetivo:** este estudio tiene como objetivo identificar y caracterizar la literatura existente sobre las herramientas para la evaluación de la accesibilidad Web. **Método:** se hace un mapeo sistemático de estudios primarios que involucran el tema de la accesibilidad Web, en los que se mapean los criterios de accesibilidad evaluados con las herramientas según el estándar WCAG. **Resultados:** el análisis de un total de 50 estudios, registró las diferentes herramientas y, en algunos casos, las características utilizadas, y se identificaron los desafíos reportados para las evaluaciones de accesibilidad. **Conclusiones:** los resultados obtenidos identificaron un total de 38 herramientas diferentes

que se utilizan para evaluar accesibilidad Web. Por otro lado, quedó en evidencia que en los últimos diez años se han utilizado con más frecuencia las herramientas y que éstas ayudan a detectar criterios de accesibilidad tales como: contenido no textual, propósito de los enlaces, idioma de la página, etiquetas o instrucciones, entre otros. Sin embargo, queda para trabajo futuro determinar cuáles son las capacidades de las herramientas para evaluar la accesibilidad Web y cuáles son las más recomendadas, así como, analizar qué se puede hacer automáticamente y que solo por humanos.

1.2. Introducción

La Web se ha convertido en un recurso esencial para distintos aspectos de la vida en sociedad, tales como la educación, el empleo, el Gobierno, el comercio, la salud, y el entretenimiento. Por ello es necesario que la Web sea accesible, de manera que las personas con alguna condición de discapacidad tengan igualdad de oportunidades para participar activamente en la sociedad [1]. La accesibilidad Web significa que las personas que tienen alguna discapacidad puedan percibir, entender, navegar e interactuar con los sitios Web [1]. Entre las discapacidades que afectan el acceso a la Web están la discapacidad auditiva, la cognitiva, la neurológica, la física, y la visual.

La Organización Mundial de la Salud estima en 15 % la población con algún tipo de discapacidad. Las tecnologías con asistencia o de apoyo son usadas para mantener o mejorar las capacidades funcionales de las personas con discapacidad y han probado ser un factor de cambio [2]. A nivel mundial se realizan esfuerzos para construir ciudades más inteligentes, donde las tecnologías digitales se vuelven esenciales para mejorar la administración y los servicios. Uno de los retos más importantes es reducir la brecha digital y extender el acceso a todas las personas para reducir la desigualdad, principalmente de las personas en condición de discapacidad que se incluyen como minorías [3]. En Costa Rica, estudios especializados han mostrado que la población con discapacidad enfrenta múltiples barreras que limitan su desarrollo y ejercicio de los derechos ciudadanos, con desigualdades en el acceso a los distintos servicios fundamentales [4]. Asimismo, se ha reconocido el potencial de las tecnologías digitales como un generador de oportunidades para esta población [5].

En el año 2017, la Unión Internacional de Telecomunicaciones designó al Progra-

ma de la Sociedad de la Información y el Conocimiento (PROSIC) y al Centro de Informática de la Universidad de Costa Rica, como el ente acreditador para desarrollos digitales accesibles [6]. Asimismo, iniciativas como el Observatorio de Tecnologías Accesibles Inclusivas del Instituto Tecnológico de Costa Rica buscan reducir las brechas existentes en cuanto a la accesibilidad digital con el fin de promover la igualdad de oportunidades.

Con la implementación de la Ley para la Igualdad de Oportunidades para las Personas con Discapacidad (Ley No.7600) y el Pacto por un País Accesible e Inclusivo del Gobierno de la República, las instituciones del sector público han realizado esfuerzos para el desarrollo de sitios Web más inclusivos. Distintas evaluaciones han determinado la accesibilidad de los sitios de estas instituciones. Por ejemplo, la Defensoría de los Habitantes y el Centro de Investigación y Capacitación en Administración Pública de la Universidad de Costa Rica aplican anualmente el Índice de Transparencia del Sector Público, donde contemplan la accesibilidad Web para determinar la disponibilidad de textos alternativos en las imágenes con enlaces, el tamaño de botones y uso de subtítulos o lenguaje de señas, entre otros. En el 2018, la accesibilidad Web fue uno de los aspectos con menor calificación en los sitios Web de las instituciones públicas. Por todo lo anterior, es esencial que la accesibilidad Web sea considerada como parte del desarrollo de los servicios que se ofrecen a través de estos sitios.

Para el desarrollo de sitios Web es deseable contar con herramientas automatizadas que permitan la evaluación de la accesibilidad Web; sin embargo, estas herramientas pueden tener limitaciones en cuanto a los criterios que plantean los estándares de accesibilidad Web [7]. Más aun, existen distintos instrumentos de evaluación de accesibilidad Web, según lo indicado por las normas WCAG. Además, constantemente se trabaja en redefinir la accesibilidad Web, de manera que incluya componentes clave considerados por investigadores y profesionales [8], por lo que las herramientas deben ir evolucionando de acuerdo con las necesidades que surjan.

El objetivo de este estudio es identificar y caracterizar la literatura existente sobre herramientas para la evaluación de la accesibilidad Web.

Para el desarrollo del estudio, se realizó un mapeo sistemático de literatura que clasificó la evidencia existente en cuanto a las herramientas para evaluar la accesibilidad Web, los criterios de accesibilidad Web evaluados, el nivel de cumplimiento con

respecto a los estándares de accesibilidad y los desafíos de las evaluaciones realizadas.

El documento consta de la siguiente estructura: la sección 1.2 es una introducción al estudio. La sección 1.3 define los conceptos más relevantes del área de investigación. La sección 1.4 detalla los trabajos de autores que están relacionados con el tema de análisis de este mapeo. La sección 1.5 explica la metodología utilizada para hacer el mapeo sistemático de literatura. La sección 1.6 analiza la calidad de los estudios . La sección 1.7 describe el proceso de extracción de la información. La sección 1.9 expone las amenazas a la validez detectadas en el desarrollo del estudio. La sección 1.10 analiza los resultados de acuerdo con las preguntas de investigación. La sección 1.11 discute sobre dichos resultados. La sección 1.12 detalla lecciones aprendidas durante el proceso, y en la sección 1.13 se presentan las conclusiones. Además, se encuentra la sección del Apéndice conformada por los anexos en los que se presentan tablas con la totalidad de los estudios analizados, así como los resultados de la evaluación de calidad y otros datos obtenidos.

1.3. Marco teórico

Las pautas de accesibilidad para el contenido de la Web (WCAG: Web Content Accessibility Guidelines) [9] del Consorcio Web [1] definen un conjunto de criterios que los sitios Web deben cumplir para alcanzar cierto nivel de accesibilidad. El WCAG constituye un estándar de accesibilidad Web que satisface las necesidades de personas, organizaciones y gobiernos a nivel internacional, y explica cómo hacer que el contenido Web sea más accesible para todas las personas.

Tradicionalmente, estos criterios han sido evaluados por expertos [10], pero en los últimos años se han propuesto y desarrollado herramientas que permiten evaluarlos automáticamente. No obstante, los resultados de dichas evaluaciones dependen de la herramienta utilizada y sus características [10], [7].

1.3.1. La accesibilidad Web

La accesibilidad Web también puede beneficiar a personas sin condición de discapacidad, pero que utilizan teléfonos móviles, relojes y televisores inteligentes y otros

dispositivos con pantallas pequeñas y diferentes modos de entrada. También puede ayudar a personas mayores con capacidades cambiantes debido al envejecimiento, con un brazo roto, anteojos perdidos, o con limitaciones situacionales (como a la luz del sol o en un entorno donde no pueden escuchar el audio). Además, a personas que utilizan una conexión a Internet lenta o que tienen un ancho de banda limitado [9]. La accesibilidad Web depende de que varios componentes trabajen juntos, incluidas las tecnologías y los navegadores Web y otros agentes de usuario, herramientas de creación y sitios Web.

El compromiso del World Wide Web Consortium (W3C) de dirigir la web a su máximo potencial incluye promover un alto grado de usabilidad para las personas con discapacidad. La Iniciativa de Accesibilidad Web (WAI) es una iniciativa del W3C, que desarrolla su trabajo a través del proceso basado en el consenso del W3C, involucrando a diferentes partes interesadas en la accesibilidad web. Estos incluyen la industria, las organizaciones de discapacidad, el Gobierno, las organizaciones de investigación de accesibilidad y más. La WAI desarrolla especificaciones técnicas, pautas, técnicas y recursos de apoyo que describen soluciones de accesibilidad. Consideran estándares internacionales para la accesibilidad Web; por ejemplo, WCAG 2.0 también es una ISO estándar: ISO / IEC 40500 (International Electrotechnical Commission) [11].

1.3.2. Estándar WCAG 2.0 y 2.1

El estándar WCAG 2.0 [12], se publicó el 11 de diciembre del 2008. WCAG 2.1 se publicó el 5 de junio del 2018. Todos los requisitos (criterios de conformidad) de la versión 2.0 están incluidos en 2.1. Un sitio Web que cumpla con WCAG 2.1 cumpliría con los requisitos de las políticas que hacen referencia a WCAG 2.0. Dicho de otra forma: si se desea cumplir tanto con WCAG 2.0 y 2.1, se pueden usar los recursos de 2.1 y no sería necesario consultar los relativos a 2.0.

WCAG 2.0 está destinado principalmente a contenido Web, con material de apoyo para desarrolladores, evaluadores de accesibilidad Web y otros que requieren información estándar relacionada para la accesibilidad Web. Incluye 12 pautas organizadas en cuatro principios [12], que sientan las bases necesarias para que cualquiera pueda acceder y utilizar el contenido Web [13]. Los cuatro principios son los siguientes:

- *Principio perceptible*: la información y los componentes de la interfaz de usuario deben ser presentables a los usuarios de manera que puedan percibirlos. considera los tres sentidos principales necesarios para percibir contenido Web: vista, audición y tacto. Este principio tiene cuatro pautas: 1.1 proporciona alternativas de texto para cualquier contenido que no sea texto, que puede sustituirse por otras formas que las personas requieren, como letra grande, braille, audio, símbolos o un lenguaje simple; 1.2 proporciona alternativas para los medios en función del tiempo; 1.3 crea contenido que se puede presentar de diferentes maneras sin perder información o estructura; y 1.4 facilita a los usuarios ver y escuchar contenido, incluida la separación en primer plano y en segundo plano. Este principio incluye 22 criterios de éxito
- *Principio operable*: los componentes de la interfaz de usuario y la navegación deben ser operables. Significa que los usuarios deben poder operar la interfaz (la interfaz no puede requerir interacción que un usuario final no puede realizar). Comprende cuatro pautas: 2.1 hace que toda la funcionalidad esté disponible desde un teclado; 2.2 proporciona a los usuarios tiempo suficiente para leer y usar contenido; 2.3 aborda el diseño de contenido para evitar la incautación de ataques; y 2.4 proporciona formas de ayudar a los usuarios a navegar, encontrar contenido y determinar dónde se encuentran en un sitio. Incluye 20 criterios de éxito.
- *Principio comprensible*: la información y el funcionamiento de la interfaz de usuario deben ser comprensibles. Esto significa que los usuarios deben poder comprender la información, así como el funcionamiento de la interfaz de usuario (el contenido o la operación no puede estar más allá de su comprensión). Incluye tres pautas: 3.1 discute, hace que el contenido de texto sea legible y comprensible; 3.2 discute cómo hacer que las páginas Web aparezcan y funcionen de manera predecible; y 3.3 ayuda a los usuarios a evitar y corregir errores. Lo conforman 17 criterios de éxito.
- *Principio robusto*: el contenido debe ser lo suficientemente robusto para que pueda ser interpretado de manera confiable por una amplia variedad de agentes de usuario, incluidas las tecnologías de asistencia. Los usuarios deben poder acceder al contenido a medida que avanzan las tecnologías (a medida

que las tecnologías y los agentes de usuario evolucionan, el contenido debe permanecer accesible). Este criterio contempla dos criterios de éxito [13].

Los cuatro principios se distribuyen en 12 pautas o lineamientos que contemplan aspectos particulares de cada principio. Éstas recogen los objetivos básicos que se han de perseguir al crear contenidos accesibles para todos los usuarios, y sirven de marco general para la comprensión de los criterios y técnicas [13].

Las pautas incluyen 61 criterios de éxito, organizados de acuerdo con tres niveles de conformidad detallados a continuación:

- *Nivel A*: (el nivel mínimo de conformidad), la página Web satisface todos los criterios de éxito del Nivel A, o se proporciona una versión alternativa conforme. Conformado por 25 criterios.
- *Nivel AA*: la página Web satisface todos los criterios de éxito de Nivel A y Nivel AA, o se proporciona una versión alternativa conforme al Nivel AA. El cual está conformado por 13 criterios.
- *Nivel AAA*: la página Web satisface todos los criterios de éxito de Nivel A, Nivel AA y Nivel AAA, o se proporciona una versión alternativa conforme al Nivel AAA [14]. Conformado por 23 criterios.

La Lista de verificación de las pautas de accesibilidad al contenido Web 2.0 proporciona una referencia rápida y una descripción general de la información. A continuación, se presenta la lista con los criterios de éxito correspondientes a cada pauta, con la indicación del nivel al que corresponden según el estándar WCAG[14].

Lista de verificación WCAG 2.0 Nivel A (principiante)

1.1.1 - Contenido no textual: todo contenido no textual que se presenta al usuario cuenta con un texto alternativo que sirve para un propósito equivalente.

1.2.1 - Solo audio y solo video (pregrabado): proporcionar una alternativa al contenido de solo video y solo de audio.

1.2.2 - Subtítulos (pregrabados): proporcionar subtítulos para videos con audio.

1.2.3 - Descripción de audio o alternativa de medios (pregrabada): el video con audio tiene una segunda alternativa.

1.3.1 - Información y relaciones: estructura lógica.

1.3.2 - Secuencia significativa: presentar contenido en un orden significativo.

1.3.3 - Características sensoriales: usar más de un sentido para las instrucciones.

1.4.1 - Uso del color: no usar presentaciones que se basan únicamente en el color.

1.4.2 - Control de audio: no reproducir audio automáticamente.

2.1.1 - Teclado: accesible solo por teclado.

2.1.2 - Sin trampa de teclado: no atrapar a los usuarios del teclado.

2.2.1 - Tiempo ajustable: los límites de tiempo tienen controles de usuario.

2.2.2 - Pausa, detener, ocultar: proporcionar controles de usuario para mover contenido.

2.3.1 - Tres flashes o menos: ningún contenido parpadea más de tres veces por segundo.

2.4.1 - Bloques de derivación: proporcionar un enlace "Saltar al contenido".

2.4.2 - Página titulada: usar títulos de página útiles y claros.

2.4.3 - Orden de enfoque: orden lógico.

2.4.4 - Propósito del enlace (en contexto): el propósito de cada enlace es claro por su contexto.

3.1.1 - Idioma de la página: la página tiene un idioma asignado.

3.2.1 - En foco: los elementos no cambian cuando reciben foco.

3.2.2 - En entrada: los elementos no cambian cuando reciben información.

3.3.1 - Identificación de errores: identificar claramente los errores de entrada.

3.3.2 - Etiquetas o instrucciones: etiquetar elementos y dar instrucciones.

4.1.1 - Procesamiento: no hay errores importantes de código.

4.1.2 - Nombre, función, valor: construir todos los elementos para accesibilidad.

Lista de verificación WCAG 2.0 Nivel AA (intermedio)

1.2.4 - Subtítulos (en vivo): los videos en vivo tienen subtítulos

1.2.5 - Descripción del audio (pregrabado): los usuarios tienen acceso a la descripción de audio para el contenido de video

1.4.3 - Contraste (mínimo): la relación de contraste entre el texto y el fondo es de al menos 4.5: 1

1.4.4 - Cambiar el tamaño del texto: el texto puede ser redimensionado al 200 % sin pérdida de contenido o función.

1.4.5 - Imágenes de texto: no usar imágenes de texto.

2.4.5 - Múltiples formas: ofrezca varias formas de encontrar páginas.

2.4.6 - Encabezados y etiquetas: usar encabezados y etiquetas transparentes.

2.4.7 - Foco visible: asegúrese de que el foco del teclado sea visible y claro.

3.1.2 - Lenguaje de partes: indicar a los usuarios cuándo cambia el idioma de una página.

3.2.3 - Navegación consistente: usar menús consistentemente.

3.2.4 - Identificación consistente: usar íconos y botones consistentemente

3.3.3 - Sugerencia de error: sugerir correcciones cuando los usuarios cometen errores.

3.3.4- Prevención de errores (legales, financieros, datos): reducir el riesgo de errores de entrada para datos confidenciales.

Lista de verificación WCAG 2.0 Nivel AAA (avanzado)

1.2.6 - Lenguaje de señas (pregrabado): proporcionar traducciones en lenguaje de señas para videos.

1.2.7 - Descripción de audio extendida (pregrabada): proporcionar una descripción de audio extendida para videos.

1.2.8 - Alternativa de medios (pregrabada): proporcionar una alternativa de texto a los videos.

1.2.9 - Solo audio (en vivo): proporcionar alternativas para audio en vivo.

1.4.6 - Contraste (mejorado): la relación de contraste entre el texto y el fondo es de, al menos 7: 1.

1.4.7 - Audio de fondo bajo o nulo: el audio es claro para que los oyentes lo escuchen.

1.4.8 - Presentación visual: ofrecer a los usuarios una variedad de opciones de presentación.

1.4.9 - Imágenes de texto (sin excepción): no usar imágenes de texto.

2.1.3 - Teclado (sin excepción): accesible solo por teclado, sin excepción.

2.2.3 - Sin sincronización: sin límites de tiempo.

2.2.4 - Interrupciones: no interrumpir a los usuarios.

2.2.5 - Volver a autenticar: guardar los datos del usuario al volver a autenticarse.

2.3.2 - Tres flashes: ningún contenido parpadea más de tres veces por segundo

2.4.8 - Ubicación: informar a los usuarios dónde están.

2.4.9 - Propósito del enlace (solo enlace): el propósito de cada enlace es claro a partir de su texto.

2.4.10 - Encabezados de sección: dividir el contenido con encabezados.

3.1.3 - Palabras inusuales: explicar cualquier palabra extraña.

3.1.4 - Abreviaturas: explicar cualquier abreviatura.

3.1.5 - Nivel de lectura: los usuarios con nueve años de escuela pueden leer su contenido.

3.1.6 - Pronunciación: explicar cualquier palabra que sea difícil de pronunciar.

3.2.5 - Cambio por solicitud: no cambiar elementos en su sitio Web hasta que los usuarios pregunten.

3.3.5 - Ayuda: proporcionar ayuda detallada e instrucciones.

3.3.6 - Prevención de errores (todos): reducir el riesgo de todos los errores de entrada.

En la Figura 1.1 se detalla cómo está distribuido y conformado el contenido del Estándar WCAG 2.0 [15]. En esta se muestran círculos de color azul que representan los cuatro principios, los círculos color naranja claro describen los lineamientos, y los rectángulos representan cada uno de los criterios de éxito, además de aclarar al nivel que pertenece, ya sea nivel A, AA o AAA.

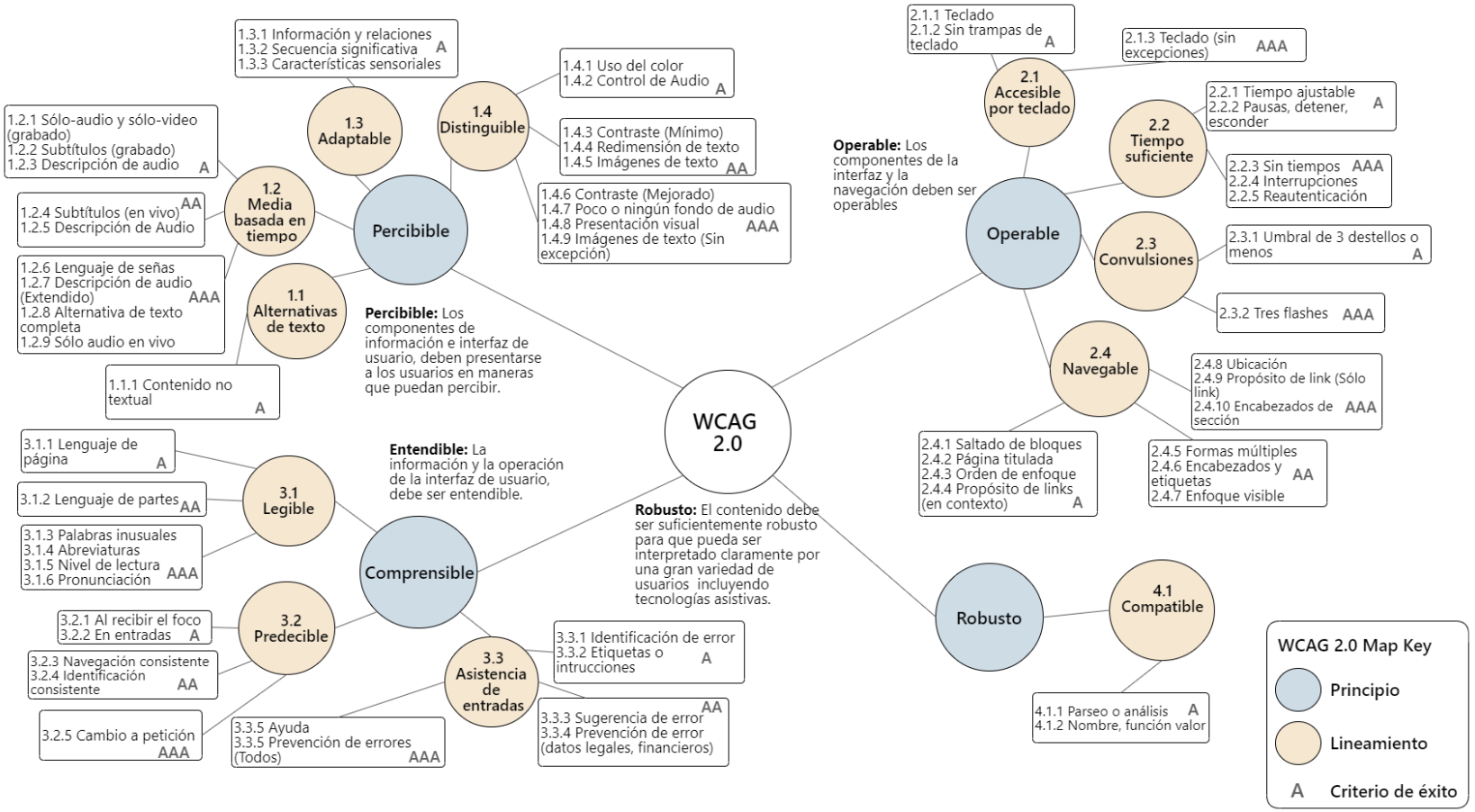


Figura 1.1: Detalle de cómo está conformado el Estándar WCAG [16].

1.3.3. Herramientas de evaluación de accesibilidad Web

Las herramientas de evaluación de accesibilidad Web son programas de software o servicios en línea que ayudan a determinar si el contenido Web cumple con las pautas de accesibilidad. Para el desarrollo de sitios Web es deseable contar con herramientas automatizadas que permitan la evaluación de la accesibilidad Web. Sin embargo, estas herramientas pueden tener limitaciones en cuanto a los criterios que plantean los estándares de accesibilidad Web [7]. Más aún, existen distintos instrumentos de evaluación de accesibilidad Web para evaluar un sitio Web según lo indicado por las normas WCAG. Es importante notar que actualmente, todavía se trabaja en definiciones de accesibilidad web que incluyen los componentes clave considerados por los investigadores y los profesionales [8], por lo que este es un campo en continua evolución y donde las herramientas deben irse actualizando a las nuevas necesidades requeridas. Las herramientas para la evaluación automatizada de la accesibilidad Web pueden apoyar a los profesionales, sin embargo, es difícil obtener el listado de los errores que las identifican [7].

Las herramientas de evaluación de la accesibilidad Web pueden apoyar al desarrollador a realizar un diagnóstico de los problemas de accesibilidad, con base en los distintos niveles de conformidad [16].

Las herramientas de evaluación de accesibilidad Web permiten ayudar a identificar rápidamente posibles problemas de accesibilidad. Pueden ser usadas en todas las fases del diseño Web y el proceso de desarrollo. Las herramientas pueden proporcionar controles completamente automatizados y ayudarlo con la revisión manual. No todos los aspectos de accesibilidad automáticamente se logran verificar. Se requiere juicio humano. A veces, las herramientas de evaluación pueden producir resultados falsos o engañosos. Las herramientas de evaluación de accesibilidad Web no pueden determinar la accesibilidad, solo pueden ayudar a hacerlo [1].

El Consorcio Web [1], por su parte, mantiene listas de herramientas que pueden apoyar los procesos de evaluación de la accesibilidad Web, y proporciona una guía de selección de herramientas.

1.4. Trabajo relacionado

Varios investigadores han realizado estudios sobre sitios Web que utilizan herramientas de evaluación de accesibilidad, con el objetivo de que personas con algún tipo de condición de discapacidad, puedan tener acceso a servicios con menor dificultad. A continuación, se mencionan los aportes de artículos secundarios que fueron revisados en esta investigación.

Nagaraju et al. [17] revisaron un total de 43 publicaciones y estudiaron cómo se lleva a cabo la investigación sobre accesibilidad, qué sugerencias hicieron los autores y qué marcos de arquitectura o métodos sugirieron para evaluar la accesibilidad del contenido de un sitio Web. Analizaron cómo se evaluó la accesibilidad Web con base en las pautas de WCAG 2.0, y cuáles métodos o herramientas fueron utilizadas. Asimismo, compararon los resultados de las evaluaciones de accesibilidad para determinar sus resultados y problemas existentes. Los autores concluyen que es necesario realizar investigaciones bibliográficas a profundidad para conocer los distintos estudios que se han empleado; y a partir de ahí hacer una comparación y un análisis para conocer las barreras existentes.

Por su parte, Tollefsen et al. [7] mencionan que las herramientas de prueba automáticas para las Pautas de Accesibilidad al Contenido en la Web (WCAG) 2.0 son importantes para los expertos profesionales en accesibilidad; estas herramientas pueden detectar aproximadamente el 50 % de los criterios de éxito. Sin embargo, las reglas utilizadas para detectar errores a menudo no están bien documentadas, y aunque se dice que se verifica un criterio de éxito, esto no garantiza que la herramienta pueda detectar todas las posibles violaciones de este criterio. Por lo tanto, una vez identificadas las herramientas han comenzado a crear un recurso para probar, no para encontrar el mejor producto de prueba, sino para obtener el conocimiento necesario sobre las herramientas que se utilizan en los proyectos de desarrollo y evaluaciones WCAG.

Kirchner [18] indica que uno de los principales problemas durante las evaluaciones de accesibilidad surge cuando se necesitan navegadores que soporten las extensiones requeridas. De lo anterior se concluye que, las herramientas utilizadas se probaron en páginas típicamente bien hechas, por lo tanto, su evaluación sigue siendo

muy subjetiva y cualitativa. Para evaluar completamente las herramientas y verificar su comportamiento con los errores más comunes se necesitan páginas de pruebas especiales, lo que arroja una evaluación cuantitativa y objetiva. Además, el autor concluyó que se ha demostrado la existencia de varias herramientas capaces de verificar la accesibilidad de las personas con discapacidad en las páginas Web, y algunas de estas pueden reparar páginas no accesibles. También, concluyen que los desarrolladores Web deben estar capacitados para hacer páginas accesibles, a fin de permitir que las personas con discapacidad tengan un fácil acceso a la red..

Por otro lado, Luque et al. [10] evalúan la cobertura de automatización para el WCAG por parte de algunas herramientas y describen sus debilidades y diferencias. Realizan una comparación entre accesibilidad Web y herramientas de evaluación. Concluyen que por no estar las herramientas normalizadas, se pueden encontrar fácilmente diferentes resultados para una página según la herramienta que se esté utilizando. Además, consideran que la normalización también es un enfoque importante, para no tener múltiples conjuntos de reglas de accesibilidad específicas del país en que nos encontremos y en implementaciones futuras. Siempre que no exista una interpretación formalizada de estas reglas, se producirán diferentes interpretaciones erróneas de los estándares WCAG.

El aporte que brinda el presente mapeo sistemático es el análisis sobre las herramientas de evaluación de accesibilidad Web que reporta la literatura, también lo es la identificación de los criterios de accesibilidad que se reportan y además, los desafíos presentes en el tema de accesibilidad Web. El análisis de las herramientas utilizadas para la evaluación de accesibilidad Web, puede permitir crear conocimiento sobre el tema y construir oportunidades de mejora sobre los desafíos que se reportan en cuanto a la accesibilidad. Además, con dicho aporte se podrá distinguir cuáles herramientas son las más utilizadas y así poder tener una herramienta capaz de mejorar los desarrollos Web accesibles.

1.5. Metodología

Para alcanzar los objetivos de la investigación, se realizó un mapeo sistemático de literatura, con base en los lineamientos de Petersen para mapeos [1] y las recomen-

daciones planteadas en [11], [21].

A continuación, se describe de manera detallada el proceso seguido para la búsqueda de información, la selección de estudios y su análisis.

1.5.1. Objetivo

El objetivo de este estudio formulado con el modelo GQM (Goal Question Metric) [3] es *analizar* las herramientas para la evaluación de la accesibilidad Web, *con el propósito de* caracterizarlas *con respecto a* los criterios evaluados para el cumplimiento del estándar de accesibilidad WCAG y los desafíos reportados *desde el punto de vista de* la investigadora *en el contexto de* evaluaciones de sitios Web.

1.5.2. Preguntas de investigación

Para guiar el diseño del estudio se establecieron las siguientes preguntas:

RQ1. ¿Cuáles herramientas se han utilizado para evaluar la accesibilidad de sitios Web?
Con las respuestas a esta pregunta es posible identificar las herramientas que se utilizan para evaluar la accesibilidad Web, analizar el porqué de su utilización y ver la constancia a través de los años.

RQ2. ¿Cuáles criterios de accesibilidad se han reportado en las evaluaciones realizadas con las herramientas?
Con esta respuesta se identificarán los criterios evaluados de accesibilidad Web basados en el estándar WCAG y el nivel de cumplimiento con respecto a los estándares de accesibilidad.

RQ3. ¿Cuáles son los desafíos reportados sobre las evaluaciones de la accesibilidad Web?
Con la respuesta de esta pregunta de investigación es posible determinar los desafíos que se presentan en el tema de la accesibilidad, las probables líneas de investigación futura y las recomendaciones que se pueden realizar con respecto al área de evaluación de accesibilidad Web.

1.5.3. Proceso de búsqueda

Para dar inicio con el proceso de búsqueda de los estudios, se realizó una exploración con el fin de identificar estudios de alta calidad que serían usados como artículos de control. La elección se basó en los siguientes criterios: el objetivo, las preguntas de investigación y términos utilizados en los estudios relacionados en el área.

Artículos de control A continuación se describen los artículos de control seleccionados [16], [22], [23] y [24].

Acosta et al. [24], se basan en tres enfoques: la investigación sobre los problemas de accesibilidad de los sitios Web de varias universidades de América Latina, la evaluación de accesibilidad Web de acuerdo con las Pautas 2.0, y un análisis de las relaciones entre los sitios Web universitarios. Estas clasificaciones fueron realizadas mediante el uso de varias herramientas de evaluación de la accesibilidad Web.

Elkabani et al. [16], hacen una introducción a accesibilidad Web; posteriormente valoran tres herramientas de evaluación, comparan los pros y contras y al final, se da una recomendación de cuál es la más apta.

Acosta et al. [22], presentan un análisis sobre la accesibilidad de los servicios interactivos de administración electrónica ofrecidos por dos entidades oficiales de los países de América Latina. La evaluación se llevó a cabo según el criterio de los expertos en accesibilidad Web y el uso de una herramienta de evaluación de accesibilidad Web. En ambos casos, se consideraron las Pautas de Accesibilidad al Contenido en la Web (WCAG) 2.0 propuestas por el World Wide Web Consortium (W3C).

Isa et al. [23], tienen como objetivo, investigar la accesibilidad de los sitios Web de Homestay, en Malasia. El estudio toma una muestra de 328 sitios, realizan la valoración con diferentes herramientas y al final proporcionan varias recomendaciones para mejorar los niveles de accesibilidad Web.

Cadena de búsqueda La cadena de búsqueda se construye, entonces, a partir de la extracción de términos clave del título y el resumen del conjunto de artículos de control. Se desarrolla el modelo PICO (Población, Intervención, Comparación, Salidas) [4].

Población: Sitios Web
Intervención: Accesibilidad
Comparación: No aplica
Salidas: Herramientas, criterios de accesibilidad, evaluación y desafíos.
El modelo PICO anterior da como resultado la siguiente cadena de búsqueda:

```
("accessibility" AND "tool*" AND ("W3C" OR "World_Wide_Web_
   Consortium" OR "WCAG*"))
```

La cadena de búsqueda es producto de un proceso de refinamiento mediante un conjunto de pruebas piloto para reducir el ruido, que consistió en determinar el orden de los elementos entre paréntesis para atacar el riesgo de que no salieran todos los artículos que iban a ser importantes y el uso de distintos paréntesis para cada uno de los cluster.

Bases de Datos Las búsquedas automatizadas se ejecutaron en las bases de datos SCOPUS [1] , IEEE Xplore [2], e ISI Web of Science [3].

Período de búsqueda El protocolo base del mapeo fue desarrollado durante el primer semestre del 2019. La búsqueda automatizada se realizó en junio del 2019 y los estudios se analizaron durante el segundo semestre del 2019. El número de estudios por base de datos fue el siguiente: Scopus (229 resultados), IEEE Xplore (43 resultados) y Web of Science (27 resultados). Los artículos por base de datos fueron tabulados en MS Excel para el subsecuente proceso de selección, y eliminación de duplicados. Se eliminó un total de 56 artículos duplicados. La Figura 1.2 muestra el detalle del proceso.

1.5.4. Proceso de selección de estudios

Para determinar si los artículos que generaron las búsquedas eran relevantes para la investigación, se definieron criterios de inclusión y de exclusión. El proceso de inclusión y extracción se realizó basado en el título, resumen y palabras clave y, en

[1] https://www2.scopus.com/home.uri
[2] http://ieeexplore.ieee.org/
[3] apps.webofknowledge.com

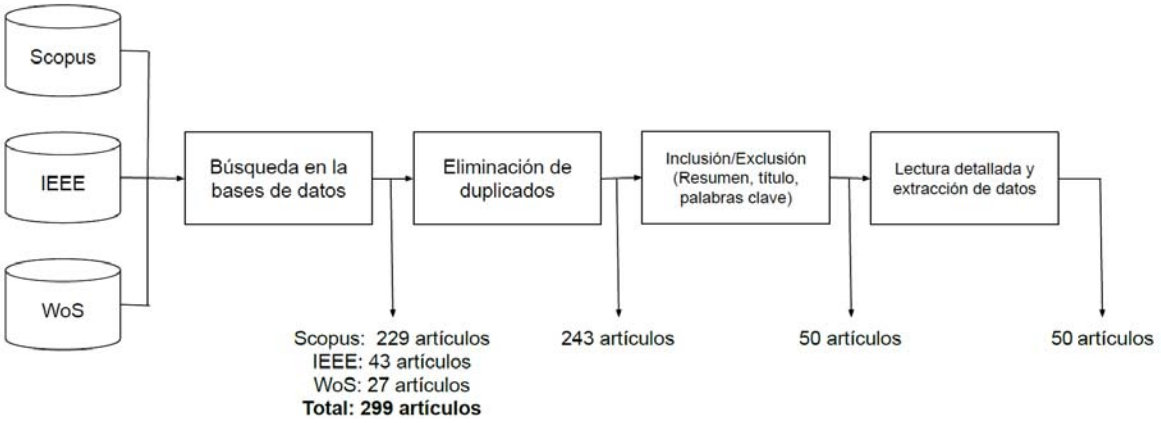


Figura 1.2: Proceso de selección de estudios.

caso de duda, se hizo la lectura del texto completo. Se excluyeron los estudios que cumplían con la fórmula (E1 OR E2) y se incluyeron los que cumplían con la fórmula (I1 AND I2 AND I3).

Inclusión:

I1. Artículos escritos en idioma inglés.

I2. Artículos del área de ingeniería del software.

I3. Artículos que evalúen un sitio Web con una herramienta.

Exclusión:

E1. Artículos no disponibles en texto completo.

E2. Artículos secundarios y terciarios.

Después de aplicar el proceso de selección (inclusión y exclusión), se obtuvieron 50 artículos que pasaron al proceso de extracción y análisis de resultados, tal como se muestra en la Figura 1.2.

1.5.5. Evaluación de la calidad

La evaluación de la calidad de los estudios seleccionados se realiza para determinar el nivel de detalle ofrecido por estos, en aspectos de interés para el análisis. La puntuación final de calidad de un trabajo primario indica el nivel de confianza en los hallazgos encontrados. Los criterios de calidad que se usaron fueron los siguientes:

El puntaje se asignó en una escala de 0 a 2, donde 0 = No cumple el criterio en lo absoluto, 1 = Cumple con el criterio parcialmente, 2 = Cumple con el criterio totalmente. Los estudios fueron evaluados de acuerdo con los siguientes criterios de calidad: (Q1) ¿El objetivo de investigación está claramente definido? (Q2) ¿ El estudio contempla una descripción en los resultados de los incumplimientos presentados en la evaluación? (Q3) ¿Las herramientas utilizadas están descritas y su elección justificada?

Los valores de calidad obtenidos por los estudios variaron entre 0 y 6, con una mediana de 5 y un promedio de 4,78, así, los estudios analizados proporcionan un nivel de detalle deseable de acuerdo con los criterios de calidad. La Figura 1.3 muestra los valores de evaluación de calidad para los artículos incluidos.

Revisando de forma más detallada el promedio obtenido por cada uno de los tres criterios de calidad evaluados, el Q1 dio como resultado 2, por tanto, se puede determinar que todos los estudios cuentan con un objetivo claro de la investigación. Para el Q2, se obtuvo un promedio de 1.48, se puede mencionar que un porcentaje alto de los estudios presentan una descripción en los resultados de los incumplimientos presentados en la evaluación. Este es un punto por considerar ya que puede ser importante para próximos estudios en el área. Con respecto al Q3, el promedio obtenido fue de 1.3, esto significa que las herramientas utilizadas casi en la mitad de los estudios, no están descritas y tampoco los autores dieron una explicación de su elección.

En el anexo de la sección 1.B, se muestra detalladamente los resultados de las preguntas de calidad para cada estudio.

1.5.6. Extracción de datos

Para cada estudio seleccionado, se extrajo la información relevante para el análisis de las preguntas de investigación. Los componentes del formulario de extracción

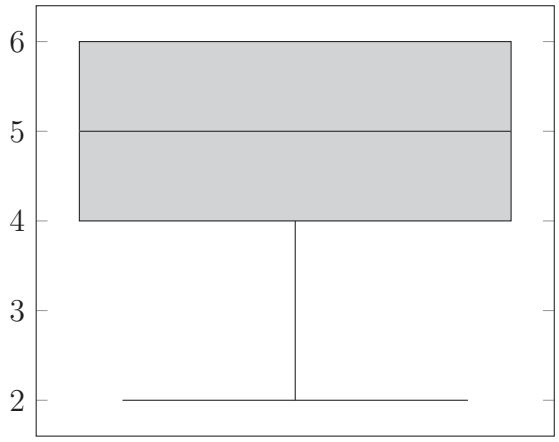


Figura 1.3: Calidad de los estudios primarios incluidos.

asociados con cada pregunta incluyeron información de las herramientas (RQ1), criterios de accesibilidad evaluados (RQ2), y desafíos reportados (RQ3). Para el análisis y síntesis de la información, se tabularon los resultados por pregunta y se usó un análisis narrativo que resumía y describía los hallazgos y la evidencia extraída. El formulario fue organizado en las secciones siguientes: datos generales y las preguntas de investigación. Los componentes del formulario de extracción se muestran en el Cuadro 1.1.

Cuadro 1.1: Componentes del formulario de extracción.

Categoría	Componentes
Generales	Id, Referencia, Título, Autores, Año, Número de páginas, Tipo de documento, Clasificación, Organización y ambiente, Base de datos, Tipo de artículo, Q1, Q2, Q3, Resultado Calidad.

Continúa en la página siguiente.

Categoría	Componentes
Información de las herramientas (RQ1)	Nombre de la herramienta, Licencia comercial o libre, Nueva herramienta o extensión, Alcance funcional.
Criterios de accesibilidad (RQ2)	Criterios Nivel A, Criterios Nivel AA, Criterios Nivel AAA del estándar WCAG
Desafíos reportados (RQ3)	Desafíos.

1.5.7. Análisis de datos

Para el diagnóstico y síntesis de la información se tabularon los resultados por pregunta y se utilizó el análisis narrativo que resume y describe los hallazgos y la evidencia extraída.

Para la RQ1, el análisis consistió en identificar aquellas herramientas de evaluación de accesibilidad reportados en los estudios y determinar si era posible obtener una descripción de las características de la herramienta.

En cuanto a la RQ2, la investigación se centró en extraer la información referente a los criterios de accesibilidad que fueron analizados por las herramientas de evaluación de accesibilidad Web, según los estudios.

La RQ3, permitió identificar los desafíos reportados en las evaluaciones de accesibilidad. Dicha información fue obtenida por medio de los componentes del formulario de extracción.

1.5.8. Amenazas a la validez

Las amenazas a la validez ayudan a determinar las fortalezas y las limitaciones del estudio con respecto a la validez de los resultados obtenidos. A continuación, se

discuten las amenazas presentes durante el mapeo de literatura.

Selección de la cadena de búsqueda y las bibliotecas digitales: Con respecto a la cadena de búsqueda, fue definida a partir de un conjunto de artículos de control y piloteada en varias pruebas para reducir el ruido. La selección de la palabra clave WCAG en la cadena de búsqueda pudo haber sesgado los estándares identificados, sin embargo, para escogerla se realizó una búsqueda exploratoria de artículos que arrojó que este era el estándar internacional más aplicado. De igual forma para mitigar el sesgo, fueron tomados en cuenta los estándares que salieron en los estudios analizados y que no eran WCAG. Las bases de datos seleccionadas (SCOPUS, IEEE Explore y Web of Science) son reconocidas por tener gran cobertura en el área de ingeniería del software.

Identificación de estudios primarios: Cuando hubo duda sobre la inclusión de un artículo se realizó la lectura completa del mismo. Se excluyó literatura gris y artículos que no estaban en idioma inglés. Dado que las búsquedas se realizaron en las bases de datos referenciales mostradas en la Figura 1.2 y la cadena de búsqueda fue en inglés, solo se contemplaron artículos en ese idioma. Durante la búsqueda se identificaron cinco artículos en español y dos en portugués, que podrían ser de interés para el lector: [26, 27, 28, 29, 30, 31, 32] y ser analizados como trabajo futuro, ya que sería importante considerar estudios que hayan evaluado sitios Web en español, del contexto regional.

Extracción y clasificación de artículos primarios: Se diseñó un formulario de extracción para la recolección de datos, como guía del proceso. El procedimiento clasificación y extracción, y la aplicación de los criterios de calidad fueron realizados solo por la investigadora y para mitigar el sesgo de información después de la extracción de los datos, se revisó nuevamente el formulario de extracción para determinar su completitud, la existencia de inconsistencias y, finalmente, verificar y depurar la información extraída.

Cada vez que se detectó una inconsistencia, la investigadora revisó el artículo en detalle para determinar la correctitud de la información extraída. Los artículos son clasificados de acuerdo con lo reportado por los autores originales y, en caso de no hacerlo claramente, la investigadora realiza la clasificación según sus descripciones o según las herramientas utilizadas.

Generalización y síntesis de resultados: la generalización y síntesis de resultados se limita a los estudios incluidos en el mapeo. Para minimizar el sesgo al presentar los resultados, toda la investigación se realizó siguiendo los protocolos previamente definidos y validados durante el procedimiento. Se reporta el proceso para facilitar el análisis y la replicación.

Consulta a expertos: para esta investigación se hicieron solo consultas generales a personas que laboran en el área de diseño web y que abordan el tema de accesibilidad dentro de la Universidad de Costa Rica. Además, el proceso de elección de los artículos de control, la búsqueda exhaustiva en las bases de datos garantizó que los datos que se obtuvieron fueran de fuentes confiables.

1.6. Análisis de resultados

En esta sección se presentan los resultados del mapeo. Los 50 estudios primarios y las preguntas de investigación fueron analizados y clasificados en diferentes dominios, entre los que están: sitios Web del Gobierno 18 (36 %), universidades 14 (28 %), industria 4 (8 %), salud 2 (4 %) y otros que no reportaron el dominio 12 (24 %). Los estudios analizados fueron publicados entre los años 2004 y 2019. El año 2018 cuenta con la mayor cantidad de artículos (10), seguido por el 2016 (8), 2017 (7) y 2019 (6), lo que indica la vigencia de las publicaciones en el área. Finalmente, los estudios se realizaron en distintos países de América Latina, de África, de la India, también en Australia, Azerbaijan, Bangladesh, China, Chipre, Estados Unidos, Ghana, Indonesia, Japón, Libia, Kazakhstan, Kyrgyzstan, Malasia, Nepal, Noruega, Arabes, Pakistán, Portugal, Sri Lanka y Turquía, lo que indica que existe interés a nivel global por mejorar el tema de la accesibilidad de los sitios Web.

Las herramientas de evaluación de accesibilidad Web no pueden verificar todos los aspectos de accesibilidad de forma automática y se requiere del juicio humano para verificarlo, los estudios [24, 33, 34, 35, 36] evidencian lo que los autores determinaron.

En anexos, en la sección 1.A se puede encontrar la lista completa de los artículos analizados en el estudio con su respectivo id, título, año y referencia. En el enlace `https://tinyurl.com/y27x5cac`, se muestran los datos recolectados de cada

artículo primario que fue procesado mediante lectura completa para obtener la información que respondiera al objetivo y las preguntas de investigación.

A continuación, se presentan los resultados del mapeo sistemático de literatura que responden las preguntas de investigación

1.6.1. Herramientas para evaluar la accesibilidad de sitios Web (RQ1)

La primera pregunta de investigación ayuda a identificar cuáles herramientas son utilizadas para evaluar la accesibilidad Web. Se identificó un total de 38 herramientas que fueron reportadas por los autores en los estudios analizados. La Figura 1.4 muestra las 12 herramientas de evaluación de accesibilidad más utilizadas y su tendencia en el tiempo.

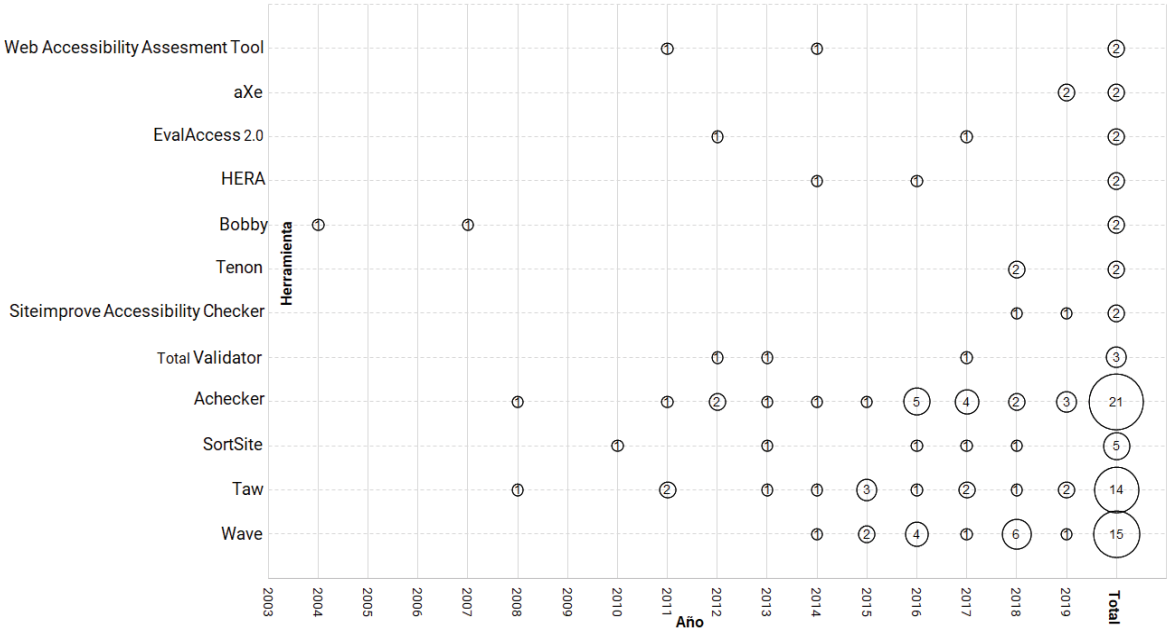


Figura 1.4: Cantidad de herramientas reportadas como utilizadas más de dos veces por año.

Desde el 2008, estas herramientas han sido regularmente reportadas en evaluaciones realizadas por los estudios de accesibilidad Web. En la Figura 1.4 se muestra que Achecker (https://achecker.ca/) fue la herramienta más utilizada con un

total de 21 estudios que la reportaron. Algunos estudios mencionan que su popularidad se debe a que es de licencia gratuita y online, y que utiliza una variedad de estándares de accesibilidad internacional. Le siguen las herramientas Wave (https://wave.webaim.org/) con 15 reportes , Taw (https://www.tawdis.net/) con 14, SortSite con 5, Total Validator con 3 y las herramientas que fueron reportadas solo dos veces: Siteimprove Accessibility Checker, Tenon, Bobby, HERA, EvalAccess 2.0, aXe y Web Accessibility Assessment Tool. Las herramientas como SortSite y Total Validator están muy por debajo del nivel de uso de las cuatro primeras, lo que podría deberse a que su licencia es de pago. Esto evidencia que las herramientas gratuitas tienden a ser las más utilizadas para evaluar los sitios Web. Un dato relevante que se obtuvo es que la herramienta Bobby fue utilizada en el 2004 y 2007, es decir, tiene varios años de no utilizarse.

Las herramientas reportadas por solo un artículo son: aDesigner[37], AMP [38], Checorser [39], Html Code Sniffer [40], CORrector [39], Cynthia Says [41], Deque [38], Evaluator 1.0.2 [42], google Mobile-Friendly Test [43], googlePage- Speed insight [43], Koa11y [35], Markup Validation Service [33], OpenWAX [36], EIII-Page Checker [44], Pingdom tool [43], PowerMapper [43], TinyMCE[45], Webpage [34], Analyzer [34], Worldspace FireEyse[46], AccessColor[47], Contrast-Finder[47], Tanaguru[47],AccessibilityToolbar[47], HTML Tidy[48], AC-TAW [48]. De esta lista, la gran mayoría de las herramientas las pueden encontrar en la dirección Web (https://www.w3.org/WAI/ER/tools/). Existen estudios que proponen nuevas herramientas a partir de la extensión de las existentes, tal es el caso de [16], [49] y [45].

En los 50 estudios analizados, la gran mayoría de los autores no realizan una descripción de la herramienta en cuanto al tipo de estándar que puede revisar y tampoco los criterios que evalúa. Por este motivo se tuvo que buscar esta información en el sitio oficial del Consorcio World Wide Web donde disponen de una amplia lista de herramientas de evaluación Web. En el apéndice 1.C, se listan todas las herramientas reportadas y una descripción de lo que evalúa cada una.

1.6.2. Criterios que se han reportado en las evaluaciones de accesibilidad (RQ2)

Esta pregunta permitió identificar, de acuerdo con el estándar WCAG, aquellos criterios que más se reportan como incumplidos en los estudios analizados. La Figura 1.5 muestra que de los 50 estudios analizados, 7 reportaron que evaluaron el criterio de conformidad Nivel A, 6 al criterio de conformidad Nivel AA y 29 al criterio Nivel AAA, además muestra la referencia del estudio que lo reportó. Se debe recordar que el criterio de conformidad Nivel A está compuesto por 25 aspectos, el criterio de conformidad Nivel AA conformado por los 25 del nivel A más 13 criterios propios y el AAA el cual acumula los 25 del A, más los 13 del AA y 23 nuevos criterios, por lo que este último es más complejo de evaluar. Del total de artículos, 40 evaluaron solo el estándar WCAG 2.0, 5 artículos evaluaron solo el estándar WCAG 1.0, 3 artículos evaluaron tanto el estándar WCAG 2.0 como Section 508, 2 artículos evaluaron ambos estándares WCAG 2.0 como el WCAG 1.0. Finalmente, 8 estudios no indicaron la evaluación de un estándar específico. De los estudios que realizaron una evaluación de accesibilidad, el 56 % (28) evaluaron los tres niveles de conformidad: A, AA, AAA.

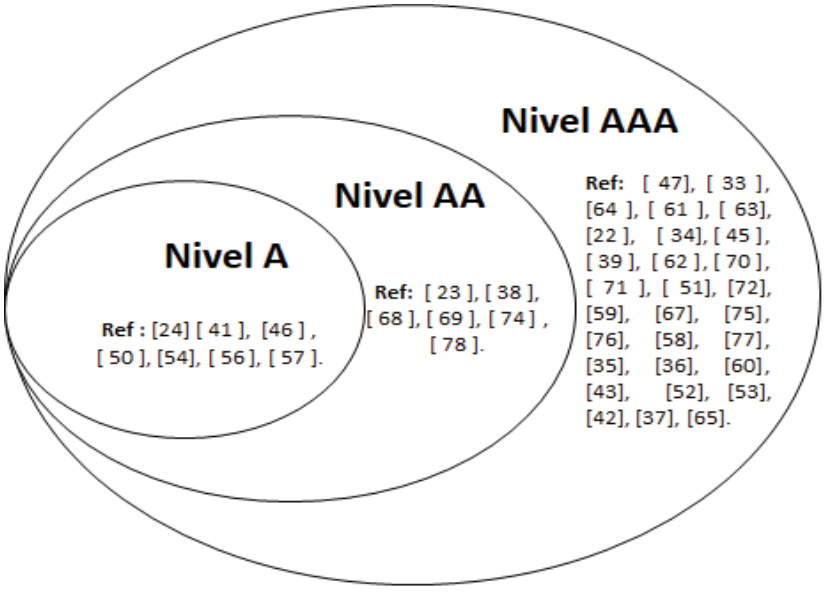


Figura 1.5: Artículos que evaluaron los diferentes niveles del estándar WCAG.

El Cuadro 1.2 muestra los criterios que no fueron reportados por los estudios. Otro

aspecto por considerar es que los criterios de accesibilidad no reportados podrían indicar que estos no pueden ser evaluados automáticamente y que aún requieren la evaluación por parte de un humano.

Cuadro 1.2: Criterios de accesibilidad no reportados en los tres niveles.

Nivel	Criterio
A	1.2.3 Audiodescripción o Medio alternativo (grabado) .
A	1.4.2 Control del audio o Medio alternativo..
AA	1.2.4 Subtítulos (en directo).
AA	1.2.5 Audio descripción (grabado).
AA	2.4.5 Múltiples vías.
AA	2.4.7 Foco visible.
AA	3.1.2 Idioma de las partes.
AA	3.2.4 Identificación coherente.
AA	3.3.3 Sugerencias ante errores.
AA	3.3.4 Prevención de errores (legales, financieros, datos).
AAA	1.2.6 Lenguaje de señas (grabado).
AAA	1.2.7 Audio descripción ampliada (grabada).
AAA	1.2.9 Sólo audio (en directo).
AAA	1.4.7 Sonido de fondo bajo o ausente.
AAA	2.2.3 Sin tiempo.
AAA	2.2.5 Re-autenticación.
AAA	2.3.2 Tres destellos.
AAA	2.4.8 Ubicación.
AAA	2.4.10 Encabezados de sección.
AAA	3.1.3 Palabras inusuales.
AAA	3.1.4 Abreviaturas.
AAA	3.1.5 Nivel de lectura.

Continúa en la página siguiente.

Nivel	Criterio
AAA	3.1.6 Pronunciación.
AAA	3.2.5 Cambios a petición.
AAA	3.3.5 Ayuda.
AAA	3.3.6 Prevención de errores (todos).

Los Cuadros 1.3, 1.4 y 1.5 listan los aspectos reportados por las herramientas de evaluación de accesibilidad. El 40 % de los artículos reportaron las inconformidades encontradas durante la ejecución de los estudios. Las agrupaciones de los aspectos de accesibilidad se recolectaron de acuerdo con los criterios de conformidad por cada uno de los niveles de éxito: A, AA, AAA. Los estudios analizados reportan cuáles son los criterios que más se incumplen; por esto, en los cuadros se hace la división por cada uno de los niveles de conformidad.

En los Cuadros 1.3, 1.4 y 1.5, se muestra el nivel evaluado, el aspecto que se incumplió, el estudio que lo reportó y la cantidad de veces que fue reportado. Otro punto por dejar claro, y como lo muestran los cuadros, es que los criterios que no se cumplieron y que están evidenciados, fueron los que el autor del estudio reportó que detectaron las herramientas utilizadas en las evaluaciones de accesibilidad, es decir, se revisaron todos los aspectos y los que más se incumplieron se reportan, tal como lo muestran los cuadros. Los resultados exponen conjuntos de criterios no cumplidos que se repiten en la mayoría de las evaluaciones, por lo que es bueno dar seguimiento a estas debilidades encontradas en los sitios Web.

Cuadro 1.3: Criterios más incumplidos reportados por los estudios en el Nivel A.

Nivel	Criterio	Estudios	Cantidad
A	1.1.1 Contenido no textual.	[24, 46, 50, 23, 38, 37, 51, 52, 53, 34, 54, 55, 56, 57, 58, 59, 36, 60, 49, 61, 22, 62, 63]	23

Continúa en la página siguiente.

Nivel	Criterio	Estudios	Cantidad
A	2.4.4 Propósito de los enlaces. (en contexto)	[46, 34, 24, 54, 22, 23, 50, 62, 55, 38, 52, 53, 37, 58, 56]	15
A	1.3.1 Información y relaciones.	[46, 64, 34, 54, 23, 50, 62, 55, 38, 51, 59, 36, 52]	13
A	3.1.1 Idioma de la página.	[46, 64, 34, 61, 23, 50, 62, 55, 51, 53, 56, 57]	12
A	3.3.2 Etiquetas o instrucciones.	[50, 22, 23, 34, 62, 55, 53, 58, 56]	9
A	4.1.1 Procesamiento.	[46, 34, 54, 55, 51, 53]	6
A	4.1.2 Nombre, función, valor.	[46, 64, 34, 55, 59]	5
A	1.4.1 Uso del color.	[34, 62, 51, 58]	4
A	2.4.1 Evitar bloques.	[46, 34, 51]	3
A	2.1.1 Teclado.	[46, 62, 51]	3
A	2.2.1 Tiempo ajustable.	[51, 34]	2
A	3.2.1 Al recibir el foco.	[51, 34]	2
A	1.3.3 Características sensoriales.	[51, 34]	2
A	2.3.1 Umbral de tres destellos o menos.	[51, 34]	2
A	3.3.1 Identificación de errores.	[50, 51]	2
A	2.4.3 Orden del foco.	[54, 37]	2
A	1.2.1 Solo audio y solo vídeo (grabado).	[51, 54]	2

Continúa en la página siguiente.

Nivel	Criterio	Estudios	Cantidad
A	1.2.2 Subtítulos (grabados).	[54, 58]	2
A	2.2.2 Poner en pausa, detener, ocultar.	[62, 46]	2
A	1.3.2 Secuencia significativa.	[54]	1
A	2.4.2 Titulado de páginas.	[55]	1
A	3.2.2 Al recibir entradas.	[55]	1
A	2.1.2 Sin trampas para el foco del teclado.	[54]	1

En la Figura 1.6 se muestra con detalle cada uno de los aspectos o criterios que reportan los autores como más incumplidos en las evaluaciones de accesibilidad Web, pertenecientes al Nivel A. Se puede notar que los incumplimientos se dan en los cuatro principios, sin embargo, en el operable es donde se reportan más aspectos incumplidos.

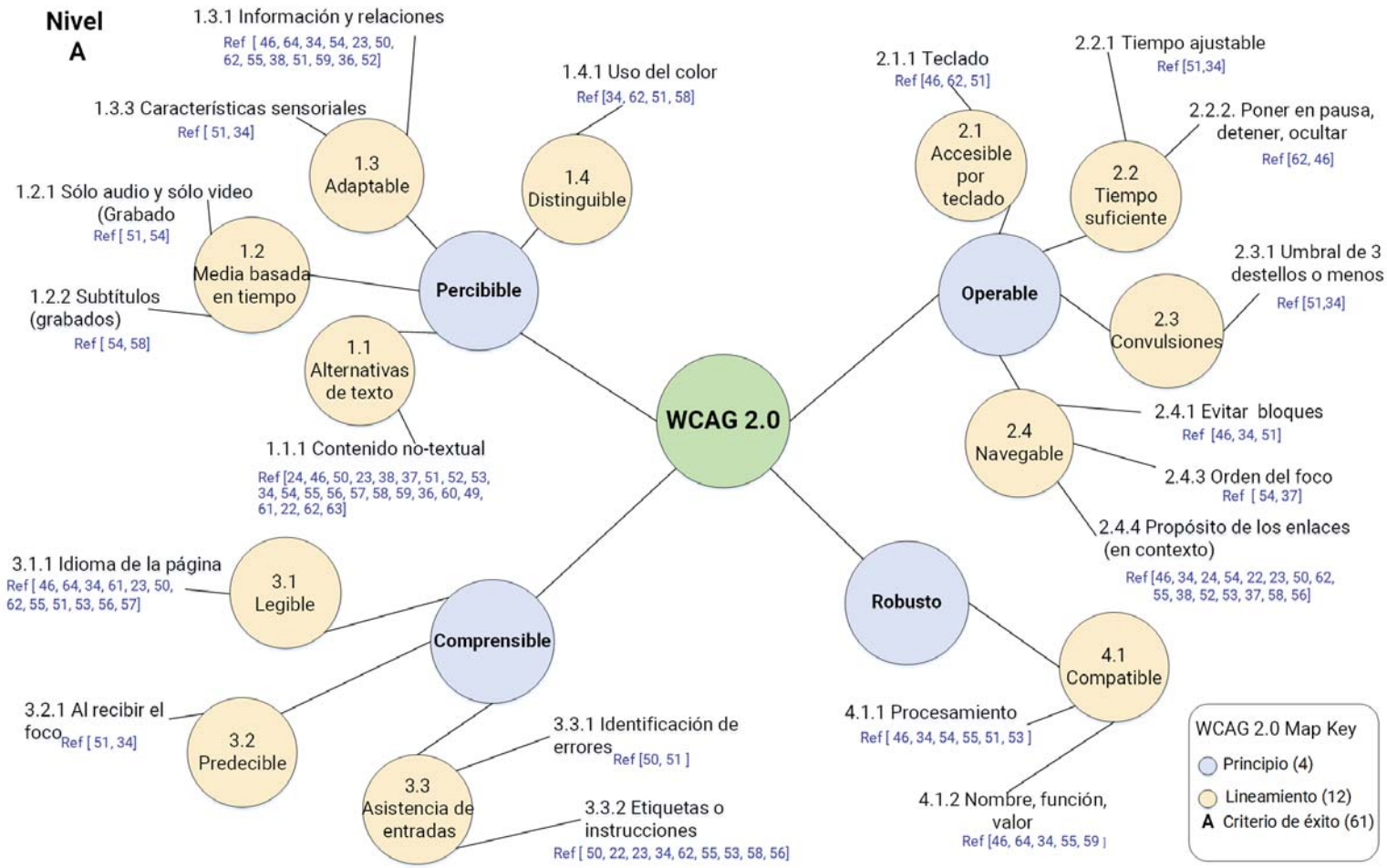


Figura 1.6: Criterios incumplidos reportados más de dos veces en el Nivel A

La Figura 1.7 muestra por año, la cantidad de aspectos que son incumplidos más de dos veces en el Nivel A, reportados por los estudios analizados que evaluaron la accesibilidad Web. Además, se puede observar que los 4 aspectos siguientes: el 1.1.1 Contenido no textual, el 2.4.4 Propósito de los enlaces (en contexto), el 1.3.1 Información y relaciones, y el 3.1.1 Idioma de la página, son los aspectos más reportados por los estudios como los que más se incumplen. Asimismo, que estos aspectos del 2016 al 2019 se reportan como incumplidos de forma continua. También, se puede observar que el aspecto 1.1.1 mostró un aumento en el 2018 comparado con los demás años.

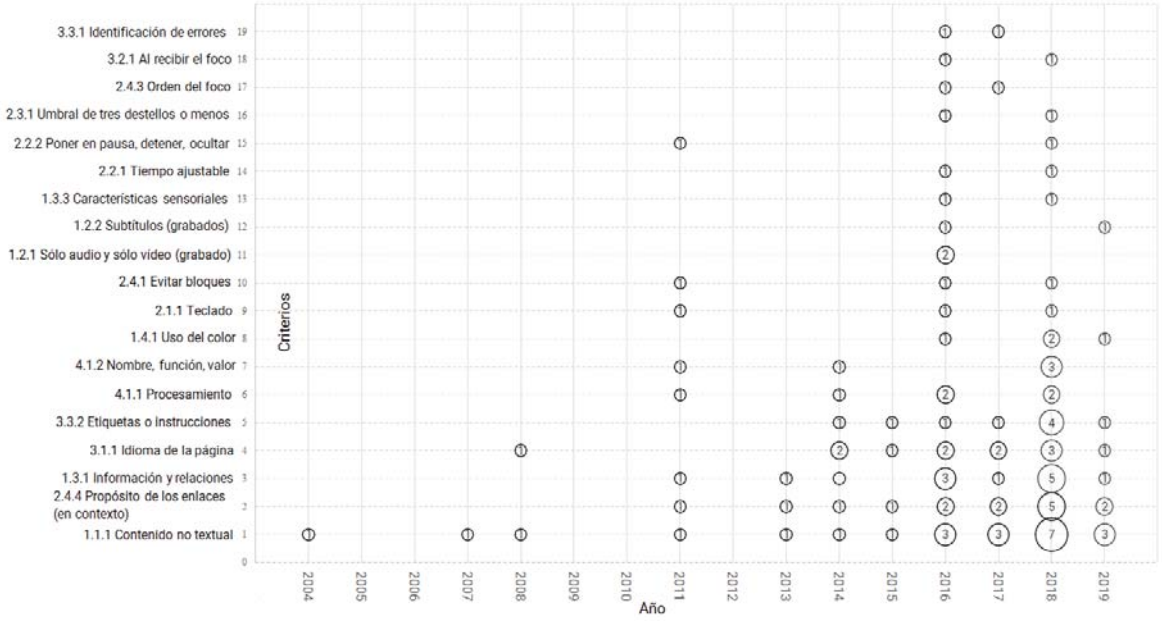


Figura 1.7: Cantidad de criterios incumplidos reportados más de dos veces en el Nivel A, por año.

Cuadro 1.4: Criterios más incumplidos reportados por los estudios en el Nivel AA.

Nivel	Criterio	Estudios	Cantidad
AA	1.4.3 Contraste (mínimo).	[50, 23, 38, 63]	4
AA	2.4.6 Encabezados y etiquetas.	[62, 37, 22]	3
AA	1.4.5 Imágenes de texto.	[65, 53, 22]	3
AA	1.4.4 Cambio de tamaño del texto.	[62, 38, 63]	3
AA	3.2.3 Navegación coherente.	[34]	1

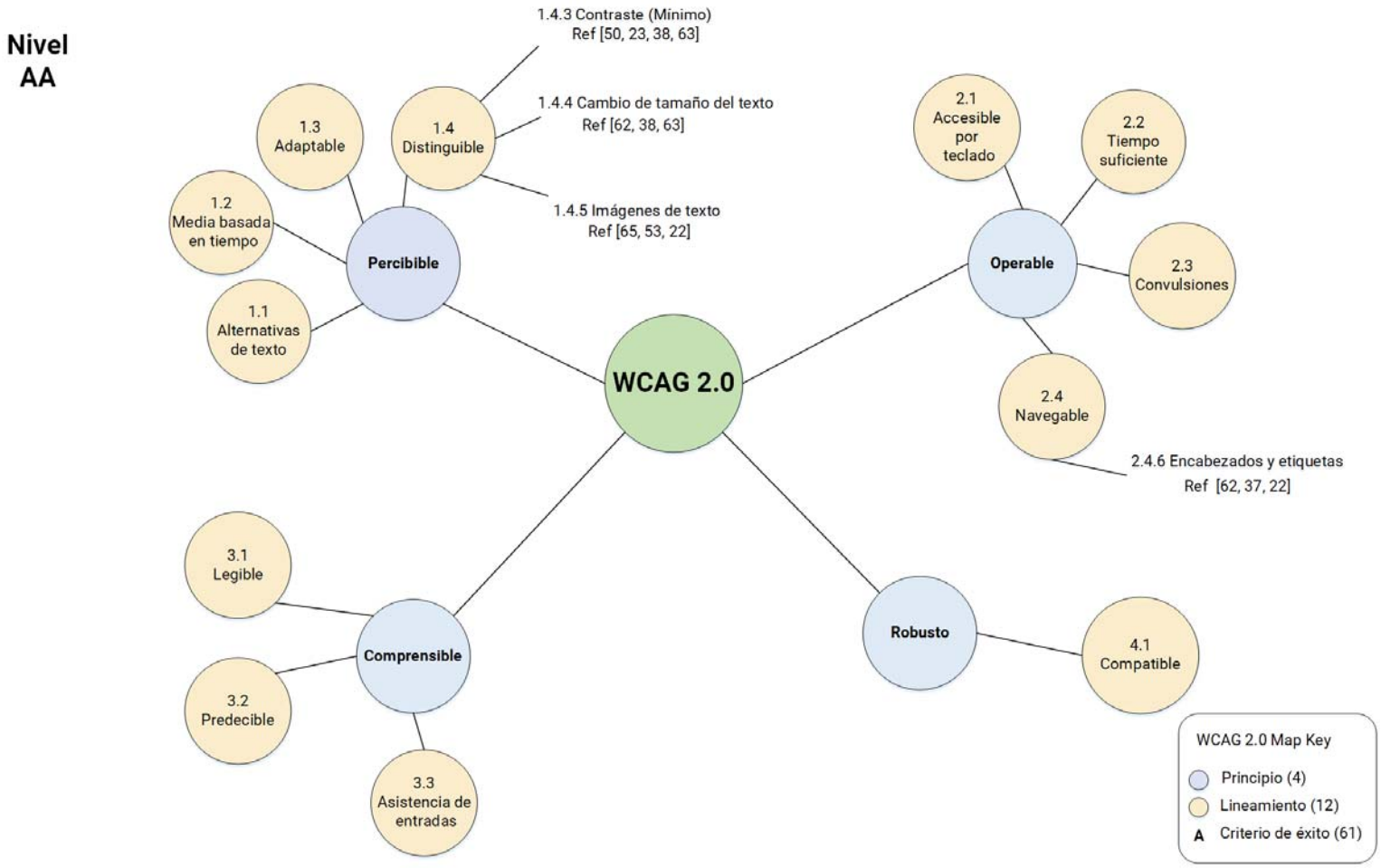


Figura 1.8: Criterios incumplidos reportados más de dos veces en el Nivel AA.

En la Figura 1.8 se puede observar que en el Nivel AA los incumplimientos reportados se dan en el principio percibible el cual contempla tres aspectos o criterios, y también se registra en el principio operable con un criterio. En los principios comprensible y robusto no se reportan incumplimientos.

Cuadro 1.5: Criterios más incumplidos reportados por los estudios en el Nivel AAA.

Nivel	Criterio	Estudios	Cantidad
AAA	2.4.9 Propósito de los enlaces (solo enlaces).	[65, 64, 22, 36, 37]	5
AAA	1.4.6 Contraste (mejorado).	[62, 16, 37]	3
AAA	1.4.9 Imágenes de texto (sin excepciones).	[65, 16, 56]	3
AAA	2.1.3 Teclado (sin excepciones).	[16, 37]	2
AAA	1.2.8 Medio alternativo (grabado).	[37]	1
AAA	1.4.8 Presentación visual.	[37]	1
AAA	2.2.4 Interrupciones.	[38]	1

En la Figura 1.9 se detallan los aspectos o criterios más incumplidos en el Nivel AAA. Se puede observar que en el principio percibible y en el operable es donde se dan dichos incumplimientos. En esta figura únicamente se detallan los que fueron reportados más de dos veces.

Nivel AAA

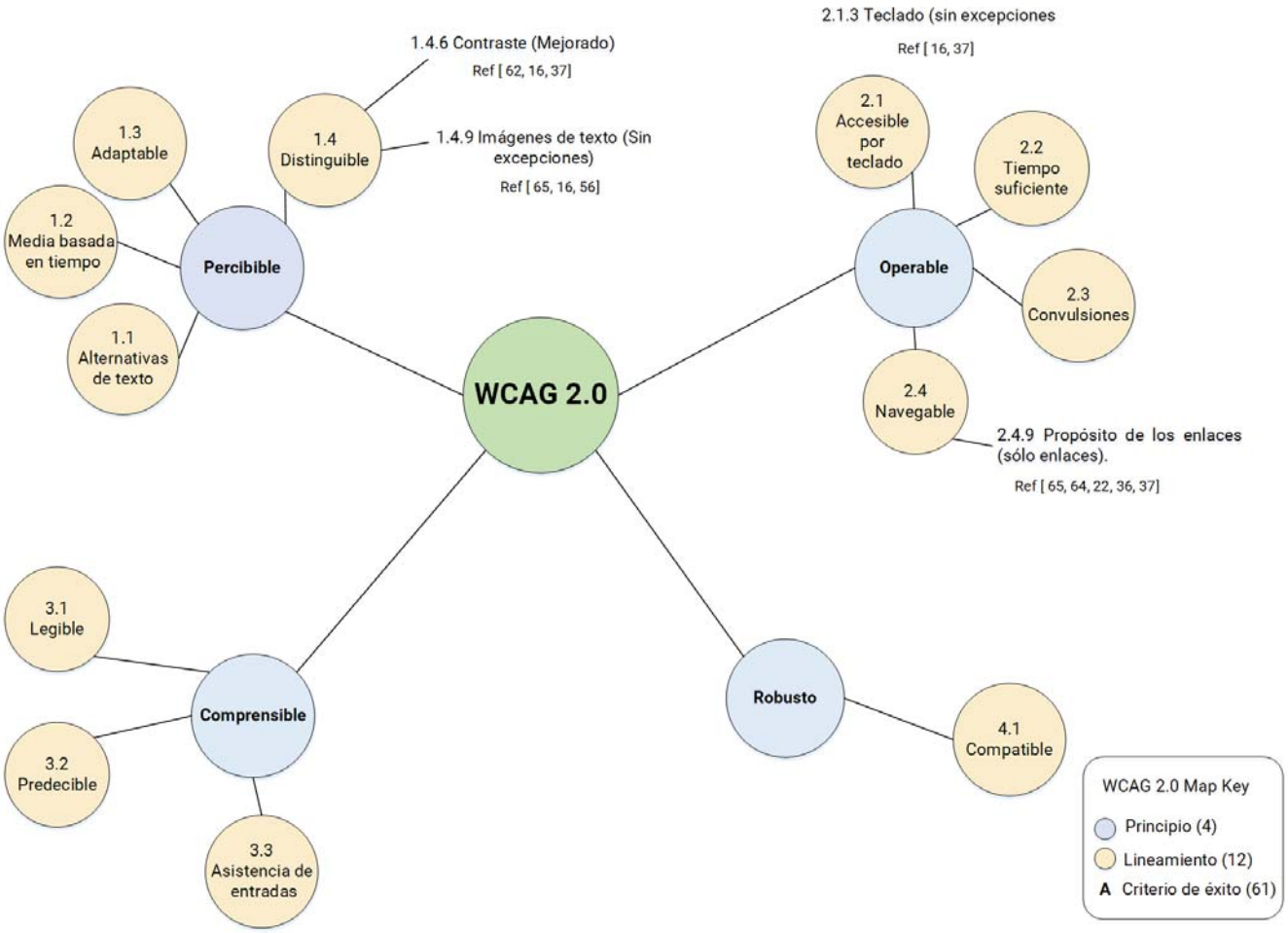


Figura 1.9: Criterios incumplidos reportados más de dos veces en el Nivel AAA.

En el apéndice, en la sección 1.E, Figura 1.10 se detalla la cantidad de aspectos que se reportan como incumplidos más de dos veces, por cada uno de los cuatro principios que conforman el estándar WCAG 2.0. Se nota claramente que el principio percibible y el comprensible son los que tienen la mayor cantidad de incumplimientos.

1.6.3. Desafíos reportados sobre las evaluaciones de la accesibilidad Web (RQ3)

La pregunta tres identificó de cada uno de los estudios analizados cuáles reportan desafíos al realizar las evaluaciones de accesibilidad Web. Se agruparon en dos categorías para permitir una mejor visualización de la información, los cuales son: aspectos técnicos y aspectos sobre regulaciones de Gobierno. Es interesante resaltar que los desafíos reportados son propiamente sobre el tema de accesibilidad Web y no tanto sobre las herramientas de evaluación.

Cuadro 1.6: Desafíos técnicos reportados.

Desafíos	Estudios
La disponibilidad del personal de desarrollo Web capacitado en accesibilidad.	[65, 33, 34, 37]
La necesidad de evaluadores expertos en accesibilidad Web.	[50, 51, 24]
La consideración de los criterios de accesibilidad durante la definición de los aspectos del diseño de sitios Web.	[54, 59, 66]
La utilización de una herramienta de evaluación única como fuente de información.	[67, 51]
El lenguaje variable de los sitios Web.	[46]

Como se muestra en el Cuadro 1.6 los aspectos técnicos reportados son de gran relevancia para evaluar la accesibilidad en los sitios Web, sin embargo, estos no siempre son cumplidos cuando se realiza el desarrollo de un sitio.

En el Cuadro 1.7 se muestran los desafíos con respecto a las regulaciones de gobierno que fueron reportados por los estudios analizados. Del Cuadro 1.7 se puede inferir que en muchos casos las regulaciones existentes no son de carácter obligatorio o bien no hay un marco legal para exigir formalmente el cumplimiento de la accesibilidad Web.

Cuadro 1.7: Desafíos con respecto a las regulaciones de Gobierno.

Desafíos	Estudio
Los responsables de las políticas deben desarrollar y promover marcos legales y normativas para abordar los problemas de accesibilidad Web.	[68, 69, 70, 71, 41, 42, 60]
Se debe hacer que la accesibilidad de los sitios Web gubernamentales sea un requisito obligatorio.	[63, 33, 67, 72]
Se deben fortalecer y compartir las políticas de accesibilidad Web de cada país, así como aplicar mejores leyes y fomentar prácticas que hagan que los sitios Web sean más accesibles.	[62, 24, 72]

En los estudios analizados, algunos autores mencionan que la evaluación de accesibilidad Web presenta desafíos, sin embargo, no detallan o mencionan claramente cuáles son los desafíos que identifican, dejando un vacío para poderlos agrupar o categorizar.

1.7. Discusión

La evaluación de la accesibilidad Web se ha convertido en una necesidad apremiante de cualquier sitio Web, independientemente del área de aplicación. Las herramientas automáticas pueden ser de mucha ayuda porque vienen a apoyar los procesos de validación, pero una evaluación manual complementaria es importante y necesaria para construir un sitio Web accesible.

Los resultados muestran que existe gran cantidad de herramientas para evaluar la accesibilidad Web pero en la práctica estas no son usadas por los desarrolladores. Se requiere una regulación que exija el cumplimiento de la accesibilidad en sitios Web para que los responsables de implementarlo, lo consideren esencial durante el proceso. Las evaluaciones de accesibilidad deben realizarse periódicamente, ya que por lo general estas pruebas se realizan al final del ciclo de desarrollo, lo que puede provocar retrabajo.

Se evidencia la existencia de herramientas para evaluar la accesibilidad de sitios Web; sin embargo, dentro de las limitaciones identificadas en este mapeo, se encuentra que la mayoría de los estudios solo mencionan el uso de una herramienta, pero no especifican el porqué de su elección y tampoco dejan en claro qué lista de criterios revisa la herramienta de forma automática y cuáles no logra detectar. Se recomienda mejorar los reportes de los artículos analizados, que los autores generen más valor al detalle cuando se realiza el análisis.

En cuanto a los desafíos reportados por los estudios se evidencia que la estandarización de la normativa y regulaciones por parte de los responsables, apenas está en proceso, y al no existir una reglamentación obligatoria, los desarrollos Web van a seguir teniendo limitaciones en cuanto a accesibilidad. Además, no todos los autores mencionan cuáles fueron los desafíos en cuanto a accesibilidad, solo mencionan que los hay, pero sin entrar en detalles.

1.8. Lecciones aprendidas

Para la realización de esta investigación se siguió la aplicación de la metodología de mapeo sistemático de literatura, nueva en mi caso como investigadora. Fue un proceso de adaptación, sin embargo me permitió aprender a aplicarla, y aunque tomó tiempo, es altamente recomendada porque el proceso es repetible y permite analizar la evidencia (estudios primarios) existente sobre un tema de investigación. No obstante, aparte del análisis de estudios mediante esta metodología, también se permitió realizar revisiones sistemáticas de literatura que condujeron al análisis en profundidad de investigaciones realizadas por otros autores acerca del tema de estudio.

Para el desarrollo del trabajo la metodología que se siguió fue esencial, ya que

permitió seguir un proceso estructurado de búsqueda, selección y análisis de cada uno de los artículos, esto garantizó la calidad en la información. La creación del protocolo preliminar en la primera fase del proceso fue de gran ayuda, ya que permitió aclarar la metodología, y poder comprender cómo guiar el proceso para avanzar exitosamente en las en las demás etapas de la investigación.

En las fases del proceso de investigación, existieron unas más retadoras que otras, dentro de las cuales se puede mencionar: el desarrollo y ajuste de la cadena de búsqueda, y la creación del formulario de extracción que permitiría obtener la información importante y necesaria para desarrollar el tema. El proceso requirió pilotajes iniciales y varias iteraciones para depurar el método y obtener los mejores resultados.

Desde la perspectiva académica, la aplicación de esta metodología permite realizar investigaciones muy completas y de calidad. Esta investigación viene a conocer la variedad de herramientas que se han desarrollado para evaluar la accesibilidad Web, así como también de la importancia que tiene el tema, lo que permite generar estudios secundarios que vienen a enriquecer el conocimiento en la materia.

A nivel profesional, es muy significativo realizar investigaciones en áreas de la Ingeniería del Software, pues son bases importantes para la solución de problemáticas en el desarrollo de proyectos tecnológicos, además, permite trabajar sobre temas actualizados que se desarrollan en el campo laboral.

El trabajo realizado en esta investigación permite marcar un objetivo que no se limita a cumplir un requisito en la conclusión de un plan de estudios, sino que el resultado de este mapeo de literatura pueda servir como referencia para otros investigadores en el área de evaluación de la accesibilidad Web.

1.9. Conclusiones

En este documento, se ha informado sobre los resultados de un estudio secundario que realizó un mapeo sistemático de estudios primarios en herramientas de evaluación de accesibilidad Web, publicados entre 2004 y 2019. El objetivo de esta investigación fue realizar un mapeo sistemático de estudios que utilizaron herramientas para evaluar accesibilidad en sitios Web. Como resultado se analizó un conjunto de 50 estudios primarios, después de aplicar criterios de exclusión e inclusión. Se logró

identificar 38 herramientas, un conjunto de criterios que se reportan como más incumplidos, cinco desafíos técnicos y cuatro desafíos sobre regulaciones de Gobierno, ambos sobre accesibilidad Web.

Existen herramientas capaces de evaluar distintos aspectos de la accesibilidad de las páginas Web, y algunas herramientas pueden identificar los errores encontrados a partir de los criterios de la normativa WCAG (versión 1.0 y 2.0). Estas herramientas pueden apoyar el proceso de evaluación de la accesibilidad Web pero no pueden reemplazar las revisiones realizadas por los expertos en el tema.

Con respecto a las herramientas reportadas para evaluar la accesibilidad Web, no se obtuvo un reporte claro de cuál fue el método de selección de la herramienta, ya que la mayoría de los autores solo hicieron mención de la o las herramientas que iban a utilizar y no especificaban el porqué de su uso o los criterios que evaluaba. Esta condición podría generarse por el formato del artículo lo que limita poder extenderse en el contenido. Además, con respecto a las herramientas, en los artículos analizados, los autores no hacen un análisis de cuáles son los criterios de éxito que no pueden ser evaluados por estas y que solo un humano lo podría hacer; sería importante analizar este vacío más a profundidad.

Es necesario continuar realizando evaluaciones objetivas sobre sitios Web controlados que permitan determinar el nivel de cumplimiento de los distintos aspectos planteados por los estándares de accesibilidad Web y cómo las herramientas los identifican y reportan. Los resultados muestran la necesidad de más estudios que aborden de forma detallada cuáles son las herramientas más recomendadas para evaluar accesibilidad Web, así, como listar los criterios que pueden ser evaluados de forma automática por las herramientas.

A medida que las organizaciones y los diseñadores tomen conciencia y pongan en práctica la accesibilidad, garantizarán que su contenido pueda ser accedido por una población más amplia. La falta de accesibilidad generalizada podría considerarse consecuencia del desconocimiento acerca de las pautas de accesibilidad del W3C.

Con esta investigación se puede concluir la importancia de implementar la accesibilidad Web en el desarrollo de los sitios. En el ámbito académico aporta como insumo en los cursos de la Escuela de Computación e Informática como el de interacción humano-computador, pruebas de software e ingeniería del software, además,

para la Maestría que tiene cursos que tratan sobre el tema de accesibilidad Web. A nivel profesional ya se identificó que hay una lista de un conjunto de herramientas que se pueden recomendar para que los profesionales puedan realizar sus evaluaciones; sin embargo, como se reportó en los desafíos, se requiere más capacitación en el tema de accesibilidad Web. En el plano de investigación académica, incentivar a que se realicen más investigaciones sobre herramientas de accesibilidad Web tomando como referencia los criterios que pueden evaluar de forma automática y que los resultados obtenidos puedan contribuir con las actividades diarias en el área de desarrollo de software.

Como trabajo futuro se plantea realizar una investigación a fin de determinar cuáles son las capacidades de las herramientas para evaluar la accesibilidad Web y cuáles son las más recomendadas, así como, analizar qué se puede hacer automáticamente y qué solo por humanos, y la mejor forma de combinar estrategias (herramientas y humanos) para mejorar la efectividad de las evaluaciones que se realizan, ya que con la lectura de los estudios analizados esto no se reporta.

A partir del capítulo de la memoria se desarrolló un artículo científico que fue enviado y aceptado en las IV Jornadas Costarricenses de Investigación en Computación e Informática JoCICI 2019, celebradas en Universidad Estatal a Distancia (UNED) durante el 19 y 20 de agosto del 2019. En el Apéndice 1.F se encuentra el artículo publicado.

Apéndice

1.A. Lista de estudios primarios incluidos

En el Cuadro 1.8 se muestran la lista completa de los artículos primarios incluidos y analizados en el estudio, la cual contiene el ID identificador del artículo, el título, el año de publicación y el número de referencia.

Cuadro 1.8: Lista de estudios primarios incluidos.

ID	Título	Año	Est.
4	A new approach to the automatic web accessibility	2011	[48]
5	A pilot study for evaluating Arabic websites using automated WCAG 2.0 evaluation tools	2011	[46]
11	A test procedure for checking the WCAG 2.0 guidelines	2016	[47]
20	Accessibility analysis and evaluation of Bangladesh government websites	2012	[33]
21	Accessibility analysis of higher education institution websites of Portugal	2019	[65]

Continúa en la página siguiente.

ID	Título	Año	Est.
23	Accessibility Analyzer: Tool for New Adaptations in Government Web Applications to Improve Accessibility	2017	[50]
34	Accessibility evaluation of top university websites: a comparative study of Kyrgyzstan, Azerbaijan, Kazakhstan and Turkey	2018	[62]
35	Accessibility Evaluation Using WCAG 2.0 Guidelines Webometrics Based Assessment Criteria (Case Study: Sebelas Maret University)	2014	[64]
41	Accessibility of eGovernment Services in Latin America	2018	[22]
42	Accessibility of Indian universities' homepages: An exploratory study	2018	[34]
45	Accessibility Verification of WWW Documents by an Automatic Guideline Verification Tool	2004	[49]
48	Accessibility evaluation improvement using Case Based Reasoning	2011	[45]
49	Accessibility Evaluation Using Web Content Accessibility Guidelines (WCAG) 2.0	2016	[23]
51	Accessibility of the cyprus Island municipal websites	2017	[44]
53	Accessibility, Quality and Performance of Government Portals and Ministry Web Sites: A View Using Diagnostic Tools	2015	[69]
59	An analysis of personalized web accessibility	2006	[73]

Continúa en la página siguiente.

ID	Título	Año	Est.
60	An Analysis of Website Accessibility in Higher Education in Indonesia Based on WCAG 2.0 Guidelines	2018	[55]
78	Automatically: An Automated Refactoring Method and Tool for Improving Web Accessibility	2018	[40]
79	Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests	2013	[38]
84	Challenges to assess accessibility in higher education websites: A comparative study of Latin america universities	2018	[24]
88	Checking and correcting the source code of web pages for accessibility	2012	[39]
101	Effect of human development level of countries on the web accessibility and quality in use of their municipality websites	2019	[63]
102	e-Government in Africa: Perceived concerns of personswith disabilities (PWDs) in South Africa	2017	[70]
105	Empowering agile project members with accessibility testing tools: A case study	2017	[66]
111	Evaluating accessibility of Malaysian ministries websites using WCAG 2.0 and Section 508 Guideline	2016	[71]
112	Evaluating accessibility of Malaysian public universities websites using AChecker and WAVE	2016	[51]

Continúa en la página siguiente.

ID	Título	Año	Est.
116	Evaluating metropolitan assembly web sites in Ghana: Accessibility, compatibility and usability	2016	[72]
117	Evaluating the accessibility of provinces' e-government websites in Indonesia	2017	[74]
119	Evaluating the web accessibility of websites of the central government of Nepal	2007	[60]
122	Evaluating web accessibility of educational websites	2015	[41]
124	Evaluation of accessibility in Mexican cybermedia	2019	[67]
125	Evaluation of accessibility of university websites: A case from turkey	2017	[75]
127	Evaluation of web accessibility in China: changes from 2009 to 2013	2016	[76]
136	Framework for Accessibility Evaluation of Hospital Websites	2018	[59]
161	Measuring the Accessability Based on Web Content Accessibility Guidelines	2018	[54]
173	Portuguese web accessibility snapshot: Status of the Portuguese websites regarding accessibility levels	2010	[77]
179	Quality evaluation of government websites	2017	[78]
194	The Arabian E-government websites accessibility: A case study	2016	[68]
204	Toward a combined method for evaluation of web accessibility	2018	[35]

Continúa en la página siguiente.

ID	Título	Año	Est.
206	Toward better web accessibility	2015	[16]
212	Towards Web Accessibility in Telerehabilitation Platforms	2018	[36]
216	Universities of the Kyrgyz Republic on the Web: accessibility and usability	2017	[61]
218	Usability and accessibility analysis of selected government websites in Sri Lanka	2016	[43]
219	Usability and accessibility evaluation of Libyan government websites	2019	[52]
222	Usability evaluation of academic websites using automated tools	2014	[53]
239	Web accessibility for disabled: A case study of government websites in Pakistan	2012	[42]
241	Web accessibility for the visually impaired: A case of higher education institutions' websites in Ghana	2017	[37]
245	Web accessibility investigation and identification of major issues of higher education websites with statistical measures: A case study of college websites	2019	[58]
256	Web content accessibility of municipal web sites in Turkey	2015	[56]
266	Website accessibility performance evaluation in Malaysia	2008	[57]

1.B. Evaluación de calidad de los estudios primarios

El Cuadro 1.9 muestra los resultados de la evaluación de calidad de todos los estudios analizados. Para cada criterio se evaluó en una escala de 0 a 2 puntos, por lo que la calidad se mide sobre 6 puntos.

Cuadro 1.9: Evaluación de calidad de los estudios primarios.

ID	Est.	Año	Q1	Q2	Q3	Total
4	[48]	2011	2	2	2	6
5	[46]	2011	2	2	2	6
11	[47]	2016	2	2	2	6
20	[33]	2012	2	2	2	6
21	[65]	2019	2	2	2	6
23	[50]	2017	2	1	1	4
34	[62]	2018	2	2	1	5
35	[64]	2014	2	2	2	6
41	[22]	2018	2	2	2	6
42	[34]	2018	2	2	2	6
45	[49]	2004	2	1	1	4
48	[45]	2011	2	0	0	2
49	[23]	2016	2	2	0	4
51	[44]	2017	2	0	0	2
53	[69]	2015	2	1	0	3
59	[73]	2006	2	1	1	4

Continúa en la página siguiente.

ID	Est.	Año	Q1	Q2	Q3	Total
60	[55]	2018	2	2	2	6
78	[40]	2018	2	0	1	3
79	[38]	2013	2	2	2	6
84	[24]	2018	2	2	2	6
88	[39]	2012	2	1	0	3
101	[63]	2019	2	1	0	3
102	[70]	2017	2	1	2	5
105	[66]	2017	2	1	2	5
111	[71]	2016	2	2	2	6
112	[51]	2016	2	2	2	6
116	[72]	2018	2	1	2	5
117	[74]	2017	2	2	2	6
119	[60]	2007	2	2	1	0
122	[41]	2015	2	1	1	4
124	[67]	2019	2	2	1	5
125	[75]	2017	2	1	1	4
127	[76]	2016	2	1	0	3
136	[59]	2018	2	2	2	6
161	[54]	2018	2	1	0	3
173	[77]	2010	2	1	1	4
179	[78]	2017	2	2	2	6
194	[68]	2016	2	2	2	6

Continúa en la página siguiente.

ID	Est.	Año	Q1	Q2	Q3	Total
204	[35]	2018	2	2	2	6
206	[16]	2015	2	2	2	6
212	[36]	2018	2	2	2	6
216	[61]	2017	2	2	1	5
218	[43]	2016	2	2	1	5
219	[52]	2019	2	2	2	6
222	[53]	2014	2	2	2	6
239	[42]	2012	2	1	2	5
241	[37]	2017	2	1	1	4
245	[58]	2019	2	1	1	4
256	[56]	2015	2	2	1	5
266	[57]	2008	2	1	0	3

1.C. Total de herramientas reportadas.

El Cuadro 1.10 muestra las 38 herramientas que reportaron los estudios, contiene el nombre de las herramientas, referencia, cantidad y la descripción.

Cuadro 1.10: Total de herramientas reportadas.

Herramienta	Estudios	Cantidad	Descripción
Achecker	[16, 46, 47, 65, 50, 62, 23, 44, 38, 39, 63, 70, 71, 51, 74, 52, 53, 33, 34, 68, 57]	21	Herramienta online que permite revisar Section 508, WCAG 1.0 y WCAG 2.0 al mismo tiempo. Es licencia libre.
Wave	[16, 22, 24, 65, 50, 71, 51, 53, 34, 68, 41, 59, 35, 36, 43]	15	Herramienta online que revisa la accesibilidad de una página web y muestra el resultado sobre la propia página. Evalúa WCAG, es licencia libre.
TAW	[46, 50, 38, 16, 52, 68, 64, 69, 55, 67, 78, 58, 56]	14	Herramienta online, creada teniendo como referencia técnica las pautas de accesibilidad al contenido web WCAG 1.0, 2.0 WCAG 2.0 del W3C. Es licencia libre.

Continúa en la página siguiente.

Herramienta	Estudios	Cantidad	Descripción
SortSite	[38, 75, 72, 77, 68]	5	Valida no solo aspectos de accesibilidad de acuerdo a las WCAG 2.0 sino también enlaces rotos. Licencia comercial y es online.
Total Validator	[38, 74, 42]	3	Consta de un validador XHTML, un validador de accesibilidad WCAG y Section 508, un validador CSS, un corrector ortográfico y un corrector de enlaces rotos. Herramienta local y licencia comercial.
Siteimprove Accessibility Checker	[66, 36]	2	Permite revisar Section 508 y WCAG 2.0. Software libre
Tenon	[59, 36]	2	Revisa pautas de accesibilidad al contenido web del W3C: WCAG 1.0, WCAG 2.0, Section 508. Licencia libre, online

Continúa en la página siguiente.

Herramienta	Estudios	Cantidad	Descripción
Bobby	[60, 49]	2	Es el validador más antiguo. La comprobación de accesibilidad se basa tanto en las pautas WCAG 1.0 como en las de la section 508. Ya no existe, de licencia comercial.
Hera	[53, 76]	2	Herramienta de revisión para facilitar la aplicación de las WCAG 2.0. Online de extensión para navegador y licencia comercial.
EvalAccess 2.0	[33, 61]	2	Permite evaluar la accesibilidad web usando la pauta WCAG 1.0. Licencia libre, online.
Web Accessibility Assessment Tool	[46, 73]	2	Es una aplicación Java. Evalúa un sitio web de acuerdo con WCAG 2.0 (nivel A, nivel AA y nivel AAA).

Continúa en la página siguiente.

Herramienta	Estudios	Cantidad	Descripción
aXe	[65, 65]	2	Permite revisar pautas WCAG 2.0, Section 508. Licencia código abierto. Online, de extensión para navegador.
Webpage	[34]	1	Revisa la optimización y la velocidad del sitio web.
Worldspace FireEyse	[46]	1	Permite evaluar sitios web y ser interpretados de acuerdo con WCAG 1.0,WCAG 2 Section 508.
Koa11y	[35]	1	Permite verificar si una página web sigue los estándares 2.0 AA Y AAA, ayuda a comprender qué hacer para mejorar la accesibilidad.
OpenWAX	[36]	1	Ayuda a resolver problemas de accesibilidad, algunos están de acuerdo con las pautas WCAG 2.0.

Continúa en la página siguiente.

Herramienta	Estudios	Cantidad	Descripción
HTML Code Sniffer	[40]	1	Viene con estándares que cubren los tres niveles de conformidad de WCAG 2.0 y la legislación de la Section 508.
aDesigner	[37]	1	Se trata de un producto de IBM que simula las condiciones con las que navega por la Web una persona que tiene algún tipo de discapacidad visual, para así poder comprobar su accesibilidad y usabilidad para una mayor variedad de usuarios que los sistemas de solo texto o de síntesis de voz.
Google PageSpeed insight	[43]	1	Informa sobre el rendimiento de las páginas tanto en dispositivos móviles como en ordenadores y ofrece sugerencias para mejorarlas.

Continúa en la página siguiente.

Herramienta	Estudios	Cantidad	Descripción
Google Mobile-Friendly Test	[43]	1	La herramienta de prueba amigable para dispositivos móviles de Search Console es una forma rápida y fácil de probar si una página de su sitio es amigable para dispositivos móviles.
Pingdom tool	[43]	1	Analiza la velocidad de carga de la web y muestra diferentes soluciones para mejorarla.
Tanaguru	[47]	1	Tanaguru es una herramienta de evaluación de sitios web de código abierto . Está dedicado a las auditorías de accesibilidad (a11y) y se centra en la confiabilidad y el alto nivel de automatización.
EIII – Page Checker	[44]	1	
Contrast-Finder	[47]	1	Proporciona soluciones para contrastes.

Continúa en la página siguiente.

Herramienta	Estudios	Cantidad	Descripción
AccessColor	[47]	1	Permite mostrar lo problemas de color en sitios web mediante la pauta WCAG.
Accessibility Toolbar	[47]	1	Se desarrolló para ayudar a la revisión manual de varios aspectos de la accesibilidad en las páginas web.
Cynthia Says	[41]	1	Revisa WCAG 1.0 y Section 508. También analiza la calidad de los textos alternativos de las imágenes.
Deque	[38]	1	Herramienta online que permite verificar WCAG 1.0, WCAG 2.0 y Section 508.
AMP	[38]	1	Permite evaluar la accesibilidad web usando la pauta WCAG 2.0, W3C 2.0,WAC 1.0 , W3C 1.0 y Secction 508.Licencia comercial.

Continúa en la página siguiente.

Herramienta	Estudios	Cantidad	Descripción
Markup Validation Service	[33]	1	Permite evaluar la accesibilidad web usando la pauta WCAG 1.0.
AC-TAW	[48]	1	Modifica el código del sitio Web para adaptarlo a la normativa WAI. Analiza y modifica el código HTML del sitio Web, pero no solamente el código HTML, sino que también es capaz de comprender y analizar el código XHTML.
CHECORSER	[39]	1	Para verificar la conformidad de WCAG de cualquier página de cliente HTML. Devuelve todos los nodos de cliente no conformes; cada uno contiene una lista de infracciones.
CORrector	[39]	1	Permite evaluar la accesibilidad web usando la pauta WCAG .

Continúa en la página siguiente.

Herramienta	Estudios	Cantidad	Descripción
Evaluator 1.0.2	[42]	1	Evalúa un sitio web o una página web única de acuerdo con los requisitos de Nivel A y AA de las Directrices de Accesibilidad al Contenido en la Web (WCAG) 2.0 del W3C .
TinyMCE	[45]	1	Tiene la habilidad de convertir un campo del tipo textarea u otros elementos de html en instancias del editor.
HTML Tidy	[48]	1	Es una aplicación de consola cuyo propósito es arreglar HTML inválido, detectar potenciales errores de accesibilidad, y mejorar el diseño y estilo de indentación del marcado resultante.

1.D. Resultados de las herramientas utilizadas por año.

En el Cuadro 1.11 se presenta la lista completa de las herramientas que reportaron los estudios analizados. Se muestra el nombre y el año del artículo.

Cuadro 1.11: Resultados de las herramientas utilizadas por año.

Herramienta	Año en el que se utilizó
Achecker	2008, 2011-2019
TAW	2008, 2011, 2013-2019
WAVE	2014-2019
Total Validator	2012, 2013, 2017
SortSite	2010, 2013, 2016- 2018
Bobby	2004 y 2007
EvalAccess 2.0	2012 y 2017
HERA	2014 y 2016
Siteimprove	
Accessibility Checker	2018 y 2019
aXe	2019
Webpage	2018
Analyzer	2018
Tenon	2018
Koa11y	2018
OpenWAX	2018
HTML code Sniffer	2018

Continúa en la página siguiente.

Herramienta	Año en el que se utilizó
aDesigner	2017
GooglePageSpeed insight	2016
Google Mobile-Friendly Test	2016
Pingdom tool	2016
Tanaguru	2016
EIII – Page Checker	2016
Contrast-Finder	2016
AccessColor	2016
AccessibilityToolbar	2016
PowerMapper	2016
Web Accessibility Assessment Tool	2011 y 2014
Cynthia Says	2015
Deque	2013
AMP	2013
Markup Validation Service	2012
AC-TAW	2011
CHECORSER	2012
CORrector	2012
Evaluator 1.0.2	2012
TinyMCE	2011
HTML Tidy	2011

Continúa en la página siguiente.

Herramienta	Año en el que se utilizó
Worldspace FireEyse	2011

1.E. Aspectos incumplidos con más de dos veces por cada uno de los cuatro principios del estándar WGAC 2.

La Figura 1.10 expone cada uno de los criterios que se reportan más de dos veces como incumplidos por cada uno de los cuatro principios del estándar WCAG.

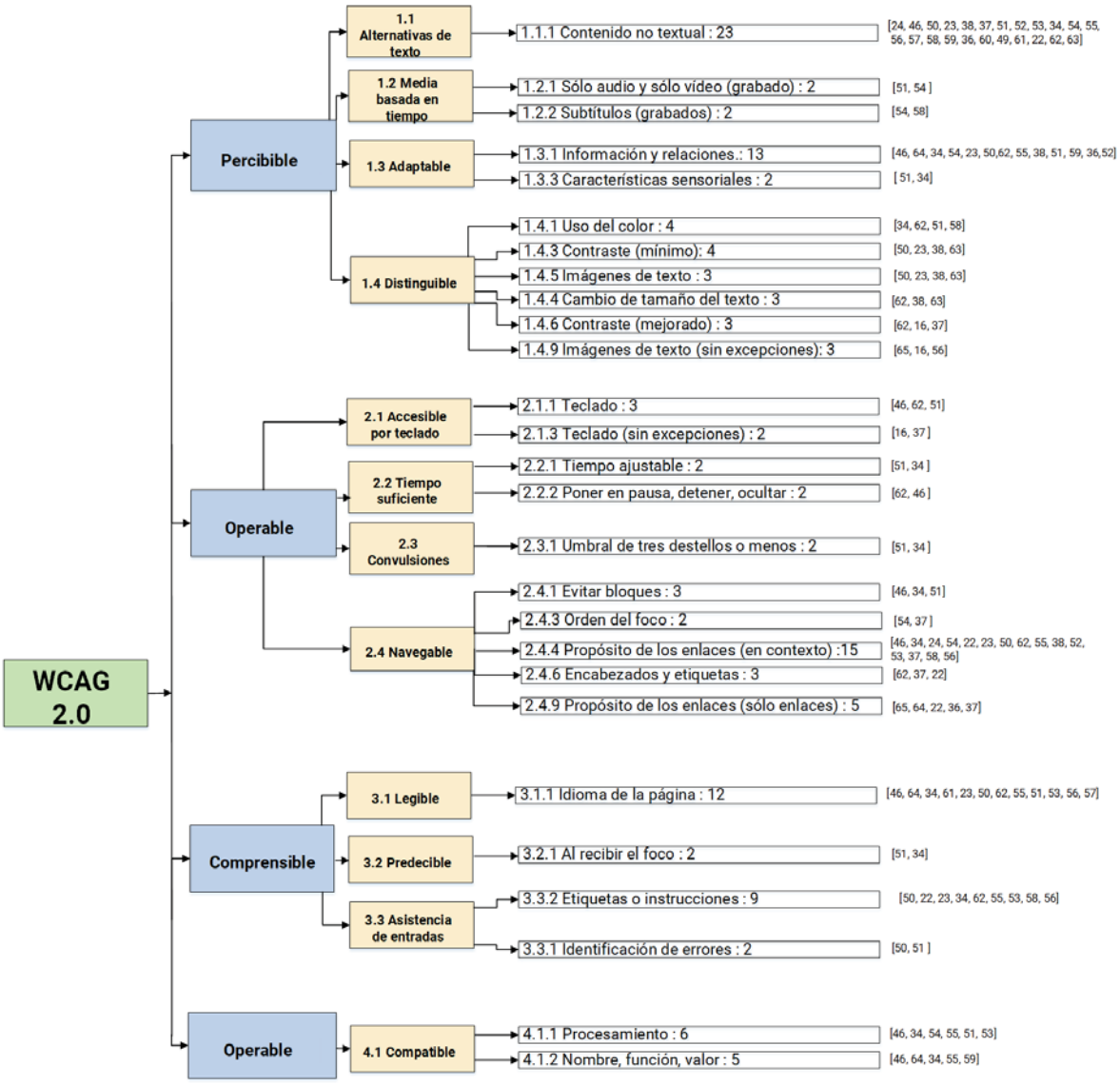


Figura 1.10: Aspectos más incumplidos por principio del estándar WGAC 2.0.

1.F. Artículo

A partir del capítulo de la memoria se desarrolló un artículo científico que fue enviado y aceptado en las IV Jornadas Costarricenses de Investigación en Computación e Informática JoCICI 2019, celebradas en Universidad Estatal a Distancia (UNED) durante el pasado 19 y 20 de agosto del 2019.

Tools for the evaluation of web accessibility: A systematic literature mapping

Patricia Agüero-Flores
Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica
San José, Costa Rica
patricia.aguero@ucr.ac.cr

Christian Quesada-López
Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica
San José, Costa Rica
cristian.quesadalopez@ucr.ac.cr

Alexandra Martínez
Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica
San José, Costa Rica
alexandra.martinez@ucr.ac.cr

Marcelo Jenkins
Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica
San José, Costa Rica
marcelo.jenkins@ucr.ac.cr

***Abstract*—In recent years, different tools have been proposed to automate the evaluation of the web contents accessibility criteria proposed by the World Wide Web Consortium (W3C). These tools can verify that a website complies with web accessibility standards such as WCAG, but the results of the evaluation may depend on the tool used. This study identifies and characterizes web accessibility assessment tools through a systematic literature mapping. A total of 50 articles were analyzed. We report the accessibility criteria evaluated by each tool as well as the main challenges related to the evaluations.**

***Index Terms*—web content accessibility, accessibility tools, W3C, WCAG.**

I. Introducción

La Web se ha convertido en un recurso esencial para distintos aspectos de la vida en sociedad, tales como la educación, el empleo, el gobierno, el comercio, la salud, y el entretenimiento. Por ello es necesario que la Web sea accesible, de manera que las personas con discapacidad tengan igualdad de oportunidades para participar activamente en la sociedad [1]. La accesibilidad web significa que las personas que tienen alguna discapacidad puedan percibir, entender, navegar e interactuar con los sitios web [1]. Entre las discapacidades que afectan el acceso a la web están la discapacidad auditiva, la cognitiva, la neurológica, la física, y la visual.

La accesibilidad web también puede beneficiar a personas sin discapacidades, pero que utilizan teléfonos móviles, relojes inteligentes, televisores inteligentes y otros dispositivos con pantallas pequeñas y diferentes modos de entrada. También puede ayudar a personas mayores con capacidades cambiantes debido al envejecimiento, personas con discapacidades temporales (como un brazo roto o anteojos perdidos), personas con limitaciones situacionales (como a la luz del sol o en un entorno donde no pueden escuchar el audio), y personas que utilizan una conexión a Internet lenta o que tienen un ancho de banda limitado o costoso [2].

La Organización Mundial de la Salud estima que el 15% de la población posee algún tipo de discapacidad [3]. Las tecnologías de apoyo son usadas para mejorar o mantener las capacidades funcionales de personas con discapacidad y han probado ser un factor de cambio [3]. A nivel mundial se realizan esfuerzos para construir ciudades más inteligentes, donde las tecnologías digitales se vuelven esenciales para mejorar la administración y los servicios. Uno de los retos más importantes es reducir la brecha digital y extender el acceso a todas las personas para reducir la desigualdad, principalmente de las minorías [4]. En Costa Rica, estudios especializados [5] han señalado que la población con discapacidad enfrenta múltiples barreras que limitan el ejercicio de sus derechos ciudadanos, con desigualdades en el acceso a los servicios fundamentales. Asimismo, se ha reconocido el potencial de las tecnologías digitales como un generador de oportunidades para esta población [6].

En el año 2017, la Unión Internacional de Telecomunicaciones designó al Programa de la Sociedad de la Información y el Conocimiento (PROSIC) y al Centro de Informática de la Universidad de Costa Rica, como entes acreditadores para desarrollos digitales accesibles. Otras iniciativas como el Observatorio de Tecnologías Accesibles Inclusivas del Instituto Tecnológico de Costa Rica buscan reducir las brechas existentes en accesibilidad digital con el fin de promover la igualdad de oportunidades.

Con la implementación de la Ley para la Igualdad de Oportunidades para las Personas con Discapacidad (Ley 7600) y el Pacto por un País Accesible e Inclusivo del Gobierno de la República, las instituciones del sector público han realizado esfuerzos para el desarrollo de sitios web más inclusivos. Similarmente, distintas evaluaciones se han realizado para determinar la accesibilidad de los sitios de estas instituciones; por ejemplo, la Defensoría de los Habitantes y el Centro de Investigación y Capacitación en Administración Pública de la Universidad de Costa Rica aplican el Índice de Transparencia del Sector Público anualmente, en el cual se contempla la accesibilidad web para determinar la disponibilidad de textos

alternativos en las imágenes con enlaces, el tamaño de botones y el uso de subtítulos o lenguaje de señas, entre otros. En el 2018, la accesibilidad web fue uno de los aspectos con menor calificación en los sitios web de instituciones públicas. Por todo lo anterior, es esencial que la accesibilidad web sea considerada como parte del desarrollo de los servicios que se ofrecen a través de sitios web.

Las pautas de accesibilidad para el contenido de la web (WCAG: *Web Content Accessibility Guidelines*) [1] del consorcio web [7] definen un conjunto de criterios que los sitios web deben cumplir para alcanzar cierto nivel de accesibilidad. El WCAG constituye un estándar de accesibilidad web que satisface las necesidades de personas, organizaciones y gobiernos a nivel internacional, y explica cómo hacer el contenido web más accesible para todas las personas. Tradicionalmente, estos criterios han sido evaluados por expertos [8], pero recientemente se han propuesto y desarrollado herramientas que evalúan automáticamente algunos de estos criterios. No obstante, los resultados de dichas evaluaciones dependen de la herramienta utilizada y sus características [8], [9].

Para desarrollar sitios web es deseable contar con herramientas automatizadas que permitan evaluar la accesibilidad web. Sin embargo, estas herramientas pueden tener limitaciones en cuanto a los criterios que plantean los estándares de accesibilidad web [9]. Más aun, existen distintos instrumentos de evaluación de accesibilidad web, según lo indicado por las normas WCAG. Además, constantemente se trabaja en redefinir la accesibilidad web, de manera que incluya componentes clave considerados por investigadores y profesionales [10], por lo que las herramientas deben ir evolucionando de acuerdo a las necesidades que surjan.

Las herramientas que evalúan la accesibilidad web pueden apoyar al desarrollador a realizar un diagnóstico de los problemas de accesibilidad, con base en distintos niveles de conformidad [11]. La normativa WCAG 2.1 establece que los contenidos web pueden cumplir con tres criterios de conformidad: A, AA, y AAA [12]. El consorcio web [7], por su parte, mantiene listas de herramientas que pueden apoyar los procesos de evaluación de la accesibilidad web, y proporciona una guía de selección de herramientas.

El propósito de nuestro estudio es identificar y caracterizar la literatura que existe sobre herramientas para la evaluación de la accesibilidad web. Las preguntas de investigación son:

- RQ1. ¿Qué herramientas han sido utilizadas para evaluar la accesibilidad web?
- RQ2. ¿Qué criterios de accesibilidad han sido reportados con las herramientas?
- RQ3. ¿Cuáles son los principales desafíos encontrados en las evaluaciones automáticas de la accesibilidad web?

Se realizó un mapeo sistemático de literatura para organizar la evidencia existente y categorizarla con respecto a las herramientas de evaluación de accesibilidad, los criterios de accesibilidad evaluados, el nivel de cumplimiento con respecto a los estándares de accesibilidad y los desafíos de las evaluaciones realizadas.

II. Trabajo relacionado

Múltiples estudios han analizado aspectos y herramientas de accesibilidad web. Nagaraju et al. [13] estudiaron cómo se evaluó la accesibilidad web usando las pautas de WCAG 2.0, cuales métodos o herramientas fueron utilizados, y cómo se comparaban los resultados de dichas evaluaciones.

Por su parte, Tollefsen et al. [9] identificaron un conjunto de herramientas para evaluar la accesibilidad web de acuerdo al WCAG 2.0. Los autores indican que alrededor del 50% de las reglas de WCAG se pueden evaluar mediante las herramientas existentes. Kirchner [14] identificó distintas herramientas capaces de evaluar la accesibilidad web, algunas de las cuales podían reparar páginas web que no eran accesibles. El autor indica que uno de los principales problemas durante las evaluaciones de accesibilidad surge cuando se necesitan navegadores que soporten las extensiones requeridas.

Por otro lado, Luque et al. [8] indican que una evaluación de accesibilidad web totalmente automatizada no es factible debido a la naturaleza de las pautas WCAG. Los autores evaluaron la cobertura del WCAG ofrecida por algunas herramientas así como sus debilidades y diferencias.

III. Metodología

En esta sección se describen brevemente el proceso de mapeo utilizado, según los lineamientos de Petersen et al. [15] y a las recomendaciones de Kitchenham [16] y Biolchini et al. [17].

El objetivo de este estudio (formulado con el modelo GQM [18]) fue *analizar* las herramientas de evaluación de la accesibilidad web, *con el propósito de* caracterizarlas *con respecto a* los criterios evaluados para el cumplimiento de los estándares de accesibilidad, y los desafíos reportados, *desde el punto de vista de* los investigadores *en el contexto de* evaluaciones de sitios web.

a) Estrategia de búsqueda y proceso de selección: Efectuamos una búsqueda exploratoria que permitió identificar estudios relevantes, que usamos de control [11], [19]–[21]. Esta búsqueda se derivó del objetivo, las preguntas de investigación, y términos usados en estudios secundarios relacionados. La cadena de búsqueda se generó usando el modelo PICO (Población, Intervención, Comparación, Salidas) en conjunto con términos clave en el título y el resumen de los artículos de control. Esto dio como resultado la siguiente cadena de búsqueda: (“accessibility” AND “tool*” AND (“W3C” OR “World Wide Web Consortium” OR “WCAG*”)). Esta cadena es producto de un proceso de refinamiento que incluyó varias pruebas piloto para reducir el ruido.

Las búsquedas automatizadas se realizaron en las bases de datos Web of Science, IEEE Xplore, y SCOPUS. El protocolo del mapeo se elaboró durante el primer semestre del 2019. La búsqueda automatizada se realizó en junio de 2019 y los estudios se analizaron durante este mismo periodo. El número de estudios recuperado para cada base de datos fue: 229 en Scopus, 43 en IEEE Xplore y 27 en Web of Science. Los artículos fueron tabulados en Microsoft Excel para los procesos de selección, evaluación y extracción de datos. Se

eliminaron los duplicados, se aplicaron los criterios de inclusión y exclusión (I/E) y finalmente se hizo la extracción y el análisis.

El proceso de I/E se hizo con base en el título y el resumen de los artículos (cuando hubo duda, se hizo lectura completa del artículo). Se excluyeron publicaciones que cumplían con la fórmula (E1 OR E2) donde (E1) Estudios no disponibles en texto completo, y (E2) Estudios secundarios o terciarios. Los estudios que no fueron excluídos y que cumplían con la fórmula (I1 AND I2 AND I3) fueron incluidos, donde (I1) Estudios en idioma inglés, (I2) Estudios que presentan herramientas de evaluación de accesibilidad web, (I3) Estudios del área de ingeniería del software. Al final de este proceso de selección, se obtiene un total de 50 artículos, tal como se muestra en la Figura 1.

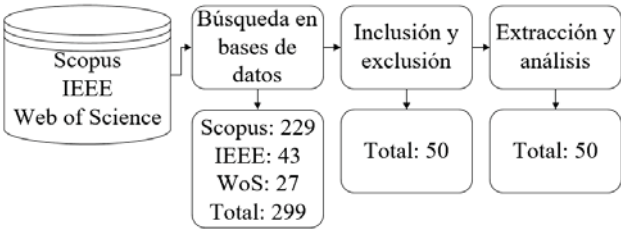


Fig. 1. Artículos incluidos durante el proceso de selección.

b) Evaluación de la calidad: La evaluación de la calidad de los artículos se llevó a cabo con el fin de identificar el nivel de detalle ofrecido sobre los aspectos de interés del análisis (especialmente relacionados con la descripción de las herramientas). El puntaje se asignó en una escala de 0 a 2, donde 0=No cumple el criterio en lo absoluto, 1=Cumple con el criterio parcialmente, 2=Cumple con el criterio totalmente. Los estudios fueron evaluados de acuerdo a los siguientes criterios de calidad: (Q1) ¿Está el objetivo de investigación claramente definido? (Q2) ¿El artículo presenta métricas para validar los resultados del estudio? (Q3) ¿Están las herramientas utilizadas descritas y su elección justificada? Los valores de calidad obtenidos por los estudio variaron entre 0 y 6, con una mediana de 5 y un promedio de 4,78, lo que indica que los estudios analizados proporcionan un nivel de detalle deseable de acuerdo a los criterios de calidad.

c) Extracción de datos, análisis y clasificación: Para cada estudio seleccionado, se extrajo la información relevante para analizar las preguntas de investigación. Los componentes del formulario de extracción incluyeron información de las herramientas (RQ1), criterios de accesibilidad evaluados (RQ2), y desafíos reportados (RQ3). La extracción fue realizada por la investigadora principal del estudio (primera autora). Para analizar y sintetizar la información, procedimos a tabular los resultados por pregunta y a realizar un análisis narrativo que resumía y describía los resultados encontrados. Los estudios relacionados a una sola herramienta fueron contabilizados por separado.

d) Amenazas a la validez: La cadena de búsqueda se construyó con base en los artículos de control, y fue refinada a través de varias iteraciones para disminuir el ruido. La palabra clave WCAG en la cadena de búsqueda puede sesgar los estándares identificados en el estudio. Las bases de datos utilizadas son reconocidas por su gran cobertura en el campos de la ingeniería de software. Cuando hubo dudas sobre si incluir un artículo o no, se procedió a su lectura completa. Excluimos literatura gris y estudios cuyo idioma no era inglés. El proceso de selección, clasificación y extracción lo realizó un solo investigador. La interpretación de los resultados también es una amenaza a la validez. Los estudios se clasificaron según lo reportado por los autores y, cuando no se reportó explícitamente, los investigadores de este estudio intentaron hacer una clasificación. Se diseñó un formulario de extracción para recolectar los datos, el cual guió el proceso y podría revisarse. Los criterios de calidad fueron aplicados por un solo investigador, representando una amenaza a la validez. Los resultados solo son generalizables en el contexto de los estudios incluidos en el mapeo. Durante todo el proceso, se aplicaron protocolos de estudios secundarios. Reportamos el proceso con el fin de facilitar el análisis y la replicación.

IV. Resultados

Aquí presentamos los resultados del mapeo de literatura, con base en 50 estudios primarios analizados. Los artículos identificados realizan análisis en diferentes dominios de negocio, entre los que estaban sitios web del gobierno (18, 36%), universidades (13, 26%), industria (7, 14%), salud (2, 4%) y otros no reportados (10, 20%).

Las publicaciones identificadas se realizan entre los años 2004 y 2019. El año 2018 cuenta con la mayor cantidad de artículos (10), seguido por el 2016 (8), 2017 (7) y 2019 (6), lo que indica la vigencia de las publicaciones en el área. Finalmente, los estudios se realizan en distintos países y regiones tales como América Latina, Africa, Australia, Azerbaijan, Bangladesh, China, Chipre, Estados Unidos, Ghana, India, Indonesia, Japón, Libia, Kazakhstan, Kyrgyzstan, Malasia, México, Nepal, Noruega, Países Arabes, Pakistán, Portugal, Sri Lanka y Turquía por lo que los resultados indican que el interés en el área es a nivel global y que existe interés mejorar el tema de la accesibilidad de los sitios web. Seguidamente presentamos los resultados que responden a cada pregunta de investigación.

A. RQ1. Herramientas utilizadas para evaluar la accesibilidad web

La Tabla I lista las herramientas identificadas en más de dos artículos. Un total de 39 herramientas fueron utilizadas para realizar la evaluación de la accesibilidad de sitios web. De las herramientas reportadas, Achecker (https://achecker.ca/), TAW (https://www.tawdis.net/) y WAVE (https://wave.webaim.org/) son las más utilizadas para evaluar la accesibilidad web. Estas herramientas son *free services* y se encuentran disponibles a través de sus sitios web.

Las herramientas reportadas en solo un artículo son: aDesigner [68], AMP [29], Checorser [30], Code Sniffer [67],

TABLE I
HERRAMIENTAS PARA LA EVALUACIÓN DE LA ACCESIBILIDAD WEB.

Herramienta	Referencia	Cant.
Achecker	[22]–[42]	21
TAW	[22], [25], [29], [36], [37], [41], [43]–[51]	15
Wave	[24], [25], [33], [34], [36], [38], [40], [41], [52]–[58]	15
SortSite	[29], [41], [59]–[61]	5
TotalValidator	[29], [62]	2
AccessColor	[23], [42]	2
aXe	[24], [49]	2
Bobby	[63], [64]	2
ContrastFinder	[23], [42]	2
EIII – Page Checker	[23], [42]	2
EvalAccess	[39], [65]	2
Hera	[38], [66]	2
HTML Tidy	[43], [67]	2
Tanaguru	[23], [42]	2
Tenon	[55], [57]	2
Web Accessibility Toolbar	[23], [42]	2

CORrector [30], Cynthia Says [54], Deque [29], Evaluator [62], google Mobile-Friendly Test [58], googlePageSpeed insight [58], iteimprove Accessibility Checker [57], Koa11y [56], Markup Validation Service [39], OpenWAX [57], Page Checker [28], Pingdom tool [58], PowerMapper [58], Siteimprove Accessibility Checker [69], TinyMCE [70], WaaT [71], Web Accessibility Assessment Tool [22], Webpage Analyzer [40], Worldspace FireEyse [22].

La Figura 2 muestra las cuatro herramientas de evaluación de accesibilidad más utilizadas y su tendencia por año. Desde el 2015, estas herramientas han sido regularmente reportadas en las evaluaciones realizadas por los estudios de accesibilidad web. Es importante notar que existen estudios que proponen nuevas herramientas a partir de la extensión de herramientas existentes [36], [64], [70].

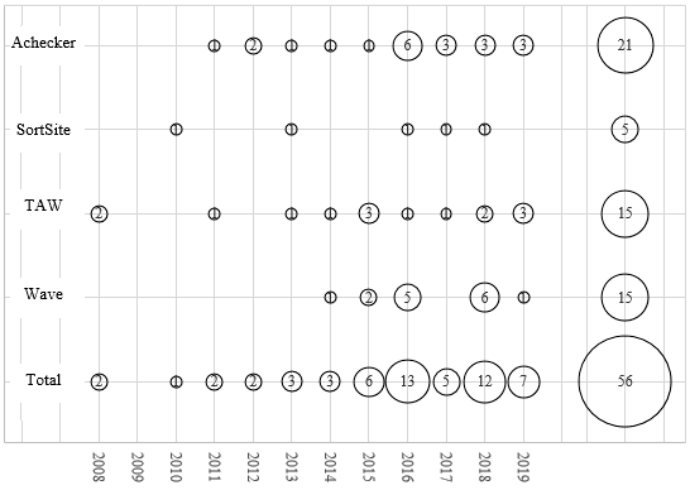


Fig. 2. Herramientas más utilizadas por año.

B. RQ2. Criterios de accesibilidad evaluados con las herramientas

La mayoría de los estudios evalúan el criterio de conformidad Nivel A, el cual está compuesto por 25 aspectos. Esto se debe a que los de conformidad Nivel AA y AAA son más difíciles de cumplir. Del total de artículos, 40 evaluaron el estándar WCAG 2.0, 5 artículos evaluaron el estándar WCAG 1.0, 3 artículos evaluaron el estándar WCAG 2.0 y Section 508, 2 artículos evaluaron el estándar WCAG 2.0 y WCAG 1.0. Finalmente, 8 artículos no indicaron la evaluación de un estándar específico. Cuando se realizó una evaluación de accesibilidad, el 56% (28) de los estudios evaluaron los tres niveles de éxito: A, AA, AAA.

Las Tabla II lista los aspectos reportados por las herramientas de evaluación de accesibilidad. El 40% de los artículos reportaron los errores encontrados durante la ejecución de los estudios. La agrupación de los aspectos de accesibilidad se recolectaron de acuerdo a los criterios de conformidad por cada uno de los niveles de éxito: A, AA, AAA. Los resultados muestran conjuntos de criterios no cumplidos que se repiten en la mayoría de las evaluaciones, por lo que con estos resultados es evidente la importancia de dar seguimiento estas debilidades encontradas en los sitios web.

Es importante notar que los aspectos de accesibilidad no reportados podrían indicar que estos no pueden ser evaluados automáticamente y que aun requieren la evaluación por parte de un humano. En el caso del Nivel A, estos aspectos son los relacionados con 1.4.2 Control del audio y 1.2.3 Audio descripción o Medio Alternativo. En el caso del Nivel AA, el 1.2.4 Subtítulos (en directo), 3.3.3 Sugerencias ante errores, 3.2.4 Identificación coherente, 1.2.5 Audio descripción (grabado), 3.3.4 Prevención de errores (legales, financieros, datos), 3.1.2 Idioma de las partes, 2.4.5 Múltiples vías. Finalmente, del Nivel AAA, el 1.4.7 Sonido de fondo bajo o ausente, 3.1.6 Pronunciación, 3.1.4 Abreviaturas, 2.2.3 Sin tiempo, 3.3.5 Ayuda, 3.3.6 Prevención de errores (todos), 3.1.3 Palabras inusuales, 1.2.6 Lengua de señas (grabado), 3.1.5 Nivel de lectura, 2.3.2 Tres destellos, 3.2.5 Cambios a petición, 2.4.10 Encabezados de sección, 1.2.7 Audio descripción ampliada (grabada), 2.4.8 Ubicación, y 2.2.5 Re-autenticación.

C. RQ3. Desafíos de las evaluaciones de la accesibilidad web

Los estudios reportan los desafíos encontrados relacionados con la accesibilidad web que se presentan al realizar las evaluaciones. Se identifican dos categorías en las cuales se agrupan los desafíos: aspectos técnicos y aspectos sobre regulaciones de gobierno. Los aspectos técnicos principales son el (1) lenguaje variable de los sitios web [22], (2) la disponibilidad del personal de desarrollo web capacitado en accesibilidad [24], [39], [40], [68], (3) la combinación de varios métodos simultáneos para evaluar la accesibilidad de sitios web considerando métricas y heurísticas [37], [53], [57], (4) son requeridos evaluadores expertos en accesibilidad web [25], [34], [52], y finalmente (6) durante la definición de los aspectos del diseño de sitios web se deben considerar los criterios de la accesibilidad [42], [55], [69].

TABLE II
ERRORES COMÚNMENTE REPORTADOS POR LOS ESTUDIOS.

Nivel	Aspecto	Referencia	Cant.
A	1.1.1 Contenido no textual	[22], [25]–[27], [29], [30], [34], [37], [38], [40], [42], [46], [50], [51], [53], [55], [57], [63]–[65]	20
A	2.4.4 Propósito de los enlaces (en contexto)	[22], [25]–[27], [29], [37], [38], [40], [42], [46], [50], [53], [57], [68]	14
A	3.1.1 Idioma de la página	[25]–[27], [34], [37], [38], [40], [44], [46], [50], [57]	11
A	1.3.1 Información y relaciones	[25]–[27], [29], [34], [37], [40], [42], [44], [46], [57]	11
A	3.3.2 Etiquetas o instrucciones	[25], [26], [38], [40], [46], [50], [57]	7
A	1.4.1 Uso del color	[26], [34], [38], [40], [57]	6
A	4.1.1 Procesamiento	[22], [34], [40], [42], [46], [57]	5
A	4.1.2 Nombre, función, valor	[40], [44], [46], [55], [57]	5
A	2.4.1 Evitar bloques	[34], [40], [57]	4
A	2.1.1 Teclado	[22], [26], [34], [57]	4
A	2.2.1 Tiempo ajustable	[34], [40], [57]	3
A	1.3.3 Características sensoriales	[34], [40]	2
A	3.2.1 Al recibir el foco	[34], [40]	2
A	2.3.1 Umbral de tres destellos o menos	[34], [40]	2
A	3.3.1 Identificación de errores	[25], [34]	2
A	1.3.2 Secuencia significativa	[42], [57]	2
A	2.4.3 Orden del foco	[42], [57]	2
A	2.4.2 Titulado de páginas	[46], [57]	2
A	1.2.1 Sólo audio y sólo vídeo (grabado)	[34], [42]	2
A	3.2.2 Al recibir entradas	[46], [57]	2
A	2.2.2 Poner en pausa, detener, ocultar	[26]	1
A	1.2.2 Subtítulos (grabados)	[42]	1
A	2.1.2 Sin trampas para el foco del teclado	[42]	1
	Total		110
AA	1.4.3 Contraste (mínimo)	[25], [27], [29], [30], [57]	5
AA	2.4.6 Encabezados y etiquetas	[26], [37], [52], [57], [68]	5
AA	1.4.5 Imágenes de texto	[24], [38], [52], [57]	4
AA	1.4.4 Cambio de tamaño del texto	[26], [29], [30], [57]	4
AA	3.2.3 Navegación coherente	[40]	1
AA	2.4.7 Foco visible	[57]	1
	Total		20
AAA	2.4.9 Propósito de los enlaces (sólo enlaces)	[24], [44], [52], [57]	4
AAA	1.4.6 Contraste (mejorado)	[26], [36], [38]	3
AAA	1.4.9 Imágenes de texto (sin excepciones)	[24], [36]	2
AAA	2.1.3 Teclado (sin excepciones)	[36], [68]	2
AAA	1.2.9 Sólo audio (en directo)	[57]	1
AAA	1.2.8 Medio alternativo (grabado)	[68]	1
AAA	1.4.8 Presentación visual	[68]	1
AAA	2.2.4 Interrupciones	[29]	1
	Total		15

Con respecto a las regulaciones de gobierno, (1) se debe hacer que la accesibilidad de los sitios web gubernamentales sea un requisito obligatorio [30], [39], [47], [59], (2) se deben establecer normativas de cumplimiento de accesibilidad [32], [33], [54], [62], [63], y (3) se deben fortalecer y compartir las políticas de accesibilidad web de cada país, así como aplicar mejores leyes y fomentar prácticas que hagan a los sitios web sean más accesibles [26], [53], [59]. En muchos casos, las regulaciones existentes no son requisitos obligatorios y no todos los gobiernos establecen reglamentos para exigir formalmente el cumplimiento de la accesibilidad web. Finalmente, (4) los responsables de las políticas deben desarrollar y promover marcos legales para abordar los problemas de accesibilidad web [41], [45].

V. DISCUSIÓN

La evaluación de la accesibilidad web se está convirtiendo en una necesidad apremiante de cualquier sitio web, independientemente del área de aplicación. Uno de los principales retos se da con la estandarización de la normativa y regulaciones por parte de los responsables, y herramientas que permitan apoyar los procesos de evaluación de los sitios web.

Los resultados muestran que existe una gran cantidad de herramientas para evaluar la accesibilidad web pero que en la práctica dichas herramientas no son usadas por los desarrolladores. Se requiere una regulación que exija el cumplimiento de la accesibilidad en sitios web para que los responsables de su desarrollo lo consideren esencial durante el proceso de desarrollo. Las evaluaciones de accesibilidad deben realizarse periódicamente, ya que por lo general estas pruebas se realizan al final del ciclo de desarrollo, lo que puede provocar retrabajo. Las herramientas de evaluación de accesibilidad pueden apoyar estos procesos de validación.

VI. CONCLUSIONES

Los resultados indican que existen herramientas capaces de evaluar aspectos de la accesibilidad web, y algunas de ellas pueden identificar errores encontrados a partir de los criterios de la normativa WCAG (versión 1.0 y 2.0). Sin embargo, algunos aspectos de accesibilidad aún deben ser evaluados manualmente. Estas herramientas pueden apoyar el proceso de evaluación de la accesibilidad web pero no pueden reemplazar las revisiones de los expertos en accesibilidad web.

Este es un trabajo en progreso que busca identificar los aspectos de accesibilidad web que pueden ser evaluados mediante herramientas automatizadas. Para ello es necesario realizar evaluaciones objetivas sobre sitios web controlados, que permitan determinar el nivel de cumplimiento de los aspectos planteados por los estándares de accesibilidad web, y cómo las herramientas los identifican y reportan.

REFERENCES

[1] WCAG. Web content accessibility guidelines (wcag) overview. [Online]. Available: https://www.w3.org/WAI/standards-guidelines/wcag/

[2] W. A. Initiative. Introduction to web accessibility. [Online]. Available: https://www.w3.org/WAI/fundamentals/accessibility-intro/what

[3] UNESCO. The world summit on the information society forum. [Online]. Available: https://en.unesco.org/events/wsis-forum-2019

[4] H. Kharas. Smart cities have an opportunity to become far more inclusive. The World Economic Forum. [Online]. Available: https://www.weforum.org/agenda/2018/06/can-smart-cities-be-equitable/

[5] K. Molina and F. Cuevas, "Tic y educación de personas con discapacidad en Costa Rica," *PROSIC*, 2014.

[6] A. Quirós-Ramírez, "Las nuevas tecnologías y la calidad de vida de personas con discapacidad en costa rica," *PROSIC*, 2017.

[7] W3C. World wide web consortium. [Online]. Available: http://www.w3.org/

[8] V. L. Centeno, C. D. Kloos, J. A. Fisteus, and L. Á. Álvarez, "Web accessibility evaluation tools: A survey and some improvements," *Electronic notes in theoretical computer science*, vol. 157, no. 2, pp. 87–100, 2006.

[9] M. Tollefsen and T. Ausland, "A practitioner's approach to using wcag evaluation tools," in *2017 6th International Conference on Information and Communication Technology and Accessibility (ICTA)*. IEEE, 2017, pp. 1–5.

[10] H. Petrie, A. Savva, and C. Power, "Towards a unified definition of web accessibility," in *Proceedings of the 12th Web for all Conference*. ACM, 2015, p. 35.

[11] I. Elkabani, L. Hamandi, R. Zantout, and S. Mansi, "Toward better web accessibility," in *2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)*. IEEE, 2015, pp. 1–6.

[12] WCAG. Web content accessibility guidelines (wcag) 2.1. [Online]. Available: https://www.w3.org/TR/WCAG21/

[13] N. M., P. Chawla, and A. Rana, "A practitioner's approach to assess the wcag 2.0 website accessibility challenges," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Feb 2019, pp. 958–966.

[14] M. Kirchner, "Evaluation, repair, and transformation of web pages for web content accessibility. review of some available tools," in *Proceedings. Fourth International Workshop on Web Site Evolution*, Oct 2002, pp. 65–72.

[15] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[16] B. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," Technical report, Ver. 2.3 EBSE Technical Report. EBSE, Tech. Rep., 2007.

[17] J. Biolchini, P. G. Mian, A. C. C. Natali, and G. H. Travassos, "Systematic review in software engineering," *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES*, vol. 679, no. 05, p. 45, 2005.

[18] V. Basili, C. Gianluigi, and D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

[19] W. A. R. W. M. Isa, A. I. H. Suhaimi, N. Ariffrn, N. F. Ishak, and N. M. Ralim, "Accessibility evaluation using web content accessibility guidelines (wcag) 2.0," in *2016 4th International Conference on User Science and Engineering (i-USEr)*. IEEE, 2016, pp. 1–4.

[20] T. Acosta, P. Acosta-Vargas, and S. Luján-Mora, "Accessibility of egovernment services in latin america," in *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*. IEEE, 2018, pp. 67–74.

[21] P. Acosta-Vargas, T. Acosta, and S. Luján-Mora, "Challenges to assess accessibility in higher education websites: A comparative study of latin america universities," *IEEE Access*, vol. 6, pp. 36 500–36 508, 2018.

[22] H. S. Al-Khalifa, M. Al-Kanhal, H. Al-Nafisah, N. Al-soukaih, E. Alhussain, and M. Al-onzi, "A pilot study for evaluating arabic websites using automated wcag 2.0 evaluation tools," in *2011 International Conference on Innovations in Information Technology*. IEEE, 2011, pp. 293–296.

[23] K. Wille, C. Wille, and R. Dumke, "A test procedure for checking the wcag 2.0 guidelines," in *International Conference on Universal Access in Human-Computer Interaction*. Springer, 2016, pp. 120–131.

[24] A. Ismail, K. Kuppusamy, and S. Paiva, "Accessibility analysis of higher education institution websites of portugal," *Universal Access in the Information Society*, pp. 1–16, 2019.

[25] S. U. Dongaonkar, R. S. Vadali, and C. Dhutadmal, "Accessibility analyzer: tool for new adaptations in government web applications to improve accessibility," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*. IEEE, 2017, pp. 1–5.

[26] R. Ismailova and Y. Inal, "Accessibility evaluation of top university websites: a comparative study of kyrgyzstan, azerbaijan, kazakhstan and turkey," *Universal Access in the Information Society*, vol. 17, no. 2, pp. 437–445, 2018.

[27] W. A. R. W. M. Isa, A. I. H. Suhaimi, N. Ariffrn, N. F. Ishak, and N. M. Ralim, "Accessibility evaluation using web content accessibility guidelines (wcag) 2.0," in *2016 4th International Conference on User Science and Engineering (i-USEr)*. IEEE, 2016, pp. 1–4.

[28] E. İ. İşeri, K. Uyar, and Ü. İlhan, "Accessibility of the cyprus island municipal websites," in *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 2017, pp. 72–76.

[29] M. Vigo, J. Brown, and V. Conway, "Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests," in *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. ACM, 2013, p. 1.

[30] D. T. Tuan, V.-H. Phan *et al.*, "Checking and correcting the source code of web pages for accessibility," in *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future*. IEEE, 2012, pp. 1–4.

[31] Y. Inal and R. Ismailova, "Effect of human development level of countries on the web accessibility and quality in use of their municipality websites," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2019.

[32] M. Dollie and S. Kabanda, "E-government in africa: Perceived concerns of persons with disabilities (pwds) in south africa," in *The Proceedings of 17th European Conference on Digital Government ECDG 2017*, 2017, p. 63.

[33] A. Ahmi and R. Mohamad, "Evaluating accessibility of malaysian ministries websites using wcag 2.0 and section 508 guideline," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 8, no. 8, pp. 177–183, 2016.

[34] ——, "Evaluating accessibility of malaysian public universities websites using achecker and wave," *Journal of Information and Communication Technology*, vol. 15, no. 2, pp. 193–214, 2016.

[35] I. G. B. N. E. Darmaputra, S. S. Wijaya, and M. A. Ayu, "Evaluating the accessibility of provinces'e-government websites in indonesia," in *2017 5th International Conference on Cyber and IT Service Management (CITSM)*. IEEE, 2017, pp. 1–6.

[36] I. Elkabani, L. Hamandi, R. Zantout, and S. Mansi, "Toward better web accessibility," in *2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)*. IEEE, 2015, pp. 1–6.

[37] N. A. Karaim and Y. Inal, "Usability and accessibility evaluation of libyan government websites," *Universal Access in the Information Society*, vol. 18, no. 1, pp. 207–216, 2019.

[38] S. Adepoju and I. Shehu, "Usability evaluation of academic websites using automated tools," in *2014 3rd International Conference on User Science and Engineering (i-USEr)*. IEEE, 2014, pp. 186–191.

[39] M. K. Baowaly and M. Bhuiyan, "Accessibility analysis and evaluation of bangladesh government websites," in *2012 International Conference on Informatics, Electronics & Vision (ICIEV)*. IEEE, 2012, pp. 46–51.

[40] A. Ismail and K. Kuppusamy, "Accessibility of indian universities' homepages: An exploratory study," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 2, pp. 268–278, 2018.

[41] Y. M. Tashtoush, D. Ala'F, and H. N. Al-Sarhan, "The arabian e-government websites accessibility: a case study," in *2016 7th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2016, pp. 276–281.

[42] K. Wille, R. R. Dumke, and C. Wille, "Measuring the accessability based on web content accessibility guidelines," in *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*. IEEE, 2016, pp. 164–169.

[43] J. M. Fernández, V. Soler, and J. Roig, "A new approach to the automatic web accessibility." in *ICEIS (5)*, 2008, pp. 393–396.

[44] P. Windriyani, R. Ferdiana, and W. Najib, "Accessibility evaluation using wcag 2.0 guidelines webometrics based assessment criteria (case study: Sebelas maret university)," in *2014 International Conference on ICT For Smart Society (ICISS)*. IEEE, 2014, pp. 305–311.

[45] W. Yaokumah, S. Brown, and R. Amponsah, "Accessibility, quality and performance of government portals and ministry web sites: a view using diagnostic tools," in *2015 Annual Global Online Conference on Information and Computer Technology (GOCICT)*. IEEE, 2015, pp. 46–50.

[46] W. Arasid, A. Abdullah, D. Wahyudin, C. Abdullah, I. Widiaty, D. Zakaria, N. Amelia, and A. Juhana, "An analysis of website accessibility in higher education in indonesia based on wcag 2.0 guidelines," in *IOP Conference Series: Materials Science and Engineering*, vol. 306, no. 1. IOP Publishing, 2018, p. 012130.

[47] R. L. Ochoa and D. M. Crovi, "Evaluation of accessibility in mexican cybermedia," *Universal Access in the Information Society*, vol. 18, no. 2, pp. 413–422, 2019.

[48] P. Acosta-Vargas, S. Luján-Mora, and L. Salvador-Ullauri, "Quality evaluation of government websites," in *2017 Fourth International Conference on eDemocracy & eGovernment (ICEDEG)*. IEEE, 2017, pp. 8–14.

[49] A. Ismail and K. Kuppusamy, "Web accessibility investigation and identification of major issues of higher education websites with statistical measures: A case study of college websites," *Journal of King Saud University-Computer and Information Sciences*, 2019.

[50] Y. Akgül, "Web content accessibility of municipal web sites in turkey," in *European Conference on e-Government*. Academic Conferences International Limited, 2015, p. 1.

[51] H. Jati and D. D. Dominic, "Website accessibility performance evaluation in malaysia," in *2008 International Symposium on Information Technology*, vol. 1. IEEE, 2008, pp. 1–3.

[52] T. Acosta, P. Acosta-Vargas, and S. Luján-Mora, "Accessibility of egovernment services in latin america," in *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*. IEEE, 2018, pp. 67–74.

[53] P. Acosta-Vargas, T. Acosta, and S. Luján-Mora, "Challenges to assess accessibility in higher education websites: A comparative study of latin america universities," *IEEE Access*, vol. 6, pp. 36 500–36 508, 2018.

[54] B. A. Shawar, "Evaluating web accessibility of educational websites," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 10, no. 4, pp. 4–10, 2015.

[55] P. Acosta-Vargas, T. Acosta, and S. Luján-Mora, "Framework for accessibility evaluation of hospital websites," in *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*. IEEE, 2018, pp. 9–15.

[56] P. Acosta-Vargas, S. Luján-Mora, T. Acosta, and L. Salvador-Ullauri, "Toward a combined method for evaluation of web accessibility," in *International Conference on Information Theoretic Security*. Springer, 2018, pp. 602–613.

[57] P. Acosta-Vargas, Y. Rybarczyk, J. Pérez, M. González, K. Jimenes, L. Leconte, and D. Esparza, "Towards web accessibility in telerehabilitation platforms," in *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*. IEEE, 2018, pp. 1–6.

[58] S. Gopinath, V. Senthooran, N. Lojenaa, and T. Kartheeswaran, "Usability and accessibility analysis of selected government websites in sri lanka," in *2016 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2016, pp. 394–398.

[59] E. Agbozo and K. Spassov, "Evaluating metropolitan assembly web sites in ghana: Accessibility, compatibility and usability." *Webology*, vol. 15, no. 1, 2018.

[60] Z. Yerlikaya and P. O. Durdu, "Evaluation of accessibility of university websites: A case from turkey," in *International Conference on Human-Computer Interaction*. Springer, 2017, pp. 663–668.

[61] R. Gonçalves, J. Martins, M. Martins, J. Pereira, and H. São Mamede, "Portuguese web accessibility snapshot-status of the portuguese websites regarding accessibility levels." in *ICEIS (5)*, 2010, pp. 223–226.

[62] M. Bakhsh and A. Mehmood, "Web accessibility for disabled: a case study of government websites in pakistan," in *2012 10th International Conference on Frontiers of Information Technology*. IEEE, 2012, pp. 342–347.

[63] B. P. Shah and S. Shakya, "Evaluating the web accessibility of websites of the central government of nepal," in *Proceedings of the 1st international conference on Theory and practice of electronic governance*. ACM, 2007, pp. 447–448.

[64] Y. Takata, T. Nakamura, and H. Seki, "Accessibility verification of www documents by an automatic guideline verification tool," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*. IEEE, 2004, pp. 10–pp.

[65] R. Ismailova and G. Kimsanova, "Universities of the kyrgyz republic on the web: accessibility and usability," *Universal Access in the Information Society*, vol. 16, no. 4, pp. 1017–1025, 2017.

[66] P.-L. P. Rau, L. Zhou, N. Sun, and R. Zhong, "Evaluation of web accessibility in china: changes from 2009 to 2013," *Universal Access in the Information Society*, vol. 15, no. 2, pp. 297–303, 2016.

[67] I. N. Ikhsan and M. Z. C. Candra, "Automatically: An automated refactoring method and tool for improving web accessibility," in *2018 5th International Conference on Data and Software Engineering (ICoDSE)*. IEEE, 2018, pp. 1–6.

[68] M. A. Agangiba, E. B. Nketiah, and W. A. Agangiba, "Web accessibility for the visually impaired: A case of higher education institutions' websites in ghana," in *International Conference on Web-Based Learning*. Springer, 2017, pp. 147–153.

[69] V. Stray, A. Bai, N. Sverdrup, and H. Mork, "Empowering agile project members with accessibility testing tools: a case study," in *International Conference on Agile Software Development*. Springer, 2019, pp. 86–101.

[70] C. Avila, S. Baldiris, R. Fabregat, and J. C. Guevara, "Accessibility evaluation improvement using case based reasoning," in *2012 Frontiers in Education Conference Proceedings*. IEEE, 2012, pp. 1–6.

[71] N. Fernandes, N. Kaklanis, K. Votis, D. Tzovaras, and L. Carriço, "An analysis of personalized web accessibility," in *Proceedings of the 11th Web for All Conference*. ACM, 2014, p. 19.

Bibliografía del capítulo

[1] W3C, "World wide web consortium." http://www.w3.org/.

[2] UNESCO, "The world summit on the information society forum." https://en.unesco.org/events/wsis-forum-2019.

[3] H. Kharas, "Smart cities have an opportunity to become far more inclusive. the world economic forum." https://www.weforum.org/agenda/2018/06/can-smart-cities-be-equitable/.

[4] K. Molina and F. Cuevas, "Tic y educación de personas con discapacidad en costa rica," *PROSIC*, 2014.

[5] A. Quirós-Ramírez, "Las nuevas tecnologías y la calidad de vida de personas con discapacidad en costa rica,"," *PROSIC*, 2014.

[6] CI, UCR, "Uit ofrecerá conferencia en la ucr sobre accesibilidad web." https://www.w3.org/WAI/fundamentals/accessibility-intro/.

[7] M. Tollefsen and T. Ausland, "A practitioner's approach to using wcag evaluation tools," in *2017 6th International Conference on Information and Communication Technology and Accessibility (ICTA)*, pp. 1–5, Dec 2017.

[8] H. Petrie, A. Savva, and C. Power, "Towards a unified definition of web accessibility," pp. 1–13, 2015.

[9] WCAG, "Web content accessibility guidelines (wcag) overview." http://www.w3.org/.

[10] V. Luque, C. Delgado, J. Arias, and L. Alvarez, "Web accessibility evaluation tools: A survey and some improvements," *Electronic Notes in Theoretical Computer Science*, vol. 157, no. 2 SPEC. ISS., pp. 87–100, 2006.

[11] Web Accessibility Initiative, "Descripción general de los estándares de accesibilidad del w3c." https://www.w3.org/WAI/about/#world-wide-web-consortium-w3c-web-accessibility-initiative/wai.

[12] Web Accessibility Initiative, "Introducción a las pautas de accesibilidad para el contenido web (wcag)." https://www.w3.org/WAI/standards-guidelines/wcag/es.

[13] W3C, "Introduction to understanding wcag 2.1."

[14] MIT , ERCIM , Keio , Beihang, "Entendiendo wcag 2.0." https://www.w3.org/TR/WCAG20/.

[15] O. Carreras, "Mapa visual de las wcag 2.0." https://olgacarreras.blogspot.com/2009/04/mapa-conceptual-de-las-wcag-20.html.

[16] I. Elkabani, L. Hamandi, R. Zantout, and S. Mansi, "Toward better web accessibility," *2015 5th International Conference on Information and Communication Technology and Accessibility, ICTA 2015.*, no. 6, 2016.

[17] P. Chawla and A. Rana, "A practitioner's approach to assess the wcag 2.0 website accessibility challenges," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 958–966, Feb 2019.

[18] M. Kirchner, "Evaluation, repair, and transformation of web pages for web content accessibility. review of some available tools," in *Proceedings. Fourth International Workshop on Web Site Evolution*, pp. 65–72, Oct 2002.

[19] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[20] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3,” *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.

[21] J. Biolchini, P. Mian, A. Natali, and G. Travassos, “Potential for electricity generation from bagasse in Kenya,” *International Sugar Journal*, vol. 107, no. 1273, pp. 32–37, 2005.

[22] T. Acosta, P. Acosta-Vargas, and S. Lujan-Mora, “Accessibility of eGovernment Services in Latin America,” *2018 5th International Conference on eDemocracy and eGovernment, ICEDEG 2018*, pp. 67–74, 2018.

[23] W. Isa, A. Suhaimi, N. Ariffrn, N. Ishak, and N. Ralim, “Accessibility evaluation using web content accessibility guidelines (wcag) 2.0,” in *2016 4th International Conference on User Science and Engineering (i-USEr)*, pp. 1–4, Aug 2016.

[24] P. Acosta-Vargas, T. Acosta, and S. Lujan-Mora, “Challenges to assess accessibility in higher education websites: A comparative study of Latin america universities,” *IEEE Access*, vol. 6, pp. 36500–36508, 2018.

[25] B. Kitchenham, E. Mendes, and G. Travassos, *A Systematic Review of Cross- vs. Within-Company Cost Estimation Studies*. IEEE Trans on SE, 2007.

[26] D. Buenaño-Fernández, T. Acosta, and S. Luján-Mora, “The use of la to evaluate the performance of students with visual disabilities when applying accessibility criteria in online courses,” 11 2018.

[27] L. Amaral, R. Fortes, and T. Bittar, “A4u - an approach to evaluation considering accessibility and usability guidelines,” pp. 295–298, 10 2018.

[28] T. Tangarife and C. Montalvão, “Estudo comparativo utilizando uma ferramenta de avaliação de acessibilidade para web,” 01 2005.

[29] M. Ocampo, E. Leal, J. Cadavid, D. Gómez, and N. Méndez, “Application of a technical tool to support the inclusion process of people with visual impairment in an educational web platform,” pp. 1–5, Oct 2016.

[30] J. López de la Fuente, "Merkur: Herramienta de transcodificación parametrizada de contenidos web para móviles," *Profesional De La Informacion - PROF INF*, vol. 18, pp. 218–222, 07 2009.

[31] D. Buenaño-Fernández and S. Luján-Mora, "Proposal of application of process mining to evaluate the learning routes of students with visual disability in online courses," *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, pp. 1035–1047, 01 2019.

[32] T. Bittar, L. Lobato, D. Neto, and R. Fortes, "Support for collaboration in wikis using graphical modeling to achieve improvements in information architecture and accessibility," vol. 2, pp. 12 – 15, 11 2010.

[33] M. Baowaly and M. Bhuiyan, "Accessibility analysis and evaluation of Bangladesh government websites," *2012 International Conference on Informatics, Electronics and Vision, ICIEV 2012*, pp. 46–51, 2012.

[34] A. Ismail and K. Kuppusamy, "Accessibility of Indian universities' homepages: An exploratory study," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 2, pp. 268–278, 2018.

[35] P. Acosta-Vargas, S. Luján-Mora, T. Acosta, and L. Salvador-Ullauri, "Toward a combined method for evaluation of web accessibility," in *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*, (Cham), pp. 602–613, Springer International Publishing, 2018.

[36] P. Acosta-Vargas, Y. Rybarczyk, J. Perez, M. Gonzalez, K. Jimenes, L. Leconte, and D. Esparza, "Towards Web Accessibility in Telerehabilitation Platforms," *2018 IEEE 3rd Ecuador Technical Chapters Meeting, ETCM 2018*, pp. 1–6, 2018.

[37] M. Agangiba, E. Nketiah, and W. Agangiba, "Web accessibility for the visually impaired: A case of higher education institutions' websites in ghana," vol. 11007, pp. 147–153, 2018.

[38] M. Vigo, J. Brown, and V. Conway, "Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests," no. October, p. 1, 2013.

[39] T. Vu, "Checking and correcting the source code of web pages for accessibility," *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future*, pp. 1–4, 2012.

[40] I. Ikhsan and M. Catur Candra, "Automatically: An automated refactoring method and tool for improving web accessibility," in *2018 5th International Conference on Data and Software Engineering (ICoDSE)*, pp. 1–6, Nov 2018.

[41] B. Abu Shawar, "Evaluating Web Accessibility of Educational Websites," *International Journal of Emerging Technologies in Learning*, vol. 10, no. 4, pp. 4–10, 2015.

[42] M. Bakhsh and A. Mehmood, "Web accessibility for disabled: A case study of government websites in Pakistan," *Proceedings - 10th International Conference on Frontiers of Information Technology, FIT 2012*, pp. 342–347, 2012.

[43] S. Gopinath, V. Senthooran, N. Lojenaa, and T. Kartheeswaran, "Usability and accessibility analysis of selected government websites in Sri Lanka," *Proceedings - 2016 IEEE Region 10 Symposium, TENSYMP 2016*, pp. 394–398, 2016.

[44] E. Iseri, K. Uyar, and U. Ilhan, "Accessibility of the cyprus island municipal websites," in *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 72–76, Sep. 2017.

[45] C. Avila, S. Baldiris, and R. Fabregat, "Accessibility Evaluation Improvement using Case Based Reasoning," *2012 Frontiers in Education Conference Proceedings*, pp. 1–6, 2012.

[46] H. Al-Khalifa, M. Al-Kanhal, H. Al-Nafisah, N. Al-soukaih, E. Al-hussain, and M. Al-onzi, "A pilot study for evaluating arabic websites using automated wcag 2.0 evaluation tools," in *2011 International Conference on Innovations in Information Technology*, pp. 293–296, April 2011.

[47] K. Wille, C. Wille, and R. Dumke, "A Test Procedure for Checking the WCAG 2.0 Guidelinesl," vol. 9737, pp. 120–131, 2016.

[48] J. Roig, "A New Approach To the Automatic Web Accessibility," pp. 393–396, 2011.

[49] Y. Takata, "Accessibility verification of www documents by an automatic guideline verification tool," vol. 00, no. C, pp. 1–10, 2004.

[50] S. Dongaonkar, "Accessibility Analyzer : Tool for New Adaptations in Government Web Applications to Improve Accessibility," *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1–5, 2017.

[51] A. Ahmi and R. Mohamad, "Evaluating Accessibility of Malaysian Public Universities Websites using Achecker and Wave .," 2016.

[52] N. Karaim and Y. Inal, "Usability and accessibility evaluation of Libyan government websites," *Universal Access in the Information Society*, vol. 18, no. 1, pp. 207–216, 2019.

[53] S. Adepoju and I. Shehu, "Usability evaluation of academic websites using automated tools," *Proceedings - 2014 3rd International Conference on User Science and Engineering: Experience. Engineer. Engage, i-USEr 2014*, pp. 186–191, 2015.

[54] K. Wille, R. Dumke, and C. Wille, "Measuring the accessability based on web content accessibility guidelines," pp. 164–169, IEEE, 2017.

[55] W. Arasid, A. Abdullah, D. Wahyudin, C. Abdullah, I. Widiaty, D. Zakaria, N. Amelia, and A. Juhana, "An Analysis of Website Accessibility in Higher Education in Indonesia Based on WCAG 2.0 Guidelines," *IOP Conference Series: Materials Science and Engineering*, vol. 306, no. 1, pp. 0–8, 2018.

[56] Y. Akgül, "Web content accessibility of municipal web sites in Turkey," *Proceedings of the European Conference on e-Government, ECEG*, vol. 2015-January, no. September, pp. 1–8, 2015.

[57] H. Jati and D. Dominic, "Website accessibility performance evaluation in malaysia," in *2008 International Symposium on Information Technology*, vol. 1, pp. 1–3, Aug 2008.

[58] A. Ismail and K. Kuppusamy, "Web accessibility investigation and identification of major issues of higher education websites with statistical measures: A case

study of college websites," *Journal of King Saud University - Computer and Information Sciences*, 2019.

[59] P. Acosta-Vargas, "Framework for Accessibility Evaluation of Hospital Websites," *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*, pp. 9–15, 2018.

[60] B. Shah and S. Shakya, "Evaluating the web accessibility of websites of the central government of nepal," no. January 2007, p. 447, 2008.

[61] R. Ismailova, "Universities of the Kyrgyz Republic on the Web : accessibility and usability," *Universal Access in the Information Society*, vol. 16, no. 4, pp. 1017–1025, 2017.

[62] R. Ismailova and Y. Inal, "Accessibility evaluation of top university websites: a comparative study of Kyrgyzstan, Azerbaijan, Kazakhstan and Turkey," *Universal Access in the Information Society*, vol. 17, no. 2, pp. 437–445, 2018.

[63] Y. Inal and R. Ismailova, "Effect of human development level of countries on the web accessibility and quality in use of their municipality websites," *Journal of Ambient Intelligence and Humanized Computing*, no. March, 2019.

[64] P. Windriyani, R. Ferdiana, and W. Najib, "Accessibility evaluation using WCAG 2.0 guidelines webometrics based assessment criteria (case study: Sebelas Maret University)," *Proceedings - 2014 International Conference on ICT for Smart Society: "Smart System Platform Development for City and Society, GoeSmart 2014", ICISS 2014*, pp. 305–311, 2014.

[65] A. Ismail, K. Kuppusamy, and S. Paiva, "Accessibility analysis of higher education institution websites of Portugal," *Universal Access in the Information Society*, no. 0123456789, 2019.

[66] V. Stray, A. Bai, and N. Sverdrup, *Empowering Agile Project Members with Accessibility Testing Tools: A Case Study*, vol. 149. Springer International Publishing, 2013.

[67] R. Ochoa and D. Crovi, "Evaluation of accessibility in Mexican cybermedia," *Universal Access in the Information Society*, vol. 18, no. 2, pp. 1–10, 2018.

[68] Y. Tashtoush, A. Darabseh, and H. Al-Sarhan, "The Arabian E-government websites accessibility: A case study," *2016 7th International Conference on Information and Communication Systems, ICICS 2016*, pp. 276–281, 2016.

[69] W. Yaokumah, S. Brown, and R. Amponsah, "Accessibility, quality and performance of government portals and ministry web sites: a view using diagnostic tools," *Proceedings - 2015 Annual Global Online Conference on Information and Computer Technology, GOCICT 2015*, pp. 46–50, 2016.

[70] C. Paper and C. Town, "E-Government in Africa : Perceived Concerns of Persons with Disabilities (PWDs) in South Africa with regards to accessibilities of services," *17th European Conference on Digital Government ECDG 2017*, no. August, 2017.

[71] A. Ahmi and R. Mohamad, "Evaluating accessibility of Malaysian Ministries Websites using WCAG 2 . 0 and Section 508 Guideline," *Journal of Telecommunication, Electronic and Computing Engineering*, vol. 8, no. 8, pp. 177–183, 2008.

[72] E. Agbozo and K. Spassov, "Evaluating metropolitan assembly web sites in Ghana: Accessibility, compatibility and usability," *Webology*, vol. 15, no. 1, p. 5, 2018.

[73] N. Fernandes, N. Kaklanis, K. Votis, D. Tzovaras, and L. Carriço, "An analysis of personalized web accessibility," no. April, pp. 1–10, 2014.

[74] I. Darmaputra, S. Wijaya, and M. Ayu, "Evaluating the accessibility of provinces' e-government websites in indonesia," in *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–6, Aug 2017.

[75] Z. Yerlikaya and O. Durdu, "Evaluation of accessibility of university websites: A case from turkey," vol. 850, pp. 663–668, 2018.

[76] P. Rau, L. Zhou, N. Sun, and R. Zhong, "Evaluation of web accessibility in china: changes from 2009 to 2013," *Universal Access in the Information Society*, vol. 15, no. 2, pp. 297–303, 2016.

[77] R. Gonçalves, J. Martins, J. Pereira, and H. Mamede, "Portuguese web accessibility in electronic public procurement platforms," *5th Iberian Conference on Information Systems and Technologies*, pp. 1–5, 2010.

[78] P. Acosta-Vargas, S. Luján-Mora, and L. Salvador-Ullauri, "Quality evaluation of government websites," *2017 4th International Conference on eDemocracy and eGovernment, ICEDEG 2017*, pp. 8–14, 2017.

Capítulo 2

Herramientas para pruebas automatizadas de seguridad Web: un mapeo sistemático

Elizabeth Gamboa Bermúdez

2.1. Resumen

Contexto: actualmente, son frecuentes los ataques cibernéticos dirigidos a aplicaciones Web para robar la información y el dinero de los usuarios. Ante esta situación, se han creado herramientas para evaluar de forma automatizada la seguridad de las aplicaciones Web, estas permiten detectar las vulnerabilidades del sistema y prevenir de los ataques cibernéticos. **Objetivo:** identificar y conocer las herramientas que han sido utilizadas para evaluar de forma automatizada la seguridad de las aplicaciones Web y cómo se ha evaluado la efectividad de la herramienta. **Metodología:** la metodología de esta investigación consistió en realizar un mapeo sistemático de estudios primarios que involucran el tema de pruebas de la seguridad en aplicaciones Web. Las herramientas se clasificaron según los tipos de metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web. **Resultados:** se encontraron 63 estudios relevantes para esta investigación, que describen 66 herramientas para realizar pruebas automatizadas las cuales evalúan la

seguridad de las aplicaciones Web. Las herramientas se clasificaron según los tipos de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web. La categoría de pruebas para detectar vulnerabilidades más comunes fue la de *Input Validation Testing (4.8)* con 55 herramientas, seguido de las pruebas de *Configuration and Deployment Management Testing (4.3)*, *Session Management Testing (4.7)*, y *Client Side Testing (4.12)* con 15 herramientas utilizadas cada una. Los tipos de pruebas más reportados fueron los de la categoría *Input Validation Testing (4.8)*. En este caso *SQL Injection (4.8.5)* con 40 herramientas, *Cross-Site Scripting (4.8.2)* con 30 herramientas, y *Testing for HTTP Incoming Requests (4.8.17)* con 19. Asimismo, existen 43 herramientas a las que se aplicaron criterios de evaluación para conocer su efectividad: efectividad y eficiencia. **Conclusiones:** hay 66 herramientas que ayudan a evaluar la seguridad de las aplicaciones Web, estas ejecutan pruebas de forma automatizada, para conocer las vulnerabilidades que tiene la aplicación. Ahora bien, no se puede indicar que las herramientas cuentan con todos los casos de pruebas automatizados necesarios para detectar las vulnerabilidades y evitar los ataques más riesgosos enlistados en el OWASP.

2.2. Introducción

Las pruebas de seguridad para aplicaciones Web son esenciales para prevenir la explotación de vulnerabilidades las cuales generalmente buscan comprometer o dañar un sistema y la información que este administra. Se estima que más del 90 % de las aplicaciones Web son vulnerables, con una media de más de 10 vulnerabilidades por aplicación [1]. Una prueba de seguridad para las aplicaciones Web es un método para evaluar las vulnerabilidades de un sistema informático mediante la validación y verificación metódica de la efectividad de los controles de seguridad de la aplicación [2]. El proceso implica un análisis activo de la aplicación en busca de debilidades, fallas técnicas o vulnerabilidades [2]. Para proteger las aplicaciones Web es necesario identificar y eliminar las vulnerabilidades que estas presentan. Una vulnerabilidad es una falla o debilidad en el diseño, implementación, operación o administración de un sistema que se podría explotar para comprometerlo [2]. Las herramientas de pruebas automatizadas pueden complementar los procesos de pruebas manuales para brindar mayor confiabilidad en la cobertura y reducir el tiempo de ejecución de los casos de

prueba de seguridad. Asimismo, pueden apoyar a los equipos de desarrolladores para ejecutar estos tipos de pruebas. Por otro lado, estas herramientas cuentan con limitantes y retos para incrementar el valor agregado que pueden ofrecer a los equipos de calidad [3, 4].

En los últimos años se ha reportado en la literatura, múltiples estudios sobre el uso de herramientas que permiten generar y ejecutar pruebas automatizadas para evaluar la seguridad de las aplicaciones Web [5]. Estas herramientas permiten conocer las vulnerabilidades que sufren las aplicaciones Web para la protección de los datos que administran; su propósito es habilitar mecanismos para mejorar la confiabilidad y la seguridad de las aplicaciones que prueban.

El objetivo de esta investigación es identificar y conocer las herramientas que han sido utilizadas para probar de forma automatizada la seguridad de aplicaciones Web. Las herramientas se clasificaron según los tipos de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web [6]. Para lograr el objetivo se realiza un mapeo sistemático de la literatura en el que se analizaron 63 estudios primarios, de los cuales se identificaron 66 herramientas utilizadas para realizar pruebas automatizadas de seguridad.

En la metodología para el análisis se aplicó un mapeo sistemático.

El documento consta de la siguiente estructura: en la sección 2.3 se encuentra el marco teórico, donde se definen los conceptos más relevantes para el estudio. La sección 2.4 detalla los trabajos relacionados con respecto a este mapeo. En la sección 2.5 explica la metodología que se siguió para realizar el mapeo de literatura. En la sección 2.6 se encuentra el análisis de resultados donde se responde las preguntas de investigación. En la sección 2.7 expone la discusión de este estudio. En la sección 2.8 se encuentra las lecciones aprendidas y en la sección 2.9 se encuentran las conclusiones.

2.3. Marco teórico

2.3.1. Ataques cibernéticos

Un ataque cibernético es un evento que interrumpe el funcionamiento de las aplicaciones Web de forma ilícita, con el fin de robar información o dinero de los usuarios de estas aplicaciones [7]. Es necesario que las aplicaciones Web cuenten con ciertos mecanismos de protección para que no sufran de ataques [8].

Existe una organización sin fines de lucro llamada OWASP (por sus siglas en inglés *Open Web Application Security Project*) [9] que provee información sobre la seguridad de las aplicaciones Web y que periódicamente publica listas de los ataques de seguridad más comunes, entre otras. Es importante conocer las tendencias en cuanto a los ataques cibernéticos más utilizados, para poder implementar mecanismos de seguridad que los contrarresten y así, proteger las aplicaciones Web.

El *top* 10 de ataques cibernéticos más críticos y frecuentes en la Web para el año 2017, según *OWASP*, se muestra en el Cuadro 2.1 [10].

Cuadro 2.1: Tipos de ataques cibernéticos.

Ranking del ataque	Ataque
1	*Injection*
2	*Broken Authentication*
3	*Sensitive Data Exposure*
4	*XML External Entities (XXE)*
5	*Broken Access Control*
6	*Security Access Control*
7	*Cross-Site Scripting (XSS)*
8	*Insecure Deserialization*
9	*Using Components with known vulnerabilities*
10	*Insufficient logging and monitoring*

A continuación se hará una breve explicación de cada tipo de ataque:

- *Injection*: de acuerdo con la lista de los diez tipos principales tipos de ataques cibernéticos publicado por *OWASP*, el ataque más crítico y frecuente es el *injection*, que consiste en inyectar código como *SQL*, *NoSQL* o *LDAP*. Este tipo de inyección de código puede actuar por medio de una entrada de datos en la aplicación Web, una variable, un parámetro o por medio de servicios internos o externos, que realizan consultas de datos valiosos en la base de datos de la aplicación Web. El objetivo de este ataque es extraer y robar información valiosa de la empresa o bien, robar sesiones de usuario para poder acceder sin problemas a la aplicación Web atacada [10].

- *Broken Authentication*: el ataque *broken authentication* es el segundo ataque más crítico y frecuente de las aplicaciones Web [10], radica en el robo de credenciales (usuario y contraseña) para acceder a una aplicación Web, ya sea por falta de encriptación de las contraseñas de los usuarios registrados en la base de datos de la aplicación o por la falta de un tiempo límite para usuarios inactivos en la aplicación Web.

- *Sensitive Data Exposure*: Este ataque consiste en el robo de datos confidenciales como los datos financieros o bien, modificar los datos financieros para producir fraudes. Este se realiza por medio de una técnica llamada *man-in-the-middle*, de forma que roba la información cuando está en tránsito a la aplicación Web [10].

- *XML External Entities (XXE)*: otro tipo de ataque que se debe de considerar son los ataques por medio de la modificación del *xml* o la presencia de datos sensibles en el *API*, pues la aplicación Web no cuenta con capas de seguridad para proteger la información que se encuentra en estos archivos.

- *Broken Access Control*: consiste en aprovechar las vulnerabilidades del sistema para acceder a funcionalidades restringidas por los usuarios o accesos a funciones no autorizadas, con el fin de robar cuentas, ver archivos confidenciales o modificación de datos [10].

- *Security Access Control*: este se aprovecha de las vulnerabilidades de la configuración incorrecta de la seguridad de la aplicación, lo cual se debe a la mala configuración del *ad hoc* del sistema, configuraciones predeterminadas inseguras, encabezados de HTTP mal configurados y mensajes de error que muestra el sistema, información técnica y rutas que deberían ocultarse a los usuarios [10]. Los atacantes aprovechan estas malas configuraciones para acceder fácilmente al sistema y robar la información o bien, modificar la información con que cuenta la aplicación.

- *Cross-Site Scripting (XSS)*: los ataques XSS (*cross-site scripting*) y CSRF (*cross-site request forgery*) se encuentran entre el ranking de los más riesgosos y frecuentes [10]. El ataque *CSRF* envía una solicitud que contiene código malicioso, de manera que cuando el usuario reciba la solicitud e ingrese con sus credenciales a la página Web maliciosa, el atacante pueda robar la sesión y realizar el fraude al usuario [11]. Entretanto, el ataque *XSS* manda un código malicioso a la aplicación Web, de forma que si el usuario accede a este código, podría redireccionarlo a otra página Web maliciosa [12].

- *Insecure Deserialization*: se funda en la obtención de códigos vulnerables para realizar ciertos servicios en el sistema. Estos códigos son inseguros, defectuoso que provoca que el sistema cuente con vulnerabilidades y se aprovechen de manipular o eliminar usuarios, realizar ataques de inyección o editar la configuración de seguridad con que cuenta la aplicación [10].

- *Using Components with known vulnerabilities*: aprovechan o ejecutan ataques en componentes como las librerías de la aplicación para realizar otros ataques y provocar pérdida de datos o la adquisición del servidor [10].

- *Insufficient logging and monitoring*: consiste en el aprovechamiento de la falta de información y monitoreo ante algún ataque del sistema, de forma que los atacantes sean persistentes hasta llegar a acceder al sistema [10].

2.3.2. Metodologías de pruebas de seguridad por OWASP

OWASP cuenta con una metodología de pruebas de seguridad para determinar vulnerabilidades de seguridad en aplicaciones Web [6]. Esta metodología en su sección 4 detalla 11 categorías donde para cada una lista el conjunto de vulnerabilidades asociadas que se deben evaluar en las aplicaciones Web: se detallan así: *Information Gathering (4.2)* que detalla 10 tipos de vulnerabilidades, *Configuration and Deployment Management Testing (4.3)* que detalla 9 tipos de vulnerabilidades, *Identity Management Testing (4.4)* que detalla 5 tipos de vulnerabilidades, *Authenticaion Testing (4.5)* que detalla 10 tipos de vulnerabilidades, *Authorization Testing (4.6)* que detalla 4 tipos de vulnerabilidades, *Session Management Testing (4.7)* que detalla 8 tipos de vulnerabilidades, *Input Validation Testing (4.8)* que detalla 17 tipos de vulnerabilidades, *Error Handling (4.9)* que detalla 2 tipos de vulnerabilidades, *Weak Cryptography (4.10)* que detalla 4 tipos de vulnerabilidades, *Business Logic Testing (4.11)* que detalla 9 tipos de vulnerabilidades y *Client Side Testing (4.12)* que detalla 12 tipos de vulnerabilidades [6]. Tal como se muestra en el Cuadro 2.2.

Cuadro 2.2: Clasificación de tipo de vulnerabilidades de seguridad Web por OWASP.

Id	Categoría	Cant.
4.1	*Web Application Security Testing*	2
4.2	*Information Gathering*	10
4.3	*Configuration and Deployment Management Testing*	9
4.4	*Identity Management Testing*	9
4.5	*Authenticaion Testing*	10
4.6	*Authorization Testing*	4
4.7	*Session Management Testing*	8
4.8	*Input Validation Testing*	17
4.9	*Error Handling*	2
4.10	*Weak Cryptography*	4
4.11	*Business Logic Testing*	9
4.12	*Client Side Testing*	12

2.3.3. Herramientas que evalúan la seguridad de las aplicaciones Web

Las herramientas que evalúan la seguridad de las aplicaciones Web realizan una lista de pruebas para detectar las vulnerabilidades que tiene la aplicación, estas pruebas se pueden ejecutar durante el desarrollo de la aplicación o bien, cuando se termine, lo cual dependería del tiempo que tengan los desarrolladores durante el ciclo de vida del *software* para hacer las respectivas pruebas de seguridad [13].

Las pruebas pueden ser automatizadas o manuales. Las manuales son aquellas en las que el desarrollador o ingeniero de calidad de software debe de realizar una serie de pasos para evaluar si la aplicación Web exhibe o no vulnerabilidades de seguridad [14]. Mientras tanto, las automatizadas son aquellas pruebas que están programadas y se ejecutan por medio de una herramienta, por lo que no es necesario que lo haga un desarrollador o ingeniero de calidad de software ejecute paso a paso la prueba [15].

Dentro de las ventajas con que cuentan las pruebas manuales se encuentran que son pruebas más exhaustivas, permiten realizar pruebas más exploratorias, económicamente podrían ser más baratas que comprar una herramienta para evaluar la seguridad por medio de pruebas automatizadas. Por otro lado, las desventajas de las pruebas manuales son: poca confiabilidad, existe el riesgo de no encontrar todas las vulnerabilidades, alto consumo del tiempo para ejecutar las pruebas y estas se deben de ejecutar una por una [16].

Las ventajas con que cuentan las pruebas automatizadas son: pruebas más rápidas, se pueden ejecutar en paralelo, brinda más confiabilidad al encontrar vulnerabilidades en el sistema, pueden ser soportadas en varias aplicaciones Web. Por otro lado, las desventajas de las pruebas automatizadas son: dificultad para ejecutar pruebas a nivel de UI, no realizan pruebas exploratorias, puede que la herramienta sea más costosa que realizar pruebas manuales [16].

2.4. Trabajo relacionado

Existen varios investigadores que han realizado estudios sobre la importancia de la seguridad de las aplicaciones Web y el contar con herramientas sobre la seguridad de las aplicaciones Web, con el fin brindar una mayor seguridad de los datos y de los usuarios en las aplicaciones Web. Se encontraron seis estudios secundarios relacionados con el tema de las herramientas que evalúan la seguridad Web, los cuales se describen a continuación:

Pfleeger et al. [17] estudiaron el problema del por qué es difícil medir la seguridad Web. Según los autores la dificultad radica en que existen diferentes métricas y que no se puede probar al cien por ciento el software, por lo que es complejo determinar si se realizó un buen trabajo al evaluar la seguridad del software. Adicionalmente ellos recalcan que existen varias herramientas para evaluar la seguridad del software con diferentes métricas, distintos propósitos y audiencia. Como parte de las conclusiones, los autores responden a cómo abordar las dificultades de manera que la medición de la seguridad Web sea más precisa y útil. Además, sugiere como estrategia el usar diferentes métricas para diversos propósitos, audiencias y objetivos.

Thompson [18] realizó otro estudio relacionado, donde también se responde a la pregunta del porqué las pruebas de seguridad son difíciles. Nuevamente, uno de los argumentos es que no siempre se puede cubrir el cien por ciento del *software*. El autor indica que complica contemplar todos los posibles escenarios de prueba para detectar las vulnerabilidades de seguridad del sistema. Thompson [18] concluye que es necesario contar con herramientas para pruebas de seguridad automatizadas, que sean eficientes en comparación con la ejecución de pruebas manuales.

Nabil et al. [19] identificaron varias herramientas que evalúan la seguridad del *software*, su objetivo fue conocer cuál herramienta era la más indicada para utilizar en el ciclo de vida de desarrollo del software (SDLC). En este estudio, las herramientas fueron evaluadas acorde con el ciclo de vida de desarrollo: en la fase de requerimiento, diseño, desarrollo. Asimismo, detecta cuál es la frecuencia de herramientas ejecutadas en cada ciclo de vida de desarrollo.

Rafique et al. [5], evalúan las vulnerabilidades encontradas en aplicaciones Web, mediante un mapeo de varias herramientas desde el 1994 hasta el 2014. Los auto-

res comentan las vulnerabilidades y las categorizan de acuerdo con la lista de los 10 tipos de ataques más riesgosos y frecuentes reportado por *OWASP*; de igual manera, indican en cuál fase del ciclo de vida de desarrollo se encuentran las mayores vulnerabilidades. Parte de la conclusión del estudio fue el mapeo de las herramientas acorde con las categorías como: herramientas que evalúan las vulnerabilidad de una aplicación Web durante el desarrollo del ciclo de vida del software y la segunda categoría fue acorde con el buen establecimiento de los principios de políticas para las aplicaciones Web presentadas en otro estudio relacionado con la automatización para las vulnerabilidades de *Cross-Site Scripting* (XSS) [5].

Curphey et al. [20] explican la importancia de contar con diferentes herramientas para evaluar la seguridad de las aplicaciones Web, cada una enfocada en distintos campos: pruebas de base de datos, de *black-box*, de *white-box*, de *Web services*, de *runtime* y de *proxy*. Parte de la conclusión de ese estudio, es la forma efectiva de probar la herramienta, la cual consiste en hacer las pruebas en un sitio o parte de un código que se conozca muy bien o que se haya analizado.

El aporte que brinda este mapeo sistemático es poder hacer una clasificación de las herramientas según los tipos de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web [6] y cuantificar las vulnerabilidades más reportadas por los estudios identificados. Se realiza un mapeo de las herramientas desde el 2006 hasta el 2019; esto es importante porque se está mostrando una lista de herramientas más actualizadas comparada con los estudios previamente mencionados, asimismo se realiza un mapeo de estudios de herramientas clasificadas de acuerdo con los tipos de pruebas de vulnerabilidad de seguridad de las aplicaciones Web realizadas por cada herramienta.

2.5. Metodología

La metodología utilizada para esta investigación fue un mapeo sistemático de la literatura, de acuerdo con los lineamientos de Petersen [1] y las recomendaciones estipuladas por Kitchenham y Charters [11].

Primero se definió el objetivo y las preguntas de investigación, luego se realizó el proceso de búsqueda de estudios y selección de los estudios, y la aplicación de

criterios de inclusión y exclusión. Posteriormente, se evaluó la calidad de los estudios y se realizó la extracción de datos para realizar su análisis.

2.5.1. Objetivo

El objetivo para esta investigación fue formulado utilizando el modelo *Goal Question Metric* (GQM) [3]. El objetivo es analizar las herramientas de pruebas automatizadas de seguridad con el propósito de caracterizarlas con respecto a los aspectos de seguridad que prueban y su efectividad, desde el punto de vista de la investigadora, en el contexto de aplicaciones Web.

2.5.2. Preguntas de investigación

Con el fin de orientar la investigación, se definieron las siguientes preguntas de investigación:

RQ1. ¿Cuáles herramientas se han reportado para pruebas automatizadas de seguridad en aplicaciones Web?

Esta respuesta facilita identificar si existen herramientas que evalúan, de forma automatizada, la seguridad de las aplicaciones Web. También ayuda a encontrar las herramientas más reportadas en la literatura, para poder así señalar cuáles son las herramientas de seguridad Web más populares. Se analizan las pruebas que ofrece cada herramienta de seguridad y se tipifican con base en los niveles y subniveles de vulnerabilidades por OWASP [6], a fin de encontrar cuáles son las más comunes para evaluar la seguridad de las aplicaciones Web.

RQ2. ¿Cómo se ha evaluado la efectividad de las herramientas que automatizan las pruebas de seguridad para aplicaciones Web?

Esta posibilita analizar cómo ha sido evaluada la efectividad de las herramientas para pruebas de seguridad Web, lo que incluye analizar la metodología que han usado los estudios para evaluar la efectividad de las herramientas.

2.5.3. Proceso de búsqueda

Para el proceso de búsqueda de estudios se realizó una exploración con el fin de identificar estudios relevantes para la investigación. Los mejores estudios encontrados en esta búsqueda inicial, con base en criterios como: objetivo, preguntas de investigación y aporte de la investigación; se definieron como artículos de control.

Artículos de control. Se encontraron los siguientes artículos de control:

Jan et al. [24] presentan una herramienta llamada SOLMI *(SOLve and Mutation-based test generation for XML Injection*) que permite evaluar el XML de las aplicaciones Web. Esta evaluación consiste en una serie de pruebas para identificar si el XML es vulnerable ante ataques maliciosos. El estudio también realiza una valoración de la herramienta SOLMI con respecto a su eficiencia para encontrar vulnerabilidades.

Dashevskyi et al. [25] presentan la herramienta llamada TESTREX, la cual consiste en una plataforma para evaluar la seguridad de las aplicaciones Web. Dentro de las pruebas que realiza se encuentran: *SQL injection*, *XSS (cross-site scripting)* y *black box*. Los autores explican el desarrollo y la funcionalidad de la herramienta, así como la evaluación que aplicaron, donde evidencian la eficiencia de la herramienta al encontrar vulnerabilidades de la seguridad en varias aplicaciones Web.

Appelt et al. [26] presentan la herramienta ML-Driven, la cual se basa en *machine learning* que evoluciona conforme vayan detectando vulnerabilidades SQL en las aplicaciones Web. El estudio explica el propósito de la herramienta y las vulnerabilidades que detecta en las aplicaciones Web.

Seguidamente, se procedió a definir la cadena de búsqueda con base en los términos utilizados por los artículos de control, el objetivo del estudio y las preguntas de investigación planteadas.

Cadena de búsqueda para la creación de la cadena de búsqueda, se utilizó el modelo PICO (Población, Intervención, Comparación, Salida) [4], que da como resultado la siguiente cadena de búsqueda:

```
("automat*" OR "tool") AND ("secur*␣test*") AND ("Web*")
```

La cadena está conformada por palabras claves que se encontraron en los estudios, tales como: “aplicaciones Web”, “pruebas de seguridad automatizadas”, “herramientas”, “aspectos de seguridad” y “efectividad”.

Las palabras “automat”, “secur” y “Web” son palabras que se les agregó un asterisco debido a que son palabras derivadas, las cuales pueden contar con más de un término, por ejemplo: “automat*” puede comprender palabras como “automated” y “automation”; la palabra “secur” puede comprender palabras como “secure”, “secured” y “security”. Por ende, para contar con la mayor cobertura de resultados de estudios relacionados con este, se procedió a asignarles asteriscos.

Bases de Datos una vez definida la cadena de búsqueda, se indagó sobre estudios relevantes en tres bases de datos: *Scopus* [1], *IEEE Xplore* [2] y *Web of Science* [3]. Estas tres bases de datos fueron elegidas por ofrecer buena cobertura de análisis en el área de computación.

Una vez corrida la cadena inicial sobre estas tres bases de datos, tuvo que ser refinada en varias ocasiones para evitar el ruido de otros estudios que no estaban relacionados con esta investigación.

Período de búsqueda la búsqueda se realizó en el primer semestre del 2019. El resultado obtenido fue: al ejecutar con la cadena de búsqueda en la base de datos *Scopus* fueron 159 estudios, en la base de datos *IEEE Xplore* fueron 81 estudios y en la base de datos *Web of Science* fueron 20 estudios. El total que se obtuvo al utilizar la cadena de búsqueda fue de 260, de los cuales 75 eran duplicados.

Una vez concluido el proceso de búsqueda, se procedió con la selección de estudios, que consistió en la aplicación de criterios de inclusión y exclusión para dejar solamente los relevantes para la investigación. Este procedimiento se muestra con más detalle en la Figura 2.1.

[1] https://www2.scopus.com/home.uri
[2] http://ieeexplore.ieee.org/
[3] apps.webofknowledge.com

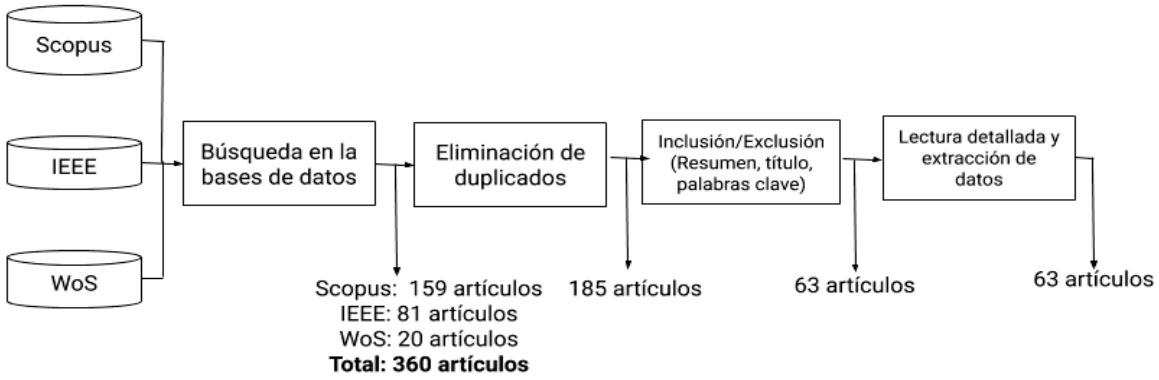


Figura 2.1: Proceso de selección de estudios.

2.5.4. Proceso de selección de estudios

Para la selección se procedió a aplicar los de criterios de inclusión y exclusión de los 185 estudios obtenidos, no duplicados y recuperados de las bases de datos *Scopus*, *IEEE Xplore* y *Web of Science*. Estos criterios fueron ejecutados sobre el resumen, título y palabras claves de los estudios.

Los siguientes fueron los criterios de inclusión utilizados:

I1. Estudios que trataran sobre herramientas de seguridad para aplicaciones Web.

I2. Estudios escritos en inglés.

Los criterios de exclusión empleados fueron los siguientes:

E1. Estudios no disponibles en texto completo.

E2. Estudios secundarios y terciarios.

Después de aplicar los criterios de inclusión y exclusión, se seleccionó un total de 63 estudios primarios. El proceso completo de selección se muestra en la Figura 2.1. En la sección del Apéndice 2.A se encuentra la lista completa de los estudios seleccionados para este mapeo de la literatura.

2.5.5. Evaluación de la calidad

La evaluación de calidad ofrece información sobre el nivel de detalle de los estudios seleccionados. El puntaje de calidad de cada uno da una idea de qué tanto aporta al responder las preguntas de investigación.

La evaluación de los artículos se realizó con base en los siguientes criterios de calidad:

1. Q1.¿El estudio describe la funcionalidad de la herramienta y cómo se ejecuta las pruebas automatizadas?

   a) No en lo absoluto: en el estudio solo menciona la herramienta aplicada. No brinda información de cómo se ejecutan las pruebas automatizadas.

   b) Parcialmente: en el estudio describe a nivel general la funcionalidad de la herramienta sin ofrecer detalles y solo menciona las pruebas automatizadas.

   c) Totalmente: en el estudio detalla la funcionalidad de la herramienta y describe cómo se ejecutan las pruebas automatizadas.

2. Q2.¿El estudio detalla el procedimiento de evaluación de la efectividad de la herramienta y sus resultados?

   a) No en lo absoluto: el estudio no evaluó la herramienta.

   b) Parcialmente: el estudio menciona muy superficialmente que se hizo una evaluación de la herramienta, y da el resultado general de misma, sin ofrecer detalles.

   c) Totalmente: en el estudio detalla el procedimiento y los resultados de la evaluación de la efectividad de la herramienta.

Cada criterio se evaluó en una escala de 0 a 2, donde 0 significa “No en lo absoluto”, 1 significa “Parcialmente” y 2 significa “Totalmente”.

El Cuadro con los resultados de la evaluación de calidad se muestra en la sección del apéndice 2.B. Los puntajes de calidad de los estudios están entre 0 y 4, donde 4 es

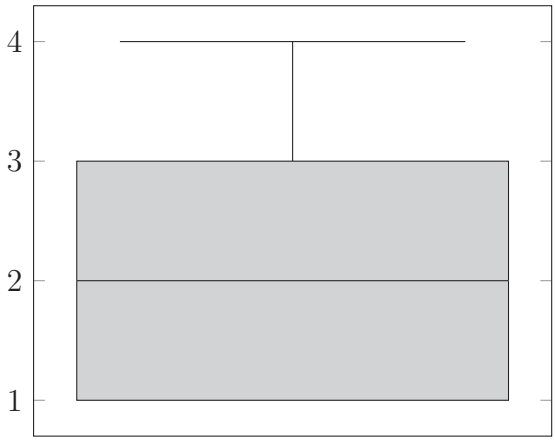


Figura 2.2: Resultado total de la evaluación de calidad

el puntaje máximo y 0 el mínimo. Los estudios que cuentan con la máxima calificación son los que proveen mayor información a la investigación.

En la Figura 2.2 de percentiles de evaluación de calidad, se identifica la mediana con un valor de 3, donde algunas evaluaciones de calidad de ciertos estudios son de 1 y otras de 4. El promedio de la evaluación de calidad es de 2,54, lo que refleja que los estudios seleccionados aportan información importante para el mapeo de esta investigación.

2.5.6. Extracción de datos

Para la extracción de información de los 63 estudios se elaboró una tabla con una serie de elementos que permitían responder las preguntas de investigación. En el Cuadro 2.3 se muestran los componentes de información considerados en el formulario de extracción.

Cuadro 2.3: Componentes del formulario de extracción.

Categoría	Componentes
Información general	Id, Año, Título, Autores, Tipo de documento (estudio / conferencia), Base de datos, Tipo de estudio, Q1, Q2
Herramientas de pruebas de seguridad (RQ1)	Nombre de la herramienta, ¿La herramienta es creada, modificada o usada?, Descripción de la herramienta, Entorno de aplicación de la herramienta, Lenguaje de programación, Link de la herramienta, ¿Cuáles son las pruebas de seguridad?, Escenarios de seguridad que evalúa, ¿Cómo aplica el escenario?
Evaluación de la efectividad de las herramientas (RQ2)	¿Evalúan la herramienta?, ¿Cuál es el procedimiento de evaluación?

Para la pregunta de investigación RQ1, se obtuvo la información acerca de la herramienta, el entorno donde se ejecuta la herramienta, la descripción de la herramienta, la información de las pruebas que se encuentran automatizadas y que ejecuta la herramienta para evaluar la seguridad de las aplicaciones Web.

Para la pregunta de investigación RQ2, se identificó si la herramienta fue evaluada o no y el procedimiento de evaluación que realizó el estudio para conocer su efectividad. Cabe mencionar que la extracción fue realizada por la investigadora del estudio. En el enlace https://tinyurl.com/y8fp26lj se detalle el formulario de extracción.

2.5.6.1. Análisis de datos

Para realizar el análisis de los datos, se tomó como base la información contenida en el formulario de extracción.

En la primera pregunta de investigación se identificaron las herramientas mencionadas por cada estudio. Con base en esta información se hizo un conteo de la cantidad de estudios que reportaban cada herramienta, para determinar las más usadas. Además, se tomó en cuenta el año de publicación de los estudios que mencionaba cada herramienta, asimismo, la descripción de la construcción y funcionalidad de la herramienta. También se consideró si la herramienta había sido creada o adaptada para evaluar la seguridad de las aplicaciones Web. Se identificaron los tipos de pruebas asociados con cada herramienta y se realizó la clasificación de estas. Por último, se analizaron los tipos de pruebas ejecutadas por año, de forma que se pudieran reflejar los tipos de pruebas más utilizados en estos años recientes.

Para la segunda pregunta de investigación, se analizaron las métricas de efectividad aplicadas a cada herramienta. Para eso, se identificó cuáles habían sido sujetas a una evaluación de efectividad, así como los tipos de criterios que se aplicaron para cada herramienta evaluada.

El mapeo aplicado en esta investigación permitió obtener la información necesaria para poder empezar y realizar un análisis de los resultados.

2.5.7. Amenazas a la validez

En esta sección se permite reconocer las limitantes que puede contener el estudio con respecto a la validez de los resultados obtenidos del mapeo.

Consulta a expertos. la consulta de un experto previa al desarrollo de la investigación permitió contar con un panorama del tema y una guía para buscar estudios de control. No obstante, se permitió mitigar el riesgo de no preguntar a un experto durante el desarrollo del tema ya que se realizaron constantes investigaciones y búsquedas exhaustivas realizadas en el tema.

Selección de la cadena de búsqueda y las bibliotecas digitales. la cadena de búsqueda fue definida a partir de una exploración en bases de datos y un conjunto de estudios de control. Además, fue refinada mediante un conjunto de pruebas piloto.

Las bases de datos seleccionadas son reconocidas por tener una buena cobertura de información en el campo de Ingeniería de *Software*. Durante el proceso de inclusión o exclusión, cuando existieron dudas sobre un estudio específico, se procedió a su lectura completa.

Extracción y clasificación de artículos primarios. para evitar el sesgo de información se revisó constantemente el formulario de extracción, hasta contar con los elementos necesarios para poder extraer con completitud la información de los estudios y así, poder llevar a cabo la investigación.

Generalización de los resultados. para mitigar este riesgo, a todo el mapeo que se realizó en esta investigación se le aplicó los protocolos previamente definidos y validados. Para la clasificación de las herramientas, se utilizó la metodología de pruebas para aplicaciones Web de OWASP. En la mayoría de los casos la clasificación fue extraída de manera explícita de los estudios; sin embargo, para algunos casos la selección de la vulnerabilidad se realizó de manera implícita a partir de los datos reportados.

Todo el proceso se reportó de forma detallada para facilitar su análisis y utilización en estudios posteriores.

2.6. Análisis de resultados

Se encontraron 63 estudios fundamentales para la realización del mapeo de la literatura. Estos estudios son conformados por 12 *journal* y 51 conferencias. Los *journal* se encuentran entre los años del 2012 al 2018, donde, en el año 2012 hay 2 *journal*, en el año 2014 hay 3, en los años 2015, 2016 y 2017 hay 1 *journal* por cada año, mientras que en el año 2018 hay 4. Se puede indicar que hubo una tendencia de mayores *journal* en el año 2018. Mientras tanto, las conferencias fueron comprendidas entre los años del 2006 al 2019. En los años 2006, 2007 y 2016 hay 1 conferencia, en los años 2008 y 2018 hay 2 conferencia, en los años 2009, 2011 y 2019 hay 3 conferencias, en los años 2010 y 2017 hay 4 conferencias, en el año 2015 hay 5 conferencias, en los años 2013 y 2014 hay 7 conferencias y en el año 2012 hay 8 conferencias. Se puede indicar que hubo una tendencia de más estudios en el año 2012. Dado lo anterior, dentro del mapeo se puede ver una tendencia mayor a ser

Figura 2.3: Cantidad de herramientas por año.

estudios de conferencia que de un *journal*. En la sección del Anexo 2.A se encuentra la lista de los estudios analizados.

2.6.1. Herramientas que se han reportado para pruebas automatizadas de seguridad en aplicaciones Web (RQ1)

En los estudios se encontraron 66 herramientas que automatizan las pruebas para evaluar la seguridad de las aplicaciones Web. Las herramientas *ZAP* [28, 29, 30], *ISTA* [31, 32], *SPaciTE* [33, 34] y *Volcano* [35, 36] fueron las más reportadas. Estas herramientas fueron referenciadas por dos o más estudios.

En la Figura 2.3 muestra la cantidad de herramientas reportadas por año. Las herramientas fueron publicadas entre los años 2006 y 2019, con la mayoría de ellas concentradas en los años 2012, 2014 y 2018.

En el año 2012 se reportaron 9 herramientas: ISTA: *Integration and System test Automation* [31], Noncespace [37], Tool-prototype [38], SPaCiTE [33, 34], Selenium IDE [39], Urls black box tests on the Web pages [40], WS-Attacker [41], Eclipse IDE [42], WSSecTool [43]. En el año 2014 se reportaron 12 herramientas: SPaCIoS [44], *Signature evaluation* [45], *Scalable Quality and Testing Lab* (SQTL) [46], Burp tool y ZAP [30], BIOFUZZ [47], RBVT [48], IMAATT [49], SECEVAL [50], Ka-

maleonFuzz [51], WSVTS [52], WSInject [53].

En el año 2018 se reportaron 10 herramientas: SAFELI y Wap [54], ZAP y DAST [29], Fortify, JBroFuzz, Paros and WebScarab [55], PBST [56], IDS [57]. En los años iniciales (2006 y 2007), solo se reportó una herramienta por año.

Cabe resaltar que en los años 2012, 2014 y 2018 se encuentran los estudios de las herramientas más reportadas (ZAP, ISTA, SPaciTE y Volcano).

En el Cuadro 2.4 se muestra la descripción de las herramientas más reportadas y las que también son recomendadas por OWASP (para escaneo de seguridad y análisis de código) [58, 59, 60] y que se encuentran dentro de las tendencias con mayor herramientas reportadas en los años 2012, 2014 y 2018.

Cuadro 2.4: Descripción de las herramientas con mayor tendencia y más reportadas.

Herramienta	Descripción	Estudios
Burp tool	Burp suite es una plataforma integrada para realizar pruebas de seguridad en aplicaciones Web. Fue diseñada para cumplir con muchas tareas de pruebas de penetración y ayudar a los profesionales de seguridad en cada paso de una herramienta de prueba.	[30]
Dast	Es un método de prueba de recuadro negro aplicado en la ejecución de aplicaciones desde el exterior. Esta herramienta funciona mediante la ejecución de *scripts* de ataque predefinidos que envían una solicitud a la aplicación Web. La respuesta de la aplicación Web a la herramienta se analiza para determinar la existencia de una vulnerabilidad. Esta herramienta tiene sus propios *scripts* y parámetros para configurar la prueba de seguridad.	[29]

Continúa en la página siguiente.

Herramienta	Descripción	Estudios
Fortify	Fortify es una herramienta de análisis estático utilizado para encontrar las causas raíz de vulnerabilidades de seguridad en el código fuente.	[55]
ISTA	ISTA es una herramienta que consiste en la ejecución de los casos de prueba de seguridad implementados desde la Descripción de Implementación del Modelo de Amenaza (TMID). El lenguaje de programación utilizado en esta herramienta es java.	[31, 32]
SPaciTE	Esta herramienta se basa en un verificador de modelos dedicado para análisis de seguridad que genera posibles ataques con respecto a vulnerabilidades comunes en aplicaciones Web.	[33, 34]
Volcano	Volcano realiza pruebas de seguridad de caja blanca para encontrar vulnerabilidades de inyección SQL en aplicaciones Web escritas en lenguaje PHP.	[35, 36]
ZAP	La herramienta realiza escaneos dinámicos de seguridad. Utiliza el complemento *FindSecBug* para realizar una verificación de seguridad de código estático y una herramienta de verificación de dependencia OWASP para verificar amenazas de seguridad en bibliotecas de terceros.	[28, 29, 30]

Con respecto a los detalles de las 59 herramientas, en el Cuadro 2.12, ubicado en el Anexo 2.C, se presentan las herramientas, en qué consisten (una breve descripción) y el lenguaje de programación en el que se crearon.

Dentro de las 66 herramientas, hay 43 que fueron usadas para los estudios y 23 que fueron creadas. Cabe mencionar que esta información es meramente implícita, los estudios no brindan de forma explícita si la herramienta mencionada en el estu-

Figura 2.4: Frecuencia de vulnerabilidades del primer nivel de OWASP por año.

dio fue usada o creada. En la sección del formulario de extracción, se encuentra la información de las herramientas que fueron usadas y creadas.

Las herramientas se clasificaron según las 11 categorías de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web de OWASP (basado en el primer nivel de la clasificación) [6]. En la Figura 2.4 se presenta la tendencia de vulnerabilidades evaluadas por categoría de OWASP del primer nivel por cada año desde el 2006 hasta el 2019.

Los resultados confirman que la categoría de vulnerabilidad *Input Validation Testing (4.8)* es la más evaluada por las herramientas entre los años del 2007 al 2019, donde los años 2014 y 2018 fueron los más evaluados con 21 y 17 ocurrencias respectivamente.

El Cuadro 2.5 agrupa las herramientas para cada una de las categorías de vulnerabilidades del primer nivel de la clasificación de OWASP. Las categorías para las cuáles se identificaron herramientas fueron: *Configuration and Deployment Management Testing (4.3)*, *Identity Management Testing (4.4)*, *Authenticaion Testing (4.5)*, *Authorization Testing (4.6)*, *Session Management Testing (4.7)*, *Input Validation Testing (4.8)*, *Error Handling (4.9)*, *Weak Cryptography (4.10)* y *Client Side Testing (4.12)*.

En la sección del Apéndice 2.D, se encuentra la Figura 2.9 donde presenta la cantidad de herramientas por categoría y subcategoría de OWASP.

Cuadro 2.5: Herramientas por categoría de OWASP (primer nivel).

Id	Herramientas y referencias	Cant.
4.3	ISTA [32], AppScan [61], Urls black box tests on the Web pages [40], Fuzz testing tool [62], HJ2IF [63], IPT-WS [64], MobSTer [65], SAML-based SSO IdP by Google [66], SECEVAL [50], WS-Attacker [41], WSFAgressor [67], WSInject [53], WSSecTool [43], WSAttaker [68], WSVTS [52]	15
4.4	SPaCiTE [33, 34], AppScan [61]	2
4.5	ZAP [28, 29], DAST [29], Fortify [55], IMAATT [49], JBroFuzz [55], OAuthTester [69], Paros [55], PBST [56], Selenium IDE [39], WebScarab [55]	10
4.6	ISTA [31], OAuthTester [69], SPaCIoS [44]	3
4.7	ZAP [28], CodePulse [70], Magento [71], MobsTer [65], PBST [56], SAMATE [72], Selenium IDE [39], AOP [73], Urls black box tests [40], ISTA [31], Deemon [74], Fortify [55], JBroFuzz [55], Paros [55], WebScarab [55]	15

Continúa en la página siguiente.

Id	Herramientas y referencias	Cant.
4.8	ZAP [28, 29, 30], ISTA [31], SPaCiTE [33, 34], Volcano [35, 36], AOP [73], Sign-WS [64], ATUSA [75], BIOFUZZ [47], BurpTool [30], CRAXweb [76], Circe [77], CodePulse [70], DAST [29], Deemon [74], XSS, black box and SQLI injection point [78], Eclipse IDE [42], Urls black box tests on the Web pages [40], Fortify [55], IAAT [79], IMATT [49], JBroFuzz [55], JWebUnit [80], JWAST [81], KamaleonFuzz [51], Magento [71], MobSTer [65], MBT [82], Noncespace [37], PHP2XMI [83], Paros [55], PURITY [84], PBST [56], RBVT [48], RADWS [64], SAFELI [54], SAMATE [72], SAP HANA XS Applications [85], Eclipse IDE test cases execution [86], SSES [87], SPaCIoS [44], SQLIVDT [88], SQLMAP [88], Selenium IDE [39], IDS [57], Signature evaluation [45], Tamper data [89], Tool-prototype [38], WAP [54], WSFAgressor [67], WSInject [53], WSVTS [52], WebScarab [55], XSSINJECTOR [90] XSS and SQLI [91]	55
4.9	ZAP [28]	1
4.10	ISTA [31], Fortify [55], JBroFuzz [55], Paros [55], SAML-based SSO IdP service provided by Google [66], WebScarab [55]	6
4.12	ZAP [29], ISTA [31], AOP [73], ATUSA [75], Code Pulse [70], DAST [29], Fortify [55], JBroFuzz [55], JWAST [81], KITE [92], Paros [55], SAP HANA XS Applications [85], SQTL [46], WebScarab [55], XSSINJECTOR [90]	15

Para cada categoría (Id) se presenta la lista de herramientas, referencias de los estudios que la evaluaron, y la cantidad de herramientas (Cant.). Los resultados indican que la categoría de pruebas para detectar vulnerabilidades más comunes fue *Input Validation Testing (4.8)* con 55 herramientas, seguido de las pruebas de *Confi-*

guration and Deployment Management Testing (4.3), *Session Management Testing (4.7)* y *Client Side Testing (4.12)* con 15 herramientas utilizadas cada una.

Para el Cuadro 2.5 se cuantificó la cantidad de estudios que trabajaron cada categoría de vulnerabilidad y el total de ocurrencias de las vulnerabilidades de una categoría. Los resultados muestran una tendencia similar a los de las herramientas donde la categoría de pruebas para detectar vulnerabilidades con más estudios fue la *(4.8)* con 52 estudios y 113 vulnerabilidades evaluadas. En el caso de las pruebas de *(4.3)* la cantidad de estudios y vulnerabilidades evaluadas fue 15. Finalmente, las categorías *(4.7) y (4.12)* fueron reportadas en 12 estudios y en el caso de la categoría *(4.7)* evaluó 12 vulnerabilidades y en la categoría *(4.12)* evaluó 17 vulnerabilidades. En la sección del apéndice 2.E se encuentra la Figura 2.10 donde presenta la cantidad de herramientas por categoría de primer nivel de OWASP.

En la Figura 2.5 se presenta la tendencia de vulnerabilidades evaluadas del segundo nivel de OWASP por cada año desde del 2007 hasta el 2019. Los resultados confirman que las vulnerabilidades *Testing for SQL Injection (4.8.5)* y *Cross-Site Scripting (4.8.2)* son las más evaluadas por las herramientas.

Figura 2.5: Frecuencia de vulnerabilidades del segundo nivel de OWASP por año.

El Cuadro 2.6 presenta las vulnerabilidades del segundo nivel de OWASP para las

categorías en que se identificaron herramientas. Las vulnerabilidades para las cuáles se identificaron herramientas fueron: *Testing for Privilege Escalation (4.6.3)*, *Testing for Insecure Direct Object References (4.6.4)*, *Testing for Cookies attributes (4.7.2)*, *Testing for Cross Site Request Forgery (CSRF) (4.7.5)*, *Testing for Reflected Cross Site Scripting (4.8.1)*, *Testing for Stored Cross Site Scripting (4.8.2)*, *Testing for SQL Injection (4.8.5)*, *Testing for LDAP Injection (4.8.6) Testing for XML Injection (4.8.8) Testing for XML injection (4.8.9) Testing for XPath Injection (4.8.10) Testing for Command Injection (4.8.13) Testing for HTTP Incoming Requests (4.8.17) Testing for DOM based Cross Site Scripting (4.12.1) Testing for JavaScript Execution (4.12.2) Testing for HTML Injection (4.12.3)*.

Cuadro 2.6: Herramientas por vulnerabilidad de OWASP (segundo nivel).

Id	Herramientas y referencias	Cant.
4.6.3	ISTA [32]	1
4.6.4	ISTA [32]	1
4.7.2	AOP [73], Evaluating the urls and performing black box tests on the Web pages [40]	2
4.7.5	ISTA [31], Deemon [74], Fortify [55], JBroFuzz [55], Paros [55], WebScarab [55]	6
4.8.1	ZAP [28], BurpTool [30], MBT [82], SPaCIoS [44]	4
4.8.2	ZAP [28, 29, 30], ISTA [31], SPaCiTE [33, 34], AOP [73], BurpTool [30], CRAXweb [76], Circe [77], CodePulse [70], DAST [29], XSS, black box and SQLI based on injection point [78], IAAT [79], IMAATT [49], JwebUnit [55], JWAST [81], KamaleonFuzz [51], MobSTer [65], MBT [65], Noncespaces [37], PHP2XMI [83], PURITY [84], RBVT [48], SSES [87], SPaCIoS [44], SQLIVDT [88], Selenium IDE [39], Tamper data [89], Tool-prototye [38], WSInject [53], XSSINJECTOR [90], XSS and SQLI evaluation [91]	30

Continúa en la página siguiente.

Id	Herramientas y referencias	Cant.
4.8.5	ZAP [28, 29], ISTA [31], SPaCiTE [33, 34], Volcano [35, 36], AOP [73], BIOFUZZ [47], CRAXweb [76], Circe [77], CodePulse [70], DAST [29], XSS, black box and SQLI based on injection point [78], Eclipse IDE [42], Fortify [55], IAAT [79], IMAATT [49], JBroFuzz [55], JWAST [81], Magento [71], MobSTer [65], PHP2XMI [83], PURITY [84], Paros [55], RBVT [48], RAD-WS [64], Eclipse IDE test cases execution [86], SAFELI [54], SAMATE [72], SAP HANA XS Applications [85], SSES [87], SPaCIoS [44], SQLIVDT [88], SQLMap [93], Selenium IDE [39], IDS [57], Signature evaluation [45], Tamper data [89], Tool-prototype [38], WAP [54], WebScarab [55], XSS and SQLI security evaluation [91]	40
4.8.6	JWAST [81], SSES [87]	2
4.8.8	Eclipse IDE [42], JWAST [81], SPaCioS [44]	3
4.8.9	Sign-WS [64]	1
4.8.10	Eclipse IDE [42], JWAST [81], RAD-WS [64], SSES [87]	4
4.8.13	JWAST [81], SSES [87]	2
4.8.17	ZAP [28, 29, 30], ATUSA [75], BurpTool [30], CRAXweb [76], DAST [29], Deemon [74], XSS, black box and SQLI based on injection point [78], Urls and performing black box tests on the Web pages [40], Fortify [55], JBroFuzz [55], JWAST [81], Paros [55], SAML-based SSO IdP service by Google [66], SECEVAL [50], SPaCIoS [44], Tamper data [89], WSFAgressor [67], WSVTS [52], WebScarab [55]	19
4.12.1	ATUSA [75]	1
4.12.2	ZAP [29], DAST [29], JWAST [81], SAP HANA XS Applications [85]	4
4.12.3	ZAP [29], ISTA [31], DAST [29]	3

En el Cuadro 2.6, en cada vulnerabilidad (Id) se presenta la lista de herramientas y referencias de los estudios que la evaluaron, y la cantidad de herramientas (Cant.). Los tipos de pruebas más reportadas fueron los de la categoría *Input Validation Testing (4.8)* en la que se encontraron las vulnerabilidades más reportadas. Las vulnerabilidades más comúnmente evaluadas fueron la *SQL Injection (4.8.5)* con 40 herramientas, *Cross-Site Scripting (4.8.2)* con 30 herramientas, y *Testing for HTTP Incoming Requests (4.8.17)* con 19 herramientas utilizadas. En la sección 2.F del Apéndice se encuentra la Figura 2.11 donde presenta la cantidad de herramientas por vulnerabilidad (segundo nivel) de OWASP.

En el Cuadro 2.7, se presenta los tipos de pruebas (acorde a la primera y segunda nivel de clasificación por OWASP) realizados por cada herramienta.

Cuadro 2.7: Herramientas para la automatización de pruebas de seguridad Web.

Herramienta	Tipo Prueba	Estudio
Zed Attack Proxy tool (ZAP)	Authentication testing, Session management testing, XSS (Reflected and Stored), SQLI, HTTP incoming request, Testing error handling, Testing javascript execution, Testing HTML injection	[28, 29, 30]
ISTA: *Integration and System test Automation*	Configuration and deployment management testing, Testing for provilege Escalation, Testing for Insecure Direct Object Preferences, CSRF, XSS, SQLI, Testing for weak cryptography, Testing for HTML Injection	[31, 32]
SPaCiTE	Identify Management Testing, XSS, SQLI	[33, 34]
Volcano	SQLI	[35, 36]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
AOP *(aspect oriented programming)*	Testing for cookies atributes, XSS (stored), SQLI, Testing client side testing	[73]
AppScan	Configuration and deployment management testing	[61]
Attack signatures and interface monitoring (Sign-WS)	Testing for SS Injection	[64]
Automatic testing of AJAX user interface (ATUSA)	HTTP Incoming Requests, Testing for DOM based Cross-Site Scripting	[75]
BIOFUZZ	SQLI	[47]
Burp tool	CSS (Reflected, Stored), testing HTTP Incoming Requests	[30]
CRAXweb	XSS (Stored), SQLI, Testing for HTTP Incoming Requests	[76]
Circe	XSS (Stored), SQLI,	[77]
CodePulse	Identify management testing, session management testing, XSS (Stored), SQLI, Client side testing	[70]
DAST *(dinamyc Application security testing) tool: Vega, ZAP, Acunetix*	Authentication testing, XSS (Stored), SQLI, testing for HTTP Incoming Requests, testing for javascript execution, testing for HTML Injection	[29]
Deemon	CSRF, testing for HTTP Incoming Requests	[74]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
Detection of vulnerabilities of Web applications (XSS, black box and SQLI) based on injection point	XSS (Stored), SQLI, testing for HTTP Incoming Requests	[78]
Eclipse IDE	SQLI, XMLI, XPath Injection	[42]
Eclipse IDE test cases execution	SQLI	[86]
Evaluating the urls and performing black box tests on the Web pages	Configuration and deployment management testing, Testing for cookies attributes, testing for HTTP Incoming requests	[40]
Fortify	Authentication testing, CSRF, SQLI, testing for HTTP Incoming requests, testing for weak cryptography, client side testing	[55]
Fuzz testing tool	Configuration and deployment management testing	[62]
HJ2If	Configuration and deployment management testing	[63]
IAAT: *Injection Aware Application Testing*	XSS (Stored), SQLI	[79]
Improved penetration testing (IPT-WS)	Configuration and deployment management testing	[64]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
Integrated MultiAgent Testing Tool (IMAATT)	Authentication testing, XSS (Stored),	[48]
JBroFuzz	Authentication testing, CSRF, SQLI, Testing for HTTP Incoming Requests, Testing for weak cryptography, client side testing	[55]
JWebUnit	XSS (Stored)	[80]
Java Web Application Security Tester (JWAST)	XSS (Stored), SQLI, LDAP injection, XMLI, XPath Injection, Testing for HTTP Incoming Requests, Testing for Javascript Execution	[81]
KITE	client side testing	[92]
KamaleonFuzz	XSS (Stored)	[51]
Magento	Session management testing, SQLI	[71]
MobSTer: *Model-based Security Testing Framework*	Session management testing, Configuration and deployment management testing, XSS (Stored), SQLI,	[65]
Model-Based Testing (MBT)	XSS(Stored and reflected)	[82]
Noncespaces	XSS (Stored)	[37]
OAuthTester	Authentication testing, Authorization testing,	[69]
PHP2XMI	XSS(Stored), SQLI	[83]
PURITY	XSS (Stored), SQLI	[84]
Paros	Authentication testing, CSRF, SQLI, Testing for HTTP Incoming Requests, Testing for Weak Cryptography, Client Side Testing	[55]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
Pattern Based Security Testing tool (PBST)	Authentication testing, Session management testing	[56]
Risk-based vulnerabilities testing (RBVT)	XSS(Stored), SQLI,	[48]
Runtime anomaly detection (RAD-WS)	SQLI, XPath Injection	[64]
SAFELI	SQLI	[54]
SAMATE	Session management testing, SQLI	[72]
SAML-based SSO IdP service provided by Google	Configuration and deployment management testing, Testing for HTTP Incoming Requests, Testing for Weak Criptography	[66]
SAP HANA XS Applications	SQLI, Testing for Javascript Execution	[85]
SECEVAL	Configuration and deployment management testing, testing for HTTP Incoming Requests	[50]
SSES: *Software security evaluation system*	XSS (Stored), SQLI, LDAP Injection, XPath Injection, Testing for Command Injection	[87]
SPaCIoS	Authorization testing, XSS (Reflected and Stored), SQLI, XMLI, Testing for HTTP Incoming Requests	[44]
SQLIVDT *(SQL Injection Vulnerability Detection Tool)*	XSS(Stored), SQLI	[88]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
SQLMap	SQLI	[93]
Scalable Quality and Testing Lab (SQTL)	client side testing	[46]
Selenium IDE	Authentication testing, session management testing, XSS (stored), SQLI	[39]
Signature based intrusion detection systems (IDS)	SQLI	[57]
Signature evaluation	SQLI	[45]
Tamper data	SQLI (Stored), SQLI, Testing for HTTP Incoming Requests	[89]
Tool-prototype	XSS(Stored), SQLI,	[38]
WAP	SQLI	[54]
WS-Attacker	Configuration and deployment management testing	[41]
WSAttaker	Configuration and deployment management testing	[68]
WSFAgressor	Configuration and deployment management testing, Testing for HTTP incoming requests	[67]
WSInject	Configuration and deployment management testing, XSS(Stored),	[53]
WSSecTool	Configuration and deployment management testing	[43]
Web service vulnerability testing system (WSVTS)	Configuration and deployment management testing, Testing for HTTP Incoming Requests	[52]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
WebScarab	Authentication testing, CSRF, SQLI, Testing for HTTP Incoming Requests, Testing for weak cryptography, client side testing	[55]
XSSINJECTOR	XSS (stored), Client side testing	[90]
XSS and SQLI security evaluation	XSS(Stored), SQLI	[91]

De acuerdo con este Cuadro, las vulnerabilidades *Testing for SQL Injection (4.8.5)* y *Cross-Site Scripting (4.8.2)* son las más evaluadas por las herramientas. Ahora bien, si comparamos con el *top* 10 de los riesgos más críticos en aplicaciones Web, publicado por el OWASP [10], vemos que las pruebas de ataques SQLI, *Configuration and deployment management testing*, XSS(*Stored y Refrected*), *Authentication testing*, XMLI son las que se encuentran en este *top*.

El Cuadro 2.8 muestra las herramientas que realizan al menos una prueba para los ataques con más riesgo y críticos en aplicaciones Web.

Cuadro 2.8: Herramientas para la automatización de pruebas de seguridad Web.

Herramienta	Tipo Prueba	Estudio
ZAP	Authentication testing, XSS (Reflected and Stored), SQLI	[28, 29, 30]
ISTA	Configuration and deployment management testing, XSS, SQLI	[31, 32]
SPaCiTE	XSS, SQLI	[33, 34]
Volcano	SQLI	[35, 36]
AOP *(Aspect Oriented Programming)*	XSS (stored), SQLI	[73]
AppScan	Configuration and deployment management testing	[61]
BIOFUZZ	SQLI	[47]
Burp tool	XSS (Reflected, Stored)	[30]
CRAXweb	XSS (Stored), SQLI	[76]
Circe	XSS (Stored), SQLI,	[77]
CodePulse	XSS (Stored), SQLI	[70]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
DAST *(Dinamyc Application Security Testing) tool: Vega, ZAP, Acunetix*	Authentication testing, XSS (Stored), SQLI	[29]
Detection of vulnerabilities of Web applications (XSS, black box and SQLI) based on injection point	XSS (Stored), SQLI	[78]
Eclipse IDE	SQLI, XMLI	[42]
Eclipse IDE test cases execution	SQLI	[86]
Evaluating the urls and performing black box tests on the Web pages	Configuration and deployment management testing	[40]
Fortify	Authentication testing, SQLI	[55]
Fuzz testing tool	Configuration and deployment management testing	[62]
HJ2If	Configuration and deployment management testing	[63]
IAAT	XSS (Stored), SQLI	[79]
IPT-WS	Configuration and deployment management testing	[64]
IMAATT	Authentication testing, XSS (Stored)	[48]
JBroFuzz	Authentication testing, SQLI	[55]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
JWebUnit	XSS (Stored)	[80]
JWAST	XSS (Stored), SQLI, XMLI	[81]
KamaleonFuzz	XSS(Stored)	[51]
Magento	SQLI	[71]
MobSTer	Configuration and deployment management testing, XSS (Stored), SQLI	[65]
MBT	XSS(Stored and reflected)	[82]
Noncespaces	XSS (Stored)	[37]
OAuthTester	Authentication testing	[69]
PHP2XMI	XSS(Stored), SQLI	[83]
PURITY	XSS (Stored), SQLI	[84]
Paros	Authentication testing, SQLI	[55]
PBST	Authentication testing	[56]
RBVT	XSS(Stored), SQLI	[48]
RAD-WS	SQLI	[64]
SAFELI	SQLI	[54]
SAMATE	Session management testing, SQLI	[72]
SAML-based SSO IdP service provided by Google	Configuration and deployment management testing	[66]
SAP HANA XS Applications	SQLI	[85]
SECEVAL	Configuration and deployment management testing	[50]
SSES	XSS (Stored), SQLI	[87]
SPaCIoS	Authorization testing, XSS (Reflected and Stored), SQLI, XMLI	[44]

Continúa en la página siguiente.

Herramienta	Tipo Prueba	Estudio
SQLIVDT *(SQL Injection Vulnerability Detection Tool)*	XSS(Stored), SQLI	[88]
SQLMap	SQLI	[93]
Selenium IDE	Authentication testing, XSS (stored), SQLI	[39]
IDS	SQLI	[57]
Signature evaluation	SQLI	[45]
Tamper data	SQLI (Stored), SQLI	[89]
Tool-prototype	XSS(Stored), SQLI	[38]
WAP	SQLI	[54]
WS-Attacker	Configuration and deployment management testing	[41]
WSAttaker	Configuration and deployment management testing	[68]
WSFAgressor	Configuration and deployment management testing	[67]
WSInject	Configuration and deployment management testing, XSS(Stored)	[53]
WSSecTool	Configuration and deployment management testing	[43]
WSVTS	Configuration and deployment management testing	[52]
WebScarab	Authentication testing, SQLI	[55]
XSSINJECTOR	XSS (stored), Client side testing	[90]
XSS and SQLI security evaluation	XSS(Stored), SQLI	[91]

De acuerdo con el Cuadro 2.7 hay 61 herramientas que evalúan los riesgos más críticos en las aplicaciones Web. Asimismo, evalúan 7 pruebas del *top* 10 declarado por el OWASP. Dentro de las evaluadas se encuentran: *Injection* (A1:2017), *Broken Authentication* (A2:2017), *Sensitive Data Exposure* (A3:2017), *XML External Entities (XXE)* (A4:2017), *Broken Access Control* (A5:2017), *Security Access Control* (A6:2017) y *Cross-Site Scripting* (XSS) (A7:2017).

Ahora bien, si comparamos con la lista de herramientas recomendadas por OWASP [58, 59, 60], existen 5 herramientas de las 66 (encontradas en este mapeo) que recomienda OWASP. Las herramientas que se encuentran mapeadas en este estudio y que también se encuentran en esta lista son: DAST [29], AppScan [61], Burp tool [30], ZAP [28, 29, 30] y Fortify [55] .

2.6.2. Evaluación de la efectividad de las herramientas para las pruebas de seguridad Web (RQ2)

De las 66 herramientas de seguridad Web identificadas, solo hay 41 a las que se les aplicó una evaluación para conocer su efectividad. Cabe resaltar que la efectividad consta de la eficacia y la eficiencia [94].

La eficacia es la capacidad de lograr el resultado deseado o esperado [95]. La eficacia puede constar de la verificación de la funcionalidad de la herramienta y la cantidad de pruebas exitosas y vulnerabilidades encontradas.

La eficiencia es la capacidad de contar con alguien o algo para conseguir un efecto esperado [95]. La eficiencia puede constar de la evaluación del tiempo de ejecución de las pruebas, comparación de características y resultados con otras herramientas, aplicación de la herramienta en casos de estudios, verificación de la cantidad de pruebas ejecutadas y tiempo que le tomó esa tarea a la herramienta, comprobación de la cantidad de falsos positivos encontrados por la herramienta y la comparación del tiempo que toma la herramienta en ejecutar la prueba versus la ejecución manual de las pruebas.

De acuerdo con el mapeo, existen 15 herramientas donde se a aplicaron criterios de eficiencia, por lo que se puede ver hay una tendencia hacia la evaluación de la eficiencia de las herramientas. Mientras tanto, existen 12 herramientas a las que se

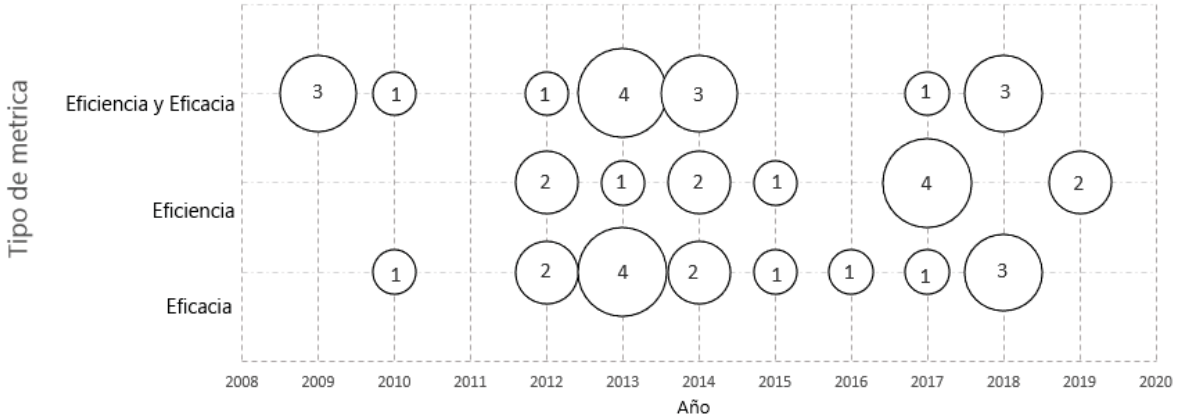


Figura 2.6: Tipos de métricas por año.

les aplicaron criterios de eficacia y a 14 herramientas se les aplicaron ambos criterios.

En la Figura 2.6 se puede ver que, en los años 2013, 2014, 2017 y 2018 se usaron más las métricas para evaluar las herramientas, mientras que en el año 2011 solo una herramienta fue evaluada. Asimismo, el tipo de métrica eficacia fue más evaluada en el año 2013 con 4 herramientas, mientras que el tipo de métrica eficiencia fue mayormente evaluada en el año 2017 con 4 herramientas. Existen 4 herramientas en las que fueron aplicadas ambos tipos de métricas (eficacia y eficiencia) en el año 2018.

En el Cuadro 2.9 se muestra el tipo de criterio usado para evaluar cada una de las herramientas, donde a 15 herramientas se les aplicaron criterios de eficiencia, a 12 herramientas se les aplicaron criterios de eficacia y a 14 herramientas se les aplicaron ambos tipos de criterios.

Cuadro 2.9: Tipos de criterios de evaluación aplicadas en las herramientas.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
MobSTer	Eficacia y eficiencia	La herramienta se ejecuta en 3 casos de estudio y se evalúa cuántas vulnerabilidades se encuentra en cada caso de estudio. Luego, también verifica el comportamiento de la herramienta, si el costo realizar alguna ejecución [65]	Se compara la herramienta con otras 4 herramientas (burp suite, ZAP, Paros y Arachni) y verifica la cantidad de *test* automatizados aplicados, como se aplica, como se usa y la configuración de la herramienta. [65]
ISTA	Eficacia	La herramienta se ejecuta en dos casos de estudio para probar su funcionalidad de la misma [31]	NA
ZAP	Eficacia	Verifica los resultados obtenidos de las pruebas de la herramienta [30]	Evalúa la herramienta por el tipo de solicitud y el tiempo que le toma la herramienta por atender la solicitud [28]

Continúa en la página siguiente.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
PURITY	Eficacia	Evalúa si la herramienta ejecuta todas las pruebas y verifica cuántas vulnerabilidades encuentra la herramienta. [84]	NA
Paros, WebScarab, JBroFuzz, Fortify	Eficacia y Eficiencia	Evalúa los resultados obtenidos al ejecutar las pruebas que cuenta la herramienta y verifica cuántas vulnerabilidades encontró. [55]	Compara los resultados obtenidos de las pruebas automatizadas de la herramienta contra los resultados obtenidos al ejecutar las pruebas de forma manual. [55]
Tool-prototype	Eficiencia	NA	Compara los resultados esperados de la herramienta contra los resultados obtenidos en un caso de estudio. [38]
SQLMap	Eficiencia	NA	Ejecuta la herramienta en 3 intervalos de tiempo (10, 30 y 60 segundos) para conocer cuántas pruebas pudo realizar la herramienta. [93]

Continúa en la página siguiente.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
CRAXweb	Eficacia	Evalúa la herramienta en aplicaciones Web con diferentes lenguajes de programación (*PHP, ruby on rails, django, asp.net*). [76]	NA
SQLIVDT	Eficacia	Evalúa la cantidad de vulnerabilidades encontradas. [88]	NA
SPaCIoS	Eficacia	Evalúa la cantidad de pruebas mutantes que ejecutó. [44]	NA
Selenium IDE	Eficacia	Evalúa las pruebas ejecutadas con éxito en cada escenario. [39]	NA
Circe	Eficacia	Evalúa la cantidad de pruebas ejecutadas en cada caso de estudio aplicado. [77]	NA
IAAT	Eficacia	Evalúa la cantidad de falsos positivos que tuvo la herramienta en cada caso de estudio. [79]	NA

Continúa en la página siguiente.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
JWebUnit	Eficacia	Evalúa las pruebas ejecutadas en una aplicaciones Web. [80]	NA
Signature evaluation	Eficacia y eficiencia	Evalúa las pruebas ejecutadas en varias aplicaciones Web. [45]	Compara los resultados obtenidos al ejecutar las pruebas en varias aplicaciones Web. [45]
XSS and SQLI security evaluation	Eficacia	Evalúa los resultados obtenidos de las pruebas automatizadas de la herramienta. [91]	NA
XSSINJECTOR	Eficiencia	NA	Evalúa la eficiencia de la herramienta al ejecutarla en varias aplicaciones Web. [90]
XSS, black box and SQLI injection point	Eficacia y eficiencia	Verifica que los resultados esperados de la herramienta. [78]	Verifica que los resultados esperados de la herramienta. [78]
KITE	Eficacia	Evalúa la cantidad de pruebas realizadas con éxito o fallidas. [92]	NA

Continúa en la página siguiente.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
Urls and performing black box tests on the Web pages	Eficacia y eficiencia	Verifica que los resultados sean los esperados de la herramienta. [40]	Compara la herramienta en diferentes aplicaciones Web, el tiempo que le llevó la herramienta para evaluar la seguridad de estas aplicaciones Web. [40]
WS-Attacker	Eficiencia	NA	Evalúa los resultados presentados en 4 Web *services*: Apache, Axis, JBoxxWS native, JBoxxWS CXF y .NET Web *services*. [41]
AppScan	Eficiencia	NA	Compara el tiempo y el resultado obtenido de las pruebas automatizadas de la herramienta contra la ejecución manual de estas pruebas. [61]

Continúa en la página siguiente.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
SAFELI	Eficiencia	NA	Evalúa la herramienta por las vulnerabilidades encontradas, objetivo de las pruebas, cantidad de falsos positivos encontrados y solución de la vulnerabilidad. [54]
WAP	Eficiencia	NA	Evalúa la herramienta por las vulnerabilidades encontradas, objetivo de las pruebas, cantidad de falsos positivos encontrados y solución de la vulnerabilidad. [54]
PBST testing tool.	Eficiencia	NA	Evalúa el comportamiento de la herramienta y la cantidad vulnerabilidades encontradas y que son falsos positivos. [56]

Continúa en la página siguiente.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
Deemon	Eficacia y eficiencia	Evalua la funcionalidad de la herramienta ejecutada en 10 populares aplicaciones Web open source. [74]	Compara los resultados obtenidos de las pruebas en 10 populares aplicaciones Web open source. [74]
MBT	Eficiencia	NA	Compara los resultados obtenidos de las pruebas en MBT y PMVT (*Pattern-driven and Model-based vulnerability testing*). [82]
Burp tool	Eficacia	Verifica los resultados obtenidos de las pruebas de la herramienta. [30]	NA
BIOFUZZ	Eficacia y eficiencia	Evalúa los resultados obtenidos al ejecutar las pruebas de la herramienta. [47]	Compara los resultados obtenidos de las pruebas en 4 aplicaciones web. [47].

Continúa en la página siguiente.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
KamaleonFuzz	Eficiencia	NA	Evalúa los resultados de la herramienta al no encontrar ningún falso positivo en las pruebas. Asimismo, compara los resultados contra otro escáner de seguridad de las aplicaciones Web, con el fin de evidenciar que la herramienta encuentra más vulnerabilidades comparado con el otro escáner. [51]
Volcano	Eficacia	Verifica la funcionalidad de la herramienta en 3 aplicaciones Web. [35]	Compara los resultados obtenidos con otra herramienta llamada Paros. [36]
ATUSA	Eficacia y eficiencia	Verifica la funcionalidad de la herramienta en dos casos de estudio. [75]	Compara los resultados obtenidos en cada caso de estudio, verifica que dentro de los resultados no existan falsos positivos. [75]

Continúa en la página siguiente.

Herramienta	T. criterio	Evaluación de la eficacia	Evaluación de la eficiencia
Sign-WS, IPT-WS, RAD-WS	Eficiencia	NA	Evalúa la funcionalidad, los resultados, el tiempo obtenido al ejecutar la herramienta y los falsos positivos en los resultados de la prueba. [64]
WSVTS	Eficacia y eficiencia	Verifica la funcionalidad de la herramienta. [52]	Compara los resultado de la herramienta contra la herramienta SOAPUI ejecutados en un caso de estudio. [52]
WSInject	Eficiencia	NA	Evalúa las vulnerabilidades encontradas y no encontradas por la herramienta. [53]
WSFAgressor	Eficiencia	NA	Evalúa la cantidad de recursos de la computadora que le toma la herramienta para evaluar la seguridad de la aplicación Web. Asimismo, verifica las pruebas que tuvieron éxito y las que fallaron. [67]

Una de las maneras en que más evaluaron la eficiencia fue la ejecución de estas

en varios casos de estudios y comparar los resultados de la herramienta (cantidad de pruebas ejecutadas y tiempo que le tomó la herramienta), de forma que se pueda conocer qué tan eficiente es la herramienta en varios casos de estudio. Las 7 herramientas evaluadas de esta manera fueron SQLMap [93], XSS Injection [90], WSAttack [41], AppScan [61], Sign-WS [64], IPT-WS [64], RAD-WS [64]. Las herramientas Safeli [54], WAP [54], Kamaleon [51], PBST [56] y WSAgressor [67], evaluaron la cantidad de falsos positivos en los resultados de ejecución de las pruebas. Mientras que Tool Prototype [38], MBT [82] y WSInject [53] evaluaron los resultados obtenidos y los compararon con otras herramientas, de forma que se pudiera conocer su eficiencia de la misma.

Mientras que, el modo en que evaluaron la eficacia fue la ejecución de casos de pruebas con éxito y la cantidad de vulnerabilidades encontradas en cada herramienta. Las herramientas evaluadas así fueron [31, 84, 76, 88, 44, 39, 77, 79, 80, 91, 92].

Ahora bien, si se combina la primer pregunta de investigación con la segunda pregunta de investigación vemos que, de acuerdo con la Figura 2.7 existen 14 herramientas que fueron evaluadas por los criterios de eficacia y eficiencia y que también realiza pruebas de *Input Validation testing*, 13 herramientas fueron evaluadas por eficiencia y realiza pruebas de *Input Validation testing* y 11 herramientas fueron evaluadas por la eficacia y *Input Validation testing*.

También vemos que en la Figura 2.7 existen 8 herramientas que fueron evaluadas por la eficacia y eficiencia y que también realiza pruebas de *Session management*, 7 herramientas que evalúa la eficiencia y que también realiza pruebas de *Authentification* y 5 herramientas que fueron evaluadas por eficacia y eficiencia y que realiza pruebas de *Client side*.

En la Figura 2.8 se puede ver las herramientas que han sido evaluadas por la efectividad y subclasificadas por las pruebas de validación de entrada. De acuerdo con la figura existe 10 herramientas que fueron evaluadas por la eficiencia y la eficacia y que realiza pruebas *SQL Injection*, 8 herramientas que fueron evaluadas por la eficiencia y la eficacia y que realiza pruebas de *HTTP request*, 6 herramientas que fueron evaluadas por la eficiencia y la eficacia y que realiza pruebas de *Stored Cross Site Scripting*. También se puede ver que hay 8 herramientas que fueron evaludas por la eficiencia y que realiza pruebas de *SQL Injection* y 9 herramientas que fueron evaluadas por la

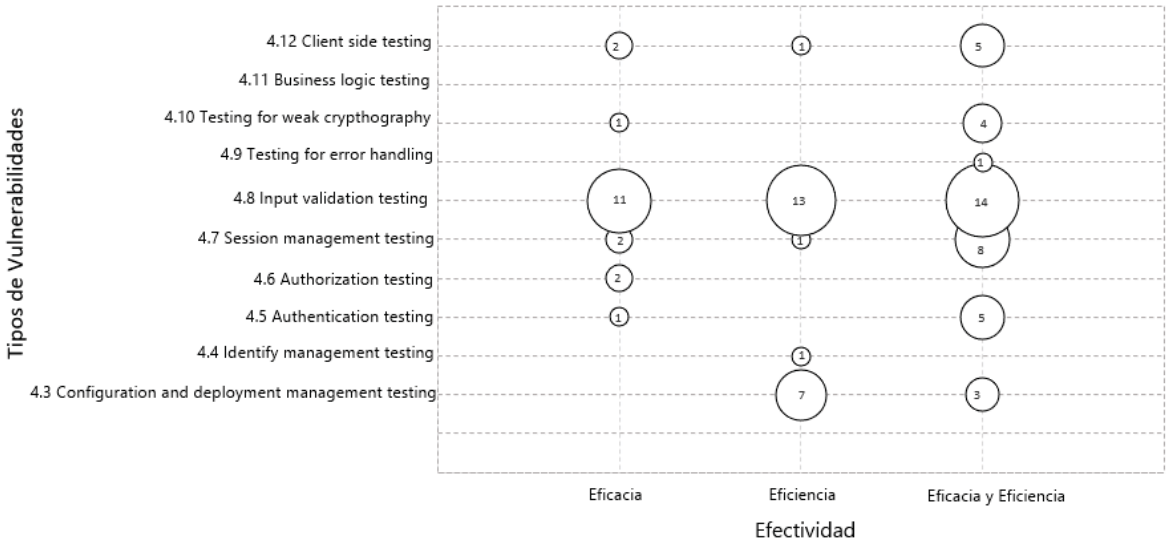


Figura 2.7: Evaluación de la efectividad de las herramientas por tipo de vulnerabilidad.

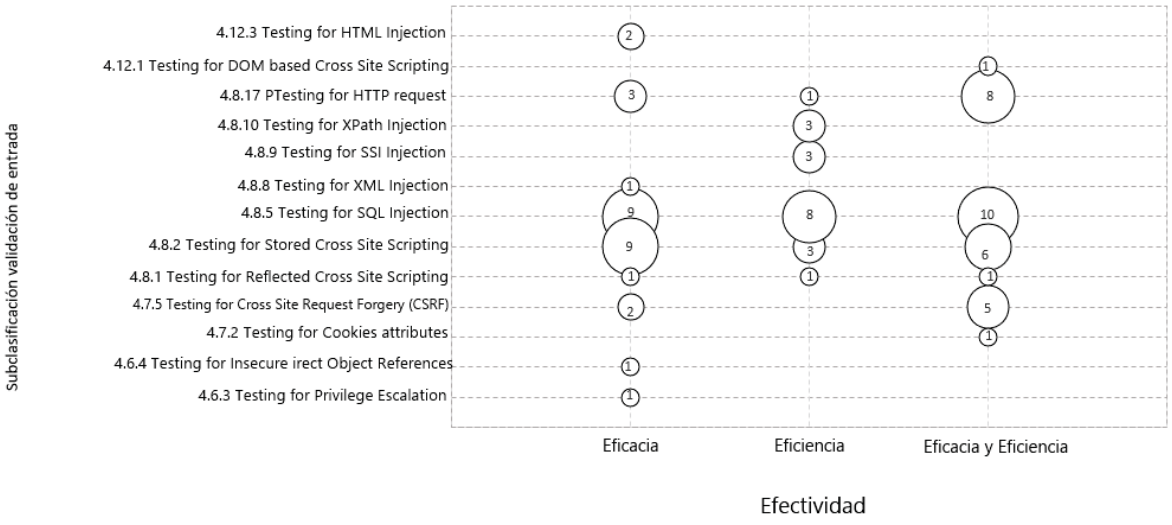


Figura 2.8: Evaluación de la efectividad de las herramientas por subclasificación de pruebas de validación de entrada.

eficacia y que realiza pruebas de *Stored Cross Site Scripting* y *SQL Injection*.

2.7. Discusión

La clasificación de las distintas herramientas disponibles para evaluar cada una de las vulnerabilidades de las aplicaciones Web puede apoyar a los profesionales en la selección de potenciales herramientas para sus procesos de pruebas de seguridad, y en la evaluación de la seguridad de las aplicaciones Web de forma automatizada en distintas fases del ciclo de vida de su desarrollo, o inclusive, cuando se encuentra en producción.

Se identificó gran variedad de herramientas que se utilizan para automatizar distintas pruebas de seguridad de acuerdo con las diferentes vulnerabilidades que existen en la actualidad. Para evaluar la seguridad de las aplicaciones Web, las herramientas deben de contar con pruebas actualizadas de acuerdo con las vulnerabilidades y ataques cibernéticos actuales. La identificación de herramientas demostró que se mantiene la tendencia de realizar pruebas para la *SQL Injection* y *Cross-Site Scripting*. Sin embargo, no se encontraron herramientas para algunos de los escenarios reportados como los 10 ataques más riesgosos y frecuentes por *OWASP*, tales como: *Insecure Deserialization, Using Components with known vulnerabilities* e *Insufficient Logging & Monitoring*.

2.8. Lecciones aprendidas

El desarrollo de un protocolo de mapeo sistemático de la literatura permite un mayor desenvolvimiento del tema de investigación. Es esencial el uso de este tipo de metodología para tener un mayor orden, contar con una base sólida al conocer los estudios con mayor calidad de información para desarrollar la investigación.

De igual manera será de gran utilidad para futuros practicantes de la Ingeniería de *Software*, ya que permite un mayor desenvolvimiento del estudio, conocer el análisis que facilitan la investigación y la calidad que ofrecen para desarrollarla.

Este tipo de metodología posibilita conocer el estado actual del tema, la problemática y desarrollar el objetivo GQM a partir de la problemática encontrada. Asimismo,

facilita llevar un proceso estructurado, al conocer cada etapa de la investigación, contando con un panorama de la estabilidad proceso.

Cabe mencionar que este sistema metodológico es algo agotador al estar constantemente realizando etapas de forma cíclica para encontrar los mejores estudios y exponer los resultados de la investigación, por lo cual no sería una buena opción integrarla en alguna organización de desarrollo de *Software*. No obstante, lo favorable de este tipo de metodología es el constante trabajo al encontrar estudios de gran calidad para el desarrollo de la investigación, permitiendo mostrar resultados puntuales, válidos y con bases consistentes.

Como una fase retadora de este tipo de metodología se puede mencionar el desarrollo de una cadena de búsqueda que cumpliera con todos los requisitos para encontrar los mejores estudios. Posteriormente, estos estudios fueron evaluados con varios puntos de calidad para definir un puntaje de la validez de la información que brinda el estudio. Para esto, se revisó cada estudio hasta extraer la información de calidad necesaria para la investigación.

A pesar de la fase retadora, este tipo de metodología permitió dar un buen panorama del tema y poder desarrollar el estudio y así definir el trabajo futuro de esta investigación.

2.9. Conclusiones

Existe una gran cantidad y diversidad de herramientas que evalúan la seguridad de las aplicaciones Web y estas pruebas se realizan de forma automatizada, donde el rendimiento es óptimo, comparado con la realización de las pruebas para evaluar la seguridad de forma manual.

Existe gran preocupación por cubrir la mayor cantidad de escenarios para encontrar cualquier vulnerabilidad con que cuentan las aplicaciones Web, de forma que los desarrolladores puedan solventar esta problemática lo más pronto posible para estas no sufran de ataques cibernéticos. Las herramientas identificadas en este estudio cubren al menos un caso de prueba para evaluar la seguridad de las aplicaciones Web. Asimismo, hay herramientas que evalúan gran cantidad de escenario, sin embargo, no cubren todos los escenarios acordes con los 10 ataques con más riesgos y más

frecuentes de acuerdo con el OWASP.

Además, aunque se identificó gran variedad de herramientas que realizan distintos tipos de pruebas de seguridad, solo pocas se encuentran en la lista de herramientas recomendadas por OWASP, lo que denota la necesidad de contar con evaluaciones empíricas de herramientas existentes en el área.

Se determinó que existen criterios de evaluación para las herramientas que evalúan la seguridad de forma automatizada de las aplicaciones Web de forma automatizada. A pesar de que este criterio se enfoca o solo en la eficiencia o solo en el rendimiento, no hay un gran porcentaje donde las herramientas sean evaluadas por ambos criterios de evaluación. Tampoco se ha encontrado un criterio de evaluación con respecto a la interoperabilidad para aquellas herramientas que evalúan la seguridad de las aplicaciones Web que se encuentran en producción.

Ahora bien, se ha mostrado en este mapeo sistemático, la existencia de las herramientas con sus diferentes tipos de pruebas para evaluar la seguridad de las aplicaciones Web. No obstante, dentro de los limitantes que se obtuvo está el acceso a estas herramientas, ya que los estudios solo se enfocaron en la creación y exposición de la herramienta para evaluar la seguridad de las aplicaciones Web, pero, no ofrecen una forma de fuente para obtener estas herramientas.

A nivel de contribuciones de este tema de investigación, en el ámbito educativo se incentiva a los estudiantes de los cursos de seguridad Web, pruebas Web, calidad de software e Ingeniería de *Software* para el desarrollo y análisis de herramientas que cuenten con pruebas de seguridad para las aplicaciones Web. A nivel profesional se incentiva el uso de las herramientas para identificar las vulnerabilidades que presenta las aplicaciones Web. En el campo de la investigación se incentiva el realizar más investigaciones de este tema y de vulnerabilidades que las herramientas aún no han podido detectar.

Como trabajo futuro, se propone seleccionar un conjunto de herramientas de pruebas de seguridad que permitan verificar las principales vulnerabilidades reportadas para las aplicaciones Web y evaluar su efectividad con el fin de generar evidencia para la industria. Asimismo, como trabajo futuro, será interesante identificar las clasificaciones de vulnerabilidades brindadas por OWASP que no fueron cubiertas por ninguna herramienta del mapeo y realizar un estudio de la razón por la cual estas

no evalúan estas vulnerabilidades y la importancia de que las herramientas ejecuten pruebas para estas vulnerabilidades.

A partir del desarrollo de este estudio, se llevó a cabo un estudio científico que fue enviado y aceptado en *Iberoamerican Conference on Software Engineering* que se desarrollará el 16-20 de Noviembre, en Curitiba, Brasil. En el Apéndice 2.G se encuentra el artículo publicado.

Apéndice

2.A. Lista de estudios primarios incluidos

El Cuadro 2.10 presenta una lista de los estudios que permitió el mapeo sistemático de este trabajo.

Cuadro 2.10: Lista de estudios primarios incluidos.

ID	Título	Año	Est.
1	MobSTer: A model-based security testing framework for Web applications	2018	[65]
2	Automated Security Test Generation with Formal Threat Models	2012	[31]
3	Noncespaces: Using randomization to defeat cross-site scripting attacks	2012	[37]
4	Security testing as a service with docker containerization	2018	[28]
5	Automated Reverse Engineering of UML Sequence Diagrams for Dynamic Web Applications	2009	[83]
6	PURITY: A Planning-based secURITY Testing Tool	2015	[84]

Continúa en la página siguiente.

ID	Título	Año	Est.
7	A case study on Web application security testing with tools and manual testing	2013	[55]
8	Grammar based oracle for security testing of Web applications	2012	[38]
9	SPaCiTE – Web Application Testing Engine	2012	[33]
10	Towards Continuous Security Certification of Software-as-a-Service Applications Using Web Application Testing Techniques	2017	[93]
11	Mutation Analysis of Magento for Evaluating Threat Model-Based Security Testing	2011	[71]
12	Supporting Security Testers in Discovering Injection Flaws	2008	[87]
13	Web Application Scanners: Definitions and Functions	2007	[72]
14	CRAXweb: Automatic Web Application Testing and Attack Generation,	2013	[76]
15	A black-box testing tool for detecting SQL injection vulnerabilities	2013	[88]
16	Model inference and security testing in the spacios project	2014	[44]
17	Mining Executable Specifications of Web Applications from Selenium IDE Tests	2012	[39]

Continúa en la página siguiente.

ID	Título	Año	Est.
18	Circe: A grammar-based oracle for testing Cross-site scripting in Web applications	2013	[77]
19	A Novel Injection Aware Approach for the Testing of Database Applications	2010	[79]
20	Automatic Web Security Unit Testing: XSS Vulnerability Detection,	2016	[80]
21	Automated Test Generation from Vulnerability Signatures	2014	[45]
22	XSS pattern for attack modeling in testing	2013	[91]
23	Code Pulse: Real-time code coverage for penetration testing activities	2015	[70]
24	Practical Combinatorial Testing for XSS Detection using Locally Optimized Attack Models	2019	[90]
25	Testing Security Policies for Web Applications	2008	[63]
26	Scalable Quality and Testing Lab (SQTL): Mission-Critical Applications Testing	2019	[46]
27	An automated vulnerability scanner for injection attack based on injection point	2010	[78]
28	Real-time communication testing evolution with WebRTC 1.0	2017	[92]

Continúa en la página siguiente.

ID	Título	Año	Est.
29	Solving Some Modeling Challenges when Testing Rich Internet Applications for Security	2012	[40]
30	Penetration Testing Tool for Web Services Security,	2012	[41]
31	Locality-Sensitive hashing for efficient Web application security testing	2019	[61]
32	Comparison of security testing approaches for detection of SQL injection vulnerabilities	2018	[54]
33	Pattern based Web security testing	2018	[56]
34	Experimental evaluation of security requirements engineering benefits	2018	[29]
35	Deemon: Detecting CSRF with dynamic analysis and property graphs	2017	[74]
36	Testing application security with aspects	2016	[73]
37	Using agent technology for security testing of WEB based applications	2015	[81]
38	Risk-driven vulnerability testing: Results from eHealth experiments using patterns and model-based approach	2015	[82]

Continúa en la página siguiente.

ID	Título	Año	Est.
39	On the applicability of combinatorial testing to Web application security testing: A case study	2014	[30]
40	Search-based security testing of Web applications	2014	[47]
41	Risk-based vulnerability testing using security test patterns	2014	[48]
42	An integrated multi-agent testing tool for security checking of agent-based Web applications	2014	[49]
43	SecEval: An evaluation framework for engineering secure systems	2014	[50]
44	KameleonFuzz: Evolutionary fuzzing for black-box XSS detection	2014	[51]
45	A tool for automated test code generation from high-level Petri nets	2011	[32]
46	Security sensitive data flow coverage criterion for automatic security testing of Web applications	2011	[35]
47	Idea: Using system level testing for revealing SQL injection-related error message information leaks	2010	[86]
48	Automated security testing of Web widget interactions	2009	[75]

Continúa en la página siguiente.

ID	Título	Año	Est.
49	Idea: Automatic security testing for Web applications	2009	[36]
50	Towards an attack signature generation framework for intrusion detection systems	2018	[57]
51	Designing vulnerability testing tools for Web services: approach, components, and tools	2017	[64]
52	Model-based security testing: An empirical study on OAuth 2.0 implementations	2017	[64]
53	Tool support for secure programming by security testing	2015	[85]
54	Security testing of orchestrated business processes in SOA	2015	[68]
55	Worst-input mutation approach to Web services vulnerability testing based on SOAP messages	2014	[52]
56	Security testing methodology for vulnerabilities detection of XSS in Web services and WS-security	2014	[53]
57	Complete Web security testing methods and recommendations	2013	[89]
58	WSFAggressor: An extensible Web service framework attacking tool	2013	[67]
59	Remote agent based automated framework for threat modelling, vulnerability testing of SOA solutions and Web services	2012	[42]

Continúa en la página siguiente.

ID	Título	Año	Est.
62	SPaCiTE - Web application testing engine	2012	[34]
61	Model-checking driven security testing of Web-based applications	2010	[66]
62	A Model-based Approach to the Security Testing of Network Protocol Implementations	2006	[62]
63	WSSecTool: A Web Service Security Analysis Tool Based on Program Slicing	2012	[43]

2.B. Evaluación de calidad de los estudios primarios

El Cuadro 2.11 muestra los resultados de la evaluación de calidad de todos los estudios analizados. Para cada criterio se evaluó en una escala de 0 a 2 puntos, por lo que la calidad se mide sobre 4 puntos.

Cuadro 2.11: Evaluación de calidad de los estudios primarios.

ID	Estudio	Año	Q1	Q2	Total
1	[65]	2018	2	1	3
2	[31]	2012	1	2	3
3	[37]	2012	1	0	1
4	[28]	2018	1	2	3
5	[83]	2009	1	0	1
6	[84]	2015	2	2	4
7	[55]	2013	1	2	3
8	[38]	2012	1	2	3
9	[33]	2012	1	0	1
10	[93]	2017	1	2	3
11	[71]	2011	1	0	1
12	[87]	2008	2	0	2
13	[72]	2007	1	0	1
14	[76]	2013	2	2	4
15	[88]	2013	1	1	2
16	[44]	2014	1	2	3
17	[39]	2012	2	2	4
18	[77]	2013	1	2	3
19	[79]	2010	1	1	2
20	[80]	2016	1	2	3
21	[45]	2014	1	2	3
22	[91]	2013	2	2	4

Continúa en la página siguiente.

ID	Estudio	Año	Q1	Q2	Total
23	[70]	2015	1	0	1
24	[90]	2019	2	2	4
25	[63]	2008	1	0	1
26	[46]	2019	1	0	1
27	[78]	2010	1	2	3
28	[92]	2017	1	1	2
29	[40]	2012	1	2	3
30	[41]	2012	2	2	4
31	[61]	2019	1	2	3
32	[54]	2018	1	1	2
33	[56]	2018	1	2	3
34	[29]	2018	1	0	1
35	[74]	2017	2	1	3
36	[73]	2016	1	0	1
37	[81]	2015	2	0	2
38	[82]	2015	1	1	2
39	[30]	2014	1	2	3
40	[47]	2014	1	2	3
41	[48]	2014	1	0	1
42	[49]	2014	2	0	2
43	[50]	2014	1	0	1
44	[51]	2014	2	2	4
45	[32]	2011	1	0	1
46	[35]	2011	1	2	3
47	[86]	2010	1	0	1
48	[75]	2009	1	2	3
49	[36]	2009	1	2	3
50	[57]	2018	1	0	1
51	[64]	2017	1	2	3

Continúa en la página siguiente.

ID	Estudio	Año	Q1	Q2	Total
52	[64]	2017	1	0	1
53	[85]	2015	1	0	1
54	[68]	2015	1	0	1
55	[52]	2014	1	2	3
56	[53]	2014	1	2	3
57	[89]	2013	1	0	1
58	[67]	2013	1	2	3
59	[42]	2012	1	0	1
60	[34]	2012	1	0	1
61	[66]	2010	1	0	1
62	[62]	2006	1	0	1
63	[43]	2012	1	0	1

2.C. Descripción de las herramientas

El Cuadro 2.12 muestra la descripción de las herramientas expuestas en los estudios analizados.

Cuadro 2.12: Descripción de las herramientas.

Herramienta	Descripción	Estudio
AOP	AOP permite encontrar errores de seguridad en aplicaciones Web utilizando servlets y también realizando pruebas de fuzzing en aplicaciones java. El lenguaje de programación utilizado en esta herramienta es java.	[73]
AppScan	Consiste en aplicar el rastreo y el mecanismo LHS para evaluar la seguridad de las aplicaciones Web.	[61]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
Attack signatures and interface monitoring (Sign-WS)	Sign-WS: es una herramienta con enfoque automatizado que utiliza firmas de ataque y monitoreo de interfaz para la detección de la vulnerabilidad de la inyección.	[64]
ATUSA	Atusa se basa en las capacidades de rastreo de *Crawljax* y proporciona detección de puntos de entrada de datos y ganchos de complementos (previos, internos y posteriores al rastreo) para probar aplicaciones *Ajax* a través de invariantes genéricos y específicos de la aplicación que sirven como oráculo para detectar fallas.	[75]
BIOFUZZ	BIOFUZZ, un probador de seguridad para aplicaciones Web que utiliza pruebas evolutivas de caja negra 3 para detectar vulnerabilidades, específicamente inyecciones SQL.	[47]
Burp tool	Burp suite es una plataforma integrada para realizar pruebas de seguridad de aplicaciones Web. Fue diseñado para cumplir con muchas tareas de pruebas de penetración y ayudar a los profesionales de seguridad en cada paso de una herramienta de prueba.	[30]
CRAXweb	Aplicar test cases (sobre todo *front-end*) en aplicaciones Web	[76]
Circe	Circe es una herramienta enfocada para construir seguridad oracle para una de las clases de código más prominentes en inyección de código, tal como lo es cross-site scripting. El lenguaje de programación utilizado en esta herramienta es php.	[77]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
CodePulse	Consiste en la ejecución de Glass box para aplicaciones Web. El lenguaje de programación utilizado en esta herramienta es java.	[70]
DAST	Es un método de prueba de recuadro negro aplicado en la ejecución de aplicaciones desde el exterior. Estas herramientas funcionan mediante la ejecución de *scripts* de ataque predefinidos que envían una solicitud a la aplicación Web. La respuesta de la aplicación Web a la herramienta se analiza para determinar la existencia de una vulnerabilidad. Cada herramienta tiene sus propios *scripts* y sus propios parámetros para configurar la prueba de seguridad.	[29]
Deemon	Es el primer marco de pruebas de seguridad automatizado para descubrir vulnerabilidades CSRF.	[74]
Detection of vulnerabilities of Web applications (XSS, black box and SQLI	Consiste en la detección de vulnerabilidades de aplicaciones Web (XSS, caja negra y SQLI) en función del punto de inyección.	[78]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
Eclipse IDE	El framwork tomará prestado implementaciones de varias pruebas de seguridad como inyección de SQL, Inyección Xpath, Escaneo difuso, Tipos inválidos, Límite Escaneo, XML con formato incorrecto, bomba XML, etc. desde código abierto probando soluciones como SOAPUI. El lenguaje de programación utilizado en esta herramienta es java.	[42]
Eclipse IDE test cases execution	Herramienta que ejecuta test cases de forma automatizada para evaluar la seguridad de una aplicación Web. El lenguaje de programación utilizado en esta herramienta es java.	[86]
Evaluating the urls and performing black box tests on the Web pages	consiste en rastrear páginas Web, evaluar las URL y realizar pruebas de recuadro negro en las páginas Web	[40]
Fortify	Fortify es una herramienta de análisis estático utilizada para encontrar las causas raíz de vulnerabilidades de seguridad en el código fuente	[55]
Fuzz testing tool	Consiste en la implementación del algoritmo propuesto en este estudio. El fin de esta herramienta es hacerse cargo de la entrada y salida del cliente MSN, desarrollando un proxy SOCKS v5 especial y forzar al cliente a usarlo.	[62]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
HJ2If	HJ2If se basa en el algoritmo *Hit-or-Jump* que se usa especialmente para las pruebas de componentes para realizar la generación de secuencias de prueba desde la especificación del sistema Web	[63]
IAAT	Esta herramienta permite tomar en cuenta los problemas de inyección SQL en diferentes aplicaciones y resolverse con soluciones a la medida. El lenguaje de programación utilizado en esta herramienta es java.	[79]
ISTA	ISTA es una herramienta que consiste en la ejecución de los casos de prueba de seguridad implementados desde la Descripción de Implementación del Modelo de Amenaza (TMID). El lenguaje de programación utilizado en esta herramienta es java.	[31, 32]
Improved penetration testing (IPT-WS)	PT-WS apunta a la detección de vulnerabilidades de inyección en servicios al alcance pero no bajo control.	[64]
IMAATT	Esta herramienta tiene como objetivo encontrar vulnerabilidades estáticas y dinámicas en aplicaciones Web.	[48]
JBroFuzz	JBroFuzz es una aplicación Web *fuzzer* escrita en java. Consiste en realizar constantes solicitudes en el HTTP con el fin de encontrar vulnerabilidades tales como *Injection, XSS, Interger OVerflox, XPATH Injection y SQLI*.	[55]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
JWebUnit	Consiste en ejecutar las pruebas en las páginas Web y verificar el resultado. El lenguaje de programación utilizado en esta herramienta es java.	[80]
JWAST	Es una herramienta de prueba estática y dinámica que se basa en nuestra técnica integrada de análisis estático y dinámico.	[81]
KITE	KITE es un código abierto, genérico, reutilizable y muy fácil de mantener un entorno de prueba automatizado para probar la interoperabilidad P2P de WebRTC en todos los tipos de clientes que cumplen con WebRTC.	[92]
KamaleonFuzz	KameleonFuzz, un *fuzzer blackbox Cross Site Scripting (XSS)* para aplicaciones Eeb. KameleonFuzz no solo puede generar entradas maliciosas para explotar XSS, sino también detectar qué tan cerca está revelando una vulnerabilidad.	[51]
Magento	Herramienta que aplica y evalúa test cases de seguridad para aplicaciones Web y también aplica pruebas mutantes.	[71]
MobSTer: *Model-based Security Testing Framework*	MobSTer es un modelo basado en un *framework* para las pruebas de seguridad. Esta herramienta apoya a un analista de seguridad en la conducción de pruebas de seguridad de aplicaciones Web.	[65]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
MBT *Model-Based Testing* (MBT)	Las pruebas basadas en modelos (MBT) son un enfoque de prueba de software en el que tanto los casos de prueba como los resultados esperados se derivan automáticamente de un modelo abstracto del sistema bajo prueba (SUT).	[82]
Noncespaces	Noncespaces es una herramienta que cuenta una técnica que permite a los clientes Web distinguir entre contenido confiable y no confiable para evitar la explotación de vulnerabilidades XSS. El lenguaje de programación utilizado en esta herramienta es php.	[37]
OAuthTester	Esta herramienta permite el descubrimiento automático de vulnerabilidades a través de pruebas y evaluaciones sistemáticas de implementaciones de *OAuth*.	[69]
PHP2XMI	PHP2XMI es una parte esencial de un *Framework* destinado a probar la conformidad de las aplicaciones Web dinámicas con las políticas de seguridad de control de acceso basadas en roles. El lenguaje de programación utilizado en esta herramienta es php.	[83]
PURITY	PURITY ejecuta casos de prueba contra un sitio Web determinado. Detecta si el sitio Web es vulnerable contra algunas de las vulnerabilidades más comunes, es decir, inyecciones SQL y secuencias de comandos entre sitios.	[84]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
Paros	Paros es una evaluación de vulnerabilidad de aplicación Web gratuita. La herramienta está escrita en java. La herramienta escanea la aplicación Web para identificar comunes vulnerabilidades como secuencias de comandos entre sitios, inyección SQL,formularios con autocompletado habilitado, versiones antiguas de archivos, etc.	[55]
Pattern Based Security Testing tool (PBST)	Es una herramienta capaz de probar el bloqueo de cuentas y la aplicación de autenticación basada en la seguridad del patrón.	[56]
Risk-based vulnerabilities testing (RBVT)	La herramienta consiste impulsar la generación de pruebas con respecto a los resultados de evaluación de riesgos y el uso de patrones de prueba de vulnerabilidad dedicados.	[48]
Runtime anomaly detection (RAD-WS)	Es una herramienta que tiene como enfoque automatizado para la detección de vulnerabilidades de inyección basadas en la detección de anomalías y eso incluye dos pasos principales.	[64]
SAFELI	SAFELI es una herramienta de análisis estático para identificar SQLIV en el código de *bytes* de las aplicaciones Web ASP.NET en tiempo de compilación. SAFELI utiliza un motor de ejecución simbólico para analizar el código fuente. El lenguaje de programación utilizado en esta herramienta es ASP.NET.	[54]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
SAMATE	El proyecto SAMATE es la identificación y medición de herramientas de garantía de seguridad de software, incluidos escáneres de aplicaciones Web	[72]
SAML-based SSO IdP service provided by Google	SAML SSO define un formato basado en XML para codificar aserciones de seguridad, así como una serie de protocolos y enlaces que prescriben cómo se deben intercambiar las aserciones en una variedad de aplicaciones y/o escenarios de implementación.	[66]
SAP HANA XS Applications	Es una integración IDE de pruebas de seguridad y análisis de código estático para detectar vulnerabilidades y patrones de codificación inseguros conocidos de acuerdo con las pautas de programación segura.	[85]
SECEVAL	SECEVAL define un modelo gráfico, que comprende un modelo de contexto de seguridad que describe propiedades de seguridad, vulnerabilidades y amenazas, así como métodos, anotaciones y herramientas	[50]
SSES: *Software security evaluation system*	El SSES se ha desarrollado a partir de un conjunto de scripts auxiliares hacia un concepto que al final puede producir un comprensión más profunda de vulnerabilidades y pruebas de seguridad. El lenguaje de programación utilizado en esta herramienta es java.	[87]
SPaCIoS	Consiste en la evaluación y metodología para evaluar la seguridad Web.	[44]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
SPaCiTE	Esta herramienta se basa en un verificador de modelos dedicado para análisis de seguridad que genera posibles ataques con respecto a vulnerabilidades comunes en aplicaciones Web.	[33, 34]
SQLIVDT: textit(SQL Injection Vulnerability Detection Tool)	Consiste en una herramienta de escáner que detecta vulnerabilidades SQLI. El lenguaje de programación utilizado en esta herramienta es java.	[88]
SQLMap	Esta herramienta se basa en un verificador de modelos dedicado para análisis de seguridad que genera posibles ataques con respecto a vulnerabilidades comunes en la Web.	[93]
SQTL: *Scalable Quality and Testing Lab*	En una plataforma para realizar pruebas de forma manual o automatizada en páginas Web	[46]
Selenium IDE	Consiste en automatizar casos de prueba previamente programados y evaluarlo en las páginas Web. El lenguaje de programación utilizado en esta herramienta es java.	[39]
IDS: *Signature based intrusion detection systems*	Sistema que tiene firmas de ataque.	[57]
Signature evaluation	Consiste en la evaluación de la firma	[45]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
Tamper data	Tamper Data es un complemento del navegador Firefox, podemos usar TAMPER DATA para ver y modificar los encabezados HTTP / HTTPS y los parámetros POST.	[89]
Tool-prototype	Es un prototipo que consiste en evaluar las vulnerabilidades de SQL en una aplicación Web. El lenguaje de programación utilizado en esta herramienta es php.	[38]
Volcano	Volcano realiza pruebas de seguridad de caja blanca para encontrar vulnerabilidades de inyección SQL en aplicaciones Web escritas en lenguaje PHP.	[35, 36]
WAP	El enfoque WAP se basa en un análisis de contaminación combinado con minería de datos para reducir la tasa de falsos positivos. Utiliza el análisis de flujo de contaminación para rastrear la propagación de la entrada no confiable a través de un árbol generador de aplicaciones que describe las rutas de flujo de control vulnerables (desde un punto de entrada a un sumidero sensible).	[54]
WS-Attacker	Consiste en una herramienta de prueba de penetración automatizada para servicios Web	[41]
WSAttaker	WS-Attacker para analizar algunas vulnerabilidades de seguridad presentes en los procesos comerciales de SOA.	[68]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
WSFAgressor	Es herramienta extensible y configurable que se puede utilizar para realizar pruebas de seguridad en marcos de servicios Web. El lenguaje de programación utilizado en esta herramienta es java.	[67]
WSInject	WSInject es una nueva herramienta de inyección de fallas, que introduce fallas o errores en los servicios Web para analizar el comportamiento en un entorno no robusto.	[53]
WSSecTool	Esta herramienta incluye tres módulos, a saber, módulo de corte, módulo de publicación de seguridad y módulo de prueba. El módulo de división analiza los códigos fuente del servicio Web para generar un gráfico de dependencia de métodos (MDG). El lenguaje de programación utilizado en esta herramienta es java.	[43]
WSVTS	WSVTS contiene cuatro módulos de funciones principales: (a) el generador de mensajes SOAP; (b) el generador de mutación de mensajes SOAP; (c) el generador de casos de prueba; y (d) el analizador de vulnerabilidades del servicio Web. El lenguaje de programación utilizado en esta herramienta es C.	[52]
WebScarab	WebScarab es una aplicacion Web open soure para pruebas de seguridad Web. Está escrito en java. Esta herramienta se puede usar para analizar aplicaciones que se comunican con HTTP Protocolos HTTPS.	[55]

Continúa en la página siguiente.

Herramienta	Descripción	Estudio
XSSINJECTOR	XSSINJECTOR ejecuta vectores de ataque generados a partir de nuestra metodología contra aplicaciones Web. La herramienta emplea un oráculo de prueba recientemente desarrollado para detectar XSS que nos permite identificar con precisión si *JavaScript* ejecutado realmente se ejecuta y, por lo tanto, eliminar falsos positivos. El lenguaje de programación utilizado en esta herramienta es python.	[90]
XSS and SQLI security evaluation	Consiste en evaluar la seguridad XSS y SQLI de las aplicaciones Web. El lenguaje de programación utilizado en esta herramienta es php.	[91]
ZAP	La herramienta realiza escaneos dinámicos de seguridad. Utiliza el complemento *FindSecBug* para realizar una verificación de seguridad de código estático y una herramienta de verificación de dependencia OWASP para verificar amenazas de seguridad en bibliotecas de terceros.	[28, 29, 30]

2.D. Cantidad de herramientas por categoría y subcategoría de OWASP

En la Figura 2.9 presenta la cantidad de herramientas por categoría y subcategoría de OWASP.

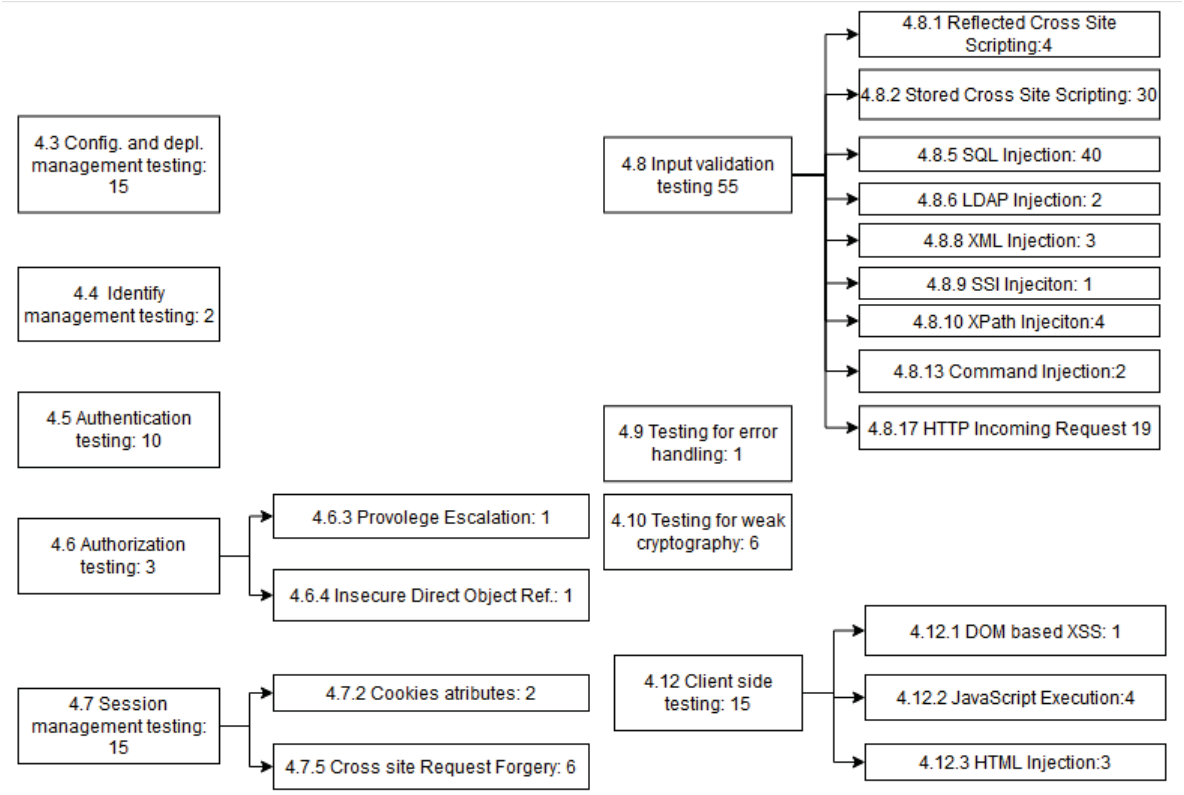


Figura 2.9: Cantidad de herramientas por categoría y subcategoría de OWASP.

2.E. Cantidad de herramientas del primer nivel de OWASP

En la Figura 2.10 presenta la cantidad de herramientas del primer nivel de OWASP

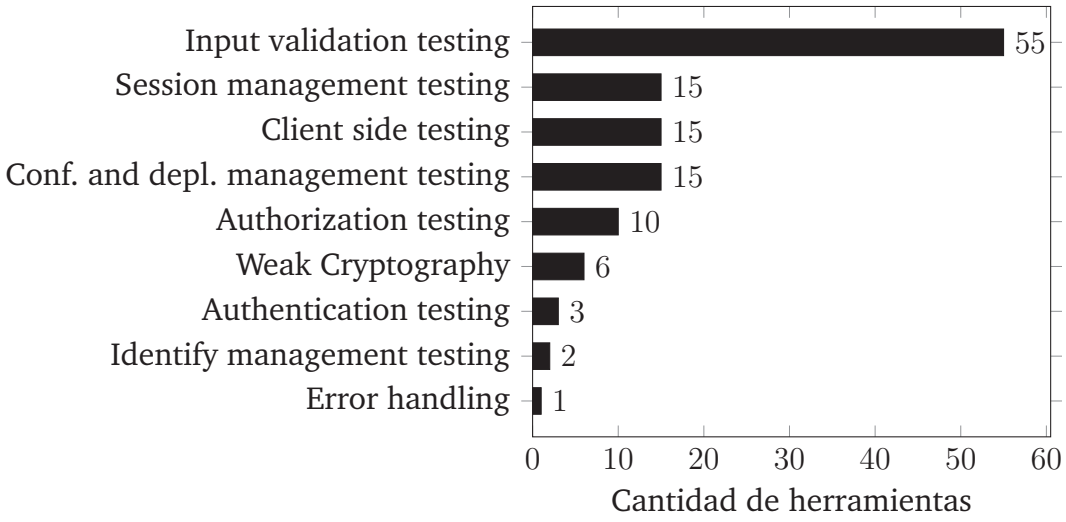


Figura 2.10: Cantidad de herramientas por categoría de primer nivel de OWASP.

2.F. Cantidad de herramientas por segundo nivel de OWASP.

En la Figura 2.11 presenta la cantidad de herramientas por segundo nivel de OWASP.

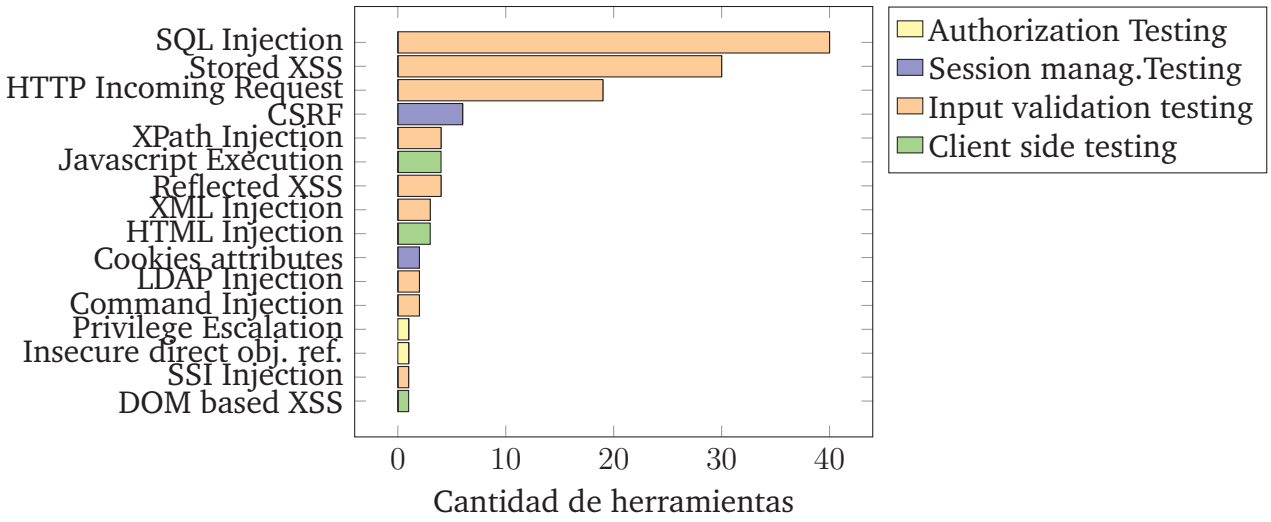


Figura 2.11: Cantidad de herramientas por subcategoría de segundo nivel por OWASP.

2.G. Artículo

A partir del desarrollo de este estudio, se generó un artículo científico el cual fue enviado y aceptado en *Iberoamerican Conference on Software Engineering* que se desarrollará el 16-20 de Noviembre, en Curitiba, Brasil.

Herramientas de pruebas automatizadas de seguridad para aplicaciones Web: Un mapeo sistemático de la literatura

Elizabeth Gamboa, Christian Quesada-López, Alexandra Martínez, Marcelo Jenkins

Universidad de Costa Rica, San Pedro, Costa Rica
{elizabeth.gamboa, cristian.quesadalopez, alexandra.martinez, marcelo.jenkins}@ucr.ac.cr

Resumen Las herramientas utilizadas para automatizar las pruebas de seguridad en aplicaciones Web son esenciales para detectar vulnerabilidades y prevenir ataques cibernéticos. En este estudio identificamos herramientas utilizadas entre el 2006 al 2019 para probar la seguridad de aplicaciones Web, en términos de los tipos de vulnerabilidades que detectan. Para ello, realizamos un mapeo sistemático de la literatura en el que se analizaron 63 estudios primarios, de los cuales identificamos en el mapeo 66 herramientas utilizadas para realizar pruebas automatizadas de seguridad. Las herramientas se clasificaron según los tipos de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web. La categoría de pruebas para detectar vulnerabilidades más común fue la de *Input Validation Testing (4.8)* con 55 herramientas, seguido de las pruebas de *Configuration and Deployment Management Testing (4.3)*, *Session Management Testing (4.7)*, y *Client Side Testing (4.12)* con 15 herramientas utilizadas cada una. Los tipos de pruebas más reportados fueron los de la categoría *Input Validation Testing (4.8)*. En este caso *SQL Injection (4.8.5)* con 40 herramientas, *Cross-Site Scripting (4.8.2)* con 30 herramientas, y *Testing for HTTP Incoming Requests (4.8.17)* con 19 herramientas utilizadas.

Palabras clave: Pruebas de seguridad, herramientas, aplicaciones Web, vulnerabilidades, OWASP, mapeo sistemático.

1. Introducción

Las pruebas de seguridad para aplicaciones Web son esenciales para prevenir la explotación de vulnerabilidades que generalmente buscan comprometer o dañar un sistema y la información que este administra. Se estima que más del 90 % de las aplicaciones Web son vulnerables, con una media de más de 10 vulnerabilidades por aplicación [1]. Una prueba de seguridad para las aplicaciones Web es un método para evaluar la seguridad de un sistema informático mediante la validación y verificación metódica de la efectividad de los controles de seguridad de la aplicación [2]. Una prueba de seguridad de aplicaciones Web se centra en evaluar la seguridad de una aplicación Web. El proceso implica un análisis

activo de la aplicación en busca de debilidades, fallas técnicas o vulnerabilidades [2]. Para proteger las aplicaciones Web es necesario identificar y eliminar las vulnerabilidades que estas presentan. Una vulnerabilidad es una falla o debilidad en el diseño, implementación, operación o administración de un sistema que podría explotarse para comprometer dicho sistema [2]. Las herramientas de pruebas automatizadas pueden complementar los procesos de pruebas manuales para brindar mayor confiabilidad en la cobertura y para reducir el tiempo de ejecución de los casos de prueba de seguridad. Asimismo, pueden apoyar a los equipos de desarrolladores para ejecutar estos tipos de pruebas. Por otro lado, estas herramientas cuentan con limitantes y retos para incrementar el valor agregado que pueden ofrecer a los equipos de calidad [3,4].

El proyecto abierto de seguridad de aplicaciones Web se dedica a promocionar el desarrollo de aplicaciones confiables. Este provee recursos abiertos sobre herramientas, documentos y capítulos para cualquier interesado en mejorar la seguridad de las aplicaciones [5]. *OWASP* detalla los lineamientos para pruebas de seguridad en el cual lista las vulnerabilidades que las pruebas de seguridad Web deben evaluar [5].

En los últimos años, múltiples estudios sobre el uso de herramientas que permiten generar y ejecutar pruebas automatizadas para evaluar la seguridad de las aplicaciones Web han sido reportadas en la literatura [6]. Estas herramientas permiten conocer las vulnerabilidades que sufren las aplicaciones Web para la protección de los datos que administran. El objetivo de las herramientas es habilitar mecanismos para mejorar la confiabilidad y la seguridad de las aplicaciones que prueban.

El objetivo de esta investigación es identificar y conocer las herramientas que han sido utilizadas para probar de forma automatizada la seguridad de aplicaciones Web. Las herramientas se clasificaron según los tipos de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web [5]. Para lograr el objetivo realizamos un mapeo sistemático de la literatura en el que se analizaron 63 estudios primarios, de los cuales identificamos 66 herramientas utilizadas para realizar pruebas automatizadas de seguridad.

El resto del artículo se estructura de la siguiente manera. La sección 2 menciona las vulnerabilidades de las aplicaciones Web. La sección 3 presenta los trabajos relacionados. La sección 4 explica la metodología del mapeo de literatura. La sección 5 analiza los resultados. La sección 6 presenta las discusiones de este estudio. Finalmente, la sección 7 presenta las conclusiones.

2. Marco Teórico

El proyecto abierto de seguridad de aplicaciones Web [7] provee información sobre la seguridad de las aplicaciones Web y periódicamente actualiza listas de los ataques de seguridad más comunes. Asimismo proporciona documentación técnica sobre estas vulnerabilidades que permite implementar mecanismos de seguridad que prevengan ataques que exploten dichas vulnerabilidades. En el 2017 *OWASP* reportó los 10 riesgos más críticos para las aplicaciones Web [8]

los cuales indicaron que las fallas de inyección (A1:2017), como SQL, NoSQL, OS o LDAP es la principal vulnerabilidad. Los siguientes riesgos fueron la pérdida de autenticación en las funciones de la aplicación relacionadas a autenticación y gestión de sesiones (A2:2017), la exposición de datos sensibles en las aplicaciones Web y APIs que no los protegen adecuadamente (A3:201), las entidades externas XML (XXE) donde los procesadores XML antiguos o mal configurados evalúan referencias a entidades externas en documentos XML (A4:2017), la pérdida de control de acceso y las restricciones sobre lo que los usuarios autenticados pueden hacer (A5:2017), la configuración de seguridad incorrecta (*ad hoc* o por omisión) (A6:2017), la secuencia de comandos en sitios cruzados (XSS) con datos no confiables enviados al navegador Web sin una validación y codificación apropiada (A7:2017), la deserialización insegura cuando una aplicación recibe objetos serializados dañinos que son manipulados o borrados por el atacante (A8:2017), los componentes con vulnerabilidades conocidas que se ejecutan con los mismos privilegios que la aplicación (A9:2017), y finalmente el registro y monitoreo insuficientes con la falta de respuesta ante incidentes (A10:2017).

OWASP mantiene una metodología de pruebas de seguridad para determinar vulnerabilidades de seguridad en aplicaciones Web [5]. Esta metodología en su sección 4 detalla 11 categorías donde para cada una lista el conjunto de vulnerabilidades asociadas que se deben evaluar en las aplicaciones Web: *Information Gathering (4.2)* que detalla 10 tipos de vulnerabilidades, *Configuration and Deployment Management Testing (4.3)* que detalla 9 tipos de vulnerabilidades, *Identity Management Testing (4.4)* que detalla 5 tipos de vulnerabilidades, *Authenticaion Testing (4.5)* que detalla 10 tipos de vulnerabilidades, *Authorization Testing (4.6)* que detalla 4 tipos de vulnerabilidades, *Session Management Testing (4.7)* que detalla 8 tipos de vulnerabilidades, *Input Validation Testing (4.8)* que detalla 17 tipos de vulnerabilidades, *Error Handling (4.9)* que detalla 2 tipos de vulnerabilidades, *Weak Cryptography (4.10)* que detalla 4 tipos de vulnerabilidades, *Business Logic Testing (4.11)* que detalla 9 tipos de vulnerabilidades, *Client Side Testing (4.12)* que detalla 12 tipos de vulnerabilidades. Tal como se muestra en la tabla 1.

Cuadro 1: Clasificación de Vulnerabilidades de seguridad Web por OWASP

Id	Categoría	Cant. TV
4.1	*Web Application Security Testing*	2
4.2	*Information Gathering*	10
4.3	*Configuration and Deployment Management Testing*	9
4.4	*Identity Management Testing*	9
4.5	*Authenticaion Testing*	10
4.6	*Authorization Testing*	4
4.7	*Session Management Testing*	8
4.8	*Input Validation Testing*	17
4.9	*Error Handling*	2
4.10	*Weak Cryptography*	4
4.11	*Business Logic Testing*	9
4.12	*Client Side Testing*	12

3. Trabajo Relacionado

Distintos estudios han discutido la seguridad del software en la literatura. En el 2003, Thompson [9] estudió por qué las pruebas de seguridad son difíciles. El autor planteó que no se puede cubrir el cien por ciento del software y que es importante contar con herramientas para pruebas de seguridad automatizadas que mejoren la eficiencia con respecto a la ejecución de pruebas manuales.

Por su parte en el 2006, Curphey y Arawo [10] listaron un conjunto de herramientas para pruebas de aplicaciones Web e indicaron que no existen herramientas que evalúen todas las vulnerabilidades. Los autores indicaron la importancia de contar con diferentes herramientas para evaluar la seguridad, cada una enfocada en distintas fases del ciclo de vida de desarrollo y para distintos tipos de pruebas tales como pruebas de base de datos, de los *web services*, en *runtime*, de *proxy*, para análisis de código, entre otras.

En el 2010, Pfleeger y Cunningham [11] estudiaron la dificultad de medir la seguridad de las aplicaciones y las estrategias para lograr medir los objetivos de seguridad. Los autores indican que la dificultad es determinar métricas efectivas que permitan una rigurosa y práctica medición de la seguridad.

En el 2015, Rafique et al. [6] identificaron soluciones disponibles para las vulnerabilidades de las aplicaciones Web mediante un mapeo de literatura. Para esto clasificaron las soluciones de acuerdo con la lista de los 10 tipos de vulnerabilidades reportadas por *OWASP* en el 2013. Asimismo identificaron la fase del ciclo de vida de desarrollo en las que propone una solución para las vulnerabilidades.

Mohammed et al. [12] identificaron 54 enfoques de pruebas de seguridad utilizadas en las fases del ciclo de vida de desarrollo del software (SDLC) mediante un mapeo de literatura en el 2017. Estos enfoques se basan principalmente en análisis estático y análisis dinámico de los artefactos de software. En el 2018, Jafari y Rasoolzadegan [13] exploraron estudios que realizan la definición de patrones de seguridad para las aplicaciones de software. Los autores determinaron que la utilización de los patrones de seguridad ha ido creciendo en las nuevas metodologías de desarrollo o la mejora de las existentes. Este estudio complementó estudios secundarios previos sobre patrones de seguridad [14,15].

En nuestro trabajo realizamos una clasificación de las herramientas según los tipos de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web [5] y cuantificamos las vulnerabilidades más reportadas por los estudios identificados. Realizamos un mapeo de las herramientas desde el 2006 hasta el 2019, por lo que es importante destacar que estamos mostrando una lista de herramientas más actualizadas comparada con los estudios previamente mencionados asimismo realizamos un mapeo de estudios de herramientas clasificadas de acuerdo a los tipos de pruebas de vulnerabilidad de seguridad de las aplicaciones web realizadas por cada herramienta.

4. Metodología

El mapeo sistemático de la literatura se realizó siguiendo los lineamientos establecidos por Petersen, Vakkalanka y Kuzniarz [16] y las recomendaciones de Kitchenham y Charters [17].

El objetivo de este estudio, formulado con el modelo *Goal Question Metric (GQM)* [18] fue *analizar* las herramientas utilizadas para realizar pruebas automatizadas de seguridad *con el propósito de* caracterizarlas *con respecto a* las vulnerabilidades de seguridad que prueban *desde el punto de vista de* los investigadores *en el contexto* de aplicaciones Web. Para guiar este estudio, se definió la siguiente pregunta de investigación: **(RQ)** ¿Cuáles herramientas se han utilizado para automatizar las pruebas de seguridad de aplicaciones Web?

4.1. Estrategia de búsqueda y proceso de selección de estudios

Primero se realizó una búsqueda exploratoria para identificar estudios relevantes, que fueron usados como artículos de control. Los artículos de control seleccionados [S01, S02, S03] fueron usados como base en la definición y validación del protocolo de investigación. Estos artículos presentan herramientas para la detección de vulnerabilidades en aplicaciones Web y se seleccionaron por contar con la información necesaria para responder la pregunta de investigación.

A partir del objetivo planteado, las preguntas de investigación, y los términos clave extraídos del título y del resumen de los artículos de control, se construyó la versión inicial de la cadena de búsqueda. Para esto se utilizó el modelo PICO (Población, Intervención, Comparación, Salidas) en el proceso de construcción. La cadena final, que se muestra a continuación, fue producto de un proceso de refinamiento que incluyó varias pruebas piloto para reducir el ruido.

(“automat*” OR “tool”) AND (“secur* test*”) AND (“web*”)

Las búsquedas automatizadas se realizaron en las bases de datos *SCOPUS*, *IEEE Xplore*, y *Web of Science*. Se realizó la búsqueda en el título, el resumen y las palabras clave de los artículos. El protocolo del mapeo se desarrolló de Marzo a Mayo del 2019, y la búsqueda automatizada se realizó en Junio del 2019. Los estudios se analizaron entre Junio y Diciembre del 2019.

El número de estudios recuperado para cada base de datos fue de 159 artículos en *SCOPUS*, 81 artículos en *IEEE Xplore*, y 20 artículos en *Web of Science*.

En total se obtuvieron 260 artículos de los cuales 75 eran duplicados, tal como se presenta en la Figura 1.

Los artículos fueron tabulados en MS Excel para los procesos de selección, evaluación y extracción de datos. Se eliminaron los duplicados, se aplicaron los criterios de inclusión y exclusión (I/E) y finalmente se hizo la extracción y el análisis de los resultados. Tras la eliminación de duplicados, se obtuvo un conjunto de 185 estudios.

El proceso de I/E se hizo con base en el título y el resumen de los artículos (cuando hubo duda, se hizo lectura completa del artículo). Se excluyeron publicaciones donde no se encontraba disponible el texto completo (E1) y los artículos

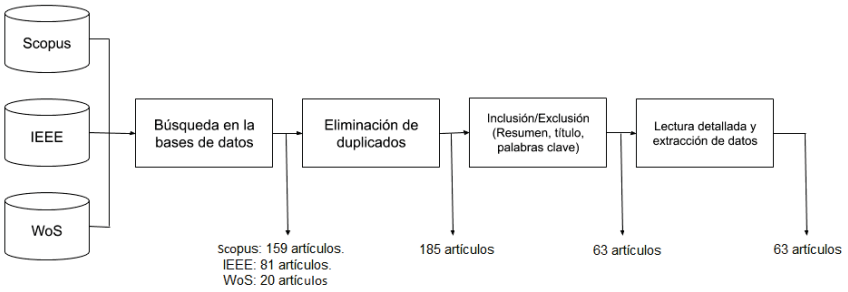


Figura 1: Proceso de búsqueda y selección de artículos

secundarios y terciarios (E2). Se incluyeron publicaciones de artículos que trataran sobre herramientas utilizadas para pruebas de seguridad en aplicaciones Web (I1) y solo los artículos en idioma inglés (I2).

A partir de la estrategia de búsqueda y el proceso de selección, se obtuvo un total de 63 artículos (Figura 1).

La lista completa de los estudios relevantes puede consultarse en el enlace: `https://tinyurl.com/w9e48g3`. En este enlace se encuentran los detalles de cada herramienta (nombre de la herramienta, nombre del artículo, autores, lenguaje de programación, entre otros), el grado de automatización de sus pruebas (automatizadas, semi automatizadas o manuales) y tipos de pruebas que realiza cada herramienta. Asimismo en esta lista se encuentra los detalles de la evaluación de calidad, el proceso de extracción que describimos a continuación.

4.2. Evaluación de calidad

La evaluación de la calidad de los estudios se realizó para determinar el nivel de detalle ofrecido sobre los aspectos de interés del análisis y basado en la pregunta de investigación. Estos permiten obtener un *ranking* de la completitud de las publicaciones para responder la pregunta de investigación. Los criterios de calidad establecidos para la evaluación de los artículos fueron los siguientes: (Q1) El artículo describe la construcción y la funcionalidad de la herramienta para realizar las pruebas de seguridad. (Q2) El artículo describe cómo se ejecutan las pruebas automatizadas de seguridad utilizando la herramienta. (Q3) El artículo detalla el procedimiento de evaluación de la efectividad de la herramienta utilizada y sus resultados. El puntaje se otorga en una escala de 0 a 2, donde 0 = No en lo absoluto, 1 = Parcialmente y 2= Totalmente.

Los valores de calidad obtenidos por los estudios variaron entre 0 y 6, con una mediana de 4 y un promedio de 3.70, lo que refleja que los estudios tienen un nivel de detalle aceptable.

4.3. Extracción y análisis de los datos

Para la extracción de la información de los 63 artículos se elaboró una tabla con los elementos que permitían responder la pregunta de investigación. Para la pregunta de investigación (RQ) se extrajo la herramienta utilizada y su

descripción, el contexto donde se ejecuta la herramienta, la información de las pruebas que realiza para detectar las vulnerabilidades de las aplicaciones Web. Una vez que se contó con la información tabulada se procedió con el análisis de las frecuencias de vulnerabilidades por herramientas y la identificación de las evaluaciones de efectividad de las herramientas.

4.4. Amenazas a la validez

A continuación se discuten algunas de las limitaciones del estudio. La cadena de búsqueda fue definida a partir de una búsqueda exploratoria en bases de datos y un conjunto de artículos de control. Además fue refinada mediante un conjunto de pruebas piloto. Las bases de datos seleccionadas son reconocidas por tener una buena cobertura de información en el campo de ingeniería de software. Durante el proceso de inclusión o exclusión, si existían dudas sobre un artículo específico, se procedió a su lectura completa.

Las clasificaciones presentadas en este estudio, así como la interpretación de los resultados, se realizaron usando el criterio de la investigadora principal y con la validación de los demás investigadores. Además, la aplicación de los criterios de calidad fue realizada por una sola investigadora. Para la clasificación de las herramientas, se utilizó la metodología de pruebas para aplicaciones Web de *OWASP*. En la mayoría de los casos la clasificación fue extraída de manera explícita de los artículos; sin embargo, para algunos casos la selección de la vulnerabilidad se realizó de manera implícita a partir de los datos reportados. Todo el proceso se reportó de forma detallada para facilitar su análisis y utilización en estudios posteriores.

5. Resultados

A continuación se presentan los resultados del mapeo sistemático de la literatura que identifican las herramientas utilizadas para la automatización de pruebas de seguridad de aplicaciones Web.

En total identificamos 66 propuestas de herramientas utilizadas para realizar pruebas automatizadas de seguridad. Las herramientas se clasificaron según las 11 categorías de la Sección 4 de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web de *OWASP* (basado en el primer nivel de la clasificación) [5].

La Tabla 2 agrupa las herramientas para cada una de las categorías de vulnerabilidades del primer nivel de la clasificación de *OWASP*. Las categorías para las cuales se identificaron herramientas fueron: *Configuration and Deployment Management Testing (4.3)*, *Identity Management Testing (4.4)*, *Authenticaion Testing (4.5)*, *Authorization Testing (4.6)*, *Session Management Testing (4.7)*, *Input Validation Testing (4.8)*, *Error Handling (4.9)*, *Weak Cryptography (4.10)* y *Client Side Testing (4.12)*.

Para cada categoría (Id) se presenta la lista de herramientas, referencias de los artículos que la evaluaron, y la cantidad de herramientas (Q). Los resultados indican que la categoría de pruebas para detectar vulnerabilidades más común fue *Input Validation Testing (4.8)* con 55 herramientas, seguido de las pruebas de *Configuration and Deployment Management Testing (4.3)*, *Session Management*

Cuadro 2: Herramientas por categoría de *OWASP* (primer nivel)

Id	Herramientas y referencias	Q
4.3	ISTA [S48], AppScan [S34], Urls black box tests on the Web pages [S32], Fuzz testing tool [S65], HJ2IF [S28], IPT-WS [S54], MobSTer [S04], SAML-based SSO IdP by Google [S64], SECEVAL [S46], WS-Attacker [S33], WSFAgressor [S61], WSInject [S59], WSSecTool [S66], WSAttaker [S57], WSVTS [S58]	15
4.4	SPaCiTE [S12, S63], AppScan [S34]	2
4.5	ZAP [S07, S37], DAST [S37], Fortify [S10], IMAATT [S45], JBroFuzz [S10], OAuthTester [S55], Paros [S10], PBST [S36], Selenium IDE [S20], WebScarab [S10]	10
4.6	ISTA [S05], OAuthTester [S55], SPaCIoS [S19]	3
4.7	ZAP [S07], ISTA [S05], CodePulse [S26], Magento [S14], MobsTer [S04], PBST [S36], SAMATE [S16], Selenium IDE [S20], AOP [S39], Urls black box tests [S32], Deemon [S38], Fortify [S10], JBroFuzz [S10], Paros [S10], WebScarab [S10]	15
4.8	ZAP [S07, S37, S42], ISTA [S05], SPaCiTE [S12, S63], Volcano [S49, S52], AOP [S39], Sign-WS [S54], ATUSA [S51], BIOFUZZ [S43], BurpTool [S42], CRAXweb [S17], Circe [S21], CodePulse [S26], DAST [S37], Deemon [S38], XSS, black box and SQLI injection point [S30], Eclipse IDE [S62], Eclipse IDE test cases [S50], Urls black box tests on the Web pages [S32], Fortify [S10], IAAT [S22], IMATT [S45], JBroFuzz [S10], JWebUnit [S23], JWAST [S40], KamaleonFuzz [S47], Magento [S14], MobSTer [S04], MBT [S41], Noncespace [S06], PHP2XMI [S08], PURITY [S09], Paros [S10], RBVT [S44], RAD-WS [S54], SAFELI [S35], SAMATE [S16], SAML [S64], SAP HANA XS Applications [S56], SECEVAL [S46], SSES [S15], SPaCIoS [S19], SQLIVDT [S18], SQLMAP [S13], Selenium IDE [S20], IDS [S53], Signature evaluation [S24], Tamper data [S60], Tool-prototype [S11], WAP [S35], WSFAgressor [S61], WSInject [S59], WSVTS [S58], WebScarab [S10], XSSINJECTOR [S27], XSS and SQLI [S25]	55
4.9	ZAP [S07]	1
4.10	ISTA [S05], Fortify [S10], JBroFuzz [S10], Paros [S10], SAML-based SSO IdP service provided by Google [S64], WebScarab [S10]	6
4.12	ZAP [S37], ISTA [S52], AOP [S39], ATUSA [S51], Code Pulse [S26], DAST [S37], Fortify [S10], JBroFuzz [S10], JWAST [S40], KITE [S31], Paros [S10], SAP HANA XS Applications [S56], SQTL [S29], WebScarab [S10], XSSINJECTOR [S27]	15

Testing (4.7) y *Client Side Testing (4.12)* con 15 herramientas utilizadas cada una.

Para la Tabla 2 se cuantificó la cantidad de artículos que trabajaron cada categoría de vulnerabilidad y el total de ocurrencias de las vulnerabilidades de una categoría en los artículos. Los resultados muestran una tendencia similar a las de las herramientas donde la categoría de pruebas para detectar vulnerabilidades con más artículos fue la *(4.8)* con 52 artículos y 113 vulnerabilidades evaluadas. En el caso de las pruebas de *(4.3)* la cantidad de artículos y vulnerabilidades evaluadas fue 15. Finalmente, las categorías *(4.7) y (4.12)* fueron reportadas en 12 artículos y en el caso de la categoría *(4.7)* evaluó 12 vulnerabilidades y en la categoría *(4.12)* evaluó 17 vulnerabilidades.

Las herramientas *ZAP* [S58, S37, S42], *ISTA* [S05, S48], *SPaciTE* [S12, S63] y *Volcano* [S49, S52] fueron las únicas reportados por dos o más artículos. Esto indica la necesidad de que los investigadores realicen replicaciones utilizando propuestas de herramientas existentes.

En la Figura 2 se presenta la tendencia de vulnerabilidades evaluadas por categoría de *OWASP* del primer nivel por cada año desde el 2006 hasta el 2019. Los resultados confirman que la categoría de vulnerabilidad *Input Validation Testing (4.8)* es la más evaluada por las herramientas entre los años 2007 al 2019, siendo el año 2014 y 2018 el más evaluado con 21 y 17 ocurrencias respectivamente.

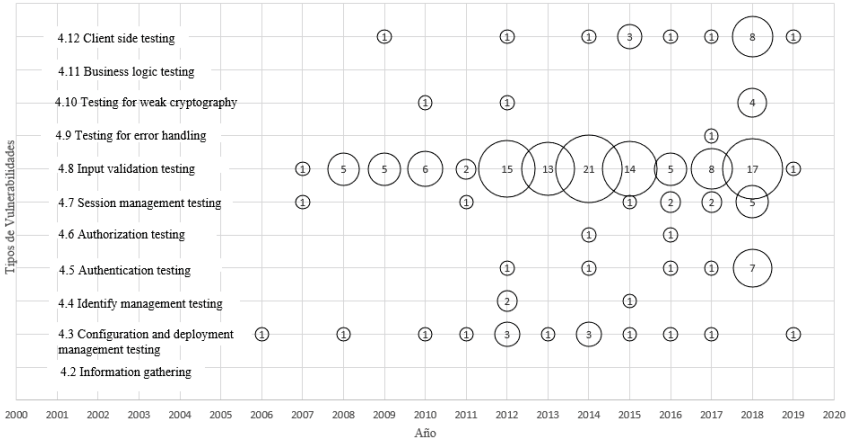


Figura 2: Tendencia de vulnerabilidades del primer nivel de *OWASP* por año

La Tabla 3 presenta las vulnerabilidades del segundo nivel de *OWASP* para las categorías en que se identificaron herramientas. Las vulnerabilidades para las cuales se identificaron herramientas fueron: *Testing for Privilege Escalation (4.6.3)*, *Testing for Insecure Direct Object References (4.6.4)*, *Testing for Cookies attributes (4.7.2)*, *Testing for Cross Site Request Forgery (CSRF) (4.7.5)*, *Testing for Reflected Cross Site Scripting (4.8.1)*, *Testing for Stored Cross Site Scripting (4.8.2)*, *Testing for SQL Injection (4.8.5)*, *Testing for LDAP Injection (4.8.6) Testing for XML Injection (4.8.8) Testing for XML injection (4.8.9)*

Testing for XPath Injection (4.8.10) Testing for Command Injection (4.8.13) Testing for HTTP Incoming Requests (4.8.17) Testing for DOM based Cross Site Scripting (4.12.1) Testing for JavaScript Execution (4.12.2) Testing for HTML Injection (4.12.3).

Para cada vulnerabilidad (Id) se presenta la lista de herramientas y referencias de los artículos que la evaluaron, y la cantidad de herramientas (Q). Los tipos de pruebas más reportadas fueron los de la categoría *Input Validation Testing (4.8)* en la que se encontraron las vulnerabilidades más reportadas. Las vulnerabilidades más comúnmente evaluadas fueron la *SQL Injection (4.8.5)* con 40 herramientas, *Cross-Site Scripting (4.8.2)* con 30 herramientas, y *Testing for HTTP Incoming Requests (4.8.17)* con 19 herramientas utilizadas.

En la Figura 3 se presenta la tendencia de vulnerabilidades evaluadas del segundo nivel de *OWASP* por cada año desde el 2007 hasta el 2019. Los resultados confirman que las vulnerabilidades *Testing for SQL Injection (4.8.5)* y *Cross-Site Scripting (4.8.2)* son las más evaluadas por las herramientas.

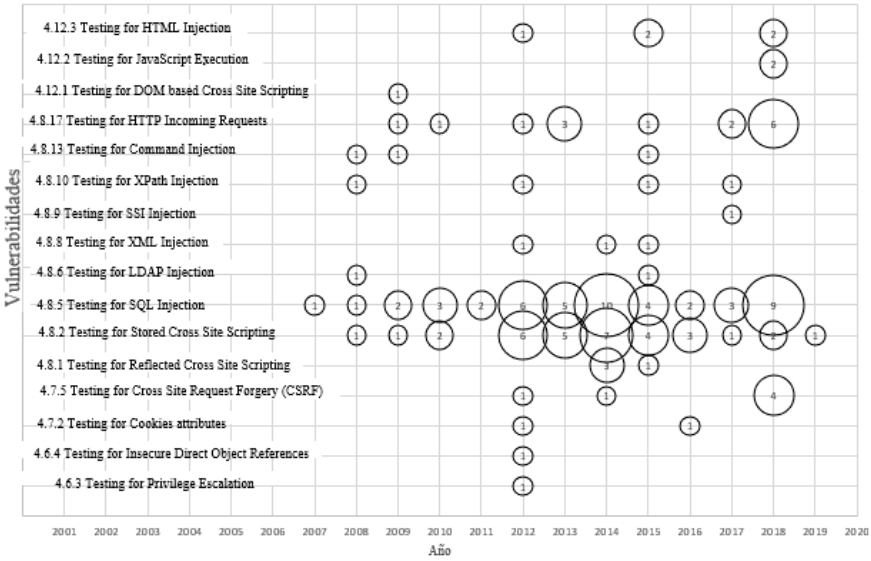


Figura 3: Tendencia de vulnerabilidades del segundo nivel de *OWASP* por año

6. Discusión

La clasificación de las distintas herramientas disponibles para evaluar cada una de las vulnerabilidades de las aplicaciones Web puede apoyar a los profesionales en la selección de potenciales herramientas para sus procesos de pruebas de seguridad. Estas herramientas pueden apoyar la evaluación de la seguridad de las aplicaciones Web de forma automatizada en distintas fases del ciclo de vida de desarrollo de las aplicaciones Web o inclusive cuando se encuentra en producción.

Cuadro 3: Herramientas por vulnerabilidad de *OWASP* (segundo nivel)

Id	Herramientas y referencias	Q
4.6.3	ISTA [S48]	1
4.6.4	ISTA [S48]	1
4.7.2	AOP [S39], Urls and performing black box tests on the Web pages [S32]	2
4.7.5	ISTA [S05], Deemon [S38], Fortify [S10], JBroFuzz [S10], Paros [S10], WebScarab [S10]	6
4.8.1	ZAP [S07], BurpTool [S42], MBT [S41], SPaCIoS [S19]	4
4.8.2	ZAP [S07, S37, S42], ISTA [S05], SPaCiTE [S12, S63], AOP [S39], BurpTool [S42], CRAXweb [S17], Circe [S21], CodePulse [S26], DAST [S37], XSS, black box and SQLI based on injection point [S30], IAAT [S22], IMAATT [S45], JwebUnit [S10], JWAST [S40], KamaleonFuzz [S47], MobSTer [S04], MBT [S04], Noncespaces [S06], PHP2XMI [S08], PURITY [S09], RBVT [S44], SSES [S15], SPaCIoS [S19], SQLIVDT [S18], Selenium IDE [S20], Tamper data [S60], Tool-prototye [S11], WSInject [S59], XSSINJECTOR [S27], XSS and SQLI evaluation [S25]	30
4.8.5	ZAP [S07,S37], ISTA [S05], SPaCiTE [S12, S63], Volcano [S49, S52], AOP [S39], BIOFUZZ [S43], CRAXweb [S17], Circe [S21], CodePulse [S26], DAST [S37], XSS, black box and SQLI based on injection point [S30], Eclipse IDE [S62], Fortify [S10], IAAT [S22], IMAATT [S45], JBroFuzz [S10], JWAST [S40], Magento [S14], MobSTer [S04], PHP2XMI [S08], PURITY [S09], Paros [S10], RBVT [S44], RAD-WS [S54], Eclipse IDE test cases execution [S50], SAFELI [S35], SAMATE [S16], SAP HANA XS Applications [S56], SSES [S15], SPaCIoS [S19], SQLIVDT [S18], SQLMap [S13], Selenium IDE [S20], IDS [S53], Signature evaluation [S24], Tamper data [S60], Tool-prototype [S11], WAP [S35], WebScarab [S10], XSS and SQLI security evaluation [S25]	40
4.8.6	JWAST [S40], SSES [S15]	2
4.8.8	Eclipse IDE [S62], JWAST [S40], SPaCioS [S19]	3
4.8.9	Sign-WS [S54]	1
4.8.10	Eclipse IDE [S62], JWAST [S40], RAD-WS [S54], SSES [S15]	4
4.8.13	JWAST [S40], SSES [S15]	2
4.8.17	ZAP [S07, S37, S42], ATUSa [S51], BurpTool [S42], CRAXweb [S17], DAST [S37], Deemon [S38], XSS, black box and SQLI based on injection point [S30], Urls and performing black box tests on the Web pages [S32], Fortify [S10], JBroFuzz [S10], JWAST [S40], Paros [S10], SAML-based SSO IdP service by Google [S64], SECEVAL [S46], SPaCIoS [S19], Tamper data [S60], WSFAgressor [S61], WSVTS [S58], WebScarab [S10]	19
4.12.1	ATUSA [S51]	1
4.12.2	ZAP [S37], DAST [S37], JWAST [S40], SAP HANA XS Applications [S56]	4
4.12.3	ZAP [S37], ISTA [S05], DAST [S37]	3

Se identificó gran variedad de herramientas que se utilizan para automatizar distintas pruebas de seguridad de acuerdo con las distintas vulnerabilidades que existen en la actualidad. Para evaluar la seguridad de las aplicaciones Web, las herramientas deben de contar con pruebas actualizadas de acuerdo con las vulnerabilidades y ataques cibernéticos actuales. La identificación de herramientas demostró que se mantiene la tendencia de realizar pruebas para la *SQL Injection* y *Cross-Site Scripting*. Sin embargo, no se encontraron herramientas para algunos de los escenarios reportados como los 10 ataques con más riesgosos y frecuentes por *OWASP*, tales como: *Insecure Deserialization, Using Components with known vulnerabilities* e *Insufficient Logging & Monitoring*.

Por otro lado, de las 66 propuestas de herramientas utilizadas para realizar pruebas automatizadas de seguridad solo 41 reportaron una evaluación de la efectividad. De las 41 herramientas, 16 fueron evaluadas a partir de criterios de eficiencia, 13 a partir de criterios de eficacia y 12 fueron evaluadas a partir de ambos criterios. Las herramientas fueron evaluadas para determinar el tiempo que toma la ejecución de las pruebas (por ejemplo: [S07, S13, S34, S54, S61]), comparar la efectividad entre herramientas (por ejemplo: [S04, S30, S58]), y determinar la relación entre eficiencia y efectividad (cantidad de pruebas ejecutadas y tiempo que tomó la ejecución de las pruebas, por ejemplo: [S11, S41, S24, S30, S32, S38, S43, S51]). Por ejemplo, las herramientas [S35, S36, S47, S54, S59, S61, S04, S51] evaluaron la cantidad de falsos positivos en los resultados de ejecución de las pruebas y la herramienta [S10] comparó los resultados obtenidos al ejecutar pruebas automatizadas contra la ejecución de pruebas manual para comparar los resultados obtenidos. En muchas de las evaluaciones se validó el éxito de la ejecución de pruebas a partir de la cantidad de vulnerabilidades encontradas (por ejemplo: [S09, S18, S19, S20, S21, S23, S25, S31, S10, S30, S32, S43, S58, S42]). Otras evaluaciones se limitaron a verificar la funcionalidad implementada por la herramienta (por ejemplo: [S22, S05, S24, S38, S51, S49]).

Del total de herramientas, 29 herramientas reportaron algunos de los retos que enfrentaron durante la ejecución de las pruebas. Diez herramientas plantean la necesidad de actualizar y cambiar sus tipos de pruebas para mejorar el desempeño, la efectividad y la escalabilidad [S05, S18, S41, S47, S52, S58, S59, S60, S62]. Ocho herramientas plantean la necesidad de crear nuevos casos de pruebas para detectar otros tipos de vulnerabilidades [S04, S08, S13, S14, S23, S30, S36, S51, S57]. Una propuesta plantea la necesidad de mejorar la configuración de la herramienta [S09], otra la mejora en la priorización de las pruebas para utilizarlas en pruebas de regresión [S20], cuatro indican la necesidad de optimizar el código, los métodos y las funcionalidades de la herramienta [S34, S38, S45, S46], así como también la mejora en la detección de los falsos positivos en los resultados [S37]. Finalmente, 3 herramientas plantean la necesidad de mejorar el soporte para distintas [S11, S26, S33].

7. Conclusiones

Este estudio reportó los resultados de un mapeo sistemático de la literatura sobre herramientas utilizadas para probar la seguridad de aplicaciones Web. Se identificaron 63 estudios primarios que reportaron 66 propuestas de herramientas

utilizadas para realizar pruebas automatizadas de seguridad. Las herramientas se clasificaron según los tipos de la metodología de pruebas de seguridad para determinar vulnerabilidades del proyecto abierto de seguridad en aplicaciones Web.

Los resultados demuestran que se mantiene la tendencia de realizar pruebas para la *SQL Injection* y *Cross-Site Scripting*. Sin embargo, los casos de prueba que realizan las herramientas identificadas solo evalúan algunos de los escenarios reportados como los 10 ataques más riesgosos y frecuentes por *OWASP*. Además, aunque se identificó gran variedad de herramientas que realizan distintos tipos de pruebas de seguridad, solo pocas herramientas se encuentran en la lista de herramientas recomendadas por *OWASP*, lo que denota la necesidad de contar con evaluaciones empíricas de herramientas existentes en el área.

Como trabajo futuro, se desea seleccionar un conjunto de herramientas de pruebas de seguridad que permitan verificar las principales vulnerabilidades reportadas para las aplicaciones Web y evaluar su efectividad con el fin de generar evidencia para la industria. Asimismo, como trabajo futuro se desea identificar las clasificaciones de vulnerabilidades brindadas por *OWASP* que no fueron cubiertas por ninguna herramienta del mapeo y realizar un estudio de la razón por la cual las actuales herramientas no evalúan estas vulnerabilidades y la importancia de que las herramientas ejecute pruebas para estas vulnerabilidades.

Referencias

1. P. Zech, M. Felderer, and R. Breu, "Knowledge-based security testing of web applications by logic programming," *International Journal on Software Tools for Technology Transfer*, vol. 21, no. 2, pp. 221–246, 2019.
2. OWASP, "Testing: Introduction and objectives," *The Open Web Application Security Project*, 2019.
3. S. De Vries, "Security testing web applications throughout automated software tests," *Corsaire Ltd*, vol. 3, 2006.
4. Y. Stefinko, A. Piskozub, and R. Banakh, "Manual and automated penetration testing. benefits and drawbacks. modern tendency," in *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*. IEEE, 2016, pp. 488–491.
5. OWASP, "Testing guide v4, web application security testing," *The Open Web Application Security Project*, 2019.
6. S. Rafique, M. Humayun, B. Hamid, A. Abbas, M. Akhtar, and K. Iqbal, "Web application security vulnerabilities detection approaches: A systematic mapping study," in *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE, 2015, pp. 1–6.
7. OWASP, "The open web application security project," *The Open Web Application Security Project*, 2019.
8. ——, "Top 10–2017–the ten most critical web application security risks," *The Open Web Application Security Project*, 2017.
9. H. H. Thompson, "Why security testing is hard," *IEEE Security Privacy*, vol. 1, no. 4, pp. 83–86, July 2003.
10. M. Curphey and R. Arawo, "Web application security assessment tools," *IEEE Security & Privacy*, vol. 4, no. 4, pp. 32–41, 2006.

11. S. Pfleeger and R. Cunningham, "Why measuring security is hard," *IEEE Security Privacy*, vol. 8, no. 4, pp. 46–54, July 2010.
12. N. M. Mohammed, M. Niazi, M. Alshayeb, and S. Mahmood, "Exploring software security approaches in software development lifecycle: A systematic mapping study," *Computer Standards & Interfaces*, vol. 50, pp. 107–115, 2017.
13. A. J. Jafari and A. Rasoolzadegan, "Security patterns: A systematic mapping study," *arXiv preprint arXiv:1811.12715*, 2018.
14. M. Bunke, "Software-security patterns: degree of maturity," in *Proceedings of the 20th European Conference on Pattern Languages of Programs*. ACM, 2015, p. 42.
15. Y. Ito, H. Washizaki, M. Yoshizawa, Y. Fukazawa, T. Okubo, H. Kaiya, A. Hazeyama, N. Yoshioka, and E. B. Fernandez, "Systematic mapping of security patterns research," in *Proceedings of the 22nd Conference on Pattern Languages of Programs*. The Hillside Group, 2015, p. 14.
16. K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," vol. 64. Elsevier, 2015, pp. 1–18.
17. B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *Technical report, EBSE Technical Report EBSE-2007-01*, pp. 1–57, 2007.
18. V. Basili, C. Gianluigi, and D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

Bibliografía del capítulo

[1] P. Zech, M. Felderer, and R. Breu, "Knowledge-based security testing of web applications by logic programming," *International Journal on Software Tools for Technology Transfer*, vol. 21, no. 2, pp. 221–246, 2019.

[2] OWASP, "Testing: Introduction and objectives," *The Open Web Application Security Project*, 2019.

[3] S. De Vries, "Security testing web applications throughout automated software tests," *Corsaire Ltd*, vol. 3, 2006.

[4] Y. Stefinko, A. Piskozub, and R. Banakh, "Manual and automated penetration testing. benefits and drawbacks. modern tendency," in *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*, pp. 488–491, IEEE, 2016.

[5] S. Rafique, M. Humayun, B. Hamid, A. Abbas, M. Akhtar, and K. Iqbal, "Web application security vulnerabilities detection approaches: A systematic mapping study," in *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 1–6, June 2015.

[6] OWASP, "Testing guide v4, web application security testing," *The Open Web Application Security Project*, 2019.

[7] F. Ureña, "Ciberataques, la mayor amenaza actual," 2015.

[8] R. Guaman-Quinche and H. Torres-Carrion, "Seguridad en aplicaciones web para sistemas de gestión académica," *Revista Tecnológica ESPOL – RTE*, vol. 28, pp. 508–519, 01 2016.

[9] "Sobre owasp," https://www.owasp.org/index.php/Sobre_OWASP, 11 2014.

[10] "Owasp top 10 -2017," https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf, 2017.

[11] "Commom weakness enumeration. cwe-352: Cross-site request forgery (csrf)," https://cwe.mitre.org/data/definitions/352.html, 12 2018.

[12] A. Domínguez, "Cross-site scripting (xss)," https://www.seguridad.unam.mx/img/XSS_rev.pdf.

[13] G. Gomez, "Herramientas de prueba de seguridad de aplicaciones," 2017.

[14] "Automated testing vs manual testing: Which should you use, and when?," *API-CA*, 11 2014.

[15] O. Cádenas, "Automatización de casos de prueba para mejorar el proceso de calidad de software.," 2016.

[16] "Automation testing vs. manual testing: What's the difference?," *Guru99*.

[17] S. Pfleeger and R. Cunningham, "Why measuring security is hard," *IEEE Security Privacy*, vol. 8, pp. 46–54, July 2010.

[18] H. Thompson, "Why security testing is hard," *IEEE Security Privacy*, vol. 1, pp. 83–86, July 2003.

[19] N. Mohammed, N. Mahmood, A. Mohammad, and M. Sajjad, "Exploring software security approaches in software development lifecycle: A systematic mapping study," *Computer Standards & Interfaces*, vol. 50, pp. 107 – 115, 2017.

[20] M. Curphey and R. Arawo, "Web application security assessment tools," *IEEE Security Privacy*, vol. 4, pp. 32–41, July 2006.

[21] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[22] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3," *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.

[23] V. Basili, G. Caldiera, and D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

[24] S. Jan, A. Panichella, A. Arcuri, and L. Briand, "Automatic generation of tests to exploit xml injection vulnerabilities in web applications," *IEEE Transactions on Software Engineering*, vol. 45, no. 4, pp. 335–362, 2019.

[25] S. Dashevskyi, D. Dos Santos, F. Massacci, and A. Sabetta, "Testrex: a framework for repeatable exploits," *International Journal on Software Tools for Technology Transfer*, vol. 21, no. 1, pp. 105–119, 2019.

[26] D. Appelt, C. Nguyen, A. Panichella, and L. Briand, "A machine-learning-driven evolutionary approach for testing web application firewalls," *IEEE Transactions on Reliability*, vol. 67, no. 3, pp. 733–757, 2018.

[27] B. Kitchenham, E. Mendes, and G. Travassos, *A Systematic Review of Cross- vs. Within-Company Cost Estimation Studies*. IEEE Trans on SE, 2007.

[28] P. Pathirathna, V. Ayesha, W. Imihira, W. Wasala, N. Kodagoda, and E. Edirisinghe, "Security testing as a service with docker containerization," in *International Conference on Software, Knowledge Information, Industrial Management and Applications, SKIMA*, vol. 2017-December, 2018.

[29] J. Boutahar, I. Maskani, and S. El Houssaïni, "Experimental evaluation of security requirements engineering benefits," in *nternational Journal of Advanced Computer Science and Applications*, 2018. Vol. 9, No. 11, 2018.

[30] B. Garn, I. Kapsalis, D. Simos, and S. Winkler, "On the applicability of combinatorial testing toweb application security testing: A case study," 2014.

[31] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and Xu.W., "Automated security test generation with formal threat models," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 4, pp. 526–540, 2012.

[32] D. Xu, "A tool for automated test code generation from high-level petri nets," in *Applications and Theory of Petri Nets* (L. M. Kristensen and L. Petrucci, eds.), (Berlin, Heidelberg), pp. 308–317, Springer Berlin Heidelberg, 2011.

[33] M. Büchler, J. Oudinet, and A. Pretschner, "Spacite - web application testing engine," in *Proceedings - IEEE 5th International Conference on Software Testing, Verification and Validation, ICST 2012*, pp. 858–859, 2012. Cited By :8.

[34] M. Büchler, J. Oudinet, and A. Pretschner, "Spacite - web application testing engine," 04 2012.

[35] T. Dao and E. Shibayama, "Security sensitive data flow coverage criterion for automatic security testing of web applications," in *Engineering Secure Software and Systems* (Ú. Erlingsson, R. Wieringa, and N. Zannone, eds.), (Berlin, Heidelberg), pp. 101–113, Springer Berlin Heidelberg, 2011.

[36] T. Dao and E. Shibayama, "Idea: Automatic security testing for web applications," in *Engineering Secure Software and Systems* (F. Massacci, S. T. Redwine, and N. Zannone, eds.), (Berlin, Heidelberg), pp. 180–184, Springer Berlin Heidelberg, 2009.

[37] M. Van Gundy and H. Chen, "Noncespaces: Using randomization to defeat cross-site scripting attacks," *Computers and Security*, vol. 31, no. 4, pp. 612–628, 2012.

[38] A. Avancini and M. Ceccato, "Grammar based oracle for security testing of web applications," in *2012 7th International Workshop on Automation of Software Test, AST 2012 - Proceedings*, pp. 15–21, 2012. Cited By :5.

[39] D. Xu, W. Xu, B. Bavikati, and W. Wong, "Mining executable specifications of web applications from selenium ide tests," in *2012 IEEE Sixth International Conference on Software Security and Reliability*, pp. 263–272, June 2012.

[40] S. Choudhary, M. Dincturk, G. Bochmann, G. Jourdan, I. Onut, and P. Ionescu, "Solving some modeling challenges when testing rich internet applications for security," in *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*, pp. 850–857, April 2012.

[41] C. Mainka, J. Somorovsky, and J. Schwenk, "Penetration testing tool for web services security," in *2012 IEEE Eighth World Congress on Services*, pp. 163–170, June 2012.

[42] P. Patil and S. Pawar, "Remote agent based automated framework for threat modelling, vulnerability testing of soa solutions and web services," in *World Congress on Internet Security (WorldCIS-2012)*, pp. 127–131, June 2012.

[43] W. Fu, Y. Zhang, X. Zhu, and J. Qian, "Wssectool: A web service security analysis tool based on program slicing," in *2012 IEEE Eighth World Congress on Services*, pp. 179–183, June 2012.

[44] M. Buchler, K. Hossen, P. Mihancea, M. Minea, R. Groz, and C. Oriat, "Model inference and security testing in the spacios project," in *2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering, CSMR-WCRE 2014 - Proceedings*, pp. 411–414, 2014. Cited By :5.

[45] A. Aydin, M. Alkhalaf, and T. Bultan, "Automated test generation from vulnerability signatures," in *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation*, pp. 193–202, March 2014.

[46] R. Hamad and M. Al Fayoumi, "Scalable quality and testing lab (sqtl): Mission-critical applications testing," in *2019 International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–7, April 2019.

[47] J. Thomé, A. Gorla, and A. Zeller, "Search-based security testing of web applications," 2014.

[48] J. Botella, B. Legeard, F. Peureux, and A. Vernotte, "Risk-based vulnerability testing using security test patterns," in *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications* (T. Margaria

and B. Steffen, eds.), (Berlin, Heidelberg), pp. 337–352, Springer Berlin Heidelberg, 2014.

[49] E. Fathy, M. Zaki, M. Ahmed, and H. Tahani, "Imatt: An integrated multi-agent testing tool for the security of agent-based web applications," 2013.

[50] M. Busch, N. Koch, and M. Wirsing, "Seceval: An evaluation framework for engineering secure systems," 2014.

[51] F. Duchene, S. Rawat, J. Richier, and R. Groz, "Kameleonfuzz:evolutionary fuzzing for black-box xss detection," 2014.

[52] J. Chen, H. Wang, D. Towey, C. Mao, R. Huang, and Y. Zhan, "Worst-input mutation approach to web services vulnerability testing based on soap messages," *Tsinghua Science and Technology*, vol. 19, pp. 429–441, Oct 2014.

[53] M. Palma and E. Martins, "Security testing methodology for vulnerabilities detection of xss in web services and ws-security," *Electronic Notes in Theoretical Computer Science*, vol. 302, p. 133–154, 02 2014.

[54] M. Najla'a Ateeq, S. Abu, G. Abdul, and Z. Hazura, "Comparison of security testing approaches for detection of sql injection vulnerabilities," in *International Journal of Engineering & Technology*, 2018.

[55] L. Dukes, X. Yuan, and F. Akowuah, "A case study on web application security testing with tools and manual testing," in *Conference Proceedings - IEEE SOUTHEASTCON*, 2013. Cited By :14.

[56] P. Araujo and A. Paiva, "Pattern based web security testing," 2018.

[57] H. Shahriar and W. Bond, "Towards an attack signature generation framework for intrusion detection systems," in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pp. 597–603, Nov 2017.

[58] "Free for open source application security tools." https://owasp.org/www-community/Free_for_Open_Source_Application_Security_Tools.

[59] "Vulnerability scanning tools." https://owasp.org/www-community/Vulnerability_Scanning_Tools.

[60] "Source code analysis tools." https://owasp.org/www-community/Source_Code_Analysis_Tools.

[61] I. Ben-Bassat and E. Rokah, "Locality-sensitive hashing for efficientweb application security testing," in *ICISSP 2019 - Proceedings of the 5th International Conference on Information Systems Security and Privacy*, pp. 193–204, 2019.

[62] Y. Hsu, G. Shu, and D. Lee, "A model-based approach to security flaw detection of network protocol implementations," in *2008 IEEE International Conference on Network Protocols*, pp. 114–123, Oct 2008.

[63] W. Mallouli, G. Morales, and A. Cavalli, "Testing security policies for web applications," in *2008 IEEE International Conference on Software Testing Verification and Validation Workshop*, pp. 269–270, April 2008.

[64] N. Antunes and M. Vieira, "Designing vulnerability testing tools for web services: approach, components, and tools," *International Journal of Information Security*, vol. 16, pp. 435–457, Aug 2017.

[65] M. Peroli, D. Meo.F., Viganò.L., and D. Guardin., "Mobster: A model-based security testing framework for web applications," *Software Testing Verification and Reliability*, vol. 28, no. 8, 2018.

[66] A. Armando, R. Carbone, L. Compagna, K. Li, and G. Pellegrino, "Model-checking driven security testing of web-based applications," in *2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, pp. 361–370, April 2010.

[67] R. Oliveira, N. Laranjeiro, and M. Vieira, "Wsfaggressor: An extensible web service framework attacking tool," 12 2012.

[68] C. Hariharan and C. Babu, “Security testing of orchestrated business processes in soa,” in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 1426–1430, May 2014.

[69] R. Yang, G. Li, W. Lau, K. Zhang, and P. Hu, “Model-based security testing: An empirical study on oauth 2.0 implementations,” 2016.

[70] H. Radwan and K. Prole, “Code pulse: Real-time code coverage for penetration testing activities,” in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, pp. 1–6, April 2015.

[71] L. Thomas, W. Xu, and D. Xu, “Mutation analysis of magento for evaluating threat model-based security testing,” in *Proceedings - International Computer Software and Applications Conference*, pp. 184–189, 2011. Cited By :2.

[72] E. Fong and V. Okun, “Web application scanners: Definitions and functions,” in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS’07)*, pp. 280b–280b, Jan 2007.

[73] M. Jain and D. Gopalani, “Testing application security with aspects,” in *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016*, 2016.

[74] G. Pellegrino, M. Johns, S. Koch, M. Backes, and C. Rossow, “Deemon: Detecting csrf with dynamic analysis and property graphs,” 2017.

[75] C.-P. Bezemer, A. Mesbah, and A. Deursen, “Automated security testing of web widget interactions,” pp. 81–90, 01 2009.

[76] S. Huang, H. Lu, W. Leong, and H. Liu, “Craxweb: Automatic web application testing and attack generation,” in *2013 IEEE 7th International Conference on Software Security and Reliability*, pp. 208–217, June 2013.

[77] A. Avancini and M. Ceccato, “Circe: A grammar-based oracle for testing cross-site scripting in web applications,” in *2013 20th Working Conference on Reverse Engineering (WCRE)*, pp. 262–271, Oct 2013.

[78] J. Chen and C. Wu, "An automated vulnerability scanner for injection attack based on injection point," in *2010 International Computer Symposium (ICS2010)*, pp. 113–118, Dec 2010.

[79] A. Anchlia and S. Jain, "A novel injection aware approach for the testing of database applications," in *2010 International Conference on Recent Trends in Information, Telecommunication and Computing*, pp. 311–313, March 2010.

[80] M. Mohammadi, B. Chu, H. Lipford, and E. Murphy-Hill, "Automatic web security unit testing: Xss vulnerability detection," in *2016 IEEE/ACM 11th International Workshop in Automation of Software Test (AST)*, pp. 78–84, May 2016.

[81] M. Imran, F. Eassa, and K. Jambi, "Using agent technology for security testing of web based applications," *sede2015*, 10 2015.

[82] A. Vernotte, C. Botea, B. Legeard, A. Molnar, and F. Peureux, "Risk-driven vulnerability testing: Results from ehealth experiments using patterns and model-based approach," in *Risk Assessment and Risk-Driven Testing* (F. Seehusen, M. Felderer, J. Großmann, and M.-F. Wendland, eds.), (Cham), pp. 93–109, Springer International Publishing, 2015.

[83] M. Alalfi, J. Cordy, and T. Dean, "Automated reverse engineering of uml sequence diagrams for dynamic web applications," in *2009 International Conference on Software Testing, Verification, and Validation Workshops*, pp. 287–294, April 2009.

[84] J. Bozic and F. Wotawa, "Purity: A planning-based security testing tool," in *Proceedings - 2015 IEEE International Conference on Software Quality, Reliability and Security-Companion, QRS-C 2015*, pp. 46–55, 2015. Cited By :5.

[85] K. Li, C. Hebert, J. Lindemann, M. Sauter, H. Mack, T. Schröer, and A. Tiple, "Tool support for secure programming by security testing," in *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 1–4, April 2015.

[86] B. Smith, L. Williams, and A. Austin, "Idea: Using system level testing for revealing sql injection-related error message information leaks," in *Engineering Secure*

Software and Systems (F. Massacci, D. Wallach, and N. Zannone, eds.), (Berlin, Heidelberg), pp. 192–200, Springer Berlin Heidelberg, 2010.

[87] S. Türpe, A. Poller, J. Trukenmüller, J. Repp, and C. Bornmann, "Supporting security testers in discovering injection flaws," in *Proceedings - Testing: Academic and Industrial Conference Practice and Research Techniques, TAIC PART 2008*, pp. 64–68, 2008. Cited By :1.

[88] Z. Djuric, "A black-box testing tool for detecting sql injection vulnerabilities," in *2013 Second International Conference on Informatics Applications (ICIA)*, pp. 216–221, Sep. 2013.

[89] L. Qian, J. Wan, L. Chen, and X. Chen, "Complete web security testing methods and recommendations," in *2013 International Conference on Computer Sciences and Applications*, pp. 86–89, Dec 2013.

[90] D. Simos, B. Garn, J. Zivanovic, and M. Leithner, "Practical combinatorial testing for xss detection using locally optimized attack models," in *2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 122–130, April 2019.

[91] J. Bozic and F. Wotawa, "Xss pattern for attack modeling in testing," in *2013 8th International Workshop on Automation of Software Test (AST)*, pp. 71–74, May 2013.

[92] A. Gouaillard and L. Roux, "Real-time communication testing evolution with webrtc 1.0," in *2017 Principles, Systems and Applications of IP Telecommunications (IPTComm)*, pp. 1–8, Sep. 2017.

[93] P. Stephanow and K. Khajehmoogahi, "Towards continuous security certification of software-as-a-service applications using web application testing techniques," in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, pp. 931–938, 2017. Cited By :4.

[94] V. De la Iglesia, "Gestión de procesos: una herramienta útil para ayudarte a ser más eficiente,"

[95] “Real academia española: Diccionario de la lengua española,” vol. 23.ª ed., 2019.

Capítulo 3

Técnicas de minería de datos y aprendizaje automático para segmentación de clientes bancarios: un mapeo sistemático de literatura

Cruz Maricel Monge Guzmán

3.1. Resumen

Contexto: La minería de datos y el aprendizaje automático pertenecen a un amplio rango de tecnologías que buscan aplicar técnicas y algoritmos a conjuntos de datos. Esto con el fin de analizarlos y extraer información útil para solventar problemas en diferentes áreas. En el caso del sector bancario, la necesidad de conocer las características de cada cliente implica una ventaja empresarial porque pueden segmentarlos y ofrecerles productos y servicios cada vez más personalizados. **Objetivo:** identificar y caracterizar la literatura existente sobre las técnicas de minería de datos y aprendizaje automático utilizadas para la segmentación de clientes bancarios, las herramientas que soportan la implementación de las técnicas, los conjuntos de datos utilizados y las métricas de evaluación aplicadas. **Metodología:** se desarrolla un mapeo sistemático de literatura que incluye 87 estudios primarios publicados en el pe-

ríodo 2005-2019. **Resultados:** los paradigmas que más se reportaron fueron *decision tree* con 88 y *linear predictors* con 55. Con respecto a los 51 estudios que reportaron herramientas: Weka con 13 y Matlab con 12, son las más referenciadas. En cuanto a los conjuntos de datos utilizados para la experimentación, el repositorio *UCI Machine Learning* de la Universidad de California fue el que más se reportó con 60 referencias. Dentro de estos conjuntos de datos los predictores más empleados fueron la edad, el trabajo, el género, la temporalidad y las características crediticias. Además, en las métricas de evaluación se mostró una clara tendencia a utilizar *accuracy* que se reportó en 66 estudios. **Conclusiones:** existen diversas técnicas de minería de datos y aprendizaje automático que se han aplicado al problema de segmentación de clientes, con tendencias claras con respecto a las técnicas, las herramientas, los conjuntos de datos y las métricas de evaluación.

3.2. Introducción

En los últimos años, con la aparición de tecnologías nuevas e innovadoras en el campo de la analítica, tales como: la minería de datos, la inteligencia artificial, el *Big Data*, el aprendizaje automático, entre otros, se ha ido ampliando la investigación en diferentes áreas de aplicación. Estas tecnologías han emergido con el fin de manejar la gran cantidad de datos que se generan cada día [1]. Un ejemplo de esto es la industria bancaria, donde el estudio y la experimentación de esas tecnologías ha permitido resolver y automatizar problemas comunes [2].

Uno de los requerimientos es conocer al cliente por parte del negocio, (sus gustos, preferencias y tendencias de uso de algún producto o servicio) para personalizar los productos y servicios que se ofrecen. Un punto relacionado es la segmentación de clientes, que se refiere a su división en grupos que tengan características en común [3]. Esta agrupación se basa en el conocimiento que se tenga acerca del cliente. Este sistema puede representar una ventaja competitiva que permite incorporar nuevos clientes y mantener la fidelidad de los que ya han sido captados por el negocio [3].

El proceso de segmentación de clientes se debe dar de forma estratégica y mediante seguimiento constante, donde se busque generar calidad en los productos o servicios ofrecidos por el negocio, de vista al cliente. Formalizar una estrategia de

este tipo, requiere de datos base que permitan visualizar, revisar y proyectar la toma de decisiones de la manera más efectiva posible. En el caso del conocimiento del cliente por parte del negocio pueden variar los datos según parámetros demográficos, económicos, sociales, entre otros [3]. Las tecnologías de análisis de datos como la minería de datos y el aprendizaje automático, buscan tomar dichos datos y manipularlos, procesarlos y generar información útil para la toma de decisiones, de una forma automatizada, rápida y efectiva [1].

Específicamente, la minería de datos se refiere al proceso de detectar patrones en un conjunto de datos [1]. Las agrupaciones de los datos se dan por medio de las variables que componen el conjunto y que poseen características que contienen alguna relación entre sí. La minería de datos no solamente implica utilizar un modelo y aplicarlo a un problema, es necesario verificar que la técnica por utilizar sea la correcta según una necesidad de análisis, normalmente planteada desde el punto de vista del negocio. El proceso de validación de la resolución del problema por medio del descubrimiento de patrones aplicado en el modelo, puede ser apoyado por la revisión de un experto en el dominio [1].

El aprendizaje automático es el proceso de entrenar un modelo o algoritmo con una serie de datos y que, a partir de estos pueda ser capaz de relacionar características e identificar nuevos caminos. El proceso de aprendizaje pasa por las etapas de entrenar el modelo, observar los cambios ocurridos y hacer una comparación con el pasado [1].

Los términos de minería de datos y aprendizaje automático son complementarios porque algunos de los algoritmos se pueden agrupar en cualquiera de las dos categorías. La minería de datos necesita de la capacidad de aprendizaje para que los algoritmos sean escalables, y el aprendizaje automático requiere del descubrimiento de patrones para hacer que el modelo pueda ampliar las características de los datos que se utilizan para el entrenamiento inicial [1].

El objetivo de este estudio es identificar y caracterizar la literatura existente sobre las técnicas de minería de datos y aprendizaje automático utilizadas para la segmentación de clientes bancarios, las herramientas que soportan la implementación de las técnicas, los conjuntos de datos utilizados y las métricas de evaluación aplicadas. Para el desarrollo del estudio, se realizó un mapeo sistemático para clasificar la literatura

existente en el área de investigación.

Este análisis está estructurado de la siguiente forma: en la sección 3.3 se presenta el marco teórico. En la sección 3.4 se explica el trabajo relacionado. Posteriormente, en la sección 3.5 se detalla la metodología (definición del objetivo, proceso de búsqueda y selección de estudios, definición y aplicación de los criterios de inclusión y exclusión, evaluación de calidad, extracción de datos, análisis de los datos y amenazas a la validez). En la sección 3.6 se muestran los resultados para cada pregunta de investigación. En la sección 3.7 se discuten los principales hallazgos. En la sección 3.8 se detallan las lecciones aprendidas durante el trabajo de investigación. Finalmente, en la sección 3.9 se detallan las conclusiones y el trabajo futuro.

3.3. Marco teórico

La segmentación del cliente es el proceso de dividir a los usuarios en diferentes grupos, con el propósito de aumentar su satisfacción y, por lo tanto, las ganancias comerciales. Esta clasificación se basa en las características comunes de los clientes. La segmentación se puede realizar en función de criterios como el grado de fidelidad, la frecuencia de compra, el volumen de compra, la demografía, entre otros [3].

El proceso de segmentación de clientes se divide en una serie de etapas que son: la identificación del cliente, la promoción de la relación activa con él, el incremento del valor del cliente y su mantención un cliente con beneficios rentables para el negocio [3]. En el caso de los bancos, la segmentación de clientes se puede dar desde dos perspectivas, la primera donde el usuario demanda la necesidad de buscar los servicios que le puede ofrecer el negocio y la segunda es la identificación de los clientes con mayor valor para hacer la oferta de servicios. En general, este proceso de segmentación de clientes es determinado realmente por la necesidad que se deba solucionar [3].

La minería de datos es un proceso de identificación y descubrimiento de patrones, tendencias y relaciones en un conjunto de datos [1]. Los patrones que se identifiquen en el análisis deben evaluarse con respecto a las expectativas del negocio y determinar si funcionan como predictores para generar más información.

Las técnicas de minería de datos pueden ser utilizadas para resolver problemas co-

munes que se presentan en las diferentes industrias, tales como el mercadeo dirigido al cliente, la segmentación de estos, la previsión de quiebra de un negocio, el manejo de riesgos, la detección de fraude, entre otros [2]. Las diferentes técnicas de minería de datos han ayudado a automatizar o semiautomatizar los procesos de análisis, con el fin de optimizar las soluciones a los problemas mencionados [1]. Específicamente en el sector de la industria bancaria, la minería de datos ha sido ampliamente utilizada. Dado su potencial de ofrecer ventajas competitivas a los bancos que las utilizan [4].

La minería de datos se divide en dos grandes áreas: la analítica descriptiva y la predictiva. La analítica descriptiva tiene como principales algoritmos los de agrupación y los de asociación [4]. En la industria bancaria estos algoritmos pueden aplicarse en el área de mercadeo ya que ayudan a descubrir relaciones entre los datos. Por ejemplo, pueden agrupar cuáles son los productos que más utiliza un cliente y ofrecerle un producto de una gama similar, o pueden analizar tendencias históricas a nivel de productos o servicios ofrecidos y hacer una promoción midiendo un porcentaje de éxito basado en datos existentes [5].

En el caso de la analítica predictiva, tiene como principales algoritmos los de clasificación y los de series de tiempo [4]. Estos algoritmos son útiles en casos como la clasificación, adquisición y lealtad del cliente. Por ejemplo, es necesario clasificar un cliente antes de otorgarle un crédito. Entre mejor puntaje posea, mejor condición le otorga el banco ya que tiene un menor riesgo.

El aprendizaje automático investiga cómo las computadoras pueden aprender o mejorar su rendimiento en función de los datos [6]. Normalmente, los algoritmos de aprendizaje automático pasan por un entrenamiento previo para que el modelo pueda tener patrones con los cuales reconocer y comparar nuevos datos. Este proceso es evolutivo, es decir, todo el tiempo el algoritmo puede crecer al reconocer patrones complejos y generar decisiones inteligentes basadas en datos [6].

Las técnicas de aprendizaje automático tienen cuatro grandes categorías: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semisupervisado y aprendizaje reforzado. Los algoritmos de aprendizaje supervisado están basados en métodos de clasificación donde deben tener un entrenamiento de datos previo. Las técnicas de aprendizaje no supervisado se basan en métodos de agrupación que intentan hacer

descubrimiento de clases en los datos. Los algoritmos de aprendizaje semisupervisado son una combinación entre las dos primeras categorías, por lo cual pueden usar cualquier método para generar más información a partir de los datos base. Por último, el aprendizaje reforzado se basa en el conocimiento de un usuario experto en el dominio, con el fin de etiquetar un conjunto de datos que no pudo ser sintetizado por el programa [6].

Los términos de minería de datos y aprendizaje automático son complementarios porque algunos de los algoritmos se pueden agrupar en cualquiera de las dos tecnologías. La minería de datos necesita de la capacidad de aprendizaje para los algoritmos sean escalables y el aprendizaje automático necesita del descubrimiento de patrones para hacer que el modelo crezca [1].

Existen diferentes clasificaciones de las técnicas de minería de datos y aprendizaje automático. En este estudio se tomó, como referencia para la clasificación, el libro *Understanding machine learning: From theory to algorithms* de Shalev-Shwartz y Ben-David [7], el cual propone la siguiente clasificación de técnicas:

- *Linear predictors:* los algoritmos que pertenecen a esta clasificación están basados en funciones lineales de predicción y en aprendizaje supervisado, ya que necesitan de patrones de entrenamiento para predecir nuevos datos. Estos tipos de algoritmos son fáciles de interpretar y son relacionados con la solución de problemas con lenguaje natural. Dentro de este paradigma se encuentran algoritmos como: *logistic regression*, *linear regression*, *linear programming* y *perceptron algorithm* [7].

- *Boosting:* se generó a partir de una pregunta teórica y se convirtió en una herramienta para el aprendizaje automático. En general, el proceso inicia con un aprendizaje básico basado según el entrenamiento del modelo por datos y conforme avanza va generando más conocimiento que enriquece la clase a la que pertenece el predictor. En este paradigma se encuentran algoritmos como: *boosting*, *bagging* y *adaptative boosting* [7].

- *Support vector machines:* busca una separación de complejidad. Los algoritmos intentan dividir el conjunto de datos de entrenamiento en separadores, que serán utilizados para segmentar las dimensiones que tienen como fin cap-

tar la mayor cantidad de convergencias entre los datos. Algunos ejemplos de técnicas dentro de este paradigma son: *sequential minimum optimization* y *fuzzy support vector machine* [7].

- *Decision trees:* se encarga de predecir si un dato corresponde a una clase "X". Esta clasificación se hace mediante el recorrido de un árbol de decisión desde el nodo raíz hasta una hoja. Existen variaciones donde los recorridos se hacen por distancias, por pesos o al azar. Ejemplos de algoritmos de este paradigma son: *random forest*, *C4.5 tree*, *clasification and regression tree* y *naïve bayes tree* [7].

- *Nearest neighbor:* se puede considerar como una de las más simples de aprendizaje automático, ya que los algoritmos se entrenan por medio de un conjunto de datos, para posteriormente predecir la etiqueta de cualquier nueva instancia, tomando como base las etiquetas de los nodos más cercanos. En este paradigma al igual que en el de *decision tree*, se dan variaciones de los algoritmos por medio de distancias y pesos. Ejemplos de algoritmos de este paradigma son: *K nearest neighbor*, *simulated annealing algorithm* y *weighted k nearest neighbor* [7].

- *Neural networks:* está inspirado en el modelo de redes neuronales del cerebro y su comportamiento. En general, consiste en una serie de capas interconectadas entre sí por una red que transporta información de un nodo a otro. Las capas internas son las más avanzadas y permiten realizar los cálculos y relaciones entre sí. La capa inicial y la capa de salida son más básicas que la capa intermedia, ya que solamente admiten los datos y los comunican correspondientemente. Algunos ejemplos de este paradigma son: *bayes network*, *backpropagation* y *neural networks* [7].

- *Clustering:* se basa en el análisis exploratorio de los datos. La idea es realizar agrupaciones de la información, considerando características que los unen o los separan de otros grupos. Las variaciones entre el tipo de agrupación que se elija dependen totalmente del problema por resolver, así, para esta clasificación existen muchos algoritmos con variaciones. Algunos ejemplos de algoritmos de este paradigma son: *K mean* y *clustering hierarchical* [7].

- *Dimensionality reduction:* se basa en el proceso de tomar datos de una dimensión alta y mapearlos a una dimensión mucho menor. Este proceso se realiza con el fin de ir reduciendo la complejidad de los datos, pero tiene el problema de que se puede dar pérdida de información. Este paradigma utiliza matrices y vectores aleatorios basados en el teorema "Johnson-Lindenstrauss". Un ejemplo de algoritmo dentro de este paradigma es: *principal component analysis* [7].

- *Generative models:* los que pertenecen a esta clase, buscan encontrar la distribución subyacente de los datos de forma paramétrica, con el fin de estimar dichos parámetros. Este tipo de algoritmos tienen el problema de que el aprendizaje es cada vez más complejo. Ejemplos de algoritmos de este paradigma son: *inear discriminant analysis* y *naïve bayesian* [7].

Otro punto por considerar al aplicar las técnicas de minería de datos y aprendizaje automático, es que hay que realizar un proceso de preparación de los datos con el fin de que estos coincidan con lo que necesita el modelo. En la Figura 3.1 se muestra un resumen de las etapas que se necesitan para la preparación del conjunto de datos. Esta preparación incluye varias etapas: la primera es la selección del conjunto de datos y consiste en elegir y extraer desde las diferentes fuentes los datos necesarios para el análisis. La segunda etapa es el preprocesamiento de los datos, que consiste en hacer una limpieza de los atributos precisos, donde se remueven todos los datos que pueden causar errores en el modelo. Estos datos pasan por el proceso de integración para tenerlos agrupados dentro de una misma fuente. Como tercera etapa está la transformación, que es donde se preparan los datos con el formato necesario para el proceso de minería. La cuarta etapa es la aplicación de las técnicas de minería de datos y aprendizaje automático a los datos manipulados con el fin de encontrar patrones. Como quinta y última etapa se encuentra la interpretación y evaluación, en donde se identifica el valor de los patrones basados en las métricas. Esta última etapa es la utilizada para la toma de decisiones de los negocios [4]. Todo este proceso puede ser reiterativo mientras se detecta cuál es la mejor solución basada en el conjunto de datos que se esté utilizando.

Con respecto a las métricas que se usan para evaluar las técnicas de minería de datos y aprendizaje automático, se encuentran las de rendimiento, que consisten en

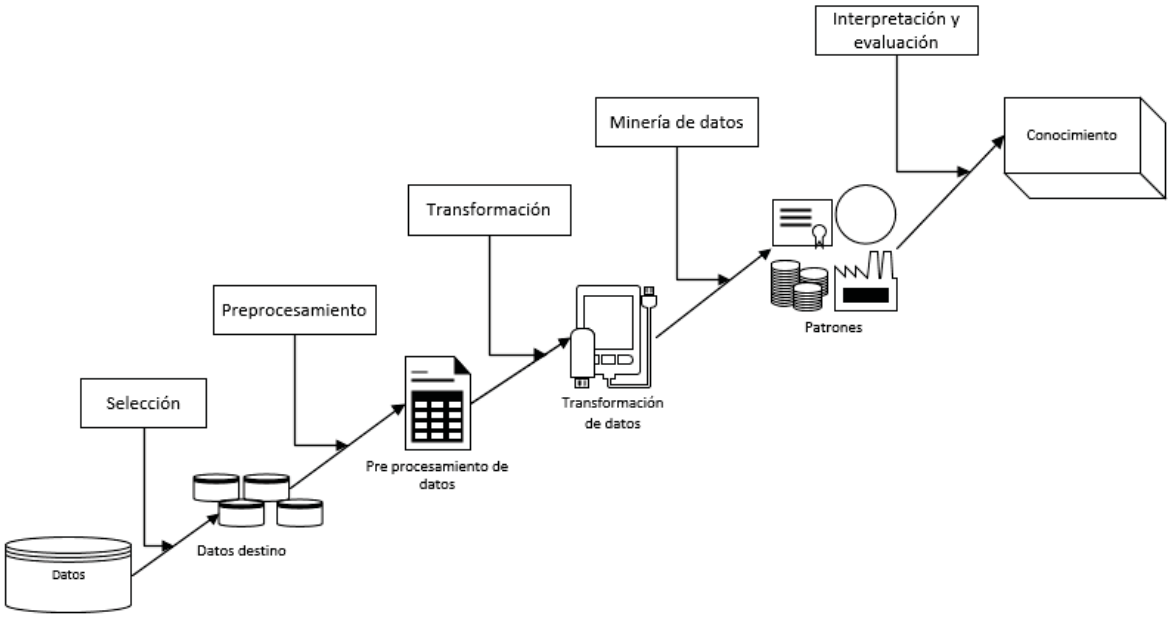


Figura 3.1: Proceso de preparación de los datos [4].

evaluar qué tan bueno es un modelo para predecir cada clase [6]. Las fórmulas que definen estas métricas están basadas en la matriz de confusión, que es útil para determinar si un clasificador reconoció correctamente una muestra según ciertos parámetros o configuraciones [1].

Las métricas de rendimiento para la evaluación de técnicas de minería datos y aprendizaje automático más comunes son: *accurancy*, *error rate*, *recall*, *specificity*, *precision* y *F-measure*. Además, existen variaciones sobre estas métricas y algunas evaluaciones de rendimiento son específicas para cada paradigma, como por ejemplo *information gain* que es para los algoritmos del paradigma *decision tree* [6].

3.4. Trabajo relacionado

En el contexto de segmentación de clientes bancarios, no se encontraron estudios similares a esta investigación. Sin embargo, sí se han analizado y evaluado diferentes técnicas de minería de datos y aprendizaje automático para solucionar problemas relacionados con la industria bancaria. Algunos de estos estudios han revisado diferentes herramientas donde se puedan aplicar experimentos de minería de datos y

aprendizaje automático y han obtenido algunas métricas de evaluación. Cabe destacar que los artículos revisados son secundarios.

Bhambri [4] estudió varias técnicas de minería de datos entre estas: asociación, agrupamiento, clasificación y pronóstico. La asociación está relacionada con la frecuencia para encontrar elementos en el conjunto de datos, incluye varios tipos tales como las siguientes reglas: de asociación multinivel, de asociación multidimensional, de asociación cuantitativa, de asociación directa y de asociación indirecta. La agrupación es la identificación de objetos similares. El pronóstico puede modelar las relaciones entre atributos dependientes e independientes, también incluye algoritmos como: regresión logística, árboles de decisión y redes neuronales, y los modelos de clasificación utilizados para identificar la población de registros. Todas estas técnicas se aplicaron para resolver problemas bancarios, como la detección de fraudes, *marketing*, gestión de riesgos y riesgos comerciales.

Pulakkazhy et al. [8] realizaron un análisis sobre el preprocesamiento de los datos antes de aplicar técnicas de minería de datos. Los pasos incluyen: selección de datos, limpieza, integración, transformación y reducción de datos. Después de aplicar las técnicas de minería de datos el siguiente paso es la evaluación de patrones y la presentación del conocimiento. Este estudio usa técnicas como *multilevel association rule*, *multidimensional association rule*, *quantitative association rule*, *direct association rule*, *indirect association rule*, *decision tree*, *neural networks* y *K-means*. Adicionalmente, mencionan diferentes casos sobre los que se puede aplicar minería de datos en la industria bancaria, como por ejemplo el manejo de riesgos y la detección de fraude, el mercado para atraer nuevos clientes al negocio y procesos para detectar el lavado de dinero.

Hasheminejad et al. [3] hicieron un amplio análisis de conjuntos de datos que han sido utilizados en evaluaciones de minería de datos, el cual detalla los parámetros usados, el tamaño del conjunto de datos y el tipo de preprocesamiento por el que se pasaron los datos. De igual manera, hacen una revisión de las técnicas más usadas de minería de datos y de los criterios de evaluación de rendimiento más comunes. Las técnicas más empleadas fueron: *self organizing map*, *k-means*, *decision trees*, *neural networks* y *naïve bayes*. Además, descubrieron que las métricas más aplicadas son *accuracy*, *precision*, *F-mesuare*, entre otros.

Goebel et al. [9] elaboraron un resumen de herramientas usadas en experimentos de minería de datos donde explican las tareas realizadas, los algoritmos evaluados y las variables empleadas. Algunas de las tareas aplicadas en las herramientas son: variedad de fuentes de datos, acceso a datos *offline/online*, lenguaje de consultas, el modelo de datos aplicado, entre otros. Las técnicas incluidas en este estudio son: predicción, regresión, clasificación, agrupación y asociación. Este estudio es importante de destacar porque hay pocos proyectos que revisan características de herramientas de minería de datos y aprendizaje automático en el contexto de datos bancarios.

Aunque todos los estudios secundarios revisados reportan técnicas en minería de datos y aprendizaje automático, ninguno se basa en el contexto de segmentación de clientes. Además, no todos mencionan las herramientas utilizadas en las implementaciones, los conjuntos de datos usados o las métricas de evaluación. No obstante, la revisión de estos trabajos da una visión sobre lo que ha sido más analizado en la literatura en el contexto de datos bancarios.

3.5. Metodología

La metodología aplicada para realizar este estudio secundario se basa en los lineamientos y recomendaciones planteadas por Petersen et al. [1] y Kitchenham et al. [11]. Estos consisten en un conjunto de pasos que incluyen procesos de selección, evaluación, extracción y análisis de los estudios primarios.

3.5.1. Objetivo

El objetivo de este estudio formulado con el modelo GQM (*Goal Question Metric*) [3] es *analizar* las técnicas de minería de datos y aprendizaje automático para la segmentación de clientes, *con el propósito de* caracterizarlas *con respecto a* las técnicas, las herramientas, los conjuntos de datos y las métricas de evaluación *desde el punto de vista de* la investigadora *en el contexto de* datos bancarios.

3.5.2. Preguntas de investigación

Las preguntas de investigación que guiaron el desarrollo de este estudio son las siguientes:

RQ1. ¿Cuáles técnicas de minería de datos y aprendizaje automático se han utilizado para la segmentación de clientes bancarios? La respuesta de esta pregunta permitirá determinar qué técnicas han sido utilizadas en el área de estudio y los paradigmas en los que se agrupan dichas técnicas.

RQ2. ¿Cuáles herramientas soportan la implementación de técnicas de minería de datos y aprendizaje automático para la segmentación de clientes bancarios? Con esta respuesta se logrará conocer las herramientas que soportan las técnicas extraídas de la pregunta de investigación 1.

RQ3. ¿Cuáles conjuntos de datos y métricas se han utilizado para evaluar las técnicas de minería de datos y aprendizaje automático para la segmentación de clientes bancarios? Esta pregunta posibilita identificar los conjuntos de datos junto con sus características y las métricas utilizadas para evaluar las técnicas.

3.5.3. Proceso de búsqueda

Para iniciar el proceso de búsqueda de los estudios, se hizo una revisión y selección de artículos de control sobre las bases de datos Scopus [1], IEEE Xplore [2] y Web of Science [3]: [13, 14, 15, 16]. Esto con el fin de tomarlos como referencia y extraer palabras claves que ayudarán a plantear la cadena de búsqueda final.

Artículos de control: a continuación se describen los artículos de control seleccionados.

Gulsoy et al. [13], realizan un estudio sobre la clasificación de clientes bancarios por medio de técnicas de minería de datos. Para el experimento hizo la evaluación de seis algoritmos bajo los mismos criterios con el fin de compararlos, evaluarlos y

[1] https://www2.scopus.com/home.uri
[2] http://ieeexplore.ieee.org/
[3] apps.webofknowledge.com

predecir cuál algoritmo es el indicado para la clasificación de clientes. Las técnicas empleadas fueron: *multi objective evolutionary fuzzy classifier*, *naïve bayes tree*, PART, C4.5, *random tree* y *simple cart*. Además, aplican sobre los resultados varias métricas de rendimiento como: *accuracy*, número de reglas aplicadas, *recall*, *precision* y *kappa statistics*.

Çiğşar et al. [14], hacen una identificación de algoritmos de minería de datos que se utilizan para predecir diferentes situaciones en la industria bancaria y en la de seguros. Este estudio aplica un experimento con un conjunto de datos reales de un banco de Turquía, tomando en cuenta variables demográficas y socioeconómicas. El trabajo hizo una comparación de seis algoritmos (*naïve bayes*, *bayesian networks*, J48, *random forest*, *multilayer perceptron* y *logistic regression*) por medio de la herramienta Weka y utilizó varias métricas de rendimiento para identificar cuál escenario era el mejor.

Basarslan et al. [15], realizan un estudio donde evalúan diferentes algoritmos (*k nearest neighbor*, *naïve bayes* y *C4.5 tree*) para la adquisición de nuevos clientes. El caso que plantean lo evalúan por medio de cuatro diferentes herramientas y bajo un conjunto de datos de prueba accesible públicamente, que contiene características de mercadeo bancario. Los algoritmos utilizados fueron comparados entre sí mediante diferentes métricas de rendimiento como: *accuracy*, *precision* y *F-mesuare*. Además, para hacer los experimentos utilizaron las herramientas R, Knime, RapidMiner y Weka.

Das [16], hace una identificación y clasificación de clientes por medio de técnicas de aprendizaje automático. Realiza una comparación con base en la efectividad presentada por los algoritmos y las características que muestra el conjunto de datos. Las técnicas que utilizaron son: *naïve bayes*, *k nearest neighbor* y *support vector machine*. La herramienta usada en este estudio fue RapidMiner.

Cadena de búsqueda: se construyó a partir de los títulos, palabras clave y resumen de los artículos de control. En este proceso se realizaron diversos pilotajes para validar que la cadena construida fuera la más adecuada, por lo que fue necesaria la lectura completa y detallada de los artículos de control fue necesaria para este proceso. Se desarrolló el modelo PICO (Población, Intervención, Comparación, Salidas) [4], don-

de se definieron los siguientes grupos:

P: Datos de clientes bancarios.

I: Técnicas de minería de datos y aprendizaje automático para la segmentación.

C: No aplica.

O: Técnicas, herramientas, conjuntos de datos y métricas utilizadas.

Como resultado, se definió la siguiente cadena de búsqueda:

```
("marketing" OR "credit") AND ("client" OR "customer") AND ("acqu*"
    OR "scor*" OR "classif*") AND ("bank*") AND ("mining" OR "
   machine_learn*" OR "predict*" OR "classif*")
```

Esta cadena se utilizó para la búsqueda en las bases de datos Scopus y Web of Science. En el caso de IEEE Xplore, se aplicó una variación donde se modificó la agrupación de los términos, ya que la búsqueda avanzada de la base de datos tiene una restricción en cuanto a la cantidad de los asteriscos (*) que pueden utilizar. Por esa razón se usó la siguiente cadena:

```
("marketing" OR "credit") AND ("client" OR "customer") AND ("acqu*"
   OR "scor*" OR "classif*") AND ("bank" OR "banking" OR "bankcard"
    OR "bankers" OR "banks" OR "banker" OR "banked" OR "bankcards"
   OR "bankings") AND ("mining" OR "predict*" OR "classif*" OR "
   machine_learning")
```

Período de búsqueda: el protocolo base del mapeo fue desarrollado durante el primer semestre del 2019. La búsqueda automatizada final se realizó en octubre del 2019. El número de estudios por base de datos fue el siguiente: Scopus: 369, IEEE Xplore: 77 y Web of Science: 88. Se descargó para cada estudio el título, el año de publicación, los autores, el resumen y las palabras clave. Con base en esta información se eliminan duplicados, para obtener un total de 409 estudios, a los que se les debía aplicar el proceso de inclusión y exclusión.

3.5.4. Proceso de selección de estudios

Para determinar si los estudios extraídos de la búsqueda automatizada eran relevantes a esta investigación, se definieron criterios de inclusión y exclusión que fueron aplicados sobre el título, las palabras clave y el resumen de cada estudio. Para este análisis se excluyeron los estudios que no cumplían con la fórmula (E1 OR E2 OR E3), donde cada componente se define como:

E1. Estudios que no estén disponibles en texto completo.

E2. Estudios que no estén escritos en el idioma inglés.

E3. Estudios que no sean primarios.

Los artículos que pasaron el filtro de la primera fórmula tuvieron que cumplir con la segunda fórmula: (I1 AND I2 AND I3), donde cada componente se define como:

I1. Estudios que utilicen datos bancarios.

I2. Estudios que usen técnicas de minería de datos y aprendizaje automático.

I3. Estudios que describan procesos de segmentación de clientes.

Una vez concluido este proceso, 100 estudios fueron seleccionados para pasar las etapas de extracción y análisis de resultados. En la Figura 3.2 se muestra la cantidad de estudios obtenidos de cada uno de los procedimientos anteriores. Después de aplicar el proceso de lectura completa de cada estudio, quedaron 87, los cuales se detallan en el Apéndice 3.A.

Se excluyeron 13 estudios después de la lectura completa, porque el resumen no era claro para identificar si el estudio era relevante para la investigación. Algunos fueron excluidos también porque eran más referidos al área de mercadeo o procesos bancarios y no basados en el área de Ingeniería de *Software*.

3.5.5. Evaluación de calidad

La evaluación de calidad se aplicó sobre los estudios primarios que pasaron el proceso de selección, con el fin de determinar si tenían calidad alta o baja. Esta evaluación se hizo con base a la relevancia y el aporte que podía tener cada estudio para

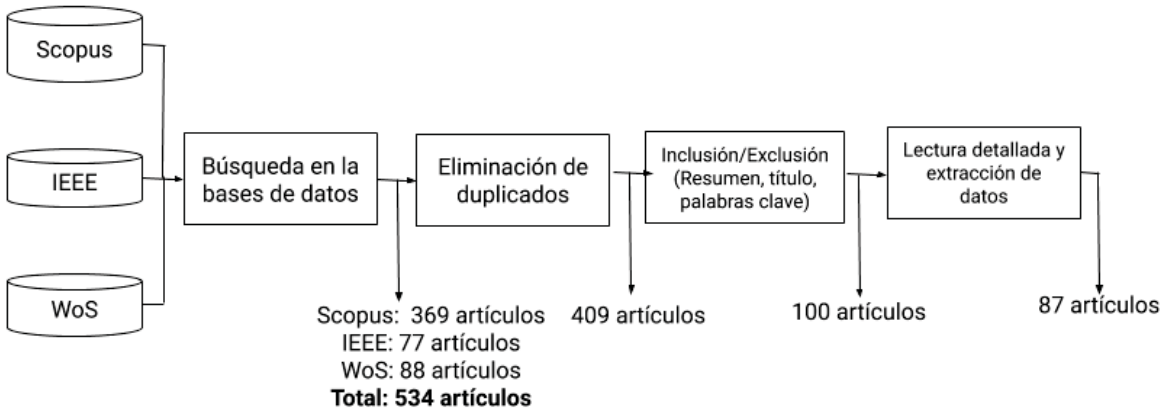


Figura 3.2: Proceso de selección de artículos.

esta investigación. Para ello se definieron los siguientes criterios de evaluación de calidad:

Q1. ¿El estudio describe las técnicas de minería de datos y aprendizaje automático utilizadas?

Q2. ¿El estudio describe las herramientas que soportan las técnicas de minería de datos y aprendizaje automático?

Q3. ¿El estudio describe las métricas de evaluación de rendimiento de las técnicas utilizadas?

Q4. ¿El estudio describe el conjunto de datos utilizado para la evaluación de las técnicas?

El puntaje se asigna a cada pregunta en una escala de 0 a 2, donde 0 = No cumple con el criterio, 1 = Cumple parcialmente el criterio y 2 = Cumple con el criterio. Según la escala definida, el máximo puntaje de calidad que puede alcanzar cualquiera de los estudios es 8 puntos y el mínimo es 0. Cabe aclarar que un estudio que obtenga un puntaje de calidad total 0 no es descartado; sin embargo, se considera como un artículo poco relevante o con bajo nivel de detalle para esta investigación.

El promedio del Q1 evaluado en los 87 estudios dio como resultado 1,41 por lo que se puede mencionar que la mayoría de artículos menciona las técnicas de minería

de datos y aprendizaje automático, pero no todos las explican con detalle. En el caso de la Q2 se obtuvo un promedio de 0,70 lo que implica que muy pocos estudios mencionaron las herramientas con las que realizaron los experimentos y el detalle explicado sobre cada una es muy bajo. Este es un punto por considerar importante para próximos estudios en el área. Con respecto a la Q3 que se refiere al reporte de las métricas de evaluación se obtuvo un promedio de 1,19, por lo que este parámetro también fue muy reportado en los estudios. Además de que aportan el detalle de los resultados obtenidos por cada métrica. Finalmente, para la Q4 se tiene un promedio de 1,34 que está dentro de la media por pregunta. Indica que algunos estudios sí reportaron los conjuntos de datos, pero se debe aumentar el detalle de los datos que se dan. En el Apéndice 3.C se muestran las evaluaciones de calidad de cada pregunta.

En la Figura 3.3 se señalan gráficamente los resultados de la evaluación de calidad sobre los 87 estudios. Se calcularon los percentiles por pregunta y del puntaje total, para obtener un promedio de 4,65 y una mediana de 5, por lo que la calidad de los artículos es aceptable. Como se puede ver en la misma figura existen bigotes en 8 y en 2. Sin embargo, estos valores no se eliminan porque el estudio que reportó un puntaje de 8 tiene gran relevancia en la investigación y si se eliminaran solo los valores mínimos, se estaría sesgando el resultado de la evaluación. En el Apéndice 3.B se detallan los resultados de calidad para cada estudio evaluado y para cada pregunta de calidad.

3.5.6. Extracción de datos

Para realizar el proceso de extracción de datos se creó un formulario con una serie de aspectos a extraer de los estudios seleccionados en los procesos anteriores. Esto con el fin de obtener la información que ayudará a responder las preguntas de investigación. El formulario creado contiene 4 categorías: información general, técnicas (RQ1), herramientas (RQ2) y métricas (RQ3), cada una con sus respectivos elementos. En el Cuadro 3.1 se detallan los elementos extraídos para cada categoría.

Para realizar con éxito este proceso fue necesario realizar la lectura completa de los 87 artículos seleccionados y extraer cada uno de los criterios incluidos en el formulario. En el enlace `https://tinyurl.com/v7gozqs` se detalla el formulario con los datos extraídos de los estudios seleccionados.

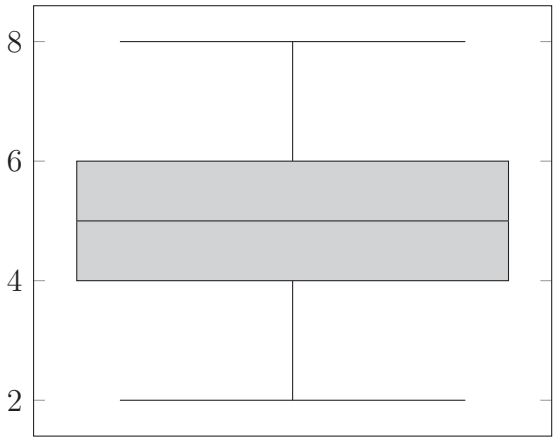


Figura 3.3: Calidad de los estudios primarios incluidos.

Cuadro 3.1: Componentes del formulario de extracción.

Categoría	Componentes
General	Base de datos, id, título, autores, año, resumen, palabras clave, tipo de artículo, artículo de control, Q1, Q2, Q3, Q4, resultado de calidad
Técnicas (RQ1)	Paradigma, nombre, descripción, configuración de la técnica, propósito del estudio
Herramientas (RQ2)	Nombre, descripción
Conjuntos y métricas (RQ3)	Nombre, descripción, fórmula, resultado obtenido, tamaño del conjunto, procesamiento del conjunto, atributos, conjunto de datos, enlace del conjunto, características del conjunto.

3.5.7. Análisis de datos

Para hacer el análisis de datos, lo primero que se hizo fue verificar que los datos tabulados eran los indicados para responder las preguntas de investigación.

Para responder la RQ1, el análisis consistió en identificar las técnicas reportadas en los estudios y hacer un conteo de los que las aplican. Para hacer la tabulación de algoritmos fue necesario utilizar una taxonomía que aplicara para técnicas de minería de datos y aprendizaje automático. Además, en algunos de los estudios se pudo extraer la configuración de la técnica porque aportaba mayor detalle a la investigación. También se extrajo el propósito de cada estudio, y así se pudo relacionar con las técnicas extraídas.

Con respecto a la RQ2 se identificaron las herramientas que soportan las técnicas de minería de datos y aprendizaje automático que se han usado para la experimentación en los estudios seleccionados. Además, se consultó las fuentes oficiales de las herramientas con el fin de extraer más características.

Para la RQ3 se identificaron los conjuntos de datos utilizados para la experimentación en los estudios. Se extrajeron datos como los atributos usados, el tamaño de los conjuntos de datos, si eran repositorios públicos o privados, entre otros. También, se obtuvieron las métricas de evaluación del rendimiento de las técnicas aplicadas y los resultados obtenidos en cada experimento.

3.5.8. Amenazas a la validez

Las amenazas a la validez determinan las fortalezas y las limitaciones de la investigación con respecto a los resultados que se obtuvieron. A continuación, se detallan las amenazas a la validez de este mapeo sistemático de la literatura.

Selección de la cadena de búsqueda y las bibliotecas digitales: la cadena de búsqueda automatizada se implementó mediante la definición de palabras claves a partir de los artículos de control y el modelo PICO. En el caso de las bases de datos elegidas (Scopus, IEEE Xplore y Web of Science), son reconocidas y recomendadas en el área de la Ingeniería de *software*.

Identificación de estudios primarios: se presentaron casos en el proceso de inclusión y exclusión donde no era claro si el estudio debía tomarse en cuenta o no, en tales casos, se decidió incluirlo y posteriormente hacer una lectura completa para validar si el estudio realmente era relevante para la investigación.

Extracción y clasificación de artículos primarios: para limitar el sesgo que pue-

de darse al momento de realizar el proceso de extracción y clasificación de los estudios por parte de la autora de esta investigación, se revisó en varias ocasiones que el formulario se ajustara a lo que se necesitaba extraer de los artículos para dar respuesta a las preguntas de investigación. En el proceso de extracción se dieron casos donde el artículo no era lo suficientemente claro para hacer una clasificación taxonómica de los algoritmos, por lo que se hizo una validación de la teoría, donde se tomó una taxonomía reconocida y que la autora utilizó cuando el autor de algún estudio no reportó la clasificación. Se aplicó la clasificación planteada por Shalev-Shwartz et al. en el libro *Understanding Machine Learning: From Theory to Algorithms* [7], con el fin de agrupar las técnicas. La definición de cada paradigma se encuentra en la sección 3.3. Esta clasificación no es la única que existe sobre minería de datos y aprendizaje automático, pero si es la mejor que se adapta a los resultados obtenidos en esta investigación.

Generalización y síntesis de resultados: la generalización y síntesis de resultados se limita a los estudios que se analizaron. Para minimizar riesgos al presentar los resultados, toda la investigación se realizó siguiendo los protocolos previamente definidos y validados. Esto con el fin de usar las mejores prácticas en el desarrollo del proyecto.

Consulta a expertos: la consulta de un experto previa al desarrollo de la investigación permitió contar con el criterio de dicho usuario y tener a consideración diversos factores de la industria bancaria principalmente. Esta validación apoyó la depuración del proceso de la cadena de búsqueda automatizada. Además, no se realizó el proceso de consulta con un usuario experto en el dominio de minería de datos y aprendizaje automático, ya que se consideró que al ser el eje central de la investigación, por medio de la metodología y el conocimiento previo de la investigadora, se podía aplicar bien el proceso.

3.6. Análisis de resultados

En esta sección se presentan los resultados de la extracción, clasificación y análisis de 87 estudios primarios, donde se abordan las preguntas de investigación anteriormente descritas. Los estudios analizados tienen un periodo de publicación del año 2005 al 2019, donde el año 2018 es en el cual se reportan más publicaciones. En el

Apéndice 3.A se puede encontrar la lista completa de artículos incluidos dentro de este trabajo.

3.6.1. Técnicas de minería de datos y aprendizaje automático para la segmentación de clientes (RQ1)

La primera pregunta de investigación ayuda a identificar cuáles son las técnicas de minería de datos y aprendizaje automático más utilizados para la segmentación de clientes bancarios. En particular, se intentó descubrir tendencias en las técnicas empleadas y las configuraciones usadas en los experimentos o casos evaluados según correspondiera. Además, se relaciona el propósito del estudio con las técnicas reportadas para relacionar mejor cada contexto.

En la Figura 3.4 se muestran los paradigmas utilizados y las técnicas que se clasificaron en cada uno. Los conceptos de minería de datos y aprendizaje automático pueden ser delimitados teóricamente. Sin embargo, en la aplicación práctica la frontera de donde termina uno y comienza el otro no se puede determinar con exactitud, ya que utilizan las mismas técnicas, herramientas, estadísticas, entre otros. La mayor diferencia entre ellas, es que el aprendizaje automático se orienta más a obtener un resultado, mientras que la minería de datos se basa en el descubrimiento de conocimiento [6]. Estos conceptos son complementarios, aunque en algunas ocasiones la minería de datos se ve como un subconjunto o aplicación del aprendizaje automático [1].

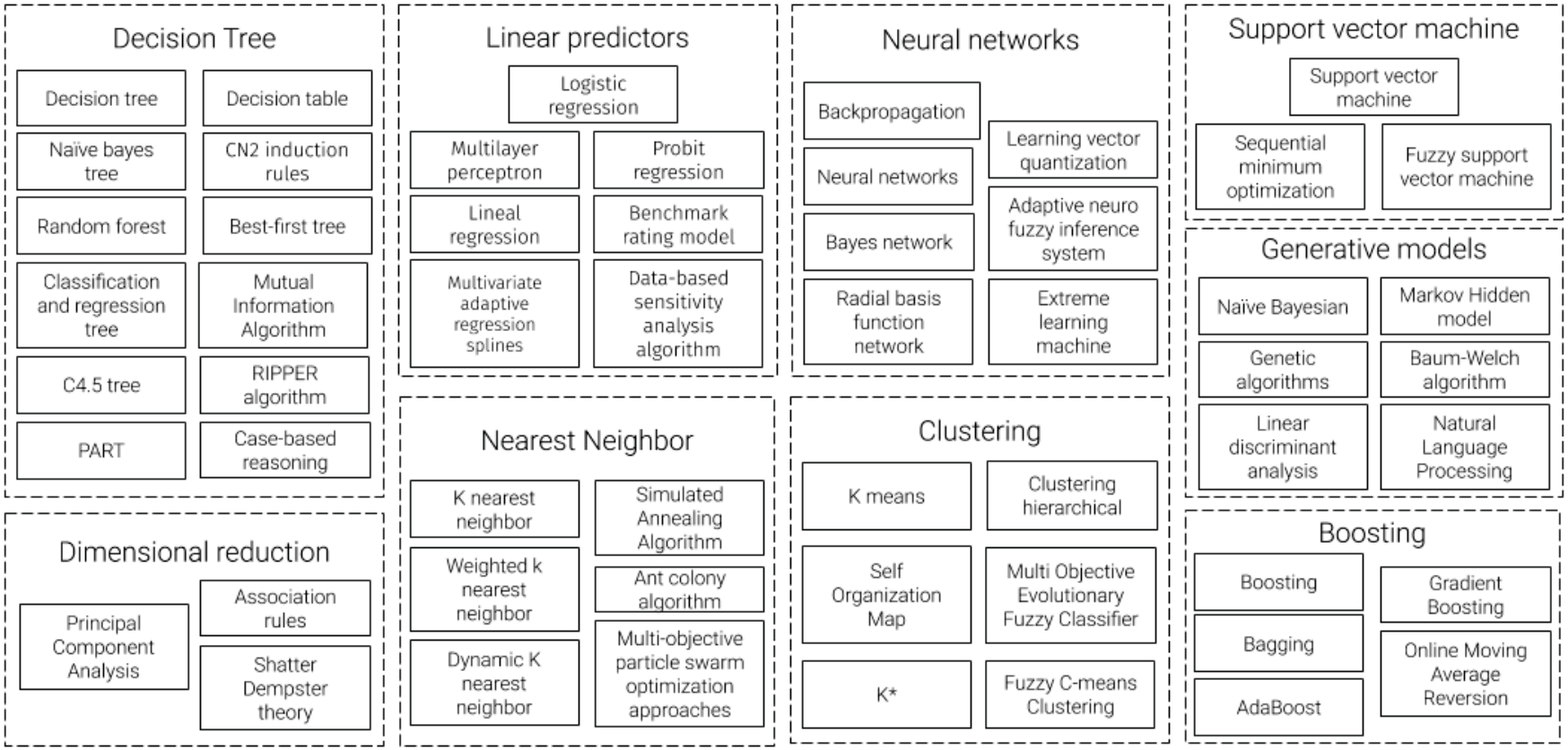


Figura 3.4: Técnicas de minería de datos y aprendizaje automático clasificados por paradigmas [7].

En el Cuadro 3.2 se muestran las técnicas reportadas en los 87 estudios primarios. Además, se señala el paradigma en el que se encuentran clasificadas, la frecuencia y la referencia a cada estudio. Como se puede notar en este cuadro, los dos paradigmas que más se reportaron son *decision tree* y *linear predictors* que agrupan el 42 % de las técnicas reportadas. Es importante destacar que ambos son basados en los algoritmos que componen la analítica predictiva, como: los de clasificación, los de regresión y los de agrupación. En el caso del paradigma *clustering* por el contexto de la investigación se supone que pudo haber tenido más reportes, pero la cantidad de veces que se referenció es baja. A pesar de que en este estudio no se puede aplicar una comparación entre paradigmas para saber cuál es el mejor para el contexto y área de aplicación planteada, por el hecho de no tener las mismas condiciones en los experimentos de los estudios analizados, sí se puede suponer que la gran diferencia entre los paradigmas *decision tree*, *linear predictors* y *clustering* es que los dos primeros son más eficientes que el último en la solución del escenario planteado. Dado que la investigación está propuesta sobre el contexto de segmentación de clientes bancarios, era esperable que la mayoría de técnicas reportadas se encontraran dentro de dichos paradigmas.

Cuadro 3.2: Técnicas de minería de datos y aprendizaje automático clasificados por paradigmas que fueron reportados en los estudios primarios.

Paradigma	Técnicas	Cant.	Estudios
Decision tree	*Decision tree*	20	[18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37]
	Naïve bayes tree	16	[13, 14, 38, 15, 20, 39, 40, 21, 23, 26, 41, 42, 43, 44, 45, 34]
	Random forest	16	[46, 13, 18, 47, 14, 21, 23, 48, 24, 49, 50, 26, 45, 35, 36, 37]
	Classification and regression tree	13	[46, 13, 51, 40, 23, 43, 29, 45, 30, 52, 53, 54, 55]
	C4.5 *tree*	13	[13, 14, 15, 48, 56, 42, 57, 58, 29, 45, 30, 59, 60]

Continúa en la página siguiente.

Paradigma	Técnicas	Cant.	Estudios
	PART	3	[13, 44, 45]
	Decision table	2	[44, 45]
	CN2 *induction rules*	1	[39]
	Best-first tree	1	[45]
	Mutual information algorithm	1	[61]
	RIPPER *algorithm*	1	[62]
	Case-based reasoning	1	[63]
Linear predictors	*Logistic regression*	22	[47, 19, 14, 64, 51, 23, 48, 42, 27, 65, 66, 67, 68, 29, 69, 45, 52, 32, 54, 70, 55, 71]
	Multilayer perceptron	16	[47, 14, 38, 20, 39, 40, 22, 48, 72, 56, 42, 73, 27, 67, 58, 28]
	Linear regression	6	[24, 74, 25, 44, 75, 55]
	Multivariate adaptive regression splines	5	[47, 64, 53, 55, 71]
	Probit regression	3	[76, 77, 78]
	Data-based sensitivity analysis algorithm	1	[61]
	Benchmark rating model	1	[79]
Neural networks	*Neural networks*	21	[20, 23, 24, 74, 25, 26, 43, 66, 80, 29, 28, 30, 52, 31, 32, 81, 54, 55, 34, 35, 82]

Continúa en la página siguiente.

Paradigma	Técnicas	Cant.	Estudios
	Backpropagation	4	[83, 77, 55, 78]
	Bayes network	3	[14, 20, 45]
	Radial basis function network	3	[42, 27, 65]
	Learning vector quantization	2	[84, 62]
	Adaptive neuro fuzzy inference system	1	[85]
	Extreme learning machine	1	[20]
Support vector machine	*Support vector machine*	27	[47, 20, 86, 87, 40, 21, 22, 23, 24, 49, 56, 26, 27, 88, 43, 44, 89, 28, 90, 31, 32, 91, 53, 83, 16, 35, 78]
	Sequential minimum optimization	3	[38, 57, 45]
	Fuzzy support vector machine	2	[88, 78]
Nearest neighbor	*K nearest neighbor*	16	[92, 38, 93, 15, 20, 94, 95, 40, 21, 56, 25, 57, 45, 54, 70, 16]
	Weighted k nearest neighbor	3	[79, 95, 39]
	Dynamic k nearest neighbor	1	[92]
	Simulated annealing algorithm	1	[60]

Continúa en la página siguiente.

Paradigma	Técnicas	Cant.	Estudios
	Ant colony algorithm	1	[83]
	Multi-objective particle swarm optimization approaches	1	[23]
Boosting	*Boosting*	8	[47, 92, 38, 15, 95, 40, 50, 96]
	Bagging	6	[40, 50, 25, 96, 36, 37]
	Adaboost	5	[86, 24, 32, 36, 37]
	Gradient boosting	2	[18, 37]
	Online Moving Average Reversion	1	[33]
Generative models	*Naïve bayesian*	8	[97, 98, 25, 58, 68, 54, 34, 16]
	Linear discriminant analysis	5	[56, 25, 54, 55, 71]
	Genetic algorithms	4	[87, 80, 91, 63]
	Markov hidden model	2	[99, 70]
	Baum-Welch algorithm	1	[99]
	Natural language processing	1	[85]
Clustering	*K means*	6	[20, 39, 21, 28, 91, 82]
	Self Organization Map	4	[84, 39, 44, 45]

Continúa en la página siguiente.

Paradigma	Técnicas	Cant.	Estudios
	*K**	2	[44, 45]
	Clustering hierarchical	1	[39]
	Multi Objective Evolutionary Fuzzy Classifier	1	[13]
	Fuzzy C-means Clustering	1	[82]
Dimensional reduction	*Association rules*	2	[41, 100]
	Principal Component Analysis	1	[49]
	Shatter Dempster theory	1	[66]

En la Figura 3.5 se muestran las frecuencias con que se reportaron los nueve paradigmas en los 87 estudios analizados. Por medio de esta figura se puede ver claramente la diferencia entre los paradigmas *decision tree* y *linear predictors*, con respecto a los demás.

El paradigma *decision tree* se basa en la estrategia de "divide y vencerás" aplicado sobre una serie de instancias relacionadas. Cada nodo del árbol representa un atributo diferente de clasificación. La iteración del algoritmo se realiza por medio de una comparación entre dos nodos y decidiendo hacia donde se puede clasificar mejor la nueva instancia. Este proceso se realiza a profundidad de recorrido según el tamaño del árbol de decisión que se esté trabajando. Un caso importante de aclarar es que si el dato por clasificar es nulo, se crea una tercera rama de clasificación. Este modelo es más aplicable con atributos nominales [1].

Con respecto al paradigma *linear predictors* es más utilizado con atributos numéricos. El problema por resolver se plantea mediante una combinación de los atribu-

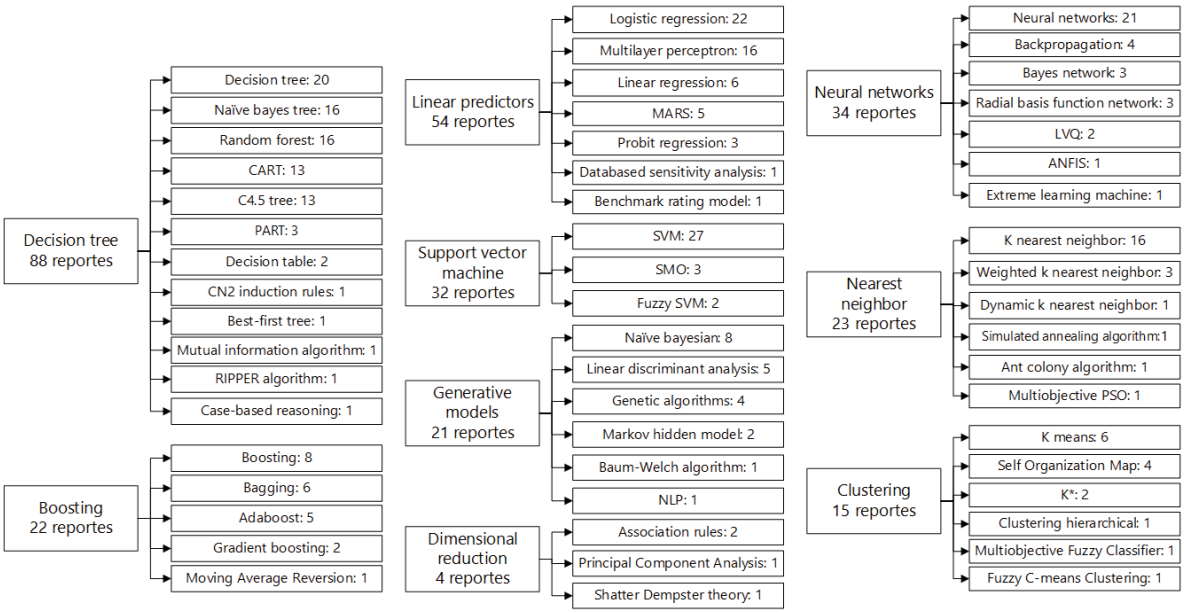


Figura 3.5: Cantidad de las técnicas de minería de datos y aprendizaje automático clasificadas en paradigmas.

tos involucrados y deben pasar por una etapa de entrenamiento para que el modelo pueda aprender las características de los datos. La segunda etapa es la validación mediante otro conjunto de datos. Esto se aplica para que el modelo pueda clasificar con mayor precisión los datos nuevos que entren [1].

Como parte de los datos extraídos, se obtuvo el objetivo para el que se aplicó cada estudio analizado. En total se extrajeron seis objetivos generales: Puntuación de clientes, Gestión de riesgo de los clientes, Campañas de mercadeo bancario, *Churn* de clientes, Toma de decisiones y Predicción de clientes potenciales. En la Figura 3.6 se muestra la relación y frecuencia entre los paradigmas y los objetivos extraídos. Los dos objetivos que más se reportaron fueron Puntuación de clientes y Gestión de riesgo de los clientes. Además de que fueron los únicos que se utilizaron en todos los paradigmas. Cabe destacar que todos los objetivos que se extrajeron son relacionados a aplicación de segmentación o conocimiento de clientes para identificar los diferentes casos de estudio.

La configuración de las técnicas se refiere a su implementación, no todos los autores reportan dicha información; sería importante que los estudios primarios reporten estos datos para que puedan ser replicados, y sus resultados validados. En los estudios

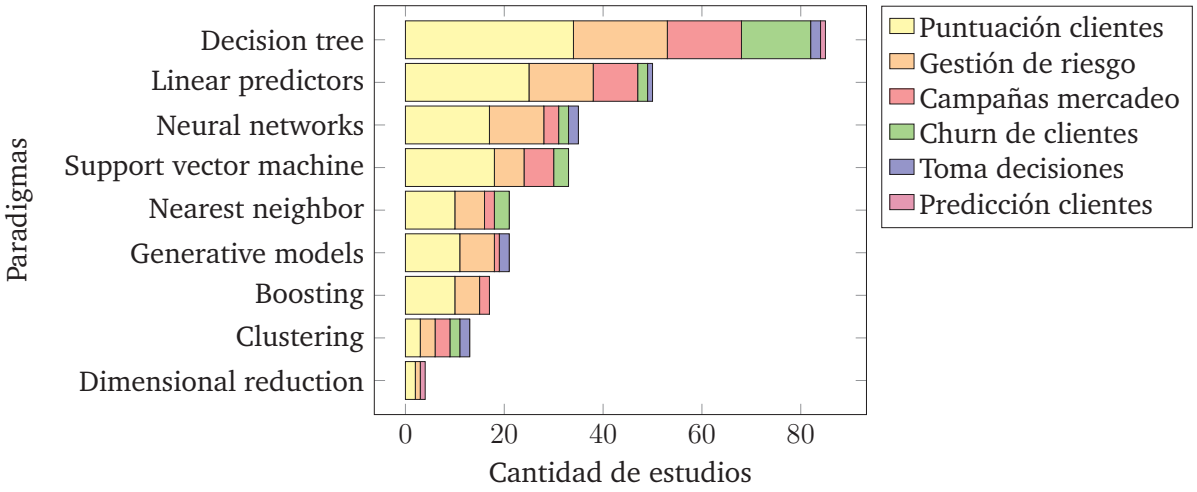


Figura 3.6: Paradigmas según el objetivo de los estudios analizados.

[18, 61, 47, 92, 14, 79, 97, 64, 98, 15, 86, 87, 40, 21, 74, 49, 56, 41, 27, 88, 100, 65, 99, 66, 89, 67, 58, 68, 69, 52, 90, 32, 53, 83, 77, 16, 60, 78, 85, 71, 63, 82], sí se reportan el detalle de la configuración utilizada. Además, es importante mencionar que todos los paradigmas han tenido una frecuencia de uso constante en el tiempo, es decir, a partir del 2005 y hasta el 2019 han sido estudiados en el contexto planteado. Esto se puede revisar mejor mediante la Figura 3.7, donde se presenta en el eje vertical los paradigmas y en el eje horizontal los años en que se reportaron los estudios.

3.6.2. Herramientas que soportan la implementación de técnicas de minería de datos y aprendizaje automático (RQ2)

La segunda pregunta de investigación corresponde a la identificación de las herramientas que soportan técnicas de minería de datos y aprendizaje automático. Se identificó un total de 22 herramientas.

En la Figura 3.8 se evidencian las herramientas que fueron reportadas más de una vez y su frecuencia. Como se ve en la figura, la herramienta más utilizada fue Weka [4] con un total de 13 estudios que la reportaron. Esta herramienta posee las técnicas ya implementadas, por lo que únicamente se debe configurar el conjunto de datos

[4]www.cs.waikato.ac.nz/ ml/weka/index.html

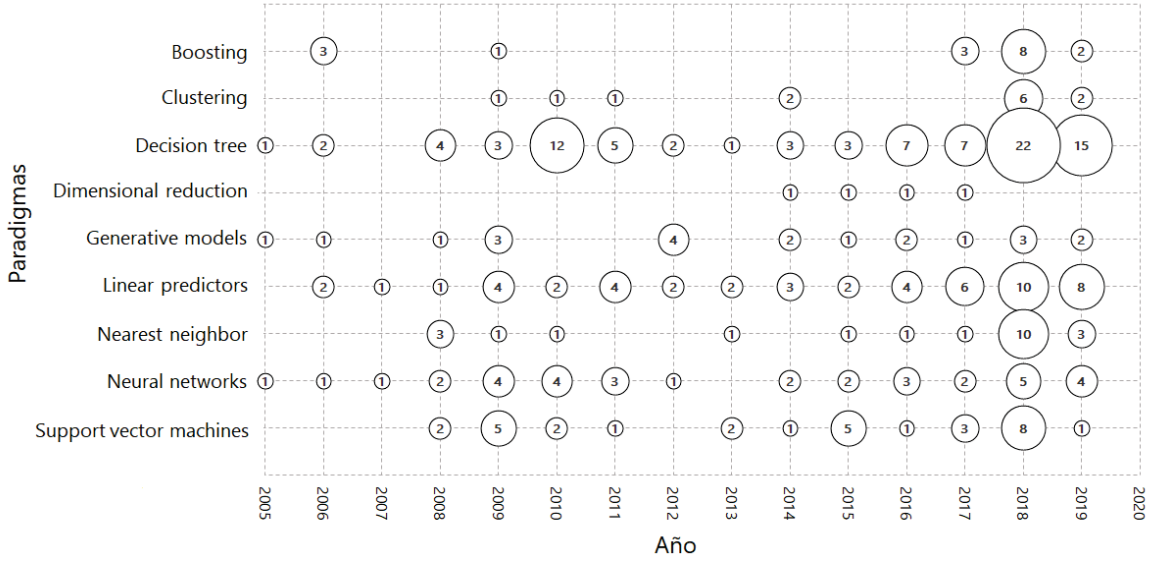


Figura 3.7: Cantidad de los paradigmas de minería de datos y aprendizaje automático reportados por año.

de entrada, las variables predictoras y las estadísticas que se quieran mostrar. Weka tiene implementadas técnicas de los paradigmas: *decision tree*, *linear predictors*, *clustering*, *support vector machines*, *boosting*, *generative models*, *nearest neighbor* y *neural networks*. También, permite la conexión a diversas bases de datos por medio de un componente Java o por medio de archivos de texto con una extensión específica para este *software*. Además, posee una biblioteca con conjuntos de datos de acceso público que contienen diferentes tipos de datos específicos para ciertos problemas. Asimismo, contiene una capa de preparación de datos previo a la entrada del modelo que se implemente, donde se pueden hacer filtros, normalizaciones de datos, transformación y combinación de atributos. Finalmente, tiene una capa gráfica donde se muestran diferentes esquemas para la comparación de los datos de salida de forma básica [1].

La siguiente herramienta más reportada fue Matlab [5] con 12 referencias. Esta herramienta está orientada a la implementación de algoritmos, al modelado, simulación y prototipo de datos. Permite análisis, exploración y visualización de resultados. El lenguaje de programación está basado en una matriz de alto nivel, donde se maneja un control de estados y las estructuras de datos. Además, contiene una colección de

[5] www.mathworks.com/products/matlab.html

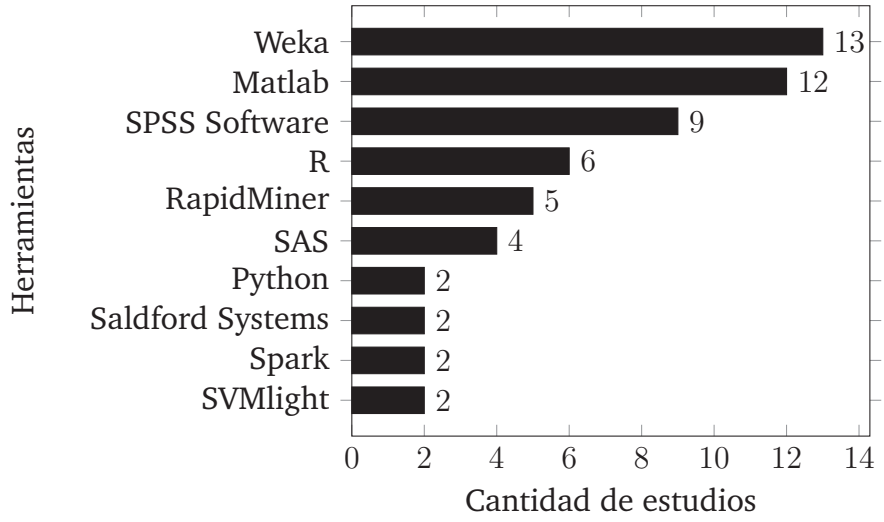


Figura 3.8: Cantidad de herramientas que soportan la implementación de técnicas de minería de datos y aprendizaje automático.

funciones ya creadas para facilitar la implementación de los modelos. Por la naturaleza de la herramienta, es posible construir técnicas de todos los paradigmas. Matlab es más utilizada en investigaciones y a nivel educativo por la complejidad técnica que se requiere al implementar en ella, no es tan fácil de aplicar en entornos productivos del día a día [101].

La Figura 3.9 muestra la relación entre las herramientas reportadas más de una vez (eje vertical) y los paradigmas en los que se clasificaron las técnicas de minería de datos y aprendizaje automático (eje horizontal). El paradigma *decision tree* fue el único que se utilizó en todas las herramientas. Por otro lado, la herramienta Matlab fue la única en la que se implementaron las técnicas de todos los paradigmas. Esto es importante porque por la naturaleza de Matlab, se puede desarrollar cualquier técnica creando el código fuente necesario. Caso contrario de Weka, que no contiene entre las técnicas por utilizar las que se encuentran dentro del paradigma *Dimensional reduction*. Esto es una limitación de la herramienta ya que restringe a la persona que vaya a utilizarla y las técnicas y configuraciones ya establecidas, sin posibilidad a adicionar más según sea la necesidad del experimento.

El Cuadro 3.3 expone las 22 herramientas reportadas en los estudios. Este cuadro contiene el nombre de la herramienta, la frecuencia con que se referencia en los estudios, el número de referencia y una descripción general. Además, es importante

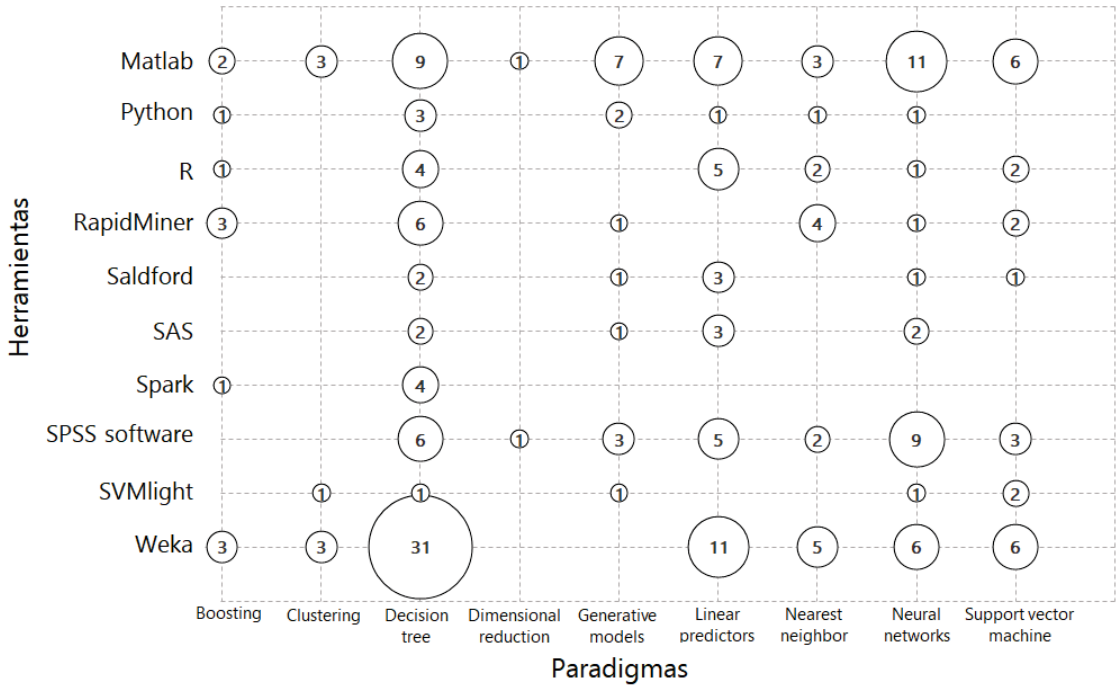


Figura 3.9: Relación entre herramientas y paradigmas.

resaltar que de 87 estudios analizados, 36 no reportaron las herramientas que utilizaron durante los experimentos. Los estudios que no reportaron el detalle de las herramientas utilizadas son: [79, 64, 84, 98, 51, 95, 86, 87, 21, 22, 24, 49, 72, 56, 42, 88, 58, 80, 75, 28, 69, 30, 91, 76, 54, 83, 70, 77, 33, 34, 59, 96, 60, 35, 71, 36].

Las tendencias de uso de las herramientas reportadas son constantes a través del tiempo, desde el año 2006 hasta el año 2019 y no muestran una tendencia clara como se puede ver en la Figura 3.10. Sin embargo, se puede ver que herramientas como Saldford y SVMlight se reportaron en los años iniciales, pero después del 2010 ya no se muestran reportes. Si más autores reportaran en sus estudios las herramientas que utilizan, se podría hacer una mejor comparación de tendencias de uso a través de los años.

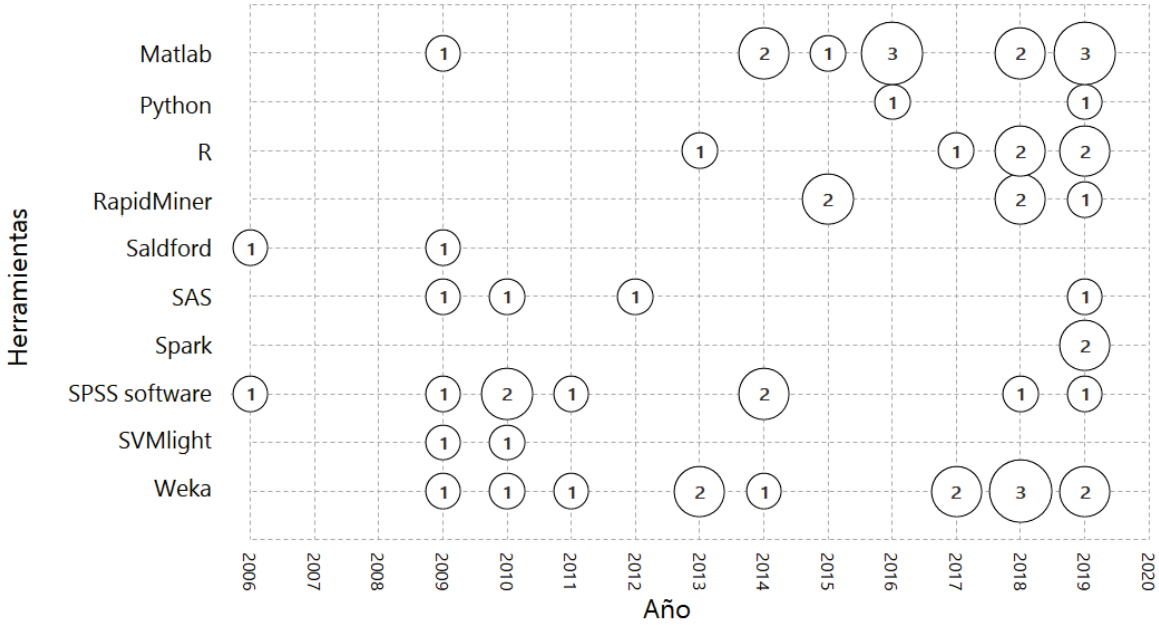


Figura 3.10: Cantidad de herramientas que soportan la implementación de técnicas de minería de datos y aprendizaje automático reportadas por año.

Cuadro 3.3: Herramientas que soportan la implementación de técnicas de minería de datos y aprendizaje automático.

Herramienta	Cant.	Estudios	Descripción
Weka	13	[13, 14, 38, 15, 23, 48, 50, 44, 57, 67, 62, 45, 32]	Es una plataforma implementada en Java que ofrece una gran variedad de algoritmos de minería de datos y aprendizaje automático. Ofrece una interfaz de usuario de fácil acceso a las funcionalidades, contiene técnicas de preprocesamiento de datos y modelado.

Continúa en la página siguiente.

Herramienta	Cant.	Estudios	Descripción
Matlab	12	[97, 20, 40, 25, 26, 73, 27, 99, 66, 78, 85, 82]	Es una herramienta que brinda la posibilidad de hacer análisis predictivo y procesos de diseño con un lenguaje de programación expresado en funciones matemáticas.
SPSS software	9	[93, 65, 66, 29, 52, 31, 55, 78, 85]	Esta plataforma ofrece análisis estadístico avanzado, una biblioteca con algoritmos de aprendizaje automático, integración con *big data*, extensión de código abierto y fácil despliegue de las aplicaciones.
R	6	[61, 47, 15, 94, 74, 89]	Es un lenguaje computacional enfocado en la estadística. Es popular en la investigación científica, principalmente en las áreas de aprendizaje automático, minería de datos, matemáticas financieras y bioinformática. Tiene una gran cantidad de bibliotecas y funcionalidades de cálculo y graficación.
RapidMiner	5	[92, 15, 43, 16, 37]	Es una herramienta para el análisis y minería de datos. Permite desarrollar procesos de análisis de datos por medio de una interfaz gráfica de encadenamiento de operadores. Tiene una integración con la herramienta Weka para poder utilizar los algoritmos que ya tiene implementados.

Continúa en la página siguiente.

Herramienta	Cant.	Estudios	Descripción
SAS	4	[19, 68, 52, 81]	Es un programa analítico de alto rendimiento que permite la implementación de análisis de datos. Además, facilita la gestión de datos de forma estándar. Posee la capacidad de generar código o de utilizar los modelos ya implementados en la herramienta.
Python	2	[18, 25]	Es un lenguaje de programación multiparadigma. Ofrece la posibilidad de integrarse con diversas herramientas y posee una gran biblioteca de funciones para ser utilizadas en analítica.
Salford Systems	2	[53, 55]	Es un modelador predictivo que incluye el desarrollo de técnicas como CART, MARS, *TreeNet* y *Random Forest*. Tiene posibilidades de automatización y modelado de datos.
Spark	2	[46, 18]	Es un entorno de computación por cluster. Ofrece la capacidad de analítica unificada por medio del procesamiento de grandes volúmenes de datos.
SVMlight	2	[31, 91]	Es una herramienta que implementa la técnica *Support vector machine*. Ofrece una rápida optimización del algoritmo, resuelve problemas de clasificación y regresión. Estima métricas de rendimiento como *error rate*, *precision* y *recall*.

Continúa en la página siguiente.

Herramienta	Cant.	Estudios	Descripción
Knime	1	[15]	Es una plataforma de minería de datos que permite el desarrollo de modelos. Está desarrollada en Java. Ofrece la manipulación de datos, visualización, creación de modelos estadísticos y minería de datos, validación de modelos y curvas ROC.
Microsoft Excel	1	[63]	Es una herramienta por medio de hojas de cálculo que ofrece gráficas, tablas dinámicas y lenguaje de programación básico.
Palisade	1	[63]	Es una plataforma se ha utilizado para el análisis de riesgo y la toma de decisiones. Tiene integración con la herramienta Microsoft Excel.
H2O	1	[46]	Es una plataforma que ofrece técnicas de aprendizaje automático e inteligencia artificial. Está enfocado en la automatización de la ciencia de datos por medio de los colaboradores de código abierto.
Powerhouse	1	[61]	Es una herramienta que ofrece el proceso de explorar datos por medio de modelos de minería de datos. Además, consta de una serie de pasos como selección de los datos para trabajar, exploración, preparación y selección de las variables y selección del algoritmo para crear el modelo.

Continúa en la página siguiente.

Herramienta	Cant.	Estudios	Descripción
Orange	1	[39]	Es una herramienta para minería de datos y análisis predictivo. Consta de componentes que implementan los algoritmos, operaciones de preprocesamiento y representación gráfica de los datos.
Visual C++	1	[23]	Es un lenguaje de programación multiparadigma. Es abierto para cualquier tipo de desarrollos con análisis de datos.
KEEL	1	[57]	Es una herramienta implementada en Java que ofrece una interfaz para diseñar el proceso completo del descubrimiento de datos. Ofrece el preprocesamiento de datos y una gran variedad de técnicas de minería de datos y aprendizaje automático ya implementadas.
Java	1	[100]	Es un lenguaje de programación y plataforma informática. Es abierto para cualquier tipo de desarrollos con análisis de datos.
Lingo	1	[31]	Es una solución diseñada y construida para resolver modelos de optimización lineal de una manera más rápida, fácil y eficiente. Es un paquete que incluye un lenguaje que proporciona soluciones integradas rápidas.

Continúa en la página siguiente.

Herramienta	Cant.	Estudios	Descripción
Pathfinder	1	[39]	Es una plataforma de inteligencia comercial. Está orientada al manejo de datos minorista.
MapReduce framework	1	[41]	Es un modelo de programación para dar soporte a la computación paralela sobre grandes colecciones de datos.

Es importante destacar que, de las 22 herramientas identificadas, 11 son de licencia gratuita. Además, es importante aclarar que estas herramientas se dividen entre las que ya tienen las técnicas de minería de datos y aprendizaje automático desarrolladas y las que el investigador necesita implementar el algoritmo. Las herramientas que ya cuentan con las técnicas implementadas son: Weka, SPSS software, RapidMiner, SAS, Saldford system, Knime, Palisade, H2O, Powerhouse, KEEL, Pathfinder y SVM-light. Las demás herramientas a pesar de que utilizan bibliotecas que ya contienen funciones que ayudan a la implementación de las técnicas, necesitan de un esfuerzo técnico adicional para poder desarrollar los modelos.

3.6.3. Conjuntos de datos y métricas de evaluación usados para las técnicas de minería de datos y aprendizaje automático (RQ3)

La tercera pregunta de investigación ayuda a identificar cuáles son los tipos de conjuntos de datos más utilizados y sus características en el contexto planteado. También, esta pregunta permite conocer cuáles son las métricas de evaluación más usadas para medir el rendimiento de las técnicas de minería de datos y aprendizaje automático durante los experimentos de cada estudio analizado.

De los conjuntos de datos utilizados para evaluar las técnicas de minería de datos y aprendizaje automático se pudo notar una tendencia de uso en los repositorios de datos. En el Cuadro 3.4 se muestran los conjuntos de datos reportados en los estudios primarios. Este cuadro contiene el nombre del conjunto, la cantidad de registros, la

cantidad de veces que se referenció y el número de referencia. Los estudios [46, 64, 94, 100, 65, 89, 67, 62, 91, 76, 96, 78, 85, 37] no reportan nombre o mayor detalle del conjunto de datos que utilizan. Esta información es útil ya que, en el caso de emplear repositorios públicos, el detalle del tratamiento del conjunto de datos para el experimento, permite a otros investigadores la posibilidad de replicar el trabajo con más fidelidad para validar resultados.

Cuadro 3.4: Conjuntos de datos reportados en los estudios primarios.

Conjunto de datos	Registros	Cant.	Estudios
German UCI Machine Learning [6]	1000	25	[38, 51, 15, 20, 87, 40, 23, 24, 49, 50, 72, 26, 42, 88, 99, 44, 57, 58, 28, 69, 31, 70, 60, 63, 36]
Australian UCI Machine Learning [7]	690	14	[38, 20, 40, 74, 49, 72, 26, 88, 99, 44, 57, 70, 60, 63]
Bank Marketing UCI Machine Learning [8]	45211	10	[18, 61, 86, 22, 48, 56, 41, 27, 83, 16]
Taiwan UCI Machine Learning [9]	30000	5	[21, 23, 25, 54, 82]
Japanese UCI Machine Learning [10]	690	4	[38, 49, 72, 26]
Bank of China	725	3	[49, 72, 53]
Chinese commercial bank	5456	3	[75, 45, 77]
SPSS credit modeling UCI Machine Learning	700	2	[72, 63]
Commercial bank in Chinese	72544	2	[29, 30]

Continúa en la página siguiente.

[6] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data
[7] http://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval
[8] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[9] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
[10] https://archive.ics.uci.edu/ml/datasets/Japanese+Credit+Screening

Conjunto de datos	Registros	Cant.	Estudios
Croatian bank	1000	2	[80, 81]
European bank	2600000	2	[79, 33]
SME commercial corporate credit scoring	103	1	[13]
Survey of Consumer Finances	8358	1	[47]
BPR Bank Jepara Artha	240	1	[92]
Atlanticus Services Corporation	12495	1	[19]
Turkish Statistical Institution [11]	59663	1	[14]
Medium size commercial bank	690	1	[97]
Superintendency of Popular and Solidarity Economy in Ecuador	20000000	1	[84]
Bank Saderat Iran	2100	1	[73]
Bank of Iran	1100	1	[43]
Commercial bank in Shenzhen	4000	1	[66]
Brazilian bank	4504	1	[68]
Credit Card Centre Department of the bank	4305	1	[52]
Modeling Data	50000	1	[32]
Bank in Taipei	8000	1	[55]
Credit Card Data Warehouse	200	1	[34]

Continúa en la página siguiente.

[11] http://ghdx.healthdata.org/organizations/turkish-statistical-institute

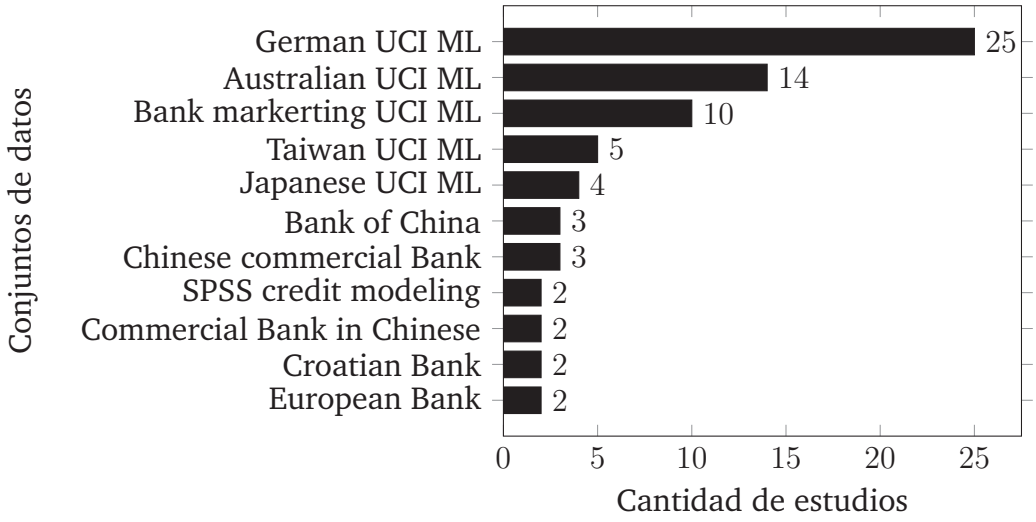


Figura 3.11: Cantidad de los conjuntos de datos reportados en los estudios primarios.

Conjunto de datos	Registros	Cant.	Estudios
Tunisian bank	603	1	[71]
Banks in Bulgaria	100	1	[93]
Public bank in Bandung	500	1	[98]
Bank of Indonesia	948	1	[95]
Pathfinder platform	200000	1	[39]

Como se ve en la Figura 3.11, 60 de las referencias son tomadas del repositorio de datos *UCI Machine Learning* de la Universidad de California Irvine [12], que es público y contiene datos de diversos tipos, entre ellos datos financieros de clientes para créditos o mercadeo. Para este repositorio existen datos por países, por lo que algunos estudios eligieron varios conjuntos de datos de diferentes países y así hacer comparaciones entre técnicas, tomando en cuenta las diferencias entre conjuntos. De este repositorio se reportaron 6 variaciones. Adicionalmente, se registraron 25 conjuntos de datos privados.

[12]https://archive.ics.uci.edu/ml/index.php

La mayoría de los estudios analizados, documentaron los atributos de estos conjuntos, los cuales se dividen entre numéricos y nominales. Los numéricos se refieren a las variables que son medibles y cuantificables. Estos atributos también pueden ser llamados continuos. En el caso de los atributos nominales son los que adquieren valores en un conjunto finito predefinido de posibilidades y pueden ser llamados categóricos. Tanto los atributos numéricos como nominales tienen la capacidad de funcionar como variables predictoras. Sin embargo, son más comunes como predictores los numéricos, ya que eliminan el sesgo que se puede dar entre los rangos de parámetros que defina un atributo nominal. Los atributos nominales son más útiles para hacer distinción entre las características de los patrones que marque el modelo [1].

En la Figura 3.12 se observan las 21 variables más reportadas en los conjuntos de datos extraídos. Como se puede ver, los cinco atributos más importantes son: la edad, el trabajo, el género, la temporalidad y las características crediticias. En el caso del atributo de edad puede ser tanto numérico como nominal, ya que algunos conjuntos de datos lo pueden colocar como un número entero o identificar por rangos ya establecidos. Para los otros cuatro atributos son nominales ya que pueden tomar valores de una lista de estándares definidos por cada conjunto. En el Cuadro 3.5 se muestran los atributos extraídos de los conjuntos de datos, la cantidad de veces que se referencian y las referencias correspondientes.

Cuadro 3.5: Atributos de los conjuntos de datos reportados en los estudios primarios.

Atributos	Cant.	Estudios
Edad	58	[18, 61, 92, 14, 79, 97, 38, 98, 51, 15, 20, 95, 39, 86, 87, 40, 22, 23, 48, 24, 49, 50, 72, 56, 25, 26, 41, 42, 73, 27, 88, 99, 44, 66, 57, 58, 68, 29, 28, 69, 30, 52, 90, 31, 32, 81, 53, 54, 83, 70, 77, 55, 16, 59, 60, 63, 36, 82]

Continúa en la página siguiente.

Atributos	Cant.	Estudios
Trabajo	49	[18, 61, 14, 38, 51, 15, 20, 95, 86, 87, 40, 22, 23, 48, 24, 49, 50, 72, 56, 26, 41, 42, 73, 27, 88, 99, 44, 66, 57, 58, 29, 28, 69, 30, 52, 90, 31, 32, 81, 53, 83, 70, 77, 55, 16, 59, 60, 63, 36]
Género	43	[14, 97, 38, 51, 15, 20, 39, 87, 40, 21, 23, 24, 49, 50, 72, 25, 26, 42, 73, 88, 99, 44, 57, 58, 68, 29, 28, 69, 30, 52, 90, 31, 32, 53, 54, 70, 77, 55, 59, 60, 63, 36, 82]
Temporalidad	40	[18, 61, 38, 51, 15, 20, 86, 87, 40, 21, 22, 23, 48, 24, 49, 50, 72, 56, 25, 26, 41, 42, 27, 88, 99, 44, 57, 58, 28, 69, 31, 54, 83, 70, 16, 59, 60, 63, 36, 82]
Créditos	32	[92, 19, 38, 98, 51, 15, 20, 87, 40, 23, 24, 49, 50, 72, 26, 42, 88, 99, 44, 57, 58, 28, 69, 31, 53, 70, 77, 55, 59, 60, 63, 36]
Estado marital	31	[18, 61, 14, 97, 38, 98, 86, 22, 23, 48, 49, 72, 56, 25, 26, 41, 73, 27, 66, 68, 29, 52, 90, 32, 81, 54, 83, 77, 55, 16, 82]
Débitos	30	[13, 38, 51, 15, 20, 95, 87, 40, 23, 24, 49, 50, 72, 26, 42, 88, 99, 44, 57, 58, 28, 69, 90, 31, 70, 77, 59, 60, 63, 36]
Teléfono	30	[38, 51, 15, 20, 87, 40, 23, 24, 49, 50, 72, 26, 42, 88, 99, 44, 57, 58, 28, 69, 31, 32, 81, 53, 70, 77, 59, 60, 63, 36]
Educación	25	[18, 61, 14, 79, 97, 86, 21, 22, 23, 48, 56, 25, 41, 73, 27, 66, 29, 90, 53, 54, 83, 77, 55, 16, 82]
Ingresos	21	[47, 92, 14, 79, 84, 95, 39, 21, 23, 25, 66, 29, 30, 90, 32, 81, 53, 54, 77, 55, 82]

Continúa en la página siguiente.

Atributos	Cant.	Estudios
Vivienda	19	[18, 61, 98, 39, 86, 22, 48, 56, 41, 27, 68, 30, 52, 90, 32, 81, 83, 55, 16]
Préstamos	14	[18, 61, 93, 95, 86, 22, 48, 56, 41, 27, 66, 52, 83, 16]
Duración	11	[18, 61, 86, 22, 48, 56, 41, 27, 52, 83, 16]
Contacto	10	[18, 61, 86, 22, 48, 56, 41, 27, 83, 16]
Campaña	10	[18, 61, 86, 22, 48, 56, 41, 27, 83, 16]
Familia	9	[47, 92, 98, 39, 29, 30, 90, 81, 53]
Balance financiero	8	[19, 21, 23, 25, 75, 45, 54, 82]
Demográficas	7	[47, 14, 32, 81, 53, 77, 55]
Salud	2	[13, 14]
Puntuación crediticia	2	[43, 34]
Tasa de interés	1	[84]

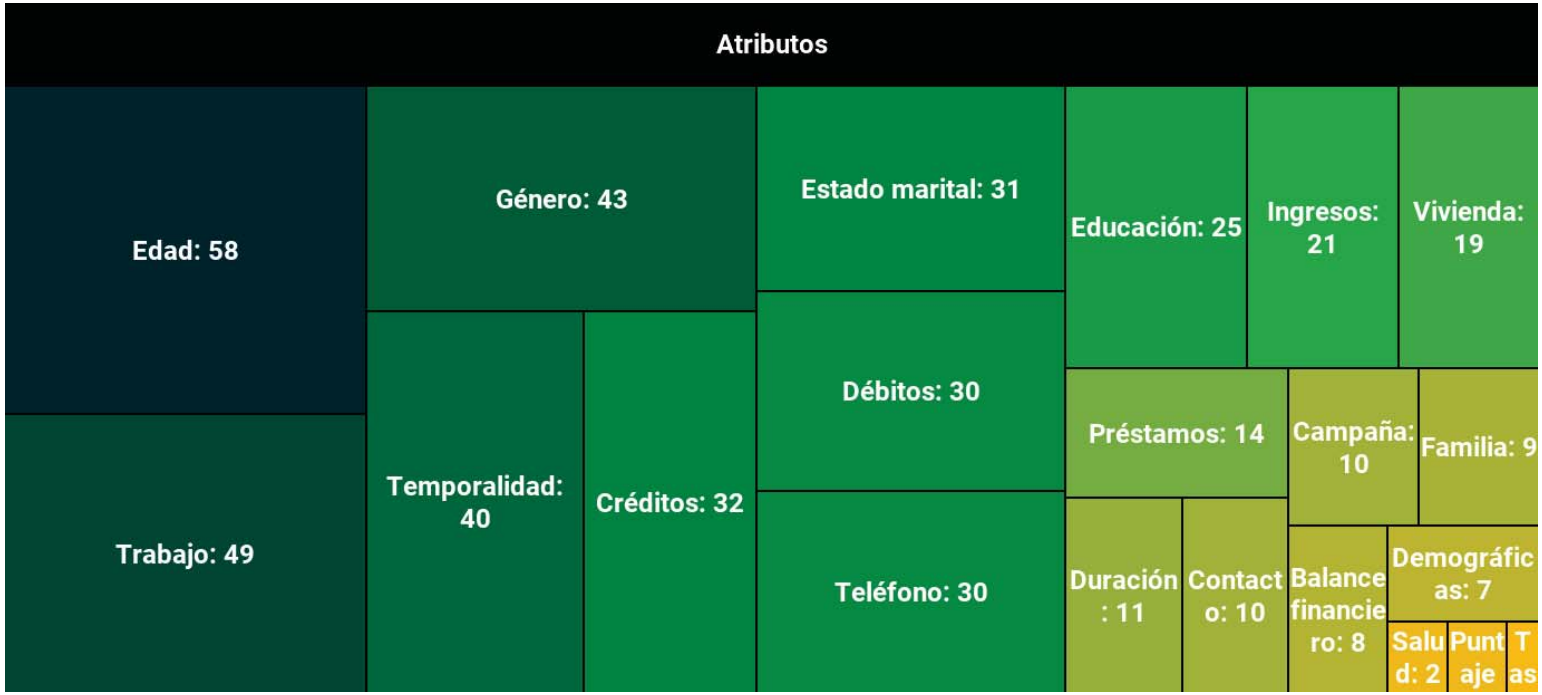


Figura 3.12: Cantidad de veces que los atributos fueron reportados.

Con respecto a las métricas de evaluación de rendimiento, de los estudios analizados, siete ([93, 39, 73, 100, 57, 62, 45]) no reportaron métodos de medición de las técnicas utilizadas. La métrica más reportada fue *accuracy*, ya que se evaluó en 66 estudios, aproximadamente el 78 % de los analizados. La sigue *recall* con 24 y *specificity* con 19. El Cuadro 3.6 muestra las métricas de rendimiento más reportadas en los estudios analizados. Este cuadro contiene el nombre de la métrica, la cantidad de veces que se referenció y el número de referencia.

Cuadro 3.6: Métricas de evaluación reportadas en los estudios primarios.

Métrica	Cant.	Estudios
Accuracy	66	[13, 61, 47, 92, 19, 14, 97, 64, 84, 38, 98, 51, 15, 20, 94, 95, 86, 40, 21, 22, 23, 48, 74, 49, 50, 72, 56, 25, 26, 42, 27, 88, 43, 65, 99, 66, 89, 67, 58, 80, 68, 29, 75, 28, 30, 52, 90, 31, 91, 81, 53, 83, 70, 77, 33, 55, 34, 16, 59, 96, 78, 35, 63, 36, 37, 82]
Recall	24	[46, 13, 18, 61, 47, 92, 14, 38, 15, 95, 86, 22, 23, 74, 25, 44, 67, 68, 31, 16, 85, 71, 37, 82]
Specificity	19	[61, 47, 38, 15, 95, 23, 74, 25, 41, 89, 67, 68, 31, 70, 55, 16, 60, 71, 82]
ROC curve	16	[13, 47, 19, 14, 79, 22, 48, 24, 50, 56, 26, 27, 43, 67, 32, 54]
Error rate	15	[15, 94, 40, 21, 74, 65, 89, 69, 53, 54, 70, 34, 16, 78, 85]
Precision	15	[46, 13, 18, 92, 14, 84, 15, 21, 22, 23, 25, 44, 28, 37, 82]
F-measure	14	[13, 18, 47, 14, 38, 15, 87, 22, 23, 56, 25, 26, 44, 82]
False positive rate	6	[22, 41, 70, 55, 16, 60]

Continúa en la página siguiente.

	Predictivo	
	Positivo	**Negativo**
Positivo	TP	FN
Negativo	FP	TN

Figura 3.13: Matriz de confusión para métricas de evaluación.

Métrica	Cant.	Estudios
True positive rate	5	[41, 89, 70, 16, 60]
False negative rate	4	[41, 70, 16, 60]

Una tendencia que se notó en el 60 % de los artículos primarios es que no reportan la fórmula de la métrica. Sin embargo, varios de ellos sí reportan la descripción teórica y el nombre, por lo que tomando como referencia a Wirten et al. [1], se pudo identificar la métrica y su fórmula. Todas las métricas de rendimiento que evalúan las técnicas de minería de datos y aprendizaje automático son basadas en la matriz de confusión [6] que se presenta en la Figura 3.13. Los términos verdadero positivo (TP por sus siglas en inglés), verdadero negativo (TN por sus siglas en inglés), falso positivo (FP) y falso negativo (FN) son los que conforman esta matriz.

Algunos estudios también reportaron métricas estadísticas para sacar desviaciones, estimaciones o relaciones entre las variables del experimento planteado. Las métricas estadísticas reportadas fueron: *RMSE criterions* con cinco referencias ([14, 94, 22, 50, 85]), *kappa statistics* con cuatro referencias ([13, 61, 22, 16]), *correlation* ([94, 87]) y *ratio information gain* ([87, 80]) con dos referencias cada una y con una referencia *top-decile rate* ([35]), *cost recovery* ([27]), *KS value* ([31]) y *potency* ([76]).

Posterior a obtener los resultados, se realizó una comparación y en la Figura 3.14 se conoce la relación entre las métricas de evaluación reportadas (eje horizontal) y los paradigmas en los que se clasificaron las técnicas de minería de datos y aprendizaje automático (eje vertical), con el fin de saber si existen relaciones exclusivas de uso

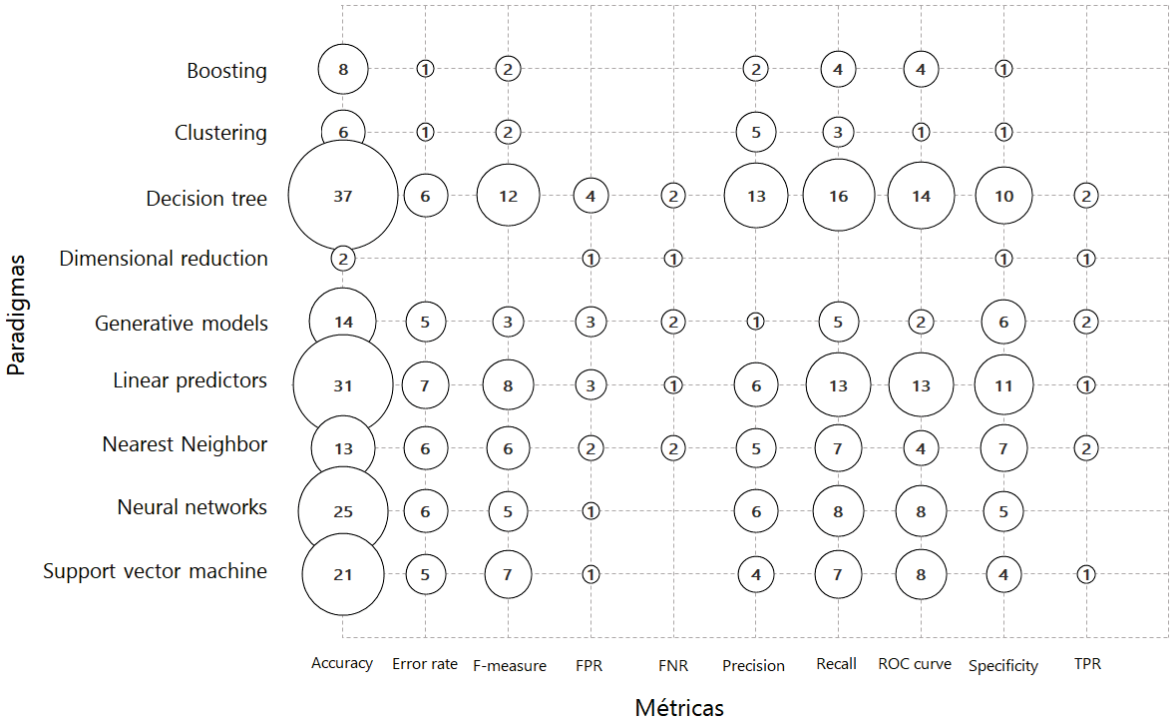


Figura 3.14: Relación entre paradigmas y métricas.

entre métricas y técnicas. En este caso como ya se había visto con la obtención de resultados, la métrica *accuracy* fue la más reportada y la única que se utilizó para evaluar todos los paradigmas. En cuanto a los paradigmas *decision tree*,*generative models*, *linear predictors* y *nearest neighbor* fueron evaluados por todas las métricas reportadas.

Todos los artículos que reportaron las métricas, mostraron los resultados que obtuvieron. Para esta investigación no fue posible hacer una comparativa completa dado que los artículos deberían presentar las mismas condiciones en cuanto a las técnicas de minería de datos y aprendizaje automático, el conjunto de datos, el tratamiento de los datos previo a la evaluación, las variables utilizadas, las herramientas, entre otros. Sin embargo, se hizo una comparación básica tomando en cuenta los estudios que evaluaron la métrica *accuracy* y que usaron el conjunto de datos *German UCI Machine Learning*. Los artículos que se utilizaron en la comparación fueron: [38, 51, 20, 40, 23, 49, 50, 72, 26, 42, 99, 58, 28, 31, 70, 59, 63, 36]. Como se puede ver en la Figura 3.15 el paradigma que obtuvo mejores resultados fue *Generative*

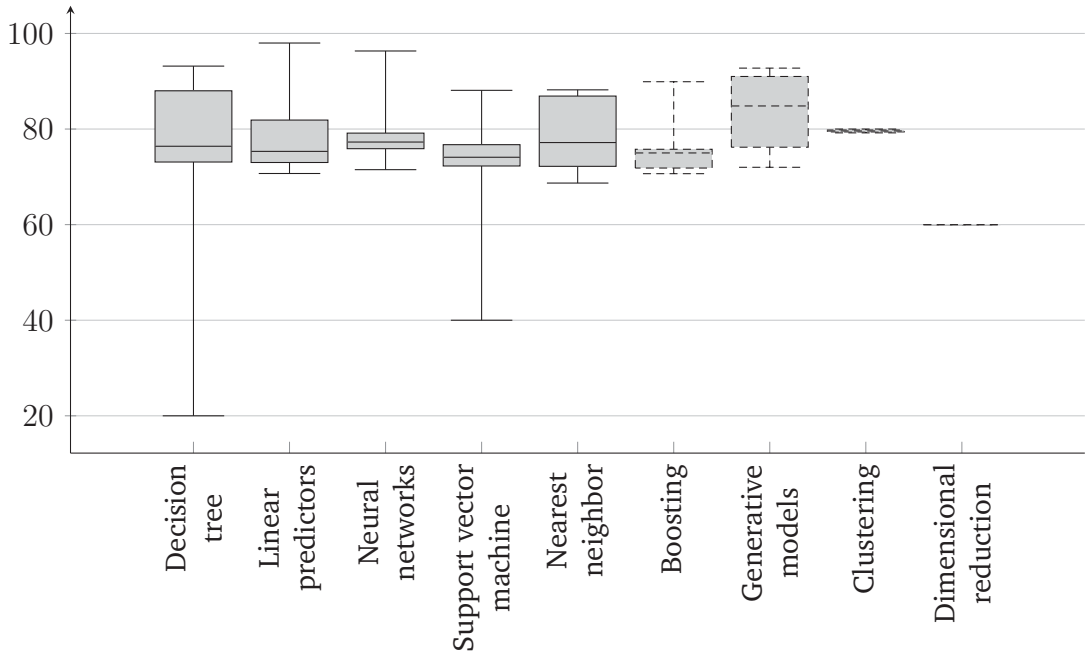


Figura 3.15: Comparación de paradigmas según el resultado de la métrica *accuracy* y el conjunto de datos *German UCI Machine Learning*.

Models que registró una mediana de 84.83 % y el que obtuvo peor resultado para la métrica fue *Dimensional Reduction* con una mediana de 60 %. Para esta comparación debe considerarse que todos los paradigmas tuvieron diferente cantidad de muestras.

3.7. Discusión

El análisis de datos es una necesidad planteada de hace ya varios años por diferentes áreas. Tecnologías como la minería de datos y el aprendizaje automático entraron a los mercados con la idea de automatizar la toma de decisiones a partir de la información que generen los datos. En el caso de la industria bancaria, el análisis de datos puede tomar diversas aristas, uno de ellos es el conocimiento del cliente y sus necesidades.

Mediante el análisis de resultados realizado, se observó que existen diferentes técnicas de minería de datos y aprendizaje automático que pueden ser aplicadas en el problema de segmentación de clientes bancarios. Estos resultados revelaron una ten-

dencia a utilizar más ciertas técnicas sobre otras. No obstante, esto no demuestra que las técnicas más aplicadas son mejores o peores para dar solución a la temática planteada, ya que para hacer una comparación de este tipo es necesario contar con un ambiente con las mismas condiciones, es decir, se debe trabajar con un conjunto que lleve igual tratamiento de datos, utilizar la misma herramienta para la implementación del experimento, evaluar las técnicas con las mismas métricas de rendimiento, entre otros.

Por medio de esta investigación se evidenció que existen diversas herramientas que soportan implementaciones de este tipo de tecnologías, poseen características y brindan diferentes posibilidades al usuario, por lo que la elección de cuál herramienta se usará depende del estudio que se esté realizando. Dentro del grupo de herramientas identificadas, se puedo valorar que existen algunas más adecuadas para ser utilizadas en la academia y otras a nivel profesional.

Con respecto a los conjuntos de datos obtenidos en la investigación, no se pudo establecer una estandarización de los tratamientos de los datos, ya que los estudios primarios brindaban poco detalle. Esto debe mejorar en el reporte de los estudios primarios. Adicionalmente, se pudo extraer variables predictoras que se repitieron en los diferentes conjuntos de datos. En el caso de la métrica de evaluación se detectó que la mayoría de métricas reportadas son estándar y poseen poca variación, ya que están planteadas mediante la matriz de confusión y tienen como objetivo evaluar diferentes parámetros de los modelos de minería de datos y aprendizaje automático aplicados.

3.8. Lecciones aprendidas

Para el desarrollo de un mapeo sistemático de literatura como éste, lo importante es tener clara y delimitada la metodología que se está aplicando. Aprender sobre este tipo de metodologías permite crear investigaciones sólidas, ya que identifica los estudios primarios que se estén trabajando en determinado contexto.

La aplicación de este tipo de metodologías permite tener una visión actualizada de los temas más recientes que se estén investigando en el campo de la Ingeniería de *software*. Además, la secuencia de lineamientos y procedimientos que se utiliza

permite ser replicada en otros estudios similares que quieran validar resultados.

Estas metodologías al tener procedimientos definidos, permiten llevar un proceso estructurado donde puede ser validado en cada una de las etapas. La iteración continua hace que el modelo e investigación se fortalezca cada vez más. Esto hace que la metodología sea escalable y de mejora continua.

La determinación del objetivo GQM a partir del problema planteado, fue la base para definir las preguntas de investigación y alcance del estudio. Este proceso fue una secuencia de iteraciones continuas con el fin de afinar la cadena de búsqueda automatizada propuesta y asegurar que los artículos analizados durante el estudio, dieran una solución adecuada al problema planteado.

Durante el proceso aplicado se validó en las etapas de evaluación de calidad y análisis de resultados para dar solución a las preguntas de investigación planteadas, se validó la necesidad de que en los estudios primarios se detallen de mejor forma los experimentos o casos de estudios aplicados. La razón de esto es porque algunos estudios estaban muy bien planteados, pero no resultaron relevantes al no poder extraer los detalles necesarios para el análisis planteado.

Todo el proceso llevado durante esta investigación fue retador, principalmente por aprender una metodología nueva. Este tipo de trabajos necesitan de tiempo y dedicación en cada una de las etapas planteadas, ya que debe existir fortaleza en lo que se esté desarrollando. La parte más complicada fue realizar la extracción y análisis de datos ya que se debía llegar a una solución que engloba a las preguntas de investigación para resolver el problema inicial planteado.

Este tipo de metodologías son recomendadas como aplicación en la academia. Sin embargo, dado el proceso y trabajo continuo que implica, es bastante difícil pensar que se apliquen en ambientes prácticos fuera del campo de investigación, es decir, compañías que no tengan un área de investigación definida, no sería fácil de adecuar este tipo de metodologías para la toma de decisiones del día a día.

El proceso aplicado durante este análisis no solamente permitió verificar las líneas de las investigaciones que se han llevado a cabo en el área planteada, sino que además, identificó trabajo futuro tanto en estudios primarios, como para estudios secundarios.

3.9. Conclusiones

El objetivo de esta investigación fue realizar un mapeo sistemático de estudios primarios que evaluaran técnicas de minería de datos y aprendizaje automático en el contexto de segmentación de clientes bancarios. Estos fueron publicados entre 2005 y 2019. Se analizó un conjunto de 87 estudios primarios, después de aplicar criterios de exclusión e inclusión. Durante este proceso también se realizó una evaluación de calidad, con el fin de verificar cuáles estudios eran más relevantes.

Los resultados mostraron una amplia gama de técnicas utilizadas por estudios en el área, donde cada configuración que se aplique puede variar los resultados. Según la clasificación utilizada durante la investigación, los paradigmas más reportados fueron *decision tree* con 88 referencias y *linear preditors* con 54 referencias. Cabe destacar que varios estudios no reportan la parametrización o configuración empleada durante sus evaluaciones. Esto es algo que se recomienda incluir en estudios primarios, ya que así se podrían hacer replicaciones o bien implementar mejoras sobre las técnicas o experimentos planteados, manteniendo las mismas condiciones y ambiente.

Con respecto a las herramientas que soportan las técnicas de minería de datos y aprendizaje automático, no se obtuvo una tendencia clara, ya que a través de los años en que se reportaron los estudios ya existían varias herramientas que tienen implementados los algoritmos y solo deben ser configurados según lo que se esté trabajando o también herramientas que permiten implementar las técnicas basándose en bibliotecas especializadas ya existentes. Con respecto al tipo de herramientas según el licenciamiento, se vio que prácticamente son: la mitad de *software* libre y la otra mitad de licencia pagada. Las herramientas más reportadas fueron Weka con 13 referencias y Matlab con 12 referencias. En el caso de Weka ya tienen las técnicas implementadas y es de licencia gratuita, por el contrario, Matlab es de licencia pagada y hay que implementar los algoritmos.

Este estudio también planteó identificar los conjuntos de datos y sus características. Se pudo validar que el repositorio público *UCI Machine Learning* de la Universidad de California fue el más utilizado con 6 de sus variaciones, con 60 reportes. Dentro de los atributos de estos conjuntos de datos se detectaron variables tanto numéricas como nominales, donde las cinco más utilizadas fueron: la edad, el trabajo, el género,

la temporalidad y las características crediticias. Además, también se investigó sobre las métricas de evaluación de rendimiento de las técnicas reportadas. Se validó mediante la investigación y la teoría, que hay métricas estándar utilizadas en este tipo estudios, que se basan en la matriz de confusión. De las métricas de evaluación se observó una tendencia por utilizar *accuracy*, que fue reportada en 66 estudios de 87 analizados.

Esta investigación puede ser aplicada para diferentes áreas e industrias. En el caso del área profesional, para las empresas que no cuentan con los recursos especializados en estas tecnologías, puede ser un buen inicio tomar como referencia estudios que hayan investigado sobre el tema y así, tener una idea de por donde se puede iniciar la experimentación de modelos que utilicen minería de datos y aprendizaje automático. En el caso del área de investigación, este estudio permite tener una revisión actualizada de los temas que han sido investigados en un área determinada y a partir de ahí, crear o identificar nuevas líneas de estudio que permitan ampliar el conocimiento sobre el contexto elegido. Finalmente, en el área académica permite que se incentive a los estudiantes a que hagan investigación mediante el tipo de metodología aplicada en este estudio, fortalece la bases con las que la Escuela de Computación e Informática de la Universidad de Costa Rica define cursos como por ejemplo, de bases de datos avanzados, de calidad del software y de inteligencia artificial.

Con el desarrollo de este proyecto se concluye que el tema ha sido estudiado y que la mayor diferencia entre los experimentos depende del ambiente que se implemente. Las configuraciones, la parametrización, los criterios de análisis, son solo algunos puntos que se deben tomar en cuenta para una implementación que incluya técnicas de minería de datos y aprendizaje automático.

Como trabajo futuro se plantea hacer una comparativa entre las técnicas de minería de datos y aprendizaje automático que fueron reportadas con más frecuencia, y teniendo en cuenta que la evaluación se debe hacer bajo las mismas condiciones, es decir, utilizar el mismo conjunto de datos, aplicar el mismo preprocesamiento de datos, elegir una herramienta adecuada para el caso de estudio y aplicar las métricas de rendimiento bajo los mismos parámetros. El propósito de realizar esta comparativa es identificar cuáles técnicas son mejores que otras contando con un ambiente controlado. Esto aplicaría como un estudio primario.

En el caso de un estudio secundario, esta investigación puede ser ampliada determinando un contexto más general, donde permita identificar diferentes clasificaciones de las técnicas de minería de datos y aprendizaje automático y así relacionar qué tanto cambian los resultados según la clasificación que se escoja.

A partir del capítulo de la memoria se desarrolló un artículo científico que fue enviado y aceptado en la Intelligent Systems Conference (IntelliSys) que se desarrollará el 3-4 de setiembre del 2020 en Amsterdam, Holanda. El artículo fue publicado en el Springer series “Advances in Intelligent Systems and Computing” e indexado en ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar y Springerlink. En el Apéndice 3.D se encuentra el artículo publicado.

Apéndice

3.A. Lista de estudios primarios incluidos.

El Cuadro 3.7 muestra la lista final de los artículos primarios incluidos en el estudio. Contiene el ID de cada artículo, el título, el año de publicación y el número de referencia.

Cuadro 3.7: Lista de estudios primarios incluidos.

ID	Título	Año	Est.
1	Direct marketing campaigns in retail banking with the use of deep learning and random forests	2019	[46]
3	A data mining application in credit scoring processes of small and medium enterprises commercial corporate customers	2019	[13]
8	Term deposit subscription prediction using spark MLlib and ML packages	2019	[18]
9	Mutual information and sensitivity analysis for feature selection in customer targeting: A comparative study	2019	[61]
11	An empirical comparison of machine-learning methods on bank client credit assessments	2019	[47]

Continúa en la página siguiente.

ID	Título	Año	Est.
12	Credit Collectibility Prediction of Debtor Candidate Using Dynamic K-Nearest Neighbor Algorithm and Distance and Attribute Weighted	2019	[92]
13	An Automatic Interaction Detection Hybrid Model for Bankcard Response Classification	2019	[19]
15	Comparison of Data Mining Classification Algorithms Determining the Default Risk	2019	[14]
20	Retail credit scoring using fine-grained payment data	2019	[79]
26	An Improved Bank Credit Scoring Model: A Naïve Bayesian Approach	2018	[97]
27	Multinomial Logistic Regression and Spline Regression for Credit Risk Modelling	2018	[64]
29	Fuzzy Credit Risk Scoring Rules using FRvarPSO	2018	[84]
32	On ensemble SSL algorithms for credit scoring problem	2018	[38]
33	A customer segmentation approach in commercial banks	2018	[93]
35	The Development of Bank Application for Debtors Selection by Using Naïve Bayes Classifier Technique	2018	[98]
37	Classification Consumer Credit for Missing Value Dataset	2018	[51]
48	Classification of a bank data set on various data mining platforms [Bir Banka Müşteri Verilerinin Farkli Veri Madenciliği Platformlarinda Siniflandirilmasi]	2018	[15]

Continúa en la página siguiente.

ID	Título	Año	Est.
49	Credit risk analysis using machine learning classifiers	2018	[20]
51	Prediction of loan status in commercial bank using machine learning classifier	2018	[94]
54	Credit scoring analysis using weighted k nearest neighbor	2018	[95]
61	From data exploration to semantic model of customer	2018	[39]
62	On identifying potential direct marketing consumers using adaptive boosted support vector machine	2018	[86]
64	A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring	2018	[87]
65	A novel classifier ensemble approach for financial distress prediction	2018	[40]
68	Improved credit card churn prediction based on rough clustering and supervised learning techniques	2018	[21]
73	A Comparative Study to the Bank Market Prediction	2018	[22]
74	A novel multi-objective particle swarm optimization for comprehensible credit scoring	2018	[23]
81	Predicting customer response to bank direct telemarketing campaign	2017	[48]
82	Prediction analysis of risky credit using Data mining classification models	2017	[24]
84	SR-based binary classification in credit scoring	2017	[74]

Continúa en la página siguiente.

ID	Título	Año	Est.
90	A new classification algorithm for the bank customer credit rating	2017	[49]
96	A relative evaluation of the performance of ensemble learning in credit scoring	2017	[50]
98	Modeling credit approval data with neural networks: an experimental investigation and optimization	2017	[72]
99	Imbalanced customer classification for bank direct marketing	2017	[56]
120	A machine learning approach for predicting bank credit worthiness	2016	[25]
127	Classifiers consensus system approach for credit scoring	2016	[26]
129	Improve the Prediction Accuracy of Naïve Bayes Classifier with Association Rule Mining	2016	[41]
131	Credit modelling using hybrid machine learning technique	2016	[42]
133	Using data mining and neural networks techniques to propose a new hybrid customer behaviour analysis and credit scoring model in banking services based on a developed RFM analysis method	2016	[73]
145	A Cluster-Based Data Balancing Ensemble Classifier for Response Modeling in Bank Direct Marketing	2015	[27]
149	Credit scoring with an improved fuzzy support vector machine based on grey incidence analysis	2015	[88]

Continúa en la página siguiente.

ID	Título	Año	Est.
154	Feasibility study for banking loan using association rule mining classifier	2015	[100]
155	A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring	2015	[43]
166	Credit risk assessment model for Jordanian commercial banks: Neural scoring approach	2014	[65]
169	Building classification models for customer credit scoring	2014	[99]
170	Credit risk prediction using fuzzy immune learning	2014	[44]
173	Combining multiple classifiers based on Dempster-Shafer theory for personal credit scoring	2014	[66]
185	Response Modeling in Direct Marketing. A Data Mining-Based Approach for Target Selection.	2013	[89]
195	Using semi-supervised classifiers for credit scoring	2013	[57]
196	The model and empirical research of application scoring based on data mining methods	2013	[67]
202	Toward a new classification model for analysing financial datasets	2012	[58]
206	Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment	2012	[80]
209	On the impact of disproportional samples in credit scoring models: An application to a Brazilian bank data	2012	[68]

Continúa en la página siguiente.

ID	Título	Año	Est.
223	An empirical study on credit scoring model for credit card by using data mining technology	2011	[29]
224	Post mining of Multiple Criteria Linear Programming classification model for actionable knowledge in credit card churning management	2011	[75]
232	Classification of customer credit data for intelligent credit scoring system using fuzzy set and MC2 - Domain driven approach	2011	[28]
233	Logistic sub-models for small size populations in credit scoring	2011	[69]
239	Behavioral rules of bank's point-of-sale for segments description and scoring prediction	2011	[62]
243	Predicting credit card holder churn in banks of China using data mining and MCDM	2010	[45]
244	Credit card customer segmentation and target marketing based on data mining	2010	[30]
245	Using data mining predictive models to classify credit card applicants	2010	[52]
247	A method combined of support vector machine and F-scores for customer classification	2010	[90]
261	Domain-driven classification based on multiple criteria and multiple constraint-level programming for intelligent credit scoring	2010	[31]
270	Solving credit scoring problem with ensemble learning: A case study	2009	[32]

Continúa en la página siguiente.

ID	Título	Año	Est.
277	Classification methods of credit rating - A comparative analysis on SVM, MDA and RST	2009	[91]
283	ANN model for corporate credit risk assessment	2009	[76]
285	Comparison procedure of predicting the time to default in behavioural scoring	2009	[81]
286	Mining the customer credit using hybrid support vector machine technique	2009	[53]
287	The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients	2009	[54]
291	Client classification on credit risk using rough set theory and ACO-based support vector machine	2008	[83]
295	Credit risk analysis using Hidden Markov Model	2008	[70]
303	A modified genetic programming for behavior scoring problem	2007	[77]
314	Towards a comprehensible and accurate credit management model: Application of four computational intelligence methodologies	2006	[33]
322	Mining the customer credit using classification and regression tree and multivariate adaptive regression splines	2006	[55]
329	A data mining approach for retailing bank customer attrition analysis	2005	[34]
354	A customer classification prediction model based on machine learning techniques	2015	[16]

Continúa en la página siguiente.

ID	Título	Año	Est.
355	A new approach based on a rough set and a decision tree to bank customer creditevaluation	2008	[59]
356	Multi-Classifier Combination for Banks Credit Risk Assessment	2006	[96]
357	A bank customer credit evaluation based on the decision tree and the simulated annealing algorithm	2008	[60]
358	Credit Risk Assessment Based on Fuzzy SVM and Principal Component Analysis	2009	[78]
363	Preventing customer churn by using random forests modeling	2008	[35]
375	A dynamic credit risk assessment model with data mining techniques: evidence from Iranian banks	2019	[85]
377	The Consumer Loan's Payment Default Predictive Model: an Application of the Logistic Regression and the Discriminant Analysis in a Tunisian Commercial Bank	2018	[71]
398	A case-based reasoning model that uses preference theory functions for credit scoring	2012	[63]
416	The Application of Tree-based model to Unbalanced German Credit Data Analysis	2018	[36]
420	Application of Ensemble Models in Credit Scoring Models	2018	[37]
424	Improve Profiling Bank Customer's Behavior Using Machine Learning	2019	[82]

3.B. Evaluación de calidad de los estudios primarios

El Cuadro 3.8 expone los resultados de la evaluación de calidad aplicada sobre los estudios primarios analizados. Contiene el ID de cada artículo, el número de referencia, el año en que fue publicado, el resultado obtenido por la Q1, Q2, Q3, Q4 y el resultado de los cuatro criterios evaluados. Cabe destacar, que la escala de evaluación es de 0 a 2 puntos, por lo que el puntaje máximo es de 8 puntos.

Cuadro 3.8: Evaluación de calidad de los estudios primarios.

ID	Est.	Año	Q1	Q2	Q3	Q4	Total
1	[46]	2019	1	2	2	2	7
3	[13]	2019	2	2	1	2	7
8	[18]	2019	2	2	1	2	7
9	[61]	2019	1	2	1	1	5
11	[47]	2019	2	1	1	1	5
12	[92]	2019	1	1	1	1	4
13	[19]	2019	1	1	1	2	5
15	[14]	2019	2	2	2	2	8
20	[79]	2019	1	0	1	1	3
26	[97]	2018	2	1	1	1	5
27	[64]	2018	2	0	1	0	3
29	[84]	2018	1	0	1	1	3
32	[38]	2018	1	2	2	2	7
33	[93]	2018	1	1	1	1	4
35	[98]	2018	1	0	1	1	3
37	[51]	2018	1	0	1	2	4
48	[15]	2018	2	2	2	2	8
49	[20]	2018	2	1	1	2	6
51	[94]	2018	1	1	2	0	4
54	[95]	2018	1	0	1	1	3

Continúa en la página siguiente.

ID	Est.	Año	Q1	Q2	Q3	Q4	Total
61	[39]	2018	1	2	0	1	4
62	[86]	2018	2	2	0	2	6
64	[87]	2018	2	0	2	2	6
65	[40]	2018	2	1	1	2	6
68	[21]	2018	2	0	1	2	5
73	[22]	2018	1	0	2	2	5
74	[23]	2018	1	1	2	2	6
81	[48]	2017	1	2	1	2	6
82	[24]	2017	1	0	1	2	4
84	[74]	2017	2	1	2	2	7
90	[49]	2017	1	0	1	2	4
96	[50]	2017	1	2	2	2	7
98	[72]	2017	1	0	1	2	4
99	[56]	2017	2	0	2	2	6
120	[25]	2016	1	1	2	2	6
127	[26]	2016	1	1	1	2	5
129	[41]	2016	2	1	1	2	6
131	[42]	2016	1	0	1	2	4
133	[73]	2016	1	1	0	1	3
145	[27]	2015	2	1	2	2	7
149	[88]	2015	1	0	1	2	4
154	[100]	2015	2	1	0	0	3
155	[43]	2015	1	1	1	1	4
166	[65]	2014	2	1	1	1	5
169	[99]	2014	2	1	1	2	6
170	[44]	2014	1	1	2	2	6
173	[66]	2014	2	1	1	1	5
185	[89]	2013	2	1	1	0	4
195	[57]	2013	1	1	0	2	4

Continúa en la página siguiente.

ID	Est.	Año	Q1	Q2	Q3	Q4	Total
196	[67]	2013	2	1	1	1	5
202	[58]	2012	2	0	1	2	5
206	[80]	2012	1	0	1	1	3
209	[68]	2012	2	1	2	1	6
223	[29]	2011	1	1	1	1	4
224	[75]	2011	1	0	1	0	2
232	[28]	2011	1	0	1	2	4
233	[69]	2011	2	0	1	2	5
239	[62]	2011	1	1	0	0	2
243	[45]	2010	1	1	0	1	3
244	[30]	2010	1	0	1	1	3
245	[52]	2010	2	1	1	1	5
247	[90]	2010	2	0	1	1	4
261	[31]	2010	1	1	1	2	5
270	[32]	2009	2	1	1	2	6
277	[91]	2009	1	1	1	0	3
283	[76]	2009	1	0	1	0	2
285	[81]	2009	1	1	1	1	4
286	[53]	2009	2	1	1	1	5
287	[54]	2009	1	0	2	2	5
291	[83]	2008	2	0	1	0	3
295	[70]	2008	1	0	1	2	4
303	[77]	2007	2	0	1	1	4
314	[33]	2006	1	0	1	0	2
322	[55]	2006	1	1	2	1	5
329	[34]	2005	1	0	2	1	4
354	[16]	2015	2	1	2	2	7
355	[59]	2008	1	0	1	2	4
356	[96]	2006	1	0	1	0	2

Continúa en la página siguiente.

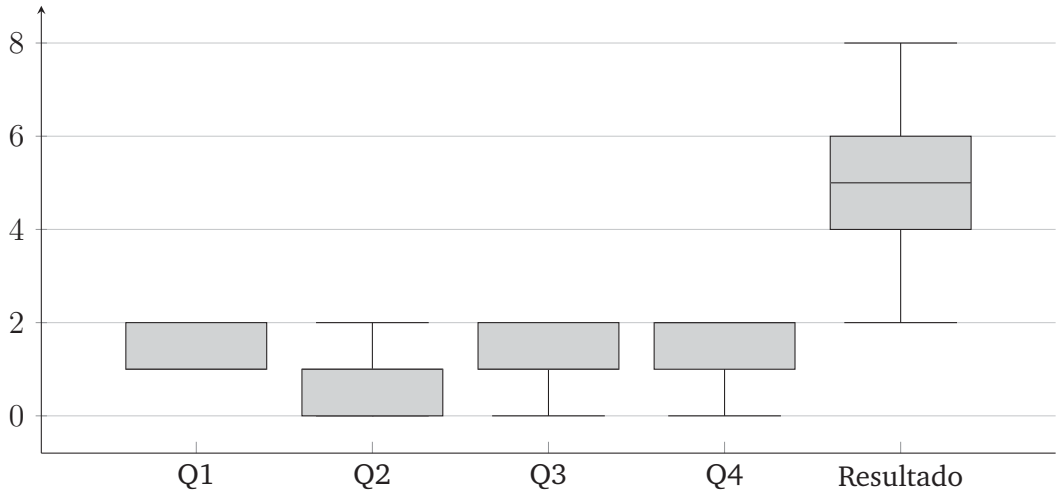


Figura 3.16: Evaluación completa de calidad.

ID	Est.	Año	Q1	Q2	Q3	Q4	Total
357	[60]	2008	2	0	1	2	5
358	[78]	2009	1	1	2	0	4
363	[35]	2008	1	0	1	1	3
375	[85]	2019	2	1	2	0	5
377	[71]	2018	2	0	1	1	4
398	[63]	2012	2	1	1	2	6
416	[36]	2018	1	0	1	2	4
420	[37]	2018	1	1	2	0	4
424	[82]	2019	1	1	1	2	5

3.C. Evaluación completa de calidad

La Figura 3.16 muestra los resultados de la evaluación de calidad aplicada sobre los 87 estudios primarios analizados. Documenta las medianas, bigotes superiores e inferiores de cada pregunta de la evaluación de calidad.

3.D. Artículo

A partir del capítulo de la memoria se desarrolló un artículo científico que fue enviado y aceptado en la Intelligent Systems Conference (IntelliSys) que se desarrollará el 3-4 de setiembre del 2020 en Amsterdam, Holanda. El artículo fue publicado en el Springer series “Advances in Intelligent Systems and Computing” e indexado en ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar y Springerlink.

Data Mining and Machine Learning Techniques for Bank Customers Segmentation: A Systematic Mapping Study

Maricel Monge(✉), Christian Quesada-López(✉), Alexandra Martínez(✉), and Marcelo Jenkins(✉)

Universidad de Costa Rica, San Pedro, Costa Rica
{cruz.monge,cristian.quesadalopez,
alexandra.martinez,marcelo.jenkins}@ucr.ac.cr

Abstract. Data mining and machine learning techniques analyze and extract useful information from data sets in order to solve problems in different areas. For the banking sector, knowing the characteristics of customers entails a business advantage since more personalized products and services can be offered. The goal of this study is to identify and characterize data mining and machine learning techniques used for bank customer segmentation, their support tools, together with evaluation metrics and datasets. We performed a systematic literature mapping of 87 primary studies published between 2005 and 2019. We found that decision trees and linear predictors were the most used data mining and machine learning paradigms in bank customer segmentation. From the 41 studies that reported support tools, Weka and Matlab were the two most commonly cited. Regarding the evaluation metrics and datasets, accuracy was the most frequently used metric, whereas the UCI Machine Learning repository from the University of California was the most used dataset. In summary, several data mining and machine learning techniques have been applied to the problem of customer segmentation, with clear tendencies regarding the techniques, tools, metrics and datasets.

Keywords: Data mining · Machine learning · Customer segmentation · Systematic literature mapping

1 Introduction

The emergence of innovative technologies in analytics, such as data mining, machine learning, big data, and artificial intelligence, has allowed handling large amounts of data that are generated every day [1]. In the banking industry, the study and experimentation with these technologies has enabled the solution and automation of common problems [2]. One of these problems is customer knowledge (knowing their tastes, preferences and trends) to customize the offer

K. Arai et al. (Eds.): IntelliSys 2019, AISC 1251, pp. 1–19, 2021.
https://doi.org/10.1007/978-3-030-55187-2_48

of products and services. A related problem is customer segmentation, which divides customers in categories with common characteristics, and builds upon customer knowledge [3]. Customer segmentation can represent a competitive advantage, as new customers can be attracted while maintaining the loyalty of current ones by increasing their satisfaction [3].

Data mining refers to the procedure of detecting model in a data set, while machine learning is the process of training an algorithm with data so that it may be able to learn from the data [1]. Data mining and machine learning are complementary approaches because data mining needs the learning capacity for algorithms to be scalable, and machine learning needs the pattern discovery to make the model grow [1].

The objective of this study is to characterize data mining and machine learning techniques used for bank customer segmentation, in terms of their support tools, evaluation metrics, and evaluation data sets. To guide this study, three research questions were defined:

RQ1. What data mining and machine learning techniques have been used for bank customer segmentation?
RQ2. What tools have been used to support data mining and machine learning techniques in bank customer segmentation?
RQ3. What metrics and data sets have been used to evaluate data mining and machine learning techniques in bank customer segmentation?

The remainder of our paper is structured as follows. Section 2 presents the background. Section 3 describes related works in the area. Section 4 explains the methodology followed. Section 5 shows and discusses the results obtained for each research question. Finally, Sect. 6 offers our conclusions.

2 Background

Customer segmentation is the process of dividing clients in different groups, with the purpose of increasing customer satisfaction and therefore, business profits. This classification is based in common features of customers. Customer segmentation can be done based on criteria such as degree of customer loyalty, purchase frequency, purchase volume, demographics, etc. [3].

Data mining is the process of identifying and discovering patterns, trends and relationships in data [1]. Patterns identified in the analysis should be evaluated with respect to business expectations, to determine if they work as good predictors. Data mining is divided into two areas: *descriptive* and *predictive* analytics. In descriptive analytics, the main algorithms are those of clustering and association [4]. In predictive analytics, the main algorithms are those of classification and time series [4].

Machine learning is the process of making models learn from data [1]. This is done through training, so that the model creates patterns against which to compare and classify new data [9]. Machine learning techniques can be divided

in three broad categories: supervised learning, unsupervised learning, and semi-supervised learning. Supervised learning algorithms are based on classification methods where they must have prior data training. Unsupervised learning algorithms are based on grouping methods where attempts are made to discover classes in the data. Semi-supervised learning algorithms are a combination between the first two categories [9].

There are different classifications of data mining and machine learning techniques. In this study, we used the classification presented by Shalev-Shwartz and Ben-David [11], which consists of the following categories:

1. **Linear predictors**: The algorithms that belong to this paradigm are based on linear prediction functions and supervised learning, because they need training patterns to predict new data. Within this paradigm are techniques such as logistic regression, linear regression, linear programming and perceptron algorithm [11].
2. **Boosting**: This approach begins with a basic learning based on training the model with data and, as it progresses, more knowledge is generated, which enriches the class to which the predictor belongs. This paradigm include algorithms like boosting, bagging and adaptative boosting [11].
3. **Support vector machines**: This paradigm seeks to separate the complexity in the data, by finding separators that will divide the dimensions intended to capture the greatest convergence among data. Examples of techniques in this paradigm are sequential minimum optimization and fuzzy support vector machine [11].
4. **Decision trees**: This type of algorithm predicts whether a data corresponds to a certain class. Classification is done by traversing a decision tree from the root node to a leaf. Random forest, C4.5 tree, classification and regression tree, and naïve bayes tree are examples of algorithms belonging to this paradigm [11].
5. **Nearest neighbor**: The algorithms from this paradigm are trained with a set of data, to subsequently predict the label of any new instance based on the labels of the closest nodes. Examples of algorithms of this paradigm are: k-nearest neighbor, simulated annealing algorithm, and weighted k-nearest neighbor [11].
6. **Neural networks**: Inspired by the brain's neural network model, this technique consists of a series of layers interconnected with each other by a network that carries information from one node to another. Inner layers are more advanced and allow calculations. The initial and output layers are simpler: they only receive and communicate the data. Bayes network, backpropagation and neural networks are included in this paradigm [11].
7. **Clustering**: This class of algorithms is based on exploratory data analysis. The idea is to form clusters, considering characteristics that unite or separate them from other clusters. Examples of algorithms in this paradigm are k-means and hierarchical clustering [11].
8. **Dimensional reduction**: This paradigm takes high-dimensional data and maps it to a smaller dimensional space, in order to reduce data complexity,

at the price of losing information. Principal component analysis is one of the main algorithms within this paradigm [11].

9. **Generative models**: Algorithms that belong to this class seek the underlying distribution of the data, in order to estimate its parameters. These algorithms have the problem that learning becomes increasingly complex. Linear discriminant analysis and Naïve Bayes are included in this paradigm [11].

3 Related Work

We were not able to find similar studies in the context of bank clients segmentation. However, previous studies have proposed and evaluated data mining and machine learning techniques to solve other problems in the banking industry. We describe these works below.

Bhambri [4] studied several data mining techniques such as association, clustering, classification and forecasting. Association is related with the frequency to find items in the data set. Association includes various types such as: multilevel association rule, multidimensional association rule, Quantitative association rule, direct association rule and indirect association rule. Clustering is the identification if similar objects. Forecasting can model relations between dependent and independent attributes. Forecasting includes algorithms like: logistic regression, decision trees, and neural networks. Classification models used to classify the population of records. All of these techniques were applied to solve banking problems such as fraud detection, marketing, risk management and business risk.

Pulakkazhy et al. [5] performed an analysis of data preparation before applying data mining process. The steps include: data selection, preprocessing, cleaning, integration, transform and reduction data. After apply data mining, the next steps are pattern evaluation and knowledge presentation. These study use techniques like multilevel association rule, multidimensional association rule, quantitative association rule, direct association rule, indirect association rule, decision trees, neural networks and k-means. They also mentioned different areas in the banking industry where data mining can be applied: risk management, fraud detection, and marketing to attract new customers to the business.

Hasheminejad et al. [3] did a comprehensive analysis of data sets that have been used in data mining assessments. This study describes the parameters used, the data set size, and the type of preprocessing done. Additionally, they reviewed the most commonly used data mining techniques and the most common performance evaluation criteria. The most used techniques were: self organizing map, k-means, decision trees, neural networks and Naive Bayes. They found that accuracy, precision and F-score were the most common evaluation criteria.

Goebel et al. [6] summarized the tools used in data mining experiments, explaining the tasks performed by the tools, the algorithms evaluated, and the variables used. Some tasks applied in the tools are: Ability to access a variety of data sources, Online/Offline data access, Query language, The underlying data model, among others. Techniques include in this study are: Prediction, Regression, Classification, Clustering and Associations. Their study is relevant

because it is one of the few studies that review the characteristics of tools in the context of bank data.

Although all reviewed secondary studies report data mining and machine learning techniques, not all of them mention the tools used, performance metrics and data sets. Our study contributes by providing this information, which gives a broader view of the area. Furthermore, none of the previous studies were targeted for customer segmentation, while our is. Yet, the review of these studies gave us insights into the area's state of the art.

4 Methodology

We conducted a systematic mapping study to select and analyze existing literature on data mining and machine learning techniques used for bank customer segmentation. We followed the methodology stated in [7] and the recommendations in [8].

4.1 Search Process and Study Selection

First, an exploratory search was conducted to identify appropriate search terms in accordance with the objective. For this purpose, we conducted search on Scopus using various combinations of terms and calibrated the search string to include papers in the scope of our study. Based on the results and insights from the exploratory search phase, we selected the following search string:

("marketing" OR "credit") AND ("client" OR "customer") AND
("acqu*" OR "scor*" OR "classif*") AND ("bank*") AND
("mining" OR "machine learn*" OR "predict*" OR "classif*")

With the search string, we conducted a search on *Scopus*, *IEEE Xplore*, and *Web of Science* digital libraries for relevant papers up to October 2019. We looked for studies that contained the search terms anywhere in the title, abstract or keywords fields.

Inclusion and Exclusion Criteria. Exclusion criteria are used to narrow down the initial set of search results by the process of elimination. Articles meeting the following criteria were excluded: (E1) Studies that are not available in full text. (E2) Studies that are not written in the English language. (E3) Books and non-peer reviewed primary studies.

Inclusion criteria defines the attributes that are essential for a study to be selected in our analysis. Articles meeting the following criteria were included: (I1) Studies that use bank data for credit and marketing. (I2) Studies that use data mining and machine learning techniques. (I3) Studies that describe customer segmentation processes.

Selection Process. Our study selection process is depicted in Fig. 1. The search string produced an initial set of 534 potentially relevant studies. The inclusion and exclusion criteria were applied to 409 studies (removing duplicates), considering only the title, abstract and keywords fields. Any paper that was irrelevant became excluded. In general, we tended to include rather than exclude potentially relevant papers. After applying the exclusion and inclusion criteria, we were left with 100 studies. Our analysis set includes a total of 87 papers after detailed reading. Each paper is assigned a unique reference code (prefixed by the letter "S") for the purpose of our analysis (S01-S86). The complete list of selected papers along with their quality assessment, and extraction form, is available at https://tinyurl.com/we6dlp7.

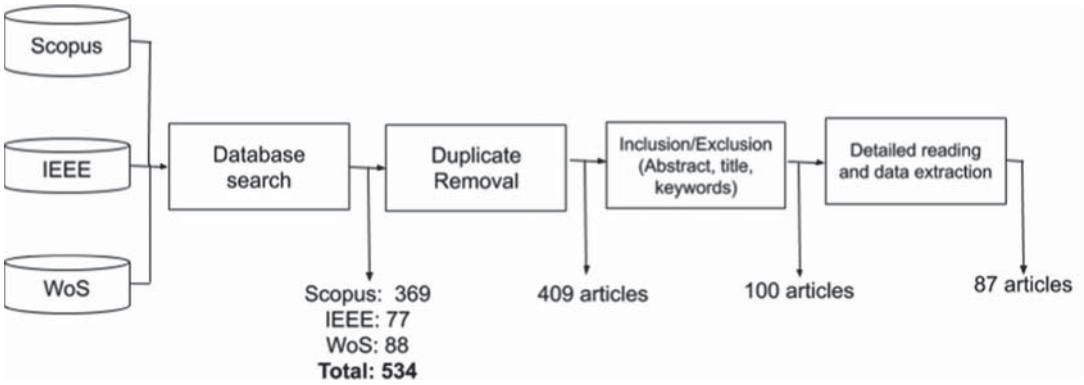


Fig. 1. Overview of the selection process used in the study.

4.2 Quality Assessment of Included Studies

The quality assessment was applied to the included primary studies, in order to determine the relevance and contribution (level of detail) that each study could have to answer our specific research questions. For this, the following quality assessment criteria were defined:

Q1. The study describes the data mining and machine learning techniques used.
Q2. The study states the tools that have been used to support the techniques.
Q3. The study discusses the performance evaluation metrics used for the evaluation of the techniques.
Q4. The study describes the data set used for the evaluation of the techniques.

The score was assigned to each question on a scale from 0 to 2, where 0 = Does not comply the criterion, 1 = Partially complies the criterion, and 2 = Complies the criterion. According to the defined scale, the maximum quality score that any of the studies can reach is 8 points and the minimum is 0. The quality assessment on average resulted in 4.11, indicating that the studies are of fairly good quality.

4.3 Data Extraction and Analysis

To apply the data extraction process, we created a form with aspects to be extracted from the selected studies. We extract each of the data items from each study and organize the data from different studies into common categories in order to obtain the information that help answering the research questions. Table 1 details the elements extracted for each category. The form contains four categories: general information, techniques (RQ1), tools (RQ2) and metrics (RQ3), each with their respective elements.

To answer RQ1, the analysis consists in identifying the techniques reported in the studies and counting the studies used by each technique. To make the tabulation it was necessary to use a classification that applies to data mining and machine learning techniques [11]. With respect to RQ2, the tools used to apply data mining techniques and machine learning was identified. For RQ3, the metrics that evaluated the performance of the techniques, the data sets, the attributes used and the results in each evaluation were identified.

Table 1. Extracted data items.

Category	Elements
General	Database, id, title, authors, year, abstract, keywords, type of article, Q1, Q2, Q3, Q4, quality result
Techniques (RQ1)	Name, description, technique configuration
Tools (RQ2)	Name, description
Metrics (RQ3)	Name, formula, result, data set, attributes, data set size

4.4 Threats to Validity

We briefly discuss the threats to validity of our mapping study. *Search terms and digital libraries.* The automated search was implemented by defining keywords based on the control articles and the PICO model. In the case of the chosen databases, they are recognized and recommended in the area of software engineering. *Selection of studies.* For the cases were presented in the inclusion and exclusion process where it was not clear if the study should be included, it was decided to include it and then make a complete reading to validate if the article was relevant for the investigation. *Data extraction and categorization.* The process of extracting and classifying the studies was conducted by the first author and validated by a second researcher. In the extraction process, each article was classified based on a recognized taxonomy [11]. The generalization and synthesis of results are limited to the studies analyzed in the investigation. To minimize risks when presenting the results, all research was applied following previously defined and validated protocols.

5 Results and Discussion

In this section we present the results of our mapping study, and address our specific research questions.

5.1 Data Mining and Machine Learning Techniques for Customer Segmentation (RQ1)

The first research question helps to identify the most commonly used data mining and machine learning techniques to segment bank customers. Attempts were made to discover trends in the techniques found.

The complete list of data mining and machine learning techniques reported by the primary studies are shown in Table 2. This table includes the paradigm in which the techniques are classified (according to [11]), the quantity of studies that report each technique, and the study reference. From this table we see that the two most common paradigms found were decision trees and linear predictors. These paradigms group 48% of the reported techniques. Interestingly, both of these paradigms comprise algorithms for predictive analytics, such as classification and regression.

Table 2. Data mining & machine learning techniques reported.

Paradigm	Technique	Qty	Studies
Decision trees	Decision tree	20	[S09, S12, S13, S16, S19, S26, S36, S38, S40, S42, S44, S51, S52, S66, S68, S72, S74, S77, S79, S85]
	Naïve bayes tree	16	[S01, S02, S03, S08, S09, S18, S29, S32, S35, S38, S39, S40, S51, S56, S59, S77]
	Random forest	16	[S01, S02, S06, S09, S13, S19, S20, S30, S31, S35, S40, S51, S53, S68, S72, S79]
	Classification and regression tree	13	[S01, S06, S09, S14, S15, S18, S26, S35, S39, S48, S64, S83, S85]
	C4.5 tree	13	[S01, S02, S03, S10, S26, S30, S35, S55, S56, S61, S71, S73, S85]
	PART	3	[S01, S32, S35]
	Decision table	2	[S32, S35]
	CN2 induction rules	1	[S08]
	Best-first tree	1	[S35]
	Mutual information algorithm	1	[S07]
	RIPPER algorithm	1	[S34]
	Case-based reasoning	1	[S05]

(*continued*)

Table 2. (*continued*)

Paradigm	Technique	Qty	Studies
Linear predictors	Logistic regression	22	[S02, S09, S15, S20, S25, S26, S30, S33, S35, S36, S42, S44, S45, S48, S56, S58, S60, S64, S67, S75, S78, S83]
	Multilayer perceptron	16	[S02, S08, S20, S29, S30, S33, S38, S39, S41, S42, S52, S54, S55, S56, S73, S74]
	Linear regression	6	[S15, S16, S22, S32, S70, S72]
	Multivariate adaptive regression splines	5	[S14, S15, S20, S45, S67]
	Probit regression	3	[S27, S69, S87]
	Data-based sensitivity analysis Algorithm	1	[S07]
	Benchmark rating model	1	[S65]
Neural networks	Neural networks	21	[S09, S12, S15, S16, S18, S22, S25, S26, S36, S38, S40, S62, S64, S68, S72, S74, S77, S81, S83, S85, S86]
	Backpropagation	4	[S15, S27, S69, S76]
	Bayes network	3	[S02, S35, S38]
	Radial basis function network	3	[S42, S56, S58]
	Learning vector quantization	2	[S34, S46]
	Adaptive neuro fuzzy inference system	1	[S28]
	Extreme learning machine	1	[S38]
Support vector machine	Support vector machine	27	[S04, S09, S12, S14, S18, S20, S23, S27, S32, S36, S38, S39, S40, S42, S50, S51, S52, S53, S55, S57, S63, S68, S72, S74, S76, S82, S84]
	Sequential minimum optimization	3	[S10, S29, S35]
	Fuzzy support vector machine	2	[S27, S57]
Nearest neighbor	K nearest neighbor	16	[S03, S04, S10, S16, S17, S21, S24, S29, S35, S38, S39, S49, S51, S55, S60, S64]
	Weighted k nearest neighbor	3	[S08, S49, S65]
	Dynamic k nearest neighbor	1	[S17]
	Simulated annealing algorithm	1	[S61]
	Ant colony algorithm	1	[S76]
	Multi-objective particle swarm optimization approaches	1	[S09]
Boosting	Boosting	8	[S03, S17, S20, S29, S31, S39, S49, S80]
	Bagging	6	[S16, S19, S31, S39, S79, S80]
	Adaboost	5	[S19, S36, S50, S72, S79]
	Gradient boosting	2	[S13, S19]
	Online Moving Average Reversion	1	[S66]

(*continued*)

Table 2. (*continued*)

Paradigm	Technique	Qty	Studies
Generative models	Naïve bayesian	8	[S04, S16, S37, S47, S64, S73, S77, S78]
	Linear discriminant analysis	5	[S15, S16, S55, S64, S67]
	Genetic algorithms	4	[S05, S63, S82, S86]
	Markov hidden model	2	[S43, S60]
	Baum-Welch algorithm	1	[S43]
	Natural language processing	1	[S28]
Clustering	K means	6	[S08, S38, S51, S62, S74, S82]
	Self Organization Map	4	[S08, S32, S35, S46]
	K*	2	[S32, S35]
	Clustering hierarchical	1	[S08]
	Multi Objective Evolutionary Fuzzy Classifier	1	[S01]
	Fuzzy C-means Clustering	1	[S62]
Dimensional reduction	Association rules	2	[S11, S59]
	Principal Component Analysis	1	[S53]
	Shatter Dempster theory	1	[S25]

Figure 3 shows the frequency of use for each paradigm, i.e., the number of studies that reported techniques from those paradigms. We observe from this figure that decision trees is the most frequently used paradigm (with 88 studies), followed by linear predictors (with 54 studies). The decision trees paradigm is based on the "divide and conquer" strategy. Each node in the tree represents a different classification attribute. The algorithm compares two nodes and decides where the new instance would better be classified. This process is applied at depth of travel, according to the size of the decision tree. This model is more applicable to nominal attributes [1]. The linear predictor paradigm, on the other hand, is more applicable to numerical attributes. These techniques must go through training so that the model can learn the relevant characteristics of the data. Then, the model is validated through another data set (the test set). Thus, the model can classify new data more accurately [1].

Not all authors report information about the configuration (implementation) of the techniques. It is important that primary studies report such information, so that they can be replicated and their results validated. Studies S04, S11, S14, S37, S43, S45, S47, S49, S55, and S73 report details of the configuration used.

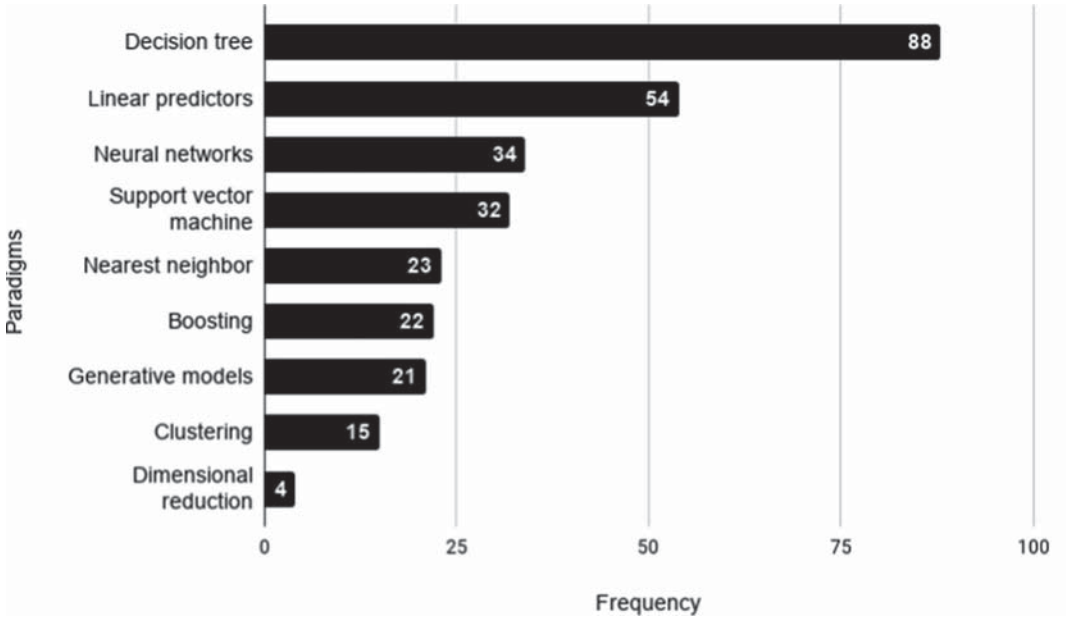


Fig. 2. Frequency of data mining and machine learning techniques.

Figure 3 shows the number of studies reported each year per paradigm. Paradigms are shown in the vertical axis, while years appear in the horizontal axis. The size of a circle is proportional to the number of studies that reported that paradigm in that year. From this figure it is hard to devise a trend in the data. Basically, there is no strong tendency of some paradigms to be used more then others over time, thus, their use has mainly been steady.

5.2 Tools that Support Data Mining and Machine Learning in Bank Customer Segmentation (RQ2)

The second research question seeks to identify the tools that support data mining and machine learning techniques in the context of bank customer segmentation. A total of 19 tools were identified.

Figure 4 shows the frequency of use for the tools that were reported more than once and Fig. 5 shows the timeline where the evolution of the tools reported in the literature is reviewed. It can be seen that Weka[1] was the most used tool, with a total of 13 studies that reported it. Weka implements techniques from the following paradigms: decision trees, linear predictors, clustering, support vector machines, boosting, generative models, nearest neighbor and neural networks. It also allows connection to various databases through a Java component or through text files with a specific extension for this software. In addition, it has a library with public data sets that contains different types of data specific to certain problems. It also contains a layer of data preparation (pre-processing) where filters, data normalizations, transformation and combination of attributes

[1] www.cs.waikato.ac.nz/~ml/weka/index.html.

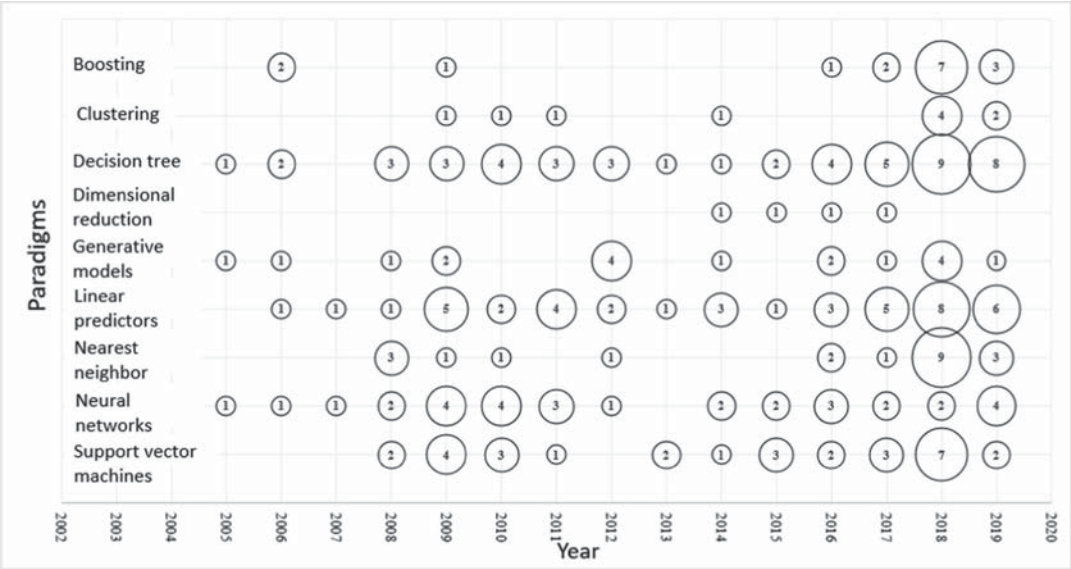


Fig. 3. Use of data mining and machine learning techniques per year.

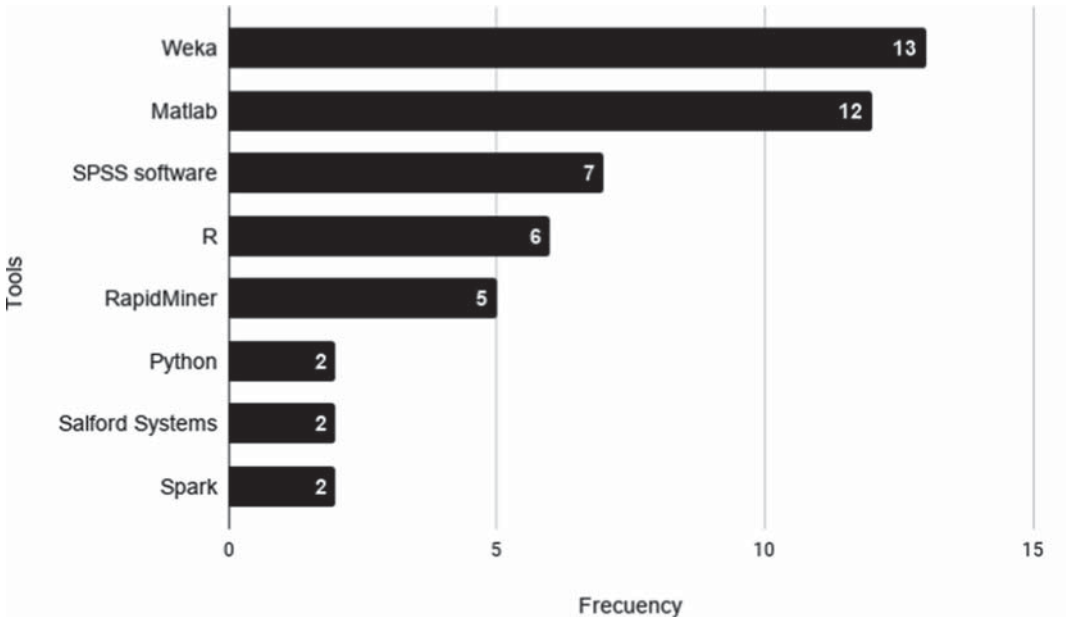


Fig. 4. Frequency of tools that support data mining and machine learning techniques.

can be made. Finally, it has a graphic layer with different schemes for comparing the output data in a basic way [1].

The next most reported tool was Matlab[2], with 12 references. This tool is oriented to the implementation of algorithms, as well as data modeling and simulation. It allows analysis, exploration and visualization of results. The programming language is based on matrices. In addition, it contains a collection of functions to facilitate the implementation of the models. Due to the nature of

[2] www.mathworks.com/products/matlab.html.

Fig. 5. Tools that support data mining and machine learning techniques per year.

the tool, it is possible to build techniques from all paradigms. This tool is used mainly in research and education because of the technical complexity required to use it, thus it is not easily applicable in industrial environments [10].

Figure 6 shows the relationship between tools and paradigms. The vertical axis represents tools reported more than once, and the horizontal axis represents data mining and machine learning paradigms for which tools were reported. An interesting finding was that decision trees is the only paradigm supported by all tools. Likewise, Matlab was the only tool that implemented techniques from all paradigms. Dimensional reduction was the only paradigm that reported no relation to the referenced tools.

Table 3 shows the 19 tools reported in the studies. This table contains the name of the tool, the frequency with which it is referenced in the studies and the reference number. It is important to note that 44 studies did not report the tools used during the experiments. These studies are: S44, S45, S46, S47, S48, S49, S50, S51, S52, S53, S54, S55, S56, S57, S58, S59, S60, S61, S62, S63, S64, S65, S66, S67, S68, S69, S70, S71, S72, S73, S74, S75, S76, S77, S78, S79, S80, S81, S82, S83, S84, S85, S86, and S87. The use of tools over time do not show a clear trend, so essentially remains constant from 2006 to 2019. If more studies reported the tools used, a better analysis of usage trends could be made in time.

We found that from the 19 tools identified, 10 are free license. In addition, it is important to clarify that these tools are divided in those that already have the data mining and machine learning techniques implemented, and those that don't (the researcher needs to implement the technique). Tools with implemented techniques are: Weka, SPSS, RapidMiner, Saldford system, Knime, Palisade, H2O, Powerhouse, KEEL and SVM Light. The other tools, although they use libraries that can help with the implementation of the algorithms, need an additional technical effort to develop the models.

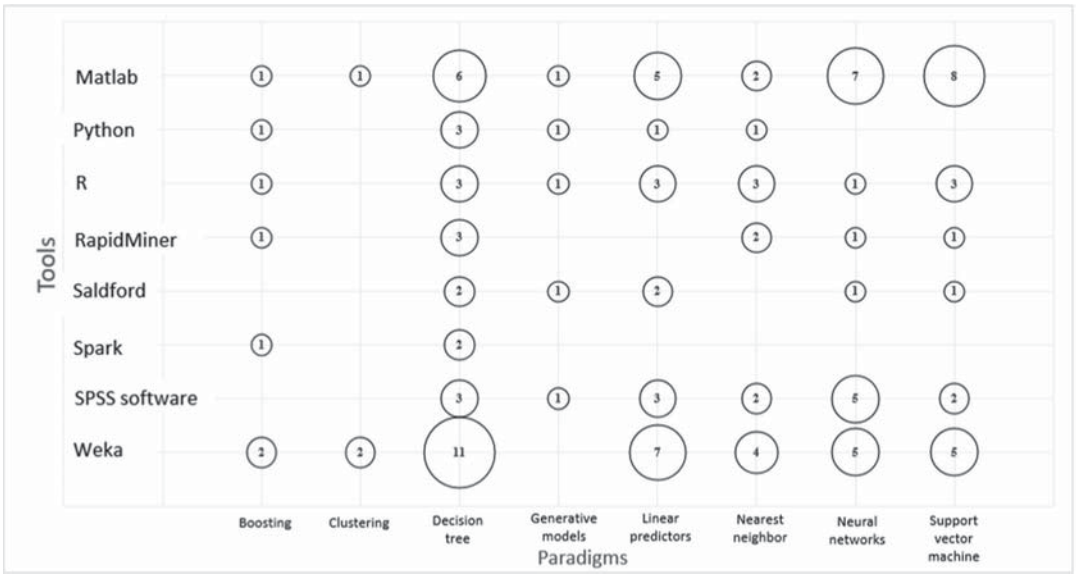


Fig. 6. Relationship between tools and paradigms.

Table 3. Tools that support data mining & machine learning techniques.

Tool	Qty	Studies
Weka	13	[S01, S02, S03, S09, S10, S29, S30, S31, S32, S33, S34, S35, S36]
Matlab	12	[S12, S16, S25, S27, S28, S37, S38, S39, S40, S41, S42, S43]
SPSS	7	[S12, S15, S24, S25, S26, S27, S28]
R	6	[S03, S07, S20, S21, S22, S23]
RapidMiner	5	[S03, S04, S17, S18, S19]
Python	2	[S13, S16]
Salford Systems	2	[S14, S15]
Spark	2	[S06, S13]
Knime	1	[S03]
Microsoft Excel	1	[S05]
Palisade	1	[S05]
H2O	1	[S06]
Powerhouse	1	[S07]
Orange	1	[S08]
Visual C++	1	[S09]
KEEL	1	[S10]
Java	1	[S11]
Lingo	1	[S12]
SVM Light	1	[S12]

5.3 Metrics and Data Sets Used to Evaluate Data Mining and Machine Learning Techniques in Bank Customer Segmentation (RQ3)

The third research question aims to identify commonly used performance metrics and datasets to evaluate data mining and machine learning techniques.

Of the studies analyzed, 5 did not report performance metrics: S08, S10, S11, S34, and S41. Table 4 shows the performance metrics reported in the primary studies, the number of times each metric was reported (used), and the study reference. The most frequently used metric was accuracy, cited by 66 studies. In second place we found recall, with 25 studies, and in third place, specificity with 20 studies.

We found that 60% of the primary studies did not report the metric formula. However, several of them do report the name and a theoretical description, so we used [1] to infer the metric and its formula. All performance metrics are based

Table 4. Performance metrics reported in primary studies.

Metric	Qty	Studies
Accuracy	66	[S01, S02, S03, S04, S05, S07, S09, S12, S14, S15, S16, S17, S18, S19, S20, S21, S22, S23, S25, S26, S27, S29, S30, S31, S33, S37, S38, S39, S40, S42, S43, S44, S45, S46, S47, S48, S49, S50, S51, S52, S53, S54, S55, S56, S57, S58, S60, S62, S66, S68, S69, S70, S71, S73, S74, S76, S77, S78, S79, S80, S81, S82, S83, S84, S85, S86]
Recall	25	[S01, S02, S03, S04, S06, S07, S09, S12, S13, S16, S17, S19, S20, S22, S28, S29, S32, S33, S49, S50, S51, S52, S62, S67, S78]
Specificity	20	[S03, S04, S07, S09, S12, S15, S16, S20, S22, S23, S29, S33, S49, S51, S59, S60, S61, S62, S67, S78]
ROC curve	16	[S01, S02, S18, S20, S30, S31, S33, S36, S40, S42, S44, S52, S55, S64, S65, S72]
Error rate	15	[S03, S04, S14, S21, S22, S23, S27, S28, S39, S51, S58, S60, S64, S75, S77]
F-measure	15	[S01, S02, S03, S09, S13, S16, S20, S24, S29, S32, S40, S52, S55, S62, S63]
Precision	15	[S01, S02, S03, S06, S09, S13, S16, S17, S19, S32, S46, S51, S52, S62, S74]
False positive rate	7	[S04, S15, S23, S52, S59, S60, S61]
True positive rate	5	[S04, S23, S59, S60, S61]
False negative rate	4	[S04, S59, S60, S61]

on the confusion matrix [9] terms: true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

Figure 7 shows the relationship between paradigms and metrics. The vertical axis displays the paradigms in which data mining and machine learning techniques were classified, while the horizontal axis corresponds to the reported performance metrics. From this figure, it can be seen that accuracy was the most frequently used metric, and the only one that was used to evaluate all 9 paradigms. Finally, the paradigms of decision trees, linear predictors, and neural networks were not only the most reported in general, but also the ones that got more evaluations reported.

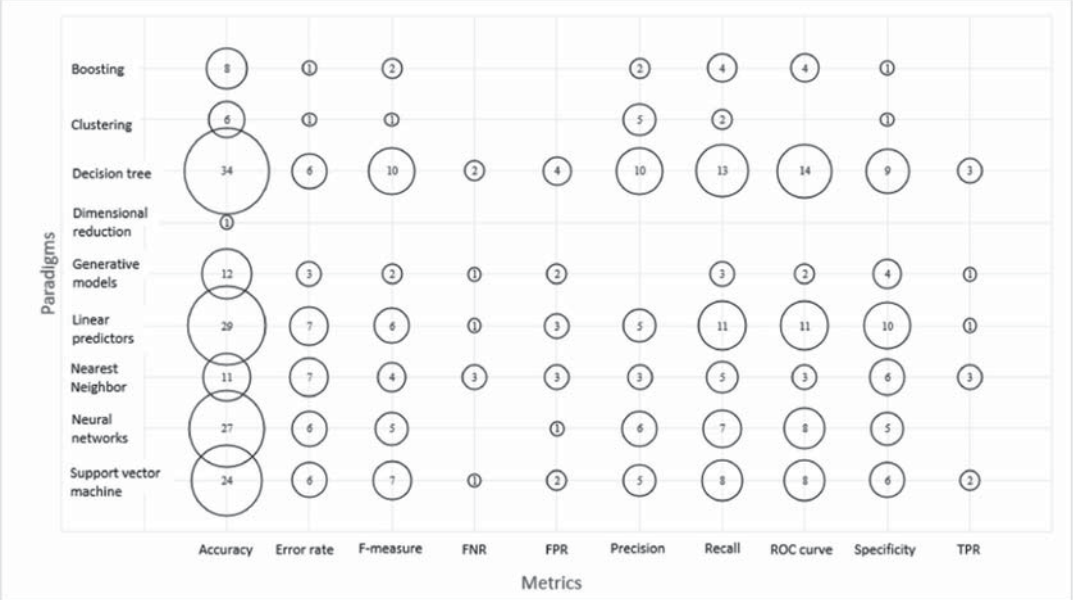


Fig. 7. Relationship between paradigms and metrics.

With respect to the data sets used to evaluate data mining and machine learning techniques in the context of bank customer segmentation, Table 5 summarizes the information reported by the primary studies. This table contains the name of the dataset, the number of records in the dataset, the number of times the dataset was reported in the studies, and the studies references. A trend of use can be noted in Table 5, where the Credit Data - UCI Machine Learning repository from the University of California[3] was reported 56 times. This repository is public and contains customer financial data for credits. In this repository there are data by countries, so some studies chose several data sets from different countries. This repository is multivariable, the attributes can be categorical or numerical, can be used for classification or prediction tasks, and depending on

[3] https://archive.ics.uci.edu/ml/index.php.

Table 5. Evaluation data sets used in primary studies.

Data set	Records	Qty	Studies
German Credit Data - UCI Machine Learning	1000	25	[S03, S05, S09, S10, S12, S29, S31, S32, S38, S39, S40, S43, S48, S53, S54, S56, S57, S61, S63, S71, S72, S73, S74, S75, S76]
Australia Credit Data - UCI Machine Learning	690	14	[S05, S10, S22, S29, S32, S38, S39, S40, S43, S53, S54, S57, S60, S71]
Credit Data - UCI Machine Learning	800000	5	[S04, S06, S16, S50, S52]
Commercial Bank in China	72544	5	[S14, S26, S35, S69, S70]
Japan Credit Data - UCI Machine Learning	653	4	[S29, S40, S53, S54]
Portugal Credit Data - UCI Machine Learning	39921	3	[S30, S42, S55]
China Credit Data - UCI Machine Learning	3111	3	[S53, S54, S68]
SPSS Credit Data - UCI Machine Learning	700	2	[S05, S54]
Turkish State Institution	59663	2	[S02, S67]
Bank marketing	11162	2	[S07, S13]
European Bank	180000000	2	[S65, S66]
Taiwan Bank	30000	2	[S51, S64]
Data modeling	50000	1	[S36]
Taipei Bank	8000	1	[S15]
Indonesian Bank	948	1	[S49]
SCF set	4245	1	[S20]
Atlanticus Services Corporation	12495	1	[S44]
Commercial bank	690	1	[S37]
Popular Superintendence of Economy of Ecuador	20000000	1	[S46]
Credit card data store	200	1	[S77]
Pathfinder platform	200000	1	[S08]
Iran Bank	4459	1	[S41]
Shenzhen Commercial Bank	4000	1	[S25]
Brazil Bank	4504	1	[S78]

the country, the number of instances varies. Besides, 17 private data sets were reported, from which it was possible to extract the amount of records used, the type of data (financial) and the nature of the datasets (multivariable).

The problem of customer segmentation had already been investigated, and there are trends on what techniques are most used, but it is not possible to determine which one is the best due to the differences in configurations and environments among the experiments.

6 Conclusions

The objective of this study was to apply a systematic mapping of primary studies that will evaluate data mining and machine learning techniques in the context of segmentation banking customers. These studies were published between 2005 and 2019. A set of 87 primary studies were analyzed, after applying exclusion and inclusion criteria. During this process a quality assessment was also applied, in order to verify which studies were most relevant for the research.

The results showed a wide range of techniques used by studies in the area, where each configuration that is applied can vary the results obtained. According to the classification used during the investigation, the most reported paradigms were decision trees with 88 references and linear predictors with 54 references. The majority of studies do not report the parameterization or configuration used during their evaluations. This is something that we recommend to include in primary studies, because in this way replications could be made or improvements can be made to the techniques or experiments proposed, maintaining the same conditions and environment.

With respect to the tools that support data mining and machine learning techniques, a clear trend was not obtained, since throughout the years in which the studies were reported there were already several tools that have the algorithms implemented and should only be configured depending on what you are working on or also tools that allow you to implement the techniques based on existing specialized libraries. With respect to the type of tools according to the licensing, it was seen that almost half are open source and the other half paid license. The most reported tools were Weka with 13 references and Matlab with 12 references. In the case of Weka, they already have the techniques implemented and it is a free license, on the contrary, Matlab is a paid license and the algorithms must be implemented.

This study also proposed identifying the performance metrics that are most evaluated. It was validated through research and theory that there are already standard performance metrics used in this type of studies. From the metrics there was a tendency to use accuracy that was reported in 66 studies of 87 analyzed.

This research can be applied for different areas and industries. In the case of the professional area, for companies that do not have the specialized resources in these technologies, it can be a good start to take as a reference studies that have already researched on the subject and have an idea of where to start the experimentation of models. In the case of education, the specialized offer of

higher education institutions in the fields of data mining and machine learning must increase. These types of studies can be the basis for defining topics of specialized courses in these areas. In the case of the academic research area, this study offers an overview of what has been worked on in the literature in recent years in the area. In addition, it groups data mining and machine learning techniques, tools, metrics and data sets. This is important because it had not been work in any other study.

As a future work, it is proposed to make a comparison between the techniques of data mining and machine learning that were most reported. Taking into account that the evaluation must be done under the same conditions, that is, use the same data set, apply the same data preprocessing, choose a tool for the case study and apply the performance metrics under the same parameters. This would apply as a primary study.

Another point for future work, its apply a similar study in another context or industry. The reason is to have alike work and compare the results. This comparison can review if the use of techniques, tools, metrics and data sets depends of context that we work.

References

1. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers, Burlington (2005)
2. Miyan, M.: Applications of data mining in banking sector. Int. J. Adv. Res. Comput. Sci. **8**(1), 109–114 (2017)
3. Hasheminejad, S.M.H., Khorrami, M.: Data mining techniques for analyzing bank customers: a survey. Intell. Decis. Technol. **12**(3), 303–321 (2018)
4. Bhambri, V.: Application of data mining in banking sector. Desh Bhagat Inst. Manag. Comput. Sci. **2**(2), 199–202 (2011)
5. Pulakkazhy, S., Balan, R.V.S.: Data mining in banking and its applications a review. J. Comput. Sci. **9**(10), 1252–1259 (2013). Cited By: 21
6. Goebel, M., Gruenwald, L.: A survey of data mining and knowledge discovery software tools. SIGKDD Explor. **1**(1), 20–33 (1999)
7. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: an update. Inf. Softw. Technol. **64**, 1–18 (2015)
8. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering version 2.3. Engineering **45**(4ve), 1051 (2007)
9. Han, J., Kamber, M., Pei, J.: Data mining: Concepts and Techniques (2012)
10. Matworks - documentation. https://la.mathworks.com/help/matlab/index.html. Accessed 01 Dec 2019
11. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms, vol. 9781107057135, pp. 1–397 (2013). Cited By: 459

Bibliografía del capítulo

[1] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2005.

[2] M. Miyan, "Applications of data mining in banking sector," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 1, pp. 109–114, 2017.

[3] S. Hasheminejad and M. Khorrami, "Data mining techniques for analyzing bank customers: A survey," *Intelligent Decision Technologies*, vol. 12, no. 3, pp. 303–321, 2018.

[4] V. Bhambri, "Application of data mining in banking sector," *Desh Bhagat Institute of Management and Computer Sciences*, vol. 2, no. 2, pp. 199–22, 2011.

[5] V. Jayasree, R. Vijayalakshmi, and S. Balan, "A review on data mining in banking sector," *American Journal of Applied Sciences*, pp. 1160–1165, 2013.

[6] H. Jiawei, K. Micheline, and J. P, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.

[7] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, vol. 9781107057135 of *Understanding Machine Learning: From Theory to Algorithms*, pp. 1–397. 2013. Cited By :459.

[8] S. Pulakkazhy and R. Balan, "Data mining in banking and its applications- a review," *Journal of Computer Science*, vol. 9, no. 10, pp. 1252–1259, 2013. Cited By :21.

[9] M. Goebel and L. Gruenwald, "A survey of data mining and knowledge discovery software tools," *SIGKDD Explorations*, vol. 1, no. 1, pp. 20–33, 1999.

[10] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[11] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3," *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.

[12] V. Basili, G. Caldiera, and D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

[13] N. Gulsoy and S. Kulluk, "A data mining application in credit scoring processes of small and medium enterprises commercial corporate customers," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 3, 2019.

[14] B. Çiğşar and D. Ünal, "Comparison of data mining classification algorithms determining the default risk," *Scientific Programming*, vol. 2019, 2019.

[15] M. Basarslan and I. Argun, "Classification of a bank data set on various data mining platforms," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting, EBBT 2018*, pp. 1–4, 2018.

[16] T. Das, "A customer classification prediction model based on machine learning techniques," in *Proceedings of the 2015 International Conference on Applied and Theoretical Computing and Communication Technology*, pp. 321–326, 2016. Cited By :3.

[17] B. Kitchenham, E. Mendes, and G. Travassos, *A Systematic Review of Cross- vs. Within-Company Cost Estimation Studies*. IEEE Trans on SE, 2007.

[18] P. Hung, T. Hanh, and T. Tung, "Term deposit subscription prediction using spark mllib and ml packages," in *ACM International Conference Proceeding Series*, pp. 88–93, 2019.

[19] Y. Wang, X. Ni, and B. Stone, "An automatic interaction detection hybrid model for bankcard response classification," in *2018 5th International Conference on Systems and Informatics, ICSAI 2018*, pp. 1111–1119, 2019.

[20] T. Pandey, A. Jagadev, S. Mohapatra, and S. Dehuri, "Credit risk analysis using machine learning classifiers," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, pp. 1850–1854, 2018.

[21] R. Rajamohamed and J. Manokaran, "Improved credit card churn prediction based on rough clustering and supervised learning techniques," *Cluster Computing*, vol. 21, no. 1, 2018. Cited By :2.

[22] S. Ghosh, A. Hazra, B. Choudhury, P. Biswas, and A. Nag, *A Comparative Study to the Bank Market Prediction*, vol. 10934 LNAI of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2018. Cited By :1.

[23] Y. Guo, J. He, L. Xu, and W. Liu, "A novel multi-objective particle swarm optimization for comprehensible credit scoring," *Soft Computing*, vol. 23, no. 18, pp. 9009–9023, 2019. Cited By :1.

[24] A. Gahlaut, K. Tushar, and P. Singh, "Prediction analysis of risky credit using data mining classification models," in *8th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2017*, 2017. Cited By :1.

[25] R. Turkson, E. Baagyere, and G. Wenya, "A machine learning approach for predicting bank credit worthiness," in *2016 3rd International Conference on Artificial Intelligence and Pattern Recognition, AIPR 2016*, pp. 81–87, 2016. Cited By :4.

[26] M. AlaRaj and M. Abbod, "Classifiers consensus system approach for credit scoring," *Knowledge-Based Systems*, vol. 104, pp. 89–105, 2016. Cited By :46.

[27] M. Amini, J. Rezaeenour, and E. Hadavandi, "A cluster-based data balancing ensemble classifier for response modeling in bank direct marketing," *International Journal of Computational Intelligence and Applications*, vol. 14, no. 4, 2015. Cited By :3.

[28] P. Marikkannu and K. Shanmugapriya, "Classification of customer credit data for intelligent credit scoring system using fuzzy set and mc2 - domain driven

approach," in *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology*, vol. 3, pp. 410–414, 2011. Cited By :3.

[29] W. Li and J. Liao, "An empirical study on credit scoring model for credit card by using data mining technology," in *Proceedings - 2011 7th International Conference on Computational Intelligence and Security, CIS 2011*, pp. 1279–1282, 2011. Cited By :6.

[30] W. Li, X. Wu, Y. Sun, and Q. Zhang, "Credit card customer segmentation and target marketing based on data mining," in *Proceedings - 2010 International Conference on Computational Intelligence and Security, CIS 2010*, pp. 73–76, 2010. Cited By :16.

[31] J. He, Y. Zhang, Y. Shi, and G. Huang, "Domain-driven classification based on multiple criteria and multiple constraint-level programming for intelligent credit scoring," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 6, pp. 826–838, 2010. Cited By :28.

[32] H. Xie, S. Han, X. Shu, X. Yang, X. Qu, and S. Zheng, "Solving credit scoring problem with ensemble learning: A case study," in *2009 2nd International Symposium on Knowledge Acquisition and Modeling, KAM 2009*, vol. 1, pp. 51–54, 2009. Cited By :2.

[33] A. Tsakonas, N. Ampazis, and G. Dounias, "Towards a comprehensible and accurate credit management model: Application of four computational intelligence methodologies," in *Proceedings of the 2006 International Symposium on Evolving Fuzzy Systems, EFS'06*, pp. 295–299, 2006. Cited By :1.

[34] X. Hu, "A data mining approach for retailing bank customer attrition analysis," *Applied Intelligence*, vol. 22, no. 1, pp. 47–60, 2005. Cited By :46.

[35] W. Ying, X. Li, Y. Xie, and E. Johnson, "Preventing customer churn by using random forests modeling," in *2008 IEEE International Conference on Information Reuse and Integration, IEEE IRI-2008*, pp. 429–434, 2008. Cited By :7.

[36] Z. Chen, "The application of tree-based model to unbalanced german credit data analysis," in *MATEC Web of Conferences*, vol. 232, 2018.

[37] A. Chopra and P. Bhilare, "Application of ensemble models in credit scoring models," *Business Perspectives and Research*, vol. 6, no. 2, pp. 129–141, 2018. Cited By :1.

[38] I. Livieris, N. Kiriakidou, A. Kanavos, V. Tampakas, and P. Pintelas, "On ensemble ssl algorithms for credit scoring problem," *Informatics*, vol. 5, no. 4, 2018. Cited By :2.

[39] I. Pawełoszek and J. Korczak, "From data exploration to semantic model of customer," in *2017 Intelligent Systems Conference, IntelliSys 2017*, vol. 2018-January, pp. 382–388, 2018.

[40] D. Liang, C. Tsai, A. Dai, and W. Eberle, "A novel classifier ensemble approach for financial distress prediction," *Knowledge and Information Systems*, vol. 54, no. 2, pp. 437–462, 2018. Cited By :4.

[41] T. Yang, K. Qian, D. Lo, Y. Xie, Y. Shi, and L. Tao, "Improve the prediction accuracy of naïve bayes classifier with association rule mining," in *Proceedings - 2nd IEEE International Conference on Big Data Security on Cloud, IEEE BigDataSecurity 2016, 2nd IEEE International Conference on High Performance and Smart Computing, IEEE HPSC 2016 and IEEE International Conference on Intelligent Data and Security, IEEE IDS 2016*, 2016.

[42] S. Dahiya, S. Handa, and N. Singh, "Credit modelling using hybrid machine learning technique," in *International Conference on Soft Computing Techniques and Implementations, ICSCTI 2015*, pp. 103–106, 2016. Cited By :3.

[43] F. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring," *Journal of Retailing and Consumer Services*, vol. 27, pp. 11–23, 2015. Cited By :42.

[44] E. Kamalloo and M. Saniee-Abadeh, "Credit risk prediction using fuzzy immune learning," *Advances in Fuzzy Systems*, 2014. Cited By :2.

[45] G. Wang, L. Liu, Y. Peng, G. Nie, G. Kou, and Y. Shi, "Predicting credit card holder churn in banks of china using data mining and mcdm," in *Proceedings -*

2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2010, pp. 215–218, 2010. Cited By :10.

[46] P. Ladyzynski, K. Zbikowski, and P. Gawrysiak, "Direct marketing campaigns in retail banking with the use of deep learning and random forests," *Expert Systems with Applications*, vol. 134, pp. 28–35, 2019.

[47] L. Munkhdalai, T. Munkhdalai, O. Namsrai, J. Lee, and K. Ryu, "An empirical comparison of machine-learning methods on bank client credit assessments," *Sustainability (Switzerland)*, vol. 11, no. 3, 2019. Cited By :2.

[48] J. Asare-Frempong and M. Jayabalan, "Predicting customer response to bank direct telemarketing campaign," in *2017 International Conference on Engineering Technology and Technopreneurship, ICE2T 2017*, vol. 2017-January, pp. 1–4, 2017. Cited By :2.

[49] H. Wang, J. Zhong, D. Zhang, and X. Zou, "A new classification algorithm for the bank customer credit rating," in *9th International Conference on Advanced Computational Intelligence, ICACI 2017*, pp. 143–148, 2017.

[50] C. Devi and R. Manicka-Chezian, "A relative evaluation of the performance of ensemble learning in credit scoring," in *2016 IEEE International Conference on Advances in Computer Applications, ICACA 2016*, pp. 161–165, 2017. Cited By :3.

[51] I. Noviandi and I. Sumitra, "Classification consumer credit for missing value dataset," in *IOP Conference Series: Materials Science and Engineering*, vol. 407, 2018.

[52] Y. Wah and I. Ibrahim, "Using data mining predictive models to classify credit card applicants," in *Proc. - 6th Intl. Conference on Advanced Information Management and Service, IMS2010, with ICMIA2010 - 2nd International Conference on Data Mining and Intelligent Information Technology Applications*, pp. 394–398, 2010. Cited By :8.

[53] W. Chen, C. Ma, and L. Ma, "Mining the customer credit using hybrid support vector machine technique," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7611–7616, 2009. Cited By :63.

[54] I. Yeh and C. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2 PART 1, pp. 2473–2480, 2009. Cited By :130.

[55] T. Lee, C. Chiu, Y. Chou, and C. Lu, "Mining the customer credit using classification and regression tree and multivariate adaptive regression splines," *Computational Statistics and Data Analysis*, vol. 50, no. 4, pp. 1113–1130, 2006. Cited By :206.

[56] G. Marinakos and S. Daskalaki, "Imbalanced customer classification for bank direct marketing," *Journal of Marketing Analytics*, vol. 5, no. 1, pp. 14–30, 2017. Cited By :1.

[57] K. Kennedy, B. Namee, and S. Delany, "Using semi-supervised classifiers for credit scoring," *Journal of the Operational Research Society*, vol. 64, no. 4, pp. 513–529, 2013. Cited By :13.

[58] F. Cai, N. Lekhac, and M. Kechadi, "Toward a new classification model for analysing financial datasets," in *7th International Conference on Digital Information Management, ICDIM 2012*, pp. 1–6, 2012.

[59] Y. Jiang, X. Zhou, and D. Zhang, "A new approach based on a rough set and a decision tree to bank customer credit evaluation," in *Proceedings of 2008 IEEE International Symposium on IT in Medicine and Education, ITME 2008*, pp. 61–65, 2008. Cited By :2.

[60] Y. Jiang, Y. Chen, Z. Zeng, and X. He, "A bank customer credit evaluation based on the decision tree and the simulated annealing algorithm," in *Proceedings - 2008 IEEE 8th International Conference on Computer and Information Technology, CIT 2008*, pp. 203–206, 2008.

[61] N. Barraza, S. Moro, M. Ferreyra, and A. de la Peña, "Mutual information and sensitivity analysis for feature selection in customer targeting: A comparative

study," *Journal of Information Science*, vol. 45, no. 1, pp. 53–67, 2019. Cited By :3.

[62] M. Bizhani and M. Tarokh, "Behavioral rules of bank's point-of-sale for segments description and scoring prediction," *International Journal of Industrial Engineering Computations*, vol. 2, no. 2, pp. 337–350, 2011. Cited By :7.

[63] S. Vukovic, B. Delibasic, A. Uzelac, and M. Suknovic, "A case-based reasoning model that uses preference theory functions for credit scoring," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8389–8395, 2012. Cited By :31.

[64] M. Adha, S. Nurrohmah, and S. Abdullah, "Multinomial logistic regression and spline regression for credit risk modelling," in *Journal of Physics: Conference Series*, vol. 1108, 2018.

[65] H. Bekhet and S. Eletter, "Credit risk assessment model for jordanian commercial banks: Neural scoring approach," *Review of Development Finance*, vol. 4, no. 1, pp. 20–28, 2014. Cited By :38.

[66] M. Jiang and J. Hu, "Combining multiple classifiers based on dempster-shafer theory for personal credit scoring," in *International Conference on Management Science and Engineering - Annual Conference Proceedings*, pp. 167–172, 2014. Cited By :1.

[67] L. Hui, S. Li, and Z. Zongfang, "The model and empirical research of application scoring based on data mining methods," in *Procedia Computer Science*, vol. 17, pp. 911–918, 2013. Cited By :1.

[68] F. Louzada, P. Ferreira-Silva, and C. Diniz, "On the impact of disproportional samples in credit scoring models: An application to a brazilian bank data," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8071–8078, 2012. Cited By :15.

[69] B. Waad, B. Farid, and G. Bel-Mufti, "Logistic sub-models for small size populations in credit scoring," in *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIDM 2011: 2011 IEEE Symposium on Computational Intelligence and Data Mining*, pp. 128–134, 2011.

[70] H. Oguz and F. Gurgen, "Credit risk analysis using hidden markov model," in *2008 23rd International Symposium on Computer and Information Sciences, ISCIS 2008*, 2008. Cited By :8.

[71] L. Abid, A. Masmoudi, and S. Zouari-Ghorbel, "The consumer loan's payment default predictive model: an application of the logistic regression and the discriminant analysis in a tunisian commercial bank," *Journal of the Knowledge Economy*, vol. 9, no. 3, pp. 948–962, 2018. Cited By :1.

[72] C. Guotai, M. Abedin, and F. Moula, "Modeling credit approval data with neural networks: an experimental investigation and optimization*," *Journal of Business Economics and Management*, vol. 18, no. 2, pp. 224–240, 2017. Cited By :1.

[73] M. Alborzi and M. Khanbabaei, "Using data mining and neural networks techniques to propose a new hybrid customer behaviour analysis and credit scoring model in banking services based on a developed rfm analysis method," *International Journal of Business Information Systems*, vol. 23, no. 1, pp. 1–22, 2016. Cited By :7.

[74] P. Wongchinsri and W. Kuratach, "Sr-based binary classification in credit scoring," in *ECTI-CON 2017 - 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 385–388, 2017.

[75] Y. Chen, L. Zhang, and Y. Shi, "Post mining of multiple criteria linear programming classification model for actionable knowledge in credit card churning management," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 204–211, 2011. Cited By :3.

[76] A. Dima and S. Vasilache, "Ann model for corporate credit risk assessment," in *Proceedings - 2009 International Conference on Information and Financial Engineering, ICIFE 2009*, pp. 94–98, 2009. Cited By :4.

[77] Q. Chen, D. Zhang, L. Wei, and H. Chen, "A modified genetic programming for behavior scoring problem," in *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2007*, pp. 535–539, 2007. Cited By :9.

[78] M. Zhao, "Credit risk assessment based on fuzzy svm and principal component analysis," in *2009 International Conference on Web Information Systems and Mining, WISM 2009*, pp. 125–127, 2009. Cited By :3.

[79] E. Tobback and D. Martens, "Retail credit scoring using fine-grained payment data," *Journal of the Royal Statistical Society.Series A: Statistics in Society*, 2019.

[80] S. Oreski, D. Oreski, and G. Oreski, "Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12605–12617, 2012. Cited By :72.

[81] N. Sarlija, M. Bensic, and M. Zekic-Susac, "Comparison procedure of predicting the time to default in behavioural scoring," *Expert Systems with Applications*, vol. 36, no. 5, pp. 8778–8788, 2009. Cited By :19.

[82] E. Dawood, E. Elfakhrany, and F. Maghraby, "Improve profiling bank customer's behavior using machine learning," *IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC*, vol. 7, pp. 109320–109327, 2019.

[83] Z. Jianguo, Z. Aiguang, and B. Tao, "Client classification on credit risk using rough set theory and aco-based support vector machine," in *2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008*, 2008. Cited By :3.

[84] P. Santana, L. Lanzarini, and A. Bariviera, "Fuzzy credit risk scoring rules using frvarpso," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 26, pp. 39–57, 2018.

[85] S. Moradi and F. Mokhatab-Rafiei, "A dynamic credit risk assessment model with data mining techniques: evidence from iranian banks," *Financial Innovation*, vol. 5, no. 1, 2019. Cited By :2.

[86] A. Lawi, A. Velayaty, and Z. Zainuddin, "On identifying potential direct marketing consumers using adaptive boosted support vector machine," in *Proceedings*

of the 2017 4th International Conference on Computer Applications and Information Processing Technology, CAIPT 2017, vol. 2018-January, pp. 1–4, 2018. Cited By :1.

[87] D. Wang, Z. Zhang, R. Bai, and Y. Mao, "A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring," *Journal of Computational and Applied Mathematics*, vol. 329, pp. 307–321, 2018. Cited By :5.

[88] B. Yi and J. Zhu, "Credit scoring with an improved fuzzy support vector machine based on grey incidence analysis," in *Proceedings of IEEE International Conference on Grey Systems and Intelligent Services, GSIS*, vol. 2015-October, pp. 173–178, 2015. Cited By :5.

[89] S. Javaheri, M. Sepehri, and B. Teimourpour, *Response Modeling in Direct Marketing. A Data Mining-Based Approach for Target Selection.*, pp. 153–180. Data Mining Applications with R, 2013. Cited By :5.

[90] Z. Huang, G. Duan, and J. Wang, "A method combined of support vector machine and f-scores for customer classification," in *Proceedings - 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010*, vol. 6, pp. 2702–2705, 2010.

[91] C. Hsu and H. Hung, "Classification methods of credit rating - a comparative analysis on svm, mda and rst," in *Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, 2009. Cited By :3.

[92] T. Fajrin, R. Saputra, and I. Waspada, "Credit collectibility prediction of debtor candidate using dynamic k-nearest neighbor algorithm and distance and attribute weighted," in *2018 2nd International Conference on Informatics and Computational Sciences, ICICoS 2018*, pp. 7–12, 2019.

[93] V. Mihova and V. Pavlov, "A customer segmentation approach in commercial banks," in *AIP Conference Proceedings*, vol. 2025, 2018.

[94] G. Arutjothi and C. Senthamarai, "Prediction of loan status in commercial bank using machine learning classifier," in *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2017*, pp. 416–419, 2018.

[95] M. Mukid, T. Widiharih, A. Rusgiyono, and A. Prahutama, "Credit scoring analysis using weighted k nearest neighbor," in *Journal of Physics: Conference Series*, vol. 1025, 2018. Cited By :2.

[96] Q. Zhou, C. Lin, and W. Yang, "Multi-classifier combination for banks credit risk assessment," in *2006 1st IEEE Conference on Industrial Electronics and Applications*, 2006. Cited By :1.

[97] O. Okesola, K. Okokpujie, A. Adewale, S. John, and O. Omoruyi, "An improved bank credit scoring model: A naïve bayesian approach," in *Proceedings - 2017 International Conference on Computational Science and Computational Intelligence, CSCI 2017*, pp. 228–233, 2018. Cited By :1.

[98] S. Ginting, J. Adler, Y. Ginting, and A. Kurniadi, "The development of bank application for debtors selection by using naïve bayes classifier technique," in *IOP Conference Series: Materials Science and Engineering*, vol. 407, 2018.

[99] B. Benyacoub, S. Bernoussi, and A. Zoglat, "Building classification models for customer credit scoring," in *Proceedings of 2nd IEEE International Conference on Logistics Operations Management, GOL 2014*, pp. 107–111, 2014. Cited By :3.

[100] A. Aribowo and N. Cahyana, "Feasibility study for banking loan using association rule mining classifier," *International Journal of Advances in Intelligent Informatics*, vol. 1, no. 1, pp. 41–47, 2015. Cited By :1.

[101] "Matworks - documentation." https://la.mathworks.com/help/matlab/index.html. Accessed: 2019-12-01.

Capítulo 4

Técnicas de aprendizaje automático y minería de datos para la clasificación de noticias Web: un mapeo de literatura

Mauricio Pandolfi González

4.1. Resumen

Contexto: las noticias descargadas desde la web han tomado gran importancia como fuente de información ciudadana, y su clasificación puede mostrar datos relevantes sobre patrones sociales o culturales de una sociedad. En el campo de la informática, la minería de datos sobre noticias permite utilizar técnicas especializadas para el procesamiento de los textos, que luego abstraen información trascendental para su clasificación temática y análisis posterior. **Objetivo:** el objetivo de este estudio es analizar las técnicas de minería de datos utilizadas para la clasificación de contenidos de noticias extraídas de la web. **Método:** esta investigación desarrolla un mapeo sistemático de 51 estudios primarios, siguiendo los lineamientos de Petersen [1]. Para cada estudio, se identifican tres aspectos esenciales: las técnicas utilizadas y su configuración, las características de los datos usados y las métricas con las que

se han sido evaluados. **Resultados:** en general, las técnicas de *clustering*, de *support vector machines* y *generative models* son las más empleadas, mientras que la métrica más utilizada para evaluar fue *F-measure*. **Conclusiones:** los estudios no muestran consenso en la respuesta de ninguna de las tres preguntas de investigación en cuanto a la especificación de las técnicas, los datos y las métricas. No obstante, el trabajo expone un mapeo de técnicas que pueden ser utilizadas desde la academia, la industria y la investigación.

4.2. Introducción

El estudio de las noticias puede ser determinante para el contexto socio-político de la sociedad. En un momento en que las noticias se consumen virtualmente, hacer un análisis temático a partir de una clasificación puede ayudar a comprender conductas y descubrir patrones sociales inmersos en nuestra cultura [2].

La incursión de nuevas tecnologías en los últimos años, tales como la móvil y las redes sociales, ha generado novedosos conceptos de comunicación como la convergencia digital, el periodismo 3.0 y el periodismo multimedia. Estos cambios han creado un nuevo paradigma, donde se han hecho propuestas de innovadores modelos para la redacción, la edición, y la producción de materiales periodísticos [3].

Con los cambios presentados en el campo del periodismo, la producción de noticias, su complejidad y su volumen han aumentado en gran medida, esto hace que surja la necesidad de hacer análisis temáticos de la información [4], lo cual solo puede darse por medio de la minería de datos, pues sus técnicas hacen posible la detección de patrones estructurales sobre los documentos, considerando además, su avanzada complejidad y creciente volumen [5, 1].

Hoy en día, los sistemas automáticos que analizan y clasifican artículos de noticias web son esenciales para el manejo de estas y para generar recomendaciones al usuario [4]. Las técnicas de minería de datos y aprendizaje automático hacen posible la detección de patrones estructurales en los documentos, basándose en sus características [5, 1] que crean retos lingüísticos y computacionales para la minería de datos: (i) indexación, categorización según taxonomías, redundancia parcial, y flujos de datos, (ii) lenguaje y significado, (iii) lenguaje no estándar y subjetividad, (iv) diversidad

temática y nuevas formas de categorización y (v) contexto y su impacto en contenido y significado [7].

En los años recientes, ha habido múltiples esfuerzos relacionados con la clasificación, agrupamiento, categorización y resumen de artículos de noticias [4]. El objetivo de esta investigación es caracterizar las técnicas de aprendizaje automático y minería de datos que se han usado para la clasificación de contenidos noticiosos extraídos de la web.

Para el desarrollo de este estudio, se realizó un mapeo sistemático de literatura, a fin de identificar las técnicas que utilizan aprendizaje automático y minería de datos para clasificación temática de noticias web, desde el punto de vista de la caracterización de las técnicas, los conjuntos de datos empleados y las métricas que se usaron para medir su efectividad.

El capítulo se organiza en varias secciones. La sección 4.2 introduce el capítulo. La sección 4.3 presenta el marco teórico. La sección 4.4 muestra el trabajo relacionado con este estudio. En la sección 4.5 se detalla la metodología empleada. En la sección 4.6 se presentan los resultados y en la sección 4.7 la discusión. Finalmente, en las secciones 4.8 y 4.9 se detallan las lecciones aprendidas y las conclusiones, respectivamente.

4.3. Marco teórico

En esta sección se describen los fundamentos teóricos sobre los cuales se basa esta investigación.

La Ingeniería de *Software* abarca el problema de la clasificación de documentos de forma que busca hallar las características que los distinguen. Tomando en cuenta que los documentos se caracterizan por las palabras que contienen, la presencia o ausencia de cada palabra puede ser tratada como un valor booleano al extraer información. Otra alternativa es tratar los documentos como conjuntos de palabras que tomen en cuenta las frecuencias de cada una de estas [1].

La clasificación de documentos se ha abordado mediante técnicas de aprendizaje supervisado y no supervisado. En el caso del aprendizaje supervisado, las categorías se conocen de antemano y están dadas para la fase de entrenamiento. En cuanto

al aprendizaje no supervisado, no se tienen clases predefinidas, sino que se buscan grupos de documentos afines, mediante técnicas de agrupación [1].

La clasificación tiene sus raíces en muchas áreas, incluyendo minería de datos, estadística, biología y aprendizaje automático. Esto refleja su gran atractivo y utilidad como un paso importante en el análisis de datos exploratorio, agrupamiento, toma de decisiones, minería de datos, recuperación de información, segmentación de imagen y clasificación de patrones [6].

El aprendizaje automático es la rama de la informática que trata sobre el uso de algoritmos para inferir estructuras de los datos y formas de validar esa estructura. No se trata de técnicas triviales, sino de soluciones que requieren una personalización y detalle técnico [1]. En este campo de estudio, se desarrollan sistemas que utilizan técnicas para automatizar procesos que emulan una comprensión de los datos sobre los que se trabaja. Por ello, el campo de la minería sobre grandes conjuntos de datos se relaciona con el aprendizaje automático [1].

Por su parte, la minería de datos trabaja para poder ver información que está, de alguna forma, escondida entre los datos que se estudian, de modo que los resultados obtenidos permitan resolver problemas. Estos procesos deben ser automáticos o semiautomáticos. El campo de minería en informática es un tema práctico sobre los datos, por lo que involucra el aprendizaje de los sistemas [1].

No hay una clara línea divisora entre los conceptos de aprendizaje automático y minería de datos. La diferencia más clara entre ellas es que el aprendizaje automático tiene como fin último un resultado, mientras que en la minería de datos este es el conocimiento generado [6]. No obstante, algunos algoritmos pueden ser clasificados en cualquiera de los dos grupos.

De igual forma, el campo de la minería de datos lleva implícito un proceso de aprendizaje para que los algoritmos puedan inferir, y el campo de aprendizaje automático requiere leer patrones en los datos. Por ello, ambas técnicas son complementarias. Algunas veces, la minería de datos es presentada como una derivación de aprendizaje automático [1].

Para el interés de este estudio, que consiste en las técnicas para encontrar clasificaciones temáticas de noticias, no se plantean diferencias significativas conceptuales que las clasifique únicamente en una de las dos ramas.

Como una extensión de minería de datos, se encuentra la minería de texto, un campo enfocado en descubrir patrones cuando los datos son textos. Esto se define como el proceso en el que se analizan textos para extraer de ellos información útil para un propósito particular [1]. El texto tiene características que pueden hacer complejo el análisis, pues se trata de información no estructurada, amorfa y difícil de manejar. No obstante, también el texto es un vehículo de gran importancia para intercambiar información, por lo que este campo puede resultar muy beneficioso en términos de poder analizar patrones [1].

El área de minería de texto está posicionada entre recuperación de información, minería de datos, procesamiento de lenguaje natural y aprendizaje automático. Incluye preprocesamiento de texto, detección automática de lenguaje, agrupación de texto y otros [9].

La minería de texto se usa para tareas como la descripción, clasificación, predicción, búsqueda, recomendación, y resumen de partes textuales de noticias y blogs, para extraer temas, opiniones, sentimientos y otros aspectos de contenido [7].

La *World Wide Web (WWW)* es una enorme base de datos, pero mucha de la información que se encuentra ahí no es simplemente texto plano, más bien, los datos están etiquetados por marcas estructurales como HTML o XML. Algunas de estas etiquetas son para uso interno e indican el formato que se sigue en cada página, mientras que otras son de uso externo y definen relaciones entre documentos. Por eso, tratar con documentos de la WWW adiciona complejidad al análisis [1]. En el caso del presente estudio, esta complejidad está presente en los datos por tratarse de noticias extraídas de la web y, por lo tanto, las técnicas utilizadas deben contemplarlo como parte de su aplicación. Esta información adicional es común en los datos y debe diferenciarse, tratarse u omitirse.

El área llamada *Web mining* se refiere al uso de minería de texto, pero en contextos donde se debe incorporar la información adicional que brindan los documentos que se abstraen de la web [1].

Debido a estas necesidades inherentes por el contexto web, desde el año 1999 se empezó a desarrollar una nueva área en la comunidad de recuperación, llamada *Topic Detection and Tracking* (TDT) [1]. Esta área trabaja sobre noticias relacionadas con eventos que se ordenan cronológicamente. Su objetivo consiste en poder detectar un

hecho a partir de un conjunto de historias y rastrearlo, es decir, darle un seguimiento al relacionarlo con otros eventos con el fin de identificar relaciones entre noticias. La TDT analiza documentos para decidir si se trata de una historia de documentos ya registrada, o bien, si es un tema nuevo. Esta área, por lo tanto, tiene que ver también con la categorización temática de noticias para poder relacionarlas unas con otras.

Las técnicas de aprendizaje automático son muy diversas, pero varias de ellas comparten las características generales en cuanto a parametrización, fórmulas utilizadas o bien en la forma de ver el problema de la clasificación. Por ello, son categorizadas en paradigmas [7]. Estos son:

- *Linear predictors*: buscan hacer una clasificación basada en funciones lineales. En esta categoría se encuentran las técnicas como *halfspace, linear regression, logistic regression*, entre otras [7].

- *Boosting*: son una generalización de los *linear predictors* para obtener mejores resultados. *Adaboost* es un ejemplo de técnica [7].

- *Support vector machines*: abordan el reto de hacer más compleja la muestra. En términos generales, separa un conjunto de entrenamiento con amplio margen de si todos los ejemplos no solo son correctos sino también están lejos del hiperplano de separación [7].

- *Decision trees*: son predictores que trabajan para determinar la etiqueta asociada con una instancia mediante la exploración de un árbol. Algunos ejemplos en este paradigma son ID3 y *random forests* [7].

- *Nearest neighbor*: la idea central de estas técnicas es tomar de parámetro un conjunto de entrenamiento y luego predecir la clasificación de una nueva instancia basándose en las similitudes con su vecino más cercano del entrenamiento. Un ejemplo de técnica de este tipo es la de *K-nearest neighbors* [7].

- *Neural networks*: basan en el desarrollo de redes neuronales artificiales, un modelo computacional que emula el comportamiento de un cerebro humano. Básicamente consiste en un conjunto de elementos, las neuronas, que se conectan en una red de comunicación compleja que produce clasificaciones [7].

- *Clustering*: identifican grupos dentro de un conjunto de elementos, de modo que los más similares terminen en el mismo grupo, y que los diferentes estén separados en diferentes grupos. Dentro de este paradigma se encuentran las técnicas basadas en enlaces, *k-means, spectral, graph cut, information bottleneck*, entre otras [7].

- *Dimensionality reduction*: toman los datos en un espacio dimensional y los mapean en un espacio más pequeño (de menos dimensiones), aplicando una transformación lineal a los datos originales. Algunas técnicas en esta categoría son *Principal component analysis* y *Compressed sensing* [7].

- *Generative models*: asumen que la distribución de los datos tiene una forma paramétrica y tienen como objetivo estimar dichos parámetros. En esta categoría se encuentran técnicas como *Maximum likehood estimator, Naive Bayes, Linear Discriminant Analysis (LDA), Latent variables, EM algorithm* y *Bayesian reasoning* [7].

En cuanto a las métricas que definen efectividad de las técnicas, Han et al. [6] explican los conceptos relacionados.

Se le llama valores positivos (P) a las tuplas que coinciden con una clase determinada, y valores negativos (N) a las que no. Luego, cada resultado de la clasificación es también etiquetado como un valor positivo o negativo. Entonces, hay cuatro términos necesarios para entender cómo funcionan estas métricas. Todas ellas forman la llamada matriz de confusión, y son:

TP Verdaderos positivos: Las tuplas positivas que fueron etiquetadas correctamente por la técnica, como positivas.

TN Verdaderos negativos: Las tuplas negativas que fueron etiquetadas correctamente por las técnicas, como negativas.

FP Falsos positivos: Las tuplas negativas que fueron incorrectamente etiquetadas como positivas.

FN Falsos negativos: Las tuplas positivas que fueron incorrectamente etiquetadas como negativas.

Cuadro 4.1: Detalles de las métricas en [8].

Métrica	**Definición**
Accuracy	(TP + TN) / (P + N)
Error rate	(FP + FN) / (P + N)
Recall	TP / P
Specifity	TN / N
Precision	TP / (TP + FP)
F-score	(2 * *precision* * *recall*) / (*precision* + *recall*)

De acuerdo con esos cuatro conceptos, las métricas son definidas por una fórmula que explica cada una. Los detalles se muestran en el Cuadro 4.1.

4.4. Trabajo relacionado

En el contexto de clasificación automática de noticias extraídas de la web, no se encontraron estudios secundarios con la misma temática de este trabajo. Los análisis encontrados que más se asemejan son primarios, y explican conceptos relacionados con las técnicas que luego aplican al contexto noticioso, con la intención de implementar o proponer sistemas, algoritmos o herramientas para lograr el cometido de hacer la clasificación y posterior análisis. Por otro lado, algunos estudios secundarios en la literatura sí se refieren al uso de técnicas de aprendizaje automático, minería de texto y *Web mining* en general, pero no sobre contextos exclusivamente noticiosos. A continuación, se presentan algunos estudios relacionados, todos ellos secundarios.

Sebastiani [11] hace un análisis sobre los algoritmos de aprendizaje automático para categorización temática de textos; aunque no específicamente relacionados con el contexto de noticias, explica la construcción de los diferentes clasificadores de texto hasta el 2002. Además, basado en varios estudios primarios, analiza los grandes grupos de algoritmos, y los distingue por su configuración. El autor encontró y describió clasificadores orientados a los grupos *probabilistic, decision trees, decision rule, regression, on-line, Rocchio, neural networks, example, y support vector machines*. Analiza también la forma en la que se evaluaron las diferentes técnicas y concluye que,

desde principios de los noventas la efectividad de los clasificadores de texto se ha mejorado mucho por el uso de técnicas de aprendizaje automático.

En otro estudio publicado en 2015, Irfan et.al. [12] enfatizan en hacer una revisión sobre las técnicas de minería de texto aplicados a contenidos de redes sociales. Categorizan las técnicas encontradas y se refieren a la clasificación como opción efectiva en el uso de minería de texto en este contexto. El estudio presenta un esquema de técnicas agrupadas por sus características, principalmente divididas en 3 grandes grupos: *hierarchical, partitional,* y *ontology-based clustering*. El trabajo concluye que extraer patrones lógicos con información detallada de datos no estructurados, es un campo crítico por ser desarrollado.

En otro estudio secundario, Bharti y Babu [13] resumen la literatura existente hasta 2017 para la extracción automática de palabras clave a partir de textos, con el fin de hacer resúmenes de la información. Presenta, dentro de sus resultados técnicas para el agrupamiento temático de los contenidos analizados. Establecen cinco tipos principales de procesos de resumido, uno de ellos se enfatiza en aprendizaje automático. Sin embargo, no encontraron ningún reporte que utilizara esta técnica. Concluyen que hay un vacío de información y estandarización de las investigaciones en este campo.

Desde otra aproximación, el estudio secundario de Castillo y Cervantes [14] presenta un vistazo general a las representaciones de algoritmos en la literatura que tratan de resolver clasificación de texto utilizando técnicas basadas en grafos. La motivación de este análisis fue mostrar cómo los grafos de coocurrencia pueden ser usados para representar documentos de texto y cómo esto puede ser un buen insumo. El estudio describe dos acercamientos principales: *feature-vector*, y *similarity*. Finalmente, menciona que los grafos son una alternativa que está al mismo nivel de otras técnicas, utilizando representaciones fáciles de construir y presentando un rendimiento rápido.

Aunque todos estos estudios secundarios reportaron técnicas relacionadas con el aprendizaje automático y minería de texto, ninguno trata específicamente sobre la clasificación temática en el contexto de noticias extraídas de la web. No obstante, dan una noción general de la clasificación de técnicas y son ejemplo sobre los intentos de clasificación de información para la posterior observación de patrones sobre los datos estudiados. Además, incorporan dentro del contexto de clasificación, el resumido de

textos y la búsqueda de palabras clave, que al inicio no se consideraron como parte de este estudio, pero luego fueron incorporadas porque se identificó la relación con dichos temas.

4.5. Metodología

La metodología descrita a continuación detalla los pasos que se llevaron a cabo para el mapeo sistemático de literatura, de acuerdo con los lineamientos de Petersen en [1] y las recomendaciones establecidas de Kitchenham [11].

Seguidamente se describe el objetivo de la investigación, las preguntas de investigación, la estrategia de búsqueda y definición de cadena, los criterios de inclusión y exclusión, así como las pautas para la evaluación de la calidad de los estudios seleccionados.

4.5.1. Objetivo

El objetivo de este estudio, de acuerdo con el modelo *Goal Question Metric* (GQM) [3] es *analizar* técnicas de aprendizaje automático y de minería de datos para la clasificación de contenidos, *con el propósito de* caracterizarlas, *con respecto a* su configuración, las características de fuentes de datos que se utilizan y las formas en que miden su efectividad *desde el punto de vista* del investigador *en el contexto de* análisis de noticias extraídas de la web.

4.5.2. Preguntas de investigación

Para guiar la presente investigación, se definieron las siguientes preguntas:

RQ1. ¿Cuáles técnicas de aprendizaje automático y minería de datos se han usado para categorizar temáticamente datos noticiosos extraídos de la web?

Conocer las técnicas de aprendizaje automático y minería de datos y su configuración es importante para identificar las características que se han usado en la categorización de datos noticiosos. Al responder esta pregunta, se plan-

tea detallar cuáles son las técnicas definidas, la frecuencia de aparición cada una y cómo han sido parametrizadas.

RQ2. ¿Cuáles fuentes de datos han sido utilizadas por las técnicas de aprendizaje automático y minería de datos para la clasificación temática de noticias extraídas de la web?

Con esta pregunta se quiere profundizar en cuál es el formato (de presentación y organización) de los datos noticiosos que se usan para la clasificación automática por medio de técnicas de aprendizaje automático y minería de datos. Además, esta pregunta comprende poder conocer el procesamiento que debe realizarse sobre los datos, en preparación, a las técnicas de minería.

RQ3. ¿Cuáles métricas se han usado para evaluar la efectividad de las técnicas de aprendizaje automático y minería de datos que clasifican temáticamente noticias extraídas de la web?

Al responder esta pregunta de investigación se busca obtener, de la literatura, cómo han hecho los estudios primarios para evaluar la efectividad de las técnicas utilizadas, midiendo su eficacia y eficiencia. Para ello se investigan las métricas que cuantifican los resultados de las técnicas empleadas. El fin es obtener información sobre su descripción general y sus características.

4.5.3. Proceso de búsqueda

A continuación, se detalla el proceso seguido para encontrar artículos relacionados al tema de estudio.

Artículos de control

A partir de las preguntas que se formularon, en primera instancia se detectaron tres artículos de control que funcionaron para el tema del estudio [17, 18, 19]. Estas investigaciones fueron seleccionadas como punto de partida, pues se referían al contexto específico de clasificación de noticias, y además presentaban explicaciones específicas y detalladas que permitían al lector, entender las técnicas y su contexto de aplicación.

Maghdid [18] plantea tres algoritmos de minería de datos de uso común, los cuales se utilizan para clasificar un set de 25 mil noticias extraídas de la web. Hace un análisis de los resultados que se obtuvieron con los tres algoritmos, en términos de la eficiencia en tiempo y espacio. Luego, propone incorporar en los análisis, variables adicionales como el tiempo en el que la noticia fue emitida, y concluye que al incluirle más variables relacionadas con locación y tiempo, se obtienen mejores resultados para las técnicas.

En el estudio de Bouras y Tsogkas [17], se investiga la aplicación de varios algoritmos de clasificación, sobre artículos de noticias extraídos de la web, y se comparan sus resultados en cuanto a eficiencia. También se examina el efecto de hacer un preprocesamiento en la clasificación de los contenidos noticiosos. El experimento mostró que el algoritmo k-means con algunos pasos previos sobre los datos, da mejores resultados que otros que se tomaron en cuenta.

En la investigación de Dadgar et. al. [19], se estudia el problema de clasificación temática de noticias como uno de los principales problemas de minería de texto. Reporta una herramienta que le permite a los usuarios de noticias, identificar grupos de noticias, el cual se basa en *support vector machines (SVM)*. Detalla los pasos de preprocesamiento, el análisis de documentos y la clasificación.

Se tomó en consideración que estos artículos debían ser parte del cuerpo de material por analizar, así, estos estudios funcionaron como guía en la etapa de construcción de la cadena de búsqueda.

Cadena de búsqueda

Para la búsqueda sistematizada de estudios sobre aprendizaje automático y minería de datos para la clasificación de datos noticiosos, se construyó una cadena de búsqueda utilizando el modelo “PICO”, que determina cuatro diferentes grupos de información: la población, la intervención, la comparación y las salidas [4]. Seguidamente se detalla la conformación de cada uno de estos grupos del modelo PICO.

La población del estudio la componen las noticias que se extraen de los sitios web de medios de comunicación masiva en Internet. Para ello se definen la palabra “news” en conjunto con la palabra “web”.

La intervención se compone de dos partes: por un lado, se busca información sobre la clasificación o agrupamiento temático de noticias, por lo que se definen las

palabras *classification* (y sus derivados), y *clustering* (y sus derivados). En segundo lugar, se identificaba que las técnicas fuesen específicas de minería de datos, por lo que se decidió utilizar la palabra *mining*. Este término incluye tanto lo relacionado con minería de datos, como con minería de texto o *Web mining*. Luego de algunos pilotajes y lecturas de artículos que se obtenían en los resultados de la búsqueda, se determinó que las técnicas tenían relación con el tema de aprendizaje automático, y eran tratadas como partes de una misma área. A partir de esos pilotajes, se definió que las técnicas encontradas se reportaban también bajo el tema de aprendizaje automático, por lo que se tomó la decisión de ampliar la cadena y el tema del estudio. Se incorporó entonces a la cadena el término "machine learning"(y sus derivados).

La comparación no aplica para este estudio, pues no se están comparando la intervención contra algo.

En cuanto a las salidas de la búsqueda, en un principio se pensó en agregar los términos “técnicas”, “estrategias” o “algoritmos”; no obstante, luego de un pilotaje, se determinó que dichos términos no eran relevantes pues limitaban demasiado la búsqueda, y se quedaban por fuera estudios que interesaban. En el contexto de la intervención y la población previamente definidas, cualquier salida relacionada con software sería potencialmente relevante para la investigación. Por ello, se decidió no aplicar ningún término al grupo de salidas.

A partir de los grupos definidos del modelo PICO, se crea la cadena de búsqueda:

```
( Web AND news ) AND ( classif* OR cluster* ) AND ( mining OR "
   machine_learn*" )
```

Se hicieron algunas pruebas para determinar si el orden de los elementos entre paréntesis, o bien la utilización de paréntesis diferentes para cada textitcluster del PICO afectarían los resultados. Se determinó que los resultados son exactamente los mismos en las tres bases de datos, por lo que no se le dio relevancia.

Durante las pruebas de pilotaje, se decidió aplicar sobre la cadena de búsqueda algunos criterios de exclusión para filtrar temas que no interesan al estudio, en particular el análisis de sentimientos, noticias en formato de video y detección automática de *fake news*. Con este fin, se agregó a la cadena los términos: AND NOT video AND NOT sentiment AND NOT ”fake news”.

Antes de tomar la decisión definitiva de agregar a la cadena de búsqueda estas

exclusiones, se hicieron pruebas para determinar si la diferencia era significativa. Además, se analizó una muestra de los artículos que serían eliminados de los resultados al agregar estas cláusulas y se determinó que el cambio implicaba descartar más de 30 artículos que no eran del contexto del estudio. Este proceso de validación se realizó para determinar que el cambio no afectaría la calidad de los resultados de la exploración.

La cadena final de búsqueda utilizada entonces fue:

```
( Web AND news ) AND ( classif* OR cluster* ) AND ( mining OR "
   machine_learn*" ) AND NOT video AND NOT sentiment AND NOT ''fake
    news''
```

Repositorios de búsqueda

Los repositorios sobre los cuales se corrió la cadena de búsqueda fueron: *Scopus*, *IEEE Xplore* y *Web Of Science*. Estas tres bases de datos son fuentes reconocidas y confiables en el campo de la Ingeniería de *Software*, las cuales permiten realizar pesquisas avanzadas sobre miles de artículos científicos.

En las tres bases de datos, se hizo la indagación sobre título, resumen y palabras clave.

Para Scopus, se hizo un ajuste particular en la cadena de búsqueda. Además, se limitó la búsqueda al campo de Ingeniería o Ciencias de la Computación, por lo que se le agregó a la cadena de búsqueda. Para esta base de datos entonces la cadena utilizada fue:

```
TITLE-ABS-KEY ( ( Web AND news ) AND ( classif* OR cluster* ) AND (
    mining OR ''machine learn*'' ) AND NOT video AND NOT sentiment
   AND NOT ''fake news'' ) AND ( LIMIT-TO ( SUBJAREA , ''COMP'' )
   OR LIMIT-TO ( SUBJAREA , ''ENGI'' ) ) .
```

Para Web of Science, se hizo el ajuste de la cadena considerando que en este buscador el AND es implícito al usar el NOT. Por lo tanto, quedó de la siguiente forma:

```
TS = ( ( Web AND news ) AND ( classif* OR cluster* ) AND ( mining
   OR "machine_learn*" ) NOT video NOT sentiment NOT ''fake news''
   ).
```

La búsqueda automatizada se realizó desde abril 2019 y los estudios fueron analizados durante mayo y hasta noviembre del 2019. El rastreo consideró estudios primarios recuperados hasta el mes de octubre del 2019.

4.5.4. Proceso de selección de estudios

Para decidir si los artículos resultantes de las búsquedas eran relevantes para la investigación, se definieron algunos criterios de inclusión y de exclusión. En este proceso se evaluaron los artículos postulantes con base en su título, palabras clave y resumen. Si no se podía determinar a partir de ahí, entonces se procedió a analizarlo y clasificarlo a partir del texto completo del artículo.

En este proceso de inclusión y exclusión, se evaluaron los artículos postulantes con base en su título, palabras clave y resumen. Si no se podía determinar a partir de ahí, entonces se procedió a analizarlo y clasificarlo a partir del texto completo del artículo.

En este estudio se excluyeron las publicaciones que cumplían con al menos uno de los siguientes criterios:

E1. Artículo no disponible en texto completo, después de una búsqueda intensiva.

E2. Artículo escrito en un idioma distinto al inglés.

E3. Artículo no es un estudio primario

E4. Las noticias que clasifica el artículo no están escritas en idioma inglés.

En relación con el criterio E1, es necesario aclarar que se hizo todo lo posible por encontrar los textos completos de cada artículo: en primera instancia, se intentó acceder al sitio original señalado por la base de datos; en segundo lugar, se procuró encontrar por acceso libre haciendo búsqueda en Internet; y en tercer lugar, se exploraron sitios de repositorios de investigación en donde era posible pedirlo directamente a los autores. A pesar de esto, hubo 37 estudios que no se pudieron encontrar en texto completo, todos ellos provenientes de las bases de datos *Scopus* o *Web of Science*.

Como criterios de inclusión, se establecieron los siguientes.

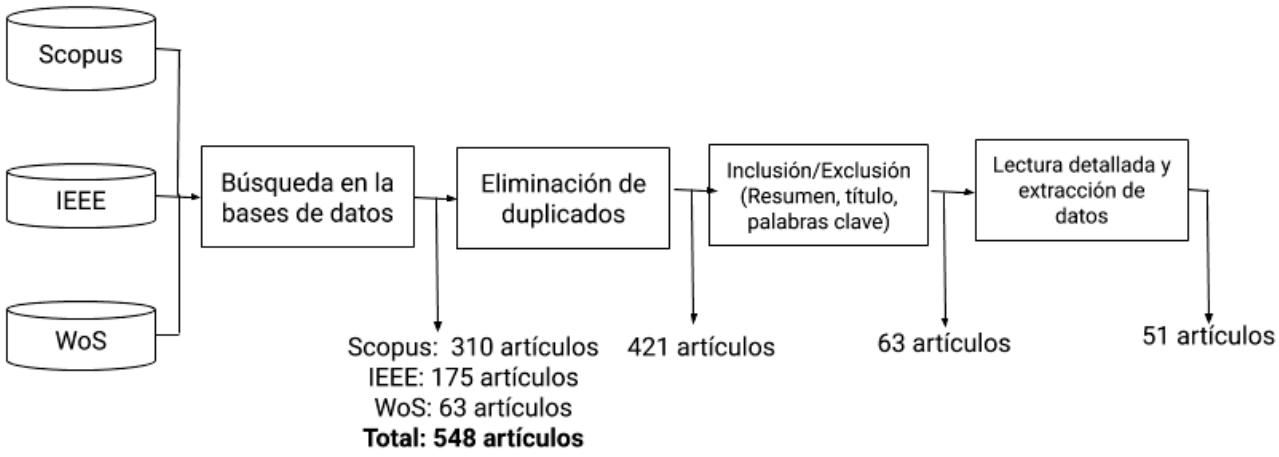


Figura 4.1: Proceso de selección de estudios.

I1. Artículos que analizan noticias por medio de técnicas reportadas como minería de datos o aprendizaje automático.

I2. Artículos que categorizan temáticamente noticias.

I3. Artículos donde las noticias que analizan están en formato de texto.

Se incluyen los artículos que cumplen con todos estos.

Resultados del proceso de selección

La Figura 4.1 muestra, de forma resumida, el proceso de selección de los artículos para este estudio secundario.

Al realizar la búsqueda automatizada en las bases de datos seleccionadas, se obtuvieron 63 resultados en Web of Science, 175 en IEEE Explore y 310 en Scopus. Luego se eliminaron los artículos duplicados, con lo que se obtuvieron 421 artículos diferentes. Sobre estos, se aplicó el proceso de inclusión y exclusión con base en resumen, las palabras clave y un escaneo general del artículo en la mayoría de las ocasiones.

Después de aplicar esos criterios se incluyeron 64 artículos en total. Al hacer la lectura completa de estos, fue necesario excluir algunos de ellos pues no cumplían con criterios establecidos, pero esto fue evidente hasta ese momento. Por esta razón, al final se seleccionó un total de 51 artículos, los cuales fueron extraídos y analizados.

Los artículos finalmente incluidos en el estudio se detallan en el apéndice 4.A.

4.5.5. Evaluación de la calidad

La evaluación de la calidad brinda información sobre el nivel de detalle que presentan los artículos seleccionados. El objetivo de esta evaluación de calidad es poder medir qué tan completa es la información presentada por los estudios primarios y darle validez al conocimiento empírico generado, de modo que se pueda saber cuáles son los que más aportan en relación con las preguntas de investigación.

Q1. ¿El artículo indica que se refiere a noticias web? (2 puntos si menciona que se refiere específicamente a datos de noticias; 1 si no lo dice pero implícitamente se infiere que aplica las técnicas sobre noticias; 0 si no indica que aplica las técnicas sobre noticias y tampoco se puede inferir).

Q2. ¿El artículo explica los pasos para llevar a cabo los procedimientos descritos? (2 puntos si describe con detalle los pasos por seguir; 1 si los describe de forma general sin profundizar; 0, si no los describe).

Q3. ¿El artículo describe los requisitos de preprocesamiento de los datos antes de utilizar las técnicas de minería de datos? (2 puntos si explica los requisitos de preprocesamiento; 1, si solamente los menciona; 0 si no se refiere al preprocesamiento de los datos del todo).

Q4. ¿El artículo explica las métricas utilizadas para medir la efectividad de las técnicas usadas en el estudio? (2 puntos si explica las métricas utilizadas; 1 si solamente menciona las métricas utilizadas; 0 si no menciona métricas).

Q5. ¿El estudio define el objetivo de investigación planteado? (2 puntos si lo define explícitamente; 1 punto si es implícito; 0 puntos si no se menciona ni se puede inferir).

Q6. ¿El estudio presenta las preguntas de investigación planteadas? (2 puntos si las define explícitamente; 1 punto si están implícitas; 0 puntos si no se mencionan ni se pueden inferir).

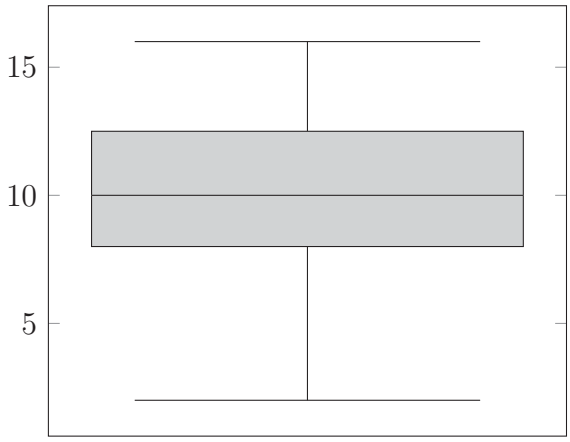


Figura 4.2: Calidad de los estudios primarios incluidos.

Q7. ¿El estudio detalla el diseño del estudio? (2 puntos si lo define explícitamente; 1 punto si es implícito; 0 puntos si no se menciona).

Q8. ¿El estudio explica la metodología empleada en el estudio? (2 puntos si lo explica claramente; 1 punto si solamente lo menciona; 0 puntos si no se menciona del todo).

Para cada uno de los criterios de evaluación de calidad, a todo artículo se le concede una puntuación de 0 a 2 puntos, de acuerdo con su nivel de cumplimiento.

Los criterios del Q1 al Q4 definen detalles del estudio de acuerdo con los intereses de esta investigación, por lo que evalúan la pertinencia de cada artículo para el análisis secundario. Por su parte, los criterios del Q5 al Q8 ayudan a dar una idea sobre la rigurosidad de las técnicas de investigación empleadas en los estudios primarios. Una puntuación baja no significará que el estudio será excluido. De esta forma, sobre una base de 16 puntos se puede tener una idea de la completitud del estudio en relación con las preguntas de investigación y su descripción metodológica.

Los resultados del proceso de evaluación de calidad se muestran en el apéndice 4.B. El balance de estos resultados se representa gráficamente en la Figura 4.2.

4.5.6. Extracción de los datos

A partir de una lectura exhaustiva de los estudios, se extraen los aspectos puntuales de las técnicas utilizadas que interesan para la investigación y que responden a las preguntas planteadas. Para ello, se clasifica la información obtenida de cada artículo y se coloca en un formulario de extracción. Los componentes del formulario de extracción se muestran en el Cuadro 4.2.

Cuadro 4.2: Componentes del formulario de extracción.

Categoría	Componentes
Identificación	ID de artículo, referencia, título, autores, año de publicación, tipo de estudio
Técnicas (RQ1)	Nombre de la técnica, nombre del paradigma base, descripción, procedimientos
Fuentes de datos (RQ2)	Tamaño, descripción, lenguaje, fuentes, formato, pasos de preprocesamiento, descripción de los pasos de preprocesamiento, representación de documentos
Métricas evaluadoras (RQ3)	Nombre, descripción, fórmulas, pruebas realizadas, hallazgo principal

En los casos en que un mismo artículo utiliza varias técnicas a la vez, estas son tomadas en cuenta como valores separados para poder comparar luego la frecuencia de cada técnica utilizada.

En el enlace `https://tinyurl.com/ydcl26qg` se detalla el formulario de extracción utilizado.

4.5.7. Análisis de datos

El análisis se centra en los aspectos específicos que permiten contestar las preguntas de investigación.

En relación con la RQ1, consistió en identificar las técnicas utilizadas de minería de datos en los estudios. Además, se determinaron las características generales de dichas técnicas para poder clasificarlas dentro de una taxonomía. Para ello, se utilizó la taxonomía de Shalev-Shwartz et al. [7].

En cuanto a la RQ2, se enfocó en las características de las fuentes de datos, de dónde provienen y qué formato tenían las noticias utilizadas, así como los procedimientos de preprocesamiento que se realizaron sobre los datos. Se tomó en consideración únicamente la información brindada por cada artículo, y el agrupamiento se hizo según el nombre de las técnicas de preprocesamiento o su explicación.

Finalmente, para la RQ3 consistió en determinar cómo se evaluaron las técnicas, cuántas métricas se utilizan a la vez para probar los resultados y qué tan útiles resultan ser. El agrupamiento de información relativa alas métricas implementadas se hizo según las métricas descritas por Han et al. [6].

4.5.8. Amenazas a la validez

Las amenazas a la validez describen las limitaciones y puntos fuertes del estudio en relación con la validez de los resultados que se obtengan. En esta investigación, las amenazas detectadas son:

Selección de la cadena de búsqueda y las bibliotecas digitales: la definición de la cadena y las bibliotecas es una amenaza porque es una decisión que delimita el estudio completamente y si se omite algún aspecto relevante, información valiosa quedaría por fuera de la investigación. Para mitigar esto, se consideró que las bases de datos seleccionadas fueran reconocidas y con gran cobertura de estudios. La cadena, por otro lado, fue producto de varios pilotajes mediante los cuales se fue refinando y depurando la cadena de búsqueda. Por lo tanto, su definición fue específica y fundamentada, no definida a la ligera.

Identificación de los estudios primarios: aunque los criterios de inclusión y exclusión se definieron de forma objetiva, el proceso de selección de artículos fue realizado únicamente por un investigador, por lo que podría haber un sesgo en la escogencia de los estudios. La forma de mitigarlo fue que cuando hubo duda en si un artículo cumplía o no un criterio, se hizo la selección con base en el texto completo

y tratando de profundizar más. Si aún así había dudas sobre si debería ser incluido o excluido, la prioridad del proceso fue tomarlos en cuenta para la etapa de lectura completa.

Extracción y clasificación de artículos primarios: puede haber un sesgo a la hora de extraer la información de los artículos incluidos en el estudio, y en la clasificación de las técnicas que estos reportan. Para ello, durante el proceso se realizó varias veces una verificación de que el formulario de extracción fuera coherente con las preguntas de investigación. Además, cuando las técnicas reportadas no fueron clasificadas por los autores de los estudios, se hicieron validaciones de la teoría, a partir de la taxonomía que se utilizó.

Generalización y síntesis de resultados: la generalización y síntesis de resultados se hace a partir de los estudios analizados solamente. Para minimizar riesgos al presentar los resultados, la investigación se hizo siguiendo en todo momento los protocolos y metodologías definidos y validados. Buenas prácticas en la metodología ayudaron a mitigar este riesgo.

Consulta a expertos: para esta investigación no se hicieron consultas a expertos en el área. No obstante, el proceso de elección de los artículos de control y la búsqueda exhaustiva y repetida sobre las bases de datos garantizó que los datos que se obtuvieron vienen de fuentes confiables, y se justifican pertinentemente durante el desarrollo del capítulo. La metodología empleada permitió generar datos objetivos, rastreables y medibles, por lo que si se tienen claras las limitaciones de esta metodología, no es necesaria la consulta directa a expertos en el área.

4.6. Análisis de resultados

En esta sección se presentan los resultados del mapeo sistemático, basado en los 51 estudios primarios que se analizaron, y de acuerdo con las respuestas a cada pregunta de investigación planteada. La lista completa de artículos incluidos puede ser consultada en el apéndice 4.A.

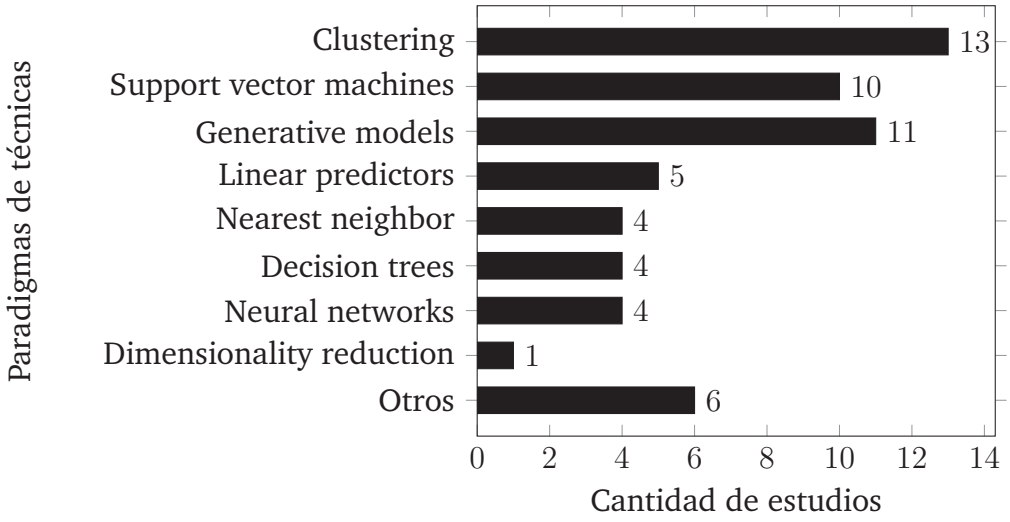


Figura 4.3: Paradigmas de técnicas reportados.

4.6.1. Técnicas de aprendizaje automático y minería de datos utilizadas para la categorización temática de noticias (RQ1)

Para esta pregunta de investigación, a todos los estudios primarios se les logró extraer información de las técnicas utilizadas.

La Figura 4.3 muestra la frecuencia de las técnicas de aprendizaje automático y minería de datos que fueron utilizados para la clasificación automática de noticias. El eje vertical expone los paradigmas de técnicas (según la taxonomía propuesta en [7]). Mientras que el eje horizontal describe la cantidad de artículos mencionan dichas técnicas. El paradigma de clasificación que más se ha usado es *clustering*, seguido por *support vector machines* y *generative models*. Estas tres categorías concentran el 58 % de los reportes. Seis estudios fueron clasificados en la categoría de ”otros”, pues no cumplían con las características de ninguno de los paradigmas anteriores: 3 estudios se refieren a *fuzzy systems*, otro es de comportamiento de *beehive* por agentes, uno se refiere a análisis de palabras claves y otro menciona una técnica nueva llamada *table based matching algorithm*.

Todos los artículos se clasificaron según las técnicas de aprendizaje automático o minería de datos que utilizan, por ello algunos artículos son mencionados en más de

una categoría, pues presentaron varias técnicas.

Las técnicas de *clustering* identifican grupos diferenciados a partir de un conjunto de elementos, basado en similitudes. En el caso de los artículos de noticias, esto significa que las técnicas dividen los contenidos de los artículos de acuerdo con la relación que la técnica halla entre los términos que se utilizan. El hecho de que este sea el paradigma más reportado puede deberse a que la definición de la técnica tiene similitud con el problema planteado en sí mismo, pues simplemente buscan hacer agrupamientos temáticos dadas las características de los datos.

Las técnicas de *support vector machines* usan aprendizaje supervisado para analizar y clasificar cada elemento en una de dos categorías. Identifica la brecha que hay en un conjunto de datos de modo que coloca cada elemento de la forma más polarizada a un lado de esa diferencia. Para su uso sobre noticias, esto implica que determina para cada categoría, si el artículo pertenece o no.

Las técnicas orientadas a *generative models*, asumen al principio que la distribución de los datos tiene un parámetro e intenta estimarlo. En el caso de las noticias, esto implica que las variables que determinan cada noticia son analizadas por aparte como independientes para determinar la pertenencia o no a un grupo específico, es decir, su clasificación temática.

El Cuadro 4.3 expone los estudios que reportan cada uno de los paradigmas de técnicas sobre aprendizaje automático y minería de datos. Como puede observarse, varios de los estudios utilizaron diversas técnicas a la vez sobre los datos. La segunda columna presenta un listado de técnicas específicas que se reportaron directamente bajo ese nombre, para cada paradigma. En el apéndice 4.D se muestra un gráfico relacionado.

Por otra parte, la Figura 4.4 muestra la distribución de cada paradigma de técnicas, por año. El eje horizontal indica el año del estudio, mientras que el eje vertical el grupo de técnicas que se usaron. Mientras que hay algunas técnicas que han mantenido una tendencia constante en el tiempo, otras no. Por ejemplo, la técnica de *linear predictors* no aparece en estudios posteriores al 2011, mientras que las técnicas de *clustering* se han mantenido durante el tiempo desde la primera aparición en 2004. La mayor concentración de estudios que emplearon *support vector machines* y *generative models* es en el año 2012.

Cuadro 4.3: Estudios que reportan técnicas de aprendizaje automático y minería de datos.

Paradigma	Técnicas	Cant.	Estudios
Clustering	*K-means, k-medians, complete-link graph, pairwise linkage, Locality Sensitive Hashing (LSH), F2N-Rank graph, RSS Organizing and Classification System (ROCS), Particle swarm optimization, Probabilistic named Entity Recognition*	13	[21, 22, 17, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]
Generative models	*Multinominal naive bayes (MNNB), Latent Dirichlet Allocation (LDA), Expectation–maximization (EM) algorithm, DBSCAN, Naive bayes*	11	[33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]
Support vector machines	*Standart, non-linear*	10	[44, 33, 45, 46, 47, 19, 48, 49, 50, 43]
Linear predictors	*Rocchio, Hierarchical Topic Model with Ontological Guidance, hierarchical temporal topic tracking, Probabilistic matrix*	5	[51, 52, 53, 54, 55]
Decision trees	*Suffix Tree, Standart Decision tree technique, Improved STC, C4.5*	4	[22, 18, 56, 39]
Nearest neighbor	*K-Nearest Neighbor*	4	[57, 58, 18, 38]
Neural networks	*Neural Preference Moore Machine, Artificial neural network technique, Dynamic artificial neural network (DAN2), recurrent plausibility networks*	4	[18, 59, 60, 61]
Dimensionality reduction	*Document Classification and Knowledge Extraction*	1	[62]
Otras	*Fuzzy systems, keyword analysis using Wordnet, Beehive, table based matching*	6	[63, 64, 4, 65, 66, 67]

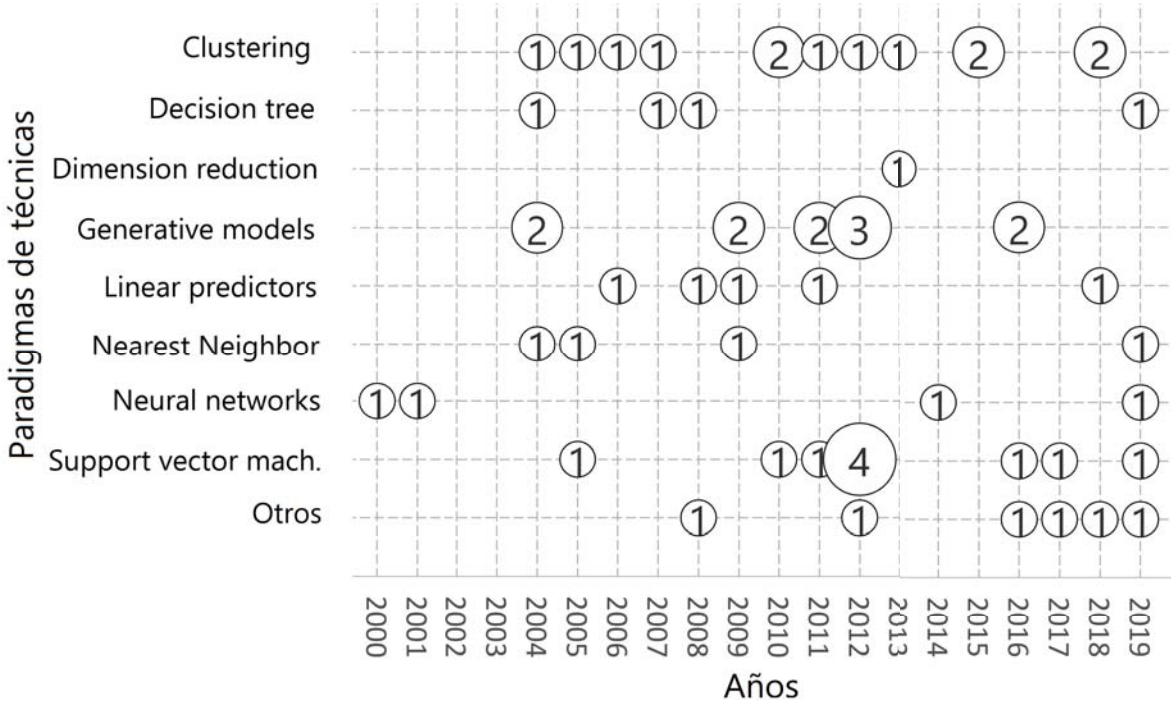


Figura 4.4: Cantidad de reportes de paradigmas de técnicas, por año de estudio.

De esta información es difícil divisar una tendencia en los datos. Básicamente, no hay una constancia marcada que diga que algunos paradigmas hayan sido más utilizados que otros a través del tiempo. No obstante, como puede observarse en el gráfico, varios paradigmas de técnicas han sido menos reportadas en los últimos años (*linear predictors, dimensionality reduction*), mientras que otras más bien parecen predominar en este período (*clustering, support vector machines*). Las técnicas de *decision trees*, *nearest neighbor* y *neural networks* estuvieron ausentes por algunos años, pero volvieron a ser reportadas en el año 2019.

4.6.2. Características de las fuentes de datos utilizadas para la clasificación temática de noticias con aprendizaje automático y minería de datos (RQ2)

En cada uno de los estudios analizados, los autores establecieron conjuntos de datos noticiosos sobre los cuales aplicaron las técnicas de aprendizaje automático y

Cuadro 4.4: Cantidad de noticias que conforman el conjunto de datos utilizado.

Cantidad de noticias	Cantidad	Estudios
Entre 1 y 5 mil noticias	26	[64, 57, 22, 44, 33, 4, 23, 45, 46, 47, 25, 26, 34, 35, 28, 29, 51, 24, 19, 54, 56, 48, 36, 32, 42, 67]
Entre 5001 y 10 mil noticias	4	[17, 37, 60, 41]
Entre 10001 y 15 mil noticias	3	[59, 65, 61]
Entre 15001 y 20 mil noticias	6	[52, 62, 53, 50, 40, 43]
Entre 20001 y 25 mil noticias	2	[18, 49]
Más de 35 mil noticias	5	[21, 58, 27, 55, 38]
Cantidad no especificada	5	[66, 30, 31, 63, 39]

minería de datos. A continuación se muestran algunas características de estos conjuntos de datos.

Tamaño de los conjuntos de datos

El Cuadro 4.4 señala el tamaño de los conjuntos de datos para cada estudio primario, organizados por conjuntos de 5 mil noticias. Veintiséis estudios usaron conjuntos de datos de 5 mil noticias o menos, 13 usaron cuerpos de entre 5 mil y 20 mil noticias, 7 usaron sets de más de 20 mil pero menos de 35 mil noticias, y 5 estudios utilizaron más de 35 mil. Además, 5 de estos no mencionaron el tamaño de su cuerpo de datos.

Origen de los cuerpos de datos

De los estudios primarios analizados, 26 extrajeron las noticias a partir de las que estaban disponibles en páginas de Internet de contenidos noticiosos, otros 18 únicamente manejaron un cuerpo de datos existente que fue descargado para realizar sus pruebas de clasificación, 7 utilizaron ambas fuentes de datos. Esta información se muestra de forma gráfica en la Figura 4.5. La figura expone en su eje vertical el tipo de fuente de datos usado, mientras que el horizontal muestra la cantidad de estudios.

En total, 33 estudios utilizaron extracción a partir de sitios web de noticias, mientras que 25 optaron por un conjunto de datos. En general, los estudios que extrajeron noticias de sitios web tuvieron acceso a información de primera mano con datos

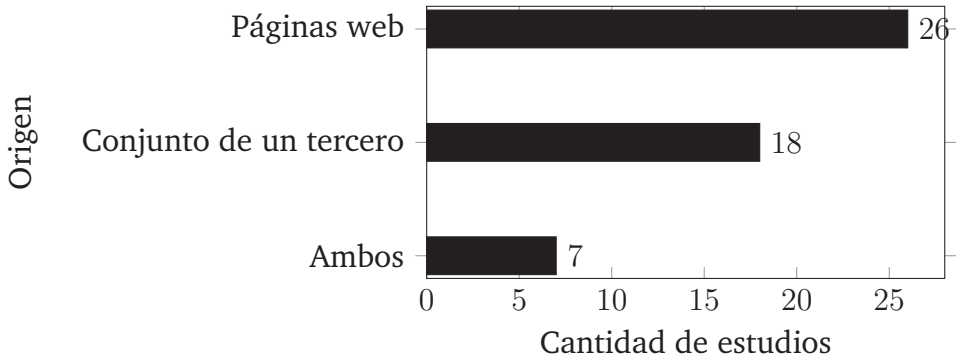


Figura 4.5: Cantidad de reportes de origen de las noticias.

reales, mientras que los análisis que usaron conjuntos de datos se limitaron a información más preparada para la clasificación, y menos actualizada para el momento del estudio.

El Cuadro 4.5 muestra el detalle del origen de los datos que usó cada estudio. Puede notarse que dentro de los conjuntos de datos preparados hay varias referencias a 20NewsGroup y Reuters. En cuanto a los sitios Web de noticias, hay gran variedad de fuentes pero se usan con frecuencia las grandes agencias como CNN, y Reuters.com. Otros estudios optaron por utilizar motores de búsqueda como Yahoo news y Google news.

Cuadro 4.5: Origen de los datos utilizados por cada estudio.

Estudio	Detalle
[21]	Noticias de las páginas web: 7am.com, bergens-tidende.no, chronicle.com, dagbladet.no, dn.no, latimes.com, nypost.com, nytimes.com, observer.guardian.co.uk, sognavis.no, vg.no
[64]	Noticias autralianas sustraídas de la página www.theage.com.au.
[57]	Noticias del medio CNN
[59]	Reuters corpus
[22]	Resultados de una búsqueda en Google que luego pasaron por ciertos filtros

Continúa en la página siguiente.

Estudio	Detalle
[44]	Noticias provenientes de varios sitios de noticias no especificados, y el conjunto de datos 20Newsgroup
[33]	Noticias provenientes de páginas como BBC, CNN y SKYNews.
[17]	Noticias de 20 portales de noticias web seleccionados.
[58]	Reuters RCV1. corpus
[4]	Noticias del New York Times (NYT) online
[23]	Noticias del UAB Reporter's archives
[45]	Noticias de 13 fuentes de noticias: ABCAU, ABCNews, Aljazeera, Boston, CNN, CSmonitor, Euronews, FoxNews, Reuters, Telegraph, The Globe, USA-Today y WSJ.
[18]	Conjunto de datos News Aggregator Data, obtenido de "Center for Machine Learning and Intelligent Systems", "University of California, Irvine".
[46]	Diferentes sitios de noticias no especificados, y el conjunto de datos 20News-Group.
[47]	Noticias de proveedores de noticias no especificados.
[65]	Noticias de Hindustan Times (http://www.hindustantimes.com) y The Guardian (https://www.theguardian.com).
[25]	Noticias a partir de Yahoo news.
[52]	Conjunto de datos 20Newsgroup, y también noticias de los portales Web bbc.co.uk, cnn.com y usatoday.com.
[62]	Conjunto de datos 20 NewsGroup.
[53]	Conjunto de datos 20newsgroup.
[26]	Noticias extraídas de finance.yahoo.com.
[34]	Noticias extraídas de la página http://www.globalissues.org/ .

Continúa en la página siguiente.

Estudio	Detalle
[35]	Home page de Google News.
[27]	Noticias extraídas de Reuters.
[28]	Noticias extraídas de los RSS de varias páginas web, menciona CNN, Reuters y Euronews.
[29]	Conjunto de datos Reuters-21578 y conjunto de datos 20NewsGroup.
[30]	Conjunto de datos Reuters 21578.
[51]	Noticias extraídas de las páginas web de 4 agencias (CNN, Reuters, France24 y DW-World).
[24]	Conjunto de datos 20 News groups.
[19]	Conjunto de datos 20Newsgroup y también noticias provenientes de la BBC.
[54]	Noticias extraídas de la agencia Reuters.
[55]	Noticias extraídas de los servicios RSS de varias páginas de noticias no expecificadas.
[56]	Conjunto de datos llamado TDT2 English corpus.
[31]	Noticias extraídas de la web de sitios no especificados, y también el conjunto de datos English Gigaword Corpus.
[63]	Noticias extraídas del sitio Reuters.
[48]	HR-Net corpus y usaron también el ATIS corpus, pero este segundo es acerca de otro dominio (información de viajes).
[49]	Noticias extraídas de CNN, Yahoo News y Reuters.com. También usaron conjunto de datos 20 newsgroup y Reuters-21579 (y otros pero no del dominio específico de noticias).
[37]	Conjunto de datos de 20 Newsgroups collection.
[50]	Conjunto de datos de 20 Newsgroups collection.

Continúa en la página siguiente.

Estudio	Detalle
[32]	De 46 fuentes diferentes de noticias, no especificadas.
[38]	Artículos de noticias de Reuters RCV1 collection.
[60]	Conjunto de datos Reuters 21578.
[39]	Noticias extraídas de The Wall Street Journal de la TIPSTER collection.
[61]	Títulos a partir de Reuters corpus filtrados por el reportado ModApte split.
[40]	Dos conjuntos de datos: Reuters-21578 y 20 Newsgroup.
[41]	Colección RCV1-v2.
[42]	Colección 20 Newsgroup.
[43]	Colección 20 Newsgroup.
[67]	Noticias extraídas de la página NewsPage.com .
[36, 66, 39]	No dan detalles de la fuente de datos.

El Cuadro 4.6 presenta los detalles de los cuerpos de datos reportados, para los 25 estudios que utilizaron un cuerpo de datos de un tercero. La mayor cantidad de reportes se dieron para 20 Newsgroup, con 14 estudios, mientras que otros 7 reportaron haber utilizado Reuters-21578.

Cuadro 4.6: Información de cuerpos de datos reportados.

Nombre	Estudios	Origen	Descripción	Tamaño	Clasific.
20 Newsgroup	14 estudios: [44, 46, 52, 53, 29, 24, 19, 49, 37, 50, 40, 42, 43, 62]	Ken Lang es acreditado como el creador de recolectar los datos.	Colección de documentos relacionados a noticias, agrupados por temas (algunos más específicos que otros).	18 846 documentos divididos en 20 grupos temáticos.	20 clasificadores divididos en 6 grandes grupos: *computers, recreation, sci, misc, talks, alternative,* y *social*.
Reuters-21578	7 estudios: [29, 30, 49, 60, 40, 61, 59]	Carnegie Group, Inc. y Reuters, Ltd.	Recolectado en 1987. La colección es en formato SGML. Dentro de los atributos para cada noticia, incluye una clasificación temática.	21 578 noticias	Divididos en estos grupos mayoritarios: *exchanges, organizations, people, places,* y *topics (economic subjects)*.

Continúa en la página siguiente.

Nombre	Estudios	Origen	Descripción	Tamaño	Clasific.
Reuters Corpus Volume I (RCV1)	3 estudios: [58, 38, 41]	Reuters, Ltd. , la agencia internacional más grande de noticias de texto y televisión.	Recolectado entre agosto de 1996 y agosto de 1997. La colección es en formato XML, y cada documento se categoriza en 3 criterios (tema, industria y región).	806 791 documentos en la v1, y 804 414 en su v2.	Códigos asignados organizados en estos grupos mayoritarios: *Corporate/Industrial, Economics, Government/Social* y *Markets*.

Continúa en la página siguiente.

Nombre	Estudios	Origen	Descripción	Tamaño	Clasific.
News Aggregator Data Set	1 estudio: [18]	Artificial Intelligence Lab, Roma Tre University, Italia.	Las noticias que presenta están agrupadas en conjuntos de datos según categoría temática, y a su vez hay grupos de noticias que se relacionan entre sí porque tratan sobre historias similares.	422 937 páginas, en 4 categorías temáticas (negocios, ciencia y tecnología, entretenimiento y salud) y cada categoría en más de mil clusters de noticias similares.	Incluye 4 mayores grupos: *business, science and technology, entertainment,* y *health*.

Continúa en la página siguiente.

Nombre	Estudios	Origen	Descripción	Tamaño	Clasific.
TDT2 English Corpus	1 estudio: [56]	Linguistic Data Consortium, University of Pennsylvania, EE.UU.	Es un texto que surge de la transcripción manual de noticias de radio y televisión a partir de audios. Las fuentes son: Associated Press's World Stream, the New York Times news service, Public Radio International's The World, Voice of America's English news, ABC's World News Tonight y CNN's Headline News.	54 mil historias divididas en 100 temas.	Incluye 100 temas diferentes, no especificados.
English Gigaword	1 estudio: [31]	Linguistic Data Consortium, University of Pennsylvania, EE.UU.	Surge a partir de las fuentes: Agence France Press English Service, Associated Press Worldstream English Service, The New York Times Newswire Service, y The Xinhua News Agency English Service. Formato en SGML.	314 archivos de noticias, sin información de categorización.	Sin información.

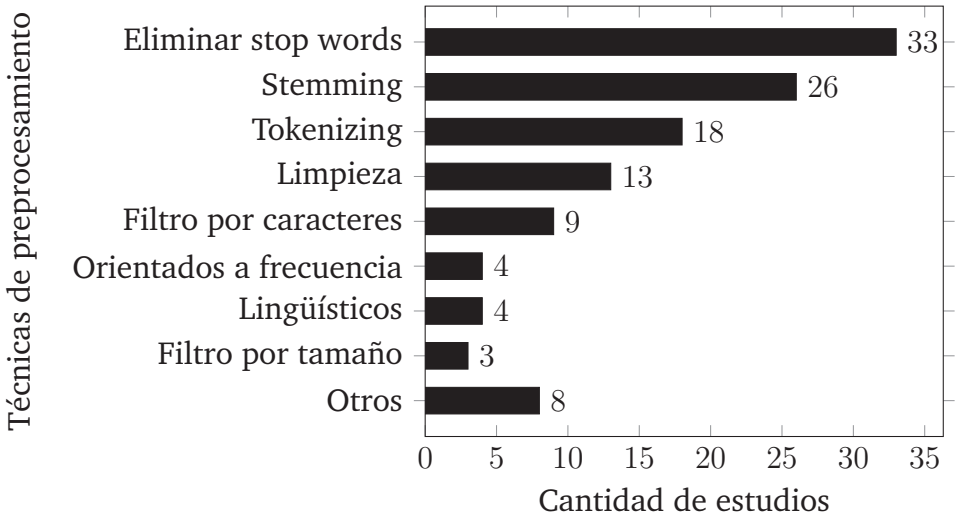


Figura 4.6: Técnicas de preprocesamiento de datos.

Algoritmos de preprocesamiento

Del total de estudios, solo 41 reportaron que utilizaron algún preprocesamiento sobre los datos con el fin de prepararlos para la técnica de minería de datos o aprendizaje automático empleada posteriormente.

La Figura 4.6 muestra la cantidad de estudios que reportaron cada técnica de preprocesamiento. El eje vertical describe las técnicas de preprocesamiento, mientras que el horizontal la cantidad de estudios que reportaron cada técnica. Algunos mencionaron varias técnicas de preprocesamiento, y cada una de estas menciones se contabilizó. Como puede observarse en la figura, la mayoría de los artículos usan como técnica de preprocesamiento la eliminación de *stop words*. Otras técnicas de preprocesamiento que se mencionan son el *stemming*, la limpieza de etiquetas innecesarias en los artículos de noticias y el proceso de *tokenizing*, entre otros.

La clasificación de las técnicas se hizo con base en la explicación que dieron los artículos, o el nombre de la técnica.

El cuadro 4.7 muestra las menciones da cada una de las técnicas de preprocesamiento que se llevaron a cabo sobre los datos, así como la descripción de las técnicas empleadas. Este detalle se extrajo a partir de las explicaciones que dieron los artículos que los mencionaron.

Cuadro 4.7: Estudios que reportaron técnicas de preprocesamiento.

Técnica	Detalle	Estudios
Eliminación de *stopwords*	Se identifican las palabras que no aportan relevancia para la clasificación y se eliminan de los documentos.	[21, 64, 57, 59, 22, 44, 33, 17, 4, 23, 46, 47, 25, 62, 26, 34, 35, 27, 28, 29, 30, 51, 19, 55, 36, 50, 32, 38, 39, 61, 41, 43, 67]
Stemming	Detectar las raíces de las palabras y eliminar el resto de cada una, de forma que se estandariza su comprensión.	[64, 57, 59, 22, 44, 33, 17, 4, 23, 18, 25, 62, 26, 35, 30, 24, 54, 55, 56, 37, 32, 38, 39, 41, 43, 67]
Tokenizing	Tomar cadenas de caracteres de los datos y separarlos en cada una de las palabras que las componen.	[64, 57, 44, 33, 17, 4, 18, 46, 26, 35, 27, 29, 30, 24, 19, 55, 39, 67]
Limpieza	Limpiar los documentos eliminando elementos innecesarios para la clasificación como comentarios y etiquetas.	[21, 57, 22, 17, 58, 47, 62, 51, 55, 56, 48, 36, 43]
Filtro por caracteres	Filtrar relacionado a los caracteres que se utilizan, por ejemplo para eliminar signos de puntuación o transformar caracteres a minúsculas	[21, 44, 33, 18, 47, 62, 19, 39, 41]
Filtro por tamaños	Eliminar palabras muy cortas o muy largas	[44, 18, 46]

Continúa en la página siguiente.

Técnica	Detalle	Estudios
Lingüísticas	Clasificación lingüística de las palabras para tomarlo en cuenta en el resto del proceso	[33, 28, 31, 38]
Orientados a frecuencia	Filtrar los datos, eliminando palabras muy poco frecuentes o demasiado frecuentes.	[21, 28, 29, 54]
Otras		[64, 47, 52, 35, 24, 56, 50, 32].

4.6.3. Métricas usadas para evaluar la efectividad de las técnicas de minería de datos y aprendizaje automático para clasificar noticias (RQ3)

De los estudios analizados, solo 44 reportaron las métricas utilizadas para medir la efectividad de las técnicas de minería de datos o aprendizaje automático que emplearon. La Figura 4.7 muestra la cantidad de veces que cada métrica fue reportada.

Las métricas fueron clasificadas con base en [6], según la explicación ofrecida por los artículos o la mención del nombre de la métrica. El Cuadro 4.8 señala el detalle. La que se utilizó más veces fue *F-measure*, con 25 reportes. Esta se obtiene al calcular la media armónica entre los valores de *precision* y *recall*. Por otro lado, 19 reportaron haber utilizado *precision* y *recall* por separado. *Accuracy* fue usada como métrica para medir resultados en 17 artículos primarios. Otros 10 estudios mencionaron medir *complexity* en términos de tiempo de ejecución o espacio. Finalmente, 13 estudios mencionan otras técnicas diferentes: [22, 17, 23, 28] cuentan los *clusters* generados y el tamaño de cada uno, los estudios [21, 23] mencionan como métrica la inspección manual de los resultados y los estudios [64, 25, 52, 29, 37, 60, 39, 40] utilizan otras métricas.

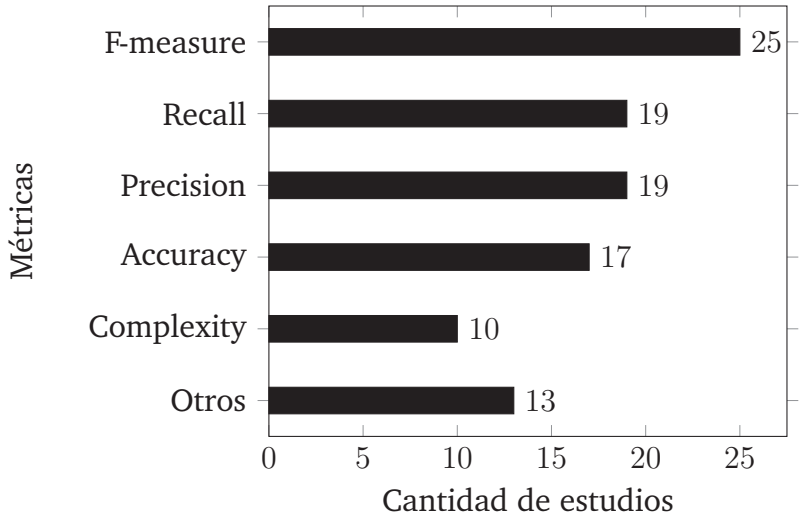


Figura 4.7: Métricas reportadas.

Cuadro 4.8: Reportes de métricas para medir efectividad.

Métrica	Cantidad	Estudios
F-measure	25	[33, 17, 45, 47, 65, 25, 52, 62, 53, 66, 35, 29, 51, 24, 19, 56, 31, 49, 32, 38, 60, 40, 41, 42, 43]
Recall	19	[57, 59, 45, 47, 65, 25, 62, 66, 51, 24, 19, 31, 48, 49, 32, 38, 60, 61, 42]
Precision	19	[57, 59, 45, 47, 65, 25, 62, 66, 51, 24, 19, 31, 48, 49, 32, 38, 60, 61, 42]
Accuracy	17	[64, 44, 58, 4, 18, 66, 35, 24, 55, 31, 37, 50, 32, 38, 60, 39, 67]
Complexity	10	[64, 17, 58, 18, 30, 56, 36, 60, 40, 41]
Otras	13	[22, 17, 23, 28, 21, 64, 25, 52, 29, 37, 60, 39, 40]

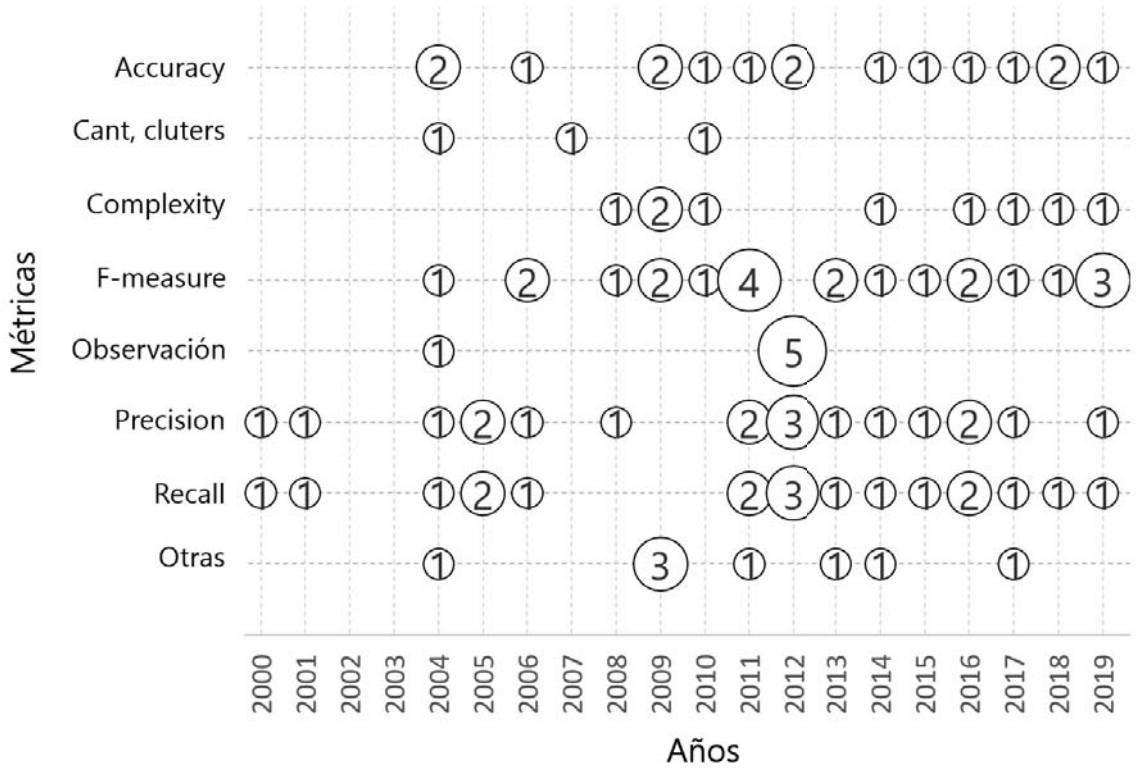


Figura 4.8: Cantidad de reportes de métricas por año.

Por otro lado, la Figura 4.8 muestra la tendencia cada métrica en el tiempo. El eje horizontal expone el año del estudio, mientras que en el eje vertical se ven las métricas. El tamaño de cada círculo indica la cantidad de estudios que reportan esa métrica en ese año. Como se observa en la figura, el uso de métricas creadas por los investigadores basadas simplemente en observaciones sobre las clasificaciones está presentes en estudios de 2005 o anteriores. Las métricas de *accuracy* y *complexity* son tomadas en cuenta en investigaciones posteriores al 2004, pero de estas dos solo *complexity* se mantiene en los estudios más recientes. Las técnicas de *precision* y *recall* - casi siempre utilizadas en conjunto- se han mantenido a lo largo del tiempo. La *F-measure*, que incluye *precision* y *recall*, también se ha mantenido y es la métrica que presenta más reportes en los años 2011, 2012, 2013 y 2019.

4.7. Discusión

Los estudios primarios sobre técnicas de aprendizaje automático y minería de datos aplicadas al contexto de datos noticiosos presentan una gran diversidad de resultados.

Los 51 estudios en conjunto, realizaron un total de 64 reportes de técnicas. De ahí, no se puede decir que hay una tendencia por utilizar siempre una misma técnica, pues el paradigma de técnicas usada con más frecuencia llega apenas a 13 menciones. También así, los formatos de los datos utilizados no son uniformes. Probablemente dada la facilidad para encontrar noticias en la web, solo 18 estudios optaron por usar un conjunto de datos preestablecido. Las características cambiantes de las noticias y el hecho de considerar distintas fuentes de artículos, además de su característica cambiante y gran volumen, dan valor al uso de técnicas de minería para realizar esta clasificación de forma automatizada.

Las métricas empleadas también varían mucho en los estudios primarios. Las técnicas más utilizadas son *precision* y *recall* y la *F-measure* que incluye a estos dos. Por lo general, los autores solo se refieren vagamente a la selección de dichas métricas para evaluar sus resultados, pues no dan muchas explicaciones al respecto. Por ello, también los resultados obtenidos de los experimentos son, en la mayoría de casos, apenas estudiados. Es decir, en la mayoría de veces los artículos no dan mayores

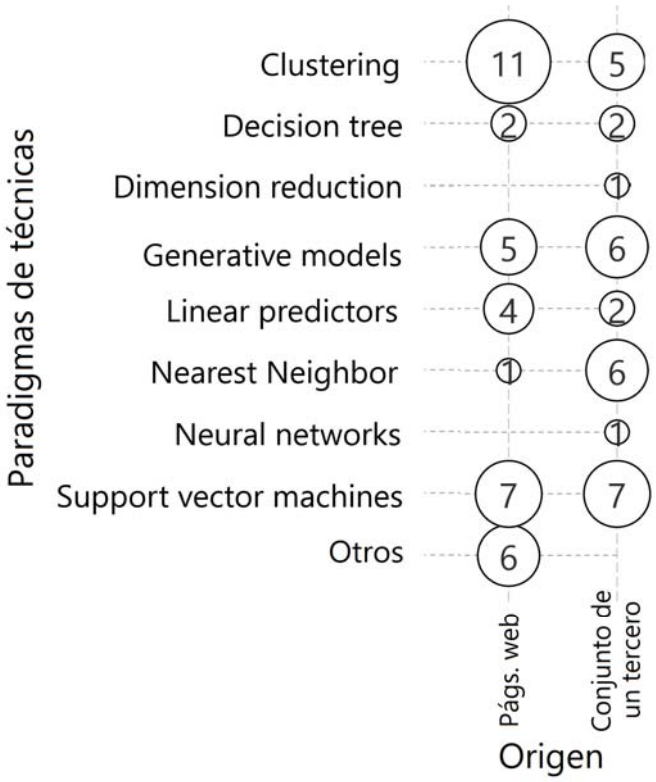


Figura 4.9: Cantidad de reportes de origen de datos por paradigmas de técnicas.

explicaciones y se limitan a apenas mencionar sus resultados.

La Figura 4.9 muestra la relación que hay entre las técnicas aplicadas por los estudios (eje vertical) y el origen de los datos usados (eje horizontal). El tamaño del círculo es proporcional a la cantidad de análisis que coinciden en dichos criterios. Puede observarse que con excepción de las técnicas de *neural networks* y *dimensionality reduction*, que se utilizan solo en estudios que trabajaron sobre un conjunto de datos de un tercero, todas las demás técnicas se relacionan con ambos tipos de documentos: los que provienen de un conjunto de datos descargado de internet y los que fueron extraídos de la Web como parte del estudio. Por otro lado, las técnicas de *clustering* se relacionan más con extracción directa de páginas Web, que con el uso de conjuntos de datos de un tercero. Esto tiene sentido puesto que el uso de conjuntos de datos de un tercero supone una preparación de los datos y agrupamientos previos, en cambio el uso de páginas Web directas requiere hacer esa agrupación prácticamente sin metadatos relacionados ni ordenados.

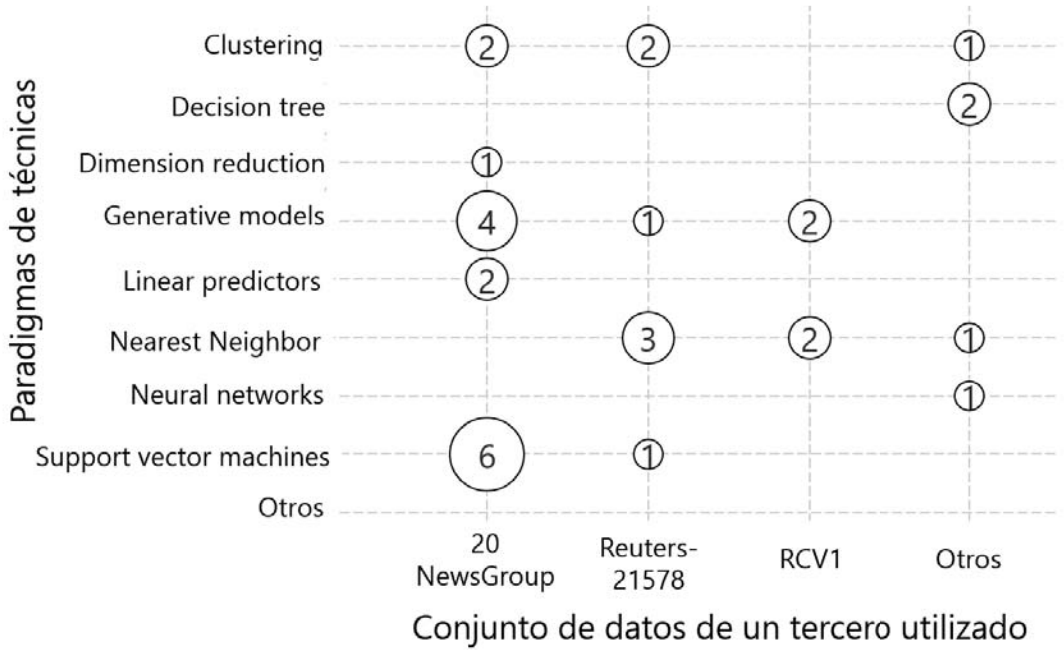


Figura 4.10: Cantidad de reportes de uso de conjuntos de datos de un tercero por paradigmas de técnicas.

La Figura 4.10 muestra el detalle de los conjuntos de datos que se utilizaron y que provienen de un tercero. De aquí, puede observarse que *20 Newsgroup* se utiliza primordialmente en estudios que usaron técnicas de *support vector machines* y *generative models*, mientras que *Reuters-21578* se utilizó en estudios mayoritariamente relacionados con *nearest neighbor* y *clustering*.

La Figura 4.11 expone las métricas que utilizaron los estudios primarios (eje horizontal), versus las técnicas que midieron (eje vertical). El tamaño de cada círculo refleja la cantidad de estudios que coinciden en los valores de los ejes. Para las técnicas de *clustering*, se han utilizado todas las métricas. La métrica *F-measure* no solo es la más empleada en los estudios, sino que también se ha usado para medir todas las técnicas. Por otro lado, las métricas *precision* y *recall* se han usado para medir la efectividad de casi todas las técnicas, con excepción del paradigma de técnicas *decision trees*. En el caso de la métrica de *complexity*, no se utiliza para medir las técnicas de *support vector machines*, *linear predictors*, ni *dimensionality reduction*.

La Figura 4.12 apunta la relación entre el origen de los datos usados por los estu-

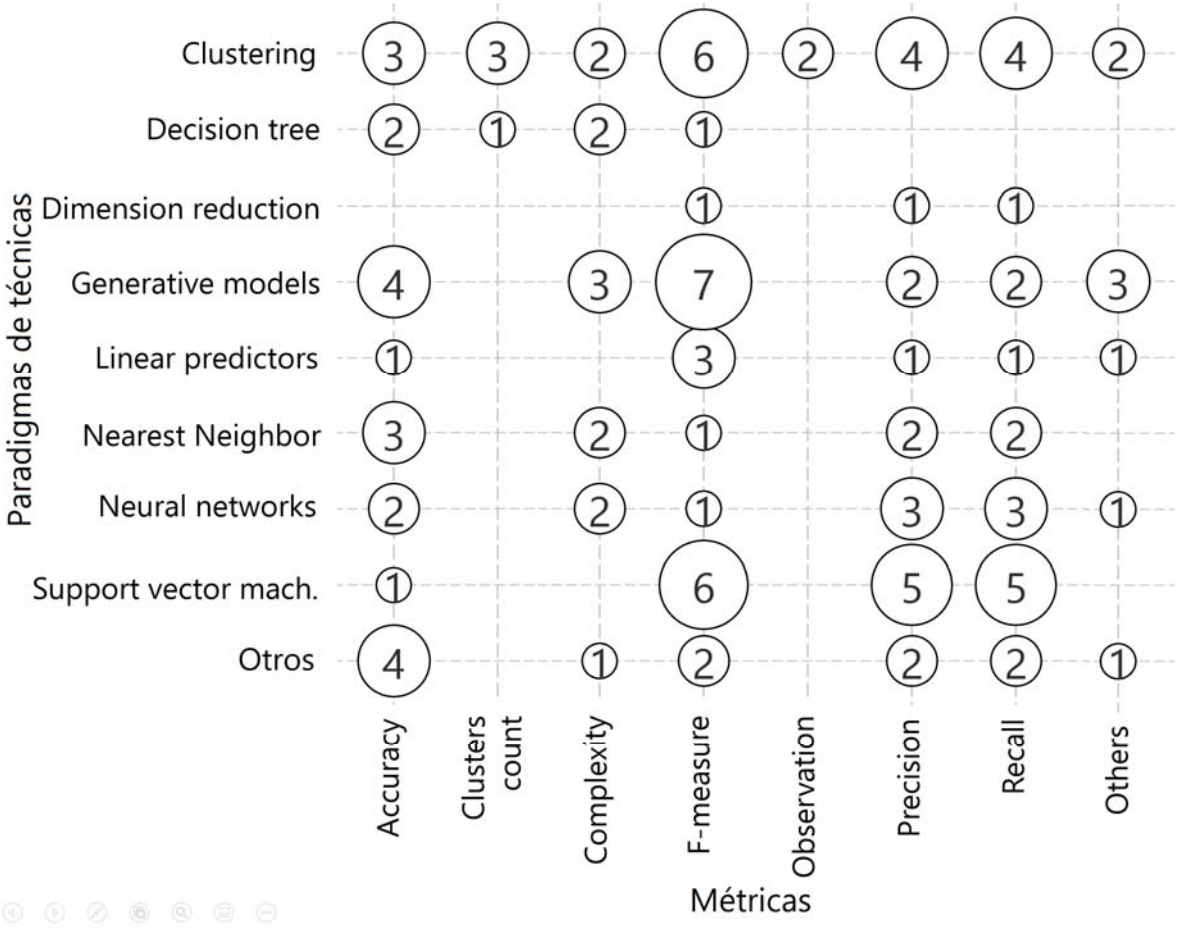


Figura 4.11: Cantidad de reportes de métricas utilizadas por paradigma de técnicas.

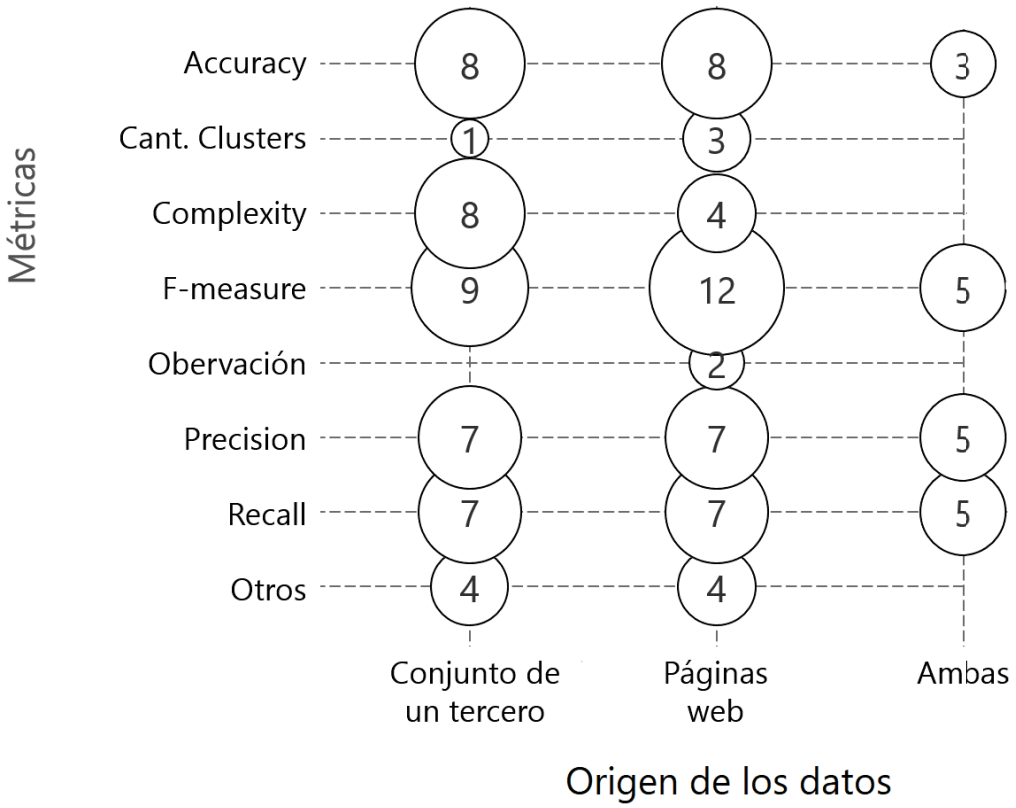


Figura 4.12: Cantidad de reportes de métricas por origen de datos sobre los que fue utilizado.

dios (eje horizontal) y las métricas que se utilizaron para medir los resultados sobre la clasificación de estos (eje vertical). El tamaño de cada círculo refleja la cantidad de estudios que coinciden en los valores de los ejes. Únicamente la métrica de simple observación se utilizó solo para datos extraídos de la Web. En todos los demás casos, existen relaciones entre todas las métricas y todos los orígenes de los datos. Además, estas relaciones se dan en frecuencias similares.

Los estudios que reportaron el mismo paradigma de técnicas y utilizaron una misma métrica no son comparables. Las características de cada experimento, sus contextos y variables que se aplicaron son diferentes, por lo que no es justo realizar una comparación de resultados absolutos. Realizar esa comparación de valores numéricos se sale del alcance de esta investigación, por lo que más adelante se plantea como trabajo futuro.

4.8. Lecciones aprendidas

El proceso de investigación de este estudio ha sido fuente de aprendizaje profesional para el investigador. Este proceso requirió disciplina en el método y atención a los detalles.

El análisis de artículos, la sistematización del proceso y la definición de los temas más importantes, hicieron que el investigador desarrollara mejores habilidades de lectura técnica, capacidad de síntesis y capacidad de observación para la investigación.

Por otro lado, a partir de la metodología empleada hubo muchas enseñanzas en relación con procesos de investigación en informática. El uso de estándares y sistematización es importante pues ayuda a agrupar grandes cantidades de datos y hacer análisis lo más objetivos posibles para obtener resultados prácticos.

La clasificación por grupos de las técnicas, las fuentes de datos y las métricas fueron acompañados por un esfuerzo adicional de estandarización que aportó también para la formación, pues permitió conocer nuevas técnicas.

En el uso de la metodología utilizada en este estudio, fue esencial poner atención al detalle. La información en la literatura toma más importancia al ser agrupada y comparada. Además, el orden de los datos recopilados y la relectura de estudios fueron herramientas que permitieron generar el conocimiento. Además, desde la pers-

pectiva de ingeniero de software, el uso de esta metodología es importante porque brinda visibilidad. En ese sentido, da la oportunidades de conocer técnicas aplicables para diferentes campos y contextos, permite enfocar los esfuerzos en el área de interés y poder hallar respuestas concretas.

Hubo esencialmente dos fases que fueron las más retadoras en el proceso. En primer lugar, el procedimiento de inclusión/exclusión de los resultados obtenidos de las bases de datos tomó mucho más tiempo de lo previsto. Esta etapa además requirió inspeccionar la mayoría de los artículos, pues no se podía detectar la pertinencia del documento a partir únicamente de su resumen y palabras clave. En segundo lugar, el sistema de extracción y agrupamiento de información no fue sencillo. El mayor reto de esta etapa consistió en estudiar cómo interpretar la información contenida en los estudios para poder interpretarla en conjunto. Ello requirió hacer mucho retrabajo, y tener una visión más profunda que permitiera comparar y contrastar los reportes de los autores.

Finalmente, el uso de una metodología sistemática permitió disgregar el problema de investigación en un conjunto de preguntas que se pudo responder. Esto hizo posible el procesamiento y clasificación de los datos presentes en la literatura. También permitió extraer la información con la seguridad de que el material sustraído era el que interesaba en la investigación. A futuro y en la práctica de la profesión, el uso de estas técnicas puede ser útil para estudiar soluciones antes de implementarlas. El mapeo permitiría, como en este caso, enfocarse en un área de interés para revisar opciones, analizar resultados, adquirir conocimientos y finalmente tomar decisiones técnicas sobre la Ingeniería de *Software*.

La metodología utilizada brinda algunas ventajas. Como se mencionó, la agregación de evidencia existente permite tomar decisiones para ser aplicadas en la práctica, pues ofrece un panorama general como insumo y de acuerdo con los resultados generados por otras investigaciones. En el caso de este estudio, hizo posible el agrupamiento de información para complementar ciertas informaciones con otras, y permitió que fueran presentadas de forma comprensible y además mitigando riesgos de sesgo. No obstante, esta metodología también tiene desventajas, no siempre permite hacer comparaciones debido a las características propias de cada estudio. Estas técnicas requieren invertir muchos recursos, y además mucho análisis sobre fuentes

que terminan siendo descartadas en la investigación.

4.9. Conclusiones

Este trabajo planteó caracterizar los estudios primarios que han tratado de crear clasificaciones o agrupamientos temáticos a partir de noticias web.

Las técnicas de minería de datos y minería de texto pueden ayudar mucho en la clasificación de los contenidos noticiosos. Dado el volumen de datos y su complejidad, se hace necesaria una herramienta automatizada que sea capaz de solventar el problema de la clasificación temática. Existen técnicas que analizan los contenidos de textos en general, pero no se encontraron estudios especializados únicamente en el contexto de noticias. Esto podría deberse a la complejidad del problema, considerando la constante generación de noticias y los diferentes estilos de escritura de los periodistas.

Las técnicas empleadas en los estudios primarios analizados buscan generar agrupamientos temáticos de la información, eliminando de los datos lo que no sea relevante y tratando de establecer similitudes entre los textos de cada artículo. No obstante, se trata de técnicas generales que se intentan aplicar al contexto noticioso y sobre documentos que describen el acontecer nacional o mundial.

En general, las técnicas de *clustering*, *support vector machines* y *generative models* son las más utilizadas, mientras que la métrica más frecuente usada para evaluar fue F-measure. Sin embargo, los estudios no muestran consenso en la respuesta de ninguna de las tres preguntas de investigación: la especificación de las técnicas, los datos y las métricas.

Hay gran variedad de resultados diferentes y casi todo tipo de combinaciones entre técnicas utilizadas, datos que se utilizaron y métricas evaluadoras. De esto se desprende que los estudios primarios no reportan una única forma de hacer clasificación temática de noticias, y no hay consenso sobre la mejor manera de hacerlo.

En relación con el ámbito profesional, este estudio provee un mapeo de técnicas que pueden ser utilizadas como directorio para aplicarse sobre diferentes tipos de datos. En el campo de la investigación, se evidencia la importancia de mejorar los reportes y de proveer mayores detalles para poder realizar comparaciones. También, el

planteamiento de este estudio secundario abre la puerta a realizar estudios primarios relacionados. Desde una perspectiva académica, la información brindada da la oportunidad de profundizar en la aplicación de las técnicas sobre diferentes contextos, y provee ideas para la exploración de dichas técnicas desde un contexto de aprendizaje y descubrimiento. Desde esta temática, se plantea un aporte sobre los cursos de bases de datos avanzados, de calidad del software y de inteligencia artificial. Sin embargo, es aplicable en general en toda la malla curricular, así como en los cursos del énfasis de Ingeniería de Software.

Como trabajo futuro, se plantea ampliar el estudio para analizar clasificación temática en contextos más amplios y generales, haciendo una investigación más extensa que permita plantear las mejores posibilidades reportadas y explorar cómo estas serían utilizadas en la clasificación automática de noticias. Además, se propone poder hallar la forma de analizar los resultados en las métricas de efectividad de cada estudio para poder realizar comparaciones de rendimientos.

Se hace la recomendación de que los estudios primarios documenten mejor lo que hicieron para incrementar su reproducibilidad, y aumentar la confiabilidad de la evidencia empírica generada. Para futuros estudios, se recomienda aumentar el nivel de detalle en explicaciones de casos, justificar mejor las selecciones de técnicas o métricas, así como mencionar las características específicas de su cuerpo de datos. Estas medidas mejorarían sustancialmente la calidad de los estudios pues permitiría profundizar, y hacer más entendibles los aspectos técnicos al contrastar propuestas más profundamente.

A partir del capítulo de la memoria se desarrolló un artículo científico que fue enviado y aceptado en la Intelligent Systems Conference (IntelliSys) 2020 que se desarrolló el 3 y 4 de setiembre del 2020 en Amsterdam, Países Bajos. El artículo fue publicado en el Springer series “Advances in Intelligent Systems and Computing” e indexado en ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar y Springerlink. En el Apéndice 4.C se encuentra el artículo publicado.

Apéndice

4.A. Lista de estudios primarios incluidos

En el Cuadro 4.9 se muestran los artículos secundarios incluidos en el estudio.

Cuadro 4.9: Lista de estudios primarios incluidos.

ID	Título	Año	Est.
1	News item extraction for text mining in Web newspapers	2005	[21]
2	Evolving type-2 Web news mining	2017	[64]
3	Dynamic pattern mining: An incremental data clustering approach	2005	[57]
4	Modular preference Moore machines in news mining agents	2001	[59]
5	Topic detection and tracking for news Web pages	2007	[22]
6	Web service to deliver filtered RSS items to a mobile application	2012	[44]
7	Content mining of microblogs	2012	[33]
8	Assigning Web news to clusters	2010	[17]
9	Scalable Web mining with newistic	2009	[58]

Continúa en la página siguiente.

ID	Título	Año	Est.
10	Web news mining in an evolving framework	2016	[4]
11	Topic-based clustering of news articles	2004	[23]
12	SVM-based Web content mining with leaf classification unit from DOM-tree	2017	[45]
13	Web News Mining Using New Features: A Comparative Study	2019	[18]
14	Delivering categorized news items using RSS feeds and Web services	2010	[46]
15	Effect of named entities in Web page classification	2012	[47]
16	Web page classification on news feeds using hybrid technique for extraction	2019	[65]
17	Scalable clustering of news search results	2011	[25]
18	Scalable text classification as a tool for personalization	2009	[52]
19	Document Clustering using message passing between data points	2013	[62]
20	Scalability of text classification	2006	[53]
21	Categorizing online news articles using penguin search optimization algorithm	2018	[66]
22	Product news summarization for competitor intelligence using topic identification and artificial bee colony optimization	2015	[26]
23	Topics modeling based on selective Zipf distribution	2012	[34]

Continúa en la página siguiente.

ID	Título	Año	Est.
24	Learning to group Web text incorporating prior information	2011	[35]
25	Building domain keywords using cognitive based sentences framework	2018	[27]
26	Supporting information retrieval in RSS feeds	2010	[28]
27	Evaluation of text document clustering approach based on particle swarm optimization	2013	[29]
28	Concept Based Document Clustering Using K Prototype Algorithm	2018	[30]
29	Large-scale hierarchical text classification without labelled data	2011	[51]
30	Graph based text representation for document clustering	2015	[24]
31	A novel text mining approach based on TF-IDF and Support Vector Machine for news classification	2016	[19]
32	Tracking Topic Evolution in News Environments	2008	[54]
33	A probabilistic framework for short text classification	2018	[55]
34	An Event Detection Algorithm Based on Improved STC	2008	[56]
35	Dynamic entity and relationship extraction from news articles	2012	[31]

Continúa en la página siguiente.

ID	Título	Año	Est.
36	What's going on out there right now? A beehive based machine to give snapshot of the ongoing stories on the Web	2012	[63]
37	Combining statistical similarity measures for automatic induction of semantic classes	2005	[48]
38	DOCUMENT CLUSTERING WITH BURSTY INFORMATION	2012	[49]
39	Event-based cross media question answering	2016	[36]
40	An adaptive personalized news dissemination system	2009	[37]
41	Text classification using ensemble of non-linear support vector machinesOpen Access	2019	[50]
42	Event extraction from heterogeneous news sources	2006	[32]
43	Managing content with automatic document classification	2004	[38]
44	Enhancing document search with a dynamic artificial neural network(Book Chapter)	2014	[60]
45	Automatic text summarization with genetic algorithm-based attribute selection	2004	[39]
46	Neural network agents for learning semantic text classification	2000	[61]
47	Self-switching classification framework for titled documents	2009	[40]

Continúa en la página siguiente.

ID	Título	Año	Est.
48	Scalable text classification with sparse generative modeling	2012	[41]
49	News text classification model based on topic model	2016	[42]
50	Comparing SVM and naïve Bayes classifiers for text categorization with Wikitology as knowledge enrichment	2011	[43]
51	List based matching algorithm for classifying news articles in NewsPage.com	2008	[67]

4.B. Evaluación de calidad de los estudios primarios

En el Cuadro 4.10 se muestran los resultados de la evaluación de calidad de los estudios incluidos en el estudio.

Cuadro 4.10: Evaluación de calidad de los estudios primarios.

ID	Año	Est.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
1	2005	[21]	1	0	1	0	1	1	1	1	6
2	2017	[64]	2	1	2	1	2	1	2	2	13
3	2005	[57]	2	2	2	2	2	1	2	1	14
4	2001	[59]	1	1	0	1	1	1	2	2	9
5	2007	[22]	0	2	1	0	1	1	0	1	6
6	2012	[44]	1	1	2	0	1	1	0	2	8

Continúa en la página siguiente.

ID	Año	Est.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
7	2012	[33]	2	0	1	1	2	0	2	2	10
8	2010	[17]	2	2	2	2	2	2	2	2	16
9	2009	[58]	2	2	0	0	1	0	1	0	6
10	2016	[4]	2	1	2	2	2	1	2	2	14
11	2004	[23]	2	2	2	1	1	0	1	2	11
12	2017	[45]	2	0	0	0	2	0	0	0	5
13	2019	[18]	2	2	2	2	2	1	2	2	15
14	2010	[46]	2	2	1	0	1	0	0	1	7
15	2012	[47]	2	1	2	2	2	1	2	2	14
16	2019	[65]	2	1	0	1	1	0	1	1	7
17	2011	[25]	2	1	1	2	1	0	2	2	11
18	2009	[52]	2	2	0	0	1	1	2	1	9
19	2013	[62]	1	2	2	2	1	0	1	1	10
20	2006	[53]	1	1	0	0	1	0	0	1	4
21	2018	[66]	2	2	0	2	1	0	1	2	10
22	2015	[26]	2	2	1	0	2	0	0	1	8
23	2012	[34]	0	1	0	0	0	1	0	0	2
24	2011	[35]	1	2	1	1	2	1	1	2	11
25	2018	[27]	1	1	0	0	1	0	0	1	4
26	2010	[28]	2	2	2	2	2	1	1	1	13
27	2013	[29]	2	2	2	2	2	2	2	2	16
28	2018	[30]	2	2	2	2	2	1	1	1	13

Continúa en la página siguiente.

ID	Año	Est.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
29	2011	[51]	2	1	1	1	1	0	1	1	8
30	2015	[24]	2	1	2	2	2	2	2	2	15
31	2016	[19]	2	1	1	2	2	0	1	2	11
32	2008	[54]	2	1	1	0	1	0	0	0	5
33	2018	[55]	2	2	2	2	2	0	1	1	12
34	2008	[56]	2	2	1	1	2	0	1	1	10
35	2012	[31]	2	2	0	1	2	0	0	2	9
36	2012	[63]	2	2	0	0	2	0	0	0	6
37	2005	[48]	2	2	1	2	2	1	2	2	14
38	2012	[49]	2	2	0	2	2	0	1	2	11
39	2016	[36]	2	2	1	0	2	0	0	1	8
40	2009	[37]	2	1	0	0	0	0	2	2	7
41	2019	[50]	2	1	1	0	2	0	2	1	9
42	2006	[32]	2	2	1	1	2	0	2	2	12
43	2004	[38]	2	2	1	2	2	1	2	2	14
44	2014	[60]	2	2	0	1	2	0	1	2	10
45	2004	[39]	2	1	1	1	2	0	2	2	11
46	2000	[61]	2	1	0	1	2	0	2	2	10
47	2009	[40]	2	2	0	1	1	0	2	2	10
48	2012	[41]	2	2	1	1	2	0	2	2	12
49	2016	[42]	2	2	0	2	2	0	2	2	12
50	2011	[43]	2	1	1	1	2	1	1	1	10

Continúa en la página siguiente.

ID	Año	Est.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
51	2008	[67]	2	2	2	1	2	1	2	2	14

4.C. Artículo

A partir del capítulo de la memoria se desarrolló un artículo científico que fue enviado y aceptado en la Intelligent Systems Conference (IntelliSys) 2020 que se desarrolló el 3 y 4 de setiembre del 2020 en Amsterdam, Países Bajos. El artículo fue publicado en el Springer series “Advances in Intelligent Systems and Computing” e indexado en ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar y Springerlink.

Automatic Classification of Web News: A Systematic Mapping Study

Mauricio Pandolfi-González( ), Christian Quesada-López( ), Alexandra Martínez( ), and Marcelo Jenkins( )

Universidad de Costa Rica, San Pedro, Costa Rica
{mauricio.pandolfi,cristian.quesadalopez,alexandra.martinez, marcelo.jenkins}@ucr.ac.cr

Abstract. The number of news articles published on the Web has had a dramatic increase. News websites are overwhelmed daily with articles, and their processing and classification is a challenge. Reading news from the web has become an important citizen's information source, and its classification can show relevant information about social or cultural patterns on society. In this context, techniques that can automatically analyze and classify news articles are essential. In particular, data mining and machine learning techniques have been applied for the classification of web news, as they can detect structural patterns based on documents characteristics. Their use requires specialized text processing and summarizing techniques. The objective of this study is to characterize data mining and machine learning techniques used for the web news classification, the datasets used, and the evaluation metrics. We performed a systematic literature mapping of 51 primary studies published between 2000 and 2019. We found that the most used techniques fall into these paradigms: clustering, support vector machines and generative models. Also, 33 studies used online data extracted from Internet's news web pages, while 25 downloaded a previously published dataset. The most common metric is the F-measure, with 25 reports. In summary, several data mining and machine learning techniques have been applied to the automatic classification of web news, showing some trends regarding the techniques, datasets, and metrics.

Keywords: Machine learning · Data mining · Topic-based clustering.

1 Introduction

Studying the news is key in the social and political context of modern information societies. Performing a thematic analysis from a classification of news can help understand behaviors and discover social patterns immersed in our culture [1].

The emergence of technologies such as mobile applications and digital social networks has generated new communication concepts: digital conversion, journalism 3.0 and multimedia journalism. These changes have enabled a new paradigm

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1251, pp. 1–17, 2021.
https://doi.org/10.1007/978-3-030-55187-2_41

that involves different models for writing, editing and production of journalism-related materials [2].

In the field of journalism, production and volume have greatly increased as well, raising the need for an automated thematic analysis of the information [3].

Nowadays, automatic systems that treat, analyze and classify web news articles are essential for the management of web news and also for user recommendation tasks [3].

Data mining and machine learning techniques make possible structural pattern detection on documents, based on their own characteristics [7,8].

News characteristics lead to various linguistic and computational challenges for text mining: (*i*) indexing, taxonomic categorization, partial redundancy, and data streams, (*ii*) language and meaning, (*iii*) nonstandard language and subjectivity, (*iv*) thematic diversity and new forms of categorization, and (*v*) context and its impact on content and meaning [5].

In recent years, there have been many approaches related to classification, grouping, categorization and summary of news articles [3].

The main objective of this research is to characterize data mining and machine learning techniques that have been used for the topic classification of news extracted from the web. We also report the datasets used, and the metrics for evaluating their results. To guide this study, we defined two research questions:

RQ1. What machine learning and data mining techniques have been used to classify news-related data extracted from the web?

RQ2. What datasets and metrics have been used to evaluate machine learning and data mining techniques in topic classification of web news?

The remainder of our paper is structured as follows. Section 2 presents the background. Section 3 discusses some related work in the area. Section 4 explains the methodology used in this study. The results are presented in Sect. 5. Finally, the conclusions are outlined in Sect. 6.

2 Background

Text mining is a research area positioned at the intersection of information retrieval, data mining, natural language processing, and machine learning. We subsume knowledge-detecting as well as knowledge-extracting techniques under the term text mining. Following this definition text mining embraces: text preprocessing, automated language detection of texts, summarizing or abstracting texts, automated text categorization, text clustering, and others [4].

Text mining is used for tasks such as description, classification, prediction, search, recommendation, or summary of the textual parts of news and blogs, extracting topics, events, opinions, sentiments, and other aspects of content [5].

Document classification refers to a process of assigning one or more labels for a document from a predefined set of labels. The main issues in document classification are connected to classification of free text giving document content,

for instance: (a) classifying Web documents on the content topic as being about arts, education, science, and others, (b) classifying news articles by their topic: politics, technology, science, health, and others, (c) classifying movie reviews by their opinion: positive review, negative review. Machine learning methods applied to document classification are based on general classification methods adjusted to handle some specifics of text data [6].

Machine learning is the branch in informatics that uses automatic algorithms to infer structures in data and ways to validate those. In this study field, systems are developed that use techniques for automation of processes, which emulates an understanding of the data. Thus, the concept of mining large amounts of data is related to machine learning [8].

Data mining is about discovering information that is hidden within the data, so that the results can help researchers solve problems. This process can be automatic or semiautomatic, and involves the system's capability to learn [8].

Web mining refers to the use of data mining techniques in contexts in which you have to incorporate additional information from the web documents, according to its nature [8].

Machine learning techniques are very diverse, but some of them share general characteristics in terms of parametrization, used formulas, or in the way of understanding the classification problem. Shalev-Shwartz and Ben-David [9] categorize them in the following paradigms:

1. *Linear predictors:* These techniques try to find a classification based on linear functions. In this category, there are techniques such as halfspace, linear regression, logistic regression, among others.
2. *Boosting:* these techniques are a generalization of the linear predictors, aiming to obtain better results. Adaboost is an example.
3. *Support vector machines:* These techniques assume the challenge of making a more complex sample. In general terms, consist in separating a training set with big margin if all the examples are not only correct but also that they are far from the separation hyperplane.
4. *Decision trees:* These are predictors that determine the label associated to an instance, through the exploration of a data tree. Some examples of this paradigm are ID3 and random forests.
5. *Nearest neighbor:* The central idea of these techniques is to take as a parameter a training set, and then predict the classification of a new instance based on the similarities with their nearest neighbor in the training step. For example, K-nearest neighbors technique.
6. *Neural networks:* These techniques are based on the development of artificial neural networks model, which emulates the behavior of a human brain. It consists of a set of elements (neurons) that connect in a complex net of communication that produce classifications of data.
7. *Clustering:* These techniques identify groups inside a set of elements, in a way that the most similar ones end in the same group, and the different ones are separated in different groups. Inside this paradigm there are techniques based on links such as k-means, spectral, graph cut, information bottleneck, among others.

8. *Dimensionality reduction:* These techniques take the data in a dimensional space and map them in a smaller space (with fewer dimensions), applying a lineal transformation on the original data. Some techniques in this category are principal component analysis and compressed sensing.
9. *Generative models:* They assume that the distribution of data has a parametric form and aim to estimate those parameters. In this category, there are techniques such as Maximum likehood estimator, Naive Bayes, Linear Discriminant Analysis (LDA), Latent variables, EM algorithm, and Bayesian reasoning.

The use of mining techniques, referred as web news mining, serves as an important and powerful idea to manage all the information and the knowledge that is encapsulated in large collections of web news [3].

3 Related Work

We were not able to find similar studies in the context of automatic classification of news extracted from the web. Some studies refer to the use of machine learning, data mining, or web mining techniques in general, but not exclusively in news-related contexts. We describe these works below.

Sebastiani [10] presents an analysis of the machine learning algorithms for the categorization of texts and explains the construction of different text classifiers until the year 2002. This survey discusses the main approaches to text categorization using the machine learning paradigm. The authors found and described probabilistic, decision trees, decision rule, regression, on-line, Rocchio, neural networks, example, and support vector machines oriented classifiers. The study also analyzes the way of evaluating different approaches and concludes that from the early nineties. The effectiveness of text classifiers has dramatically improved by the use of machine learning methods for the text classification field.

In 2015, Irfan et al. [11] report a review of the text mining techniques applied to artificial social networks contents. These data share some characteristics with on-line news, such as their volatility and their huge volume. They provided a study focused on the application of classification text mining techniques where the data is unstructured. The study presents a schema of techniques grouped by their characteristics, mainly divided into 3 great groups: hierarchical, partitional, and ontology-based clustering. The study concludes that extracting logical patterns with accurate information from unstructured data is a critical approach to perform.

Bharti and Babu [12] identify the literature up to 2017 on the automatic extraction of keywords from texts, to make summaries. They established five main types of summarization process. One of them emphasizes in machine learning techniques, but the study did not find any reports of using this approach. They conclude that there is a lack of information and standardization of researches in this field.

Castillo and Cervantes [13] present a general view on the representation of algorithms in the literature, that try to solve text classification that uses graph-based techniques. The motivation of this analysis was to show how co-occurrence graphs can be used to represent text documents and how this can be a valuable asset. This study described 2 main approaches for classification: feature-vector, and similarity. This study finally mentioned that graphs are an alternative that is at the same level as many other of the state of art techniques, by implementing easy to build representations and having a relatively fast performance.

4 Methodology

We conducted a systematic mapping study to select and analyze existing literature on data mining and machine learning techniques used for automatic classification of web news. We followed the methodology stated in the guidelines of Petersen [14] and the recommendations of Kitchenham [15].

The objective of this research, based on the Goal Question Metric (GQM) model [16] is *to analyze* machine learning and data mining techniques for classification of news *with the purpose of* characterizing them, *in terms of* the techniques, data sources, and metrics for measuring effectiveness *from the point of view of* the researchers *in the context of* news extracted from the web.

4.1 Search Strategy and Selection Process

As the first step, an exploratory search was made to identify appropriate search terms in accordance with the objective. We calibrated the search string based on a set of relevant studies selected as control articles [18–20]. Based on the results and insights from the exploratory search phase, we selected the following search string:

```
((web AND news) AND (classif* OR cluster*) AND (mining OR
''machine learn*")␣AND␣NOT␣video␣AND␣NOT␣sentiment␣AND␣NOT
␣''fake␣news")
```

The automatic search for relevant papers were conducted up to October 2019 in three digital libraries: *Scopus*, *IEEE Xplore*, and *Web of Science*. The mapping and analysis was made during 2019. We looked for studies that contained the search terms anywhere in the title, abstract or keywords fields.

The resulting amount of studies in each digital library was: 310 results in *Scopus*, 175 in *IEEE Xplore* and 63 in *Web of Science*. The articles were retrieved in a spreadsheet, duplicates were eliminated, inclusion and exclusion criteria were applied, the information from each was extracted, and finally they were analyzed based on the research questions.

Inclusion and Exclusion Criteria. Exclusion criteria are used to narrow down the initial set of search results by the process of elimination. Studies that met one (or more) of these criteria were excluded from the research:

E1. Not available in full text, after an intensive search.
E2. Not written in English.
E3. Not a primary study.
E4. The news articles that the study classifies are not in English.

Inclusion criteria defines the attributes that are essential for a study to be selected for analysis. Included studies had to fulfill all the inclusion criteria. The inclusion criteria were as follows:

I1. The study analyzes news by techniques reported as data mining or machine learning.
I2. The study categorizes news by topic.
I3. The study analyzes news in text format.

The inclusion and exclusion process was made based on title, summary, keywords and in some cases full read. Figure 1 summarizes the process of study selection for this secondary research. The search string produced an initial set of 548 potentially relevant studies. The inclusion and exclusion criteria were applied to 421 studies (removing duplicates), considering only the title, abstract and keywords fields. Any paper that was irrelevant became excluded. In general, we tended to include rather than exclude potentially relevant papers. After applying the exclusion and inclusion criteria, we were left with 53 studies. It is important to mention that only 13 studies were excluded by the first exclusion criterion. Our analysis set includes a total of 51 papers after detailed reading. Each paper is assigned a unique reference code (prefixed by the letter "S") for the purpose of our analysis (S01-S51). The complete list of selected papers along with their quality assessment, and extraction form, is available at https://tinyurl.com/tw37ovg.

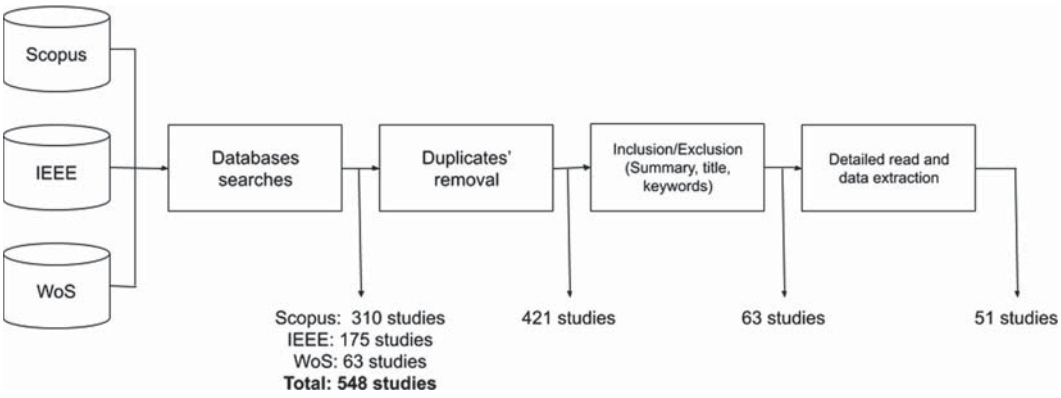


Fig. 1. Summary of the studies' selection process.

4.2 Quality Assessment of Included Studies

The quality assessment of the studies determined the level of detail related to the aspects that was included in this research. Each criterion was pointed between 0 and 2, according to this scale: 2 points if it fulfilled the criterion, 1 point if partially, and 0 points if it does not.

The criteria used for the evaluation were: (Q1) The study describes the web news dataset. (Q2) The study explains the steps for implementing the proposed approach. (Q3) The study describes the preprocessing steps performed on the data before applying the techniques. (Q4) The study explains the metrics used for measuring the effectiveness of the used techniques. (Q5) The study defines its main objective. (Q6) The study presents the research questions. (Q7) The study details the study design. (Q8) The study explains the methodology.

According to the defined scale, the maximum quality score that any of the studies can reach is 16 points and the minimum is 0. The average quality score obtained was 10.04, indicating that, on average, the studies were of fair quality.

4.3 Data Extraction and Classification

To apply the data extraction process, we created a form with aspects to be extracted from the selected studies based on the research questions. We extract each of the data items from each study and organize the data from all studies into common categories.

The extracted information was ordered in terms of the used techniques (RQ1), the datasets and the metrics used for measuring the effectiveness of the techniques (RQ2). When one article used several techniques, datasets or metrics, these were considered as separate values.

The analysis was focused on the specific aspects that allow answering the research questions.

To answer RQ1, the analysis consists in identifying the techniques reported in the studies and counting the studies used by each technique. The techniques classification was conducted according to the taxonomy proposed by Shalev-Shwartz et al. [9].

For RQ2, the metrics that evaluated the performance of the techniques and the data sets were identified.

The classification of metrics was done according to the description by Han et al. [17].

4.4 Threats to Validity

In this research, there were detected some validity threats for the mapping study. We briefly discuss the threats to validity of our mapping study.

The selection of the databases used for this research was made considering that these databases are recognized in the research community and to the area of software engineering, and they have great coverage of studies. The search string was a result of making several pilot searches and tests that helped to refine it.

The criteria for inclusion and exclusion were established in order to avoid bias. In case of doubt in deciding whether a study fulfilled or not a criterion, the decision was made based on a full read. Also, due to the scope of the research, there were not executed exhaustive researches over other sources different than the mentioned online databases.

The process of analyzing the studies and extracting data was performed by the first author and validated by a second one. Also, during the selection process, several times the extraction form was verified and validated, so that it was always coherent with the research questions. When the reported techniques were not classified by the authors of the study, some validations were made according to the theory and the taxonomy used [9]. Finally, for minimizing the risks on the way of presenting the results, this research respected all the established protocols and methodology.

5 Results and Discussion

Here we present the results our systematic mapping, according to our specific research questions.

5.1 Machine Learning and Data Mining Techniques Used for Classification of Web News (RQ1)

The first research question identifies the most commonly used machine learning and data mining techniques to classify web news topics. All 51 studies contributed to answer this research question.

The complete list of data mining and machine learning techniques reported by the primary studies are shown in Table 1. This table includes the paradigm in which the techniques are classified (according to [9]), the quantity of studies that report each technique, and the study reference. Notice that each study could use several techniques.

From this table we see that the three most common paradigms found were clustering, support vector machines and generative models. These paradigms group 58% of the reported techniques. These paradigms comprise algorithms for classification.

Figure 2 shows the frequency of use for each paradigm (the number of studies that reported techniques from those paradigms). The results show that clustering is the most frequently used paradigm (with 13 studies), followed by support vector machines (with 10 studies), and generative models (with 11 studies). Clustering related techniques identify differentiated groups from a set of elements, according to their similarities [9]. In news data, this means that clustering techniques divide the contents of the articles according to the relation of their terms. Support vector machine techniques use supervised learning. Given a training set, these techniques analyze classify each element in one of two categories, trying for the gap between the groups to be as big as possible. Then, new examples use this mapping for deciding to which of the 2 categories they belong to [9]. Generative

models oriented techniques, on the other hand, start assuming that the distribution of data has a parametric form and tries to estimate those parameters [9].

Figure 3 shows the number of studies reported each year per paradigm. Paradigms are shown in the vertical axis, while years appear in the horizontal axis. The size of a circle is proportional to the number of studies that reported that paradigm in that year. As shown, the included studies were published from 2000 to 2019. From this figure it is hard to devise a trend in the data. Basically, there is no strong tendency of some paradigms to be used more then others over time.

Some paradigms of techniques are being less reported (linear predictors, dimensionality reduction), and others seem to predominate in the most recent years (clustering, support vector machines). The decision trees, nearest neighbor, and neural networks related techniques were absent during several years, but they appeared again in the year 2019.

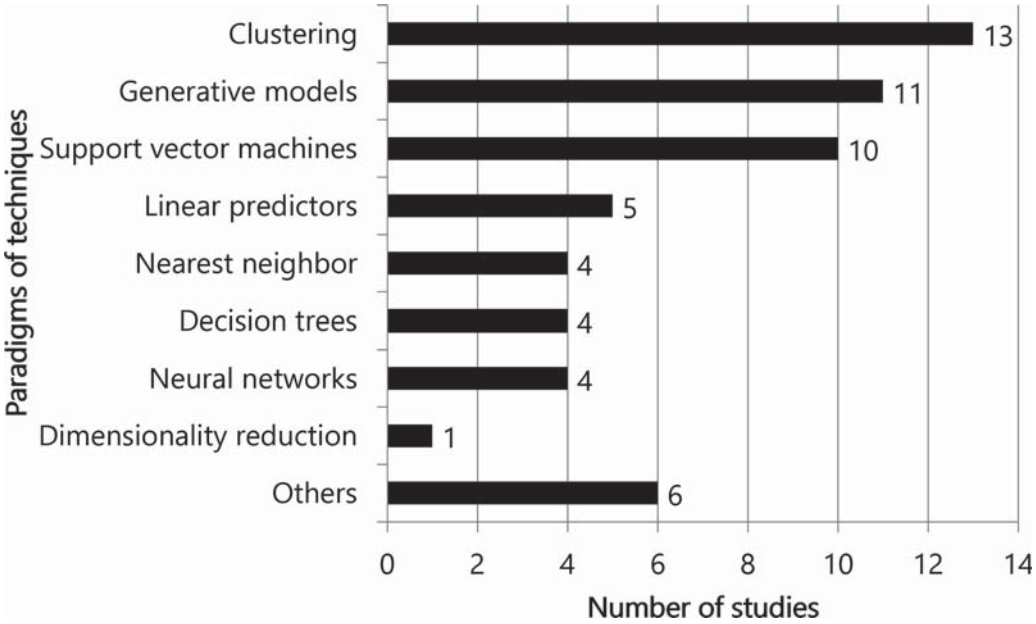


Fig. 2. Frequency of data mining and machine learning techniques.

5.2 Datasets and Metrics Used to Evaluate Data Mining and Machine Learning Techniques (RQ2)

The second research question aims to identify commonly used performance metrics and datasets to evaluate data mining and machine learning techniques.

First, we identify and characterize the data sources used for classification of news, together with the preprocessing techniques applied to the datasets. Second, we report the metrics used for measuring the effectiveness of data mining or machine learning techniques.

Table 1. Data mining and machine learning techniques reported.

Paradigm	Technique	Qty	Studies
Clustering	K-means, k-medians, complete-link graph, pairwise linkage, Locality Sensitive Hashing (LSH), F2N-Rank graph, RSS Organizing and Classification System (ROCS), Particle swarm optimization, Probabilistic named Entity Recognition	13	[S02, S04, S05, S06, S07, S08, S09, S10, S11, S12, S13, S14, S49]
Generative models	Multinominal naive bayes (MNNB), Latent Dirichlet Allocation (LDA), Expectation–maximization (EM) algorithm, DBSCAN, Naive bayes	11	[S16, S24, S25, S26, S27, S28, S29, S30, S31, S23, S50]
Support vector machines	Standart, non-linear	10	[S03, S15, S16, S17, S18, S19, S20, S21, S22, S23]
Linear predictors	Rocchio, Hierarchical Topic Model with Ontological Guidance, hierarchical temporal topic tracking, Probabilistic matrix	5	[S32, S33, S34, S35, S36]
Decision trees	Suffix Tree, Standart Decision tree technique, Improved STC, C4.5	4	[S01, S05, S28, S37]
Nearest neighbor	K-Nearest Neighbor	4	[S01, S27, S38, S39]
Neural networks	Neural Preference Moore Machine, Artificial neural network technique, Dynamic artificial neural network (DAN2), recurrent plausibility networks	4	[S01, S40, S41, S42]
Dimensionality reduction	Document Classification and Knowledge Extraction	1	[S43]
Others	Fuzzy systems, keyword analysis using Wordnet, Beehive, table based matching	6	[S44, S45, S46, S47, S48, S51]

Characterization of Datasets. Table 2 summarizes the datasets used by primary studies. This table contains the dataset name, the number of times the dataset was reported (Qty), the studies' references, the description, the size of the dataset, and the classifiers.

In each of the analyzed studies, the authors established sets of news for which they applied machine learning and data mining techniques. The characteristics of these datasets is described in Table 2.

In total, 33 studies used extraction from news web pages, and 25 used a dataset from the web.

A trend of use can be noted, where the *20 Newsgroup* dataset was reported 15 times. This is a collection of documents related to news, retrieved by Ken Lang, and grouped by topic.

Seven studies reported the use of *Reuters-21578*. This is a collection from Carnegie Group Inc. and Reuters Ltd.

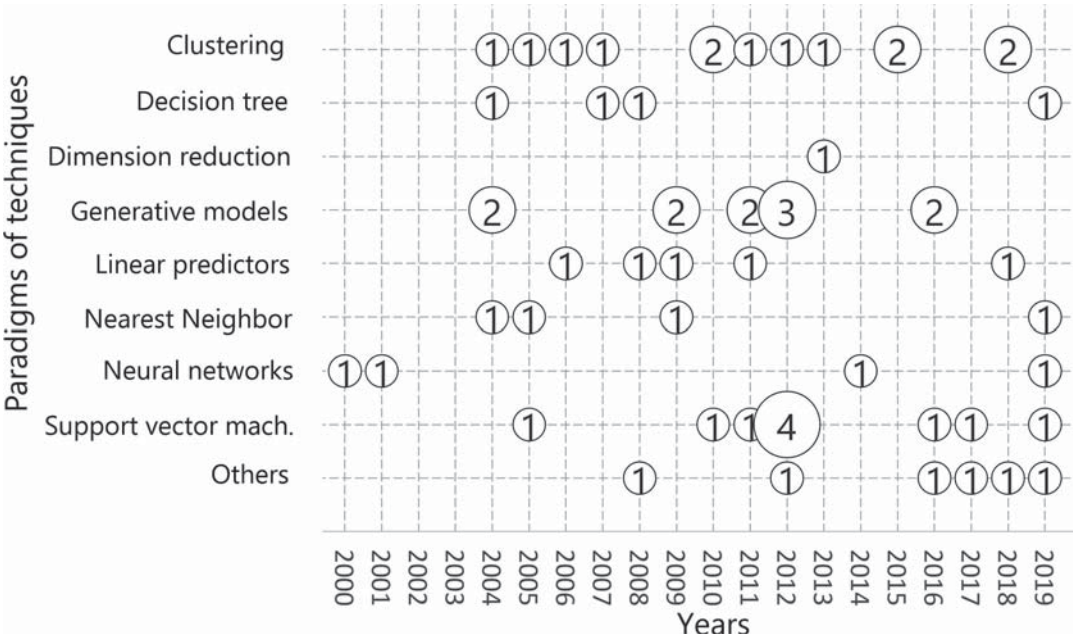


Fig. 3. Use of data mining and machine learning techniques per year.

Finally, thirty-three studies used data sources from the same study, where the authors extracted the news articles directly from the web.

In terms of the size of the datasets, 26 studies used 5 thousand news articles or less, 13 used between 5 and 20 thousand, 7 used sets of more than 20 thousand but less than 35 thousand, and only 5 studies used more than 35 thousand news articles. Five studies did not provide this information.

Figure 4 shows the relationship between paradigms and datasets.

The vertical axis displays the paradigms in which data mining and machine learning techniques were classified, while the horizontal axis corresponds to the reported dataset.

As observed, the neural networks and dimensionality reduction related techniques were reported to be used only in studies that used a third party dataset. All the remaining techniques were reported in both types of sources: from a third party or directly from web pages. On the other side, the clustering techniques were more related to the extraction directly from web pages than with the use of third party datasets.

Based on the third party datasets, it can be observed that *20 Newsgroup* is used primarily in studies that used support vector machines and generative models techniques, while *Reuters-21578* was used in studies related to nearest neighbor and clustering paradigms.

Preprocessing Techniques. Of the 51 studies analyzed, 41 reported the preprocessing techniques applied to the data before implementing the data mining or machine learning technique. The following preprocessing techniques were found:

Table 2. Data sets used in primary studies.

Dataset	Qty	Studies	Description	Size	Classifiers
20 Newsgroup	14	[S03, S07, S11, S15, S18, S21, S22, S23, S26, S29, S31, S33, S34, S43]	Collection of documents related to news, retrieved by Ken Lang, and grouped by topic (some specific and some general)	18 846 documents	20 classifiers divided in 6 great groups: computers, recreation, sci, misc, talks, alternative, and social
Reuters-21578	7	[S11, S12, S21, S29, S40, S41, S42]	From Carnegie Group, Inc. and Reuters, Ltd. Recollected in 1987. The collection is in SGML format. Attributes for each news include topic classification	21 578 articles	Subjects divided in these mayor groups: exchanges, organizations, people, places, and topics (economic subjects)
Reuters Corpus Volume I (RCV1)	3	[S27, S30, S39]	From Reuters, Ltd., the international greatest text and tv news agency. Collected between August 1996 and August 1997. It is in XML format, and each document is categorized into 3 criteria (topic, industry and region)	806 791 documents in v1, and 804 414 in v2	Different codes organized in these mayor groups: Corporate/Industrial, Economics, Government/Social and Markets
Other data corpus from a third party	3	[S01, S13, S37]	These studies mention News Aggregator Data Set from Artificial Intelligence Lab, Roma Tre University, Italy; TDT2 English Corpus, from Linguistic Data Consortium, University of Pennsylvania, EE.UU.; and English Gigaword, from Linguistic Data Consortium, University of Pennsylvania, EE.UU	ADS: 422 937 pages. TDT2 E: 54 thousand stories. E. Gigaword: 314 archives	NADS: Includes four mayor groups of topics: business, science and technology, entertainment, and health. TDT2 includes 100 different topics not specified
Data corpus extracted from the Internet in the same study	33	[S02, S03, S04, S05, S06, S08, S09, S10, S12, S13, S14, S15, S16, S17, S18, S19, S21, S24, S25, S28, S32, S33, S35, S36, S38, S44, S45, S46, S47, S48, S49, S50, S51]	Seeveral sources such as New York Times, CNN, BBC, SKYNews, ABCNews, FoxNews, Reuters, The Guardian, Yahoo news, among other	Varied	Varied

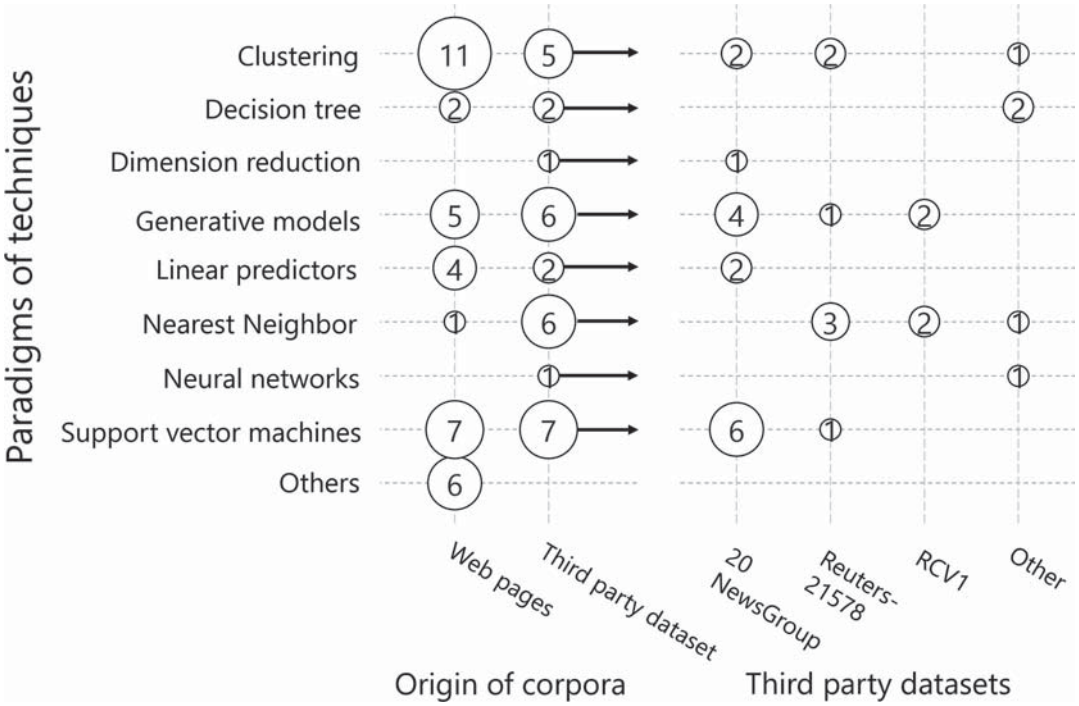


Fig. 4. Relation between paradigms and datasets.

- Stop words removal: identifying and then removing words that were not relevant for the classification, such as articles or pronouns [S02, S03, S04, S05, S06, S08, S09, S10, S11, S12, S14, S15, S16, S18, S19, S22, S23, S24, S25, S27, S28, S30, S32, S36, S38, S40, S42, S43, S45, S48, S49, S50, S51].
- Stemming: detecting the roots of the words and eliminating the rest of each word, in order to standardize their meaning [S01, S02, S05, S06, S07, S08, S12, S14, S15, S16, S23, S26, S27, S28, S30, S35, S36, S37, S38, S40, S43, S45, S48, S49, S50, S51].
- Tokenizing: splits longer strings of text into smaller pieces, or tokens [S01, S02, S03, S07, S09, S11, S12, S15, S16, S18, S28, S36, S38, S45, S48, S49, S50, S51].
- Cleaning: removing unnecessary elements from the data such as comments and labels [S02, S04, S05, S19, S20, S23, S25, S32, S36, S37, S38, S39, S43].
- Characters filtering: for example, removing punctuation signs, or transforming letter cases [S01, S03, S04, S15, S16, S19, S28, S30, S43].
- Size filtering: removing too short or too long words [S01, S15, S18].
- Linguistic filtering: using a linguistic classification of words so that it can be part of the posterior analysis [S10, S13, S16, S27].
- Frequency-oriented filtering: removing frequent words [S04, S10, S11, S35].
- Others [S07, S14, S19, S22, S33, S37, S45, S50].

Performance Metrics. From all studies analyzed, 44 reported performance metrics used for measuring the effectiveness of data mining or machine learning techniques.

Such metrics were classified based on Han et al. [17]. Table 3 shows the performance metrics reported in the primary studies, the number of times each metric was reported (used), and the study reference.

The most frequently used metric was *F-measure*, reported by 25 studies. This is obtained by the harmonic mean of the *precision* and *recall* values.

In second place we found *precision* and *recall*, with 19 studies, and in third place, *Accuracy* with 17 studies.

Ten studies reported measuring *complexity* in terms of execution time or space, 13 studies reported other different metrics counted on the generated clusters and their size, and 2 studies reported as a metric the manual analysis of the results.

Table 3. Performance metrics reported in primary studies.

Metric	Qty	Studies
F-measure	25	[S02, S03, S07, S08, S11, S13, S14, S16, S17, S19, S21, S23, S27, S29, S30, S31, S32, S33, S34, S37, S41, S43, S46, S47, S50]
Recall	19	[S03, S07, S08, S13, S14, S17, S19, S20, S21, S27, S31, S32, S38, S40, S41, S42, S43, S46, S47]
Precision	19	[S03, S07, S08, S13, S14, S17, S19, S20, S21, S27, S31, S32, S38, S40, S41, S42, S43, S46, S47]
Accuracy	17	[S01, S07, S13, S14, S15, S22, S26, S27, S28, S36, S39, S41, S45, S47, S48, S50, S51]
Complexity	10	[S01, S02, S12, S25, S29, S30, S37, S39, S41, S45]
Others	13	[S02, S04, S05, S06, S08, S10, S11, S26, S28, S29, S33, S41, S45]

Figure 5 shows the relationship between paradigms and metrics. The vertical axis displays the paradigms in which data mining and machine learning techniques were classified, while the horizontal axis corresponds to the reported performance metrics.

For the *clustering* related techniques, all the metrics were reported to be used. On the other side, the *F-measure* metric is not only the most used in all the studies, but also has been used to measure the performance in all the paradigms. The *precision* and *recall* metrics have been used to measure effectiveness of almost all the paradigms of techniques, except the *decision trees* paradigm. In the case of the *complexity* measure, it was not used with the *support vector machines*, *linear predictors*, nor *dimensionality reduction* techniques paradigm.

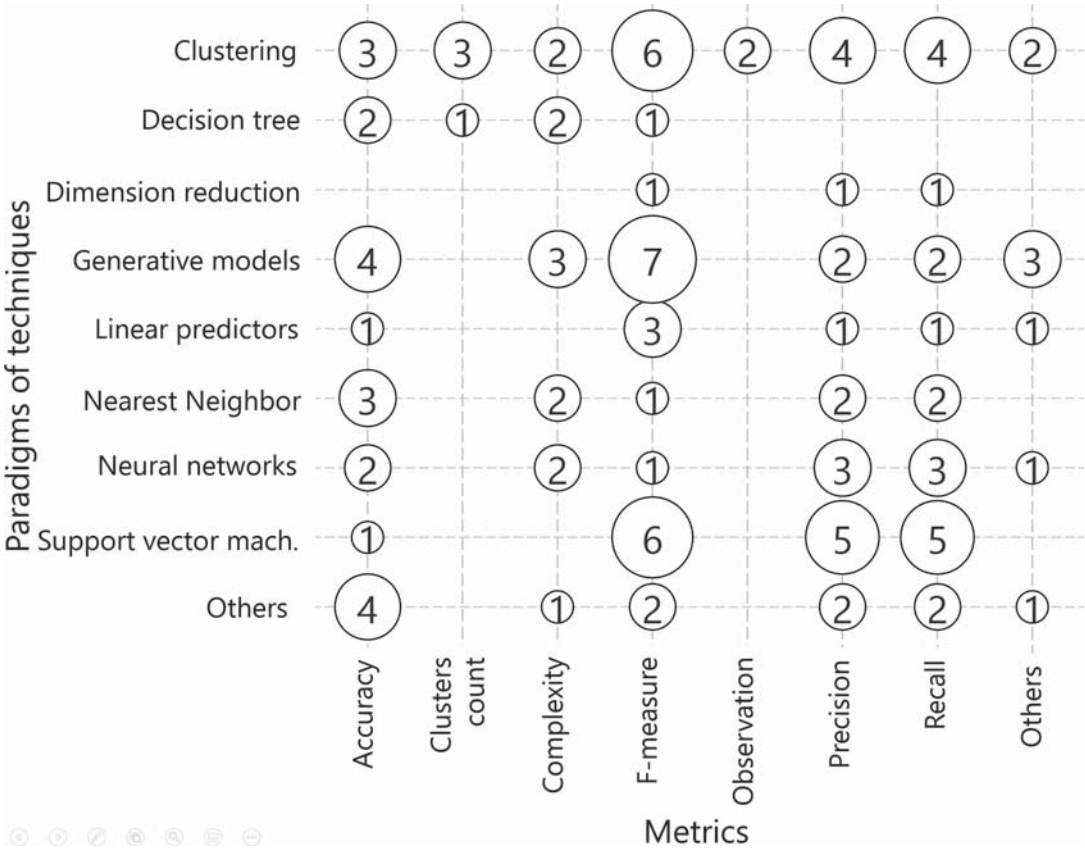


Fig. 5. Relation between reported metrics and data sources

6 Conclusions

In this paper, we presented a systematic literature review of 51 studies, with the goal of characterizing data mining and machine learning techniques applied to classification of web news, together with the datasets used, and their evaluation metrics.

Data mining and machine learning techniques can help in the classification of news, considering their large volume and complexity.

The techniques used in the analyzed primary studies aim to generate topic groupings from the news, removing irrelevant information and trying to establish similarities among the texts of articles.

We found that the most used paradigm of techniques were clustering, support vector machines, and generative models. We also found that the most reported metric for measuring effectiveness was the F-measure. Regarding the datasets, most studies used online data extracted from news web pages.

Primary studies analyzed in this study did not report one single way of classifying the news, but rather a variety of techniques, hence we cannot say that there is a *best* technique for topic classification in the news context.

A possible future work could be to make a comparison based on the reported most used techniques presented here, in order to study them in a deeper analysis.

Also, to reproduce experiments based on the reported techniques, and compare their performance.

A finally recommendation for researchers is that the studies should have more information about what they did so that the studies can be easily be reproducible and increase the reliability of the empirically generated evidence. Also it is recommended to be more specific on documenting the cases, justifying the selection of techniques, metrics, or data sources.

References

1. Fisher, D., Hoff, A., Robertson, G., Hurst, M.: Narratives: a visualization to track narrative events as they develop. In: 2008 IEEE Symposium on Visual Analytics Science and Technology, pp. 115–122, October 2008
2. Arce, J.: Medios de comunicación de masas en costa rica: entre la digitalización, la convergencia y el auge de los "new media", PROSIC, Informe del Programa de la Sociedad de la Información el Conocimiento, pp. 283–307. Universidad de Costa Rica, San José (2012)
3. Iglesias, J., Tiemblo, A., Ledezma, A., Sanchis, A.: Web news mining in an evolving framework. Inf. Fusion **28**, 90–98 (2016)
4. Mittermayer, M., Knolmayer, G.: A survey. Institut für Wirtschaftsinformatik der Universität Bern, Text mining systems for market response to news (2006)
5. Berendt, B.: Text mining for news and blogs analysis. In: Encyclopedia of Machine Learning, pp. 968–972 (2017)
6. Mladenić, M., Brank, J., Grobelnik, M.: Document Classification. Encyclopedia of Machine Learning, pp. 968–972 (2017)
7. Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B., Liu, X.: Learning approaches for detecting and tracking news events. Intell. Syst. Appl. 14, 32–43 (1999)
8. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers, Burlington (2005)
9. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: from theory to algorithms. In: Volume 9781107057135 of Understanding Machine Learning: From Theory to Algorithms, pp. 1–397 (2013). Cited by: 459
10. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**, 1–47 (2002)
11. Irfan, R., King, C.K., Grages, D., Ewen, S., Khan, S.U., Madani, S.A., Kolodziej, J., Wang, L., Chen, D., Rayes, A., Tziritas, N., Xu, C., Zomaya, A.Y., Alzahrani, A.S., Li, H.: A survey on text mining in social networks. Knowl. Eng. Rev. **30**(2), 157–170 (2015). Cited by: 39
12. Bharti, D., Babu, K.: Automatic Keyword Extraction for Text Summarization: A Survey, April 2017. http://arxiv.org/abs/1704.03242
13. Castillo, E., Cervantes, O., Vilariño, D.: Text analysis using different graph based representations. Computacion y Sistemas **21**(4), 581–599 (2017). Cited by: 1
14. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: an update. Inf. Softw. Technol. **64**, 1–18 (2015)
15. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. Engineering **45**(d), 1051 (2007)

16. Basili, V., Gianluigi, C., Rombach, D.: The goal question metric approach. In: Encyclopedia of Software Engineering, pp. 528–532 (1994)
17. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techiques. Morgan Kaufmann Publishers, Burlington (2012)
18. Maghdid, H.: Web news mining using new features: a comparative study. IEEE Access **7**, 5626–5641 (2019)
19. Bouras, C., Tsogkas, V.: Assigning web news to clusters, pp. 1–6 (2010)
20. Dadgar, S.M.H., Araghi, M.S., Farahani, M.M.: A novel text mining approach based on TF-IDF and support vector machine for news classification. In: 2016 IEEE International Conference on Engineering and Technology (ICETECH), pp. 112–116, March 2016

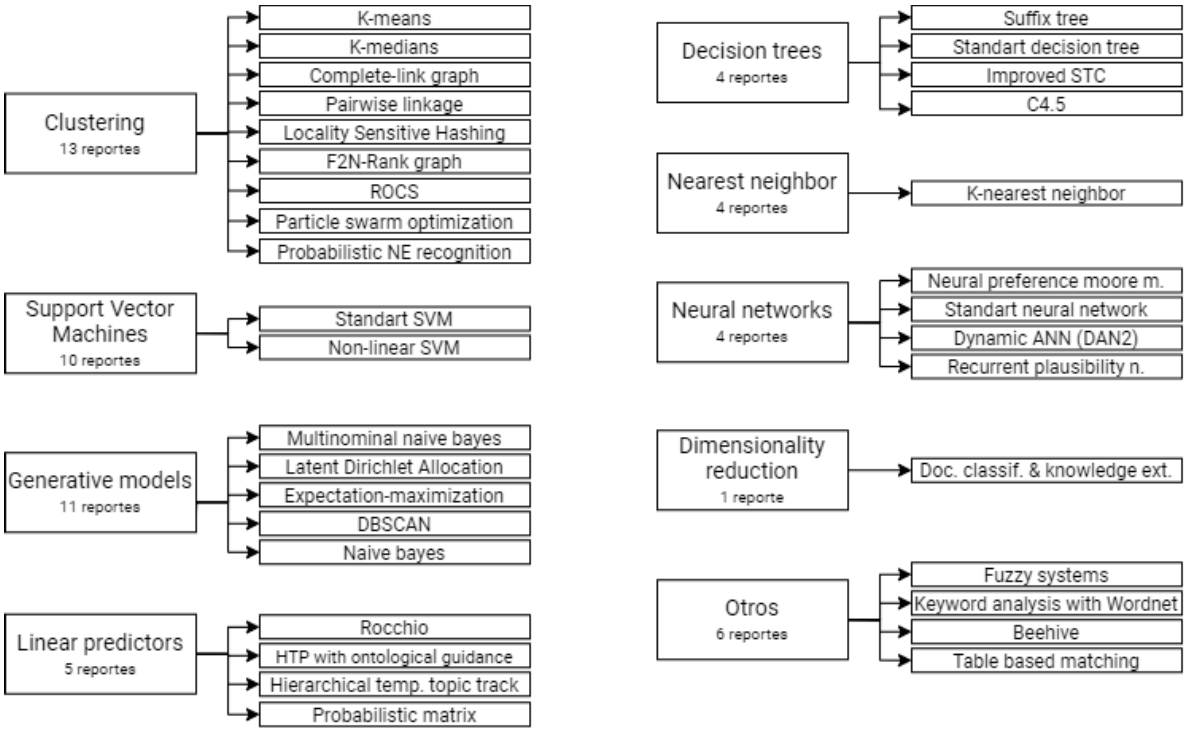


Figura 4.13: Técnicas de minería de datos y aprendizaje automático por paradigma.

4.D. Agrupamiento de técnicas

En la Figura 4.13 se detallan las técnicas específicas de minería de datos y aprendizaje automático que fueron reportadas para cada paradigma estudiado.

Bibliografía del capítulo

[1] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[2] D. Fisher, A. Hoff, G. Robertson, and M. Hurst, "Narratives: A visualization to track narrative events as they develop," in *2008 IEEE Symposium on Visual Analytics Science and Technology*, pp. 115–122, Oct 2008.

[3] J. Arce, "Medios de comunicación de masas en costa rica: entre la digitalización, la convergencia y el auge de los "new media".," *PROSIC, Informe del Programa de la Sociedad de la Información el Conocimiento*, pp. 283–307, 2012. San José: Universidad de Costa Rica.

[4] J. Iglesias, A. Tiemblo, A. Ledezma, and A. Sanchis, "Web news mining in an evolving framework," *Information Fusion*, vol. 28, pp. 90–98, 2016.

[5] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. Archibald, and X. Liu, "Learning approaches for detecting and tracking news events," *Intelligent Systems and their Applications, IEEE*, vol. 14, pp. 32 – 43, 08 1999.

[6] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2005.

[7] B. Berendt, *Text Mining for News and Blogs Analysis*, pp. 968–972. Springer US, 2010.

[8] H. Jiawei, K. Micheline, and J. P, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.

[9] M. Mittermayer and G. Knolmayer, “Text mining systems for market response to news: A survey,” 09 2006.

[10] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, vol. 9781107057135 of *Understanding Machine Learning: From Theory to Algorithms*, pp. 1–397. 2013. Cited By :459.

[11] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Comput. Surv.*, vol. 34, pp. 1–47, Mar. 2002.

[12] R. Irfan, C. King, D. Grages, S. Ewen, S. Khan, S. Madani, J. Kolodziej, L. Wang, D. Chen, A. Rayes, N. Tziritas, C. Xu, A. Zomaya, A. Alzahrani, and H. Li, “A survey on text mining in social networks,” *Knowledge Engineering Review*, vol. 30, no. 2, pp. 157–170, 2015. Cited By :39.

[13] D. Bharti and K. Babu, “Automatic keyword extraction for text summarization: A survey,” *http://arxiv.org/abs/1704.03242*, 04 2017.

[14] E. Castillo, O. Cervantes, and D. Vilariño, “Text analysis using different graph-based representations,” *Computacion y Sistemas*, vol. 21, no. 4, pp. 581–599, 2017. Cited By :1.

[15] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3,” *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.

[16] V. Basili, G. Caldiera, and D. Rombach, “The goal question metric approach,” *Encyclopedia of software engineering*, pp. 528–532, 1994.

[17] C. Bouras and V. Tsogkas, “Assigning web news to clusters,” pp. 1–6, 2010.

[18] H. Maghdid, “Web news mining using new features: A comparative study,” *IEEE Access*, vol. 7, pp. 5626–5641, 2019.

[19] S. Dadgar, M. Araghi, and M. Farahani, “A novel text mining approach based on tf-idf and support vector machine for news classification,” in *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pp. 112–116, March 2016.

[20] B. Kitchenham, E. Mendes, and G. Travassos, *A Systematic Review of Cross- vs. Within-Company Cost Estimation Studies*. IEEE Trans on SE, 2007.

[21] K. Nørvåg and R. Øyri, "News item extraction for text mining in web newspapers," vol. 2005, pp. 195–204, 2005.

[22] M. Mori, T. Miura, and I. Shioya, "Topic detection and tracking for news web pages," pp. 338–342, 2007.

[23] N. Shah and E. ElBahesh, "Topic-based clustering of news articles," pp. 412–413, 2004.

[24] A. Abdulsahib and S. Kamaruddin, "Graph based text representation for document clustering," *Journal of Theoretical and Applied Information Technology*, vol. 76, no. 1, pp. 1–13, 2015. Cited By :4.

[25] S. Vadrevu, C. Hui Teo, S. Rajan, K. Punera, B. Dom, A. Smola, Y. Chang, and Z. Zheng, "Scalable clustering of news search results," pp. 675–683, 2011.

[26] S. Chakraborti and S. Dey, "Product news summarization for competitor intelligence using topic identification and artificial bee colony optimization," in *Proceeding of the 2015 Research in Adaptive and Convergent Systems, RACS 2015*, pp. 1–6, 2015. Cited By :3.

[27] Z. Xu, W. Liu, Y. Zhu, and S. Zhang, *Building domain keywords using cognitive based sentences framework*, vol. 422 of *Lecture Notes in Electrical Engineering*. 2018.

[28] G. Dubus, M. Bruyen, and N. Bennacer, "Supporting information retrieval in rss feeds," in *WEBIST 2010 - Proceedings of the 6th International Conference on Web Information Systems and Technology*, vol. 1, pp. 307–312, 2010.

[29] S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," *Open Computer Science*, vol. 3, no. 2, pp. 69–90, 2013. Cited By :29.

[30] S. Pasarate and R. Shedge, "Concept based document clustering using k prototype algorithm," in *2018 International Conference on Control, Power, Communication and Computing Technologies, ICCPCCT 2018*, pp. 579–583, 2018.

[31] M. Haq, H. Ahmed, and A. Qamar, "Dynamic entity and relationship extraction from news articles," in *2012 International Conference on Emerging Technologies*, pp. 1–5, Oct 2012.

[32] M. Naughton, N. Kushmerick, and J. Carthy, "Event extraction from heterogeneous news sources," in *AAAI Workshop - Technical Report*, vol. WS-06-07, pp. 1–6, 2006. Cited By :16.

[33] M. Özgür and B. Diri, "Content mining of microblogs," pp. 835–838, 2012.

[34] J. Zeng, J. Duan, W. Cao, and C. Wu, "Topics modeling based on selective zipf distribution," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6541–6546, 2012. Cited By :19.

[35] Y. Cheng, K. Zhang, Y. Xie, A. Agrawal, W.-K. Liao, and A. Choudhary, "Learning to group web text incorporating prior information," pp. 212–219, 2011.

[36] X. Liu and B. Huet, "Event-based cross media question answering," *Multimedia Tools and Applications*, vol. 75, pp. 1495–1508, FEB 2016.

[37] I. Katakis, G. Tsoumakas, E. Banos, N. Bassiliades, and I. Vlahavas, "An adaptive personalized news dissemination system," *Journal of Intelligent Information Systems*, vol. 32, no. 2, pp. 191–212, 2009. Cited By :47.

[38] R. Calvo, J. Lee, and X. Li, "Managing content with automatic document classification," *Journal of Digital Information*, vol. 5, no. 2, 2004. Cited By :24.

[39] C. Silla, G. Pappa, A. Freitas, and C. Kaestner, "Automatic text summarization with genetic algorithm-based attribute selection," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 3315, pp. 305–314, 2004. Cited By :18.

[40] H. Guo, L. Zhou, and L. Feng, "Self-switching classification framework for titled documents," *Journal of Computer Science and Technology*, vol. 24, no. 4, pp. 615–625, 2009. Cited By :3.

[41] A. Puurula, *Scalable text classification with sparse generative modeling*, vol. 7458 LNAI of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2012. Cited By :4.

[42] Z. Li, W. Shang, and M. Yan, "News text classification model based on topic model," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1–5, June 2016.

[43] S. Hassan, M. Rafi, and M. S. Shaikh, "Comparing svm and naïve bayes classifiers for text categorization with wikitology as knowledge enrichment," in *2011 IEEE 14th International Multitopic Conference*, pp. 31–34, Dec 2011.

[44] A. Sajjanhar and Y. Zhao, "Web service to deliver filtered rss items to a mobile application," pp. 128–133, 2012.

[45] Y. Kim and S. Lee, "Svm-based web content mining with leaf classification unit from dom-tree," pp. 359–364, 2017.

[46] S. Saha, A. Sajjanhar, S. Gao, R. Dew, and Y. Zhao, "Delivering categorized news items using rss feeds and web services," pp. 698–702, 2010.

[47] S. Samarawickrama and L. Jayaratne, "Effect of named entities in web page classification," pp. 38–42, 2012.

[48] A. Pangos, E. Iosif, A. Potamianos, and E. Fosler-Lussier, "Combining statistical similarity measures for automatic induction of semantic classes," in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pp. 278–283, Nov 2005.

[49] A. Hoonlor, B. Szymanski, M. J., and V. Chaoji, "Document Clustering with Bursty Information," *Computing and Informatics*, vol. 31, no. 6, SI, pp. 1533–1555, 2012.

[50] S. Sharma and N. Sharma, "Text classification using ensemble of non-linear support vector machines," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 3169–3174, 2019.

[51] V. Ha-Thuc and J. Renders, "Large-scale hierarchical text classification without labelled data," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining, WSDM 2011*, pp. 685–694, 2011. Cited By :13.

[52] L. Antonellis, C. Bouras, and V. Poulopoulos, "Scalable text classification as a tool for personalization," *Computer Systems Science and Engineering*, vol. 24, no. 6, pp. 399–408, 2009.

[53] I. Antonellis, C. Bouras, V. Poulopoulos, and A. Zouzias, "Scalability of text classification," in *WEBIST 2006 - 2nd International Conference on Web Information Systems and Technologies, Proceedings*, vol. IT, pp. 408–413, 2006. Cited By :1.

[54] M. Viermetz, M. Skubacz, C. Ziegler, and D. Seipel, "Tracking topic evolution in news environments," in *2008 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, pp. 215–220, July 2008.

[55] M. Ali, S. Khalid, M. I. Rana, and F. Azhar, "A probabilistic framework for short text classification," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 742–747, Jan 2018.

[56] L. Qiu, Bin-Pang, and L. Zhao, "An event detection algorithm based on improved stc," in *2008 IEEE International Conference on Networking, Sensing and Control*, pp. 528–532, April 2008.

[57] S. Chung and D. McLeod, "Dynamic pattern mining: An incremental data clustering approach," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3360 LNCS, pp. 85–112, 2005.

[58] O. Dan and H. Mocian, "Scalable web mining with newistic," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5476 LNAI, pp. 556–563, 2009.

[59] S. Wermter and G. Arevian, "Modular preference moore machines in news mining agents," vol. 3, pp. 1792–1797, 2001.

[60] M. Ghiassi and M. Olschimke, *Enhancing document search with a dynamic artificial neural network*, pp. 1–33. Advances in Machine Learning Research, 2014.

[61] S. Wermter, "Neural network agents for learning semantic text classification," *Information Retrieval*, vol. 3, no. 2, pp. 87–103, 2000. Cited By :43.

[62] N. Sahu, K. Mohbey, and G. Thakur, "Document clustering using message passing between data points," in *Proceedings - 2013 International Conference on Communication Systems and Network Technologies, CSNT 2013*, pp. 591–596, 2013.

[63] P. Navrat and S. Sabo, "What is going on out there right now? a beehive based machine to give snapshot of the ongoing stories on the web," in *2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pp. 168–174, Nov 2012.

[64] C. Za'in, M. Pratama, E. Lughofer, and S. Anavatti, "Evolving type-2 web news mining," *Applied Soft Computing Journal*, vol. 54, pp. 200–220, 2017.

[65] A. Patel and Y. Sharma, "Web page classification on news feeds using hybrid technique for extraction," *Smart Innovation, Systems and Technologies*, vol. 107, pp. 399–405, 2019.

[66] D. Nithya and S. Sivakumari, "Categorizing online news articles using penguin search optimization algorithm," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 4, pp. 2565–2568, 2018.

[67] J. Taeho and Y. Gwyduk, "List based matching algorithm for classifying news articles in newspage.com," in *2008 IEEE International Conference on System of Systems Engineering*, pp. 1–5, June 2008.

Conclusiones de la memoria

En esta memoria se analizó un conjunto de temas avanzados en Ingeniería de *Software* para la carrera de Bachillerato en Computación de la Escuela de Computación e Informática, mediante la aplicación de metodologías empíricas. Con estas investigaciones, se generó conocimiento en relación con cuatro áreas avanzadas de la Ingeniería del *Software* y con los resultados presentados en este documento se ofrecen materiales actualizados que pueden ser utilizados por la carrera de Bachillerato en Computación con Énfasis en Ingeniería de *Software* de la Escuela de Computación e Informática de la Universidad de Costa Rica. Las principales temáticas corresponden a herramientas de evaluación de accesibilidad, herramientas de pruebas de seguridad Web y técnicas de minería de datos y de aprendizaje automático.

Los temas investigados funcionan como insumo para los cursos del énfasis en Ingeniería de *Software*. Los tópicos relacionados con herramientas para evaluar accesibilidad, actúan como material para los cursos sobre interacción humano-computador. Estos, junto con las premisas de pruebas automatizadas también aporta a los cursos con contenido sobre desarrollo y calidad de aplicaciones Web y la seguridad relacionada con estas. En el caso de los temas orientados a herramientas de minería de datos y aprendizaje automático, serían de gran utilidad para los cursos que tratan bases de datos avanzados y de calidad del *software*. En general, los cuatro temas analizados son aplicables en toda la malla curricular establecida para la carrera, así como en los cursos específicos del énfasis.

Para el desarrollo de los estudios secundarios que se presentaron en esta memoria, se siguieron metodologías de investigación rigurosas que aseguraron la calidad y validez necesaria. Por ello, fue parte fundamental aplicar cada etapa de investigación desde un acercamiento sistemático. La estructuración y orden de las etapas asegu-

raron que cada capítulo tenga la posibilidad de ser replicado en el futuro por otros investigadores y así conseguir resultados que puedan ser comparados y validados.

A continuación se detallan las conclusiones para cada una de las investigaciones empíricas que integran esta memoria:

En "Herramientas para la evaluación de la accesibilidad Web: un mapeo sistemático de literatura", como su nombre lo indica, se realizó un mapeo sistemático de literatura que abarca el periodo comprendido entre el 2004 y el 2019, donde se analizaron 50 estudios, con el objetivo de identificar las herramientas, criterios y desafíos en dicho contexto. Este mapeo logró identificar 38 herramientas de las cuales tres fueron las más más reportadas como utilizadas, un conjunto de criterios que se reportan como más incumplidos, cinco desafíos técnicos y cuatro desafíos sobre regulaciones de Gobierno, ambos sobre accesibilidad Web. Con respecto a las herramientas reportadas para evaluar la accesibilidad Web, no se obtuvo un reporte claro de cuál fue el método de selección de la herramienta, ya que la mayoría de los autores solo hicieron mención de la o las herramientas que iban a usar y no especificaron el porqué de su uso o los criterios que evaluaba. En cuanto a los desafíos, solo algunos autores los reportan, otros simplemente mencionan que los hay, pero sin entrar en detalles. Los resultados muestran la necesidad de más estudios que aborden de forma detallada cuáles son las herramientas más recomendadas para evaluar accesibilidad Web, así como listar los criterios que pueden ser evaluados de forma automática por las herramientas. Como trabajo futuro se plantea realizar una investigación a fin de determinar cuáles son las capacidades de las herramientas para evaluar la accesibilidad Web y cuáles son las más recomendadas, así como, analizar qué se puede hacer automáticamente y qué solo por humanos, y la mejor forma de combinar estrategias (herramientas y humanos) para mejorar la efectividad de las evaluaciones que se realizan, ya que con la lectura de los estudios analizados esto no se reporta.

"Herramientas para pruebas automatizadas de seguridad Web: un mapeo sistemático de literatura" realizó un mapeo sistemático sobre las herramientas que evalúan la seguridad de las aplicaciones Web, con el propósito de identificar las herramientas que han sido utilizadas para detectar vulnerabilidades por medio de pruebas automatizadas. En este mapeo se realizó un análisis de 63 estudios publicados entre los años del período 2006-2019. Se logró identificar 66 herramientas que evalúan la tenden-

cia de vulnerabilidades de las aplicaciones Web. Los resultados revelan que existe una gran variedad de herramientas que evalúan la mayor cantidad de las tendencias de vulnerabilidades definidas por OWASP. Las herramientas identificadas cubren al menos un caso de pruebas para evaluar la seguridad de las aplicaciones Web. Además, aunque se identificó gran variedad de herramientas que realizan distintos tipos de pruebas de seguridad, solo pocas se encuentran en la lista de herramientas recomendadas por OWASP, lo que denota la necesidad de contar con evaluaciones empíricas de herramientas existentes en el área. Como trabajo futuro, se plantea seleccionar un conjunto de herramientas de pruebas de seguridad que facilite verificar las principales vulnerabilidades reportadas para las aplicaciones Web y evaluar su efectividad con el fin de generar evidencia para la industria. Asimismo, como otra alternativa de trabajo futuro se propone identificar las clasificaciones de vulnerabilidades brindadas por OWASP que no fueron cubiertas por ninguna herramienta del mapeo. Luego, realizar un estudio de las razones por las cuales las herramientas actuales no evalúan estas vulnerabilidades.

En "Técnicas de minería de datos y aprendizaje automático para segmentación de clientes bancarios: un mapeo sistemático de literatura" se realizó un análisis de 87 estudios primarios que fueron publicados durante el periodo 2005-2019, con el fin de caracterizar la literatura según las técnicas, las herramientas, los conjuntos de datos y las métricas de evaluación. Se logró identificar entre los principales hallazgos 55 técnicas de minería de datos y aprendizaje automático que fueron clasificadas dentro de nueve paradigmas. Los paradigmas más reportados fueron *decision tree* y *linear predictors* con 88 y 54 referencias respectivamente. De igual manera, con respecto a las herramientas se identificaron 22 en total. Estas herramientas mantuvieron tendencias de reportes a través de los años. El 55 % de las herramientas eran de licencia gratuita. Con respecto a la capacidad de implementación de estas herramientas, se identificó que el 50 % de ellas ya contienen las técnicas implementadas y solamente se deben configurar, mientras que para el restante 50 % se necesita de un esfuerzo adicional para implementar el modelo. En el caso de los conjuntos de datos se identificaron 31, de los cuales se pudo obtener para la mayoría su cantidad de registros, el enlace para consulta y sus atributos más determinantes, entre ellos, los más utilizados fueron: la edad, el trabajo, el género, la temporalidad y las características crediticias del cliente. En el caso de las métricas de evaluación se detectó que la mayoría de es-

tudios reportan las métricas estándar. La métrica *accuracy* fue la más reportada, pues se presentó en el 78 % de los estudios analizados. Como trabajo futuro se plantea hacer una comparativa entre las técnicas de minería de datos y aprendizaje automático que fueron más reportadas, teniendo en cuenta que la evaluación se debe hacer bajo las mismas condiciones: utilizar el mismo conjunto de datos, aplicar el mismo preprocesamiento de datos, elegir una herramienta adecuada para el caso de estudio y aplicar las métricas de evaluación de rendimiento bajo los mismos parámetros. El propósito de realizar esta comparativa sería identificar cuáles técnicas son mejores que otras contando con un ambiente controlado. Asimismo, esta investigación puede ser ampliada determinando un contexto más general, donde permita identificar diferentes clasificaciones de las técnicas de minería de datos y aprendizaje automático y así relacionar qué tanto cambian los resultados según la clasificación que se escoja.

En "Técnicas de aprendizaje automático y minería de datos para la clasificación de noticias Web: un mapeo de literatura" se hizo un mapeo sistemático que comprendió 51 estudios primarios publicados entre 2000 y 2019. Esta investigación tuvo como objetivo caracterizar las técnicas utilizadas en este contexto, los conjuntos de datos que los estudios emplearon y las métricas que usaron para evaluarlas. De todos los estudios, se realizaron un total de 64 reportes de técnicas. Las técnicas de los paradigmas *clustering*, *support vector machines* y *generative models* fueron las más frecuentes. Las de *linear predictors* fueron disminuyendo sus apariciones con el tiempo, mientras que las técnicas de *clustering* se han mantenido desde su primera aparición en 2004. De los estudios primarios analizados, el 65 % de ellos extrajeron las noticias que utilizaron a partir de las que estaban disponibles en páginas de internet de contenidos noticiosos. El 47 % utilizaron un cuerpo de datos existente que fue descargado. Dentro de los cuerpos de datos utilizados hubo varias referencias principalmente a 20NewsGroup y Reuters. La métrica *F-measure* no solo fue más aplicada para evaluar, sino que también reportó usos para medir todos los paradigmas de técnicas encontrados. Existieron reportes sobre relaciones entre todas las métricas estudiadas con todos los orígenes de los datos presentados. Aunque los estudios no mostraron un consenso sobre técnicas, cuerpos de datos o métricas muy dominantes en los estudios, la especificación de cada uno de esos aspectos puede ser aplicada en la academia, la industria y la investigación. Además, este estudio plantea como trabajo futuro ser ampliado para analizar clasificación temática en contextos más amplios y más generales, ha-

ciendo una investigación más extensa que permita plantear las mejores posibilidades reportadas, y explorar cómo estas serían utilizadas en la clasificación automática de noticias. Como otra posible investigación, se plantea hallar una forma de estandarizar los resultados de las métricas de efectividad de cada estudio, para poder realizar comparaciones en relación con sus rendimientos.

El esfuerzo académico reflejado en esta memoria permite realizar aportes desde tres áreas principales. En el ámbito profesional, los mapeos sistemáticos que identificaron técnicas y herramientas posibilitaron ampliar criterios técnicos para considerar su aplicabilidad en la industria. En el área de la investigación, el trabajo evidenció la utilidad de la aplicación de mapeos sistemáticos sobre metodologías empíricas, para considerar los estudios en la literatura relacionados con un tema específico, con el fin de recopilar sus resultados para análisis posterior. Además, evidenció la importancia de mejorar los detalles de los reportes sobre los estudios que se realizan. Finalmente, desde el campo académico, la información brindada sirve de insumo para identificar áreas de interés que sirvan para reforzar la formación que se da en la carrera de Ciencias de la Computación e Informática.

Las investigaciones realizadas sirven como insumo para ser considerados dentro de los cursos del plan de estudios (Énfasis de Ingeniería de Software). La audiencia meta está dirigida principalmente a profesores o evaluadores de los planes de estudios. Los temas sobre herramientas para evaluar accesibilidad aportarían a los cursos sobre interacción humano-computador. Este, junto con el de pruebas automatizadas, aportarían a los cursos con contenido sobre desarrollo y calidad de aplicaciones Web, la seguridad relacionada con estas, pruebas de software e ingeniería de software. Por otro lado, en relación con los temas de herramientas de minería de datos y aprendizaje automático, se daría un aporte sobre los cursos de bases de datos avanzados, de calidad del software y de inteligencia artificial. Los cuatro temas desarrollados, finalmente, son aplicables en toda la malla curricular establecida para la carrera, así como en los cursos específicos del énfasis.

Bibliografía

[1] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[2] B. Kitchenham and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*. IEEE Trans on SE, 2007.

[3] V. Basili, G. Caldiera, and D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

[4] B. Kitchenham, E. Mendes, and G. Travassos, *A Systematic Review of Cross- vs. Within-Company Cost Estimation Studies*. IEEE Trans on SE, 2007.

[5] P. Bourque and R. Fairley, *SWEBOK: Guide to the Software Engineering Body of Knowledge*. Los Alamitos, CA: IEEE Computer Society, version 3.0 ed., 2014.

[6] R. Feldt and T. Zimmermann, *Empirical Software Engineering*. Springer, 2018.