

Análisis de las Técnicas de Aprendizaje Automático aplicadas en la Ingeniería de Requerimientos de Software: Un Mapeo Sistemático de la Literatura

María Guadalupe Gramajo, Luciana Ballejos, Mariel Ale

CIDISI, Centro de Investigación y Desarrollo de Ingeniería en Sistemas de Información
CONICET-UTN Santa Fe, Argentina
{mgramajo, lballejo, male}@frsf.utn.edu.ar

Resumen. En este trabajo se presenta un mapeo sistemático de la literatura, a los fines de proporcionar una revisión comprensiva de las investigaciones que detallan la aplicación de las técnicas de Aprendizaje Automático (AA) en el área de Ingeniería de Requerimientos (IR) para enriquecer técnicas y procedimientos tradicionales. Los resultados obtenidos destacan propuestas alentadoras orientadas principalmente, a la predicción de fallas y clasificación de los requisitos de software. Sin embargo, se detectó que los estudios que utilizan técnicas de AA en sus propuestas, aún precisan de una mayor validación empírica para ser ampliamente aceptadas y aplicadas en entornos de desarrollo de software. Consecuentemente, esta situación denota la necesidad de continuar investigando y profundizando sobre posibles aplicaciones del AA en el área de IR.

1 Introducción

En la actualidad, el software resulta un activo esencial en todas las organizaciones. Las restricciones y expectativas de los stakeholders hacen de su desarrollo un proceso poco trivial, que consume tiempo y esfuerzo. Esta situación, conduce a los ingenieros y gerentes de proyecto a centrar su trabajo en la resolución de los diversos problemas que surgen durante el proceso de desarrollo del software. Es por ello que, en los últimos años, ha surgido un creciente interés por las técnicas que emergen de las áreas de la Inteligencia Artificial, y particularmente en el Aprendizaje Automático (AA), a los fines de, explorar su utilidad en aplicaciones de software, y enriquecer técnicas y procedimientos tradicionales que contribuyan a solucionar los desafíos presentes en la Ingeniería del Software [1]. Los algoritmos de AA han demostrado ser de gran valor práctico en una variedad de dominios de aplicación, la Ingeniería de Requerimientos (IR) es uno de ellos. Múltiples técnicas de AA, tales como la lógica difusa, redes neuronales artificiales, clasificación, sistemas basados en conocimiento, computación evolutiva, sistemas expertos, entre otras, son actualmente utilizadas para abordar una amplia gama de problemas y desafíos [2, 3]. Consecuentemente, para responder al reto de desarrollar y mantener sistemas de software de gran tamaño y complejidad en entornos dinámicos, los métodos de AA continúan demostrando avances significativos en las actividades relacionadas a la trazabilidad y clasificación de requisitos [4, 5], predicción de cambios en el código [6] y pruebas de software [7].

Esta situación, ha motivado la elaboración de un mapeo sistemático de la literatura (MSL), siguiendo las pautas reportadas por Petersen y otros [8] y Kitchenham y Charters [9]. El principal objetivo de este trabajo, es identificar y analizar los avances en el área de IR, respecto a la aplicación de técnicas de Aprendizaje Automático Supervisado (AAS), en la resolución de los desafíos que surgen en cada una de sus fases. Un MSL se enfoca principalmente, en probar la existencia de estudios en un campo de interés particular, proporcionando una visión general del área de investigación. Además, permite identificar áreas temáticas que aún precisan investigación y determinar las tendencias de estudios futuros [8]. Resultando así, una herramienta útil y de gran valor para los profesionales que desean iniciar sus investigaciones en un campo de estudio determinado.

Este estudio puede considerarse relevante y de gran utilidad para proyectos que se deseen abordar a futuro en el área de IR basados en la aplicación de AA. Las principales contribuciones de este estudio son: i) identificación y análisis de las técnicas de AA que apoyan las fases de la IR; ii) identificación de las fuentes de datos utilizadas por los modelos de AA en los estudios identificados.

El resto del trabajo se encuentra organizado de la siguiente manera. La Sección 2 describe brevemente los principales conceptos abordados en este MSL tales como, Ingeniería de Requerimientos, Aprendizaje Automático y Aprendizaje Automático Supervisado, en virtud de establecer un marco teórico introductorio. La Sección 3 presenta la metodología de investigación aplicada. La Sección 4 presenta el análisis y discusión de los resultados. La Sección 5 describe conclusiones y futuras líneas de investigación.

2 Marco Teórico

En esta sección se describirán brevemente los principales conceptos abordados en el MSL, a modo de establecer un marco teórico introductorio, necesario para abordar la búsqueda de los estudios primarios.

2.1 Aprendizaje Automático

En los últimos años, el interés y popularidad por el AA ha crecido de manera exponencial. Si bien no es un concepto nuevo, la disponibilidad de grandes volúmenes de datos y el aumento de la capacidad de procesamiento en los ordenadores, permitieron experimentar e indagar aún más sobre sus posibles aplicaciones en diversos dominios.

El AA es una disciplina de la Inteligencia Artificial, en esencia, es un conjunto de técnicas que permiten a las máquinas “aprender” de los datos y luego tomar decisiones o realizar predicciones sobre ellos. Samuel [10], describe el concepto de AA como el campo de estudio que le da a los ordenadores la habilidad de aprender algo para lo cual no han sido explícitamente programados. Una definición más formal es la expresada por Mitchell [11], quien sostiene que una máquina aprende una tarea particular T , considerando las experiencias de tipo E , respecto de una medida de performance P , si la

máquina efectivamente mejora su performance P , en la tarea T , a partir de la experiencia E . Considerando las definiciones aportadas por los autores mencionados, este trabajo toma como referencia para su desarrollo el concepto de AA como *disciplina que estudia los métodos y algoritmos computacionales, y, además, permite a los ordenadores simular el aprendizaje humano y adquirir conocimiento del mundo real, con lo que es posible mejorar el rendimiento de algunas tareas en base al nuevo conocimiento adquirido*.

2.2 Aprendizaje Automático Supervisado

El AAS es un subconjunto de técnicas del AA que operan bajo supervisión, su objetivo es construir un modelo que realice predicciones basadas en la evidencia en presencia de incertidumbre. En el AAS, los algoritmos trabajan con datos “etiquetados”, intentado encontrar una función hipótesis que, dadas las variables de entrada, les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un “histórico de datos” o “datos de entrenamiento” y, de esta manera, “aprende” a asignar la etiqueta de salida adecuada a un nuevo valor de entrada [12]. La Figura 1 esquematiza el AAS.

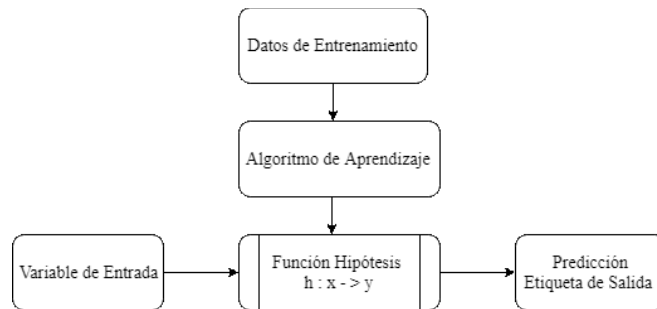


Fig. 1. Esquema genérico de Aprendizaje Supervisado.

2.3 Ingeniería de Requerimientos de Software

La IR proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional [13]. En otras palabras, la IR es una disciplina de la Ingeniería del Software que genera artefactos transversales a todas las actividades involucradas en el ciclo de vida del desarrollo del software, es por ello que resulta relevante su revisión, a los fines de detectar áreas que aún precisan investigación, siendo útiles para futuras contribuciones en el área.

Este trabajo, analiza la aplicación de AAS en la IR, estas técnicas se caracterizan por su gran capacidad de predicción y clasificación. El AAS, permite aprovechar las fuentes de datos disponibles para realizar predicciones de comportamiento, con lo cual es posible desarrollar sistemas de soporte y recomendación en distintos dominios de aplicación, área de fundamental interés para los autores. En las siguientes secciones se describe la metodología utilizada en el desarrollo de este MSL.

3 Metodología

La metodología utilizada para desarrollar este MSL, se basa en las pautas propuestas por Petersen y otros [8] y las consideraciones establecidas por Kitchenham y Charters [9], ambas propuestas son ampliamente aceptadas en la comunidad científica para realizar estudios de mapeo y revisiones de la literatura en el área de Ingeniería del Software. El proceso de investigación consta de cinco pasos principales: (a) definición de preguntas de investigación, (b) búsqueda e identificación de estudios primarios utilizando cadenas de búsqueda en bibliotecas académicas digitales, (c) definición de criterios de inclusión y exclusión, (d) selección y clasificación de estudios primarios considerando los criterios definidos anteriormente, a fin de eliminar aquellos que no son relevantes a la investigación y, por último, (e) extracción y análisis de datos. La Figura 2 sintetiza la metodología aplicada en esta investigación.



Fig. 2. Proceso de Mapeo Sistemático propuesto por Petersen y otros [8].

Las subsecciones posteriores, describen los elementos de entrada y las salidas obtenidas, como resultado del proceso metodológico aplicado.

3.1 Definición de las Preguntas de Investigación

Las preguntas de investigación son esenciales en un MSL, porque sirven de guía al proceso de búsqueda y selección de estudios. La Tabla 1 presenta las preguntas de investigación y el criterio que motiva su formulación, a los fines de delimitar el alcance de la revisión.

Tabla 1. Preguntas y Objetivos de Investigación.

PREGUNTA DE INVESTIGACION	MOTIVACION
Q1. ¿Cuáles son las actividades, tareas y desafíos de la IR respaldadas por las tecnologías de AA?	M1. Identificar propuestas de investigación que emplean técnicas de AA para abordar las actividades, tareas y problemas que surgen en la IR.
Q2. ¿Cuáles son los algoritmos de AA utilizados con mayor frecuencia en las propuestas identificadas?	M2. Detectar los algoritmos de AA utilizados para resolver dificultades en las actividades de la IR.
Q3. ¿Cuáles son las principales fuentes de datos utilizadas en los algoritmos de AA?	M3. Detectar repositorios y tipos de datos utilizados como entrada para la ejecución de los algoritmos de AA.

3.2 Ejecución de la Búsqueda: Estrategia y Profundidad Temporal

Los estudios primarios se identificaron consultando cuatro bibliotecas en línea: (1) IEEE Xplore; (2) ScienceDirect; (3) SpringerLink y (4) Scopus. Estas bibliotecas fueron seleccionadas porque son consideradas fuentes importantes de estudios en el área Informática en general, y de Ingeniería del Software en particular, poseen repositorios de conferencias y revistas relevantes. Asimismo, sus motores de búsqueda admiten la posibilidad realizar búsquedas avanzadas, basadas en diferentes criterios y formatos de cadena. Resulta necesario destacar, además, el acceso institucional gratuito a los trabajos indexados en las mencionadas bibliotecas digitales.

Los términos utilizados para la construcción de la cadena de búsqueda fueron acordados entre los autores, considerando las actividades involucradas en la IR definidas por Pohl [14] y, además, las pautas propuestas por Petersen y otros [8], éstas se mencionan a continuación: (1) Identificar las palabras principales de las preguntas de investigación; (2) Extraer palabras clave en documentos relevantes a la temática; (3) Identificar sinónimos y abreviaturas alternativas; (4) Construir cadenas de búsqueda concatenando las palabras identificadas mediante los operadores booleanos AND para vincular términos principales y OR para incorporar sinónimos alternativos; (5) Construir las cadenas de búsqueda avanzadas para cada repositorio. Finalmente, la cadena de búsqueda genérica resultante se muestra en la Figura 3, la misma se adaptó a los motores de búsqueda de cada repositorio para su ejecución.

("requirements engineering" OR "software requirements") AND ("machine learning" OR "supervised learning")

Fig. 3. Cadena de Búsqueda Genérica.

Para definir la profundidad temporal, este trabajo toma como referencia el año 2002, en el cual se liberó Torch, una librería de AA bajo licencia de código abierto que incluye los algoritmos y modelos más populares, tales como Perceptrón Multicapa, Máquina de Soporte Vectorial (SVM), Modelos Gaussianos Mixtos, Clasificadores Bayesianos, K-Vecinos más cercanos, entre otros [15]. La difusión de este hecho y el creciente interés en indagar más aún sobre la utilidad de las técnicas de AA, impulsó su aplicación en diversos dominios. Consecuentemente, por esta razón, la búsqueda de los estudios primarios se acotó a las últimas décadas (2002- 2017), planteando como desafío descubrir oportunidades de investigación y lagunas en el campo del AA aplicado a la IR.

La cadena de búsqueda se aplicó en los repositorios seleccionados, en títulos, resúmenes y palabras clave de los estudios primarios. El proceso de búsqueda se realizó entre los meses octubre - diciembre del año 2017. Como resultado de la ejecución se identificaron un total de doscientos sesenta y siete (267) documentos. IEEE Xplore devolvió veinticuatro (24) artículos; Scopus sesenta y dos (62); ScienceDirect identificó sólo cinco (5) documentos y finalmente; SpringerLink fue la biblioteca que retornó una mayor cantidad, identificando ciento setenta y seis (176) artículos. Los estudios resultantes pueden visualizarse en ¹ accediendo a GitHub. GitHub es una plataforma de desarrollo colaborativo que proporciona herramientas para facilitar el desarrollo distribuido, especialmente útiles para proyectos de software de código abierto. Además, permite alojar repositorios de forma pública y gratuita [16]. La Figura 4 muestra la cantidad de artículos recuperados en cada repositorio.

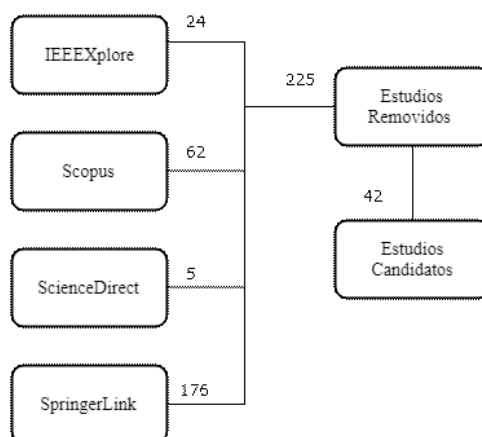


Fig. 4. Número de estudios identificados en las bibliotecas digitales.

1. <https://github.com/gramajoguadalupe/-Arg2018-Systematic-Review-2018/blob/master/RepositorioArticulos.xlsx>

3.3 Selección de Artículos Relevantes

Los estudios primarios candidatos al análisis fueron seleccionados abordando, tal como proponen Medina-López y otros [17], criterios de inclusión y exclusión que se detallan en la Tabla 3.

Tabla 2. Criterios de Inclusión y Exclusión.

CRITERIOS DE INCLUSIÓN
CI 1. Trabajos de investigación con propuestas concretas en la aplicación de técnicas de AAS en las actividades de la IR.
CI 2. Estudios publicados en librerías digitales entre los años 2002 y 2017.
CI 3. Estudios publicados en formato: Journal Article y Conference Paper.
CI 4. La cadena de búsqueda debe aparecer en el título, resumen o palabras clave del estudio.
CRITERIOS DE EXCLUSIÓN
CE 1. Trabajos de investigación cuya aplicación de técnicas de AAS, pertenezcan a un contexto distinto de las actividades de la IR.
CE 2. Propuestas e investigaciones cuya profundidad temporal excedan del intervalo de tiempo especificado para la elaboración del mapeo sistemático.
CE 3. Estudios publicados en formato de libro de texto, tesis, capítulo de libro, position paper, keynotes, opinión, tutorial, poster o panel.
CE 4. Estudios que no contengan en título, resumen o palabras clave, los términos utilizados en la cadena de búsqueda. Este criterio contribuye a identificar artículos que focalizan en el tema de análisis.

Como resultado del proceso de filtrado se identificaron un total de cuarenta y dos (42) documentos que detallan la aplicación del AAS en propuestas concretas relacionadas a la IR. Además, se detectaron doscientos veinticinco (225) falsos positivos, es decir, documentos que describen propuestas de AA, pero que se encuentran fuera del alcance de la revisión, es por ello que fueron excluidos. Posteriormente, los estudios identificados se categorizaron según el año de publicación. La Figura 5 muestra la distribución de artículos publicados por año, en ella, se puede apreciar que la popularidad del AA aplicado en la IR crece abruptamente en los dos últimos años (2016-2017).

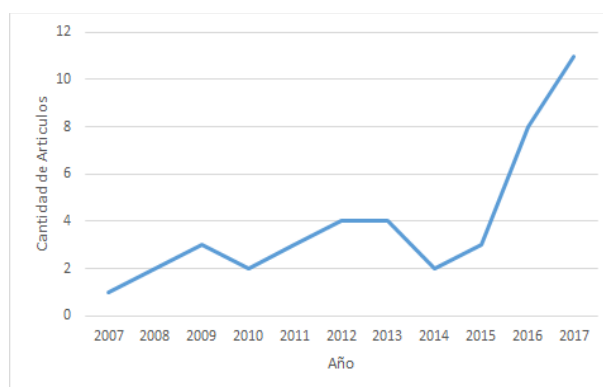


Fig. 5. Distribución de artículos seleccionados por año de publicación.

3.4 Búsqueda de Conceptos Claves

Luego de identificar los estudios candidatos, los autores leyeron y analizaron sus resúmenes y palabras claves, a fin de categorizar los tópicos más relevantes.

Los estudios hallados describen principalmente propuestas que abordan la identificación de requisitos especificados en lenguaje natural [18], investigaciones que proponen utilizar AA para la identificación de riesgos en los requerimientos [19] y enfoques para la clasificación de requisitos de software [20–22]. Se identificaron otras propuestas, que abordan la trazabilidad de los requisitos mediante técnicas de AA [4], otras proponen modelos de predicción para detectar artefactos propensos al cambio [23], algunos estudios, proponen modelos para predecir la performance de las configuraciones del sistema que pueden afectar el cumplimiento de determinados requerimientos [24].

La Figura 6 grafica la frecuencia de palabras claves en los artículos candidatos, con la intención de detectar las temáticas relevantes.

Frecuencia de Palabras Clave

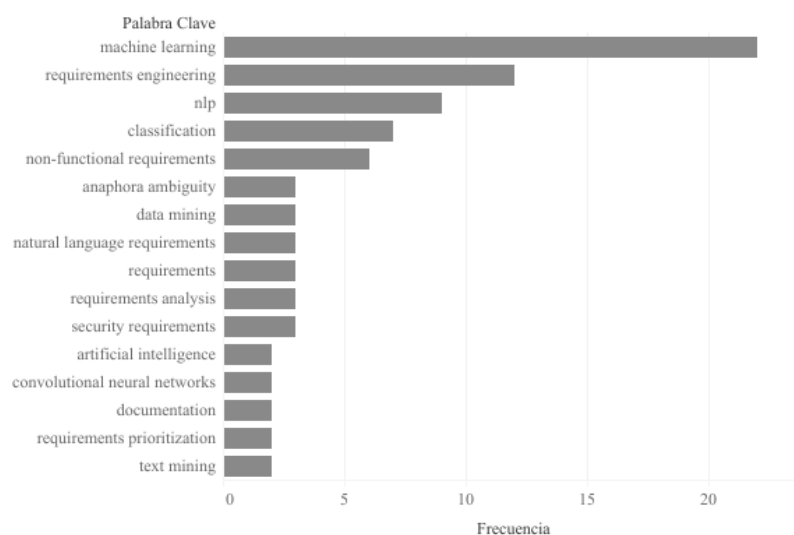


Fig. 6. Frecuencia de palabras claves por artículos seleccionados.

La frecuencia de las palabras claves proporciona una visualización general de los tópicos más relevantes en los artículos seleccionados. Mediante esta representación, se puede apreciar la presencia de las técnicas de clasificación, minería de datos y procesamiento de lenguaje natural en el campo del AA aplicadas a la IR.

3.5 Proceso de Extracción y Mapeo de Datos

El proceso de extracción de datos consiste en la lectura y análisis de los artículos identificados como relevantes, a fin de responder las preguntas de investigación. A continuación, se describen los resultados obtenidos.

Q1. ¿Cuáles son las actividades, tareas y desafíos de la IR respaldadas por las tecnologías de AA?

A partir de los estudios seleccionados es posible evidenciar la aplicación de las técnicas de AA en las actividades de la IR. Los artículos identificados demuestran que ingenieros e investigadores de la ciencia de datos utilizan el AA para obtener nuevas perspectivas y soluciones de diversos desafíos, que estaban hasta ese momento fuera de su alcance, aprovechando la infraestructura de hardware y disponibilidad de grandes volúmenes de datos. Esta revisión detectó que las actividades que se encuentran respaldadas por las tecnologías de la AA son las relacionadas a la gestión de requisitos. En cuanto a las actividades de educación de requisitos, Sharma y otros [25] proponen un método para identificar automáticamente los requisitos de seguridad, combinando el análisis lingüístico con técnicas de AA. Bagheri y otros [26] proponen un framework de apoyo a la toma de decisiones que emplea procesamiento de lenguaje natural y una ontología para procesar especificaciones de requisitos a modo de facilitar y brindar soporte a ingenieros y gerentes de proyectos. Otras propuestas abordan los problemas de priorización de requisitos. McZara y otros [27] proponen un método para ordenar y seleccionar los requerimientos de acuerdo a las prioridades de los stakeholders, mediante el empleo de técnicas de procesamiento del lenguaje natural. Duan y otros [28], proponen el uso de técnicas de minería de datos y AA para priorizar los requisitos de acuerdo con los intereses de las partes interesadas, los objetivos empresariales y las preocupaciones transversales, tales como los requisitos de seguridad y rendimiento. La eficacia del enfoque se evalúa a través de un caso de estudio basado en un conjunto de requisitos extraídos de un proyecto de código abierto denominado SugarCRM.

Por otro lado, se proponen modelos predictivos para la detección de datos y artefactos propensos al cambio, enfatizando en los requisitos del software. El modelo de predicción propuesto por Malhotra y Khanna [23] utiliza técnicas híbridas basadas en la búsqueda. Este estudio analiza y compara el rendimiento predictivo de cinco técnicas híbridas basadas en búsquedas y cuatro técnicas de aprendizaje automático, así como una técnica estadística para predecir clases propensas al cambio en seis paquetes de aplicaciones del sistema operativo Android. Guo y otros [24], aplica el AA para predecir la performance de ciertas configuraciones del sistema que puedan afectar a los requerimientos. Fitzgerald y otros [29] proponen una herramienta capaz de predecir los requerimientos que pueden fallar, dada la ambigüedad y escasez de información en la especificación de los requisitos. Sagrado y Aguila [30] proponen el uso de las redes Bayesianas para predecir las especificaciones de requisitos que deben ser revisadas.

Sharma y otros [18] proponen un método para detectar ambigüedad en los documentos de especificación de requisitos mediante el uso de redes bayesianas. Otros estudios, resuelven los desafíos de la IR mediante las técnicas de clasificación. Aguila y Sagrado [19] proponen el uso de clasificadores bayesianos para clasificar los riesgos en los re-

querimientos. Kurtanovic y Maalej [20] proponen clasificar automáticamente los requisitos de software en funcionales y no funcionales con técnicas de AAS. Paralelamente, se evalúa la precisión con la que se pueden identificar varios tipos de requerimientos no funcionales, en particular, los requisitos de usabilidad, seguridad, operación y rendimiento. Dekhtyar y Fong [31] consideran la aplicación de AA guiado por tensorflow, usando word2vec para resolver el problema de clasificación de requerimientos. Las técnicas de AA también son aplicadas para dar solución a los problemas de trazabilidad [4].

Q2. ¿Cuáles son los algoritmos de AA utilizados en las propuestas identificadas?

Los algoritmos detectados con mayor frecuencia de aplicación son Naives Bayes, Máquina de Soporte Vectorial, Redes Neuronales y Word2Vec. Sharma y otros [25], presentan un enfoque para identificar las reglas de negocio en las especificaciones de requisitos. El estudio tiene como objetivo determinar la eficacia con la que las reglas de negocio pueden ser identificadas y clasificadas en categorías mediante técnicas de AA. Las observaciones realizadas por los autores muestran que el algoritmo Máquina de Soporte Vectorial presenta un mejor rendimiento que otros clasificadores, tales como RandomForest y Naives Bayes. Sharma y otros [18], presentan una solución a la detección de ambigüedades en los requisitos de software, utilizando redes bayesianas para clasificar las declaraciones de los requisitos con ambigüedad anáfora pronominal a partir de un corpus de declaraciones potencialmente ambiguas, el cual es usado como un conjunto de datos de entrenamiento para el clasificador. En la propuesta de Dekhtyar y Fong [31], la aplicación del AA es guiado por tensorflow y se utiliza word2vec para resolver el problema de clasificación de requerimientos. Además, se comparan tres técnicas de AA para la identificación de los requisitos no funcionales. Las redes neuronales son aplicadas por Winkler y Vogelsang [32], para clasificar automáticamente el contenido de una especificación de requisitos expresados en lenguaje natural como "requisito" o "información", utilizando Redes Neuronales Convolucionales.

Q3. ¿Cuáles son las principales fuentes de datos utilizadas en los algoritmos de AA?

A partir de los trabajos analizados se identificaron tres repositorios de datos en [28], [33], [34], un proyecto de código abierto llamado SugarCRM, un documento de requisitos de formato libre proporcionado por Siemens Logistics and Automotive Organization e INCOSE, respectivamente. Esto denota escasez de repositorios de acceso abierto destinados a la investigación en el área de requerimientos, lo que genera dificultades en el proceso de validación de las propuestas. La tabla 4 identifica los artículos que mencionan las fuentes de datos utilizadas en sus propuestas.

Tabla 4. Fuentes de Datos de Requerimientos de Software.

Artículo	Repositorio	Actividad en IR
Duan, C., Laurent, P., Cleland-Huang, J., Kwiatkowski, C.: Towards automated requirements prioritization and triage. <i>Requir. Eng.</i> 14, 73–89 (2009).	SugarCRM	Priorización de Requisitos de software.

Cleland-Huang, J., Settimi, R., Zou, X., Solc, P.: Automated classification of non-functional requirements. <i>Requir. Eng.</i> 12, 103–120 (2007).	Siemens Logistics and Automotive Organization.	Detección y clasificación de Requisitos no funcionales de software.
Parra, E., Dimou, C., Llorens, J., Moreno, V., Fraga, A.: A methodology for the classification of quality of requirements using machine learning techniques. <i>Inf. Softw. Technol.</i> 67, 180–195 (2015).	International Council on Systems Engineering (INCOSE).	Evaluación de la calidad de los requisitos de software.

4. Resultados y Discusiones Generales

Uno de los objetivos de este MSL fue identificar las revistas científicas que utilizan a menudo los investigadores para difundir su trabajo del área de IR aplicando AA. La Figura 7 lista las revistas y conferencias más populares y su frecuencia de publicación.

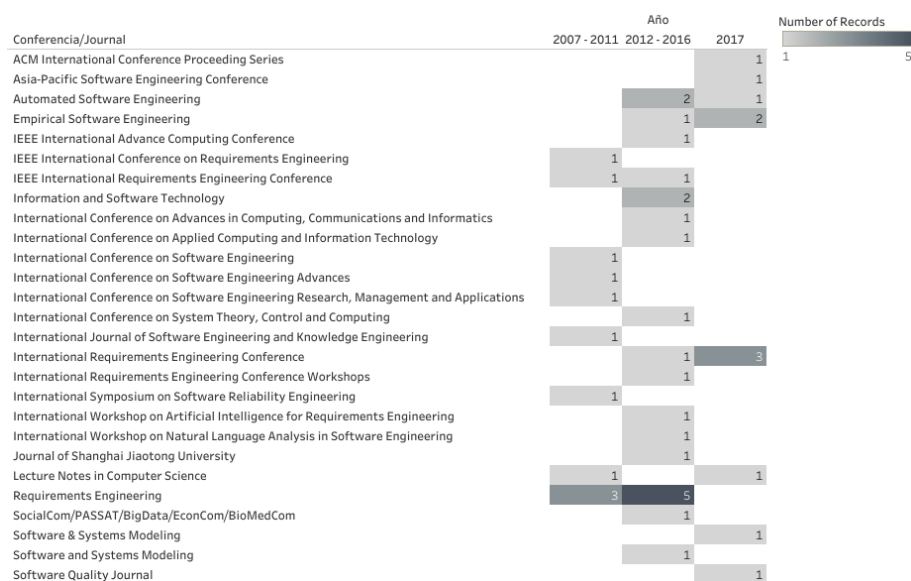


Fig. 7. Frecuencia de artículos seleccionados por Revista y Conferencias científicas.

Se puede apreciar que las revistas y conferencias que presentan mayor frecuencia de publicaciones en área de AA aplicada a la IR, son Requirements Engineering e International Requirements Engineering Conference representando un 18,6% y 11,6% respectivamente, de la muestra total de artículos seleccionados. Por otro lado, los estudios

que superaron los criterios definidos en la Sección 3 fueron cuarenta y dos (42). Específicamente, en este MSL se identificaron cuatro grandes áreas de aplicación de las técnicas de AA en las actividades de la IR, ellas son las relacionadas a la identificación y detección de requisitos, clasificación de requisitos en funcionales y no funcionales, predicción de fallas y las propuestas orientadas a la detección de ambigüedad en los documentos de especificación.

Sólo se hallaron tres (3) estudios que aplican técnicas de AA supervisado a las actividades asociadas a la trazabilidad requisitos. Además, se identificaron dos (2) estudios relacionados a la priorización de requisitos. Las propuestas analizadas han demostrado la posibilidad de ejecución y prueba de un amplio conjunto de algoritmos de ML. Los algoritmos de mayor presencia en los estudios son Naives Bayes, Máquina de Soporte Vectorial y Redes Neuronales Artificiales. Una de las conclusiones más notables en este trabajo, es la escasa presencia de propuestas relacionadas a las actividades de educación, actividad esencial para el desarrollo de cualquier producto de software. Resulta importante destacar la necesidad de continuar analizando los aportes específicos en las áreas detectadas, a fin de poder avanzar en la generación de propuestas concretas del uso de AA en los requisitos, como artefactos fundamentales del desarrollo de software. La Figura 8 visualiza los tópicos (actividades, tareas y desafíos) en el área de IR que emplean técnicas de AA con mayor frecuencia.

Frecuencia de los Topicos Analizados

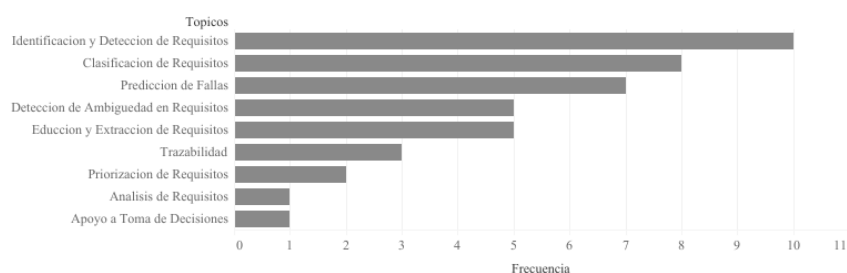


Fig. 8. Frecuencia de artículos seleccionados por desafíos de la IR.

En la siguiente sección se presentan las conclusiones a partir de los resultados obtenidos y se describen futuras líneas de investigación.

5. Conclusiones y Trabajos Futuros

Este trabajo analiza las diversas propuestas de investigación que emplean técnicas AA para abordar problemas tradicionales en la IR. Específicamente, se identificó que las actividades relacionadas a la clasificación de requisitos en funcionales y no funcionales son las que presentan una mayor incidencia en la aplicación de técnicas de AA del tipo supervisado. Los resultados obtenidos permiten evidenciar el valor práctico y la utilidad al aplicar algoritmos de AA sobre las actividades de la IR. Por otro lado, es

necesario destacar que no se encontraron trabajos que desestimen la aplicación de técnicas de AA en la obtención de mejoras y resolución de problemas en las actividades de la IR. Este hecho demanda la necesidad de continuar indagando en los beneficios y usos posibles de las técnicas de AA en el área de IR.

Los autores proponen como futura línea de investigación, el estudio de la aplicación de las técnicas de AAS para abordar la reutilización de artefactos de software a partir de documentos, modelos o representaciones provenientes de proyectos cuyos artefactos ya fueron validados. Esta propuesta, intenta contribuir con las tareas de los ingenieros y gerentes de proyecto durante la generación de nuevos artefactos en otros proyectos con características similares.

Referencias

1. Zhang, D.: *Advances in machine learning applications in software engineering*. IGI Global (2006).
2. Malhotra, R.: A systematic review of machine learning techniques for software fault prediction. *Appl. Soft Comput.* 27, 504–518 (2015).
3. Welling, M.: A first encounter with Machine Learning. Irvine, CA. Univ. Calif. 1–93 (2011).
4. Li, Z., Chen, M., Huang, L., Ng, V., Geng, R.: Tracing requirements in software design. In: *ACM International Conference Proceeding Series*. pp. 25–29 (2017).
5. Jirapanthong, W., Zisman, A.: Xtraque: traceability for product line systems. *Softw. Syst. Model.* 8, 117–144 (2009).
6. Giger, E., Pinzger, M., Gall, H.C.: Can we predict types of code changes? an empirical analysis. In: *Mining Software Repositories (MSR)*, 2012 9th IEEE Working Conference on. pp. 217–226 (2012).
7. Felbinger, H., Wotawa, F., Nica, M.: Test-Suite Reduction Does Not Necessarily Require Executing the Program under Test. In: *Proceedings - 2016 IEEE International Conference on Software Quality, Reliability and Security-Companion, QRS-C 2016*. pp. 23–30 (2016).
8. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic Mapping Studies in Software Engineering. In: *EASE*. pp. 68–77 (2008).
9. Kitchenham, B., Charters, S.: *Guidelines for performing Systematic Literature Reviews in Software Engineering*. (2007).
10. Samuel, A.L.: Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.* 3, 210–229 (1959).
11. Mitchell, T.M., others: *Machine learning*. 1997. Burr Ridge, McGraw Hill. 45, 870–877 (1997).
12. Russell, S.J., Norvig, P.: *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, (2016).
13. Thayer, R.H., Bailin, S.C., Dorfman, M.: *Software requirements engineering*. IEEE Computer Society Press (1997).
14. Pohl, K.: *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated (2010).
15. Collobert, R., Bengio, S., Mariéthoz, J.: Torch: a modular machine learning software library. (2002).
16. Squire, M.: Forge++: The changing landscape of FLOSS development. In: *System Sciences (HICSS)*, 2014 47th Hawaii International Conference on. pp. 3266–3275 (2014).
17. Medina-López, C., Marín-García, J.A., Alfalla-Luque, R.: Una propuesta metodológica para la realización de búsquedas sistemáticas de bibliografía (A methodological proposal for the

- systematic literature review). *WPOM-Working Pap. Oper. Manag.* 1, 13–30 (2010).
18. Sharma, R., Sharma, N., Biswas, K.K.: Machine Learning for Detecting Pronominal Anaphora Ambiguity in NL Requirements. In: *Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD)*, 2016 4th Intl Conf on. pp. 177–182 (2016).
 19. del Águila, I.M., Del Sagrado, J.: Requirement risk level forecast using Bayesian networks classifiers. *Int. J. Softw. Eng. Knowl. Eng.* 21, 167–190 (2011).
 20. Kurtanović, Z., Maalej, W.: Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning. In: *Requirements Engineering Conference (RE)*, 2017 IEEE 25th International. pp. 490–495 (2017).
 21. Abad, Z.S.H., Karras, O., Ghazi, P., Glinz, M., Ruhe, G., Schneider, K.: What works better? a study of classifying requirements. In: *Requirements Engineering Conference (RE)*, 2017 IEEE 25th International. pp. 496–501 (2017).
 22. Winkler, J.P., Vogelsang, A.: What does my classifier learn? A visual approach to understanding natural language text classifiers. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*. 10260 LNCS, 468–479 (2017).
 23. Malhotra, R., Khanna, M.: An exploratory study for software change prediction in object-oriented systems using hybridized techniques. *Autom. Softw. Eng.* 24, 673–717 (2017).
 24. Guo, J., Yang, D., Siegmund, N., Apel, S., Sarkar, A., Valov, P., Czarnecki, K., Wasowski, A., Yu, H.: Data-efficient performance learning for configurable systems. *Empir. Softw. Eng.* 1–42 (2017).
 25. Sharma, R., Bhatia, J., Biswas, K.K.: Automated identification of business rules in requirements documents. In: *Advance Computing Conference (IACC)*, 2014 IEEE International. pp. 1442–1447 (2014).
 26. Bagheri, E., Ensan, F., Gasevic, D.: Decision support for the software product line domain engineering lifecycle. *Autom. Softw. Eng.* 19, 335–377 (2012).
 27. McZara, J., Sarkani, S., Holzer, T., Eveleigh, T.: Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment. *Empir. Softw. Eng.* 20, 1721–1761 (2015).
 28. Duan, C., Laurent, P., Cleland-Huang, J., Kwiatkowski, C.: Towards automated requirements prioritization and triage. *Requir. Eng.* 14, 73–89 (2009).
 29. Fitzgerald, C., Letier, E., Finkelstein, A.: Early failure prediction in feature request management systems: an extended study. *Requir. Eng.* 17, 117–132 (2012).
 30. del Sagrado, J., del Águila, I.M.: Stability prediction of the software requirements specification. *Softw. Qual. J.* 1–21 (2017).
 31. Dekhtyar, A., Fong, V.: RE Data Challenge: Requirements Identification with Word2Vec and TensorFlow. In: *Requirements Engineering Conference (RE)*, 2017 IEEE 25th International. pp. 484–489 (2017).
 32. Winkler, J., Vogelsang, A.: Automatic classification of requirements based on convolutional neural networks. In: *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference Workshops, REW 2016*. pp. 39–45 (2016).
 33. Cleland-Huang, J., Settini, R., Zou, X., Solc, P.: Automated classification of non-functional requirements. *Requir. Eng.* 12, 103–120 (2007).
 34. Parra, E., Dimou, C., Llorens, J., Moreno, V., Fraga, A.: A methodology for the classification of quality of requirements using machine learning techniques. *Inf. Softw. Technol.* 67, 180–195 (2015).