

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/342603805>

Seizing Requirements Engineering Issues through Supervised Learning Techniques

Article in IEEE Latin America Transactions · July 2020

DOI: 10.1109/TLA.2020.9099757

CITATION

1

READS

77

3 authors:



[Maria Guadalupe Gramajo](#)

National University of Technology

6 PUBLICATIONS 4 CITATIONS

SEE PROFILE



[Luciana C. Ballejos](#)

National University of Technology

24 PUBLICATIONS 263 CITATIONS

SEE PROFILE



[Mariel Alejandra Ale](#)

National University of Technology

20 PUBLICATIONS 105 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Gestión de dependencias en la nube: Una propuesta eficiente para Java [View project](#)



Conceptual model for the development of competency-based curriculum generation systems in engineering careers [View project](#)

Seizing Requirements Engineering Issues through Supervised Learning Techniques

Maria Guadalupe Gramajo, *Graduate Member, IEEE*, Luciana Ballejos, and Mariel Ale

Abstract—In recent years, the popularity of machine learning techniques has grown due to the availability of large volumes of data and the increased processing capacity of computers. Despite the inherent value of these techniques, few studies have attempted to summarize how machine learning algorithms, especially supervised learning have contributed to task automation and resolving challenges in Requirements Engineering. This paper proposes a systematic mapping of the literature to identify and analyze proposals detailing the uses and applications of supervised learning in Requirements Engineering between 2002-2018. The objective of this research is to identify trends, data sets and methods used. Thirty-three studies were selected based on defined inclusion and exclusion criteria. The results revealed that researchers using these techniques focuses on eight broad categories: Detection of linguistic problems in requirements documents and artifacts written in natural language; Classification of document content ; Traceability; Effort estimation; Requirements analysis; Failure prediction; Quality and Detection of business rules. The most used supervised learning techniques were Support Vector Machine, Naive Bayes, Decision Tree, K-Nearest Neighbour and Random Forest. Twenty-five (25) public and twenty-eight (28) private data sources were identified. Among the most used public data sources are Predictor Models In Software Engineering (PROMISE), iTrust Electronic Health Care System and Metric Data Program (MDP) from NASA.

Index Terms—Machine Learning, Supervised Learning, Software Requirement, Requirements Engineering.

I. INTRODUCCIÓN

EN LA actualidad el software resulta un activo esencial en todas las organizaciones. Las restricciones y expectativas de las partes interesadas 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 desafíos que surgen durante el proceso de desarrollo de software [1]. La necesidad de abordar este escenario ha generado un creciente interés por la integración de nuevas tecnologías, especialmente por aquellas que emergen del área de la Inteligencia Artificial (IA). El surgimiento de nuevas estrategias de aprendizaje y el aumento de la capacidad de procesamiento en los ordenadores posicionaron a la IA como una herramienta poderosa y accesible, capaz de ser implementada como un componente clave durante el desarrollo de sistemas de software [2]. Estudios recientes han demostrado resultados prometedores al combinar técnicas tradicionales como el procesamiento de lenguaje natural, razonamiento basado en reglas y representación del

conocimiento con técnicas de aprendizaje automático y aprendizaje profundo para brindar soluciones inteligentes a las tareas involucradas en el desarrollo de software [3].

La integración de las tecnologías de IA en el área de la Ingeniería de Software, tiene por objetivo automatizar tareas que demandan esfuerzos intensivos, optimizar el proceso de desarrollo y consecuentemente, obtener sistemas de calidad. Diversas técnicas y algoritmos de IA, especialmente las técnicas de aprendizaje automático (AA) han impactado en las actividades involucradas en el ciclo de vida de desarrollo de software. En este sentido, la Ingeniería de Requerimientos (IR), subdisciplina esencial de la Ingeniería del Software dado que se enfoca en capturar y definir los requerimientos del sistema a desarrollar, se ha visto ampliamente beneficiada a partir de la implementación de dichas técnicas.

Algunas de las aplicaciones halladas en la literatura abordan la predicción de trazabilidad y clasificación automática de requerimientos [4], predicción de cambios en el código [5], detección de ambigüedad en documentos escritos en lenguaje natural [6] y pruebas automáticas de software [7]. Sin embargo, a pesar del valor inherente de estas técnicas, pocos estudios han intentado proporcionar una visión general acerca de cómo éstos algoritmos, especialmente los algoritmos de aprendizaje del tipo supervisado, han contribuido a la resolución de los desafíos en la IR. Por esta razón, definimos como propósito general de este artículo la elaboración de un mapeo sistemático de la literatura, a fin de analizar la intersección de éstas dos grandes áreas: la "*Ingeniería de Requerimientos*" y el "*Aprendizaje Automático Supervisado*" y, consecuentemente, analizar la evolución en el tiempo de esta línea de investigación.

Un mapeo sistemático se enfoca en probar la existencia de estudios en un campo de interés particular, para proporcionar una visión general del área de investigación. Además, permite identificar áreas temáticas que aún precisan exploración y determinar las tendencias de estudios futuros [8], resultando una herramienta útil y de gran valor para los profesionales que desean iniciar sus investigaciones en un dominio determinado.

Creemos que este estudio puede promover el uso de las técnicas de AA en el diseño de nuevas propuestas e investigaciones, para abordar aspectos aún no resueltos y servir de guía respecto al uso de los algoritmos de aprendizaje supervisado en el área de la IR.

Las principales contribuciones de este trabajo son: i) revisión comprensiva de las investigaciones que apuestan a la aplicación de técnicas de aprendizaje supervisado en el área de IR, ii) identificación y análisis de las técnicas de aprendizaje supervisado que apoyan las actividades de la IR, iii) la iden-

tificación de las fuentes de datos utilizadas en las propuestas de investigación y, iv) detección de oportunidades y lagunas de investigación en el campo del aprendizaje supervisado aplicado a la IR.

El contenido de este artículo se encuentra organizado de la siguiente manera. La Sección 2 introduce conceptos preliminares respecto al Aprendizaje Automático, enfatizando especialmente en el Aprendizaje Supervisado y la Ingeniería de Requerimientos a fin de establecer un marco teórico introductorio. La Sección 3 describe algunos trabajos relacionados en el área para detectar antecedentes de investigación. La Sección 4 detalla la metodología aplicada en el desarrollo del mapeo sistemático de la literatura. La Sección 5 presenta los resultados y discusiones. La Sección 6 describe posibles amenazas a la validez. Finalmente, la Sección 7 presenta las conclusiones y la futura línea de investigación.

II. CONCEPTOS PRELIMINARES

Esta sección incluye los principales conceptos sobre el AA, enfatizando especialmente en el aprendizaje supervisado y la IR. Posteriormente, se describen los algoritmos aplicados con mayor frecuencia en el área de la IR.

A. Aprendizaje Automático

El interés y la popularidad del AA ha crecido en los últimos años. Aunque no es un concepto nuevo, la disponibilidad de grandes volúmenes de datos y el aumento de la capacidad de procesamiento en los ordenadores permitió experimentar e indagar aún más sobre sus usos y aplicaciones en diversos dominios. El AA es una disciplina de la IA, compuesta por un conjunto de técnicas que permiten a los ordenadores aprender de los datos, realizar generalizaciones y predicciones a partir de ellos, lo cual facilita la toma de decisiones. Samuel [9] es un pionero en el AA y describe este concepto como el campo de estudio que 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 [10], quien sostiene que un ordenador aprende una tarea particular T , considerando las experiencias del tipo E , con respecto a una medida de rendimiento P , si el ordenador mejora efectivamente su rendimiento P , en la tarea T , a partir de la experiencia E .

Estas técnicas permiten a los ordenadores simular el aprendizaje humano por medio de algoritmos y adquirir conocimiento sobre un dominio en particular, con lo cual es posible mejorar el rendimiento de algunas tareas bajo ese dominio en base al nuevo conocimiento adquirido.

De acuerdo al enfoque utilizado en el proceso de aprendizaje, los algoritmos de AA, puede clasificarse en cuatro grandes categorías, éstas incluyen: (a) *aprendizaje supervisado*, (b) *aprendizaje no supervisado*, (c) *aprendizaje semi-supervisado* y (d) *aprendizaje por refuerzo*.

En el aprendizaje supervisado, los algoritmos razonan desde instancias suministradas externamente y correctamente etiquetadas para producir hipótesis generales, las cuales permiten hacer predicciones sobre instancias futuras [11]. En el aprendizaje no supervisado, los algoritmos disponen de datos con estructura desconocida y sin etiquetar. Su objetivo es

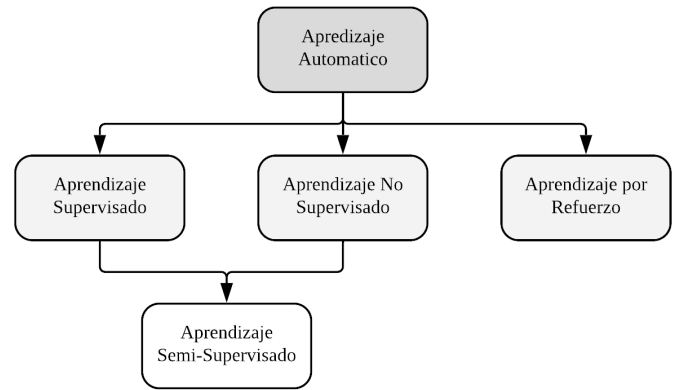


Fig. 1. Clasificación de algoritmos de aprendizaje automático..

la extracción de información mediante la exploración de la estructura de dichos datos. Los algoritmos de aprendizaje no supervisado se centran en la búsqueda de patrones ocultos y el análisis de agrupamiento a partir de conjuntos de datos [12]. El aprendizaje semi-supervisado permite utilizar datos no etiquetados para mejorar las tareas de aprendizaje supervisado cuando los datos etiquetados son escasos [13]. En el aprendizaje por refuerzo los algoritmos aprenden basándose en la retroalimentación externa proporcionada por una entidad o entorno pensante [14]. La Fig. 1 presenta la clasificación de los algoritmos de AA mencionados anteriormente.

Esta investigación establece como alcance el análisis de los estudios que emplean aprendizaje supervisado en el área de la IR; estas técnicas se caracterizan por su gran capacidad de predicción y clasificación. El aprendizaje supervisado, permite aprovechar las fuentes de datos disponibles para realizar predicciones de comportamiento, lo que permite desarrollar sistemas de apoyo y recomendación en diferentes dominios de aplicación, un área de interés fundamental para los autores.

B. Aprendizaje Supervisado

El aprendizaje supervisado es una técnica de AA que emplea datos etiquetados, llamados datos de entrenamiento, para construir un modelo predictivo a fin de predecir la etiqueta de datos sin etiquetar.

El conjunto de datos de entrenamiento incluye datos de entrada y valores de respuesta. A partir de éste, el algoritmo de aprendizaje supervisado busca crear un modelo que pueda realizar predicciones acerca de los valores de respuesta para un nuevo conjunto de datos. Con frecuencia se utiliza un conjunto de datos de prueba para validar el modelo. Si se utilizan conjuntos de datos de entrenamiento de mayor tamaño, es posible generar modelos cuya capacidad predictiva es mayor, obteniendo buenos resultados sobre nuevos conjuntos de datos [15].

C. Algoritmos de AA aplicados en la IR

En esta subsección se describen los algoritmos de AA del tipo supervisado aplicados con mayor frecuencia en la IR.

El algoritmo de aprendizaje supervisado, *Máquina de Soporte Vectorial* (SVM), puede ser utilizado para tareas de

clasificación o regresión binaria. Las SVM proporcionan un enfoque alternativo a la categorización de entidades y han sido ampliamente utilizadas en aplicaciones relacionadas al procesamiento de lenguaje natural, reconocimiento de voz e imagen.

Una SVM mapea los puntos de entrada a un espacio de características de una dimensión mayor (i.e. si los puntos de entrada están en \mathbb{R}^2 estos son mapeados por la SVM a \mathbb{R}^3). Este algoritmo clasifica los datos al encontrar el mejor hiperplano que separe los puntos de entrada y maximice el margen M entre las clases en el espacio de características. Los vectores de soporte son los puntos de datos más cercanos al hiperplano de separación [16].

Naive Bayes (NB) es un algoritmo de aprendizaje automático basado en el Teorema de Bayes con una fuerte suposición de independencia condicional entre los predictores respecto de su clase o etiqueta. En otras palabras, un clasificador NB supone que la presencia de una característica particular en una clase no está relacionada con la presencia de otra característica [17]. Esta suposición de independencia condicional de los predictores permite al clasificador NB estimar los parámetros necesarios para una clasificación precisa mientras usa menos datos de entrenamiento respecto a otros clasificadores. Esto lo hace particularmente efectivo para conjuntos de datos que contienen muchos predictores.

El algoritmo *K-Nearest Neighbour* (KNN) se caracteriza por ser flexible y con una mecánica fácil de entender. Este algoritmo se basa en el aprendizaje por analogía, es decir, comparando un determinado individuo de prueba o punto de interés con ejemplos de entrenamiento que son similares a él. Los ejemplos de entrenamiento se describen mediante n atributos. Cada ejemplo o individuo representa un punto en un espacio n -dimensional. El valor de K representa la cantidad de puntos vecinos considerados para clasificar un individuo de prueba o punto de interés. La proximidad de los K vecinos más cercanos se define en términos de una métrica de distancia como la distancia euclídea [18].

El algoritmo *Decision Tree* (DT) es usado para resolver problemas de clasificación y regresión. Este algoritmo está basado en la lógica, donde los conjuntos de datos se modelan en estructuras jerárquicas en forma de árbol compuestas por decisiones lógicas. Cada nodo representa un atributo a evaluar, mientras que las ramas indican las opciones de decisión sobre el atributo dado y cada hoja representa un resultado. El resultado final será una estructura de árbol donde cada nodo interno representa un condicional para el valor de un atributo en particular y cada hoja representa la decisión para una clase en particular. En el caso de un nuevo individuo desconocido, el árbol procede a evaluar cada condicional hasta llegar al final de una de las hojas del árbol para etiquetar este nuevo caso. La complejidad del árbol se controla mediante la utilización de criterios de parada y métodos de poda. Mientras que las métricas utilizadas para medir la complejidad del árbol son el número de nodos, el número de hojas, la profundidad de los árboles y el número de atributos utilizados [19].

El algoritmo *Random Forest* combina un conjunto de árboles de decisión y entrena a cada uno con un conjunto de observaciones tomadas al azar. Las predicciones finales del al-

goritmo se generan promediando las predicciones individuales de cada árbol. Random Forest es un método de aprendizaje utilizado para problemas de clasificación y regresión [20].

D. Ingeniería de Requerimientos

La IR provee el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional [21]. Las actividades de este proceso incluyen: obtención, análisis, especificación, validación, verificación y gestión de requerimientos [22].

Sin embargo, dada las exigencias de comunicación y colaboración entre las partes interesadas, la IR está sujeta a diversos desafíos. Especialmente, durante las actividades de educación y especificación de requerimientos, en las cuales es posible incurrir en la inserción de errores y malentendidos. Es por ello, que a menudo esta fase debe afrontar diversas dificultades, tales como especificaciones de requerimientos incompletas y ambiguas, intereses en conflicto de las partes interesadas, volatilidad en los requerimientos y restricciones de tiempo, entre otros, que influyen en la calidad y el éxito del proyecto de software. Cuando estas dificultades no son resueltas, se generan fallas, esfuerzo adicional para el equipo de desarrollo y consecuentemente, incrementan los costos del proyecto de software.

Considerando estos retos y el surgimiento de nuevas técnicas de aprendizaje de IA, cuyo valor ha sido ampliamente reconocido en diversos dominios de aplicación, la comunidad científica ha comenzado a experimentar sobre la aplicación de diversos métodos de AA en las actividades de la IR, a fin de resolver dificultades y aspectos aún no resueltos. En los últimos años, investigadores y profesionales han explorado la utilidad del AA en aplicaciones de software para enriquecer técnicas y procedimientos tradicionales que conduzcan a contribuciones significativas en la IR. Por lo tanto, los aportes y mejoras que se puedan realizar en el área favorecerá a todo el ciclo de vida del software. Por esta razón, en esta investigación se propone brindar una visión general de la literatura acerca de cómo los investigadores han aprovechado los algoritmos de AA para automatizar y optimizar las diversas actividades involucradas en dicha área.

III. TRABAJOS RELACIONADOS

En esta sección se describirán brevemente algunos trabajos relevantes, que buscan responder preguntas de investigación y detectar oportunidades de mejora en la IR, a fin de identificar antecedentes. La Tabla I presenta los artículos analizados y sus correspondientes preguntas de investigación.

Ambreen y otros [23] presentan un mapeo sistemático en el cual definen preguntas de investigación enfocadas en la identificación de evidencia empírica en la IR. El estudio de mapeo se basa en 270 estudios empíricos extraídos de cuatro bases de datos ACM, IEEE, SpringerLink y ScienceDirect, durante el período 1990-2012. La obtención, el análisis y la gestión de requerimientos, presentan mayor frecuencia de publicación y

TABLA I
ESTUDIOS RELACIONADOS

Estudio	Pregunta de Investigación
Empirical research in requirements engineering: trends and opportunities	RQ1. ¿Cuál es el estado del arte en estudios empíricos de la IR? RQ2. ¿Cuál es la evidencia empírica en la literatura de IR?
Requirements Engineering Techniques: A Systematic Literature Review	RQ1. ¿Cuáles son las técnicas utilizadas en la IR? RQ2. ¿Cuáles son las limitaciones de las técnicas existentes utilizadas en la IR? RQ3. ¿Cómo afectan los cambios en los requerimientos de software al análisis de requerimientos?
A Requirements Engineering Techniques Review in Agile Software Development Methods	RQ1. ¿Qué enfoques ágiles existen que aplican las técnicas tradicionales de la IR en los procesos de desarrollo ágil? RQ2. ¿Qué técnicas de IR se utilizan habitualmente para obtener los requerimientos de los usuarios en los procesos de desarrollo ágil?

consecuentemente, se identifican como actividades de mayor interés.

Los resultados obtenidos reflejan que la verificación y validación de requerimientos, a pesar de su relevancia carece de evidencia empírica. Este trabajo de investigación evidencia que las tareas relacionadas a los requerimientos no funcionales y la IR en general, son emergentes en relación a las fases que conforman el ciclo de vida de desarrollo de software. Por otro lado, el uso de ontologías y propuestas relacionadas a la detección de patrones de aplicación de las técnicas de IR en pequeñas y medianas empresas, han comenzado a ser investigadas. Finalmente, la investigación concluye que existe un interés limitado en evaluar y comparar propuestas existentes, y que la comunidad científica se enfoca más bien, en proponer otras nuevas. Solo el 6% de los estudios se identificaron como trabajos empíricos. Los autores de este artículo también detectaron que dos tercios de los artículos analizados se efectuaron en entornos empresariales controlados, sin la suficiente información para replicar los resultados. Adicionalmente, se presenta una clasificación de las áreas emergentes de la IR en relación a las propuestas de investigación.

Matyokurehwa y otros [24] examinan diversas técnicas utilizadas en la IR durante el período 2000-2016. El objetivo de la investigación es encontrar técnicas que puedan ser aplicadas a casos concretos durante el desarrollo de un proyecto de software, sus limitaciones y cómo los cambios en los requerimientos pueden afectar su análisis. Los estudios primarios analizados presentaron cuarenta y tres (43) técnicas que abordan los requerimientos entregados por los stakeholders, pero ninguna puede resolver de manera precisa todas las situaciones que se presentan durante el desarrollo de un proyecto. En cuanto a las limitaciones se observaron que las técnicas identificadas están circunscritas a una sola actividad de la IR. Por último, los autores de este trabajo identificaron que los cambios no previstos pueden generar potenciales problemas con el presupuesto y la estimación del tiempo de entrega de los hitos del proyecto. Esta investigación

señala como potencial laguna de investigación, el desarrollo de técnicas que aborden las actividades de la IR. Considerando que es necesario idear técnicas que sean fáciles de monitorear y adoptar en los proyectos de desarrollo de software.

Zamudio y otros [25] presentan una revisión que aborda temas relacionados a las aplicaciones de las técnicas tradicionales de la IR en las metodologías ágiles de desarrollo de software. En particular los autores analizaron las siguientes metodologías ágiles: SCRUM, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD) y Crystal Family. Los resultados obtenidos permiten determinar que las técnicas tradicionales de la IR son aplicadas en las metodologías ágiles sin identificar un patrón de acerca de su uso. En el caso de eXtreme Programming (XP), Scrum y Crystal se identificaron las siguientes técnicas para la tarea de educación de requisitos: entrevistas, lluvia de ideas, priorización, modelado de requerimientos, construcción de escenarios y casos de usos. Mientras que para el análisis de los requerimientos frecuentemente se adoptan en la mayoría de metodologías, los diagramas de lenguaje unificado de modelado (UML). Los autores concluyen su trabajo sosteniendo que los métodos ágiles no poseen técnicas que difieren con las usadas en las metodologías tradicionales para la extracción de requerimientos. Y aún más, las metodologías ágiles poseen flujos de trabajos donde se aplican las técnicas tradicionales de la IR continuamente a medida que los requerimientos van mutando.

Luego del análisis de estas propuestas es posible concluir que la IR es reconocida como una subdisciplina de la Ingeniería de Software emergente y con lagunas de investigación. El objetivo del análisis de los trabajos mencionados anteriormente es obtener antecedentes y detectar puntos de interés en común que permiten argumentar y justificar el desarrollo de esta investigación. A pesar que las investigaciones analizadas abordan alcances y preguntas de investigación diferentes a las definidas en este artículo, todas coinciden en la necesidad de enriquecer y mejorar dicha área, la cual es esencial para el éxito de un proyecto de software. Estamos convencidos que la IR puede hallar oportunidades de mejora y beneficiarse de la aplicación de nuevas tecnologías y métodos, tales como el AA y es en este aspecto en el que se enfoca esta investigación.

IV. METODOLOGÍA

El método utilizado para desarrollar este mapeo sistemático de la literatura se basa en los lineamientos establecidos en las contribuciones de Petersen y otros [8], una propuesta ampliamente aceptada por la comunidad científica para realizar estudios de mapeo en el área de la Ingeniería de Software. El proceso de investigación consta de cinco etapas principales: (a) definición de las preguntas de investigación, (b) búsqueda e identificación de estudios primarios mediante cadenas de búsqueda en bibliotecas académicas digitales, (c) selección de los estudios mediante criterios de inclusión y exclusión, (d) revisión de estudios seleccionados a través de la lectura comprensiva de resúmenes, búsqueda de palabras clave y conceptos que reflejen el contexto de investigación, y, por último, e) extracción y análisis de datos. La Fig. 2 sintetiza el método aplicado en esta investigación.

TABLA II
PREGUNTAS DE INVESTIGACIÓN

Pregunta de Investigación	Motivación
Q1. ¿Cuáles son las actividades de la IR respaldadas por el aprendizaje supervisado?	M1. Identificar propuestas que utilizan el aprendizaje supervisado para abordar las actividades de la IR.
Q2. ¿Cuáles son los algoritmos de aprendizaje supervisado utilizados para resolver dificultades en las actividades de la IR?	M2. Detectar los principales algoritmos de aprendizaje supervisado aplicados para resolver dificultades y desafíos presentes en la IR.
Q3. ¿Cuáles son las principales fuentes de datos utilizadas para ejecutar los algoritmos de aprendizaje supervisado?	M3. Identificar repositorios y fuentes de datos utilizados como entrada para la ejecución de los algoritmos de aprendizaje supervisado.

Las subsecciones posteriores, describen los elementos de entrada y salidas obtenidos, como resultado de la metodología aplicada.

A. Preguntas de Investigación

Esta investigación identifica propuestas que aplican técnicas de AA del tipo supervisado para resolver desafíos que surgen durante las actividades de la IR. A fin de definir el alcance de este artículo se formularon tres preguntas de investigación. La Tabla II presenta las preguntas y los criterios que motivan su formulación.

La primera pregunta tiene por objetivo identificar las tareas y actividades de la IR que han sido abordadas, mediante el uso de las técnicas de AA en diversas contribuciones académicas halladas en la literatura. La finalidad es descubrir tendencias respecto a la aplicación de estas técnicas y determinar las actividades que han resultado más beneficiadas, a partir de estos aportes y propuestas. El propósito de la segunda pregunta es determinar posibles correspondencias de uso entre los algoritmos de AA aplicados en las diversas propuestas y las actividades y/o desafíos de la IR. Por último, la tercera pregunta pretende identificar y difundir los conjuntos de datos disponibles en la web y la literatura, a fin proveer a los profesionales de este campo de estudio, recursos que faciliten la prueba y validación de los modelos AA.

B. Estrategia de Búsqueda

Los estudios se identificaron consultando cuatro bibliotecas en línea: (i) IEEEExplore; (ii) ScienceDirect; (iii) SpringerLink y (iv) Scopus. Estas bibliotecas fueron seleccionadas porque son consideradas fuentes importantes de estudios en el área Informática en general, y de Ingeniería de Software en particular, poseen repositorios de conferencias y revistas relevantes. Asimismo, sus motores de búsqueda admiten la posibilidad de realizar búsquedas avanzadas basadas en diferentes criterios y formatos de cadena.

Los términos utilizados para construir la consulta de búsqueda fueron acordados entre los autores, teniendo en cuenta las directrices propuestas por Petersen y otros [8]. Estos

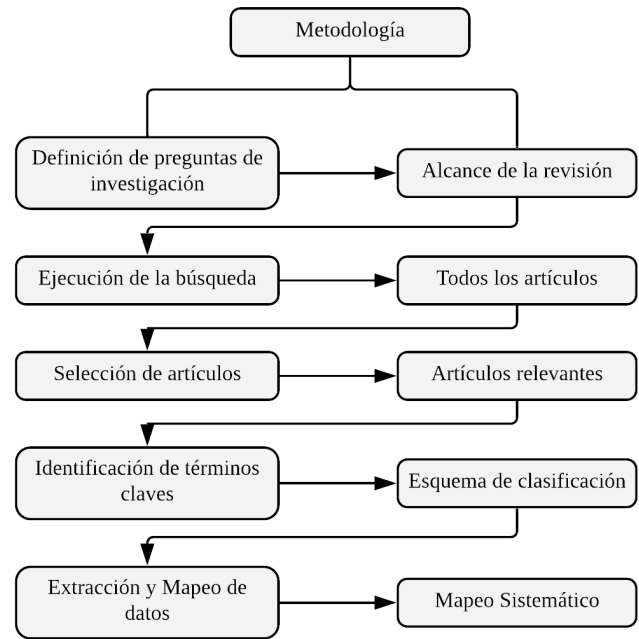


Fig. 2. Metodología de investigación [8].

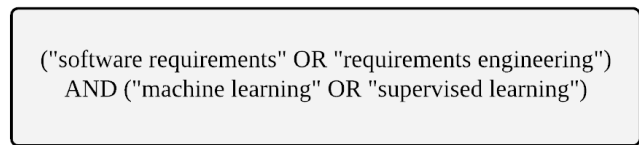


Fig. 3. Cadena de búsqueda genérica.

se mencionan a continuación: (1) Identificar los términos principales en las preguntas de investigación; (2) Extraer palabras clave de documentos relevantes a la temática; (3) Identificación de términos de búsqueda en títulos, resúmenes y palabras clave relevantes; (4) Identificar sinónimos y abreviaturas alternativas; (5) Construir cadenas de búsqueda, concatenando las palabras identificadas mediante operadores booleanos "AND" para enlazar los términos principales y "OR" para incorporar sinónimos alternativos. La cadena de búsqueda genérica resultante se muestra en la Fig. 3.

La misma fue adaptada a la sintaxis de los motores de búsqueda de cada repositorio. Si bien, este trabajo aborda el análisis de las investigaciones que aplican aprendizaje supervisado, en las simulaciones de búsqueda realizadas se observó la presencia de estudios vinculados a la temática que utilizaron como palabra clave el término aprendizaje automático sin especificar el tipo de enfoque de aprendizaje aplicado (supervisado, no supervisado, semi-supervisado, por refuerzo).

Es por ello, que se añadió este término a la cadena de búsqueda.

Con la intención de impulsar a los investigadores a trabajar en propuestas innovadoras y alentar a la comunidad en general a trabajar en nuevas tecnologías de AA, en el año 2002 fue liberada la librería Torch. Una librería de AA bajo licencia de

TABLA III
CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN

Criterios de Inclusión	Criterios de Exclusión
CI1. Trabajos de investigación con propuestas concretas en la aplicación de técnicas de aprendizaje automático supervisado en las actividades de IR	CE1. Trabajos de investigación cuya aplicación de técnicas de aprendizaje automático supervisado, pertenezcan a un contexto distinto al de IR.
CI2. Estudios publicados en librerías digitales entre los años 2002 y 2018	CE2. Propuestas e investigaciones cuya profundidad temporal excedan del intervalo de tiempo especificado para la elaboración del mapeo sistemático.
CI3. La cadena de búsqueda debe aparecer en el título, resumen o palabras clave del estudio	CE3. Estudios que no contengan en título, resumen o palabras clave, los términos utilizados en la cadena de búsqueda.
CI4. Estudios publicados en idioma inglés.	CE4. Trabajos de investigación no redactados en idioma inglés.

código abierto BSD, distribución de software Berkeley, que incluye los algoritmos y modelos más populares, tales como máquina de soporte vectorial (SVM), modelos gaussianos mixtos, clasificadores bayesianos, K-vecinos más cercanos (KNN), redes neuronales artificiales (ANN), entre otros [26].

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 estudios en este trabajo se acotó al período que parte desde el citado hito hasta el año 2018, planteando como desafío descubrir oportunidades de investigación y lagunas en el campo del aprendizaje supervisado aplicado a la IR.

C. Criterios de Inclusión y Exclusión

Esta investigación se centra en el análisis de propuestas cuyo foco de investigación es la aplicación de técnicas de aprendizaje automático supervisado en la IR. Fueron incluidas publicaciones redactadas en idioma inglés, debido al bajo nivel de difusión de publicaciones y artículos redactados en otros idiomas en revistas relevantes de Ingeniería de Software.

Los términos incluidos en la cadena de búsqueda deben aparecer al menos en el título, resumen o palabras clave, lo que contribuye a identificar los artículos que se centran en el tema de análisis. Se incluyeron artículos revisados por pares publicados en el período 2002-2018, considerando como un hito relevante en el campo del AA la liberación de la librería Torch.

Fueron excluidos los estudios no revisados por pares, duplicados, no redactados en inglés, documentos redundantes de la misma autoría, documentos del tipo tesis, position papers, panel de discusiones y, aquellos que no incluían técnicas de aprendizaje automático supervisado para apoyar las actividades de IR. La Tabla III resume los criterios de inclusión y exclusión considerados para seleccionar los estudios candidatos al análisis.

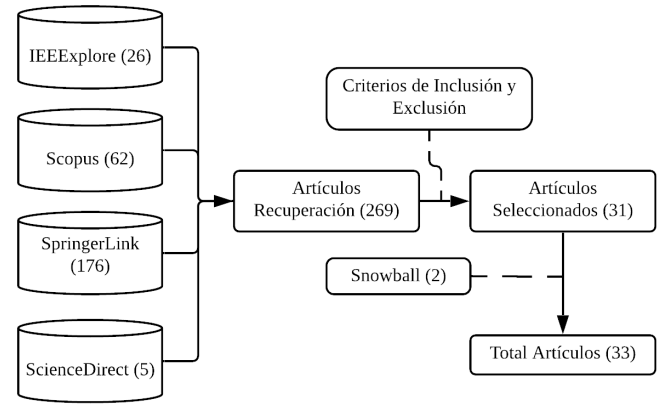


Fig. 4. Número de estudios identificados por repositorio académico.

D. Extracción de Datos

La cadena de búsqueda se aplicó en los repositorios seleccionados, sobre títulos, resúmenes y palabras clave. Fueron identificados doscientos sesenta y nueve (269) documentos como resultado de la ejecución. La biblioteca IEEEExplore devolvió veintiséis (26) 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. La Fig. 4 sintetiza la cantidad de estudios recuperados por repositorio académico y los artículos seleccionados.

Los artículos recuperados de los repositorios académicos fueron organizados en un formulario de extracción de datos, por título, año de publicación y revista o conferencia. Cada uno de estos artículos fueron analizados en busca de aquellos que cumplieran los criterios de inclusión, mientras que los que no cumplieran con éstos fueron excluidos.

Como resultado del proceso de filtrado, fueron seleccionados treinta y uno (31) documentos que detallan explícitamente la implementación de técnicas de aprendizaje supervisado en el área de la IR. La estrategia de "bola de nieve hacia atrás" [27] se aplicó posteriormente para detectar estudios asociados con la temática de investigación. Esta estrategia consiste en revisar las referencias bibliográficas citadas en los estudios identificados como relevantes para reconocer artículos relacionados a la temática de investigación. Como resultado de este proceso, se añadieron dos (2) estudios. El número total de estudios candidatos para el proceso de revisión y análisis fue de treinta y tres (33).

E. Síntesis de Datos

Esta investigación analiza treinta y tres (33) estudios, resultantes del proceso de selección. A continuación, se detallan los resultados obtenidos y luego, se responden las preguntas de investigación.

La distribución de los estudios publicados en función del tiempo permite visualizar las tendencias respecto al empleo de técnicas de aprendizaje supervisado en el área de la IR. La Fig. 5 muestra la distribución de los artículos seleccionados por año de publicación.

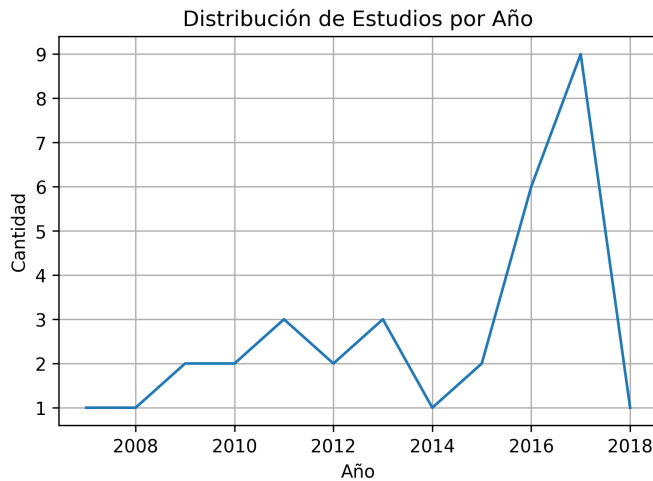


Fig. 5. Distribución de estudios seleccionados por año de publicación.



Fig. 6. Distribución de estudios publicados en conferencias científicas.

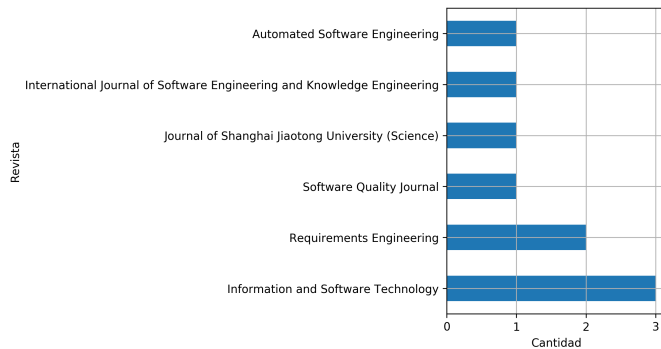


Fig. 7. Distribución de estudios seleccionados por año de publicación.

La Fig. 5 muestra un aumento significativo del 67% respecto a propuestas de investigación que utilizan métodos de aprendizaje supervisado para dar soporte a las diversas tareas de la IR durante el período (2016-2018). En consecuencia, es notable el posicionamiento de las tecnologías de AA como tema de interés en los últimos dos años. Probablemente, esto se deba a los importantes avances respecto al incremento de la capacidad de procesamiento en los ordenadores y el surgimiento de nuevas técnicas que otorgan mayor poder de aprendizaje. Otro aspecto relevante, es el reconocimiento de la comunidad científica respecto al valor del AA para automatizar tareas y resolver dificultades en diversos dominios, lo que sin

dudas ha motivado a los profesionales de la IR a investigar sobre su uso en dicha área. Con la finalidad de ampliar las capacidades y recursos de los desarrolladores para brindar soluciones inteligentes a los desafíos presentes en la IR.

Como se mencionó anteriormente, un mapeo sistemático tiene como objetivo identificar las revistas y conferencias científicas utilizadas frecuentemente por los investigadores para difundir sus trabajos de investigación. Es por ello, que en la Fig. 6 se identifican las conferencias más populares y su frecuencia de publicación, a fin de reconocer los potenciales espacios para presentar futuras propuestas.

En esta investigación se identificaron veinticuatro (24) estudios publicados en conferencias internacionales. La conferencia que presenta mayor frecuencia de publicación, ocho (8) estudios, es la “International Requirements Engineering Conference”. Esta conferencia está clasificada con la categoría A en el ranking de Excelencia en Investigación en Australia (ERA) y con la categoría A1 en el ranking de Qualis del Ministerio de Educación de Brasil. Probablemente su elevada categoría, sea una de las razones que justifican la frecuencia de publicación en dicha conferencia.

En cuanto a publicaciones en revistas científicas fueron identificados nueve (9) artículos. La Fig. 7 señala las revistas más populares y su frecuencia de publicación. De acuerdo a los resultados obtenidos, se observa que la revista “Information and Software Technology” posee tres (3) artículos publicados, mientras que “Requirements Engineering” posee dos (2) artículos publicados en los últimos quince años. Ambas revistas forman parte del cuartil (Q2), escala que ofrece scimagojr¹, para clasificar revistas relevantes.

A continuación, se responden las preguntas de investigación definidas.

Q1. ¿Cuáles son las actividades de la IR respaldadas por las técnicas de aprendizaje supervisado?

Los artículos identificados demuestran que el aprendizaje supervisado, es utilizado en la actualidad por ingenieros e investigadores para obtener nuevas perspectivas y soluciones a problemas que anteriormente se encontraban fuera de su alcance, aprovechando la infraestructura de hardware y una mayor disponibilidad de los datos. El Anexo A presenta un resumen de los estudios seleccionados.

En esta investigación se identificaron métodos y propuestas que apoyan las tareas de obtención y especificación de requerimientos. La obtención de requerimientos es la tarea que determina, a través de la comunicación con los clientes y usuarios, las necesidades y restricciones del producto software a desarrollar.

Los requerimientos se obtienen a través de entrevistas, inspecciones, cuestionarios, entre otras técnicas, a los stakeholders. Es una tarea compleja, porque requiere una comunicación eficiente, a los fines de especificar posteriormente lo recopilado en un documento formal de requerimientos. Es por ello, que los requerimientos deben concebirse sin ambigüedades para definir de manera correcta lo que los stakeholders esperan que haga el sistema.

¹<https://www.scimagojr.com/>

En esta revisión, se identificaron cinco (5) propuestas que abordan los desafíos relacionados con la detección de incertidumbre [28], [29] y ambigüedad [6], [30], [31] en documentos de requerimientos y artefactos escritos en lenguaje natural. Estas propuestas tienen por objeto ayudar a los analistas e ingenieros a identificar los requerimientos propensos a generar malentendidos entre las partes interesadas.

Por otro lado, determinar la coherencia respecto a un documento de requerimientos también es una tarea abordada mediante las técnicas de aprendizaje supervisado [32]. Sólo se identificó un (1) estudio en el área. Esta propuesta utiliza técnicas de procesamiento de lenguaje natural y aprendizaje supervisado para identificar tipos específicos de requerimientos dentro de un conjunto de especificaciones y generar subconjuntos. Esto facilita el análisis de cada subconjunto respecto a aspectos tales como consistencia, integridad y ambigüedad en relación a otros requerimientos. La agrupación de requerimientos en muestras de menor tamaño minimiza la probabilidad de errores durante su análisis.

Otras investigaciones abordan la clasificación automática de los requerimientos escritos en lenguaje natural en requerimientos funcionales (RF) y las diversas subcategorías de los requerimientos no funcionales (RNF). Esto se debe particularmente, al hecho de que los stakeholders, al igual que los ingenieros de requerimientos, usan diversas terminologías y estructuras de oraciones para describir el mismo tipo de requerimiento. Dados estos problemas, es imperativo desarrollar nuevas técnicas y herramientas que puedan respaldar la clasificación de requerimientos. En este trabajo se identificaron ocho (8) propuestas que apuestan a resolver este desafío mediante la aplicación de técnicas de aprendizaje supervisado [33]–[39].

Merten y otros [40] en su propuesta aplican técnicas de aprendizaje supervisado y procesamiento de textos para detectar y clasificar solicitudes de funcionalidades de software presentes en los sistemas de seguimiento y soporte a problemas.

Los documentos de requerimientos, además de contener especificaciones, poseen contenido auxiliar, tales como, resúmenes, explicaciones, referencias a otros documentos, entre otros. Se detectó un (1) estudio que aborda la categorización del contenido de las especificaciones, clasificándolo por requerimiento o información [41], a fines de simplificar el trabajo de ingenieros y analistas de requerimientos.

Otras investigaciones tienen como foco de estudio los enlaces de trazabilidad, fundamentales para el mantenimiento del producto software porque establecen vínculos entre los artefactos de software relacionados, por ejemplo, código fuente, documentación, especificaciones de requerimientos, entre otros, dentro de un sistema. Esto implica que un requerimiento sea trazable desde su definición y durante todo el desarrollo del software, lo cual garantiza una adecuada gestión de cambios y evaluación de impacto en el resto del sistema. La identificación y clasificación de la trazabilidad entre los requerimientos es abordada en seis (6) propuestas [4], [42]–[46].

La identificación de reglas de negocio es una tarea importante del proceso de la IR. Sin embargo, resulta desafiante ya que estas reglas a menudo, no se establecen explícitamente en

los documentos de requerimientos. En caso de que las reglas de negocio sean explícitas, pueden no ser de naturaleza atómica, o pueden ser vagas. Los hallazgos en esta investigación demuestran que el aprendizaje supervisado ha sido empleado para detectar reglas de negocio sobre las especificaciones de requerimientos [47]. Se identificó sólo un (1) estudio que propone una solución a dicho desafío.

Uno de los factores más importantes y desafiantes en el desarrollo de un proyecto de software, es la obtención de requerimientos de calidad. Los requerimientos erróneos no detectados en fases tempranas pueden provocar costos adicionales, insatisfacción de las partes interesadas y retrasos en la entrega del producto software, consecuentemente, esto podría generar la cancelación del proyecto. Por estas razones, la investigación sobre dicha temática resulta clave. En este trabajo se identificaron tres (3) estudios que abordan la clasificación y evaluación de la calidad de requerimientos [48]–[50].

Algunas investigaciones utilizan el aprendizaje supervisado para hacer predicciones respecto a la necesidad de revisión en los documentos de requerimientos [51] con el objetivo de ayudar a los analistas e ingenieros a determinar si una especificación de requerimientos es lo suficientemente estable o necesita revisión adicional. Además, estas técnicas también son utilizadas para predecir el rendimiento operativo del sistema mediante el análisis de la calidad de los requerimientos [52]. Otros aplican el aprendizaje supervisado para predecir fallas en las funcionalidades solicitadas por los stakeholders en los sistemas de gestión de solicitudes de funciones en línea [53], predicción de clases en el código propensas al cambio [54], predicción de riesgos en los requerimientos, con la intención de obtener predictores que indiquen la necesidad de utilizar técnicas para la gestión de riesgos y así mitigar su impacto [55]. Se identificaron cinco (5) propuestas que resuelven desafíos de la IR mediante la generación de modelos predictivos.

Por otro lado, se detectó un enfoque que emplea el aprendizaje supervisado para cuantificar el impacto de los RNF en la estimación del esfuerzo de desarrollo de un proyecto de software [56]. También, se identificó una propuesta que aborda el análisis de requerimientos para extraer automáticamente información semántica [57].

A partir del análisis de los estudios seleccionados, es notable el amplio abanico de aplicaciones de modelos de aprendizaje supervisado en el campo de la IR. Especialmente, se destacan las propuestas destinadas a resolver problemas lingüísticos presentes en los documentos de requerimientos y artefactos escritos en lenguaje natural. Por otro lado, también se distinguen las investigaciones enfocadas en la clasificación del contenido de los documentos de requerimientos.

La revisión comprensiva de los estudios seleccionados demuestra que las técnicas de aprendizaje supervisado en la IR se centran en ocho grandes categorías: (a) Análisis de requerimientos; (b) Identificación de reglas de negocio; (c) Detección de fallas; (d) Estimación del esfuerzo; (e) Resolución de problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural (PDNL); (f) Trazabilidad; y (g) Calidad. La Tabla IV mapea los artículos identificados con las actividades de la IR y su correspondiente categoría asociada.

TABLA IV
ARTÍCULOS POR CATEGORÍA Y ACTIVIDADES DE LA IR

Categoría	ID Artículo	Actividades de la IR
Análisis de Requerimientos	[57]	Análisis de Requerimientos
Reglas de Negocio	[47]	Especificación de Requerimientos
Clasificación de Contenido	[33]–[41]	Análisis de Requerimientos
Predicción de Fallas	[51]–[55]	Validación y Verificación
Estimación de Esfuerzo	[56]	Validación y Verificación
Problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural (PDNL)	[6], [28]–[32]	Validación y Verificación
Trazabilidad	[4], [42]–[46]	Especificación de Requerimientos
Calidad	[48]–[50]	Especificación de Requerimientos

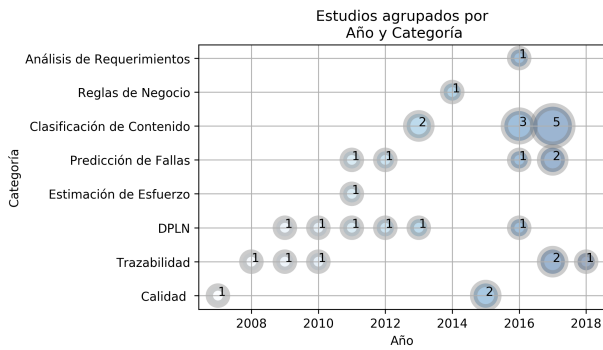


Fig. 8. Distribución de artículos por año y categoría.

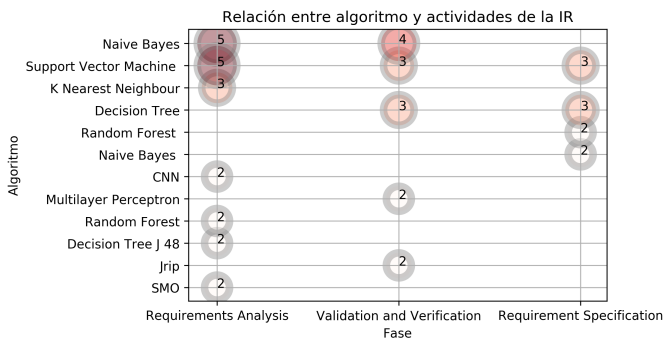


Fig. 9. Relación entre algoritmos de AA y actividades de la IR.

En la Tabla IV, se puede observar que las categorías clasificación de contenido, PDNL y trazabilidad, poseen una mayor cantidad de propuestas que aplican métodos de AA. Consecuentemente, es posible determinar que las actividades de la IR de mayor interés y experiencia respecto a la utilización de dichas técnicas corresponden al análisis de requerimientos, validación y verificación.

La Fig. 8 muestra la distribución de artículos que emplean el aprendizaje supervisado en función a las categorías definidas.

En la Fig. 8 es posible apreciar que a pesar de la libe-

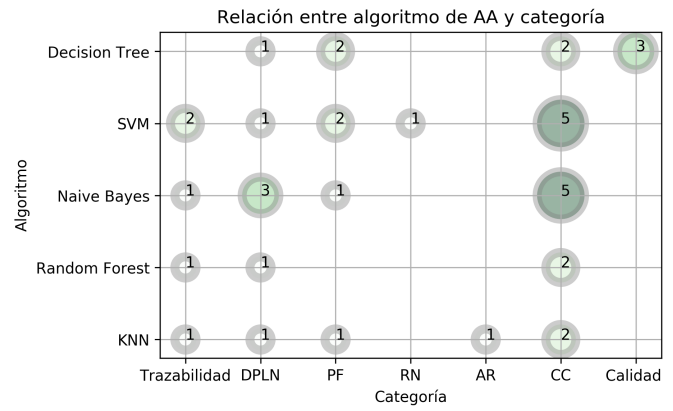


Fig. 10. Relación entre algoritmo de AA y categoría.

eración en el año 2002 de librerías y nuevas tecnologías que permitieron explorar el uso de las técnicas de AA en diversos dominios, recién a partir del año 2006 comenzaron a ser aplicadas en el área de la IR. Las propuestas iniciales, estuvieron orientadas a la resolución de desafíos relacionados a la calidad y trazabilidad de los requerimientos, sin embargo, las investigaciones sobre estas temáticas no mostraron grandes variaciones a largo del tiempo.

Por otro lado, los estudios que abordaron la clasificación de contenidos en los documentos de requerimientos son los que muestran mayor crecimiento entre los años 2016 y 2017. También cabe destacar, que la detección de problemas lingüísticos en documentos y artefactos escritos en lenguaje natural resultó ser uno de los retos más abordados hasta el año 2013.

A fin de proporcionar un mayor entendimiento e identificar tendencias respecto al uso de los métodos de AA, la Fig. 9 presenta la relación entre las actividades de la IR y los algoritmos de AA aplicados en las propuestas identificadas.

En la Fig. 9, es posible apreciar los diez (10) algoritmos de AA más frecuentemente aplicados, en relación, a las actividades de la IR que presentan mayor cantidad de publicaciones. Consecuentemente, se observa que durante la actividad análisis de requerimientos, a menudo se aplican los algoritmos NB, SVM y KNN. Mientras que, en las actividades de validación y verificación de requerimientos las propuestas identificadas hacen uso de los algoritmos NB, SVM y DT. Por último, durante la especificación de requerimientos los algoritmos SVM y DT son los algoritmos aplicados con mayor frecuencia. Por otro lado, se observa que el análisis de requerimientos es la actividad de la IR que presenta una variedad superior de algoritmos aplicados, en relación, a las actividades de validación verificación y especificación de requerimientos.

Q2. ¿Cuáles son los algoritmos de aprendizaje supervisado utilizados para resolver dificultades en las actividades de la IR?

Una amplia variedad de algoritmos de AA fueron detectados durante el análisis de las propuestas halladas en la literatura, éstos son utilizados para automatizar y optimizar las actividades involucradas en la IR. Fueron identificados cincuenta y cinco (55) algoritmos en los estudios seleccionados. El Anexo

A presenta los algoritmos identificados por estudio.

Con el propósito de identificar los algoritmos más utilizados, la Fig. 10 muestra los cinco algoritmos de AA con mayor frecuencia de aplicación, en relación, a las cinco categorías que presentan mayor cantidad de propuestas. Las técnicas más utilizadas corresponden a NB, SVM, DT, KNN y Random Forest.

El algoritmo NB es aplicado en propuestas orientadas a la clasificación de contenido de los documentos de requerimientos (CC), detección y resolución de problemas lingüísticos en documentos y artefactos escritos en lenguaje natural (DPLN). Mientras que es usado con menor frecuencia en tareas relacionadas a la trazabilidad y predicción de fallas (PF). Por lo que podemos concluir que dada la simplicidad que ofrece el algoritmo NB para construir modelos con un buen rendimiento, éste es usado en un amplio abanico de tareas y desafíos de la IR.

Mientras que el algoritmo SVM ha sido utilizado para abordar desafíos relacionados a la clasificación de texto y contenido de los documentos de requerimientos (CC), trazabilidad y predicción de fallas (PF). Posiblemente la selección de dicho algoritmo se deba a su efectividad en la resolución de tareas que involucren la categorización de entidades. También, fueron detectadas investigaciones que analizan la aplicación del algoritmo SVM en propuestas relacionadas a la trazabilidad.

El algoritmo DT es utilizado para clasificar el contenido de los documentos (CC), predicción de fallas (PF) y en actividades relacionadas a la calidad de los requerimientos. Posiblemente la aplicación de este algoritmo se debe a su capacidad para representar y categorizar ciertas condiciones que ocurren en forma sucesiva a fin de resolver un problema determinado. Este algoritmo facilita la interpretación de los datos, dado que provee una visión gráfica de la toma de decisión necesaria, las variables evaluadas, las acciones que deben ser tomadas y el orden en la cual la toma de decisión será efectuada.

El algoritmo KNN ha sido aplicado en actividades relacionadas a la clasificación de contenido (CC), trazabilidad, detección de problemas lingüísticos (DPLN), predicción de fallas (PF) y análisis de requerimientos (AR), lo cual evidencia su amplia flexibilidad y facilidad de aplicación.

Por último, el algoritmo Random Forest es utilizado frecuentemente para resolver problemas relacionados a la clasificación de contenido (CC) y en menor medida en propuestas que abordan la trazabilidad de requerimientos y detección de problemas lingüísticos en documentos de requerimientos y artefactos de software (DPLN).

Resulta clave destacar que los estudios analizados no detallan una justificación respecto a los algoritmos aplicados. Y además, aproximadamente el 62% de los estudios identificados utilizaron más de un algoritmo en sus propuestas de investigación, con la intención de seleccionar aquel que retorna mayor rendimiento. Esto denota la necesidad de un método formal que sirva de guía al proceso de selección de algoritmos de AA para ser aplicados en el área de IR.

Q3. ¿Cuáles son las principales fuentes de datos utilizadas para ejecutar los algoritmos de aprendizaje supervisado?

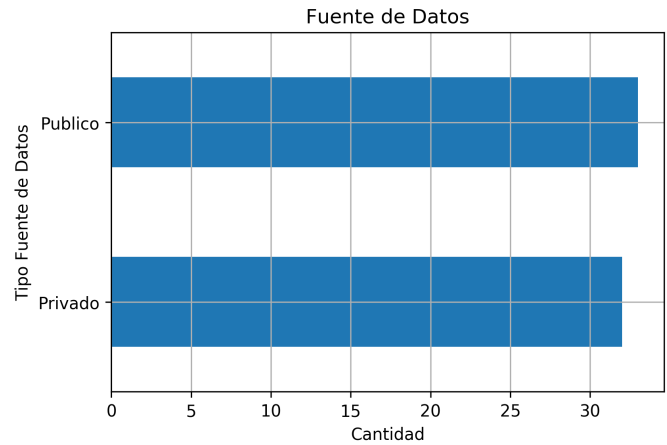


Fig. 11. Distribución de fuentes de datos públicas y privadas.

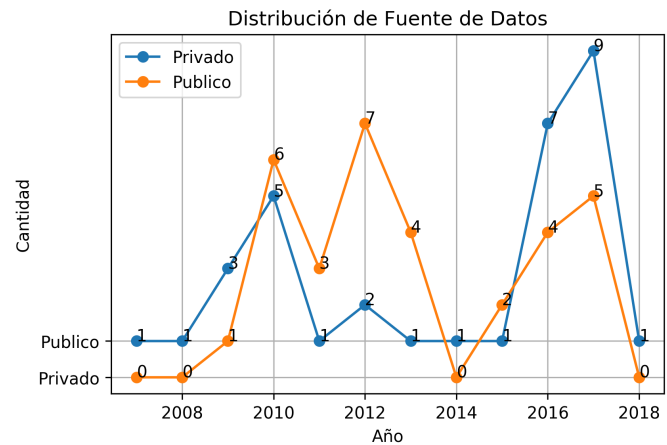


Fig. 12. Distribución de fuentes de datos públicas y privadas.

En los estudios seleccionados, se han detectado diversas fuentes de datos, las cuales se clasificaron como públicas o privadas. Las fuentes de datos públicas están disponibles en repositorios y se distribuyen libremente. Mientras que las fuentes de datos privadas no son compartidas por los investigadores y, consecuentemente, resulta compleja la reproducción de sus propuestas. La Fig. 11 muestra la distribución de fuentes de datos públicas y privadas identificadas.

A partir de los estudios analizados se identificaron veinticinco (25) fuentes de datos públicas y veintiocho (28) privadas. Entre las fuentes de datos públicas utilizadas con mayor frecuencia se encuentra PROMISE (Predictor Models In Software Engineering) utilizado en seis (6) estudios [34], [36]–[38], [53], [54], cuatro (4) estudios utilizaron la fuente de datos iTrust Electronic Health Care System [38], [39], [41], [47] y dos (2) estudios [53], [54] utilizaron el repositorio MDP (Metric Data Program) de la NASA. Cabe señalar que hay algunos estudios que utilizaron más de un conjunto de datos para validar sus propuestas. La Fig. 12 muestra la distribución de las fuentes de datos privadas y públicas a lo largo de los años 2002–2018, considerados en este trabajo.

El empleo de fuentes de datos públicas y privadas en las diversas propuestas de investigación han presentado variaciones

a lo largo de los años. Se observa que en período 2016-2017, han aumentado las propuestas que utilizan conjuntos de datos privados, probablemente consecuencia de la competencia global entre organizaciones. Sin embargo, el uso de conjuntos de datos públicos sigue siendo considerablemente elevado. Durante el período 2010-2012, la tendencia del uso de repositorios públicos ha sido notable. El Anexo A mapea las fuentes de datos utilizadas por estudio.

Como se mencionó en secciones anteriores, el AA utiliza conjuntos de datos para entrenar un modelo y en base a ello hacer predicciones útiles. Es por esta razón, es necesario destacar que el uso de conjuntos de datos públicos permite que las investigaciones propuestas sean repetibles, refutables y verificables [58] proporcionando de esta manera, valor a la comunidad científica para encarar futuras investigaciones. En relación a esto, en la década de 1990, se creó un repositorio llamado UCI ² (University of California Irvine) como un servicio a la comunidad que utiliza el AA, el cual actualmente posee 468 conjuntos de datos.

Inspirados en el esfuerzo de la UCI, posteriormente, los investigadores involucrados en el área de Ingeniería de Software desarrollaron el repositorio PROMISE ³, que cuenta con numerosos conjuntos de datos públicos desde el año 2005. El repositorio PROMISE fue creado para fomentar modelos predictivos repetibles, verificables, refutables y/o mejorables, lo que resulta esencial para la madurez de cualquier disciplina de investigación.

Otra de las fuentes de datos disponibles es MDP de la NASA ⁴, que actualmente consta de 13 conjuntos de datos destinados explícitamente a la investigación de métricas de software. Cada conjunto de datos representa un sistema/subsistema de software de la NASA, contiene las métricas de código y los datos respecto a las fallas de cada módulo que lo compone.

Por otro lado, iTrust ⁵ es un proyecto de software de código abierto, su objetivo es el desarrollo de una aplicación médica que mantiene el registro de los pacientes y su historial médico, además, permitir la comunicación entre médicos. La documentación del proyecto consiste en 59 casos de uso y 11 módulos de código, todos sus artefactos se encuentran disponibles en línea públicamente.

V. ANÁLISIS Y DISCUSIONES DE LOS RESULTADOS

En esta investigación se analizan los estudios que aplican técnicas de aprendizaje supervisado para resolver dificultades en la IR. Como resultado del proceso de análisis, se observó la existencia de artículos que utilizan un conjunto de algoritmos para entrenar sus modelos y luego comparar su rendimiento. Los artículos identificados fueron dieciocho (18). Especialmente, se destaca la propuesta de Nikora y Balcom [32] quienes emplean un conjunto de treinta y un (31) algoritmos utilizando la herramienta WEKA. Por otro lado, se detectó que los algoritmos SVM y NB en combinación con las

técnicas de procesamiento de lenguaje natural son utilizados para resolver problemas relacionados a aspectos lingüísticos en documentos de requerimientos escritos en lenguaje natural y clasificación de contenido. Las ANN fueron utilizadas para eliminar la necesidad de extracción manual de características en los documentos de requerimientos. Para abordar problemas de clasificación fueron utilizados los algoritmos DT, KNN y Regression Logistic (algoritmos clasificadores). Mientras que las técnicas de ensamblado, tales como Bagging y Boosting fueron los algoritmos usados en menor medida debido a que requieren para su entrenamiento grandes conjuntos de datos. En este sentido, Ott [31] destaca la necesidad real de continuar trabajando e impulsando el uso de datos públicos en las investigaciones, a fin de disponer de conjuntos de datos de mayor volumen para el entrenamiento de los algoritmos, pilar fundamental del campo del AA. Entre las actividades de la IR más abordadas se identifican el análisis, especificación, validación y verificación de requerimientos. Consecuentemente, es posible determinar que las actividades relacionadas a la obtención y extracción de requerimientos constituyen una laguna de investigación en el área.

Otro aspecto para destacar es la gran variedad de algoritmos aplicados en las propuestas de investigación, lo que denota la necesidad de un procedimiento o consideraciones formales que facilite el proceso de selección de los mismos para su aplicación en el área de requerimientos. Aún más, no se halló evidencia de una hoja de ruta formalmente documentada que especifique las condiciones necesarias para aplicar las técnicas de AA en el campo de la IR. En relación a las pautas necesarias para la preparación de las fuentes de datos para su posterior procesamiento, tampoco se halló evidencia. Esta situación, demuestra la necesidad de procesos y/o consideraciones formales en dicha área de investigación.

La Fig. 13 ofrece un mapa mental que resume los algoritmos más frecuentemente usados, las tareas de la IR en las que se aplican las técnicas de aprendizaje supervisado y las fuentes de datos públicas más utilizadas en los artículos. Además, las revistas y conferencias más significativas en el área también fueron consideradas.

Durante el desarrollo de esta investigación, no se abordaron ciertas temáticas de interés dada la exhaustividad que requiere su análisis, pero sin dudas valen la pena su estudio y discusión en trabajos posteriores. Este artículo no aborda un análisis comparativo respecto al rendimiento de los algoritmos de AA aplicados en la resolución de un mismo problema o actividad de IR. El flujo o metodología de trabajo utilizado en las propuestas de investigación para aplicar los modelos de AA, no fueron discutidos. Sin embargo, resulta fundamental determinar las metodologías de trabajo utilizadas, a fin de identificar posibles patrones de aplicación. Por último, en este trabajo no es evaluado el impacto y el nivel de aceptación de las industrias que apuestan a la implementación de propuestas que utilizan técnicas de AA.

VI. AMENAZAS A LA VALIDEZ

En esta sección se discuten las posibles amenazas a la validez respecto a los resultados obtenidos durante el proceso

²<https://archive.ics.uci.edu/ml/index.php>

³<http://promise.site.uottawa.ca/SERepository>

⁴<http://mdp.ivv.nasa.gov>

⁵<https://bensmith.s3.amazonaws.com/website/papers/sst2011.pdf>

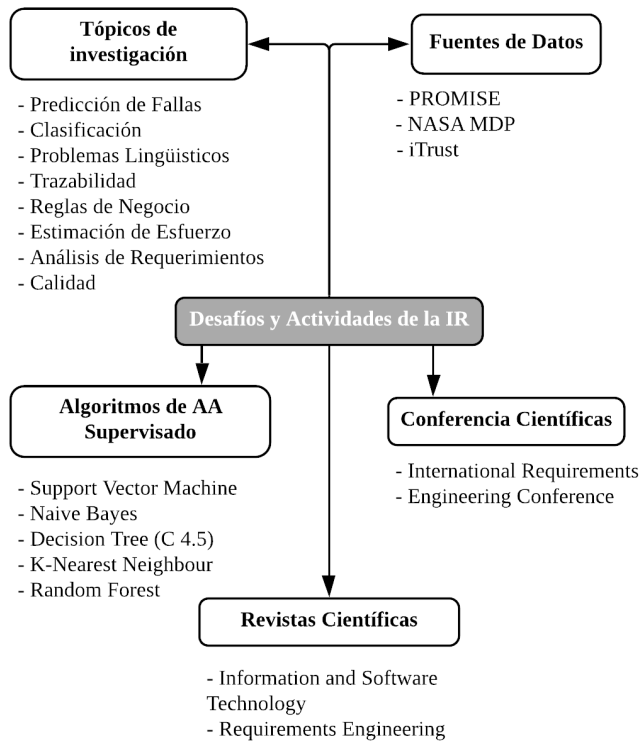


Fig. 13. Mapa mental resultante del mapeo sistemático de la literatura.

de revisión de los artículos seleccionados. El mapeo sistemático propuesto ofrece un estudio de las investigaciones en el área de la IR que aplican técnicas de aprendizaje automático supervisado para resolver diversas tareas y desafíos. A pesar de los esfuerzos de los autores para disminuir los sesgos respecto a la selección de los artículos y resultados obtenidos, existen potenciales amenazas que podrían afectar su validez. Las posibles amenazas identificadas en esta investigación están representadas por el sesgo en la selección de estudios incluidos, la extracción de datos de diferentes fuentes y la síntesis de datos. Para mitigar las amenazas potenciales a la validez, se aplicó un protocolo de investigación definido y reconocido por la comunidad científica para realizar estudios de mapeo. Tal como proponen Petersen y otros [8], este protocolo propone la definición de preguntas de investigación, criterios de inclusión/exclusión y una estrategia de investigación. Además, se analizaron y debatieron los términos incluidos en la cadena de búsqueda de acuerdo al ámbito de interés de los autores en la temática. Posteriormente, se realizaron pruebas piloto de la cadena de búsqueda para comprobar el alcance definido en esta investigación. Durante el proceso de revisión, todos los autores experimentados en el área participaron y debatieron en la toma de decisiones acerca de los documentos incluidos en el estudio.

Otro de los posibles sesgos presente en la investigación está relacionado con la inclusión de artículos redactados sólo en inglés, dado el bajo nivel de difusión de publicaciones y artículos redactados en otros idiomas, en revistas relevantes de Ingeniería de Software. Cabe destacar, que los autores definieron este criterio considerándolo además, como un límite

del alcance de la investigación propuesta. En particular, no se propuso un procedimiento para mitigar este posible sesgo, sino que fue planteado como una futura línea de investigación.

Por otro lado, es necesario considerar que los resultados obtenidos pueden variar en relación a la fecha de ejecución de la cadena de búsqueda y respecto a los accesos a las bibliotecas científicas digitales otorgados a las Universidades e Instituciones de pertenencia de los autores. Consecuentemente, esta situación puede conducir a otro posible sesgo en la investigación dada la no inclusión de algunas bibliotecas relevantes.

VII. CONCLUSIONES

El mapeo sistemático de la literatura propuesto tiene como objetivo identificar y analizar tendencias, conjuntos de datos y métodos de aprendizaje supervisado utilizados para resolver tareas y desafíos presentes en la IR entre los años 2002 y 2018. Considerando los criterios de inclusión y exclusión definidos, se analizaron treinta y tres (33) estudios publicados en revistas y conferencias significativas en el área de Ingeniería de Software. El análisis de los estudios seleccionados reveló que las técnicas de aprendizaje supervisado pueden ser aplicadas para resolver y automatizar tareas relacionadas con la IR, enfocándose principalmente en ocho categorías: detección de problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural, clasificación de contenido de documentos, trazabilidad, estimación del esfuerzo, análisis de requerimientos, predicción de fallas, calidad y detección de reglas de negocio. En esta investigación, se detectaron cincuenta y cinco (55) algoritmos aplicados en las diversas propuestas. Los cinco algoritmos más aplicados son NB, SVM, DT, KNN y Random Forest. En relación a las fuentes de datos utilizadas para entrenar los modelos de AA propuestos, se identificaron veinticinco (25) fuentes de datos públicas y veintiocho (28) privadas. Las fuentes de datos públicas utilizadas con mayor frecuencia fueron PROMISE, iTrust Electronic Health Care System y MDP, aplicados en seis (6), cuatro (4) y dos (2) estudios respectivamente. Es necesario destacar, la presencia de algunos estudios que utilizaron más de un conjunto de datos para validar sus modelos. Los resultados obtenidos en esta investigación permiten evidenciar el valor de los algoritmos de aprendizaje supervisado en el área de la IR. Dado que, tal como los demuestran las propuestas analizadas éstos permiten la optimización y automatización de las actividades involucradas. Aún más, es necesario destacar que no se encontraron trabajos que desestimen la aplicación de las técnicas de AA en la obtención de mejoras y resolución de problemas en dicha área de investigación. Este hecho demanda la necesidad de continuar indagando acerca de sus beneficios y usos posibles. Adicionalmente, se identificaron lagunas de investigación también, específicamente en tareas relacionadas a la obtención y extracción de requerimientos, las cuales son fundamentales para el éxito de un proyecto de software. Consecuentemente, este hecho representa una oportunidad de investigación en el área, por lo que las contribuciones y mejoras que se puedan realizar favorecerán a todo el ciclo de vida del software. Los autores proponen como futura línea

de investigación la definición de un proceso formal que sirva de guía para la elección de algoritmos de AA, de acuerdo al tipo de problema o tarea que se desea resolver en la IR, basado en las experiencias y estudios analizados.

AGRADECIMIENTOS

Los autores agradecen el apoyo brindado por las siguientes instituciones: CONICET y Universidad Tecnológica Nacional (SIUTIFE0004923TC).

REFERENCIAS

- [1] S. Asghar and M. Umar, "Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-Shelf (COTS)," *International Journal of Software Engineering*, vol. 1, no. 2, pp. 32–50, 2010.
- [2] R. Feldt, F. G. de Oliveira Neto, and R. Torkar, "Ways of applying artificial intelligence in software engineering," in *Proceedings of the 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering*, pp. 35–41, ACM, 2018.
- [3] D. L. Giudice and L. L. Humans, "How ai will change software development and applications," 2016.
- [4] Z. Li, M. Chen, L. Huang, V. Ng, and R. Geng, "Tracing requirements in software design," in *ACM International Conference Proceeding Series*, vol. Part F1287, pp. 25–29, 2017.
- [5] E. Giger, M. Pinzger, and H. C. Gall, "Can we predict types of code changes? an empirical analysis," in *Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on*, pp. 217–226, IEEE, 2012.
- [6] H. Yang, A. De Roeck, V. Gervasi, A. Willis, and B. Nuseibeh, "Extending nocuous ambiguity analysis for anaphora in natural language requirements," in *Requirements Engineering Conference (RE), 2010 18th IEEE International*, pp. 25–34, IEEE, 2010.
- [7] H. Felbinger, F. Wotawa, and M. Nica, "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] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, and Petersen, "Systematic Mapping Studies in Software Engineering," *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE'08)*, vol. 8, no. June 2008, pp. 68–77, 2008.
- [9] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.
- [10] T. M. Mitchell, "Machine learning," *Burr Ridge, IL: McGraw Hill*, vol. 45, no. 37, pp. 870–877, 1997.
- [11] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.
- [12] M. E. Celebi and K. Aydin, *Unsupervised Learning Algorithms*. Springer, 2016.
- [13] S. W. Knox, *Machine learning: a concise introduction*, vol. 285. John Wiley & Sons, 2018.
- [14] R. S. Sutton, A. G. Barto, and Others, *Reinforcement learning: An introduction*. MIT press, 1998.
- [15] M. Darnstädt, H. U. Simon, and B. Szörényi, "Supervised learning and co-training," *Theoretical Computer Science*, vol. 519, pp. 68–87, 2014.
- [16] J. Shawe-Taylor and N. Cristianini, "Support vector machines," *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, pp. 93–112, 2000.
- [17] C. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.
- [18] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.
- [19] L. Rokach and O. Maimon, "Decision trees," in *Data mining and knowledge discovery handbook*, pp. 165–192, Springer, 2005.
- [20] G. Williams, "Random forests," in *Data Mining with Rattle and R*, pp. 245–268, Springer, 2011.
- [21] R. H. Thayer, S. C. Bailin, and M. Dorfman, *Software requirements engineering*. IEEE Computer Society Press, 1997.
- [22] K. Pohl, *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.
- [23] T. Ambreen, N. Ikram, M. Usman, and M. Niazi, "Empirical research in requirements engineering: trends and opportunities," *Requirements Engineering*, vol. 23, no. 1, pp. 63–95, 2018.
- [24] K. Matyokurehwa, N. Mavetera, and O. Jokonya, "Requirements engineering techniques: A systematic literature review," *International Journal of Soft Computing and Engineering*, vol. 7, no. 1, pp. 14–20, 2017.
- [25] L. Zamudio, J. A. Aguilar, C. Tripp, and S. Misra, "A requirements engineering techniques review in agile software development methods," in *International Conference on Computational Science and Its Applications*, pp. 683–698, Springer, 2017.
- [26] R. Collobert, S. Bengio, J. Mariétoz, and J. Mariétoz, "Torch: a modular machine learning software library," tech. rep., Idiap, 2002.
- [27] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, p. 38, ACM, 2014.
- [28] H. Yang, A. De Roeck, V. Gervasi, A. Willis, and B. Nuseibeh, "Speculative requirements: Automatic detection of uncertainty in natural language requirements," in *Requirements Engineering Conference (RE), 2012 20th IEEE International*, pp. 11–20, IEEE, 2012.
- [29] A. Knauss, D. Damian, X. Franch, A. Rook, H. A. Müller, and A. Thomo, "Acon: A learning-based approach to deal with uncertainty in contextual requirements at runtime," *Information and Software Technology*, vol. 70, pp. 85–99, 2016.
- [30] H. Yang, A. de Roeck, V. Gervasi, A. Willis, and B. Nuseibeh, "Analysing anaphoric ambiguity in natural language requirements," *Requirements Engineering*, vol. 16, no. 3, pp. 163–169, 2011.
- [31] D. Ott, "Automatic requirement categorization of large natural language specifications at mercedes-benz for review improvements," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 50–64, 2013.
- [32] A. P. Nikora and G. Balcom, "Automated identification of LTL patterns in natural language requirements," in *Software Reliability Engineering, 2009. ISSRE'09. 20th International Symposium on*, pp. 185–194, IEEE, 2009.
- [33] T. Li, "Identifying Security Requirements Based on Linguistic Analysis and Machine Learning," in *Asia-Pacific Software Engineering Conference (APSEC), 2017 24th*, pp. 388–397, IEEE, 2017.
- [34] R. Jindal, R. Malhotra, and A. Jain, "Automated classification of security requirements," in *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*, pp. 2027–2033, 2016.
- [35] Z. Kurtanovic and W. Maalej, "Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning," *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 490–495, 2017.
- [36] A. Dekhtyar and V. Fong, "RE Data Challenge: Requirements Identification with Word2Vec and TensorFlow," in *Requirements Engineering Conference (RE), 2017 IEEE 25th International*, pp. 484–489, IEEE, 2017.
- [37] Z. S. H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, and K. Schneider, "What works better? a study of classifying requirements," in *Requirements Engineering Conference (RE), 2017 IEEE 25th International*, pp. 496–501, IEEE, 2017.
- [38] J. Slankas and L. Williams, "Automated extraction of non-functional requirements in available documentation," *2013 1st International Workshop on Natural Language Analysis in Software Engineering, NaturaliSE 2013 - Proceedings*, pp. 9–16, 2013.
- [39] J. Slankas and L. Williams, "Access control policy extraction from unconstrained natural language text," *Proceedings - Social-Com/PASSAT/BigData/EconCom/BioMedCom 2013*, pp. 435–440, 2013.
- [40] T. Merten, M. Falis, P. Hübner, T. Quirchmayr, S. Bürsner, and B. Paech, "Software Feature Request Detection in Issue Tracking Systems," *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference, RE 2016*, pp. 166–175, 2016.
- [41] J. Winkler and A. Vogelsang, "Automatic classification of requirements based on convolutional neural networks," in *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference Workshops, REW 2016*, pp. 39–45, IEEE, 2016.
- [42] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker, "A machine learning approach for tracing regulatory codes to product specific requirements," *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10*, vol. 1, p. 155, 2010.
- [43] G. Gokyer, S. Cetin, C. Sener, and M. T. Yondem, "Non-functional requirements to architectural concerns: ML and NLP at crossroads," in *Software Engineering Advances, 2008. ICSEA'08. The Third International Conference on*, pp. 400–406, IEEE, 2008.
- [44] C. Mills and S. Haiduc, "A machine learning approach for determining the validity of traceability links," *Proceedings - 2017 IEEE/ACM 39th*

- International Conference on Software Engineering Companion, ICSE-C 2017*, pp. 121–123, 2017.
- [45] A. Sardinha, R. Chitchyan, N. Weston, P. Greenwood, and A. Rashid, “EA-Analyzer: Automating conflict detection in a large set of textual aspect-oriented requirements,” *Automated Software Engineering*, vol. 20, no. 1, pp. 111–135, 2013.
 - [46] Atas M., Samer R., and Felfernig A., “Automated Identification of Type-Specific Dependencies between Requirements,” in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 688–695, IEEE, 2018.
 - [47] R. Sharma, J. Bhatia, and K. K. Biswas, “Automated identification of business rules in requirements documents,” in *Advance Computing Conference (IACC), 2014 IEEE International*, pp. 1442–1447, IEEE, 2014.
 - [48] E. Parra, C. Dimou, J. Llorens, V. Moreno, and A. Fraga, “A methodology for the classification of quality of requirements using machine learning techniques,” *Information and Software Technology*, vol. 67, pp. 180–195, 2015.
 - [49] J. H. Hayes, W. Li, T. Yu, X. Han, M. Hays, and C. Woodson, “Measuring Requirement Quality to Predict Testability,” in *2015 IEEE Second International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pp. 1–8, 2015.
 - [50] I. Hussain, O. Ormandjieva, and L. Kosseim, “Automatic quality assessment of SRS text by means of a decision-tree-based text classifier,” in *Seventh International Conference on Quality Software (QSIC 2007)*, pp. 209–218, 2007.
 - [51] J. del Sagrado and I. M. del Águila, “Stability prediction of the software requirements specification,” *Software Quality Journal*, vol. 26, no. 2, pp. 585–605, 2018.
 - [52] J. L. Dargan, J. S. Wasek, and E. Campos-Nanez, “Systems performance prediction using requirements quality attributes classification,” *Requirements Engineering*, vol. 21, no. 4, pp. 553–572, 2016.
 - [53] C. Fitzgerald, E. Letier, and A. Finkelstein, “Early failure prediction in feature request management systems: an extended study,” *Requirements Engineering*, vol. 17, no. 2, pp. 117–132, 2012.
 - [54] R. Malhotra and M. Khanna, “An exploratory study for software change prediction in object-oriented systems using hybridized techniques,” *Automated Software Engineering*, vol. 24, no. 3, pp. 673–717, 2017.
 - [55] I. M. del Águila and J. Del Sagrado, “Requirement risk level forecast using Bayesian networks classifiers,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 21, no. 02, pp. 167–190, 2011.
 - [56] R. Abdukalykov, I. Hussain, M. Kassab, and O. Ormandjieva, “Quantifying the impact of different non-functional requirements and problem domains on software effort estimation,” in *Proceedings - 2011 9th International Conference on Software Engineering Research, Management and Applications, SERA 2011*, pp. 158–165, 2011.
 - [57] Y. Wang, “Automatic semantic analysis of software requirements through machine learning and ontology approach,” *Journal of Shanghai Jiaotong University (Science)*, vol. 21, no. 6, pp. 692–701, 2016.
 - [58] C. Catal and B. Diri, “A systematic review of software fault prediction studies,” *Expert systems with applications*, vol. 36, no. 4, pp. 7346–7354, 2009.

TABLA V: ANEXO A

ID	Resumen	Categoría	Algoritmo	Fuente de Datos	
[4]	Li y otros proponen un enfoque basado en ontologías para predecir la trazabilidad de los requerimientos y utilizan el algoritmo Support Vector Machine para entrenar los datos.	Trazabilidad	Support Vector Machine	Vector	Pine System
[6]	Yang y otros proponen desarrollar un sistema para clasificar automáticamente las ambigüedades en los requerimientos redactados en lenguaje natural, como nocivos o inofensivos, e informar al analista de los casos potencialmente peligrosos. Desarrollaron un clasificador utilizando técnicas de aprendizaje automático, específicamente Naives Bayes. Los autores destacan la necesidad de un conjunto de datos más grande para mejorar la precisión de la predicción.	Problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural	Naive Bayes		REUTS
[28]	Yang y otros presentan un enfoque para detectar automáticamente la incertidumbre sobre los requerimientos expresados en lenguaje natural. El enfoque aplicado consta de dos etapas. En la primera los autores identifican frases especulativas mediante la aplicación de campos aleatorios condicionales (CRF). Mientras que en la segunda etapa utilizan un conjunto de heurísticas lingüísticas para determinar el nivel de incertidumbre alcanzado en las oraciones analizadas.	Problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural	Conditional Field	Random	REUTS
[29]	Knauss y otros proponen ACon (Adaptation of Contextual Requirements), un proyecto para abordar la incertidumbre que afecta a los requerimientos en tiempo de ejecución. ACon utiliza un bucle de retroalimentación para detectar las necesidades contextuales afectadas por la incertidumbre e integra el algoritmo JRip (un clasificador basado en reglas) para analizar y determinar el contexto en el cual los requisitos son válidos.	Problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural	Jrip		TOTEM

[30]	Yang y otros describen un enfoque para identificar automáticamente ambigüedades potencialmente nocivas, centrándose en la ambigüedad anafórica, que ocurre cuando los lectores no están de acuerdo en cómo deben interpretarse los pronombres en una oración. A través de la definición de heurísticas capturan información acerca de los aspectos que pueden llevar a un lector a una interpretación particular del texto. A partir de estas heurísticas construyen un clasificador utilizando los algoritmos Naives Bayes y K-Nearest Neighbor para predecir las interpretaciones del lector.	Problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural	K- Nearest Neighbor; Naive Bayes	REUTS
[31]	Ott propone un método para detectar defectos de consistencia o integridad, entre los requerimientos de software utilizando algoritmos de clasificación de texto. Se evaluaron dos algoritmos de clasificación, Multinomial Naive Bayes (MNB) y Support Vector Machine (SVM). Para validar la propuesta se utilizaron dos especificaciones automotrices de Mercedes-Benz. El autor destaca la necesidad de continuar trabajando para mejorar la adquisición de datos de entrenamiento (mayor volumen).	Problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural	Multinomial Naive Bayes; Support Vector Machine	Mercedes Benz
[32]	Nikora y Balcom proponen aplicar aprendizaje automático para identificar tipos específicos de requerimientos redactados en lenguaje natural dentro de un conjunto de especificaciones. En este caso, se evaluaron treinta y un (31) algoritmos de clasificación sobre el conjunto de datos de entrenamiento, utilizando la herramienta WEKA. Los resultados en esta investigación demuestran que no hay una diferencia significativa en cuanto al rendimiento de los algoritmos utilizados.	Problemas lingüísticos en documentos de requerimientos y artefactos escritos en lenguaje natural	AD Tree; Bayes Network; Complement Naives Bayes; Conjunctive Rule; Decision Stump; Decision Table; Hyper Pipes; IB1; Ibk ; Decision Tree; Jrip; Kstar, LMT; Simple Logistic; LWL; Multilayer Perceptron; Naive Bayes Multinomial; Naive Bayes Updateable; Naive Bayes, NB Tree; Nnge; OneR; PART; Random Forest; Random Tree; RBF Network; Ridor ; Sequential minimal optimization; VFI; Voted Perceptron; Zero R	Jet Propulsion Laboratory (JPL)

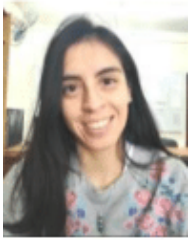
[33]	Li propone un método basado en técnicas de aprendizaje automático y análisis lingüístico para identificar requerimientos de seguridad. Este trabajo utiliza cuatro tipos de algoritmos, incluyendo los basados en Bayes, árboles, funciones y reglas. El autor tomó dos algoritmos de cada uno de estos tipos respectivamente, Naive Bayes (NB), BayesNet (BN), Logistic Model Tree (LMT), J48, Sequential minimal optimization (SMO), Logistic, Decision Table (DT) y PART	Clasificación Contenido	de	Naive Bayes; BayesNet, Logistic Model Tree; J48; Sequential minimal optimization; Logistic; Decision Table; PART	CEPSCO; TISPAN; Global Platform
[34]	Jindal y otros proponen extraer las descripciones de requisitos de seguridad presentes en los documentos de requerimientos (ERS) y desarrollar un modelo de clasificación utilizando el algoritmo árbol de decisión J48. Esta propuesta es evaluada con las especificaciones de requerimientos procedentes de 15 proyectos desarrollados por estudiantes de la Universidad De Paul disponibles públicamente en el repositorio PROMISE.	Clasificación Contenido	de	Decision Tree (J48)	Predictor Models In Software Engineering (PROMISE)
[35]	Kurtanović y Maalej investigan sobre la precisión con la cual es posible clasificar automáticamente los requerimientos como funcionales (RF) y no funcionales (RNF). Para ello, evaluaron los algoritmos Support Vector Machine, Adaptive Boost, Gradient Boosting y Gradient Boosting.	Clasificación Contenido	de	Support Vector Machine; Adaptive Boost; Extra-Trees; Gradient Boosting; Random Forest	Amazon
[36]	Dekhtyar y Fong presentan un enfoque para clasificar los RNF utilizando Word2vec, Convolutional Neuronal Network y Naive Bayes.	Clasificación Contenido	de	Naive Bayes; Word2Vec; Convolutional Neuronal Network	PROMISE; Fachgebiet Software Engineering
[37]	Abad y otros proponen un enfoque de pre-procesamiento que estandariza y normaliza los requerimientos antes de aplicar algoritmos de clasificación. Además, los autores investigan sobre diversos algoritmos (Decision Tree, Latent Dirichlet Allocation, Naive Bayes) para clasificar de manera automática los RNF.	Clasificación Contenido	de	Decision Tree; Latent Dirichlet Allocation; K-means; Naive Bayes	PROMISE
[38]	Slankas y Williams proponen extraer y categorizar los RNF a partir de la documentación disponible (acuerdos de uso, manuales de instalación, reglamentos, especificaciones de requerimientos y manuales de usuario) mediante procesamiento de lenguaje natural y aprendizaje automático. En esta propuesta se utiliza K-Nearest Neighbor, Naive Bayes, Support Vector Machine, Sequential Minimal Optimization (SMO) y para agrupar las oraciones relacionadas utilizaron el algoritmo K-medoids.	Clasificación Contenido	de	K-medoids; K- Nearest Neighbor; Naive Bayes; Support Vector Machine; Sequential minimal optimization	iTrust; Promise; CCHIT

[39]	Slankas y Williams proponen un proceso basado en técnicas de aprendizaje automático para analizar documentos escritos en lenguaje natural. El objetivo es extraer temas, acciones y recursos relevantes para la especificación de una política de control de acceso. El proceso utiliza una combinación de los algoritmos K-Nearest Neighbor, Naive Bayes y Support Vector Machine. También se utiliza el algoritmo de agrupamiento K-medoids para agrupar las oraciones relacionadas.	Clasificación de Contenido	de	K-Nearest Neighbor; Naive Bayes; Support Vector Machine; K-medoids	iTrust
[40]	Merten y otros utilizan técnicas de procesamiento de lenguaje natural y aprendizaje automático para detectar requerimientos en los sistemas de seguimiento de problemas, para lo cual utilizaron las librerías NLTK y Scikit-learn. En esta propuesta se utilizaron siete (7) algoritmos de aprendizaje automático Naive Bayes; Multinomial Naive Bayes; Support Vector Machine; Logistic Regression; SGD (Descenso Gradiente Estocástico); Decision Tree; Random Forest.	Clasificación de Contenido	de	Naive Bayes; Naive Bayes Multinomial, Support Vector Machine; Logistic Regression; SGD (Descenso Gradiente Estocástico); Decision Tree; Random Forest	cgeo; Lighttpd; Radiant; Redmine
[41]	Winkler y Vogelsang presentan un enfoque para clasificar automáticamente el contenido de una especificación de requerimientos como "requerimiento" o "información", utilizando redes neuronales convolutivas.	Clasificación de Contenido	de	Convolutional Neural Network	Mercedes Benz
[42]	Cleland-Huang y otros presentan dos métodos de aprendizaje automático para mejorar la calidad de los enlaces de trazabilidad entre los requerimientos. Los autores utilizan un algoritmo de clasificación propio desarrollado en una investigación anterior	Trazabilidad		Propio	Care2x; CCHIT; Clear Health; Physician; iTrust; Trial Implementation; PatientOS; Practice One; Lauesen; World VistA
[43]	Gokyer y otros proponen un enfoque para extraer automáticamente aspectos arquitectónicos y atributos de calidad desde RNF expresados en texto plano. Para alcanzar dicho objetivo se utiliza el algoritmo Support Vector Machine y técnicas de procesamiento de lenguaje natural.	Trazabilidad		Support Vector Machine	Cybersoft
[44]	Mills y Haiduc proponen un enfoque predictivo que proporciona una clasificación booleana (es decir, válida o inválida) para todos los posibles vínculos o enlaces de trazabilidad entre los conjuntos de artefactos de software. El enfoque utiliza el clasificador de aprendizaje de automático Random Forest implementado en la herramienta Weka.	Trazabilidad		Random Forest	eAnci; eTour; SMOS

[45]	Sardinha y otros presenta EA-Analyzer, una herramienta automatizada para identificar dependencias conflictivas entre requerimientos en los documentos de especificación, utilizando el algoritmo Naive Bayes.	Trazabilidad	Naive Bayes	Health Watcher; Smart Home; CAS
[47]	Sharma y otros presentan un enfoque para la identificación de las reglas de negocio en los documentos de requerimientos. En esta propuesta se utilizaron los algoritmos Support Vector Machine, Naive Bayes, BayesNet y Random Forest.	Reglas de Negocio	Support Vector Machine; Random Forest; Bayes Network; Naive Bayes	Propio
[48]	Parra y otros proponen evaluar automáticamente la calidad de los requerimientos, emulando la evaluación que haría un experto en calidad de un proyecto. Los autores utilizan algoritmos de aprendizaje automático (Decision Tree, PART, Boosting, Bagging, Induction Rules) para generar clasificadores y predecir la calidad de los requisitos.	Calidad	Decision Tree; PART; Boosting, Bagging; Induction Rules	INCOSE
[49]	Hayes y otros proponen utilizar los algoritmos Decision Tree y Logistic Regression para determinar si los requerimientos cumplen con los aspectos de calidad a fin de ser sometidos a las etapas de testing.	Calidad	Regression Logistic; Decision Tree	Browser; iTrust
[50]	Hussain y otros aplican técnicas de aprendizaje automático (Decision Tree) para detectar especificaciones ambiguas en los documentos de requerimientos, considerando indicadores de calidad definidos en un modelo de calidad propuesto por los autores.	Calidad	Decision Tree	Designfest
[51]	Del Sagrado y Águila utilizan Bayesian Network para evaluar las especificaciones de requerimientos y determinar si cumplen con los atributos de calidad o requieren revisión.	Predicción de Fal- las	Bayes Network	RALIC
[52]	Dargan y otros introducen un modelo estadístico utilizado para predecir el rendimiento operativo del sistema a partir de los requerimientos de calidad. Esta propuesta evalúa los algoritmos Logistic Regression, Naive Bayes Classifier, Support Vector Machine y K-Nearest Neighbor para desarrollar un modelo predictivo.	Predicción de Fal- las	Naive Bayes; Support Vector Machine; K-Nearest Neighbor; Logistic Regression	ORD (Operational Requirements Document); KPPs (Key Performance Parameters); Operational Test Report
[53]	Fitzgerald y otros presentan un modelo de predicción de fallas relacionadas a las funcionalidades solicitadas durante la obtención de requerimientos. Esta propuesta emplea los algoritmos Decision Table, Naive Bayes, Linear Regression y M5P-Tree.	Predicción de Fal- las	Naive Bayes; Lineal Regression; Decision Tree	Eclipse; FireFox; KDE; Netbeans; Wikimedia; Thunderbird; Apache

[54]	Malhotra y Khanna sostienen que la volatilidad de los requerimientos, actualización tecnológica y ocurrencia de los defectos requieren cambios en el software para su uso efectivo. Es por ello que proponen un enfoque para la predicción de clases en el código propensas al cambio. Esta propuesta utiliza Decision Tree, Support Vector Machine, CART y Multilayer Perceptron.	Predicción de Fal-las	Decision Tree; Support Vector Machine; CART; Multilayer Perceptron	Ice Cream Sandwich; Jelly Bean
[55]	Águila y Del Sagrado proponen un modelo para predecir la ocurrencia de riesgos a partir de las métricas de requerimientos, utilizando Bayesian Network y Decision Tree.	Predicción de Fal-las	Bayes Network; Decision Tree	PROMISE; MDP (Metrics Data Program)
[56]	Abdukalykov y otros proponen utilizar técnicas de aprendizaje automático para estimar el esfuerzo del proyecto de software. El modelo propuesto se basa en datos históricos de proyectos anteriores y en base a ello, estima el esfuerzo de desarrollo durante la fase de especificación de requerimientos. Se utilizan K Nearest Neighbor y Linear Regression.	Estimación de Es-fuerzo	Artificial Network; Regression	Neuronal Linear ISBSG COSMIC
[57]	Wang propone un enfoque para analizar automáticamente las especificaciones de los requerimientos de software y extraer la información semántica utilizando el algoritmo K-Nearest Neighbour.	Análisis de Requer-imientos	K-Nearest Neighbour	Propio
[35]	Kurtanović y Maalej proponen un enfoque que utiliza metadatos, características léxicas y sintácticas para clasificar automáticamente RF y RNF (usabilidad, seguridad, funcionamiento y rendimiento). Para desarrollar esta propuesta utilizaron el algoritmo Support Vector Machine, alcanzando medidas de precisión (recall y precision) de aproximadamente el 92% para ambas clases de requerimientos. El conjunto de datos utilizado corresponde a una muestra aleatoria de revisiones de software de Amazon (SR) rastreadas entre 2014 y 2016.	Clasificación de Contenido	Support Machine Vector	Amazon

[46]	<p>Atas y otros presentan un enfoque para identificar automáticamente las dependencias entre requerimientos mediante el uso de técnicas de clasificación supervisada. Se evaluaron los algoritmos Naive Bayes, Linear Support Vector Machine, K-Nearest Neighbors y Random Forest. Los resultados de esta propuesta demuestran que el algoritmo Random Forest puede detectar dependencias entre requerimientos con una calidad alta de predicción (aproximadamente 82%). Para evaluar este enfoque se utiliza un conjunto de requerimientos derivados de un proyecto que tiene como objetivo el desarrollo de un reloj deportivo (smartwatch).</p>	Trazabilidad	<p>Naive Bayes; Linear Support Vector Machine; K-Nearest Neighbour; Random Forest</p>	Propio
------	--	--------------	---	--------



María Guadalupe Gramajo , Recibió el título de Ingeniero en Sistemas de Información en la Universidad Tecnológica Nacional, Facultad Regional Tucumán en el año 2015. Actualmente, se desempeña como becario doctoral en el Centro de Investigación y Desarrollo de Ingeniería en Sistemas de Información, Santa Fe, Argentina. Es miembro activo y voluntaria IEEE, Sección Argentina. Sus principales áreas de investigación incluyen Ingeniería de Requerimientos, Aprendizaje Automático y Big Data



Mariel A. Ale Recibió el título de Doctor con mención Ingeniería en Sistemas de Información en la Universidad Tecnológica Nacional, Facultad Regional Santa Fe, en el año 2009. Actualmente, se desempeña como investigador docente en el Centro de Investigación y Desarrollo en Ingeniería en Sistemas de Información, Santa Fe, Argentina. Sus principales áreas de investigación incluyen, Aprendizaje Automático, Interoperabilidad Semántica y Entornos e-learning



Luciana C. Ballejos Recibió el título de Doctor con mención Ingeniería en Sistemas de Información en la Universidad Tecnológica Nacional, Facultad Regional Santa Fe en el año 2009. En la actualidad, se desempeña como investigador docente en el Centro de Investigación y Desarrollo de Ingeniería en Sistemas de Información, Santa Fe, Argentina. Sus áreas de investigación incluyen Ingeniería del Software, Aprendizaje Automático y Cloud Computing.