Received: 9 June 2023 | Revised: 21 February 2024 | Accepted: 29 February 2024

CAAI Transactions on Intelligence Technology

DOI: 10.1049/cit2.12348

ORIGINAL RESEARCH

A systematic mapping to investigate the application of machine learning techniques in requirement engineering activities

Shoaib Hassan[1] | **Qianmu Li**[1] | **Khursheed Aurangzeb**[2] | **Affan Yasin**[3] | **Javed Ali Khan**[4] | **Muhammad Shahid Anwar**[5]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu, China

[2]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

[3]School of Software, Northwestern Polytechnical University, Xian, Shaanxi, China

[4]Department of Computer Science, School of Physics, Engineering & Computer Science, University of Hertfordshire, Hatfield, UK

[5]Department of AI and Software, Gachon University Seongnam-si, Seongnam, South Korea

Correspondence

Qianmu Li.
Email: qianmu@njust.edu.cn

Funding information

Research Center of the College of Computer and Information Sciences, King Saud University, Grant/Award Number: RSPD2024R947; King Saud University

Abstract

Over the past few years, the application and usage of Machine Learning (ML) techniques have increased exponentially due to continuously increasing the size of data and computing capacity. Despite the popularity of ML techniques, only a few research studies have focused on the application of ML especially supervised learning techniques in Requirement Engineering (RE) activities to solve the problems that occur in RE activities. The authors focus on the systematic mapping of past work to investigate those studies that focused on the application of supervised learning techniques in RE activities between the period of 2002–2023. The authors aim to investigate the research trends, main RE activities, ML algorithms, and data sources that were studied during this period. Forty-five research studies were selected based on our exclusion and inclusion criteria. The results show that the scientific community used 57 algorithms. Among those algorithms, researchers mostly used the five following ML algorithms in RE activities: Decision Tree, Support Vector Machine, Naïve Bayes, K-nearest neighbour Classifier, and Random Forest. The results show that researchers used these algorithms in eight major RE activities. Those activities are requirements analysis, failure prediction, effort estimation, quality, traceability, business rules identification, content classification, and detection of problems in requirements written in natural language. Our selected research studies used 32 private and 41 public data sources. The most popular data sources that were detected in selected studies are the Metric Data Programme from NASA, Predictor Models in Software Engineering, and iTrust Electronic Health Care System.

KEYWORDS

data analysis, machine learning, software engineering

1 | INTRODUCTION

Software is an important aspect of all types of organisations and software development is a critical task because it takes cost, effort, and time to build it. Developers and programmers need to deal with challenges during software development [1]. Developers can deal with these situations by integrating new technologies, especially those from Machine Learning (ML) and Artificial Intelligence (AI). These emerging areas of computer science have boosted this field and made a revolution in this field [2]. The integration of new technologies based on AI with the Software Engineering (SE) field automates all those tasks of SE that require so much effort and time. AI has also improved the quality of software. Current studies show better and quality-oriented software development tasks due to the integration of AI techniques with traditional techniques such as rule-based reasoning and Natural language processing (NLP) [3]. In this regard, Requirement Engineering (RE) a sub-area of software engineering that focuses on requirements gathering and requirements specification took great benefit from implementing those emerging techniques. Requirement engineering is a human-based activity that produces many challenges [4]. Different human aspects such as motivation, communication, and domain knowledge impact the RE process [5]. Human-centric practices that we used in RE for ML-based systems produced issues [6]. Machine learning introduces many qualities in requirements including accuracy and fairness [7]. In the literature, some research articles address ambiguity detection in requirement specification documents [8], automatic sorting of requirements [9], code change due to requirement change [10], and software verification and validation [11]. Some articles also address the algorithms, especially supervised-type algorithms that play the main role in resolving the issue in RE. Due to this reason, the main objective of this paper is to explain the systematic mapping, to analyse and investigate the connection and evolution of these two large areas: Supervised ML Techniques and RE.

Our systematic mapping will focus on the literature study in a specific field to provide an overview of that research area. It will also explain those areas that still require attention and define the future of these areas [12]. Our study will give an overview of existing ML applications in the area of RE and how can we use ML techniques to solve issues and challenges that arise in RE. Our study will guide the use of supervised machine-learning techniques in the RE area.

The main objectives of our research work are (1) reviewing a comprehensive study of all observations of applications of supervised learning techniques in the RE field, (2) investigating and analysing all those techniques of supervised learning that help RE tasks, (3) Identifying all the data sources that used in research articles, (4) Identifying all the journals and conferences that highly contribute in the application of ML techniques in RE activities, and (5) investigating gaps and research opportunities in the field of supervised learning applied to RE. The organisation of this article is given as: Section 2 introduces the main concepts relevant to ML techniques and RE. Section 3 describes related work in the area of ML and RE to analyse past research work. Section 4 explains the main methodology for the systematic mapping of supervised learning techniques in RE. Section 5 explains the results and discussions. Section 6 provides the main possible threats to validity. And finally, Section 7 gives the conclusion and main future work.

2 | BASIC CONCEPTS

2.1 | Machine learning

Machine learning has gained popularity in the last decade. Due to a large amount of data and high processing capacity, most organisations use ML in their applications. Machine learning is a sub-area of AI, composed of techniques that allow computers to learn data, make predictions, and make decisions according to those predictions. Machine learning has a variety of algorithms [13]. Sam [14] is a pioneer in ML and explains this concept as a study that allows computers to learn something, for which they have not been programmed. Mitchell [15] also defines that a computer learns from some specific tasks considering the experiences of E type, concerning performance measure P, whether the computer improves performance P, in task T, from experience E. These techniques address computers to check human learning through algorithms and obtain knowledge about a specific domain, thereby possible to increase the performance of some tasks on the new knowledge acquired. ML algorithms can be categorised into four areas, these include (a) supervised learning, (b) unsupervised learning, (c) semi-supervised learning, and (d), reinforcement learning. In supervised learning, algorithms have correctly labelled instances to produce general hypotheses, and allow to make predictions about future instances [16]. In unsupervised learning, algorithms have unlabelled data [17]. In semi-supervised learning, algorithms use unlabelled data to improve supervised learning tasks when labelled data is sparse [18]. In reinforcement learning, computers learn based on the external feedback provided by an external object or environment [19]. In self-supervised learning, the model trains itself by its input rather than depending on the external data provided by the humans.

Figure 1 shows the classification of ML techniques. Our research focuses on all the studies that use supervised learning in the RE field. Supervised learning predicts according to given

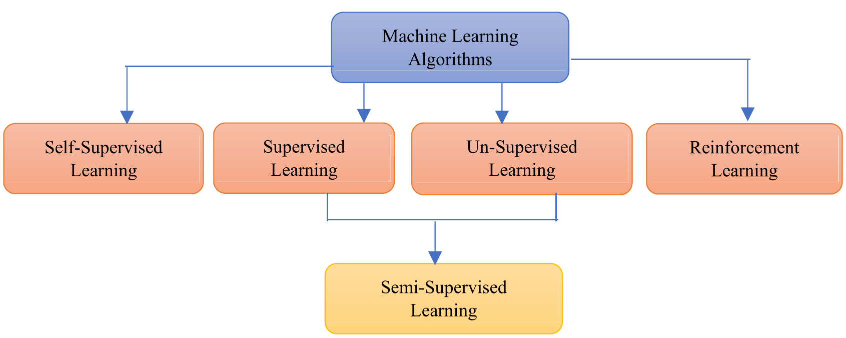

FIGURE 1 Classification of machine learning algorithms.

labelled data which gives us support to develop recommendation systems for different domains.

2.2 | Supervised learning

In the supervised learning technique, we use labelled data which is called training data. From training data, we build a predictive model to predict unlabelled data labels. The training data set contains data that comes from response values. From supervised learning algorithms, we can predict values for new upcoming data. For validation of these algorithms, we use a test data test. If we are using large training data sets, then we can also create models that have advanced predictive ability and gain good results on new upcoming sets [20]. We have a variety of supervised learning algorithms [21].

2.3 | Machine learning algorithms in RE

This section discusses the supervised learning techniques that are mostly used in RE. One of the most common supervised algorithms is Naive Bayes (NB) which is based on the Bayes theorem. This algorithm is based on independence between the predictors according to their labels. Simply, we can say that in NB, we suppose that the occurrence of a characteristic species in a class is not related to another occurrence of another characteristic [22]. Due to this assumption, the NB classifiers estimate the essential parameters for a precise classification using less training data, relative to other classifiers. This ability of NB makes it more attractive for those data sets that contain many predictors. Support Machine Vector (SVM) is another most common supervised learning algorithm. SVM is mainly used for classification or binary regression. SVMs provide a substitute approach to the classification of entities and are mostly used in speech recognition and NLP-based applications. This algorithm sorts the data according to the best hyperplane that splits the input points and maximises the margin between the classes in feature space. Support vectors are the nearest data points separation hyperplane [23].

The K-nearest neighbour (KNN) is the most simple and easy algorithm of supervised learning. It is based on learning by analogy, based on associating a given test individually with training examples relevant to it. The training data are described by n attributes. Each data value represents a point in an n-dimension. The value of K shows the number of points that are considered to classify a test individual point of interest [24]. In KNN, we describe the proximity of K nearest neighbours defined in terms of distance, such as Euclidean distance [25]. The decision tree (DT) algorithm is also one of the supervised learning algorithms that are used to solve regression and classification problems. In a DT, we model data sets in a tree-like structure based on logical decisions. Each node of the tree presents attributes for evaluation and the branches present the decision options for the given attribute. Each leaf on the tree presents a result. In the final result of the DT algorithm, each node represents a condition for the attribute value and each sheet represents the decision for a specific class. When we get a new unknown individual, the tree proceeds to evaluate each conditional unit until it reaches the end of the sheets to label this new unknown value. We can control the complexity of the tree by using pruning methods and stopping criteria. The metrics that we used to measure the complexity of the algorithm are the number of nodes, the number of sheets, and the number of attributes [26]. DT is broadly used for text classification [27]. The random forest (RF) algorithm is also a supervised learning algorithm. It combines a set of decision trees and then trains each set with a set of random observations. We can find the final prediction by averaging the individual predictions of each tree. This algorithm is also used for regression and classification problems [28]. Machine learning is also used to take emotional requirements from different stakeholders [29]. Besides these algorithms, many other techniques are used for RE. Encoding/Decoding and transformers are the most commonly used techniques for semantic understanding of textual requirements. Transformers give weight to words of requirements according to their importance. Transformers can classify the requirements into different categories according to the context of the requirement. Transformers can be helpful to find the dependencies between the requirements. This will be helpful for the requirements traceability. Machine learning helps create the link automatically among requirements, design, code, and testing activities to manage and handle the changes efficiently. The encoder part of the transformer takes the input sequence and converts it into context information. The decoder part again converts the context information into the sequence of words. Transformers with their deep architecture are very helpful for natural language and text information understanding in software requirements.

Deep learning produced a great impact on the software industry [30]. Deep learning is widely used in NLP to identify the true meaning of the content [31]. Deep learning models can also be used in the RE process. Deep learning models can improve the RE process in multiple ways. These are given below:

Requirement Classification: Deep learning models such as convolutional neural networks (CNN) can be used to classify the requirements [32]. Through these models, we can classify the requirements into functional, non-functional, and emotional categories.

Requirements Prioritisation: Requirements prioritisation can also be done through deep learning models. These models are useful to find business values, risks, and priority of requirements.

Image and Voice Identification: If we gathered requirements in voice or video form, then deep learning models can convert audio and video files into textual data.

Understanding the Semantics: Fasttext and word2vec embedding techniques are useful for understanding the semantics and phrases that are hidden in the requirements [33]. These techniques are the most commonly used techniques for word embedding [34]. Deep learning models are also useful for opinion mining [35].

Natural Language Processing: Sentiment analysis is a major research direction of NLP and text mining [36, 37]. Sentiment analysis is a crucial task to identify subjective information from a bulk amount of data [38]. Identifying the required bulk amount of data efficiently is the main responsibility of NLP [39]. Deep learning models like neural networks are used to analyse emails, user stories, and documentation to filter out the requirements [40].

Analysing Traceability: Deep learning models can be used to connect the traceability links between requirements, architecture, and coding to ensure that requirements meet their basic criteria.

Natural Language Generation (NLG): NLG models can be used to generate the requirements content from user input. These models are useful for communicating requirements to stakeholders. Generative AI, represented by models like ChatGPT, can be used to create the requirements automatically by user input. It will make a revolution in the RE field.

Deep learning methods are very different from classical ML models in multiple ways. One of the main differences between deep learning and supervised learning is feature engineering. In deep learning, feature identification is performed as a part of training the model. However, in the supervised learning method, we manually identify the features among the datasets. Supervised learning uses simple models for datasets. Those models have very low complexity. In contrast, deep learning uses complex models with multiple layers. The internal structure of deep learning models is very complex. Deep learning models need high computational power and resources for processing the models on big and complex datasets. While classical ML algorithms require very little computation power and resources.

The understandability and interpretation of ML models are very easy. We can easily explain the supervised learning model methodology. In contrast, deep learning models especially neural networks are very complex and their interpretability is very difficult. Machine learning models work very well for small datasets that have limited relations and features. However, for large datasets, classical supervised algorithms face difficulties. For large datasets, deep learning models work efficiently. Deep learning models work very well in speech recognition, image recognition, and NLP-related tasks. Natural language processing plays a great role in solving RE-related problems. Requirements ambiguity and completeness can be checked by NLP techniques.

Natural language processing is widely used for resolving RE problems. Through NLP techniques, we can identify the pattern and hidden data in requirement documents. NLP is helpful in eliciting requirements from different data sources. Different topic modelling algorithms can identify the topics in requirement documents to show the trend of the requirements. The major problem that NLP solves is requirements classification. In the RE process, requirements classification is a difficult task. NLP techniques can classify the requirements into different categories according to their nature. Requirements prioritisation can also be done with the help of NLP techniques. Regression models can be used to assess the priority of software requirements by using different project factors. BERT and GPT models can be used to produce natural language specifications by giving some inputs. Requirements traceability can also be achieved by NLP techniques. NLP is useful for linking requirement documents and other software artefacts. By integrating NLP techniques in the RE process, we can increase the efficiency and performance of the RE process.

Research studies show that ML is very important in every RE activity. The following Table 1 will explain briefly the application of ML techniques in RE activities.

These are the major fields of ML algorithms that have applications in RE. Machine learning has a variety of algorithms for different tasks of RE. Tokenisation and name entity

TABLE 1 Application of ML techniques in RE activities.

No.	Application of machine learning techniques	RE activities
1	Requirements classification	Requirements management and prioritisation
2	Understanding of natural language	Requirements analysis
3	Anomaly detection	Requirements specification
4	Sentiment analysis	Requirements analysis
5	Requirements traceability	Requirements analysis and specification
6	Automated requirements elicitation	Requirements elicitation
7	Requirements prioritisation	Requirements prioritisation
8	Cross domain learning	Requirements feasibility and domain study
9	Conflict resolution	Requirements analysis
10	Automated document generation	Requirements specification
11	Feedback analysis	Requirements implementation and deployment
12	Recommendation systems	Requirements design and architecture

recognition are the main algorithms of NLP that are used to break the requirements into different parts and understand the semantics. Naïve Bayes, RF, and support vector machine are used to classify the requirements into different categories. K-means and hierarchal clustering algorithms are used for grouping similar requirements based on their characteristics. LDA algorithm is used to select the topics in the requirements. CNN and deep reinforcement learning models are used for image and video analysis in requirements. The selection of a specific algorithm depends upon the nature of the RE task. Machine learning makes the RE process fast and easy.

2.4 | Requirement engineering

RE provides a systematic way to gather the requirements from the clients, analyse those requirements, specify the requirements, validate the requirements, and then manage the requirements that transform into a functional system [41]. The activities of RE include elicitation, analysis, specification, validation, and management of requirements [42]. Due to changes in the trends of the software industry, RE is facing many challenges. Every requirement model has its pros and cons [43]. There are no validated RE techniques for any specific type of problem [44]. Elicitation and requirements specification are two main challenging tasks due to the volatility of requirements [45]. As customers' demands are changing day by day, these two RE tasks face many challenges. If we cannot resolve these RE issues then it may cause failure that requires additional effort and cost for the development team. It is necessary to provide RE education to stakeholders [46].

Due to the challenges of RE, and the emergence of AI techniques, the scientific community has begun to conduct experiments on the application of different AI techniques in RE activities to solve the difficulties and issues that have not yet been resolved. The contributions of AI techniques in RE activities will add improvements to the overall software development life cycle. For this reason, in this investigation, we provide an overview of the literature on how researchers have connected ML algorithms with various activities of RE to automate and optimise those activities.

3 | RELATED WORK

In this section, we will briefly explain some related works, to look at some opportunities for improvements in RE, to find research gaps in previous systematic studies. Table 2 presents the articles and the questions defined. Ambreen et al. [48] present a systematic mapping that defines research questions focused on the identification of proposals with empirical evidence of RE. The mapping study was based on 270 studies that were drawn from four main databases ACM, IEEE, Springer, and ScienceDirect during the period 1990–2012. The results obtained from this study reflect that verification and validation of the requirement activity of RE is not very focused. While the elicitation, analysis, and, management of requirements have a high frequency of publication and are identified as activities of greater interest. On the other hand, the use of ontologies and the application of different techniques of RE in small-sized organisations have begun to be investigated. Finally, the investigation concludes that there is limited interest in comparing existing proposals rather than the scientific community being busy introducing new ones. Only 6% of studies were identified as empirical works.

Matyokurehwa et al. [49] investigated RE techniques that were frequently used in software projects from 2000 to 2016. The main objective of this research is to identify the relationship between RE techniques and possible application strategies in particular cases during software development. It also focuses on RE limitations and how changes affect the analysis. This study analysed 43 techniques that address the Requirements Analysis (RA), but no technique can solve all scenarios accurately. This study also focused on the problems that come with the budget and time estimation of different milestones of the project. This research explains the gaps and the techniques that should address the problems in different RE activities. In a study [50], authors focused on the systematic mapping of supervised learning techniques in RE activities from the 2002–2018 years. This study focused on 5 ML algorithms and three data sources used in this period. In 2021, a group of researchers conducted a study to analyse the usage of RE activities for ML-based systems [51]. They found RE activities useful in ML-based systems. A. Khan discussed non-functional requirements handling in IOT-based ML systems. In his study, he described the effects of poor handling of non-functional requirements in ML-based systems. Non-functional requirements had a great impact on the system's architecture [52]. Pei et al. [53] focused on RE activities in ML applications from a cross-domain perspective. The main focus of this study was to analyse the collaborative RE process. S.Dey defined the impact of uncertainties on RE methods due to ML-based systems. The study discussed all the difficulties of ML-based systems like requirement analysis and decision-making that uses goal-oriented RE [54]. Zamani [55] focused on the systematic mapping of RE activities with ML techniques. It discussed all the literature from 2010 to 2020 related to RE and ML and found the effect of ML techniques in RE activities. In 2021, a group of researchers found the challenges with non-functional requirements for ML-based systems [56]. They focused on the importance of non-functional requirements in ML-based systems. Villamizar et al. [57] discussed the systematic mapping of RE activities with ML techniques. It focused on 35 research articles that discussed the quality properties of ML-based systems like data quality, safety, and transparency.

Zamudio et al. [58] present a review study that addresses the application of traditional RE techniques in agile software development methodologies. The author analysed the SCRUM, Adaptive Software Development, Crystal Family, and Dynamic Systems Development Method. The results of this study allow us to use RE techniques in agile software development without concerning the usage pattern. In the case of extreme programming (XP) and crystal, brainstorming, interviews,

TABLE 2 Related studies of RE.

Studies	Research questions
Empirical research in requirement engineering: Trends and opportunities	RQ1. What is the state of the art in empirical studies of requirement engineering?
	RQ2. What is empirical evidence on RE literature?
Requirement engineering techniques review in Agile software development methods	RQ1. Which agile approaches use traditional techniques of RE in the agile development process?
	RQ2. Which RE techniques are used to get the user requirement for agile software development?
Requirement engineering techniques: A systematic literature review	RQ1. Which RE techniques we are using nowadays?
	RQ2. What are the limitations of the existing RE technique?
	RQ3. How we can change the requirements in the requirement analysis phase?
Agile software requirements engineering challenges-solutions- a conceptual framework from systematic literature review [47]	RQ1. What classification challenges are faced in agile, especially in requirements engineering?
	RQ2. How proposed solutions will address those challenges?

construction of scenarios, and use cases are used for requirements elicitation. While, for RA, Unified Modelling Language is used frequently. The author argues that there is no difference between the use of RE techniques in traditional software development and agile software development methodologies. Furthermore, there are some cases in which traditional RE techniques are applied regularly as requirements change. After the analysis of these studies, we can conclude that RE is a subdiscipline of Software Engineering with research gaps. There is no sufficient technique for requirements classification and analysis automatically [59]. The objective of the analysis of previous work is to obtain past information and identify the point of interest. Although all the research articles agree on the need to enrich and improve this area which is necessary for the success of software projects. We are influenced that RE can find opportunities for improvement and take benefits from the application of new technologies like ML in the RE field.

4 | METHODOLOGY

The methodology that is used for the systematic mapping of literature is based on the guidelines given by Petersen [12], a methodology globally accepted by the scientific community to build systematic mapping in the field of software engineering. The research process has five steps: (a) define research questions, (b) search and identify relevant studies through specific keywords in digital academic libraries, (c) selection of studies using inclusion and Exclusion criteria (EC), (d) review of selected studies by reading abstracts, keywords and major methodology that reflect the core research context, and (e data extraction and analysis). Figure 2 shows the method that is applied in our study. All the following subsections will explain the inputs and outputs of our research methodology.

4.1 | Research questions

Our study analyses all those research articles that apply supervised ML techniques to solve challenges and issues that arise during RE activities. We constructed three research questions to define the scope of our article. Table 3 provides all the research questions and the criteria that motivate their construction. The objective of the first question is to identify all the tasks and RE activities that have been addressed through the use of ML algorithms in multiple academic contributions found in the literature. The main aim is to analyse the trend regarding the application of ML techniques in RE activities and identify those RE issues that have been resolved by ML techniques from various past research contributions. The objective of the second research question is to determine those supervised learning algorithms that resolve challenges and issues of RE activities from past research contributions. The third question aims to identify available data sets in the literature and web. These data sets will facilitate the validation of ML models for software professionals.

4.2 | Search strategy

Studies were searched by referring to four digital online libraries: (i) Springer Link; (ii) Scopus; (iii) IEEEXplore; and (iv) ScienceDirect. These four libraries are very important sources of studies in the area of software engineering. They have repositories of related journals and conferences. This is the main reason to select these libraries. These libraries allow us to search research articles based on different criteria. The terms used to build the search query for research articles are according to the guidelines given by Petersen et al. [12]. These guidelines are the following: (1) Identify the main terms in research questions; (2) Extract documents that are relevant to keywords; (3) Identify and review search terms in titles,

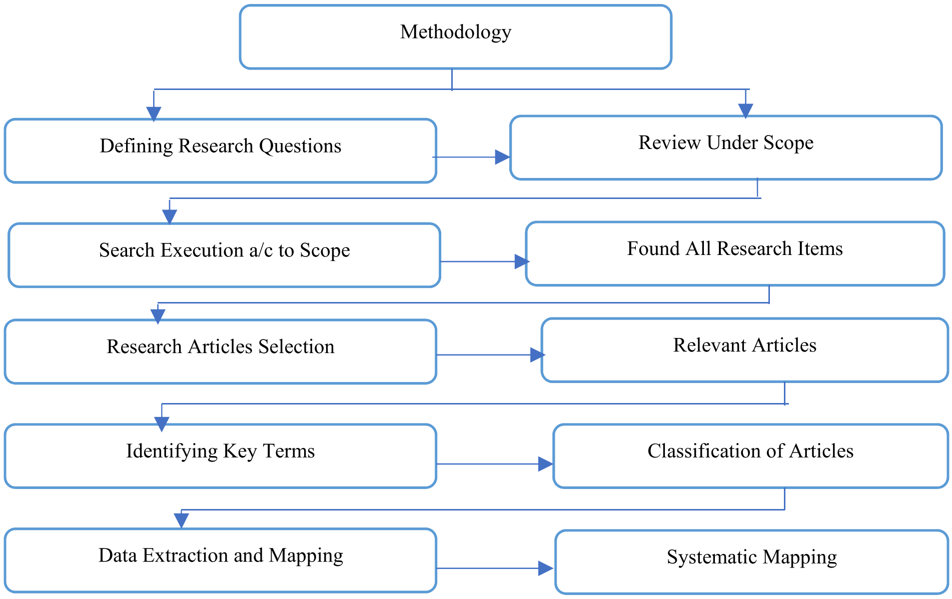

FIGURE 2 Research methodology.

TABLE 3 Research questions.

Research questions	Motivation
Q1. Which RE activities support supervised learning techniques?	M1. Identify all those studies that use supervised learning techniques in RE activities
Q2. Which algorithms are used to resolve the problems in RE activities?	M2. Identify the significant algorithms that are used to resolve the problems in RE activities
Q3. What are the main data sources that are used to run ML models?	M3. Identify all those data sources and repositories that are used for the execution of algorithms

abstracts, and keywords; (4) Identify alternative abbreviations and synonyms; (5) Construct search strings, concatenate the words by Boolean operator 'AND' to link the main term and 'OR' to integrate no alternate anonyms. Table 4 shows the search queries made to libraries. Although this study addresses the analysis of research that applies ML techniques in RE activities. But if we construct a search query without specifying the type of ML technique, then we will get results of all ML approaches (supervised, unsupervised, semi-supervised, and reinforcement). That is why we use the supervised learning approach term to filter our results. The torch library that was introduced in 2002 contains the most common algorithms such as Bayesian classifier, SVM, KNN, and artificial neural networks, among others [60]. Due to this, more research focused on applications of these various algorithms in different domains. For this reason, the studies were limited to the period starting from 2002 until 2023, posturing as a task to investigate research opportunities and gaps in the field of supervised learning applied to RE activities.

4.3 | Inclusion and exclusion criteria

Our research mainly focuses on the analysis of all the literature whose focus is on the application of supervised learning techniques in RE. Our research will focus only on the literature that has been published in the English language. The terms that we used in the search string must be included in the abstract, title, or keywords of a research article. It will help us to identify the articles that focus on the analysis topic. All the peer-reviewed articles that were published from 2002 to 2023 are considered. All the non-peer-reviewed research articles, not written in English articles, duplicate articles, and redundant articles of the same author's documents of type discussion panel or thesis were excluded. All those articles that did not comprise supervised learning techniques were also excluded. Table 5 briefly explains the inclusion and EC to select studies for analysis.

4.4 | Data extraction

We have applied search strings to selected libraries on titles and keywords. We identified two hundred and ninety-five (295) research articles as a result of search string execution. SpringerLink was the repository that gave a maximum number of research articles, finding (180) articles. Scopus gave sixty-three (66) documents, the IEEE Xplore repository gave twenty-eight (42) articles and ScienceDirect gave six (7) articles. Figure 3 summarises all the studies that were retrieved from

TABLE 4 Search queries made to libraries.

No.	Search queries made to libraries	Libraries
1	Machine learning and requirements engineering	• IEEExplore • Springer link • Scopus • Science direct
2	Machine learning and software requirements	
3	Machine learning algorithms and requirements engineering	
4	Machine learning algorithms and software requirements	
5	Supervised learning and requirements engineering	
6	Supervised learning and software requirements	

TABLE 5 Inclusion and exclusion criteria.

Inclusion criteria (IC)	Exclusion criteria (EC)
IC1. Investigating only those studies that focus on the application of supervised learning techniques in RE activities	EC1. Research studies that focus on the application of supervised learning techniques other than RE
IC2. Studies published between the period of 2002–2023	EC2. Research studies that do not exist in this period
IC3. The search string must appear in the abstract, title, and keyword	EC3. Research studies that do not contain search string words in the abstract, title, or keyword
IC4. Studies that were published in English language only	EC4. Research papers that were not written in the English language

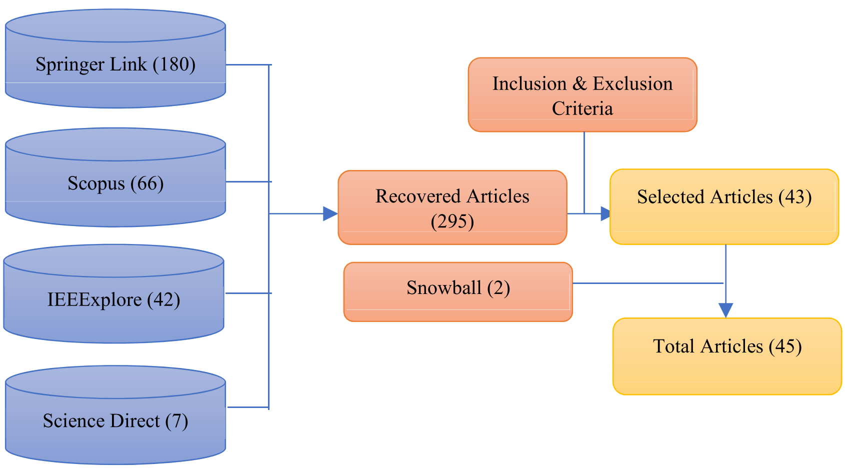

FIGURE 3 No. of research studies identified in repositories.

libraries and selected documents. Each article was analysed according to inclusion criteria. The articles which did not meet IC were excluded from this study. As a result of the cleaning process, we have selected 43 documents that focused on the application of supervised learning techniques in the RE field. The snowball [61] strategy was also applied to all studies to detect research articles that are associated with our research topic. The snowball strategy focused on reviewing those bibliographical references that are cited in these studies to find the relevance of research articles to our research topic. Due to the snowball strategy, two more research articles were added. The total number of research articles related to our research topic is 45.

4.5 | Data synthesis

Our research study analyses 45 research articles. The distribution of published articles according to the time represents the trends regarding applications of supervised learning techniques in the RE field. Figure 5 represents the research trends according to the year of publication. Figure 4 presents that from 2016 to 2018, there was a gradual increase in research articles related to applications of supervised learning algorithms in RE activities. So, ML technology has been a hot topic in the last 3 years. The main reason for the interest is the advancements in the processing power of computers and the emergence of new technologies that enhance the learning of

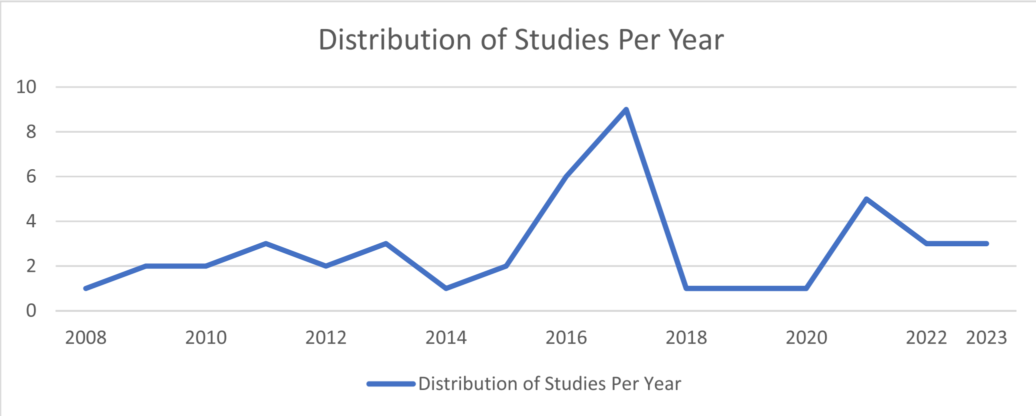

FIGURE 4 Distribution of selected studies by year of publication.

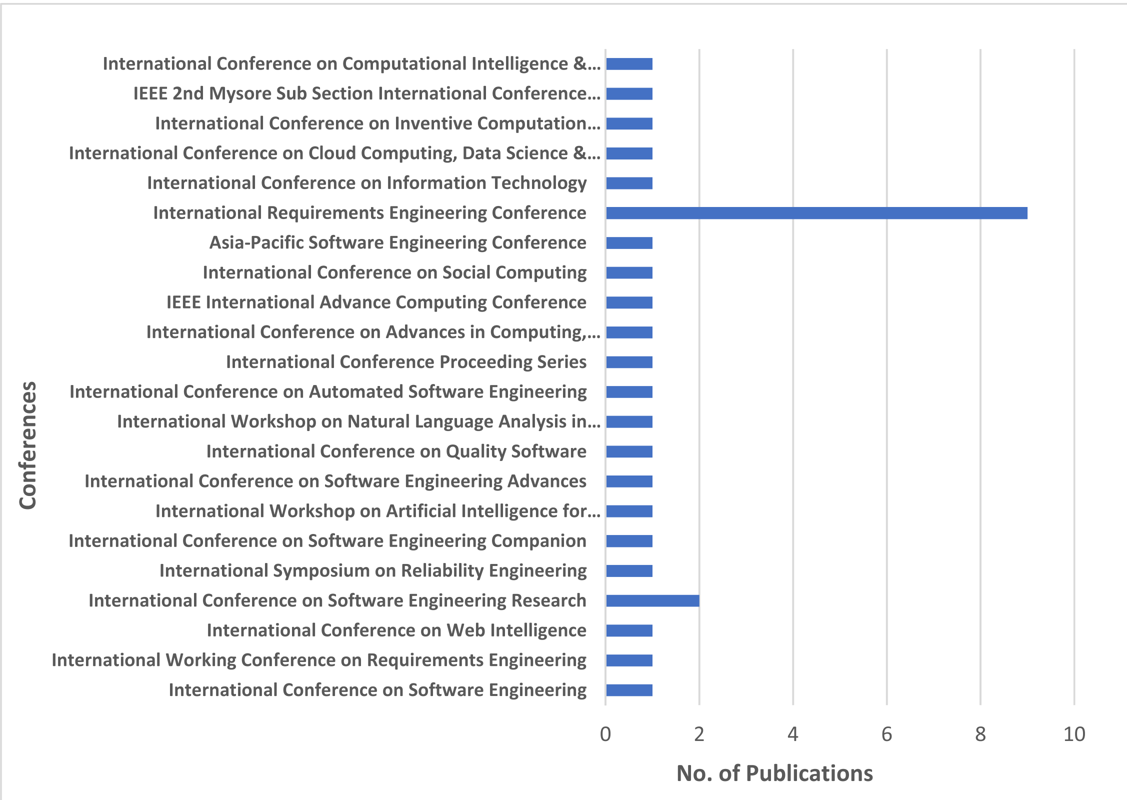

FIGURE 5 Distribution of studies published in conferences.

computers. Another major reason is an acknowledgement of the scientific community regarding ML techniques to automate activities. These reasons have motivated RE professionals to investigate the applications of ML in the RE field. One of the main challenges in RE is to identify a specific ML algorithm to classify requirements in a specific case [62].

Our systematic mapping focused on the scientific techniques that were used by researchers to address their research work. Figure 5 summarises all those conferences and their rate of publication. In this investigation, 31 articles were identified and published at international conferences. The conference which has the highest rate of publication is the International RE Conference. This conference is categorised as an A-rank conference in the field of RE. The publications in scientific journals are 14. Figure 6 represents the most popular journals and their rate of publication. According to the results, the magazine Information and Software Technology has 4 articles, while the RE journal has two articles published in the last 21 years. Both journals are ranked as Q2 category journals, a scale that Scimagojr gives to categorise the journals. Now we will answer the research questions that we have defined previously.

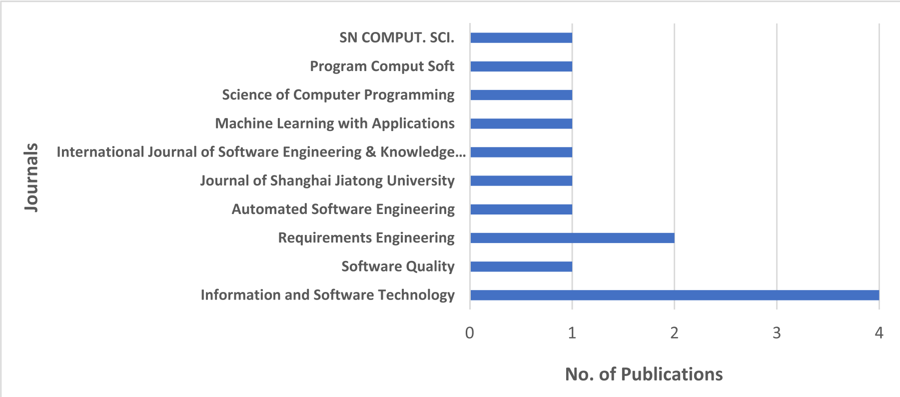

FIGURE 6 Distribution of studies published in journals.

Question No.1 What are the primary RE activities that can be facilitated by supervised learning techniques?

The analysed articles address that supervised learning techniques are widely used by programmers and the scientific community to solve new problems in various domains. Annexure A presents all those selected studies. In our research study, we focus only on those research articles that address the tasks of requirements elicitation, requirements classification, and requirements specification. Requirements elicitation is a task to gather the requirements from customers to obtain the customer needs. The requirements are obtained through different techniques including interviews, surveys, inspections, and questionnaires. It is the most difficult task among all RE tasks because it requires good communication to make requirement documents unambiguous and consistent. R. M. Balajee discussed the requirement identification and elicitation by using ML algorithms during the COVID-19 pandemic time [63]. In our study, five research articles have been identified that address the issues related to ambiguity and uncertainty in requirement documents written in natural language [8, 64–67].

These articles will help requirement engineers identify those requirements that create ambiguity between different stakeholders. One article [68] is identified that address the consistency of requirement documents through supervised learning techniques. This article uses NLP techniques to analyse specific types of requirements within a set of specifications and generate subsets of requirements. The author has analysed those subsets regarding aspects such as integrity, consistency, and ambiguity about other requirements. To group the requirements in small groups, minimise the risks of errors during its analysis. Perez-Verdejo et al. [69] focused on automatic requirements classification through ML techniques. Models were tested by using five open-source software projects at GitHub. Other research addresses the automatic classification of functional requirements written in natural language and categorises non-functional requirements. Requirement engineers use various terminologies and techniques to classify the requirements. It is difficult to develop new techniques that can automate the requirements classification process. In this study, we identified eight research articles that solve these challenges through supervised learning techniques and applications [70–77].

Merten et al. [77] use supervised learning techniques and word processing to identify and classify all the requests for software features that are present in problem support systems. Vogelsang [78] addresses the classification of requirements based on specifications or information. Other research focuses on the traceability to link different software artifacts including source code and documentation. This indicates that the requirement is traceable from its origin throughout the development process, which assures good requirement change management. The classification of traceability between requirements is addressed in eight studies [9, 79–85]. In the RE process, it is very important to define the business rule. These rules are often not defined in the requirements specification document. Sharma [86] addresses that supervised learning techniques have been used to identify the business rules on the requirements specification documents. One of the main challenges to developing a good software project is to obtain good quality requirements. Only one study gave a solution to address this problem. If incorrect and inconsistent requirements are not identified in the early phase of software development then it may cause many problems including cost problems, customer dissatisfaction, and project delays which result in project failure. Due to these issues, research on this topic become a hot area nowadays. In our study, we analysed three articles that focus on the evaluation and classification of quality of requirements [87–89]. Alashqar [90] focused on the classification of non-functional requirements through ML techniques. The Predictor Models in Software Engineering (PROMISE) data set was used to classify non-functional requirements. Manal et al. focused on requirements classification through ML algorithms. It focused on the ML-based HC4RC approach to classify the requirements [91]. In 2023, a group of researchers focused on requirements classification and requirements tracing by ML techniques. Researchers have used zero-shot learning for requirements classification without any training data [92].Quba et al. [93] used ML techniques to classify the requirements. The study used the PROMISE_exp dataset to classify the requirements into functional and non-functional categories. SVM and KNN algorithms are used to classify the requirements. P. Talele discussed the usage of ML algorithms to classify and prioritise software requirements. He focused on 6 ML algorithms that were used to classify and prioritise the requirements [94]. Perez-Verdejo [95] focused on the systematic mapping of ML techniques for requirements classification. The study discussed that NB, decision trees, and NLP algorithms are the most common algorithms used for requirements classification. U. Akshatha Nayak has discussed the requirements classification through ML algorithms. His study focused on requirements definition and description framework [96]. S. Hauser described a method for requirements classification and analysis using ML algorithms. They used a neural network and term frequency algorithm to classify the requirements [97].

The main objective of the study [98] is to help requirement engineers determine requirements specification document stability by using supervised learning techniques for making predictions about changes in specification documents. These techniques will also predict the performance of the system by investigating the requirements quality [99]. Other studies have also used different supervised learning techniques to predict the faults and failures in functional requirements, predict classes in code that need to be changed, predict risks in functional and non-functional requirements of the system, and also predict the impact of those risks on the system [100, 101]. Quba et al. [102], a technique was suggested to check software requirements from documents automatically. In our study, we have identified five research articles that address the RE issues by creating predictive models. Abdukalykov et al. [103] focus on the usage of supervised learning techniques to measure the effect of non-functional requirements on the Effort Estimation (EE) of software projects. Sakhrawi et al. presented a systematic mapping to investigate the usage of ML techniques to predict software project effort. They have done systematic mapping from the period of 1995 to 2020 [104]. A study [105] focuses on the requirement analysis phase to filter the semantic information automatically. By analysing the selected studies, we observed that there is a wide application of supervised learning techniques in the field of RE. Selected studies also focused on the classification of requirements specification document content. Table 6 shows studies from 2007 to 2023. Before 2007, no study was found on the application of ML techniques in RE activities. The frequency of RE activities from 2007 to 2023 is given in the Figure 7 below.

The complete review of selected studies shows that applications of supervised learning techniques in RE focus on eight main areas: (a)Business Rules identification (BR); (b) RA; (c) Failure Prediction (FP); (d) Traceability; (e) Resolving Linguistics issues in requirement document that are specified in natural language (RPNL); (f) EE; (g) Content Classification (CC); (h) Quality.

Table 7 shows all the RE activities and their relevant categories. We can see that most of the research articles have RPNL, CC, and traceability categories that use ML methods. Therefore, we can determine that ML techniques are mostly used in the specification, validation, analysis, and classification phase of the RE life cycle. In Table No. 6, we show all the articles that use supervised learning techniques according to the above categories. Despite the fact of introducing the ML library in 2002, the use of ML in RE began in 2006. The early articles were based on the resolution of issues related to requirements traceability and quality. The studies relevant to CC in requirement documents show the highest growth in the period of 2016–2019. Detecting linguistics problems in software requirements written in natural language was one of the most important challenges till 2013.

Table 8 summarises the relationship between ML techniques and RE activities in selected studies to analyse the trends of usage of ML techniques. In Table 8, we show which ML algorithms are mostly used in RE activities according to the number of publications. Therefore, we observed that during the requirement specification phase, DT and SVM algorithms are mostly used. While, during the RA and classification phase, NB, KNN, and SVM are mostly used and in the requirements validation phase, SVM, DT, and NB are mostly used. According to the above observations, we can say that the RA and classification phase has a variety of supervised learning algorithms as compared to the requirements specification and validation phases.

Question No.2 Which supervised learning algorithms are employed to tackle challenges and issues in RE activities?

By analysing the selected studies, it is observed that there is a wide range of ML algorithms used to automate RE activities. Fifty-seven algorithms were detected during the analysis of studies. Annexure A summarises all the algorithms used in selected studies. Table 9 shows those algorithms that have a high rate of application on the defined categories of RE in selected studies.

The algorithms that are widely used in selected studies are SVM, KNN, DT, RF, and NB. The SVM algorithm is widely used in issues related to FP, traceability, and CC. The selection of the algorithm is based on the impact of the algorithm in the resolution of challenges in RE activities. We have observed that SVM is also used in BR and RPNL. NB is widely used in those studies that have CC and RPNL categories of RE. NB is not widely used in FP and traceability tasks of RE. So, we can say that NB is widely used in resolving the challenges of RE activities. DT algorithm is widely used in FP, CC, and activities that are related to requirements quality. DT algorithm provides a good interpretation of data and it gives a graphical view of the decision the actions that should be taken and the sequence in which the decision should be taken. The RF algorithm is widely used to solve the issues relevant to CC. It is used in a lesser amount in requirements traceability and RPNL in selected studies. Finally, the KNN algorithm is widely used in BR, FP, RPNL, Traceability, and CC-related activities. It is observed that most of the ML algorithms are used in requirements classification. By analysing the selected studies, we conclude that these studies do not give a justification regarding the usage of a specific algorithm. We observed that about 62% of selected studies used more than one algorithm in their

TABLE 6 No. of research studies grouped by year and categories.

	No. of publications per year (2007–2023)																
Categories	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
RA										1			1			1	
BR								1							1		
CC							2			3	5			1	3	2	2
FP					1	1				1	2						
EE					1										1		
RPNL			1	1	1	1	1			1							
Traceability		1	1	1							2	1					
Quality	1								2								

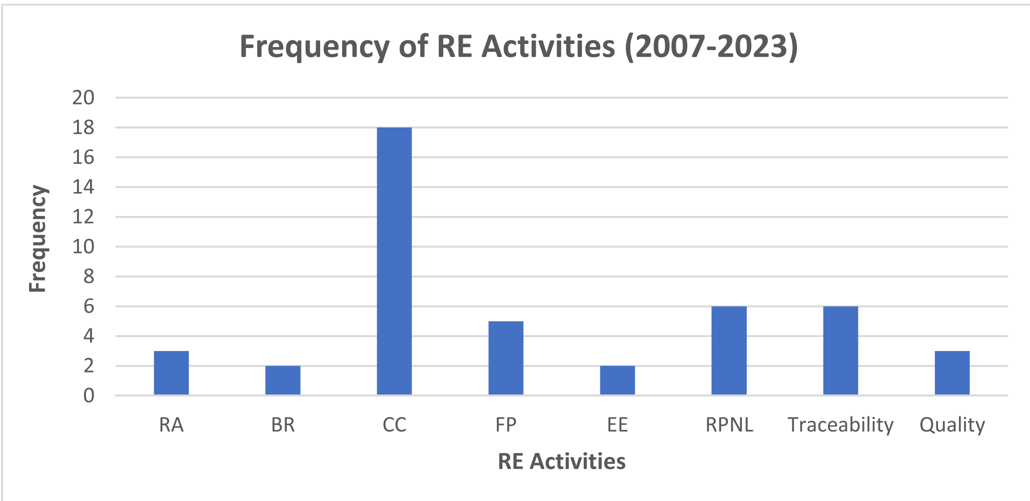

FIGURE 7 Frequency of requirement engineering activities from the year 2007 to 2023.

TABLE 7 Research studies by categories and RE activities.

Categories	Article ID	RE activities
RA	[7, 97, 105]	Requirements analysis
BR	[63, 86]	Requirements specification and identification
CC	[62, 69–78, 90–93, 95, 96]	Requirements analysis and classification
FP	[98–101, 106]	Verification & validation
EE	[103, 104]	Verification & validation
RPNL	[8, 64–68]	Verification & validation
Traceability	[9, 79, 85]	Requirements specification
Quality	[87–89]	Requirements specification

articles. This shows that we should have some proper methods to select the ML algorithms to be applied in the RE field.

Research studies show that ML algorithms are widely used to resolve all the issues of RE. Machine learning techniques especially NLP techniques have resolved the requirements ambiguity problem. Natural language processing techniques can extract useful information from the requirements automatically. Requirements classification become also easy due to ML models. ML models can classify the requirements into different categories in an efficient manner. Through the sentiment analysis technique, we can find the hidden emotions and sentiments in the requirements. ML models especially

TABLE 8 Frequency of the relationship between ML algorithms and RE activities in research studies.

	RE activities		
ML algorithms	**Requirements specification**	**Verification & validation**	**Requirements analysis & classification**
SVM	3	3	7
KNN			6
Decision tree	3	3	2
Random forest	2		3
CNN			5
Multilayer perceptron		2	
Decision tree J48			2
Jrip		2	
SMO			2
Naïve Bays	2	4	7
Term frequency algorithm			2
HC4RC model			1
Zero-shot learning			1

TABLE 9 Frequency of the relationship between most used ML algorithm and RE categories.

	RE categories						
ML algorithms	**Traceability**	**RPNL**	**FP**	**BR**	**RA**	**CC**	**Quality**
Decision tree		1	2			5	3
SVM	2	1	2	1		8	
Naive Bayes	1	3	1			8	
Random forest	1	1				4	
KNN	1	1	1	1	2	5	

unsupervised learning techniques are useful to find anomalies in the requirements. It will be helpful to find inconsistencies in the requirements. ML models can establish the links between requirements, design, and code artifacts. It will be helpful to make traceability links and find errors and mistakes in the requirements. ML models can explain the requirements to requirement engineers in an efficient manner. It will be helpful for requirement engineers to understand the requirements and make the systems successful.

Question No.3 What are the key data resources employed to operate supervised learning algorithms?

By analysing selected studies, we observed multiple data sources classified as public and private data sources. Public data sources are those sources that are easily and freely available in repositories. Private data sources are those sources that are not shared by the scientific community and it is difficult to use that source. Figure 8 shows all the public and private data sources.

By analysing the selected studies, we observed 32 private and 41 public data sources. The most common public data source that is used in selected studies is PROMISE data (PROMISE). This data set is widely used in eight studies [71, 73–75, 90, 93, 100, 106]. Another four studies used iTrust Electronic Health Care System data sources [75, 76, 78, 86]. The Metric Data Program (MDP) data source is used in two studies [100, 106]. Perez-Verdejo et al. [69] used five public data sources placed on GitHub. It is observed that some selected studies used more than one data source to validate their research methodology.

Figure 9 presents all the public and private data sources throughout the period from 2002 to 2023. It is seen that from 2016 to 2023, most of the selected studies used private data sets. The main reason for the increase in private data sets is the increase in competition between organisations. However, the usage of public data sets is also very high. In the period from 2008 to 2012, the usage of public data sets is extraordinary. In Table 11. Annexure A, we show all the data sources used in our selected studies. According to previous sections, we see that ML algorithms use these data sets to train a model and then make useful predictions according to that model. Thus, public data sets allow the scientific community to conduct future research on the applications of ML algorithms in the RE field. University of California Irvine gives a service that uses ML algorithms that have 468 data sets. Software engineering researchers also developed the PROMISE repository, which has had multiple data sets since 2005. Another available data source is MDP which contains 13 data sets that are used in the findings of software metrics. Another data source is the iTrust system. It is a software project that focuses on the development of a medical application that handles patients' data and allows communication between doctors and patients. This project document contains 59 use cases and 11 code modules. All these artifacts are available online.

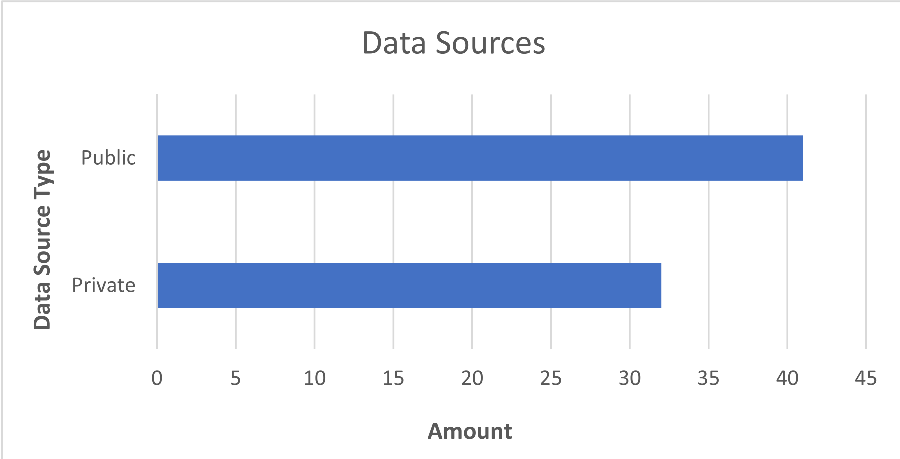

FIGURE 8 Total No. of private and public data sources.

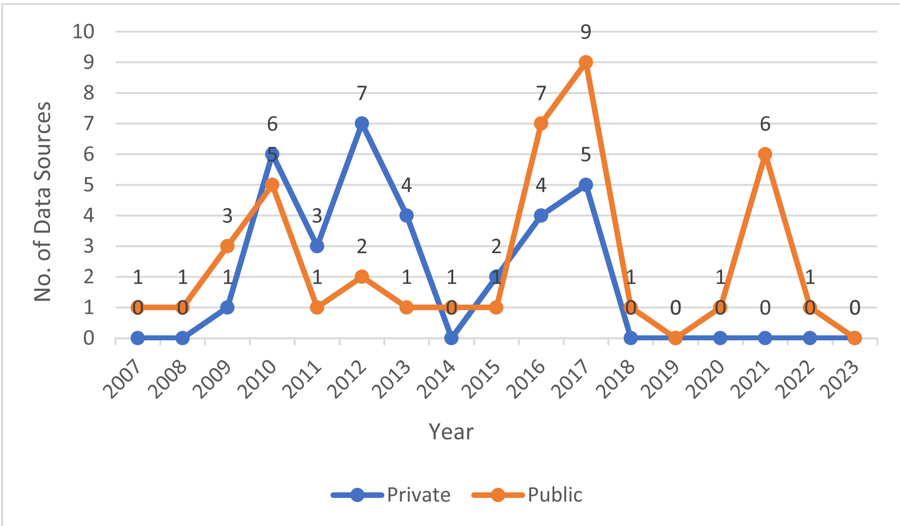

FIGURE 9 Distribution of private and public data sources.

5 | RESULTS ANALYSIS AND DISCUSSION

Our study focuses on research documents that use supervised learning techniques to solve the issues in different RE activities. During observation of selected studies, it is observed that some articles use a set of algorithms to train the models and then compare the performance of all those algorithms. Twenty-two articles have addressed this observation. Khatian et al. [83] used five algorithms to classify non-functional requirements and did their comparative analysis. Atoum et al. [84] used 66 studies to address the different aspects of requirements validation. Diri et al. [107] used 74 research papers relevant to fault prediction in software projects. Balcom [68] used a set of the 30 algorithms through the WEKA tool. Quba et al. [93] used SVM and KNN classifiers to classify software requirements. Pratvina Talele [62] used six different types of ML algorithms to classify and prioritise the requirements. Perez-Verdejo [95] used naïve Bayes, decision trees, and NLP algorithms to classify the requirements. Hauser et al. [97] used neural network and term frequency algorithms to classify the software requirements. We have also analysed that the combination of NB and SVM algorithms with NLP techniques are used to solve the issues that are relevant to linguistics aspects in requirement specification documents written in natural language. We have made a taxonomy tree to summarise all our proposed research in a figure. The following Figure 10 represents a taxonomy tree.

This study also focused on CC. Artificial Neural Networks were used to remove the manual characteristics extraction in requirement specification documents. KNN, DT, and Regression Logistic algorithms were used for the requirements

TABLE 10 Comparative analysis of our systematic study with existing systematic literature studies.

Serial No.	Characteristics	Related existing studies [50–53, 55, 57, 62, 63, 69, 90, 93, 95–97]	Our systematic study
1	Study period coverage	Study [50] covered all the literature studies from the 2002–2018 time period. Study [51] covered all the literature studies till 2020. Study [53] covered all the literature studies till 2021. Study [55] covered all the literature studies from 2010 to 2020. Study [57] covered all the literature studies till 2020.	Our study focused on the period from 2002 to 2023.
2	ML algorithms	Study [50] covered 5 machine learning algorithms. Study [52] focused only on those ML algorithms that are related to IOT. Study [69] focused on 5 classification algorithms. Study [90] focused on 5 ML algorithms. Study [93] only focused on 2 ML algorithms named SVM and KNN. Study [94] focused on 6 ML algorithms. Study [95] focused on 3 ML algorithms named naïve Bayes, decision tree, and NLP algorithms. Study [97] focused on two algorithms named TFA and neural network.	Our study covered 57 ML algorithms. Details of algorithms are given at the end of this article in Table 11. as Annexure A
3	RE activities	Study [51] focused on those RE activities that are connected with machine learning-based AI systems. Study [52] used ML algorithms to handle non-functional requirements only. Study [53] focused on two RE activities named requirement elicitation and requirement design decision. Study [55] discussed two RE activities named traceability analysis and ambiguity detection. Study [44] mainly focused on the quality of software requirements. Study [69] focused on requirements classification and quality characteristics. Study [90] focused on the classification of non-functional requirements only. Study [93] used ML algorithms to classify the requirements only. Study [94] explained two RE activities named requirements identification and requirements prioritisation. Study [95] focused on the classification of requirements only. Study [63] focused on requirement identification and elicitation. Study [96] discussed requirement classification using ML algorithms. Study [97] described a method for requirement classification and analysis.	Our study discussed eight (8) RE activities that used different ML algorithms. The most used RE activity investigated in our research study is requirement classification
4	Data sources	The study [50] identified 25 public and 28 public data sources that used ML algorithms. The study [69] used five open-source data sources at GitHub. Study [90] used the PROMISE data set for requirements classification. Study [93] focused on the PROMISE data set for requirements classification.	Our study discussed 41 public and 32 private data sources that used ML algorithms.

classification issues. Some assembly techniques such as boosting and bagging were also used. However, these techniques were used in lesser amounts because these techniques require large data sets. Ott [67] addresses the need to endorse the use of public data sets to obtain access to large data sets. Many RE activities were addressed in selected studies. However, the most popular RE activities in our selected studies are RA and classification, requirements specification, and requirements validation. Therefore, from our selected studies, we observed the main research gap is to find the best method to obtain and extract requirements. In our selected studies, there is a variety of algorithms. So, there is a high need for a well-structured method to select some specific algorithm for its application in the RE field. There are no specific guidelines to address the conditions in which we should apply ML techniques in the RE field. This demonstrates the need for a structured method in this research field. Figure 11 presents the scheme of study obtained from our systematic mapping of literature. This diagram presents all the algorithms that are widely used in our selected studies, RE activities on which those algorithms were applied, the most frequently used data sources, and the conference and journal names that are most

TABLE 11 Annexure A.

Paper ID	Summary	Category	Algorithm	Data source
[7]	Jennifer et al. proposed a methodology to overview the applications of machine learning techniques for the analysis of non-functional requirements	RA	KNN, naïve Bayes, logistic regression	Own
[9]	Li et al. proposed a method to predict the requirements traceability and use SVM to train the data	Traceability	SVM	Pine system
[8]	Yang et al. proposed a methodology to classify requirements ambiguities automatically written in natural language and use a large data set to improve accuracy	RPNL	Naïve Bayes	REUTS
[50]	Gramajo et al. used systematic mapping from the period of 2002–2018. He proposed that the CC category is mostly used in his systematic mapping	CC	KNN, naïve Bayes, decision tree, SVM, random forest	PROMISE
[53]	The study focused on RE activities in machine learning applications from a cross-domain perspective	RA	A variety of algorithms	No data source
[59]	This study used the PROMISE_exp dataset to classify the requirements into functional and non-functional categories. SVM and KNN algorithms are used to classify the requirements	CC	SVM, KNN	PROMISE
[62]	P. Talele discussed the usage of ML algorithms to classify and prioritise software requirements. He focused on 6 ML algorithms that were used to classify and prioritise the requirements	CC	A combination of 6 algorithms	No data source
[64]	Yang et al. proposed a methodology to detect uncertainty in requirements automatically	RPNL	Conditional random field	REUTS
[65]	Knauss et al. proposed a methodology that addresses the uncertainty affecting the requirements' execution time. They used the Jrip algorithm to determine the situation in which requirements are valid	RPNL	Jrip	TOTEM
[66]	Yang et al. proposed an approach to investigate harmful ambiguities that occur when readers do not know how pronouns in a sentence should be interpreted. They used KNN and naïve Bayes algorithms to identify the reader's interpretation	RPNL	KNN, naïve Bayes	REUTS
[67]	Ott used a text classification algorithm to identify consistency and defects between software requirements. They used large data sets and two algorithms: SVM and multinomial naïve Bayes	RPNL	Multinomial naïve Bayes, SVM	Mercedes benz
[68]	Nikora used the WEKA tool to apply machine learning techniques to identify specific types of requirements in large data of requirements. They used 31 algorithms for classification	RPNL	Decision tree, Bayes network, complement naïve Bayes, conjunctive rule, decision stump, decision table, hyper pipes, IB1, ibk, Jrip, Kstar, LMT, simple logistic, LWL, multilayer perceptron, naïve Bayes multinomial, naïve Bayes updateable, naïve Bayes tree, Nnge, OneR, part, random forest, random tree, RBF network, Ridor, sequential minimal optimisation, VFI, voted perceptron, zero R	Jet propulsion laboratory
[69]	The study focused on automatic requirements classification through machine learning techniques. Models were tested by using five open-source software projects at Github	CC	A combination of 5 classification models	Open-source project data sets
[70]	Li proposed a method to identify security requirements by using machine learning techniques and linguistic analysis. They used four algorithms in their study	CC	Naïve Bayes, Bayesnet, PART, decision table, sequential minimal optimisation, logistic model tree, J48	CEPSCO, global platform, TISPAN

TABLE 11 (Continued)

Paper ID	Summary	Category	Algorithm	Data source
[71]	Jindal and others proposed a method to extract requirements from requirements specification documents and build a classification model based on a decision tree	CC	Decision tree (J48)	PROMISE
[72]	Kurtanovic and others proposed a method to classify functional and non-functional requirements and check their accuracy. They used various algorithms for classification	CC	SVM, random forest, gradient boosting, adaptive boost, extra trees	Amazon
[72]	Kurtanovic and others proposed a method to classify functional and non-functional requirements. They have used lexical and syntactic features to classify quality attributes. They have used various algorithms for this purpose	CC	SVM	Amazon
[73]	Dekhtyar proposed an approach to classify non-functional requirements using CNN, naïve Bayes, and Word2vec	CC	CNN, naïve Bayes, and Word2vec.	PROMISE
[74]	Abad et al. proposed a preprocessing approach for requirements to normalise requirements before applying classification algorithms to them	CC	LDA, decision tree, K-means, naïve Bayes	PROMISE
[75]	Slankas and others proposed a method to extract and categorise non-functional requirements from requirement documents through natural language processing and automatic learning. They have used various algorithms	CC	K-medois, KNN, naïve Bayes, SVM, sequential minimal optimisation	iTrust, PROMISE
[76]	Slankas and others proposed a method to analyse documents written in natural language based on machine learning techniques. They used various algorithms for analysis	CC	SVM, naïve Bayes, KNN, K-medois	iTrust
[77]	Merten and others used NLP and machine learning techniques to identify requirements in problem-tracking systems. They have used various algorithms	CC	Naive Bayes multinomial, SVM, logistic regression, descent Gradient´ stochastic, decision tree, random forest	Cgeo, Lighttpd, radiant redmine
[78]	Winkler and others proposed an approach to classify requirements content using CNN.	CC	Convolutional neural network	Mercedes benz
[79]	Cleland-Huang et al. proposed two machine learning algorithms to improve the quality of traceability between the requirements.	Traceability	Developed own algorithm	iTrust
[80]	Gokyer and others proposed an approach to extract quality attributes from non-functional requirements written in plain text automatically.	Traceability	SVM	Cybersoft
[81]	Mills and others proposed an approach to give a Boolean ranking to all the traceability links between different software artefacts.	Traceability	Random forest	eTour, eAnci,
[82]	Sardinha and others proposed a tool to identify conflicting dependencies between requirements in documents.	Traceability	Naïve Bayes	Health watcher, smart home, CAS
[85]	Atas and others proposed a method to identify dependencies between requirements automatically. They have used several supervised learning algorithms for this purpose	Traceability	Naïve Bayes, SVM, KNN, random forest	Own
[86]	Sharma et al. proposed a method to detect business rules in requirement documents. They used various algorithms to detect business rules.	BR	SVM, random forest, Bayes network, naïve Bayes	Own
[87]	Parra and others proposed a method to evaluate the quality of requirements in a software project automatically. They have used various machine algorithms to classify the quality of requirements	Quality	Decision tree, boosting, bagging, induction rules, PART	INCOSE

(Continues)

TABLE 11 (Continued)

Paper ID	Summary	Category	Algorithm	Data source
[88]	Hayes and others proposed a method to determine the quality of requirements and whether they meet testing criteria or not by using DT and logistic regression algorithms	Quality	Logistic regression, decision tree	Browser, iTrust
[89]	Hussain and others proposed a method to detect ambiguity in documents by using machine learning techniques.	Quality	Decision tree	Designfest
[90]	The study focused on the classification of non-functional requirements through machine learning techniques. This study uses a PROMISE data set to classify non-functional requirements.	CC	A combination of 5 ML classifiers	PROMISE
[91]	The study focused on requirements classification through machine learning algorithms. It focused on ML ML-based HC4RC approach to classify the requirements.	CC	HC4RC algorithm	No data source
[92]	The study focused on requirements classification and requirements tracing by machine learning techniques. Authors have used zero-shot learning (ZSL) for requirements classification without any training data	CC	Zero-shot learning (ZSL)	No data source
[95]	The study focused on the systematic mapping of machine learning techniques for requirements classification. The study discussed that naive Bayes, decision trees, and natural language processing algorithms are the most common algorithms used for requirements classification	CC	Naïve Bayes, decision tree, natural language processing algorithm	Academic databases and collected user reviews
[97]	S. Hauser described a method for requirements classification and analysis using machine learning algorithms. They used a neural network and term frequency algorithm to classify the requirements	CC	Term frequency algorithm, neural network	No data source
[98]	Del Sagrado and others used the Bayesian network to evaluate requirement specification documents to check whether they meet quality criteria or not	FP	Bayes network	RALIC
[99]	Dargan and others used a statistical model to predict the operating performance of the system based on the quality of requirements	FP	SVM, KNN, naïve Bayes, logistic regression	Operational test report, key performance parameters
[100]	Malhotra and others address the problem of requirements volatility. Due change in requirements causes the code change. They have proposed an approach to predict the class in the code that needs to be changed	FP	Decision tree, SVM, CART, multilayer perceptron	Ice cream sandwich, Jelly beans
[101]	Aguila and del Sagrado proposed an approach to predict the risk's occurrence from the requirements metrics	FP	Bayes network, decision tree	PROMISE, MDP
[103]	Abdukalykov and others proposed a method to estimate the effort of a software project using machine learning techniques. They have used the data history of past projects to predict effort	EE	Artificial neural network, linear regression	ISBSG COSMIC
[105]	Wang proposed a method to analyse requirements specification automatically and extract semantic information	RA	KNN	Own
[106]	Fitzgerald et al. proposed a failure prediction model related to functional requirements obtained during requirements elicitation	FP	Naïve Bayes, linear regression, decision tree	ECLIPSE, Firefox, WHERE, Netbeans

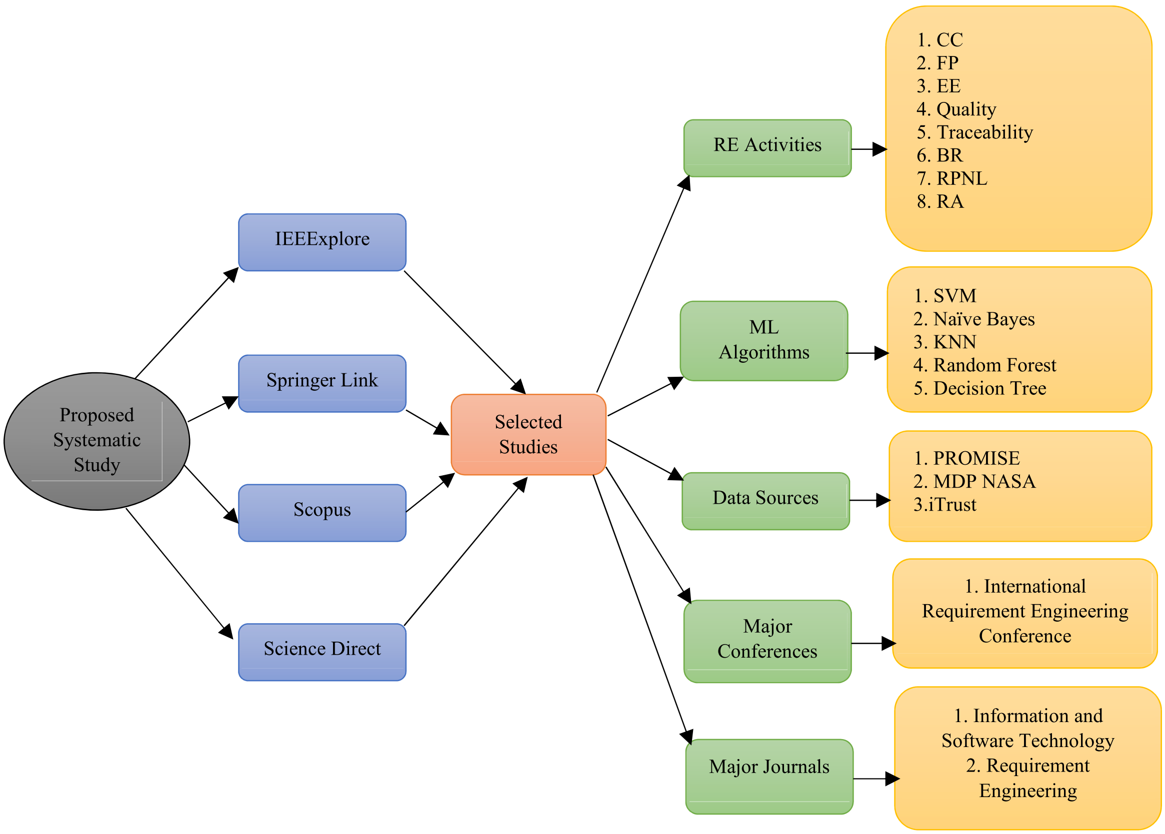

FIGURE 10 Taxonomy tree for summarisation of proposed systematic study.

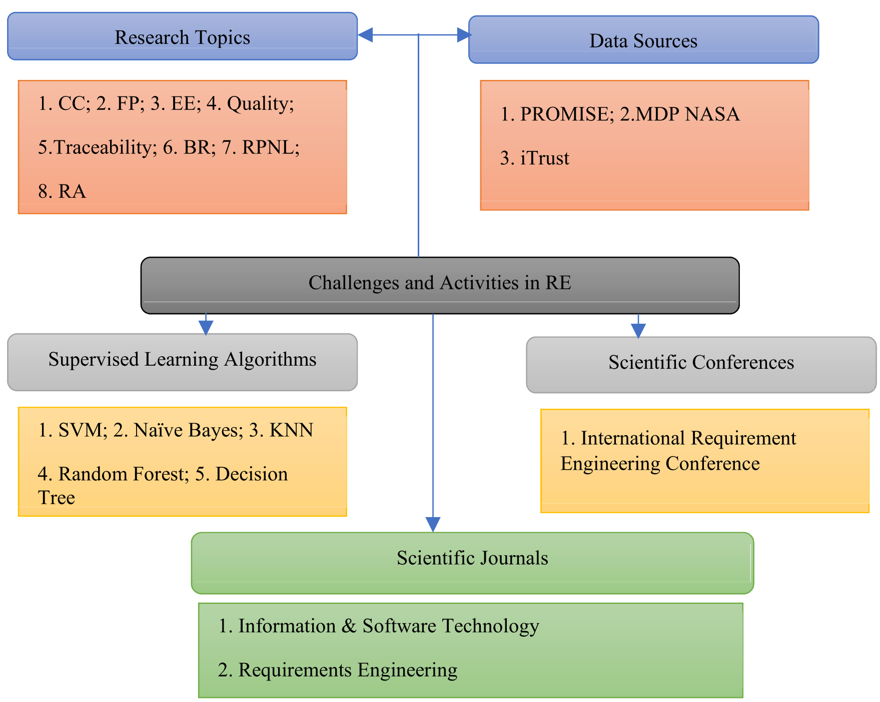

FIGURE 11 Scheme of study obtained from systematic mapping of literature.

significant in this field. The selected articles do not address the comparison of different ML algorithms applied in the resolution of some issues in any RE activity. The working methodology to apply ML techniques in selected studies was not discussed in our systematic study. Therefore, it is necessary to observe the working methodologies that were used in selected studies to investigate the application patterns. Our study focuses on the implementation of ML algorithms in the RE field to help the scientific community make revolutions in this area.

A lot of research has been done on the systematic mapping of ML with RE activities. However, our proposed systematic mapping gives the latest information about the integration of ML with RE activities. We have compared our systematic study with other systematic literature studies and found more contributions as compared to others. A comparative analysis of our systematic study with other systematic literature studies is given in Table 10.

Implementing ML techniques and models introduces many research challenges. Domain adaptability is one of the major challenges. As requirements come from different application domains, it is difficult to decide which ML algorithm will be best suited to which domain. The wrong classification of requirements is also a challenging task. Requirement security and privacy is also one of the major challenges that we are facing when we use ML techniques. We often face issues with requirements labelling. Requirements are labelled through deep learning techniques which require expertise and may be expensive [108].

There are multiple future directions in which ML can be used in different RE activities. One of the major future research directions is automatic requirement elicitation by using ML techniques from emails, interviews, or other media. Another future direction can be the accurate cost, time, and resource prediction of the software requirements. Machine learning models could be used to detect the biasness and fairness in requirements specification documents. Machine learning tools can also be used to check requirements consistency and completeness. This future direction will make a revolution in the RE field by automating different RE tasks using ML algorithms.

6 | THREATS TO VALIDITY

This section explains all the possible threats to validity concerning the results gained through the process of analysis of selected studies. Our systematic study focuses on the applications of ML techniques in the RE field to solve the issues of the RE field. Despite the efforts to minimise the biasness of the author regarding the selection of studies and their results, there are still some threats that could affect the validity of our research. The possible threats can be found in the biasness of the selection of studies, data extraction from different data sources, and data synthesis. To remove possible threats in our study, well-structured and recognised guidelines introduced by the scientific community for systematic mapping were applied in our research work. Petersen [12] defines all the guidelines for the systematic mapping of studies. The study [12] defines research questions, research patterns, and inclusion/EC for the systematic mapping of studies. This guideline also discussed the terms and keywords in the search string according to the researcher's scope of interest in the area.

During the review process of studies, all authors participated in the decision of the selection of documents for our research study. Another biasness in this study is the inclusion of only those articles that were written in the English language only published in relevant journals. The authors defined that this inclusion criterion will limit the scope of our study. We have not mitigated this biasness. We have set this biasness as a future line of research. Another point of consideration is that the results obtained through the search string may change about the execution time of the search string and the time when access to digital repositories was given to the institutes from which authors belonged. Therefore, it can also lead to another biasness which is the non-inclusion of some other repositories that are relevant to our research.

7 | CONCLUSION

The main objective of our research study is to investigate the trends of applications of supervised learning techniques in the RE field in the period 2002 to 2023. According to inclusion and EC, we have selected forty-five (45) research documents from different well-reputed software engineering journals and conferences for analysis. By analysing these research documents, we have observed that supervised learning techniques can be used in RE activities to solve the problems and issues in RE activities. Supervised learning techniques have focused mainly on eight RE activities. These activities are business rule detection, quality, FP, CC, traceability, requirement analysis, EE, and linguistic problem detection in the requirements specification documents.

By analysing the selected studies, it is observed that fifty-seven (57) algorithms have been applied in these studies. The most common and frequent algorithms that were applied in those studies are KNN, SVM, NB, DT, and RF. In our study, 41 public and 32 private data sources were used to train the ML models. The most popular and frequent data sources were PROMISE, iTrust Electronic Health Care System, and MDP which are applied in eight, four, and two studies respectively. Some studies also used more than one data source to validate their study. By analysing selected studies, we see that there is a high value in supervised learning techniques applications in the RE field. Supervised learning techniques automate many RE activities. It is observed that most of the research articles have used ML algorithms to automate the requirement classification activity. Each study supports the usage of supervised learning algorithms in the RE field. Some gaps were also identified by analysing the selected studies. One of the major gaps is in the tasks and activities relevant to gathering and extracting requirements which are the major tasks for the success of any software project. This gap provides us an opportunity to work in this area to give advantages in the overall software engineering life cycle. In the future, we can give a formal and well-structured method to select specific ML algorithms to solve the problems in different RE activities. We can also extend our research by analysing the research studies written in other languages. The research study can also be extended by analysing literature studies that focused on the relation of emotion-based RE with ML.

ACKNOWLEDGEMENTS

This research is funded by researchers supporting project number (RSPD2024R947), King Saud University, Riyadh, Saudia Arabia.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analysed in this study.

ORCID

Shoaib Hassan https://orcid.org/0000-0002-8265-7932
Affan Yasin https://orcid.org/0000-0002-0166-2239

REFERENCES

1. Asghar, S., Umar, M.: Requirement engineering challenges in development of software applications and selection of customer-off the-shelf (COTS). Int. J. Software Eng. 1(2), 32–50 (2010)
2. Feldt, R., de Oliveira Neto, F.G., Torkar, R.: Ways of applying artificial intelligence in software engineering. In: 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, pp. 35–41. ACM (2018)
3. Giudice, D.L., et al.: How Ai Will Change Software Development and Applications (2016)
4. Madampe, K., Hoda, R., Grundy, J.: The emotional roller coaster of responding to requirements changes in software engineering. IEEE Trans. Software Eng. 49(3), 1171–1187 (2023). https://doi.org/10.1109/tse.2022.3172925
5. Hidellaarachchi, D., et al.: The influence of human aspects on requirements engineering-related activities: software practitioners' perspective. ACM Trans. Software Eng. Methodol. 32(5), 1–37 (2023). https://doi.org/10.1145/3546943
6. Ahmad, K., et al.: Requirements practices and gaps when engineering human-centered Artificial Intelligence systems. Appl. Soft Comput. 143, 110421 (2023). https://doi.org/10.1016/j.asoc.2023.110421
7. Horkoff, J.: Non-functional requirements for machine learning: challenges and new directions. In: IEEE 27th International Requirement Engineering Conference (2019)
8. Yang, H.: Extending nocuous ambiguity analysis for anaphora in natural language requirements. In: IEEE International Requirements Engineering Conference (RE) (2010)
9. Li, Z., et al.: Tracing requirements in software design. ACM Int. Conf. Proc. F1287, 25–29 (2017). https://doi.org/10.1145/3084100.3084102
10. Giger, E., et al.: Can we predict types of code changes? an empirical analysis. In: Mining Software Repositories (MSR), 9th IEEE Working Conference on, pp. 217–226 (2012)
11. Felbinger, H., et al.: Test-suite reduction does not necessarily require executing the Program under test. In: Proceedings of 2016 IEEE International Conference on Software Quality, Reliability and Security-Companion, QRS-C 2016 (2016)
12. Petersen, K., et al.: Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE'08), vol. 8 (2008). https://doi.org/10.14236/ewic/ease2008.8
13. Biswas, S., Rajan, H.: Fair preprocessing: towards understanding compositional fairness of data transformers in machine learning pipeline. In: 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, USA (2021)
14. Samuel, L.: Some studies in machine learning using the game of checkers. IBM J. Res. Dev. 3(3), 210–229 (1959). https://doi.org/10.1147/rd.33.0210
15. Mitchell, T.M. Machine Learning,Burr Ridge: McGraw Hill, 45(37), 870–877 (1997)
16. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised Machine Learning: A Review of Classification Techniques (2007)
17. Celebi, M.E., Aydin, K.: Unsupervised Learning Algorithms. Springer (2016)
18. Knox, S.W.: Machine Learning: A Concise Introduction. John Wiley & Sons (2018)
19. Barto, A.G., Sutton, R.S.: Reinforcement Learning: An Introduction. MIT press (1998)
20. Darnst¨adt, M., Simon, H.U., Szörényi, B.: Supervised learning and co-training. Theor. Comput. Sci. 519, 68–87 (2014). https://doi.org/10.1016/j.tcs.2013.09.020
21. Onan, A.: Consensus Clustering-Based Undersampling Approach to Imbalanced Learning. Scientific Programming, Hindawi (2019)
22. Manning, C.: Introduction to information retrieval. Nat. Lang. Eng. 16(1), 100–103 (2010). https://doi.org/10.1017/s1351324909005129
23. Cristianini, N.: Support vector machines. In: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, pp. 93–112 (2000)
24. Onan, A.: An ensemble scheme based on language function analysis and feature engineering for text genre classification. J. Inf. Sci. 44(1), 28–47 (2018). https://doi.org/10.1177/0165551516677911
25. Alpaydin, E.: Introduction to Machine Learning. MIT press (2014)
26. Maimon, O.: Decision trees. In: Data Mining and Knowledge Discovery Handbook, pp. 165–192. Springer (2005)
27. Onan, A., Korukoğlu, S., Bulut, H.: Ensemble of keyword extraction methods and classifiers in text classification. Expert Syst. Appl. 57, 232–247 (2016). https://doi.org/10.1016/j.eswa.2016.03.045
28. Williams, G.: Random forests. In: Data Mining with Rattle and R, pp. 245–268. Springer (2011)
29. Mia Mohammad, I., et al.: Data augmentation for improving emotion recognition in software engineering communication. In: 37th IEEE/ACM International Conference on Automated Software Engineering, USA (2023)
30. Onan, A.: Hierarchical graph-based text classification framework with contextual node embedding and BERT-based dynamic fusion. J. King Saud Univ.—Comput. Inf. Sci. 35(7), 101610 (2023). https://doi.org/10.1016/j.jksuci.2023.101610
31. Onan, A.: Mining opinions from instructor evaluation reviews: a deep learning approach. Comput. Appl. Eng. Educ. 28(1), 117–138 (2020). https://doi.org/10.1002/cae.22179
32. Onan, A.: Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. . King Saud Univ.—Comput. Inf. Sci. 34(5), 2098–2117 (2022). https://doi.org/10.1016/j.jksuci.2022.02.025
33. Onan, A.: Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering. IEEE Access 7, 145614–145633 (2019). https://doi.org/10.1109/access.2019.2945911
34. Onan, A.: Topic-enriched word embeddings for sarcasm identification. In: Computer Science On-Line Conference (2019)
35. Onan, A.: Sentiment analysis on massive open online course evaluations: a text mining and deep learning approach. Comput. Appl. Eng. Educ. 29(3), 572–589 (2021). https://doi.org/10.1002/cae.22253

36. Onan, A., Korukoğlu, S.: A feature selection model based on genetic rank aggregation for text sentiment classification. J. Inf. Sci. 43(1), 25–38 (2016). https://doi.org/10.1177/0165551515613226
37. Onan, A.: Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. Concurrency Comput. Pract. Ex. 33(23) (2020). https://doi.org/10.1002/cpe.5909
38. Onan, A., Korukoğlu, S., Bulut, H.: A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification. Inf. Process. Manag. 53(4), 814–833 (2017). https://doi.org/10.1016/j.ipm.2017.02.008
39. Onan, A., Tocoglu, M.A.: A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification. IEEE Access 9, 7701–7722 (2021). https://doi.org/10.1109/access.2021.3049734
40. Onan, A.: GTR-GA: harnessing the power of graph-based neural networks and genetic algorithms for text augmentation. Expert Syst. Appl. 232, 120908 (2023). https://doi.org/10.1016/j.eswa.2023.120908
41. Thayer, R.H.: Software Requirements Engineering. IEEE Computer Society Press (1997)
42. Pohl, K.: Requirements Engineering: Fundamentals, Principles, and Techniques. Springer Publishing Company, Incorporated (2010)
43. Bao, T., et al.: RM2Doc: a tool for automatic generation of requirements documents from requirements models. In: ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, USA (2022)
44. Villamizar, H.: Requirements engineering for machine learning: a systematic mapping study. In: 47th Euromicro Conference on Software Engineering and Advance Applications (SEAA) (2021)
45. Fatima, R., et al.: Improving software requirements reasoning by novices: a story-based approach. IET Softw. 13(6), 564–574 (2019). https://doi.org/10.1049/iet-sen.2018.5379
46. Yasin, A., et al.: Design and preliminary evaluation of a cyber security requirements education game (SREG). Inf. Software Technol. 95, 179–200 (2018). https://doi.org/10.1016/j.infsof.2017.12.002
47. Hoy, Z.: Agile software requirements engineering challenges-solutions—a conceptual framework from systematic literature review. Information, MDPI 14(6) (2023)
48. Ambreen, T., et al.: Empirical research in requirements engineering: trends and opportunities. Requir. Eng. 23(23), 63–95 (2018)
49. Matyokurehwa, K., Mavetera, N., Jokonya, O.: Requirements engineering techniques: a systematic literature review. Int. J. Soft Comput. Eng., 7(1), 14–20 (2017). https://doi.org/10.23956/ijarcsse/v6i9/01917
50. Gramajo, M., Ballejos, L., Ale, M: Seizing requirements engineering issues through supervised learning techniques. IEEE Lat. Am. Trans., 18(07), 1164–1184 (2020), https://doi.org/10.1109/tla.2020.9099757
51. Yoshioka, N., et al.: Landscape of requirements engineering for machine learning-based AI systems. In: 28th Asia-Pacific Software Engineering Conference Workshops (APSEC Workshops), Taiwan (2021)
52. Khan, A., et al.: Handling non-fuctional requirements in IoT-based machine learning. In: Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer (2022)
53. Pei, Z., et al.: Requirements engineering for machine learning: a review and reflection. In: 2022 IEEE 30th International Requirements Engineering Conference Workshops (REW), Melbourne, Australia (2022)
54. Dey, S.: Evidence-driven data requirements engineering and data uncertainty assessment of machine learning-based safety-critical systems. In: 2022 IEEE 30th International Requirements Engineering Conference (RE), Melbourne, Australia (2022)
55. Zamani, K.: Machine learning in requirements engineering: a mapping study. In: 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), Notre Dame, IN, USA (2021)
56. Horkoff, J.: Non-functional requirements for machine learning: understanding current use and challenges in industry. In: 2021 IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA (2021)
57. Villamizar, H.: Requirements engineering for machine learning: a systematic mapping study. In: 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Palermo, Italy (2021)
58. Zamudio, L., et al.: A requirements engineering techniques review in agile software development methods. In: International Conference on Computational Science and its Applications, pp. 683–698. Springer (2017)
59. Quba, G.Y., et al.: Software requirements classification using machine learning algorithm's. In: International Conference on Information Technology (2021)
60. Collobert, R., et al.: Torch: A Modular Machine Learning Software Library. tech. rep., Idiap (2002)
61. Wohlin, C.: 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, pp. 38. ACM (2014)
62. Pratvina Talele, R.P.: Classification and prioritization of software requirements using machine learning- A systematic review. In: 11th International Conference on Cloud Computing, Data Science and Engineering (2021)
63. Balajee, R.M., et al.: Requirements identification on automated medical Care with appropriate machine learning techniques. In: 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India (2021)
64. Yang, H.: Speculative requirements: automatic detection of uncertainty in natural language requirements. In: 20th IEEE International Requirements Engineering Conference (RE) (2012)
65. Knauss, A., et al.: Acon: a learning-based approach to deal with uncertainty in contextual requirements at runtime. Inf. Software Technol. 70, 85–99 (2016). https://doi.org/10.1016/j.infsof.2015.10.001
66. Yang, H., et al.: Analyzing anaphoric ambiguity in natural language requirements. Requir. Eng. 16(3), 163–169 (2011)
67. Ott, D.: 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 (2013)
68. Balcom, G.: Automated identification of LTL patterns in natural language requirements. In: 20th International Symposium on Software Reliability Engineering, ISSRE'09 (2009)
69. Pérez-Verdejo, J.M., et al.: Requirements and GitHub issues: an automated approach for quality requirements classification. Program. Comput. Software 47(8), 704–721 (2021). https://doi.org/10.1134/s0361768821080193
70. Li, T.: Identifying security requirements based on linguistic analysis and machine learning. In: IEEE Asia-Pacific Software Engineering Conference (APSEC) (2017)
71. Jindal, R.: Automated classification of security requirements. In: 2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI (2016)
72. Maalej, W.: Automatically classifying functional and non-functional requirements using supervised machine learning. In: IEEE 25th International Requirements Engineering Conference (RE) (2017)
73. Fong, V.: RE data challenge: requirements identification with Word2Vec and TensorFlow. In: IEEE 25th International Requirements Engineering Conference (RE) (2017)
74. Abad, Z.S.H., et al.: What works better? a study of classifying requirements. In: IEEE 25th International Requirements Engineering Conference (RE) (2017)
75. Slankas, J., Williams, L.: Automated extraction of non-functional requirements in available documentation. In: 1st International Workshop on Natural Language Analysis in Software Engineering, NaturaLiSE 2013 - Proceedings (2013)
76. Williams, L.: Access control policy extraction from unconstrained natural language text. In: Proceedings -SocialCom/PASSAT/BigData/EconCom/BioMedCom (2013)

77. Merten, T., et al.: Software feature request detection in issue tracking systems. In: IEEE 24th International Requirements Engineering Conference, RE 2016 (2016)
78. Vogelsang, A.: Automatic classification of requirements based on convolutional neural networks. In: IEEE 24th International Requirements Engineering Conference Workshops, REW 2016 (2016)
79. Cleland-Huang, J., et al.: A machine learning approach for tracing regulatory codes to product specific requirements. In: 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10 (2010)
80. Gokyer, G., et al.: Non-functional requirements to architectural concerns: ML and NLP at crossroads. In: The Third International Conference on Software Engineering Advances, 2008. ICSEA'08 (2008)
81. Haiduc, S.: A machine learning approach for determining the validity of traceability links. In: IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017 (2017)
82. Sardinha, A., et al.: EA-Analyzer: automating conflict detection in a large set of textual aspect-oriented requirements. Autom. Software Eng. 20(1), 111–135 (2013). https://doi.org/10.1007/s10515-012-0106-7
83. Khatian, V.M, et al.: Comparative analysis for predicting non-functional requirements using supervised machine learning. In: 1st International Conference on Artificial Intelligence and Data Analytics (2021)
84. Atoum, I., et al.: Challenges of software requirements quality assurance and validation: a systematic literature review. IEEE Access 9, 137613–-137634 (2021). https://doi.org/10.1109/access.2021.3117989
85. Atas, M.: Automated identification of type specific dependencies between requirements. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 688–695 (2018)
86. Sharma, R.: Automated identification of business rules in requirements documents. In: 2014 IEEE International Advance Computing Conference (IACC) (2014)
87. Parra, E., et al.: A methodology for the classification of quality of requirements using machine learning techniques. Inf. Software Technol. 67, 180–195 (2015). https://doi.org/10.1016/j.infsof.2015.07.006
88. Hayes, J.H., et al.: Measuring requirement quality to predict testability. In: 2015 IEEE Second International Workshop on Artificial Intelligence for Requirements Engineering (AIRE) (2015)
89. Hussain, I.: Automatic quality assessment of SRS text by means of a decision-tree-based text classifier. In: Seventh International Conference on Quality Software (QSIC 2007) (2007)
90. Alashqar, A.M.: Studying the commonalities, mappings and relationships between non-functional requirements using machine learning. Sci. Comput. Program. 218, 102806 (2022). https://doi.org/10.1016/j.scico.2022.102806
91. Binkhonain, M., Zhao, L.: A machine learning approach for hierarchical classification of software requirements. Mach. Learn. Appl. 12, 100457 (2023). https://doi.org/10.1016/j.mlwa.2023.100457
92. Alhoshan, W., Ferrari, A., Zhao, L.: Zero-shot learning for requirements classification: an exploratory study. Inf. Software Technol. 159, 107202 (2023). https://doi.org/10.1016/j.infsof.2023.107202
93. Quba, G.Y., et al.: Software requirements classification using machine learning algorithm's. In: 2021 International Conference on Information Technology (ICIT), Amman, Jordan, (2021)
94. Phalnikar, R.: Classification and prioritisation of software requirements using machine learning – a systematic review. In: 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 2021. Noida, India (2021)
95. Pérez-Verdejo, J.M.: A systematic literature review on machine learning for automated requirements classification. In: 2020 8th International Conference in Software Engineering Research and Innovation (2020)
96. Nayak, U.A.: Feasibility study of machine learning & AI algorithms for classifying software requirements. In: 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India (2022)
97. Hauser, S., et al.: An improved assessing requirements quality with ML methods. In: 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN). Al-Khobar, Saudi Arabia (2022)
98. del Águila, I.M.: Stability prediction of the software requirements specification. Software Qual. J. 26(2), 585–605 (2018). https://doi.org/10.1007/s11219-017-9362-x
99. Dargan, J.L., Wasek, J.S., Campos-Nanez, E.: Systems performance prediction using requirements quality attributes classification. Requir. Eng. 21(4), 553–572 (2016). https://doi.org/10.1007/s00766-015-0232-4
100. Khanna, M.: An exploratory study for software change prediction in object-oriented systems using hybridized techniques. Autom. Software Eng. 24(3), 673–717 (2017). https://doi.org/10.1007/s10515-016-0203-0
101. Del Sagrado, J.: Requirement risk level forecast using Bayesian networks classifiers. Int. J. Software Eng. Knowl. Eng. 21(2), 167–190 (2011). https://doi.org/10.1142/s0218194011005219
102. Quba, G.Y., et al.: An intelligent prototype for requirements validation process using machine learning algorithms. In: International Conference on Information Technology (2021)
103. Abdukalykov, R., et al.: Quantifying the impact of different non-functional requirements and problem domains on software effort estimation. In: 9th International Conference on Software Engineering Research Management and Applications (2011)
104. Sakhrawi, Z., Sellami, A., Bouassida, N.: Software enhancement effort prediction using machine-learning techniques: a systematic mapping study. SN COMPUT. SCI. 2(6), 468 (2021). https://doi.org/10.1007/s42979-021-00872-6
105. Wang, Y.: Automatic semantic analysis of software requirements through machine learning and ontology approach. J. Shanghai Jiaot. Univ. 21(6), 692–701 (2016). https://doi.org/10.1007/s12204-016-1783-3
106. Fitzgerald, C., Letier, E., Finkelstein, A.: Early failure prediction in feature request management systems: an extended study. Requir. Eng. 17(2), 117–132 (2012). https://doi.org/10.1007/s00766-012-0150-7
107. Diri, B.: A systematic review of software fault prediction studies. Expert Syst. Appl. 36(4), 7346–7354 (2009). https://doi.org/10.1016/j.eswa.2008.10.027
108. Onan, A.: SRL-ACO: a text augmentation framework based on semantic role labeling and ant colony optimization. J. King Saud Univ.-Comput. Inf. Sci. 35(7), 101611 (2023). https://doi.org/10.1016/j.jksuci.2023.101611

How to cite this article: Hassan, S., et al.: A systematic mapping to investigate the application of machine learning techniques in requirement engineering activities. CAAI Trans. Intell. Technol. 1–23 (2024). https://doi.org/10.1049/cit2.12348