Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059832

ARTICLE

An Improved Hybrid Deep Learning Approach for Security Requirements Classification

Shoaib Hassan[1,*], Qianmu Li[1,*], Muhammad Zubair[2], Rakan A. Alsowail[3] and Muhammad Umair[2]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China
[2]Faculty of Information Technology and Computer Science, University of Central Punjab, Lahore, 5400, Pakistan
[3]Computer Skills, Self-Development Skills Development, Deanship of Common First Year, King Saud University, Riyadh, 11362, Saudia Arabia

* Corresponding Authors: Shoaib Hassan. Email: shoaibcomsats11@yahoo.com; Qianmu Li. Email: qianmu@njust.edu.cn

Received: 17 October 2024; Accepted: 13 December 2024; Published: 06 March 2025

ABSTRACT: As the trend to use the latest machine learning models to automate requirements engineering processes continues, security requirements classification is tuning into the most researched field in the software engineering community. Previous literature studies have proposed numerous models for the classification of security requirements. However, adopting those models is constrained due to the lack of essential datasets permitting the repetition and generalization of studies employing more advanced machine learning algorithms. Moreover, most of the researchers focus only on the classification of requirements with security keywords. They did not consider other nonfunctional requirements (NFR) directly or indirectly related to security. This has been identified as a significant research gap in security requirements engineering. The major objective of this study is to propose a security requirements classification model that categorizes security and other relevant security requirements. We use PROMISE_exp and DOSSPRE, the two most commonly used datasets in the software engineering community. The proposed methodology consists of two steps. In the first step, we analyze all the nonfunctional requirements and their relation with security requirements. We found 10 NFRs that have a strong relationship with security requirements. In the second step, we categorize those NFRs in the security requirements category. Our proposed methodology is a hybrid model based on the Convolutional Neural Network (CNN) and Extreme Gradient Boosting (XGBoost) models. Moreover, we evaluate the model by updating the requirement type column with a binary classification column in the dataset to classify the requirements into security and non-security categories. The performance is evaluated using four metrics: recall, precision, accuracy, and F1 Score with 20 and 28 epochs number and batch size of 32 for PROMISE_exp and DOSSPRE datasets and achieved 87.3% and 85.3% accuracy, respectively. The proposed study shows an enhancement in metrics values compared to the previous literature studies. This is a proof of concept for systematizing the evaluation of security recognition in software systems during the early phases of software development.

KEYWORDS: Requirements engineering; security requirements; deep learning; CNN; XGBoost; classification

1 Introduction

In recent years, there has been a notable growth in software system breaches. These breaches have mainly affected systems used in banking, healthcare, e-commerce, and other web applications [1–3]. This problem is aggravated by the increase in advanced persistent threats, which are hard to recognize by nature [4]. Many businesses actively work to put security measures in place to protect software systems from the frequent risks of today. However, when these precautions are included at a later stage of development, like the deployment

phase, they intensify vulnerabilities and increase the possibility of software failure. Furthermore, its regular implementation has to offer a degree of protection commensurate with the complexity of the techniques used by attackers [5]. Ensuring the safety of the program during its development process guarantees its resilience and robustness against any threats. Additional security precautions are taken during deployment in addition to this [6,7]. This requirement emphasizes how important it is to start thinking about security requirements engineering early in the software development life cycle (SDLC). This approach finds and records the security requirements (SR) necessary for creating safe software systems, in addition to integrating security into the design and testing it concurrently with functional requirements [8]. Using this method promises a more thoughtful application of security as opposed to treating it as an afterthought, which would require reworking the architecture and result in increased maintenance expenses.

Security requirements (SR), a subset of non-functional requirements (NFR) that impose constraints on the system's behavior. Typically, these boundaries stem from industry best practices or standards that specify security features or services necessary to protect the software from possible intrusions [9]. They are gathered via user stories, software evaluations, and documentation supplied by different stakeholders, together with additional software requirements. The security requirements elicitation procedure that is used to collect SR contributes to the security requirements analysis process [10]. As security requirements are written in natural language, it is hard to perform the study using traditional statistical tools. The majority of software engineers have a low understanding of security, which makes this problem worse. As part of the analysis process, security requirements are classified. To do this, each security requirement is categorized into a specific model according to the security services the software offers [11]. Research in this vital area is discouraged because, although many studies concentrate on functional and non-functional requirements, including security in a broad sense, those that give a more thorough model do not provide a comparable dataset [12]. Many non-functional requirements can be seen as only non-functional requirements and have nothing to do with security. Maintainability is frequently seen as a non-functional requirement that has nothing to do with security. The security defenses of the system may be compromised by maintenance operations, which may also introduce fresh vulnerabilities into the program that did not exist before. The categorization of software requirements becomes a challenging process in these conditions. Several machine learning and natural language processing methods are available to help with tasks like reliability prediction and software requirement classification. However, suitable datasets for machine learning model testing, validation, and training are required to extract and categorize software requirements [13]. The software requirements datasets that are now available are either excessively tiny or skewed toward functional and non-functional needs, with little attention paid to security issues. Replicable research on the classification of security requirements is delayed by this constraint. Furthermore, there are multiple machine learning algorithms, which adversely affects the classification outcomes [14].

In this study, we use three datasets for the classification methodology of security requirements. One dataset is the DOSSPRE Multi-class dataset (Dataset of Student's Software Projects Requirements), which contains 1400 requirements. That dataset contains different classes, including non-security and different types of security requirements. Another dataset is the DOSSPRE binary, which contains only two security and non-security requirements classes. The third dataset is PROMISE_exp, the most famous dataset in empirical software engineering research. The PROMISE_exp data set includes 1030 requirements labeled with security and non-security requirements. All three datasets have functional and non-functional requirements. All datasets have a large number of security requirements. The main reason for selecting these datasets is their capability to meet many security requirements. The main objective of this study is to provide a novel approach to security requirements classification that is validated by these three datasets. Numerous machine learning models have applications in requirements classification tasks. Our proposed study uses convolutional neural

network (CNN) and Extreme Gradient Boosting (XGBoost) machine learning techniques. We use a hybrid machine learning model to achieve maximum accuracy in the security requirements classification task. Both machine learning techniques have their structure and usage. CNN is one of the most widely used deep learning models in image processing tasks. CNN is commonly used to identify the pattern in the data using convolutional layers [15]. The major key components of CNN are convolutional layers, pooling layers, fully connected layers, and activation functions [16]. Each component has its responsibility. CNN is mainly used to classify text or images. XGBoost is an ensemble learning model that uses the prediction of numerous decision trees to make a strong model [17]. XGBoost has features like regularization, handling missing values, parallel processing, and gradient boosting. Classification and regression are the most common applications of XGBoost. The integration of CNN and XGBoost achieves maximum accuracy. CNN uses multiple layers of neurons, while XGBoost uses a decision tree with ensemble learning [18]. XGBoost works with low resources and low computational power, while CNN requires a high level of resources. The combination of both techniques creates a useful impact on our proposed methodology. The strength of CNN and XGBoost is that they produce maximum accuracy in security requirements classification tasks. The major contributions of our research work are given below:

- Highlights the critical relationship between security requirements and other non-functional requirements, expanding the scope of security analysis beyond traditional keyword-based approaches.
- Expands the DOSSPRE and PROMISE_exp datasets by adding 84 and 61 new instances, respectively, and updating the requirement type column to classify requirements into security (SR) and non-security (NSR) categories.
- Provides comprehensive SR and NSR labeling for all 1400 instances in the DOSSPRE dataset and 1030 instances in the PROMISE_exp dataset, ensuring consistent categorization across both datasets.
- Evaluate the effectiveness of a combined CNN and XGBoost model for accurately classifying requirements into security (SR) and non-security (NSR) categories.
- Improves validation accuracy by leveraging a hybrid CNN and XGBoost model to classify security requirements.

The document is structured with a literature review discussed in Section 2. The training dataset, preprocessing, and feature engineering procedure will be described in Section 3. Section 4 presents and discusses the categorization and mapping of security requirements suggested in this study. In Section 5, the hybrid models' evaluation is performed, and the findings are comprehensively discussed. Finally, Section 6 concludes the study with possible future work.

2 Literature Review

Machine learning (ML) has many software requirements classification techniques [19]. After the advancement in ML techniques, much research has been done on the classification of requirements into functional and non-functional categories. However, there is very little research on the classification of security requirements. Security is one of the most important non-functional requirements for every system [20]. Most researchers focus on the functional requirements that define the services or features of the system or quality attributes (non-functional requirements) that define the system's quality. Performance, security, reliability, usability, maintainability, and portability are the most used non-functional requirements. Without these requirements, the system cannot meet its quality criteria. Many non-functional requirements come under the category of security requirements due to their features and applications. The study [21] addressed four NFR categories: security, performance, usability, and operational. The study [22] addressed different NFRs including Legal = L, Availability = A, Look and Feel = LF, Security = SE, Maintainability = MN, Performance = PE, Usability = US, Fault Tolerance = FT, Portability = PO, Scalability = SC, and Operational = O. Some

researchers categorized fault tolerance in the security requirement category as it deals with survivability. The classification of security requirements is based on the security services suggested in some research studies. The study [23] suggested that the CIA model stands for confidentiality, integrity, and availability. Alonge et al. focused on information assets and proposed a framework for information assets classification for risk assessment [24]. The study [25] addressed four security principles authentication, authorization, identification, and audit. Access control is also related to authentication and authorization. ISO7498-2:1989 addressed authentication, authorization, availability, confidentiality, integrity, non-repudiation, and audit attributes. Availability, confidentiality, integrity, authentication, and accountability are identified as core security characteristics in the study [26] while authorization is defined as a fundamental security service.

2.1 Security Requirements (SR)

The literature studies categorized some other NFRs into security requirements. This categorization is helpful in our research study in analyzing those NFRs and putting them in the security requirements category. Those requirements that are relevant to security are given in Table 1 below.

Table 1: Nonfunctional requirements relevant to security requirement

Sr. no.	NFRs	Description
1	Availability (AVA)	The duration the system is available for the end user is called availability. Numerous security attacks on the system may affect the availability.
2	Authentication (THE)	To verify any user entering into the system is called authentication. It identifies whether the user is who they claim to be. It is directly related to the security of the system.
3	Authorization (THO)	Authorization comes after authentication. It defines how much access should be given to some users and ensures the system's security by giving specific access to the users.
4	Integrity (INT)	Integrity confirms accuracy and consistency in the software life cycle. It is directly related to authorization.
5	Immunity (IMM)	Immunity refers to the system's capability to resist or recover from numerous attacks. It also ensures data protection with continuous operations.
6	Intrusion Detection (IND)	It refers to the process used to identify unauthorized access. It is important in analyzing network traffic and identifying malicious attacks [27].
7	Auditing (AUD)	Auditing is a process to examine, review, and evaluate the history of all transactions in a software system. It involves reviewing databases, networks, and servers to ensure these are secure and safe.
8	Confidentiality (CON)	It is a software attribute that secures software data from unauthorized access and attacks. It maintains trust between the software system and the users.
9	Maintainability (MAI)	It is defined as how easily a software or its component can be maintained over time. It is used to reduce the time, effort, and cost required to fix defects and issues in the system.

(Continued)

Table 1 (continued)

Sr. no.	NFRs	Description
10	Survivability (SUR)	It refers to the capability of the system to perform its operations despite the presence of threats and issues in the system. It is helpful to recover the system after dealing with the threats.

Firesmith divided security requirements into 12 different categories [28]. Those categories are authentication, authorization, identification, integrity, immunity, intrusion detection, privacy, survivability, non-repudiation, security auditing, system maintenance, and physical protection. In the literature study, no dataset exists that supports this classification scheme. The study [29] addressed a security requirements taxonomy supported by two abstraction layers. The first layer described the basic security features, and the second described the other security sub-factors. Another study addressed the security requirements taxonomy for protecting the Internet of Things (IoT). In this study, the authors proposed a classification strategy by considering all possible threats and attacks. The authors focused on vulnerabilities in IoT systems that result from unrealized security requirements. There are also many challenges and issues in other domains, as there are very limited datasets that support the research and development of security requirements.

2.2 Non-Security Requirements

All requirements that do not contribute to the security aspects of the system are referred to as non-security requirements (NSRs). Normally, all functional requirements fall into this category. However, some NFRs fall into the NSR category. Some major NSRs are given in Table 2 below.

Table 2: Requirements irrelevant to security requirement

Sr. no.	Requirements	Description
1	Functional	A statement of service that a system should provide. Functional requirements are the major features that a system offers to its users.
2	Usability	It is defined as how easy it is to use the system. Usability is strongly related to learnability, look & feel, and human-computer interaction aspects of the software system.
3	Performance	Performance refers to how efficiently a system operates to meet its tasks. It includes efficiency, throughput, and response time.
4	Portability	Portability is defined as the ability of software to run on different platforms or environments.
5	Scalability	Scalability refers to the ability of a software to expand its resources without compromising its performance to meet its operations [30]. Scalability is helpful for those application that have growth ability.
6	Look & feel	It refers to the type of NFR related to design elements, physical appearance, and user experience with the software system. It is helpful to make the system attractive and appealing.
7	Fault tolerance	A software system can continue its operations in the presence of hardware or software faults. Fault tolerance is helpful for robust systems to handle faults and failures.

(Continued)

Table 2 (continued)

Sr. no.	Requirements	Description
8	Reliability	Reliability refers to the ability of the system to continue its operations without failure under different circumstances [31]. Reliability is very crucial for healthcare and finance systems.
9	Legal & standards	Legal requirements refer to the laws and regulations of organizations. They define the legal issues and standards of systems and organizations.
10	Response time	Response time defines the time a system takes to respond to an input. There is a difference in time between the request being made and the response being received.

All these requirements come under the category of non-security requirements. The above-mentioned requirements have no connection with the security.

2.3 Datasets in Security Requirements Classification

Multiple datasets exist that contain functional and non-functional requirements. One of the most popular datasets is the PROMISE dataset which includes 625 requirements. Of 625 requirements, 255 are functional, and 375 are NFRs. The dataset contains three attributes requirement, class, and project ID. Out of 375 NFRs, 66 are security requirements, which constitute 10.5% of the whole dataset [9,19,22]. The extension of the PROMISE dataset is called the PROMISE_exp dataset. In the extended dataset of PROMISE_exp, there are 1030 instances, of which 478 are functional requirements and 552 NFRs. Out of 552 NFRs, most of them are related to security requirements. Another dataset from the requirements specification document of three industry projects is called SeqReq [32]. This dataset contains 507 requirements. Out of 507, 187 requirements are related to security. This dataset contains only two attributes requirement and requirement class. The DOSSPRE dataset is a new dataset in the requirements engineering community. There are two versions of DOSSPRE datasets. One version contains binary labels for the requirement class. That version is called the DOSSPRE binary dataset. Another version is the DOSSPRE multi-class dataset, which contains multiple labels for class attributes that require it. The DOSSPRE dataset contains 876 non-security and 524 security requirements. Another dataset that is widely used for requirements engineering tasks is PURE (Public Requirements Dataset). This dataset contains 79 requirements documents that are extracted from the web. In 79 documents, 34,268 sentences were written in natural language, which is helpful for requirements engineering tasks [24]. In these sentences, there is very little contribution towards security requirements. From literature studies, we observe that the existing datasets have very low security requirements compared to the other requirements. This makes it challenging to deploy machine learning techniques to classify security requirements. We observed that there is not a single classification model that classifies security requirements efficiently. Therefore, we proposed a classification model for security requirements that will be supported by three datasets PROMISE_exp, DOSSPRE Multi class, and DOSSPRE binary datasets. The evaluation of the model through three datasets will increase its effectiveness.

2.4 Deep Learning in Requirements Engineering

Deep learning has various applications in requirements engineering (RE) [33]. One of the major applications of deep learning is requirements classification [34]. Deep learning has numerous models that can classify the requirements into different categories efficiently [35]. It is very useful for large applications

in which there is a bulk number of requirements. Different deep learning models like Bidirectional Encoder Representation from Transformer (BERT) which is helpful for natural language processing, Generative Pre-trained Transformers (GPT), which is a neural network model, and Convolutional Neural Network (CNN), which is also a most common neural network model are very useful for extracting useful requirements from a large volume of requirements [36]. In requirements prioritization, deep learning plays an important part in analyzing the feedback of different stakeholders [37]. Deep learning is also helpful for detecting anomalies and ambiguities that exist in the requirements. The requirements changes can easily be tracked by using deep learning approaches. Translation of requirements has become easy since the evolution of deep learning techniques. Deep learning enhances the efficiency and accuracy of the RE process. The latest deep learning models improve the RE process. CNN is helpful for requirements classification. Transformers like BERT and GPT are also useful for extracting useful information from the requirements. Recurrent neural networks (RNN) also help the RE process by executing text processing and classification tasks. Deep reinforcement learning and graph neural networks (GNN) are also beneficiaries of requirements prioritization and traceability tasks.

The combination of CNN and XGBoost techniques gives efficient results. Chen used CNN to classify user requirements [38]. Chen found CNN helpful in improving the accuracy of classification tasks by 13.5% as compared to previous studies. The study [39] used the CNN model to classify requirements into functional and non-functional categories. The study used the PROMISE dataset to classify software requirements using the CNN model. Shatnawi [40] applied CNN and support vector machine (SVM) which is useful for classification and regression tasks for classifying online datasets. Shatnawi found an accuracy of 85% for the requirements classification into functional and non-functional categories. Hidayat et al. [41] addressed the classification of NFRs from application review datasets using the CNN model. Hidayat found 80% accuracy with the CNN model for NFR classification. The study [42] used a multiclass CNN model for the functional requirements classification extracted from the requirements specification document and achieved a maximum of 77% accuracy. Ramraj addressed the effectiveness of the XGBoost algorithm for accurate classification across different datasets [43]. Ramraj found an accuracy of 77.95% with the XGBoost algorithm. The study [44] shows the effectiveness and strength of using both CNN and XGBoost for classification tasks. It showed an accuracy of 84.2% with CNN and 83.6% with the XGBoost model.

3 Materials and Methods

We used two datasets for our research study. One is PROMISE_exp, and the other is the DOSSPRE dataset. We extend the PROMISE_exp and DOSSPRE datasets by adding healthcare system requirements. First, we review the healthcare system document, and after our analysis, we incorporate the relevant requirements into our datasets. Experts validated those extracted requirements. After extension, the size of our datasets increases. The major reason for choosing healthcare system requirements is its diverse nature. There are also many other reasons for choosing healthcare requirements. PROMISE_exp dataset contains healthcare requirements, and DOSSPRE contains students' project requirements. Both datasets contain requirements that are very relevant to healthcare. So, it is useful to add healthcare requirements to extend the datasets. Another reason is healthcare requirements standards. Healthcare requirements strictly follow healthcare standards, which ensure the security parameters of the system are met. Healthcare requirements are very sensitive and secure. So, it is necessary to include those requirements in the security requirements classification task. Healthcare requirements have many security risks. The addition of these requirements to the datasets will help us analyze the security risks in detail. Healthcare requirements contain a diverse nature of users. So, all types of security aspects can be covered by adding security requirements to the datasets. Adding healthcare requirements to the datasets may be helpful in showing their other perspectives on. The

PROMISE_exp dataset has 969 labeled requirements, while DOSSPRE contains 1316. Both datasets have been written in English and used in multiple studies. The studies [39,45], and [46], used these two datasets. The studies [47–49] also used these two famous datasets of the software engineering domain. Three significant attributes exist in both datasets. Those attributes are requirement ID, requirement text, and requirement class. The requirement ID indicates the project from which the requirement originated. The requirement text presents the requirement description, while the requirement class describes the requirement category, whether it is a functional or non-functional requirement. PROMISE_exp dataset contains functional and 11 different types of NFRs. In comparison, the DOSSPRE dataset contains functional and ten types of NFRs that will be discussed in detail in a later section.

3.1 Proposed Methodology Overview

The proposed methodology offers several advantages over existing competitive studies. The proposed methodology works on two well-known datasets of the software engineering community. Those datasets contain a variety of non-functional requirements. Our proposed methodology extends both datasets by adding requirements from healthcare system documents. In extension, we added 84 requirements to the DOSSPRE dataset. In addition, DOSSPRE contains 1400 instances of requirements. We added 61 new requirements instances to the PROMISE_exp dataset. In addition, PROMISE_exp contains 1030 requirements. After adding the requirements, we label the datasets with security and non-security requirements (NSR) according to the nature of non-functional requirements. DOSSPRE multi-label dataset contains NSR and different types of security requirements. We classify this dataset into two categories. We categorized all requirements except NSR into the security requirements category. We convert the DOSSPRE multi-label dataset into a binary label dataset by classifying it into binary labels. Likewise, we classify the PROMISE_exp dataset into a binary label dataset by classifying the requirement class into security and NSR categories. After categorizing, we do feature engineering on our extended labeled datasets using a combination of CNN and XGBoost algorithms. After feature engineering, we classify both datasets into security and non-security categories with validation accuracies of 0.87 and 0.86. The overall workflow of the proposed methodology can be presented in Fig. 1 below.

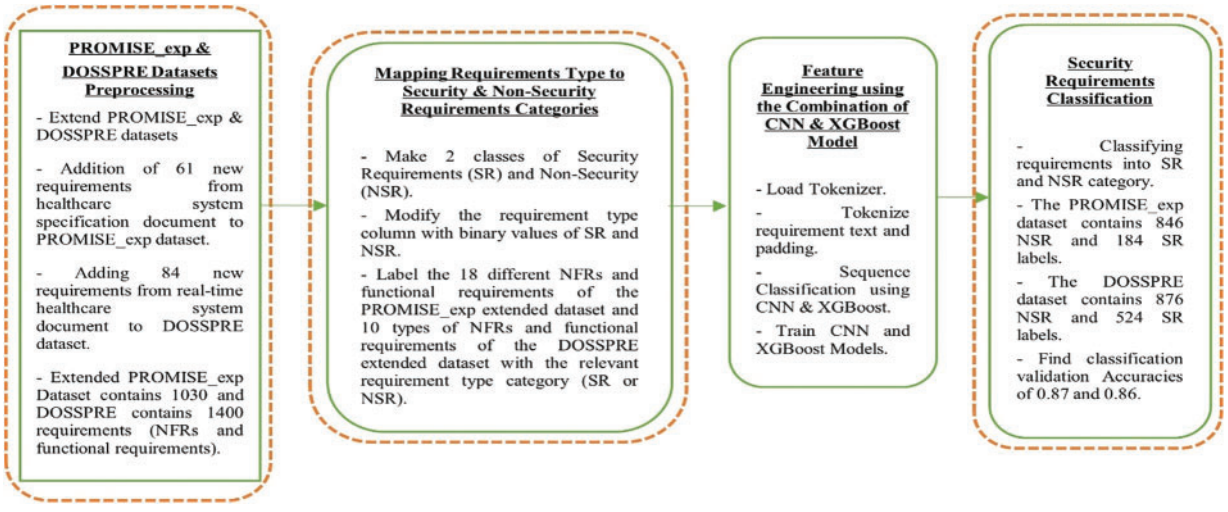

Figure 1: Workflow of the proposed study

The red boxes highlighted in the figure above illustrate the key contributions of our proposed research methodology. The first, second, and fourth boxes show the major novelty of our study. Datasets preprocessing and training are executed in the first box. The second box contains the mapping and labeling requirements for SR and NSR categories. The feature engineering process is being executed in the third box. In the last box, we find the classification accuracies for the security requirements of both datasets.

3.2 Datasets Extension and Labeling

We extend the datasets as a part of our research contribution. All the steps executed in preprocessing, dataset extension and labeling are represented in Fig. 2.

All the steps mentioned in Fig. 2 above are the preprocessing steps. In the first step, we searched healthcare specification documents. After searching, we analyzed the retrieved documents. All the requirements of the documents were examined. We extracted functional and different types of NFRs from those documents. After software experts validated those requirements, we added them to both datasets. In validation, experts check the completeness and consistency of the requirements. This process extends the PROMISE_exp and DOSSPRE datasets. After extension, the extended DOSSPRE dataset contains 1400 requirements while PROMISE_exp contains 1030 requirements. The requirement type column in both datasets is a multi-label column with multiple values of different NFRs and functional requirements. The extended datasets were unlabeled. We will describe the labeling process in the next section.

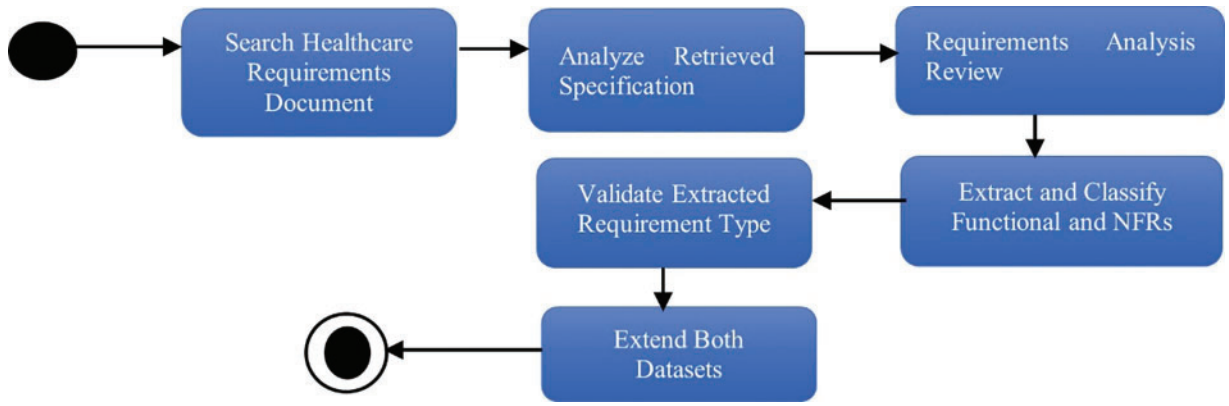

Figure 2: Datasets expansion and labeling

3.3 Datasets Labeling

Manual labeling was chosen based on the characteristics of the DOSSPRE and PROMISE_exp extended datasets. A software professional was designated to review each requirement and give suitable labels. These labels help as a good means to present the context of security requirements among different NFRs. The labeling process is presented in Fig. 3.

Before labeling, the PROMISE_exp dataset contains 11 types of NFRs and functional requirements. The distribution of NFRs of the PROMISE_exp dataset before labeling can be presented in Fig. 4.

As in Fig. 4 above, the highest number of NFR in the PROMISE_exp dataset is security requirement, having 128 requirements. In the labeling process, we put some other categories of NFRs in the security requirements category due to their relevance with security features. The linkage between those NFRs and security requirements has been discussed in the above section. After labeling, the PROMISE_exp dataset contains 184 SR and 846 NSR labels in the requirement type column. Before labeling, the extended DOSSPRE

dataset contains 11 types of labels in the requirement type attribute. The distribution of those labels can be shown in Fig. 5 below.

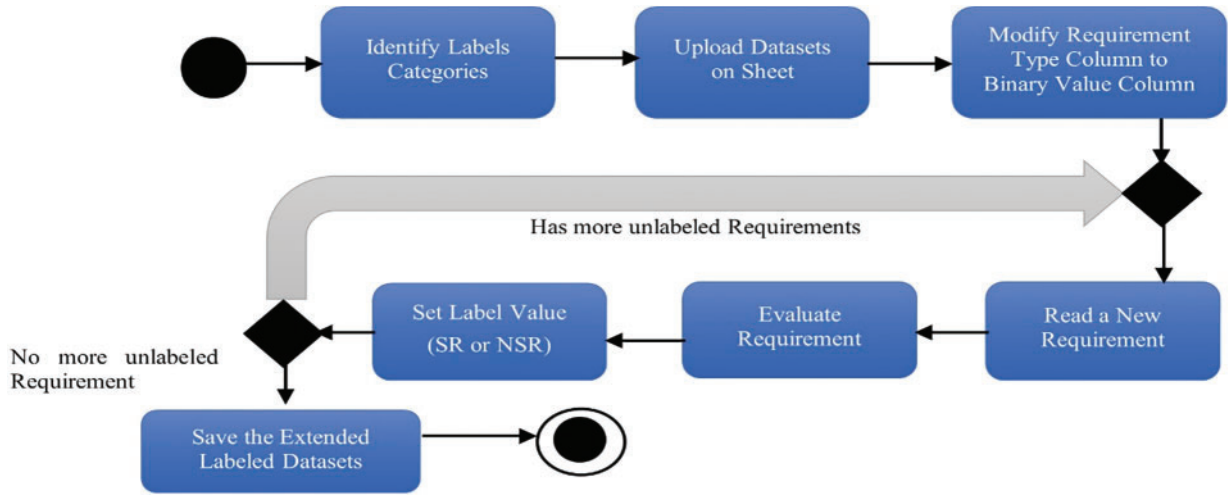

Figure 3: Labeling process in the extended datasets

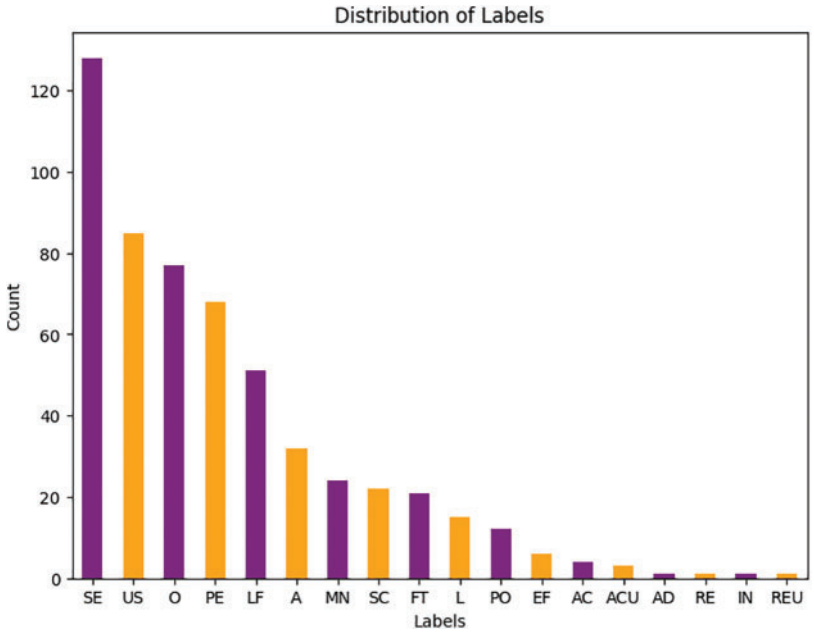

Figure 4: Distribution of NFRs in PROMISE_exp dataset before labeling

As in Fig. 5 above, the highest number of labels in the DOSSPRE dataset is NSR, which has 876 requirements. In the labeling process, we put all other categories of NFRs in the security requirements category due to their relevance to security features. The linkage between these NFRs and security requirements has been discussed in the above section. After labeling, the DOSSPRE dataset contains 524 SR and 876 NSR labels in the requirement type column.

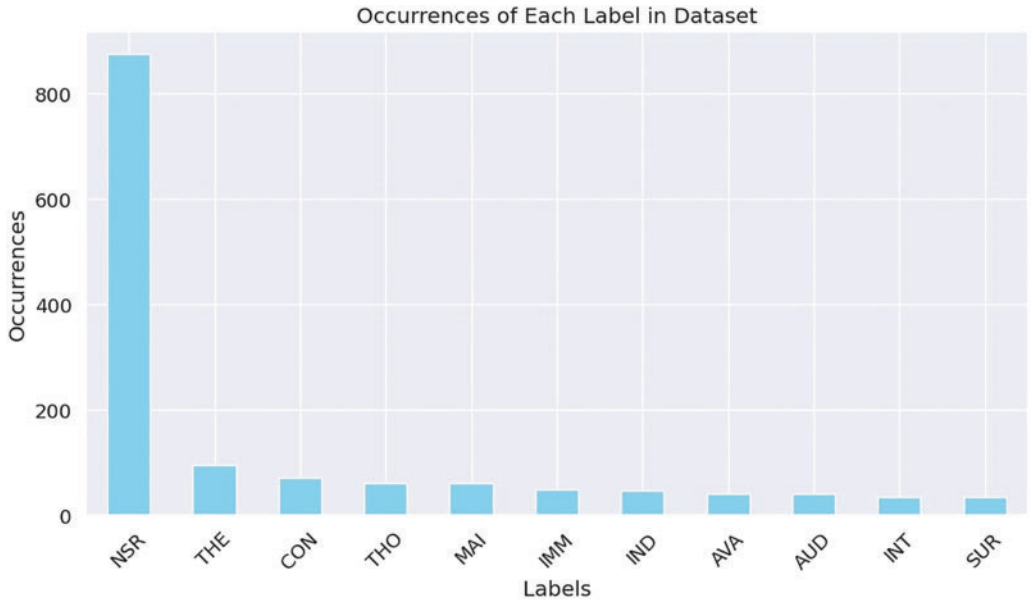

Figure 5: Distribution of requirement type labels in DOSSPRE dataset before labeling

3.4 Feature Engineering

This section describes feature engineering, which obtains the features of security requirements in extended datasets. Multiple feature engineering methods exist, but we selected a combination of CNN and XGBoost models. CNN and XGBoost models perform well in natural language processing (NLP) and classification tasks. CNN reduces the manual feature engineering process. CNN extracts features directly from the raw data. CNN can be extended in deep and complex architectures to model patterns, while XGBoost provides high accuracy in classification-related tasks. XGBoost is a scalable and efficient model with broader flexibility that can be used in numerous problems.

This study used the sklearn library for pre-processing tasks and pandas for data manipulation. We can also use other Python libraries. After importing libraries, we convert the labels into numerical values using a label encoder. Then tokenization and padding are performed on the datasets. After padding, we apply CNN and XGBoost models on our extended datasets and calculate the accuracies across both datasets. The hybrid model improves the accuracy of classifying the security requirements by combining the strengths of both CNN and XGBoost models. The next section will present a detailed discussion regarding the feature engineering process.

4 Proposed Methodology

This section gives an internal overview of the proposed methodology. Our study is novel in mapping some NFRs to the security requirement category and classifying them with the highest accuracy. Our proposed methodology contains many steps. Firstly, we find the requirements relevant to the security requirement category. Then, we categorized all the requirements into two categories: security and non-security requirements. After categorization, we used a hybrid model of CNN and XGBoost to classify the security requirements in both datasets. According to the literature review, no study uses expanded datasets and classifies security requirements with the highest validation accuracies. The relevance and linkage of requirements with security requirements are given below.

4.1 Mapping Relevant Requirements to Security Category

Some NFRs are directly or indirectly related to security requirements. We examine both datasets and analyze all the NFRS. The NFRs that impact the system's security fall under the security requirement category. All other requirements come under the category of non-security requirements. The linkage of NFRs with security requirements is given in Table 3 below.

Table 3: The relationship between NFRs and security requirement

Sr. no.	NFRs	Linkage with security requirement
1	Availability (AVA)	Availability is directly related to security. If a system is available, it is safe from different threats and attacks. The unavailable systems due to security attacks compromise the user's trust and may affect significant business operations.
2	Authentication (THE)	Authentication is very important for the security of the systems. It ensures that access is granted to only authorized users or the systems. It protects the system from unauthorized users and attacks. It is also helpful in auditing the systems.
3	Authorization (THO)	Authorization gives users access rights. It decides how much access should be given to specific users. It is useful for protecting the resources within the system. Authorization can mitigate security risks and protect the system from unauthorized access.
4	Integrity (INT)	Integrity is a significant element of security. It confirms that the system and data resources are consistent and accurate and protects them from unauthorized modifications and corruption. It is also helpful for maintaining the reliability of information.
5	Immunity (IMM)	It is directly related to security. It is a capability to resist various security attacks and ensure that a system performs its operations in a normal mode. It is also helpful in the recoverability of the systems after security attacks.
6	Intrusion Detection (IND)	Intrusion detection has a direct relationship with security as it detects security attacks in systems and networks. It enhances security and informs the organizations about the strategies for protecting the systems from security attacks.
7	Auditing (AUD)	Auditing is a method to examine, review, and evaluate the operations of a software system. It is directly related to security as it detects the security loopholes in the system.
8	Confidentiality (CON)	It is related to the security and trust of the users. It secures software data from unauthorized access. It maintains trust and confidence between the software system and the users.
9	Maintainability (MAI)	The more maintainable software systems are more secure and resilient. It is a capability to maintain the system with all its operations and protect it from unnecessary attacks and risks [50].
10	Survivability (SUR)	Survivability is the capability of the system to perform its processes despite security threats and challenges. It is helpful for the system's recoverability.

The reason for selecting the above requirements in Table 3 is their strong relationship with the security requirement. All the above requirements are directly or indirectly related to the security. Availability is very crucial for the security of the system as availability is the capability of the user to access the system and its resources without any disruption. Availability ensures that the system should be available to its authorized users all the time. It is also related to authentication and authorization requirements of the system that are also relevant to the security requirement. Authentication maintains the user's identity, while authorization establishes the user's accessible area. According to a security point of view, integrity is also very important for the system. Integrity ensures that the system's data is secure and reliable and has not been altered. Integrity is the most important requirement to achieve the security objective of the system. A secure system also has great immunity against threats, as immunity ensures that a system recovers from different attacks and vulnerabilities. Intrusion detection also has an important relationship with the security of the system. Intrusion detection helps us to increase the security of the system by identifying unauthorized access or problems within a system. Security is dependent on the auditing and confidentiality of the system. Auditing refers to the tracking of user actions and security activities of a system, while confidentiality ensures that the system's data is only available to authorized users. A system that is difficult to maintain may cause different security risks, and it may also impact the system's survivability. All of the above-selected NFRs have a strong association with security requirements due to their capability to address numerous directions of security.

4.2 Internal Workflow of Proposed Methodology

The overview of the proposed methodology in Section 3 describes all the significant steps executed in this study. Fig. 1 shows the proposed methodology abstractly. All steps contain further sub-steps that have already been discussed before. In this section, we will discuss the internal processing of the proposed methodology. The internal processing of the proposed methodology is shown in Fig. 6.

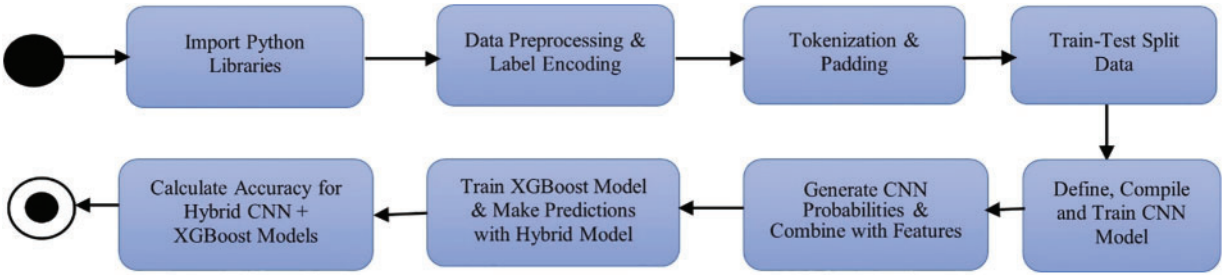

Figure 6: Internal workflow of proposed methodology

In Fig. 6 above, we have a total of 8 steps that cover our proposed methodology. Firstly, we import Python libraries. We import the panda's library for data manipulation, sklearn for preprocessing, Tensorflow for the CNN model, XGboost for the XGBoost model, and numpy for numerical operations. After importing libraries, we perform data processing label encoding. In data processing, we convert our data into labels and features. In label encoding, we convert the labels into numerical values. Tokenization and padding are performed after label encoding. The main purpose of tokenization is to convert text data into numerical sequences. The tokenizer splits the text into words and assigns a unique integer index to each word according to its frequency in the dataset. While padding ensures that all sequences have the same length. After padding, we break the data into two parts, called training and testing data.

To evaluate the model, we define the CNN model to perform text classification. After defining the CNN model, we compiled the model and used the accuracy metric to show the performance of our model. After

compilation, the CNN model is trained for 40 epochs and a batch size of 64. The trained CNN model is used to predict the probabilities for both training and testing data. After CNN, we train the XGBoost model and predict the class labels using the combined features. The predictions made by XGBoost compared with the true class labels. Lastly, the proposed model accuracy is calculated for both CNN and hybrid models. Our proposed hybrid model combines the strength of both CNN and XGBoost models and gives us the best results. We assess the results using performance metrics, including accuracy, recall, precision, and F1 Score values [51]. Table 4 provides the formulas to find performance metrics values.

Table 4: Performance metrics values calculation formulas

Sr. no.	Performance metric	Equation
1	Precision	$\frac{TP}{TP+FP}$
2	Recall	$\frac{TP}{TP+FN}$
3	Accuracy	$\frac{TP+TN}{TP+FN+TN+FP}$
4	F1 Score	$\frac{2*(Precision*Recall)}{Precision+Recall}$

Evaluation metrics like accuracy, recall, precision, and F1 Score are very important for assessing the performance of deep learning models in classification tasks. Each metric has its perspective to evaluate the model's performance. From Table 4, we can see that precision explains the correct predictions out of all positive predictions. It is very important in those cases where the cost of false positives is very high. Precision is also significant in real-world applications like disease screening or fault detection. High precision in those systems ensures that diseases are accurately predicted or frauds are detected correctly. Recall is also an important evaluation metric which is also called sensitivity. Recall value is crucial for the model when the false positive cost is high. It plays an important role in safety-critical systems like medical systems, where missing a positive case can be dangerous. Accuracy is the most used evaluation metric in machine learning. It defines the correct predictions among all instances. Accuracy can be misleading in imbalanced datasets. This evaluation metric is very useful in balanced datasets. The F1 Score is the harmonic mean of recall and precision. It reflects the importance of both recall and precision metrics. The F1 Score is very helpful in imbalanced datasets as it provides a single metric containing false positive (FP) and false negative (FN) values. The F1 Score is useful in sentiment analysis and information retrieval tasks. All these metrics are helpful to assess the performance of the model in different directions. We select these metrics according to the problem domain.

5 Results and Discussion

This section describes the results and evaluates the performance of our proposed hybrid deep learning model. We trained our hybrid model on the extended datasets that contain two types of requirements classes. One class is a security requirement (SR), and the other is a non-security requirement (NSR). We have used two extended datasets that were not balanced. PROMISE_exp dataset contains 184 SR and 846 NSR. In contrast, the DOSSPRE dataset contains 524 SR and 876 NSR. The hybrid model evaluation on both datasets is given in the next section.

5.1 Hybrid Model Evaluation

Our proposed model is defined to assess the capability of combining CNN and XGBoost models for the binary classification of SR and NSR categories. The core objective of our proposed model is to improve the performance metrics values. To assess the hybrid model, we partitioned our extended data set into 70:30 for training and testing. The batch size and epochs are two significant constraints that were measured for evolving a hyperparameter tuning technique to improve the performance of the hybrid model. The epoch number describes the total number of cycles a model concludes for a training dataset. In contrast, the batch size is described as how many samples continue before the model changes. We trained the hybrid model over epoch numbers from 1 to 20, 28, and 40 with batch sizes 32 and 64. The following metrics were chosen to assess the hybrid models' performance, including precision, accuracy, F1 Score, and recall. A specific random value is used to handle the randomization break of the testing and training data to overcome the overfitting issue. The evaluation scheme followed in our research methodology is given in Fig. 7.

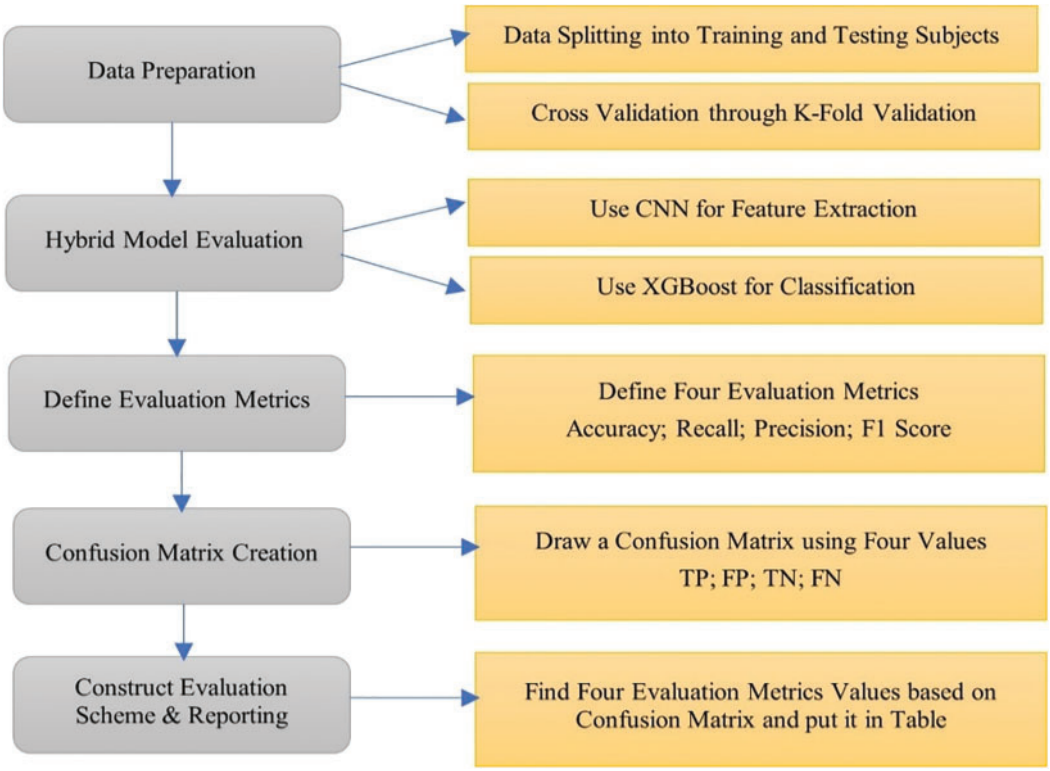

Figure 7: Evaluation scheme of proposed study

Through this evaluation scheme, we analyze how well our proposed hybrid model performs in security requirements classification tasks. The results obtained will be discussed in the next section.

5.2 Results Analysis

For analyzing the performance of our proposed hybrid model, we used a confusion matrix. In Fig. 8, we provide a confusion matrix for the PROMISE_exp dataset with 20 epoch numbers and 32 batch sizes. We have two classes in the confusion matrix. Those classes are SR and NSR and are represented by 1 and 0. We choose the SR label as positive and NSR as negative. It will give us four values; true negative (TN), false

negative (FN), true positive (TP), and false positive (FP). As in Fig. 8, 27 requirements of the SR category were precisely recognized by the hybrid classification model known as TP. 12 instances were predicted by the hybrid model as SR but were NSR known as FP. Likewise, 153 instances of NSR were precisely recognized by the hybrid deep learning model known as TN. 14 instances were recognized as NSR but were from SR, known as FN. In Fig. 9, we show a confusion matrix of the DOSSPRE dataset having binary label values with 28 epoch numbers and 32 batch sizes. In Fig. 9, 71 instances of SR were precisely recognized by the proposed model. 28 requirements were predicted by the hybrid model as SR but were NSR. Likewise, we have 168 instances of NSR that were precisely recognized by our proposed mode. l13 instances were recognized as NSR, but they were from SR. The proposed model's performance appears rationally stable between the two classes, with maximum accuracy in finding NSR instances in contrast to SR ones. In Fig. 10, we present a confusion matrix of the DOSSPRE dataset having multiple class labels with epoch number 40 and batch size 64. In this matrix, we have 11 label values from 0 to 10. The confusion matrix has a high number of predicted instances on label number 7, which shows the high number of non-security requirements among other nonfunctional requirements in the dataset. This examination of the confusion matrix brings valuable insights into the model's benefits and disadvantages in differentiating between SR and NSR classes, helping in more improvement of its projecting capabilities.

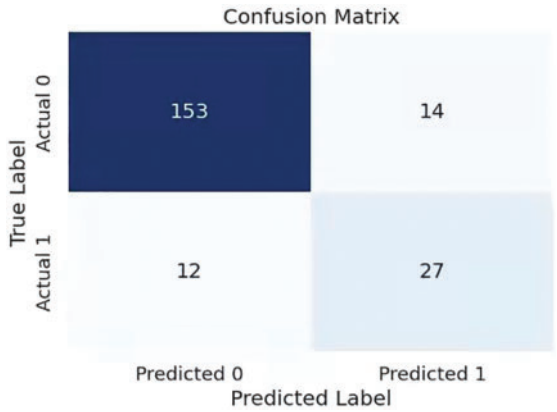

Figure 8: Confusion matrix of PROMISE_exp dataset with batch size 32 and epoch number 20

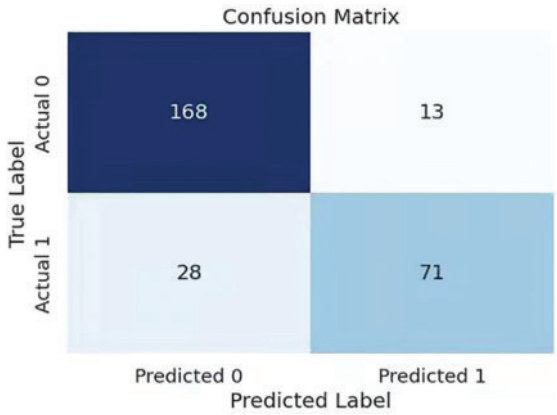

Figure 9: Confusion matrix of DOSSPRE (binary) dataset with batch size 32 and epoch number 28

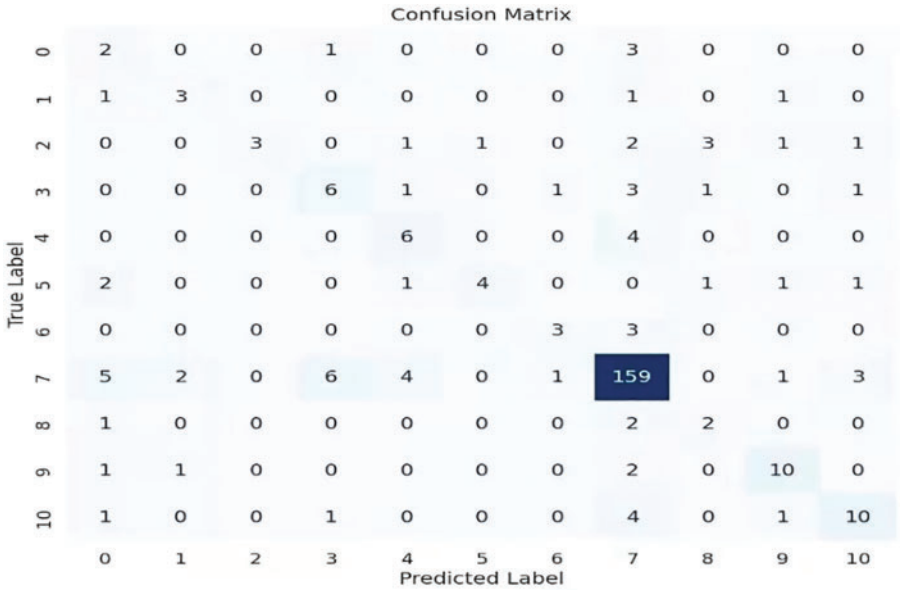

Figure 10: Confusion matrix of DOSSPRE (multi-label) dataset with batch size 64 and epoch number 40

We find validation loss and validation accuracy for all datasets. Validation loss is a metric used to assess the performance of the proposed hybrid model on the validation set. It is analogous to training loss and can be calculated by adding the sum of all errors for each instance in the validation set. While validation accuracy indicates how well our hybrid model performs on the validation set. Table 5. gives the validation loss and accuracy of the PROMISE_exp dataset using the proposed hybrid model. It can be seen that the model gives maximum validation accuracy at epoch numbers 8, 10, and 12. Table 6 shows the validation loss and accuracy of the DOSSPRE dataset, which has binary label values using the proposed hybrid model. It can be seen that the model gives maximum validation accuracy at epoch number 11. Table 7 shows the validation loss and accuracy of the DOSSPRE dataset, which has multiple label values using the proposed hybrid model. It can be seen that the model gives maximum validation accuracy at epoch numbers 17, 23, and 28.

Table 5: Validation loss and accuracy of PROMISE_exp dataset

Epoch number	Loss	Accuracy	Validation loss	Validation accuracy
1	0.525	0.821	0.493	0.810
2	0.447	0.824	0.457	0.810
3	0.413	0.824	0.428	0.810
4	0.325	0.825	0.359	0.815
5	0.183	0.930	0.315	0.864
6	0.083	0.983	0.309	0.854
7	0.030	0.997	0.414	0.864
8	**0.011**	**0.998**	**0.470**	**0.868**
9	0.005	1.000	0.451	0.859
10	0.003	1.000	0.507	0.868
11	0.002	1.000	0.511	0.864
12	0.001	1.000	0.532	0.868

Table 6: Validation loss and accuracy of DOSSPRE (binary) dataset

Epoch number	Loss	Accuracy	Validation loss	Validation accuracy
1	0.666	0.619	0.648	0.646
2	0.634	0.620	0.611	0.646
3	0.466	0.802	0.420	0.817
4	0.222	0.914	0.374	0.835
5	0.096	0.970	0.407	0.842
6	0.048	0.988	0.444	0.839
7	0.027	0.995	0.476	0.835
8	0.022	0.993	0.513	0.828
9	0.025	0.993	0.529	0.828
10	0.014	0.995	0.550	0.828
11	**0.017**	**0.995**	**0.598**	**0.850**
12	0.012	0.997	0.582	0.825

Table 7: Validation loss and accuracy of DOSSPRE (multi-label) dataset

Epoch number	Loss	Accuracy	Validation loss	Validation accuracy
1	2.172	0.515	1.755	0.646
2	1.616	0.619	1.500	0.646
3	1.513	0.619	1.465	0.646
4	1.461	0.619	1.439	0.646
5	1.396	0.619	1.385	0.646
6	1.282	0.619	1.300	0.646
7	1.125	0.630	1.210	0.646
8	0.992	0.681	1.171	0.685
9	0.889	0.717	1.148	0.689
10	0.789	0.759	1.116	0.682
11	0.689	0.783	1.102	0.692
12	0.582	0.813	1.063	0.696
13	0.477	0.867	1.028	0.707
14	0.379	0.921	1.024	0.725
15	0.287	0.950	1.048	0.742
16	0.216	0.970	1.018	0.771
17	**0.158**	**0.981**	**1.052**	**0.782**
18	0.113	0.986	1.086	0.778
19	0.085	0.988	1.102	0.778
20	0.065	0.992	1.175	0.767
21	0.051	0.993	1.132	0.778
22	0.046	0.994	1.302	0.771
23	0.048	0.996	1.176	0.782
24	0.031	0.994	1.241	0.771
25	0.055	0.993	1.283	0.775

(Continued)

Table 7 (continued)

Epoch number	Loss	Accuracy	Validation loss	Validation accuracy
26	0.310	0.995	1.208	0.778
27	0.024	0.994	1.394	0.767
28	0.023	0.995	1.234	0.782
29	0.019	0.995	1.303	0.775
30	0.015	0.996	1.309	0.775

Table 5 presents the loss, validation loss, accuracy, and validation accuracy of the PROMISE_exp dataset based on our proposed hybrid deep learning model. From the above table, we can see that at epoch 12, we have the minimum loss. Loss is used in the training process to assess the model's performance. In the above table, we get minimum loss at 12 epoch numbers in the training process. At epoch number 12, we achieved maximum accuracy, which shows the classification accuracy in the training process. Validation loss is computed on the dataset that is not used for training. We compute the validation loss of the validation dataset. The validation loss is helpful in checking whether either model is overfit or underfit. If validation loss increases and training loss decreases, it shows the overfitting problem. In the above table, we have minimum validation loss on epoch number 6. We achieve the highest accuracy in 8 epochs. Validation accuracy indicates how well our proposed model performs on the validation dataset.

Table 6 presents the four performance metric values of the DOSSPRE (binary) dataset based on our proposed hybrid deep learning model. From the above table, we can see that at epoch 12, we have the minimum loss. In Table 6 above, we get minimum loss at 12 epoch numbers in the training process. At epoch number 12, we achieved maximum training accuracy, which shows that the model is accurately classified in the training process. In the above table, we have minimum validation loss on epoch number 4. We achieve the highest accuracy in 11 epochs. The major difference between valuation and training accuracy shows the overfitting issue. The difference in our scenario is very low, which shows the better performance of our proposed model.

Table 7 presents the loss, validation loss, accuracy, and validation accuracy of the DOSSPRE (multi-label) dataset based on our proposed model. From the above table, we can see that at epoch 30, we have the minimum loss. The highest training accuracy was also achieved at epoch number 30. Validation loss is computed on the dataset that is not used for training. The validation loss and accuracy are used to generalize the proposed hybrid model's abilities. In the above table, we have minimum validation loss on the epoch number 16. We achieved the highest accuracy in 17 epochs. Validation loss and validation accuracy values indicate how well our proposed model performs on the validation datasets.

Table 8 discusses the performance metrics of the proposed hybrid deep learning model for PROMISE_exp, DOSSPRE (Binary Value), and DOSSPRE (Multi-label) datasets with different batch sizes and epoch values. The tables indicate that the hybrid model's performance improves as the values of accuracy, recall, precision, and F1 Score increase. In the PROMISE_exp dataset, we use 20 epochs and 32 batch sizes to achieve maximum accuracy. We achieved the highest accuracy of 87.3% on 20 epochs with batch size 32 for the PROMISE_exp dataset, among other datasets, as shown in Table 8. To achieve maximum accuracy, we use 28 epochs with batch size 32 for the DOSSPRE (Binary Value) dataset. We achieve 85.3% accuracy for the DOSSPRE (Binary Value) dataset. We see that to maximize accuracy, the epoch number gradually increases. For the DOSSPRE (Multi-label) dataset, we achieve a maximum accuracy of 74.2% with epoch

number 40 and batch size 64. High accuracies in Table 8 show that the hybrid model accurately predicts the SR and NSR classes.

Table 8: Performance metrics values of all datasets according to different parameters

Serial no.	Parameters	PROMISE_exp binary dataset	DOSSPRE binary dataset	DOSSPRE multi-label dataset
1	Epoch	20	28	40
2	N_Estimators	350	200	300
3	Batch Size	32	32	64
4	Maximum Depth	3	3	3
5	Random State used	42	42	42
6	CNN + XGBoost Accuracy	**0.873**	**0.853**	**0.742**
7	Precision	0.666	0.845	0.772
8	Recall	0.666	0.717	0.742
9	F1 Score	0.666	0.775	0.744

Our hybrid model precisely predicts both classes, SR and NSR. The confusion matrix shows that the maximum accuracy is attained in the SR class with the positive label. The performance of our model improves with batch sizes 32 and 64 and epoch numbers 20, 28, and 40. The recall value also increased from 66.6% to 74.2.1% across all datasets, which shows that our proposed model perceives the SR accurately 87.3% of the time. Likewise, the highest accuracy, 87.3%, was achieved in the PROMISE_exp dataset with 20 epoch numbers and 32 batch sizes. The above table shows that the accuracy improved as batch size increased. Our proposed hybrid model gives the highest accuracy (87.3% and 85.3%) for binary classification of security and non-security factors.

Our results disclose a sole connection between different NFRs and security requirements. Contrasting previous literature studies, our proposed model used the classification method of security requirements that improves the security requirement engineering model. Our proposed hybrid model gives exclusive results as compared to literature studies. The proposed hybrid model attains 87.3% accuracy for the PROMISE_exp dataset and 85.3% for the DOSSPRE dataset, which is the maximum compared to the previous research studies. The study [40] applied CNN and SVM models to classify online datasets. It gained 85% accuracy, while the study [41] addressed the classification of NFRs from application review datasets using the CNN model and found 80% accuracy with the CNN model for NFR classification. The study [42] used a multiclass CNN model for the functional requirements classification and achieved a maximum of 77% accuracy. One study also used the XGBoost algorithm for accurate classification across different datasets and gained 77.95% accuracy [43]. The latest study [44] used the combination of CNN and XGBoost models, like our methodology, and achieved an accuracy of 84.2% with CNN and 83.6% with the XGBoost model. Our proposed methodology achieved the highest accuracy of 87.3% with the combination of CNN and XGBoost models, which has the highest accuracy compared to all of the above studies. Most of the research studies used only one dataset to compute the performance. However, in our proposed study, we used three datasets to analyze the behavior of the proposed hybrid model. By using a large number of epochs and a big batch size, we achieved higher performance metrics compared to previous studies. We recognized 10 different NFRs relevant to security requirements in both datasets. We mapped all the requirements of both datasets to 2 categories (SR and NSR) that had not been examined in the literature studies. The enhancement in

performance metric values presents the main research input towards mapping the relevant NFRs to the security requirement category and categorizing those requirements using the hybrid deep learning model. The results of our hybrid model show the revolution in security requirement engineering, which could lead to the development of applications that can protect systems from future security threats.

5.3 Discussion

Our proposed study emphasizes classifying security requirements by a hybrid deep learning model. Due to this drive, we use the PROMISE_exp and DOSSPRE datasets extensively in the software research community. Both datasets are well-tested datasets by many literature studies [52]. To contribute to the research, the DOSSPRE and PROMISE_exp datasets were extended. PROMISE_exp dataset consists of 969 different types of requirements. We extend this dataset by merging 61 more new requirements from the healthcare specification document. The extended PROMISE_exp dataset comprises 1030 requirements, 552 different nonfunctional and 478 functional requirements. Before the extension of datasets, the dataset contains 11 different types of NFRs. After extension, the dataset includes 18 types of NFRs. The DOSSPRE dataset contains 1316 labeled requirements. After the extension, it includes 1400 requirements. We analyze all security aspects that can directly influence other nonfunctional requirements. After doing literature research, we found ten nonfunctional requirements directly or indirectly related to security requirements. By merging the requirements into two groups, we find security and non-security requirement classes. Experts analyze all the requirements and suggest security category labels for certain NFRs. After labeling, the PROMISE_exp dataset contains 184 SR and 846 NSR labels. Meanwhile, the DOSSPRE dataset contains 524 SR and 876 NSR labels. After that, we do feature extraction of both extended datasets. We combine CNN and XGBoost models to classify precise security factors. The proposed model performs tokenization and padding on the datasets and creates training and testing datasets. We define, train, and compile the CNN and XGBoost models on the datasets to find the model's all-performance metric values. We select epoch numbers 20, 28, and 40 with batch sizes 32, 32, and 64. Our proposed model does not overfit due to the selection of this epoch value. To check the overfitting issue of the model, we inspect the training and validation losses and accuracies across numerous epochs. The problem of overfitting occurs when the proposed model studies to execute fine on the training data, but for the unobserved data, it fails. For the datasets used in our study, we have trained the model for several epochs and have numerous training and validation accuracies and losses. The validation accuracy for the PROMISE_exp dataset gradually improves from 0.81 to 0.86, which is promising. Likewise, the validation accuracy for the DOSSPRE (binary) dataset also gradually increases from 0.64 to 0.85 and 0.64 to 0.78 for the DOSSPRE (multi-label) dataset. Confusion metrics present that the proposed hybrid model attains maximum accuracy. It is observed that our proposed study attained the maximum accuracy of 87.3% and 85.3% with epoch numbers 20 and 28 and batch size 32, which was not achieved before. Two studies used the CNN model for security requirements classification for the PROMISE_exp and DOSSPRE datasets individually and achieved an overall accuracy of 84% and 85%. Our model is novel in using multiple datasets and achieving the highest accuracy. Our model produces a significant impact by increasing the accuracy to 87.3% by using epoch numbers 20 and 32 batch size, a significant improvement of our proposed research. The evaluation of the proposed hybrid model shows improvements in accuracy, yielding an improvement in accuracy and all other performance parameters compared to the previous studies.

We have achieved 87% and 85.3% accuracy for both datasets, which is good enough. In cybersecurity systems or national security systems, these accuracies might not be good enough. However, we are dealing with datasets that contain mostly less critical healthcare requirements. In the healthcare requirements scenario, these accuracies are acceptable. CNN and XGBoost models are very strong models. CNN is mostly used for image-type data, while XGBoost mainly focuses on tabular data. In the case of data on security

requirements, achieving 87% accuracy could be seen as a good result due to the complex domain. In our proposed methodology, we deal with all nonfunctional requirements that fall under the scope of security. We have categorized 10 NFRs into the security category. But there could still be other NFRs that we have missed and come under the security requirements. In this case, achieving this level of accuracy shows the better performance of our proposed model. The healthcare domain is one of the crucial domains that strictly follows the standards and rules. By keeping and classifying healthcare requirements with those standards and attaining this level of accuracy is a great achievement. According to the previous studies, 87% and 85.3% accuracies are the highest achieved for security requirements classification tasks yet. Integrating security features into RE is very beneficial in terms of efficiency and system protection. It gives the system an immunity to resist security attacks. Integrating security features with requirement engineering makes software projects more reliable and efficient. Security requirement engineering is valuable for making secure software systems that provide integrity and confidentiality simultaneously. Integrating security factors in requirement engineering benefits us in classifying the relevant NFRs into SR and NSR classes.

Requirements such as authentication, authorization, and confidentiality are the major requirements that influence security requirements. Authentication is associated with security as it allows users to enter the system. Authorization also specifies the users' access rights. Immunity also ensures how much a system can resist security attacks and perform normal operations. Availability ensures that a system is safe from security attacks and available to users. Mapping different requirements with security requirements makes the software system competent and reliable. In security requirements, such software systems can be created to fulfill the security features, which is the most important feature of any software system. Risks can also be identified through effective security requirement engineering. Security eliminates all the possible risks associated with the protection and confidentiality of software systems. Integrating security features in the software engineering process aids us in creating systems that fulfill functional requirements and influence all security aspects of the system. Many business organizations follow security parameters in their software projects to protect the system and gain maximum benefit from it. Security requirements engineering fosters technical advancements in software development, focusing on safeguarding systems against cyber threats, enhancing data scalability, and ensuring compliance with industry standards.

5.4 Complexities and Limitations

The hybrid model for security requirements classification produces many complexities and limitations. The combination of CNN and XGBoost models needs a careful merging strategy to ensure that data imported to the XGBoost model is compatible and appropriate with the attributes learned from CNN. The hybrid model also produces complexity in the preprocessing step as data first passes through CNN and then is formatted according to the XGBoost model, creating more complexity in the system. The training process of the hybrid model is also crucial as it requires additional resources and computation power. The hybrid model contains CNN and XGBoost models requiring proper maintenance and upgrade. Upgrading the hybrid model according to the changes is also challenging. It produces many complexities and ambiguities in the model. A large labeled dataset is required to get better performance from the CNN model. Normally, security-related data is imbalanced, and CNN requires more effort to train it accordingly. One of the major limitations of the CNN model is its latency in requirements classification tasks. The integration of CNN with XGBoost will also produce a significant delay in data processing. Many organizations have different regulatory and legal requirements. The hybrid model can face problems in meeting those standards and requirements. The hybrid model has been applied to two common datasets of the software engineering domain. The results can be compromised if we apply the model to multiple datasets from different domains. Security requirements contain deep semantics that are hard to understand. CNN and XGBoost might not be useful in understanding

the true semantics of security requirements. CNN model produces an overfitting issue when dealing with smaller datasets. The overfitting issue also occurs in the XGBoost model if there is no cross-validation. Scalability is also crucial in the hybrid model. As security data increases, the hybrid model faces difficulty scaling efficiently without compromising the performance. CNN and XGBoost models are like black-box models. It is very difficult to explain their classification tasks. The combination of CNN and XGBoost models strengthens the security requirements classification task, but it also introduces many limitations and complexities, such as data processing, model integration, and maintenance. These complexities must be resolved before applying this hybrid model to real-time applications.

5.5 Threats to Validity

The major threats that exist in the proposed study are given below.

5.5.1 External Threat

It deals with the hybrid model results and findings. We have applied the proposed hybrid model on the PROMISE_exp and DOSSPRE to mitigate this threat. The results generated by these datasets reduced this threat and presented a broad view of the proposed model.

5.5.2 Internal Threat

It relates to the internal validity of the study. The selection of the CNN, and XGBoost models and the relevant security requirements formed this type of threat. However, we have tried to mitigate this threat by providing explanations for selecting those models and security requirements. These strategies mitigated the risk of personal biases in selecting the hybrid model.

5.5.3 Evaluation Threat

Our study used validation accuracy, loss, and performance metrics values. A confusion matrix is also used to show the proposed model's performance. Performance metrics give the value of the hybrid deep learning model [53]. However, these performance metrics values may not achieve security requirements elicitation and classification efficiency. Robust evaluation metrics can be used to resolve this threat.

6 Conclusion

Our proposed novel methodology focuses on different requirements and analyzes their impact on the system's security. We observed that 10 types of nonfunctional requirements lie under the security requirement category. Using different deep learning models to classify the requirements into security and non-security categories, our study emphasizes a significant innovation in understanding the correct meaning of requirements and rendering their security goals. After observation, two categories were formed. We extended DOSSPRE and PROMISE_exp datasets according to those categories to perform the requirements for text classification. These categories are SR and NSR. Labeling is performed on all the DOSSPRE and PROMISE_exp datasets. We use the combination of CNN and the XGBoost model, having the ability to learn for security requirements classification. We use the hybrid deep learning model to recognize the meaning of the requirement within security requirement engineering. We observe and monitor the model's performance by evaluating its influence on requirements classification. Our proposed model provides worthwhile and better results. We attained the highest accuracy of 87.3% and 85.3% with epoch numbers 20 and 28 and batch size 32 for PROMISE_exp and DOSSPRE datasets, which is the highest accuracy till now. The confusion matrix also shows an acceptable degree of accuracy for both datasets. The metrics values show enhancement

in all parameters compared to the previous literature studies. Moreover, reducing false positives in the confusion matrix shows better predictive ability. The hybrid deep learning model shows high strength in these findings by calculating the best metric parameter values. This proposed model helps as a foundation for security requirements engineering research. The correct classification of security requirements through the hybrid deep learning model displays the model's efficiency with both datasets. We can attain security goals by integrating all security requirements into the software system. Our hybrid model guarantees that the system meets technical and all types of security requirements.

In the future, we can increase the accuracy by observing more NFRs related to security or taking more datasets to present the efficiency of the hybrid model with numerous datasets. We can execute more model tuning, which depends on how important each indicator is in a given software system. The performance of the hybrid model can also be increased by optimizing the CNN and XGBoost models. This can be done by decreasing model size and increasing computational efficiency. The combination of CNN and XGBoost models holds great potential for classification tasks related to security requirements. Future research can focus on handling different imbalanced datasets, increasing model interpretability, and increasing performance. This hybrid model can be adapted in emerging security research areas like edge computing, and cyber security. Combining CNN with recurrent neural networks (RNNs) will also improve security classification tasks by leveraging sequential attributes. We can improve the accuracy by investigating domain-specific feature extraction methods for security requirements. The hybrid model will also help to detect the unexplored security requirements that have not been labeled in the datasets. Researchers can handle modern security issues and challenges by improving security requirements classification tasks.

Acknowledgement: Not applicable.

Funding Statement: The authors of this study extend their appreciation to the Researchers Supporting Project number (RSPD2025R544), King Saud University, Riyadh, Saudia Arabia.

Author Contributions: Shoaib Hassan: Conceptualization, Methodology, Writing Original Draft Preparation. Qianmu Li: Supervision, Writing, Reviewing, Editing, Project Administration. Muhammad Zubair: Formal Analysis, Writing, Reviewing, and Editing. Rakan A. Alsowail: Data Visualization, Writing, Reviewing, Editing, Funding Acquisition. Muhammad Umair: Formal Analysis, Writing, Reviewing, and Editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, S. H., Q. L., upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Almulihi AH, Alassery F, Khan AI, Shukla S, Gupta BK, Kuma R. Analyzing the implications of healthcare data breaches through computational technique. Soft Comput. 2022;32(3):1763–79. doi:10.32604/iasc.2022.023460.
2. Salini P, Kanmani S. Model oriented security requirements engineering (MOSRE) framework for web applications. In: Advances in computing and information technology. Berlin: Heidelberg; 2013.
3. Arogundade T, Misra S, Abayomi-Alli OO, Fernandez-Sanz L. Enhancing misuse cases with risk assessment for safety requirements. IEEE Access. 2020;8:12001–14. doi:10.1109/ACCESS.2019.2963673.
4. Jung JW, Park SH, Lee SW. A tool for security requirements recommendation using case-based problem domain ontology. In: 2021 IEEE 29th International Requirements Engineering Conference (RE); 2021; Notre Dame, IN, USA.

5. Riaz M, King J, Slankas J, Williams L. Hidden in plain sight: automatically identifying security requirements from natural language artifacts. In: 2014 IEEE 22nd International Requirements Engineering Conference (RE); 2014; Karlskrona, Sweden.
6. Slankas J, Williams L. Automated extraction of non-functional requirements in available documentation. In: 2013 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE); 2013.
7. Mukalazi A, Boyaci A. The Internet of Things: a domain-specific security requirement classification. In: 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA); 2022; Ankara, Turkey; p. 1–8.
8. Dias Canedo E, Cordeiro Mendes B. Software requirements classification using machine learning algorithms. Entropy. 2021;22(9):1057.
9. Mengmeng L, Peng L. Automatic classification of non-functional requirements from augmented app user reviews. New York, NY, USA: Association for Computing Machinery; 2017. p. 344–53.
10. Rouland Q, Gjorcheski S, Jaskolka J. Eliciting a security architecture requirements baseline from standards and regulations. In: 2023 IEEE 31st International Requirements Engineering Conference Workshops (REW); 2023; Hannover, Germany.
11. Shukla A, Katt B, Nweke LO, Yeng PK, Weldehawaryat GK. System security assurance: a systematic literature review. Comput Sci Rev. 2022;45(1):100496. doi:10.1016/j.cosrev.2022.100496.
12. Kim J, Lee SW. Understanding and recommending security requirements from problem domain ontology: a cognitive three-layered approach. J Syst Softw. 2020;169(1):110695. doi:10.1016/j.jss.2020.110695.
13. Ferrari A, Spagnolo GO, Gnesi S. PURE: a dataset of public requirements documents. In: 2017 IEEE 25th International Requirements Engineering Conference; 2017.
14. Kadebu P, Thada V, Chiurunge P. Security requirements extraction and classification: a survey. In: 2018 3rd International Conference on Contemporary Computing and Informatics (IC3I); 2018; Gurgaon, India. p. 129–34.
15. Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET); 2017; Antalya, Turkey.
16. Dai D. An introduction of CNN: models and training on neural network models. In: 2021 International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR); 2021; Shanghai, China.
17. Li H, Cao Y, Li S, Zhao J, Sun Y. XGBoost model and its application to personal credit evaluation. IEEE Intell Syst. 2020;35(3):52–61. doi:10.1109/MIS.2020.2972533.
18. Wu J, Li Y, Ma Y. Comparison of XGBoost and the neural network model on the classbalanced datasets. In: 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer; 2021; Greenville, SC, USA; p. 457–61.
19. Hassan S, Li Q, Aurangzeb K, Yasin A, Khan Ali J, Shahid Anwar M. A systematic mapping to investigate the application of machine learning techniques in requirement engineering activities. CAAI Trans Intell Technol. 2024;1(2):1–22. doi:10.1049/cit2.12348.
20. Yahya S, Kamalrudin M, Sidek S, Jaimun M, Yusof J, Hua AK, et al. A review paper: security requirement patterns for a secure software development. In: 2019 1st International Conference on Artificial Intelligence and Data Sciences; 2019; Ipoh, Malaysia; p. 146–51.
21. Singh P, Singh D, Sharma A. Classification of non-functional requirements from SRS documents using thematic roles. In: 2016 IEEE International Symposium on Nanoelectronic and Information Systems, iNIS 2016; 2016
22. Hassan S, Li Q, Zubair M, Alsowail R, Qureshi M. Unveiling the correlation between nonfunctional requirements and sustainable environmental factors using a machine learning model. Sustainability. 2024;16(14):5901. doi:10.3390/su16145901.
23. Ham JVD. Toward a better understanding of cybersecurity. Digital Threats: Res Practice. 2021;2(3):1–3. doi:10.1145/344244.
24. Alonge Y, Arogundade OT, Adesemowo K, Ibrahalu FT, Adeniran OJ, Mustapha AM. Information asset classification and labelling model using fuzzy approach for effective security risk assessment. In: 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS); 2020.

25. Silva MA, Danziger M. The importance of security requirements elicitation and how to do it. In: PMI® Global Congress 2015—EMEA; 2015; London, UK.
26. Viega J. Building security requirements with CLASP. In: Proceedings of the 2005 Workshop on Software Engineering for Secure Systems—Building Trustworthy Applications; 2005. p. 1–7.
27. Halimaa AA, Sundarakantham K. Machine learning based intrusion detection system. In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI); 2019; Tirunelveli, India.
28. Khanneh S, Anu V. Security requirements prioritization techniques: a survey and classification framework. Software. 2022;1(4):450–72. doi:10.3390/software1040019.
29. Rjaibi N, Rabai LB. Developing a novel holistic taxonomy of security requirements. Procedia Comput Sci. 2015;62(1):213–20. doi:10.1016/j.procs.2015.08.442.
30. Heiko H, Andreagiovanni R. Scalability in computing and robotics. IEEE Trans Comput. 2022;71(6):1453–65. doi:10.1109/TC.2021.3089044.
31. Lou Y. Research of model-based reliability requirement transformation technology. In: 2023 Global Reliability and Prognostics and Health Management Conference; 2023; Hangzhou, China.
32. Houmb SH, Islam S, Knauss E, Jürjens J, Schneider K. Eliciting security requirements and tracing them to design: an integration of common criteria, heuristics, and UMLsec. Requir Eng. 2010;15(1):63–93. doi:10.1007/s00766-009-0093-9.
33. Dafaalla H. Deep learning model for selecting suitable requirements elicitation techniques. Appl Sci. 2022;12(18):9060. doi:10.3390/app12189060.
34. Vijayvargiya S, Kumar L, Murthy LB, Misra S. Software requirements classification using deep- learning approach with various hidden layers. In: 2022 17th Conference on Computer Science and Intelligence Systems; 2022; Sofia, Bulgaria; p. 895–904.
35. Wang S, Huang L, Gao A, Ge J, Zhang T, Feng H, et al. Machine/deep learning for software engineering: a systematic literature review. IEEE Trans Softw Eng. 2023;49(3):1188–231. doi:10.1109/TSE.2022.3173346.
36. Arora C, Grundy J, Abdelrazek M. Advancing requirements engineering through generative AI. In: Generative AI for effective software development. Cham: Springer; 2024.
37. Talele P, Phalnikar R. Classification and prioritisation of software requirements using machine learning—A systematic review. In: 2021 11th International Conference on Cloud Computing; 2021; Noida, India: Data Science & Engineering.
38. Chen X, Huang M, Feng S, Chen Y, Li W. Automatic classification of user requirements information based on convolutional neural network. In: 2021 5th Asian Conference on Artificial Intelligence Technology; 2021; Haikou, China.
39. Bisi Manjubala KK. CNN-BPSO approach to select optimal values of CNN parameters for software requirements classification. In: 2020 IEEE 17th India Council International Conference; 2020; New Delhi, India.
40. Audat A, Saraireh M, Shatnawi M. Intelligent requirements engineering: applying machine learning for requirements classification. In: 2023 14th International Conference on Information and Communication Systems; 2023; Irbid, Jordan.
41. Hidayat T, Rochimah S. NFR classification using keyword extraction and CNN on app reviews. In: 2021 4th International Seminar on Research of Information Technology and Intelligent Systems; 2021; Yogyakarta, Indonesia; p. 211–6.
42. Jp S, Menon VK, Soman K, Ojha AKR. A non-exclusive multi-class convolutional neural network for the classification of functional requirements in AUTOSAR software requirement specification. IEEE Access. 2022;10:117707–14. doi:10.1109/ACCESS.2022.3217752.
43. Ramraj S, Nishant U, Sunil R, Banerjee S. Experimenting XGBoost algorithm for prediction and classification of different datasets. Int J Control Theory Appl. 2016;9(40):651–62.
44. Thongsuwan S, Jaiyen S, Padcharoen A, Agarwal P. ConvXGB: a new deep learning model for classification problems based on CNN and XGBoost. Nucl Eng Technol. 2021;53(2):522–31. doi:10.1016/j.net.2020.04.008.
45. Kadebu P, Sikka S, Tyagi RK, Chiurunge P. A classification approach for software requirements towards maintainable security. Sci Afr. 2023;19(1):e01496. doi:10.1016/j.sciaf.2022.e01496.

46. Jindal R. Automated classification of security requirements. In: 2016 International Conference on Advances in Computing, Communications and Informatics; 2016.
47. Fong V. RE data challenge: requirements identification with Word2Vec and TensorFlow. In: 2017 IEEE 25th International Requirements Engineering Conference; 2017.
48. Abad ZSH, Karras O, Ghazi P, Glinz M, Ruhe G, Schneider K. What works better? A study of classifying requirements. In: 2017 IEEE 25th International Requirements Engineering Conference; 2017.
49. Alashqar AM. Studying the commonalities, mappings and relationships between nonfunctional requirements using machine learning. Sci Comput Program. 2022;218(11):102806. doi:10.1016/j.scico.2022.102806.
50. Riaz M, Mendes E, Tempero E. A systematic review of software maintainability prediction and metrics. In: 2009 3rd International Symposium on Empirical Software Engineering and Measurement; 2009; Lake Buena Vista, FL, USA.
51. GeeksforGeeks. Evaluation metrics in machine learning. Uttar Pradesh, India: Sanchhaya Education Private Limited; 2013. Available from: https://www.geeksforgeeks.org/metrics-for-machine-learning-model/. [Accessed 2024 Jun 26].
52. Lima M, Valle V, Costa E, Lira F, Gadelha B. Software engineering repositories: expanding the PROMISE database. In: SBES '19: Proceedings of the 33rd Brazilian Symposium on Software Engineering; 2019 Sep; Salvador, Brazil; p. 427–36.
53. Zhou J, Gandomi AH, Chen F, Holzinger A. Evaluating the quality of machine learning explanations: a survey on methods and metrics. Electronics. 2021;10(5):593. doi:10.3390/electronics10050593.