International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 9, pp 1-9, September-2025

https://doi.org/10.47001/IRJIET/2025.909001

A Binary Classification of Software Security Requirements: A Deep Learning Approach

Landry Giraud Wandji T.

Ph.D. Student at the Hochschulinstitut Schaffhausen, Switzerland

Abstract **- The current work proposes a hybrid deep learning approach for classification of software security requirements based on the DOSSPRE dataset. [12] Publish a work on software requirements classification where a software requirement dataset named DOSSPRE have been generated. They mentioned that the non classification of security requirements has been identified as a major source of security concerns in the software development process. For that reason, they give an approach to classify security requirement, but they didn't consider deep learning as classification technique. The literature review show that, the application of machine learning techniques for software requirements classification is increasing these last years but in the most of application cases classical assemblage techniques are prioritized and deep learning techniques are used in just few cases, due to his complexity and the necessary resources like the needed amount of input data and a high computational power to achieved the expected results. This study contributes to fill the gap in application of deep learning technique for software security requirement classification. A novel approach for hybrid machine learning model is proposed based on naïve bayes model (MNB_Model) and a deep learning model. Three hybrid machine learning models are compared: a deep neural network (DNN-MNB_Model), a convolutional neural network (CNN-MNB_Model) and a recurrent neural network (RNN-MNB_Model). The results of the comparison show that the CNN-MNB_Model get the best performance with 80.2% accuracy.**

Keywords: requirement engineering, deep learning, machine learning.

I. INTRODUCTION

Due to the continuously increase of security threat on developed system this recent years a multiplication of solution have been developed in other to solve the identified software security problems, but the existing solution are only partially applicable to different systems. The involved systems are including web applications, banking systems, applications for healthcare and e-commerce [1] [2]. The identification and mitigation of software security threat are becoming furthermore challenging because of the multiplication of recurrent attacks [11]. To overcome those increasing challenges, it becomes much more important to develop and implement specifics strategies corresponding to the identified security issue.

In addition to security measure at the deployment stage [22] - [10], the protection of the software during the development stage will assure his robustness and his ability to withstand possible attacks. For that reason, the requirement engineering responsible shall include the management of security requirement (SR) early in the software development life cycle. This approach will prioritize software security requirements and will focus early on necessary SR for the development of a secure software system.

Security requirements belong to the category of non-functional requirement (NFR) that are the most of time taken form industry application or standards which specify security features that a software must have, in other to be protected against possible attacks [4] - [15]. Together with other software requirements, they are also acquired through user stories, software assessments, and documentation provided by various stakeholders. The process of gathering SR through the security requirements elicitation procedure aids in the security requirements analysis process [19].

Classifying security requirements is a step in the analysis process that involves assigning each security requirement to a particular classification model for the software's security services [16]. The majority of studies concentrate on both functional and non-functional requirements, including security, though usually in a broad sense [13] - [14]. Software requirements can be categorized more quickly and easily thanks to a variety of machine learning and natural language processing techniques [21] - [5]. The most of work are focusing on the application of supervised learning for requirement classification, so the consideration of deep learning techniques is not frequent.

Therefore, this study presents the application of a hybrid deep learning technique for automated classification of software requirements as security requirement (SR) and non-security requirements (NSR). The DOSSPRE dataset generate by Kadebu et al [12] containing around 1400 software requirements will be used for this classification task. This paper is structured with a literature review discussed in the

ISSN (online): 2581-3048
Volume 9, Issue 9, pp 1-9, September-2025
https://doi.org/10.47001/IRJIET/2025.909001

section 2, then a description of the material and methodology is done in the section 3. In the section 4 the results of the proposed methodology are presented. This works is concluded with a discussion of the results in the section 5.

II. RELATED WORKS

Security requirements are commonly cited as might be reason of a low system quality if these are not adequately addressed, but there is very few research on the classification of software security requirements. The Comprehensive, Lightweight Application Security Process (CLASP) [23] outlines secure software development processes and designates Authentication and Integrity, Confidentiality, Availability, and Accountability as essential security services, with Authorization (Access Control) being the most fundamental. This highlights the importance of the prioritization of security requirement in the requirement management process.

Firesmith classified security requirements into 12 categories which are "Identification, Authentication, Authorization, Immunity, Integrity, Intrusion Detection, Non-Repudiation, Privacy, Security Auditing, Survivability, Physical Protection, and System Maintenance Security Requirements" [7]. This classification introduces System Maintenance as a Security Requirement and is fairly fine-grained. However, this classification method did no consider the application of machine learning technique and the usage of datasets to support the classification method. Therefore, several studies have classified software requirement using different techniques as natural language processing and supervise learning techniques.

Which machine learning techniques are mostly used in RE activities according to the number of publications is presented by [9]. It is observed that DT and SVM algorithms are most frequently utilized during the requirement specification stage. While SVM, DT, and NB are primarily used in the requirements validation phase, NB, KNN, and SVM are primarily used in the RA and classification phase. They conclude from their observations that, in contrast to the requirements specification and validation phases, the requirement analysis and classification phase offer a wide range of supervised learning algorithms.

[17] Compares and extends ML algorithms to classify requirements in terms of precision and accuracy. It mentions that, from their literature survey decision tree (DT) algorithm can identify different requirements and outperforms existing ML algorithms. As the number of features increases, the accuracy using the DT is improved by 1.65%. To overcome the limitations of DT, they propose a multiple correlation coefficient based DT algorithm. When compared to existing ML approaches, the results showed that the proposed algorithm can improve classification performance. The accuracy of the proposed algorithm is improved by 5.49% compared to the DT algorithm. However, this study does not investigate the application of deep learning techniques.

[8] Proposed automatic NFR classification approach for quality software development by combining machine learning feature extraction and classification techniques. An empirical study with machine learning algorithms such as MNB, Gaussian Naïve Bayes (GNB), KNN, Decision tree, Support Vector Machine (SVM), Stochastic Gradient Descent SVM (SVM SGD) and four feature selection approaches have been applied to automatically classify NFR for finding out the best pair. The experiments were measured with statistical analysis including precision, recall, F1-score, and accuracy of the classification results through all the combinations of the techniques and algorithms. It is found that, SGD SVM classifier achieves best results where precision, recall, F1-score, and accuracy reported as 0.66, 0.61, 0.61, and 0.76 respectively. Therefore, deep learning techniques were not considered by this study.

[6] Shows a comparison among the text feature extraction techniques, and machine learning algorithms to the problem of classifying software requirements into functional requirements and non-functional requirements, and the subclasses of non-functional requirements and which machine learning algorithm provides the best performance for the requirements classification task. The data used to perform the research was the PROMISE_exp. The algorithms used for classification were Logistic Regression, Support Vector Machine, Multinomial Naive Bayes and k-Nearest Neighbors. It has been noticed that the use of TF-IDF followed by the use of LR had a better classification result to differentiate requirements, with an F-measure of 0.91 in binary classification (tying with SVM in that case), 0.74 in NF classification and 0.78 in general classification. Deep learning for requirements classification was no in scope of this work.

Regarding deep learning applications for requirement engineering activities, there are existing number of models that can be used for the classification of requirements into different categories [24]. Helpful for obtaining valuable information from the requirements are transformers such as BERT and GPT. Recurrent neural networks (RNN) support the RE process by carrying out tasks related to text processing and classification. Convolutional neuronal networks (CNN) is also efficiently used in requirement classification [3]. Graph neuronal networks (GNN) are applicable on requirement prioritization and traceability activities.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 9, pp 1-9, September-2025

https://doi.org/10.47001/IRJIET/2025.909001

[18] Presents a deep-learning framework for NFR classification. This framework leverages a more profound architecture that naturally captures feature structures, possesses enhanced representational power, and efficiently captures a broader context than shallower structures. To evaluate the effectiveness of the proposed method, an experiment was conducted on two widely-used datasets, encompassing 914 NFR instances. Performance analysis was performed on the applied models, and the results were evaluated using various metrics. Notably, the proposed model outperforms the other models in classifying NFR, achieving precision between 81 and 99.8%, recall between 74 and 89%, and F1-score between 83 and 89%. These results showcasing the potential of the proposed deep learning framework for advancing the field of NFR analysis and classification, but the focus on security requirement is not in the scope of this work.

[20] Explores the potential of filter-based feature selection techniques and traditional machine learning classifiers, it investigates combined potential of traditional feature selection and diverse types of deep learning predictors. Then it presents a hybrid computational predictor namely that reaps combine benefits of traditional feature selection and a deep learning predictor based on attention mechanism. Over two public benchmark datasets, large-scale experimental results reveal feature selection not only improves predictive performance of traditional machine learning predictors, but it also improves performance of deep learning predictors. The proposed predictor was compared with state-of the-art functional and non-functional requirements classification predictors over Promise and Promise-exp datasets. Therefore, this work does no focus on classification of security requirement.

The table 1 shows the comparison of the relevant literature. They are different machine learning approaches related to the classification of software requirements. The main classification task is to classify the requirements into FR and NFR, so the focus on software security requirement is missing. Another observation is that, just a small part of the existing paper describes deep learning technique as a solution to solve requirement engineering problems. The most of the study present mainly the application of classical supervised learning technique on requirement engineering activities. So, this paper will contribute to fill the gap in the field of software security requirement classification using deep learning techniques. The DOSSPRE datasets, a new available data for the requirement engineering community will be used for this requirement engineering activity.

Table 1: Comparison of the relevant literatures

Related work	Req. class.	Binary class.	Security req.	Machine learning	Deep learning	Hybrid deep learning
[7]	Yes	No	Yes	No	No	No
[17]	Yes	Yes	No	Yes	No	No
[8]	Yes	No	Yes	Yes	No	No
[6]	Yes	Yes	No	Yes	No	No
[18]	Yes	No	Yes	Yes	Yes	No
[20]	Yes	Yes	No	Yes	Yes	Yes
This work	Yes	Yes	Yes	Yes	Yes	Yes

III. METHODOLOGY

The classification of raw requirement into security and non-security requirements is becoming very important in the software development process. This task is done manually in the most of industry applications. Literature review on the topic show that there is a lot of effort done in the scientific community in other the automated this classification. But one of the main challenges is to develop a useful and efficient framework making the requirement classification task easier. Another challenge is to find corresponding datasets in other to train and test the build models.

Different software requirements datasets are existing in online repositories. The most of them are collection of functional and non-functional requirement without focus on software security requirement. The DOSSPRE dataset cover this gap by give a rich amount of software security requirements, useful for training and testing machine learning models. The DOSSPRE datasets are available in two versions, binary labels for the requirement classification are included in one version. The DOSSPRE dataset is a recent addition to the existing dataset in the requirements engineering community. It contains 876 non-security and 524 security requirements.

The target of the proposed novel approach is to build a hybrid deep learning model using the strength of a multinomial Naives Bayes Model (MNB_Model) and the efficiency of a deep learning model. The figure 1 shows the different steps of the proposed methodology. The first step

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Issue 9, pp 1-9, September-2025
https://doi.org/10.47001/IRJIET/2025.909001

will be a data preprocessing then, split the data into a train and test data set. The second step will be to build a Naive Bayes model. The third step will be to build deep learning models. Different deep learning architecture will be investigated in this work: a Deep Neural Network Model (DNN_Model), a Convolutional Neural Network Model (CNN_Model) and a Recurrent Neural Network Model (RNN_Model). The combination of the Naive Bayes algorithm with a deep leaning model to be trained and tested on the same DOSSPRE dataset will be the last step of this approach.

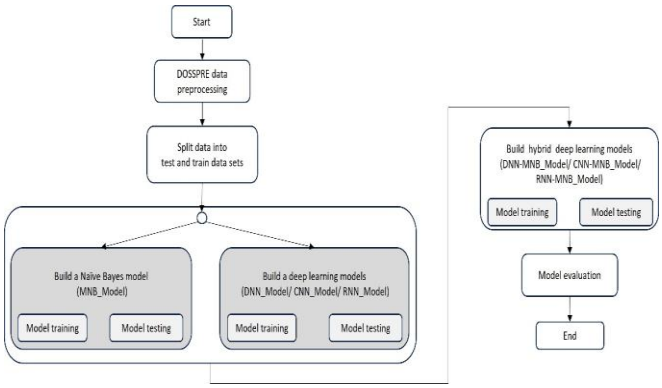

Figure 1: Proposed methodology for the hybrid deep learning approach

3.1 Collected data

The DOSSPRE dataset is a recent addition to the existing dataset in the requirements engineering community and will be use to implement the defined approach. It contains non-security and security requirements. It gives a rich amount of software security requirements, useful for training and testing machine learning models. The DOSSPRE datasets are available in two versions, binary labels for the requirement classification are included in one version.

3.2 Preprocessing

This session describes the data preprocessing. It includes the following sub steps:

- Data visualization
- Data cleaning
- Tokenization
- Stopwords removal
- Token normalization
- Vectorization

Before starting the cleaning step, it is important to visualize the data and have an overview about his content. The datasets have more than 1315 requirements labeled in two classes SR and NSR. A closer look inside the data present on the figure 2 61% NSR and 39% SR. This repartition makes the data as well balance for the use in machine learning research project.

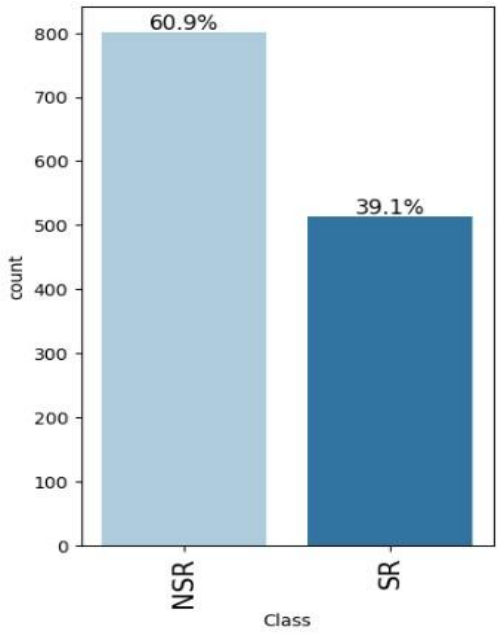

Figure 2: Percentage of NSR and SR inside the datasets

The data cleaning: Before we start handle with the data, we need to clean the data to get it all in a consistent format. We need to clean and convert our data into a matrix. In that step we also need to remove unnecessary columns. After that we can normalize the column "Class" in other to make the data easy to be used later by the machine learning algorithm. The data after cleaning and normalization steps are presented in the table 2.

Table 2: Data overview after data cleaning

Class	Requirements
1	The system shall detect sql injection and xss ...
0	The system shall analyze the detected vulnerab...
0	The system shall identify acceptable low false...
0	The system shall inspect ssl certificate
0	The system shall generate a report of the dete...

Tokenization: During this phase the sentences with the label requirement are tokenized, which turns them into a list of words.

Stopwords removal: After tokenization it is necessary to remove words which occurs very frequently but do not have possible value like: are, the, an etc. So, all the conjunction words, auxiliary verbs are removed from the requirement sentences.

Token normalization: In this step the morphological affixes are removed from the words, leaving only the root word. Converting various tokens to their most basic forms is known as token normalization. You can accomplish this by:

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 9, pp 1-9, September-2025

https://doi.org/10.47001/IRJIET/2025.909001

- Stemming: is the process of eliminating and substituting suffixes to reach the word's root form, or stem
- Lemmatization: This process yields a word's lemma, or base or dictionary form.

Features extraction: This step help to prepare the data in the needed form as input for the machine learning algorithms. For classical machine learning algorithms features are extracted through TF-IDF. The extracted features through TFID can also be used by an easy artificial neural network as ANN or DNN. But for deep learning model as CNN or RNN the features are extracted through embedding layer. The figure 3 shows the features extraction for the different models.

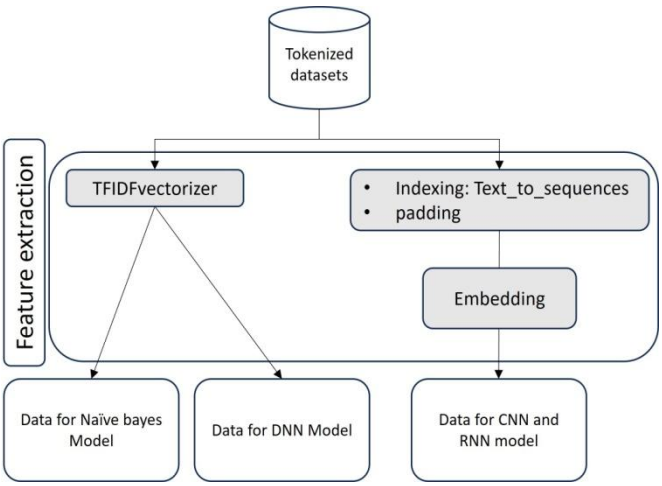

Figure 3: Feature extraction for machine learning and deep learning

Split the data into train and test data: After the preprocessing stage, the data are divided into train and text data. The figure 4 shows the shape of the splited data. We can see that the data are divided into 70% of training data and 30% of test data.

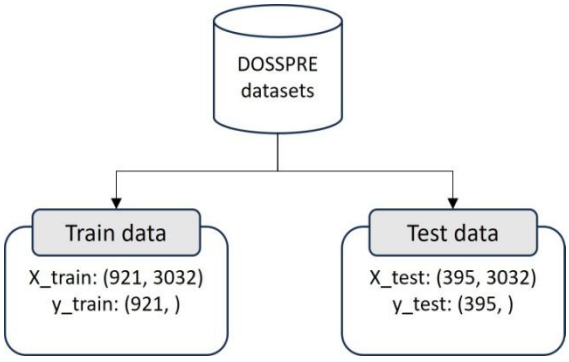

Figure 4: Splitting the data into train and test data

3.3 Build the hybrid models

The combination of the Naive Bayes model and a deep leaning model is supposed to fit together the strength of both model in other to get an improved classification of the software security requirements. The depending on the combined deep leaning model the hybrid deep learning model can be made using different methods. By using the CNN or the RNN model, the predicted result can be added to the predicted results of the MNB and the average will give the prediction from the hybrid model. Another possibility is to use the deep learning model for feature extraction and input the extracted features to the MNB model for the classification. The figure 5 describes the functionality of the DNN-MNB_model. When the model gets the input data, the first step is the feature extraction through the first layers of the deep neural model. Then the results of the features extraction are used for the classification by the MNB_Model.

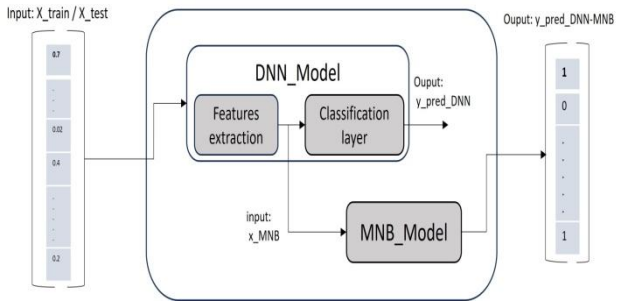

Figure 5: Structure of the DNN-MNB_Model

IV. RESULTS

4.1 Result of Naive Bayes Model (MNB_Model)

Multinomial Naives Bayes Model (MNB_Model) is built as a baseline in other to have a basic to be compare with the deep learning models. The train data are used to fit Naives Bayes Model. After the training, the model is tested using the test data. The MNB_Model achieved an accuracy of 78.2%, a F1 score of 77.2% and a precision of 78.1%. The figure 6 show the results of the builtMNB_Model.

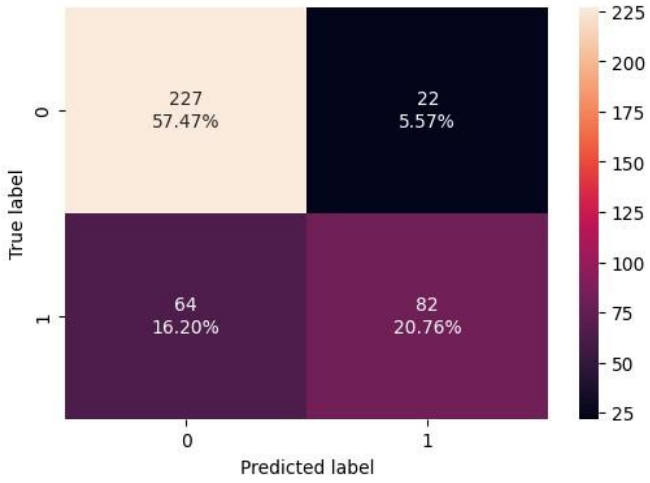

Figure 6: Confusion matrix if the MNB_Model

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 9, pp 1-9, September-2025

https://doi.org/10.47001/IRJIET/2025.909001

4.2 Results of deep learning Models

4.2.1 Results of Deep Neural Networks Model (DNN_Model)

In this subsection a deep learning model is built for the classification of security requirements. The target is to prepare the deep learning model for the combination with the naives bayes model MNB_Model. The table 3 presents the architecture of the deep learning model. The DNN_Model includes an input layer, which get the input data and feed them to the dense layers. After the features extraction in the first dense layers, the classification is done in the last layer.

Table 3: Architecture of the DNN_Model

Layer (type)	Output shape
input_layer	(None, 2846)
dense	(None, 10)
dense_1	(None, 20)
dense_2	(None, 10)
dense_3	(None, 1)

After a training and a validation of the DNN_Model, his result is evaluated in other to have an idea about the model accuracy. The result of the DNN_Model is show in the figure 7. The model achieves a validation accuracy of 79.4%.

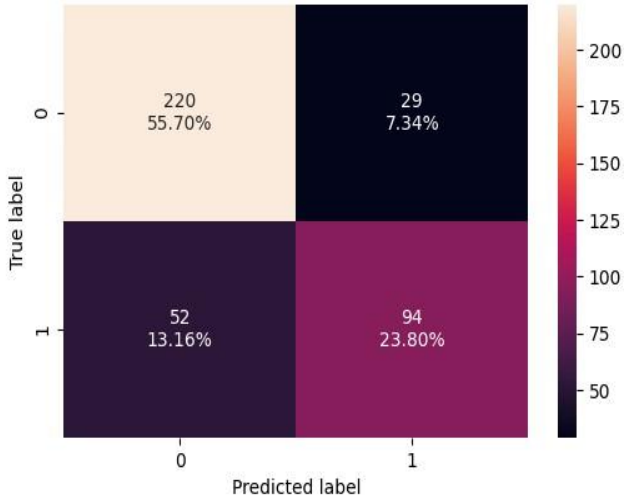

Figure 7: Result of the DNN_Model

4.2.2 Results of the Convolutional Neural Networks Model (CNN_Model)

In this subsection a convolutional neural network model is built for the classification of security requirements. The target is to prepare the deep learning model for the combination with the naives bayes model MNB_Model. The table 4 presents the architecture of the deep learning model.

Table 4: Architecture of the CNN_Model

Layer (type)	Output shape
embedding_3	(None, 2846, 128)
conv1d	(None, 2846, 128)
global_max_pooling1d_2	(None, 128)
dense_7	(None, 10)
dense_8	(None, 1)

After a training and a validation of the CNN_Model, his result is evaluated in other to have an idea about the model accuracy. The result of the CNN_Model is shown in the figure 8. The model achieves a validation accuracy of 81.2%.

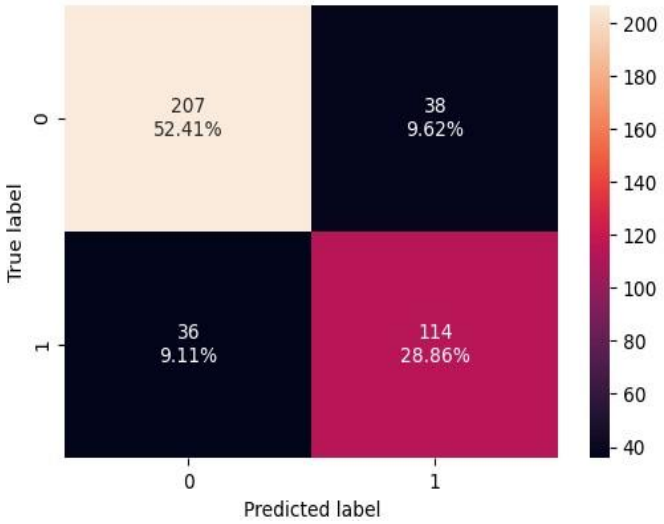

Figure 8: Result of the CNN_Model

4.2.3 Results of the Recurrent Neural Networks Model (RNN_Model)

In this subsection a recurrent neural network model is built for the classification of security requirements. The target is to prepare the deep learning model for the combination with the naives bayes model MNB_Model. The table 5 presents the architecture of the recurrent neural network model.

Table 5: Architecture of the RNN_Model

Layer (type)	Output shape
embedding_3	(None,2846,128)
spartial_dropout1d_1	(None, 2846, 128)
lstm_1	(None, 128)
dense_9	(None, 1)

After a training and a validation of the RNN_Model, his result is evaluated in other to have an idea about the model accuracy. The result of the RNN_Model is show in the figure 9. The model achieves a validation accuracy of 72.2%.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 9, pp 1-9, September-2025

https://doi.org/10.47001/IRJIET/2025.909001

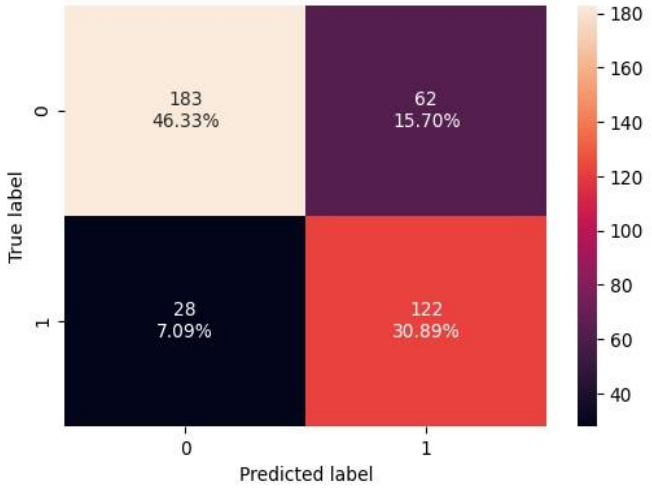

Figure 9: Result of the RNN_Model

4.3 Results of the hybrid models

The combination of the Naive Bayes model and a deep leaning model is supposed to fit together the strength of both model in other to get an improved classification of the software security requirements. The table 6 presents the results of the classification model with a direct comparison between the classical machine learning model, the deep learning model and the hybrid model.

Table 6: Results of the classification models

Model	F1-score (%)	Recall (%)	Accuracy (%)	Precision (%)
MNB	77.2	56.1	78.2	78.1
DNN	79.0	64.4	79.4	76.3
DNN-MNB	78.3	55.2	79.5	83.5
CNN	81.3	76	81.2	75
CNN-MNB	80.4	80.7	80.2	71.2
RNN	77.5	81.3	72.2	66.3
RNN-MNB	71.4	86.7	71.1	58

The results show that the applied classification models achieved good performance for the classification task (see figure 10). The MNB model taken as baseline performed with an accuracy of 78.2%. This performance is 6% better than the performance of the RNN model. The best performer for this classification is the CNN model with 81.2% accuracy. The combination of CNN and MNB also lead to a better performance as the baseline and the other deep learning model. The combination of the MNB and deep learning models does not really show a better performance compared to the deep learning models alone.

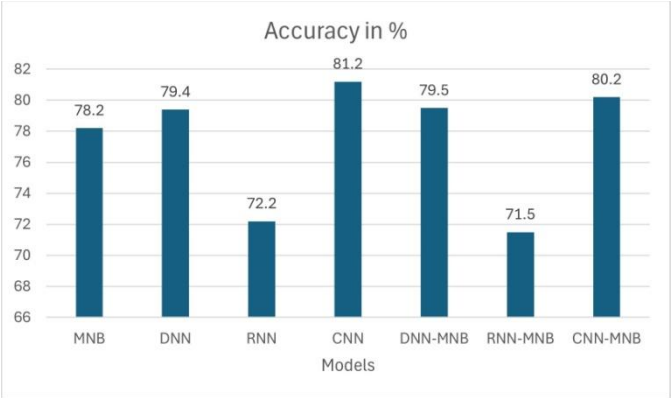

Figure 10: Accuracy of the implemented models

V. CONCLUSION

In this study we investigate the application of deep learning and hybrid deep learning model for a binary classification of software security requirements compare to a machine learning model based on multinomial naive bayes. The hybrid models were built by combining a deep learning model with the multinomial naive bayes model. The results were compared based on accuracy, precision, recall and F-score.

For this work the DOSSPRE datasets as new datasets in the requirement engineering community was use for the classification task. After the data preprocessing the approach of this work consist to a combination of a deep leaning models and a multinomial naives bayes model in other to use the strength of the deep learning model for the features extraction and the strength of the multinomial naive bayes model for the classification tasks. In summary the MNB as baseline achieved good classification performance. The CNN achieved the best performance and the RNN the worst performance. Regarding the hybrid model the best combination was CNN-MNB. So, we can notice that the combination of deep leaning model and naive bayes model did not really brought a benefit. It is important to consider that deep learning models need a considerable amount of data in other to get his best performance. In this work just 1315 requirements were used for the classification task, so it should be important as future work to increase the amount of data in other to investigate the performance of the deep learning models. The variation of the hyperparameter can also influence the performance of the hybrid model. This could also be a subject of a new study. In this work the computational time was less than 10 minutes, but other studies mentioned computational time longer than 10 hours, so the impact of the computational time on the model performance can also be investigated.

ISSN (online): 2581-3048
Volume 9, Issue 9, pp 1-9, September-2025
https://doi.org/10.47001/IRJIET/2025.909001

REFERENCES

[1] Ahmed H Almulihi et al. "Analyzing the Implications of Healthcare Data Breaches through Computational Technique." *In: Intelligent Automation & Soft Computing* 32.3 (2022).

[2] Oluwasefunmi T Arogundade et al. "Enhancing misuse cases with risk assessment for safety requirements". *In: IEEE Access* 8 (2020), pp. 12001– 12014.

[3] Xiang Chen et al. "Automatic classification of user requirements information based on convolutional neural network". *In: 2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT). IEEE.* 2021, pp. 137– 141.

[4] Jane Cleland-Huang et al. "Automated classification of non-functional requirements". *In: Requirements engineering* 12 (2007), pp. 103–120.

[5] Fabiano Dalpiaz et al. "Requirements classification with interpretable machine learning and dependency parsing". *In: 2019 IEEE 27th International Requirements Engineering Conference (RE). IEEE.* 2019, pp. 142–152.

[6] Edna Dias Canedo and Bruno Cordeiro Mendes. "Software requirements classification using machine learning algorithms". *In: Entropy* 22.9 (2020), p. 1057.

[7] Donald Firesmith et al. "Engineering security requirements." *In: J. Object Technol.* 2.1 (2003), pp. 53–68.

[8] Md Ariful Haque, Md Abdur Rahman, and Md Saeed Siddik. "Nonfunctional requirements classification with feature extraction and machine learning: An empirical study". *In: 2019 1st international conference on advances in science, engineering and robotics technology (ICASERT). IEEE.* 2019, pp. 1–5.

[9] Shoaib Hassan et al. "A systematic mapping to investigate the application of machine learning techniques in requirement engineering activities". *In: CAAI Transactions on Intelligence Technology* 9.6 (2024).

[10] Rajni Jindal, Ruchika Malhotra, and Abha Jain. "Automated classification of security requirements". *In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE.* 2016, pp. 2027–2033.

[11] Ji-Wook Jung, Sihn-Hye Park, and Seok-Won Lee. "A tool for security requirements recommendation using case-based problem domain ontology". *In: 2021 IEEE 29th international requirements engineering conference (RE). IEEE.* 2021, pp. 438–439.

[12] Prudence Kadebu et al. "A classification approach for software requirements towards maintainable security". *In: Scientific African* 19 (2023), e01496.

[13] Eric Knauss et al. "Supporting requirements engineers in recognising security issues". *In: Requirements Engineering: Foundation for Software Quality: 17th International Working Conference, REFSQ 2011, Essen, Germany,* March 28-30, 2011. Proceedings 17. Springer. 2011, pp. 4–18.

[14] Zijad Kurtanović and Walid Maalej. "Automatically classifying functional and non-functional requirements using supervised machine learning". *In: 2017 IEEE 25th international requirements engineering conference (RE). Ieee.* 2017, pp. 490–495.

[15] Mengmeng Lu and Peng Liang. "Automatic classification of non-functional requirements from augmented app user reviews". *In: Proceedings of the 21st international conference on evaluation and assessment in software engineering.* 2017, pp. 344–353.

[16] Nancy R Mead and Ted Stehney. "Security quality requirements engineering (SQUARE) methodology". *In: ACM SIGSOFT Software Engineering Notes* 30.4 (2005), pp. 1–7.

[17] J Manuel Pérez-Verdejo et al. "Requirements and github issues: An automated approach for quality requirements classification". *In: Programming and Computer Software* 47 (2021), pp. 704–721.

[18] Kiramat Rahman et al. "A deep learning framework for non-functional requirement classification". *In: Scientific Reports* 14.1 (2024), p. 3216.

[19] Quentin Rouland, Stojanche Gjorcheski, and Jason Jaskolka. "Eliciting a security architecture requirements baseline from standards and regulations". *In: 2023 IEEE 31st International Requirements Engineering Conference Workshops (REW). IEEE.* 2023, pp. 224–229.

[20] Summra Saleem et al. "FNReq-Net: A hybrid computational framework for functional and non-functional requirements classification". *In: Journal of King Saud University-Computer and Information Sciences* 35.8 (2023), p. 101665.

[21] Geet Sandhu, Shally Pal, and Pratap Pal. "Knowledge Extraction in Requirement Engineering with Machine Learning Perspective". *In: International Journal of Computer Applications* 975 (2015), p. 8887.

[22] John Slankas and Laurie Williams. "Automated extraction of non-functional requirements in available documentation". *In: 2013 1st International workshop on natural language analysis in software engineering (NaturaLiSE). IEEE.* 2013, pp. 9–16.

[23] John Viega. "Building security requirements with CLASP". *In: ACM SIGSOFT Software Engineering Notes* 30.4 (2005), pp. 1–7.

ISSN (online): 2581-3048
Volume 9, Issue 9, pp 1-9, September-2025
https://doi.org/10.47001/IRJIET/2025.909001

[24] Simin Wang et al. "Machine/deep learning for software engineering: A systematic literature review". *In: IEEE Transactions on Software Engineering* 49.3 (2022), pp. 1188–1231.

Citation of this Article:

Landry Giraud Wandji T. (2025). A Binary Classification of Software Security Requirements: A Deep Learning Approach. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(9), 1-9. Article DOI https://doi.org/10.47001/IRJIET/2025.909001

****

