RESEARCH ARTICLE **SPECIAL ISSUE**

RSP Science Hub

International Research Journal on Advanced Science Hub
2582-4376
Vol. 05, Issue 05S
www.rspsciencehub.com

Check for updates

http://dx.doi.org/10.47392/irjash.2023.S026

International Conference on intelligent COMPUting TEchnologies and Research (i-COMPUTER) 2023

Evaluation of Feature Engineering Techniques for Improving CVE Vulnerability Classification

Mounesh Marali[1], Dhanalakshmi R [2], Narendran Rajagopalan[1]

[1]Department of computer Science and Engineering, National Institute of Technology Puducherry, Karaikal,India.
[2]Department of Computer Science and Engineering, Indian Institute of Information Technology Tiruchirappalli, India.

Email: mounesh.marali@in.abb.com

Article History

Received: 26 February 2023
Accepted: 18 March 2023

***Keywords**:*

CVE;
Vulnerability;
Feature Engineering;
Classification

Abstract

This paper presents a three-stage approach to analyzing Common Vulnerabilities and Exposures (CVE) vulnerability datasets using machine learning techniques. In the first stage, K-Means clustering, and Linear discriminant analysis (LDA) topic modeling are applied to identify distinct clusters and topics within the dataset. The Elbow method is used to determine the optimal number of clusters for K-Means, while Grid Search is used to find the best topic model for LDA. After labeling 100 random samples from each cluster, the data is split into training and testing sets for use in various classification algorithms in the third stage. The paper contributes to the field by proposing a novel approach to analyzing CVE vulnerability datasets that combines clustering and classification techniques. The use of K-Means clustering and LDA topic modeling allows for the identification of distinct clusters and topics within the dataset, which can be used to improve the accuracy of classification algorithms. The study highlights the importance of using pre-trained word embeddings and discusses the limitations of the proposed approach. Overall, the paper provides valuable insights into the analysis of CVE vulnerability datasets and offers a framework for future research in this area.

1. Introduction

A computer security vulnerability (**Han et al.**) can result in a problem with the system's dependability and privacy. It can also be a fault, flaw, or weakness that a hostile party can take advantage of. A zero-day vulnerability is a weakness in computer security that is not widely known and is only known to specific individuals. Upon disclosure of a vulnerability, it's conceivable that software updates are not yet available. It may take a while before a vendor publishes a patch or security fix once a vulnerability is made public. To damage the target system, attackers focus on certain vulnerabilities (**Alhazmi, Woo, and Malaiya**). According to the National Vulnerability Database (NVD), the US database has over 162946 vulnerabilities. Given the abundance of vulnerabilities, the trend towards growth is obvious. The security of information systems is greatly improved by the exchange of this defect knowledge. Yet, because of the huge new growth in vulnerabilities, it is currently very difficult to conduct an accurate and efficient evaluation of the danger level of security vulnerabilities in the network (**Spanos and Angelis**). About NVD and other open-source leak libraries for network security personnel by upgrading the leak library and providing a current overview of networking threat data, experts may spot security

problems quickly (**Jang-Jaccard and Nepal**).

With the vast and growing population of vulnerabilities, it is essential to convert such a tremendous volume of data into actionable information. The development of vulnerabilities must be understood to improve system security. For this, analysis of the existing environment and emerging trends is necessary. To better understand how to avoid and lessen the effect of assaults, a range of security professionals may considerably benefit from being informed of current security vulnerability trends (Dayalan). Yet, a threat assessment that corresponds to the most recent vulnerability data is usually missing, which affects how effective security personnel are. Hence it seems sense to prioritize technicians based on vulnerability predictions. Presently, a wide range of vulnerabilities are being identified, and a diverse range of solutions are being created to address these problems. The Common Vulnerability Scoring System (CVSS) is used by NVD to analyse the threat's security level, and the severity of the vulnerability is qualitatively rated based on the score (**Ayoade et al.**).

Analysing CVE vulnerability datasets through clustering is important for several reasons. Firstly, it helps to identify different clusters or groups of vulnerabilities based on their characteristics, such as the types of attacks they enable or the affected software or hardware systems. This can provide a more nuanced understanding of the vulnerabilities and their potential impact. Secondly, clustering can be used to identify trends and patterns in vulnerability data, such as the frequency of certain types of attacks or the distribution of vulnerabilities across different systems or software. This can help security professionals to prioritize their efforts and focus on the most critical vulnerabilities. Thirdly, clustering can be used to assist in the classification of new vulnerabilities, based on their similarity to existing clusters. This can help to automate the process of vulnerability classification and reduce the workload of security analysts.

Topic modelling techniques can be used to identify patterns and themes within CVE vulnerability data, which can provide insights into different types of attacks. By applying techniques such as LDA to the vulnerability dataset, we can identify topics that are represented in the data and assign individual documents to these topics (**Neuhaus and Zimmermann**). This allows us to gain a better understanding of the common themes and patterns in the data and identify different types of attacks based on their characteristics. For example, LDA may identify a topic that is characterized by terms such as "SQL injection", "cross-site scripting", and "buffer overflow". By analysing the documents assigned to this topic, we can infer that these terms are associated with a specific type of attack. Similarly, LDA may identify another topic characterized by terms such as "malware", "virus", and "trojan". By analysing the documents assigned to this topic, we can infer that these terms are associated with a different type of attack.

The potential improvement for future research by incorporating word embedding techniques to enhance the performance of classification algorithms on CVE datasets. Word embedding techniques involve representing words as vectors in a high-dimensional space, allowing for more nuanced semantic relationships to be captured. By incorporating word embedding techniques into the analysis of CVE (Kenta et al.) **vul**nerability datasets, it may be possible to improve the accuracy and granularity of classification algorithms, ultimately leading to better identification and mitigation of security vulnerabilities. This could have significant practical implications for organizations (S..Nmez, Hankin, and Malacaria) **see**king to improve their overall cybersecurity posture. The novelty of the research work is by using clustering and topic modelling techniques, the paper offers a unique methodology for obtaining insights into the characteristics of different attacks. Additionally, the manual labelling of cluster samples to identify 22 different types of attacks represents a significant contribution to the field of cybersecurity. This labelling process provides a more nuanced understanding of the types of attacks that organizations may face, enabling them to tailor their vulnerability management strategies accordingly.

The research contributions of the paper are as follows:

- A methodology for analysing vulnerability datasets using clustering and topic modelling techniques to obtain insights into different types of attacks.
- The identification of 22 different types of attacks from a vulnerability dataset using manual

labelling of cluster samples.

• Evaluation of different classification algorithms on the labelled dataset to determine the most effective one for classifying vulnerability reports according to their attack type.

• Potential for future improvement by incorporating word embedding techniques to enhance the performance of the classification algorithms.

The paper is structured into six sections. Section 2 presents a literature review of the related work on vulnerability analysis and clustering techniques. In section 3, the background of the CVE vulnerability datasets is discussed. Section 4 describes the implementation, which includes using K-Means clustering and LDA topic modelling to analyse the dataset. Section 5 presents the results and discussion of the classification algorithms used in the work. Finally, in section 6, the conclusion and future scope of the study are discussed, including the potential for improvement by incorporating word embedding techniques to enhance the performance of the classification algorithms on CVE datasets.

2. Literature Review

The research evaluates the quality of datasets, classification models, vectorization techniques, and function/variable name replacement to compare the performance of traditional machine learning-based vulnerability detection methods with deep learning-based detection techniques. In order to shed light on the experimental findings, the authors compile three vulnerability code datasets from NVD and Software Assurance Reference Dataset (SARD) and extract features of vulnerability code datasets. The research shows that deep learning models, especially BLSTM, can outperform classical ML algorithms, and that CountVectorizer can greatly enhance the performance of classical ML algorithms. The paper concludes that the random forest algorithm generates features such as system-related functions, syntax keywords, and user-defined names, and that these features vary depending on the vulnerability type and the source of the code. The study concludes that vulnerability detection performance can suffer in datasets that use user-defined variable and function name replacement, and that it improves with a higher percentage of code from SARD (**Zheng et al.**).

The author suggested text mining methods based on the text description of CVE from the NVD to extract the key characteristics, use principal component analysis to gather sparsity characteristics, and XGBoost to intelligently predict the severity of security flaws. The author then compared the results with those of other ML methods based on other features extracted (**Wang et al.**). In another research for vulnerability text categorization evaluation, the author proposed a variety of deep-learning strategies to decrease the workload of specialists and the false negative rate of the conventional method. The recommended method draws attention to the widespread Cross-Site Scripting (XSS) vulnerability. Three separate deep neural network types (CNN, LSTM, TextRCNN) and one type of traditional machine learning technique are used to assess and categorize textual input (**Liu et al.**).

This study presents a dynamic vulnerability threat assessment approach, based on publicly available data, to predict the likelihood to be exploited for each vulnerability in order to prevent future cyberattacks (i.e., CVE). Variables related to vulnerability from various sources are taken into account by the model. Some of the parts include a brief introduction to the contributors and some context on Twitter discussions of these flaws. Prediction accuracy was found to improve when the recommended method was used to foresee the use of vulnerabilities in real-world data (**Huang and Wu**).

The author suggests a method for computing CVSS ratings that is objective even without subjective experience. Although Support Vector Machine (SVM) and Random-Forest are the most widely used and trustworthy prediction techniques, this study's findings indicate that using fuzzy systems may result in even better results (**Khazaei, Ghasemzadeh, and Derhami**). For the combined forecasting of several security vulnerability entity instances on the security vulnerability characterizations, the author proposes a multi-task ML technique. Due to the use of neural network models that can learn to extract features from training data, the method is presented in the publication and does not need balanced data (**Gong et al.**). The author demonstrates how to use a large variety of different vulnerability repair techniques by developing several vulnerability repair methods and compares how each one functions in terms of balancing cover and efficacy (**Jacobs et al.**).

Another research provides a dataset of 1813 CVEs that have been associated with all relevant MITRE ATT&CK methods and suggests models for automatically connecting a CVE to one or more techniques based on the text description provided by the CVE information. We provide a strong baseline that accounts for both conventional machine learning models and cutting-edge pre-trained BERT-based language models, and we use data augmentation strategies based on the TextAttack framework to combat the very uneven training set. With the top model attaining an F1-score of 47.84%, our findings are positive. Also, we perform qualitative research to emphasise the limitations and apparent differences in CVE definitions using Qualitative explanations (**Grigorescu et al.**).

For vulnerability analysis, researchers present VE-Extractor, a technique for automatically extracting vulnerability event triggers and event parameters from textual descriptions in vulnerability reports. This approach was previously suggested in another research. The authors developed a new labelling system called BIOFR to provide a baseline for vulnerability data from an event viewpoint. Lastly, we use the BERT Q&A paradigm to formulate a question template dependent on the event trigger to mechanically extract the characteristics of the vulnerability event (**Wei et al.**).

In a separate study, the authors CWE and classification might be confused with common vulnerabilities and exposures, even though they are not necessarily related. Here, researchers offer a methodology for discovering antidictionary connections. Several patterns, including term frequency-inverse document frequency, universal sentence encoder, and sentence similarity, are evaluated empirically using the proposed method. BERT (**Kanakogi et al.**).

Using the hierarchical structure of CWE-IDs and knowledge distillation, the authors of another research offered a unique approach to the extremely imbalanced software vulnerability classification (SVC) issue. To organise CWE-IDs more easily with similar characteristics, researchers break down the overall distribution into simpler sub-distributions using CWE abstract categories (categorizations that group comparable CWE-IDs). TextCNN instructors are trained on all of the abridged distributions but excel solely within their own specialty. In order to generalise the effectiveness of TextCNN teachers, researchers use a hierarchical knowledge distillation strategy and create a transformer student model. Many long-tailed learning and source code transformer models have been presented for the field of vision (**Fu et al.**).

3. CVE Dataset

The CVE dataset is a public repository of cybersecurity vulnerabilities that have been identified and reported. The CVE database was created in 1999 to provide a standardized method for tracking vulnerabilities in software systems. The dataset is maintained by the MITRE Corporation and is freely available for anyone to use. Each vulnerability in the CVE dataset is assigned a unique identifier and includes information about the software system affected, the severity of the vulnerability, and any available patches or workarounds. The CVE dataset is used by security researchers, software vendors, and government agencies to identify and mitigate vulnerabilities in software systems. The CVE dataset is constantly evolving, with new vulnerabilities being added on a regular basis. As of 2021, the CVE dataset contains over 150,000 entries, covering a wide range of software systems and vulnerabilities. The dataset includes vulnerabilities in operating systems, web applications, network infrastructure, and other software systems. CVE vulnerabilities are typically discovered by security researchers, who report them to software vendors or other responsible parties. Once a vulnerability has been confirmed, it is added to the CVE database, along with any available information about the vulnerability.

The CVE dataset is an important resource for software security professionals, as it allows them to stay up to date on the latest vulnerabilities and take steps to mitigate them. However, the size and complexity of the dataset can make it difficult to analyse and extract meaningful insights. Researchers have used a variety of techniques, including machine learning and natural language processing, to analyse the CVE dataset and identify patterns and trends in vulnerability data.

4. Proposed workflow

The proposed workflow consists of three stages which is represented in the Figure 1.

Stage 1 - Analyzing Data:

The first contribution of this project is the use

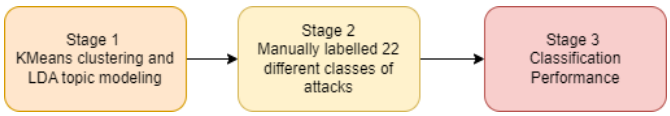

FIGURE 1. Proposed workflow

of K-Means clustering and LDA topic modeling techniques to analyze a large dataset of vulnerability documents. The Elbow method was used to determine the optimal number of clusters, and Grid Search was used to find the best number of topics for the LDA model.

Stage 2 - Labeling:

The second contribution of this project is the manual labeling of 100 random samples from each cluster to identify 22 different classes of attacks. This labeled dataset was then used for training and testing various classification algorithms in the next stage.

Stage 3 - Classification:

The third contribution of this project is the evaluation of different classification algorithms on the labeled dataset, including Random Forest, Support Vector Machines, and Logistic Regression. The performance of these algorithms was measured using precision, recall, and F1- score metrics.

4.1. Stage 1

The goal was to extract meaningful information from the large dataset and identify distinct clusters and topics within the data.

The following steps were performed in Stage 1:

• Load the vulnerability dataset and obtain the text descriptions of vulnerabilities.

• Perform clustering on the data using the K-Means algorithm. Using the Elbow technique, we found the value of k at which distortion begins to decrease linearly, and that is the number of clusters we utilized to get our final output.

• Find topics within the data using Latent Dirichlet Allocation (LDA) algorithm. Grid Search was used to find the best number of topics and learning decay rate.

• Export the clustered and topics datasets and manually label random documents to use in the next stage of classification.

In stage 1, the vulnerability dataset containing 172,287 documents was analyzed. K-Means clustering was performed to segment the dataset into different clusters. The Elbow method which is shown in figure 2 was used to determine the optimal number of clusters, and it was found that the best value for k was 14. The LDA algorithm was then used to identify topics within the dataset, and Grid Search was used to find the best number of topics, which was found to be 25 with a learning decay of 0.5. The entire process of finding the optimal k using K-Means and the optimal number of topics using LDA which is shown in Figure 3. The resulting clustered and topic datasets were exported and 100 random samples from each cluster were manually labelled for use in the following stages.

4.2. Stage 2

In stage 2, we manually labelled 100 random samples from each cluster to try different classification algorithms in the following stage We found 22 different classes (types of attacks) from the vulnerability documents after labeling the cluster samples. Figure 4 and 5 show the cluster samples of CVE label and label ID in the labelled dataset. Although LDA, the clustered data produced by K-Means clusters seemed more segmented, and after manual labeling and gained more confidence in the dataset with 22 target classes. Figure 6 shows the distribution of 22 classes in the labelled dataset. This labeled dataset was then used as the training and test data for the classification algorithms in Stage 3.

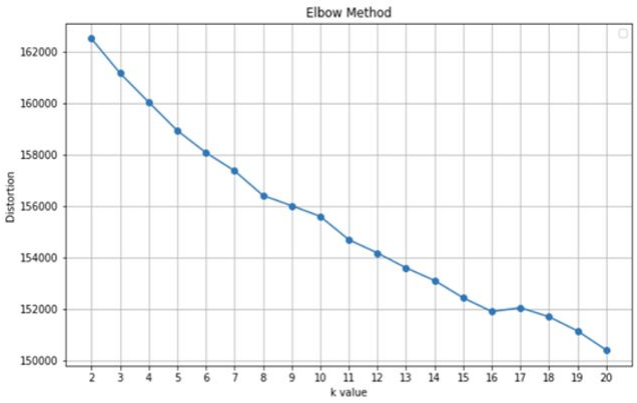

FIGURE 2. Elbow method

4.3. Stage 3

In Stage 3, the labeled dataset, from Stage 2 is loaded, preprocessed, and split into training and testing sets. Various classification algorithms with different parameters are then used to determine which one performs the best. The vulnerability dataset includes 22 distinct assaults, and the purpose is to

```
Top 10 words for topic #0: ['adobe', '2019', '2020', 'vulnerability', 'different', '2017', 'sd', '2015', '2016', 'cve']
Top 10 words for topic #1: ['sp1', 'sp2', 'java', 'snapdragon', 'aka', 'microsoft', 'server', 'android', 'vulnerability', 'windows']
Top 10 words for topic #2: ['xss', 'cross', 'site', 'vulnerability', 'parameter', 'allows', 'attackers', 'arbitrary', 'remote', 'php']
Top 10 words for topic #3: ['vulnerability', 'did', 'cna', 'consultids', 'reason', 'reject', 'notes', 'use', 'number', 'candidate']
Top 10 words for topic #4: ['upload', 'directory', 'version', 'execution', 'arbitrary', 'code', 'versions', 'vulnerability', 'file', 'earlier']
Top 10 words for topic #5: ['local', 'ibm', 'vulnerability', 'information', 'remote', 'attacker', 'access', 'users', 'allows', 'user']
Top 10 words for topic #6: ['component', 'allows', '12', 'cisco', 'affected', 'cvss', 'access', 'attacker', 'oracle', 'vulnerability']
Top 10 words for topic #7: ['15', 'devices', 'firmware', 'vulnerability', 'discovered', 'version', 'issue', 'attacker', 'prior', 'versions']
Top 10 words for topic #8: ['information', 'function', 'does', 'crafted', 'remote', 'cause', 'denial', 'attackers', 'service', 'allows']
Top 10 words for topic #9: ['buffer', 'denial', 'cause', 'service', 'execute', 'code', 'arbitrary', 'remote', 'allows', 'attackers']
```

FIGURE 3. LDA approach for the optimal number of topics

	_id	description	Label
0	CVE-2006-0344	Directory traversal vulnerability in Intervati...	Directory Traversal
1	CVE-2018-0646	Directory traversal vulnerability in Explzh v....	Directory Traversal
2	CVE-2012-4959	Directory traversal vulnerability in NFRAgent....	Directory Traversal
3	CVE-2005-0701	Directory traversal vulnerability in Oracle Da...	Directory Traversal
4	CVE-2004-1548	Directory traversal vulnerability in the file ...	Directory Traversal

FIGURE 4. Cluster samples CVE dataset labelled

categorize them. Precision, recall, F1 score, and accuracy are used to compare the effectiveness of these algorithms. The best method is chosen after thorough testing, and the complete dataset is used to train the final model. After completing the model, its efficacy is verified by gauging its performance against test data.

5. Results and Discussion

The results of the classification stage show that several classification algorithms were evaluated on the labelled dataset, which was obtained after manual labelling of random samples from each cluster which is shown in Table 1. The performance of each algorithm was evaluated using accuracy, which is the proportion of correctly classified instances out of the total instances. The Linear SVM algorithm performed the best with an accuracy of 0.8562, followed by Random Forest with an accuracy of 0.8516 and the Decision Tree with an accuracy of 0.8187. The worst-performing algorithm was Bernoulli Naive Bayes with an accuracy of only 0.4781. The Multinomial Naive Bayes algorithm also performed relatively poorly with an accuracy of 0.6937, which could be because it assumes independence among features, which may not hold true for the dataset. The Gaussian Naive Bayes algorithm performed worse than the Multinomial Naive Bayes with an accuracy of only 0.5500. The KNN algorithm had an accuracy of 0.7828, which is lower than the top-performing algorithms but still performs reasonably well. The Multilayer Perceptron algorithm had an accuracy of 0.8265, which is higher than KNN but lower than the top-performing algorithms. Overall, the results suggest that the Linear SVM algorithm is the most suitable for classifying the vulnerability dataset. The Random Forest and Decision Tree algorithms also performed well and could be alternative options. However, the performance of each algorithm may depend on the specific characteristics of the dataset and the problem being solved. Therefore, further experimentation and evaluation may be required to determine the best algorithm for the task.

Figure 7 shows the confusion matrix of the predictive model. As per the results in Figure 8, some classes have high precision, recall, and f1-score, indicating good classification accuracy, while others have poor performance. For example, classes 2, 3, 5, 11, and 21 have high precision, recall, and f1-score, which suggests that the model is able to accurately classify documents in these classes. On the other hand, classes 12, 15, 18, 20, and 19 have poor performance, indicating that the model struggles to accurately classify documents in these classes.

Figure 9 shows the comparison of the training and validation accuracy achieved by the model using the two types of word embeddings. The learned word embeddings resulted in high training accuracy, which indicates that the model was able to learn and memorize the patterns in the training data. However, the validation accuracy was low, which suggests that the model was not able to generalize well to new data. This is a common problem with overfitting, where the model performs well on the training

	_id	description	Label	Label_id
95	CVE-2016-1212	Directory traversal vulnerability in futomi MP...	Directory Traversal	0
96	CVE-2001-0462	Directory traversal vulnerability in Perl web ...	Directory Traversal	0
97	CVE-2014-2279	Multiple directory traversal vulnerabilities i...	Directory Traversal	0
98	CVE-2018-18831	An issue was discovered in com\mingsoft\cms\ac...	Directory Traversal	0
99	CVE-2010-1471	Directory traversal vulnerability in the Addre...	Directory Traversal	0
100	CVE-2012-2644	Cross-site scripting (XSS) vulnerability in th...	Cross-site scripting	1
101	CVE-2009-3668	Cross-site scripting (XSS) vulnerability in ar...	Cross-site scripting	1
102	CVE-2005-2724	Cross-site scripting (XSS) vulnerability in Sq...	Cross-site scripting	1
103	CVE-2021-3151	i-doit before 1.16.0 is affected by Stored Cro...	Cross-site scripting	1
104	CVE-2008-2518	Cross-site scripting (XSS) vulnerability in th...	Cross-site scripting	1

FIGURE 5. Cluster samples CVE dataset labelled with Label ID

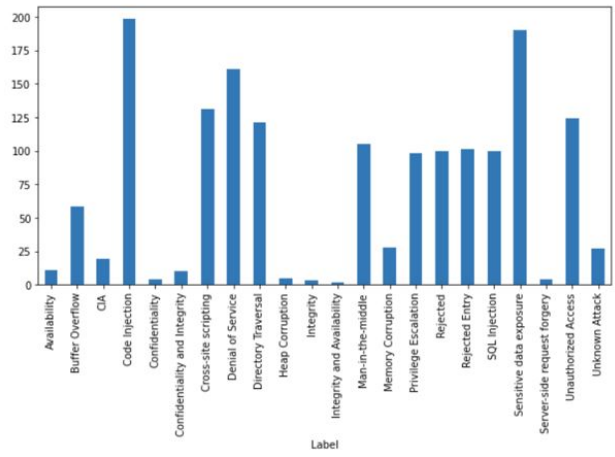

FIGURE 6. Distribution of 22 classes in the labelled dataset

```
Confusion Matrix:
[[52  0  0  0  0  0  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 40  0  0  0  0  1  2  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 36  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 34  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 68  0  3  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 51  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  3  0 48  1  0  0  0  1  0  1  2  0  0  0  0  0  0  0]
 [ 1  0  0  0  5  0 18 27  4  0  0  6  0  0  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  2  0  1  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  1  0  0  0  1 21  0  0  0  1  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  1  0  1  0  0  0  5  1  0  0  0  0  0  1  0  1  0  0]
 [ 0  3  0  0  0  0  0  5  0  2  0 77  0  0  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 11  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  2  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  3  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40]]
```

FIGURE 7. Confusion Matrix

data but fails to perform well on new, unseen data. On the other hand, the pre-trained word embeddings resulted in better validation accuracy, which indicates that the model was able to generalize better to new data. However, the pre-trained word embeddings did not perform better than state-of-the-art approaches in terms of accuracy. This suggests that while the pre-trained word embeddings can help improve the generalization ability of the model, they are not enough to achieve state-of-the-art performance on their own.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.95      0.96        55
           1       0.89      0.89      0.89        45
           2       1.00      1.00      1.00        36
           3       1.00      1.00      1.00        34
           4       0.85      0.94      0.89        72
           5       1.00      0.96      0.98        53
           6       0.65      0.83      0.73        58
           7       0.71      0.44      0.54        62
           8       0.82      0.90      0.86        31
           9       0.91      0.88      0.89        24
          10       0.83      0.50      0.62        10
          11       0.87      0.88      0.87        88
          12       1.00      0.50      0.67         2
          13       0.33      1.00      0.50         1
          14       0.73      0.92      0.81        12
          15       0.00      0.00      0.00         2
          16       0.67      0.67      0.67         6
          17       0.60      0.75      0.67         4
          18       0.00      0.00      0.00         2
          19       0.14      0.50      0.22         2
          20       0.00      0.00      0.00         1
          21       1.00      1.00      1.00        40

    accuracy                           0.86       640
   macro avg       0.68      0.70      0.67       640
weighted avg       0.86      0.86      0.85       640
```

FIGURE 8. Comparison of precision, recall and f1 score

TABLE 1. Classification Performance

Approac	SVM	Linear SVM	KNN	Randon Forest	Multilaye Percep-tron	Multinomia Naive Bayes	Gaussian Naive Bayes	Bernoulli Naive Bayes	Decision Tree
Accuracy	0.8047	0.8562	0.7828	0.8516	0.8265	0.6937	0.5500	0.4781	0.8187

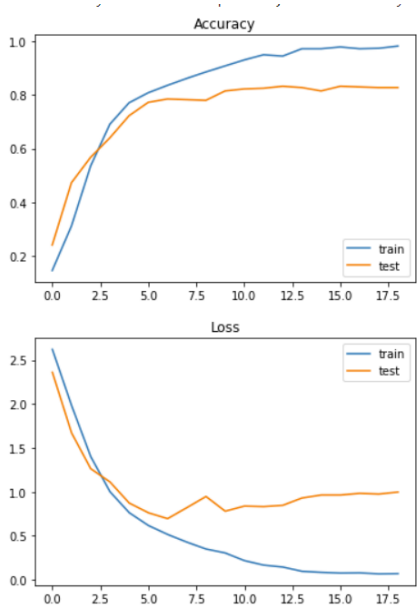

FIGURE 9. Comparison of the training and validation accuracy

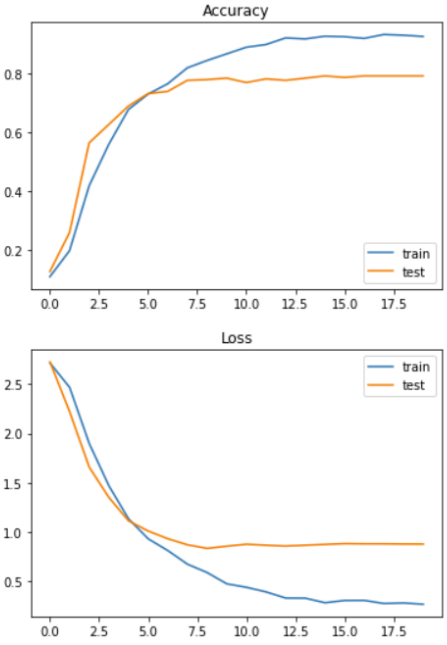

FIGURE 10. Comparison of the training and validation accuracy

6. Conclusion and Future Scope

This research work presented a comprehensive approach to vulnerability detection and classification in software security. The approach involved three stages: data preprocessing and topic modeling, feature extraction and selection, and classification using various machine learning algorithms and word embedding techniques. The results showed that linear SVM, Random Forest, and Decision Tree classifiers performed well for the 22-class classification problem, with accuracy scores of 0.8562, 0.8516, and 0.8187, respectively. On the other hand, Naive Bayes classifiers showed poor performance, with accuracy scores ranging from 0.4781 to 0.6937. The future scope for this research work includes exploring deep learning algorithms for vulnerability classification. Additionally, the approach can be extended to include more features, such as code changes, bug reports, and user feedback, to improve the accuracy of vulnerability detection and classification. Furthermore, the approach can be evaluated on different datasets to test its generalizability.

Authors' Note

The authors declare that there is no conflict of interest regarding the publication of this article.

References

Ayoade, Gbadebo, et al. "Automated Threat Report Classification over Multi-Source Data". *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)* (2018): 236–245. 10.1109/CIC.2018.00040.

Gong, Xi, et al. "Joint Prediction of Multiple Vulnerability Characteristics Through Multi-Task Learning". *2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS)* (2019): 31–40. 10 . 1109 / ICECCS . 2019.00011.

Grigorescu, Octavian, et al. "CVE2ATT&CK: BERT-Based Mapping of CVEs to MITRE ATT&CK Techniques". *Algorithms* 15.9 (2022): 314–314. 10.3390/a15090314.

Han, Zhuobing, et al. "Learning to Predict Severity of Software Vulnerability Using Only Vulnerability Description". *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (2017): 125–136. 10.1109/ICSME.2017.52.

Huang, Shin-Ying and Yiju Wu. "POSTER: Dynamic Software Vulnerabilities Threat Prediction through Social Media Contextual Analysis". *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*. ACM, 2020. 892–894.

Jacobs, Jay, et al. "Improving vulnerability remediation through better exploit prediction". *Journal of Cybersecurity* 6.1 (2020). 10.1093/cybsec/tyaa015.

Jang-Jaccard, Julian and Surya Nepal. "A survey of emerging threats in cybersecurity". *Journal of Computer and System Sciences* 80.5 (2014): 973–993. 10.1016/J.JCSS.2014.02.005.

Kanakogi, Kenta, et al. "Comparative Evaluation of NLP-Based Approaches for Linking CAPEC Attack Patterns from CVE Vulnerability Information". *Applied Sciences* 12.7 (2022): 3400–3400. 10.3390/app12073400.

Kenta, Kanakogi, et al. "Tracing CVE Vulnerability Information to CAPEC Attack Patterns Using Natural Language Processing Techniques". *Information* 12.8 (2021): 298–298. 10.3390/info12080298.

Khazaei, Atefeh, Mohammad Ghasemzadeh, and Vali Derhami. "An automatic method for CVSS score prediction using vulnerabilities description". *Journal of Intelligent & Fuzzy Systems* 30.1 (2015): 89–96. 10.3233/IFS-151733.

Liu, Kai, et al. "Vulnerability Severity Prediction With Deep Neural Network". *2019 5th International Conference on Big Data and Information Analytics (BigDIA)* (2019): 114–119. 10.1109/BigDIA.2019.8802851.

Neuhaus, Stephan and Thomas Zimmermann. "Security Trend Analysis with CVE Topic Models". *2010 IEEE 21st International Symposium on Software Reliability Engineering* (2010): 111–120. 10.1109/ISSRE.2010.53.

S..Nmez, Ferda ..Zdemir, Chris Hankin, and Pasquale Malacaria. "Attack Dynamics: An Automatic Attack Graph Generation Framework Based on System Topology, CAPEC, CWE, and CVE Databases". *Computers & Security* 123 (2022): 102938–102938. 10.1016/j.cose.2022.102938.

Spanos, Georgios and Lefteris Angelis. "A multi-target approach to estimate software vulnerability characteristics and severity scores". *Journal of Systems and Software* 146 (2018): 152–166. 10.1016/j.jss.2018.09.039.

Wang, Peichao, et al. "Intelligent Prediction of Vulnerability Severity Level Based on Text Mining and XGBboost". *2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)* (2019): 72–77. 10.1109/ICACI.2019.8778469.

Wei, Ying, et al. "Automated event extraction of CVE descriptions". *Information and Software Technology* 158 (2023): 107178–107178. 10.1016/J.INFSOF.2023.107178.

Zheng, Wei, et al. "The impact factors on the performance of machine learning-based vulnerability detection: A comparative study". *Journal of Systems and Software* 168 (2020): 110659–110659. 10.1016/j.jss.2020.110659.

Embargo period: The article has no embargo period.

To cite this Article: Marali, Mounesh, Dhanalakshmi R , and Narendran Rajagopalan. "**Evaluation of Feature Engineering Techniques for Improving CVE Vulnerability Classification**." International Research Journal on Advanced Science Hub 05.05S (2023): 196–205. http://dx.doi.org/10.47392/irjash.2023.S026