Classification of Vulnerability using NLP

3 Pager Report

Dibyendu Mandal
5th year Dual Degree,Electrical Engineering
IIT Madras
Chennai, India
ee18b108@smail.iitm.ac.in

***Abstract*—The Common Vulnerabilities and Exposures (CVE) system plays a crucial role in identifying and categorizing security vulnerabilities in software and hardware products. However, the manual classification of CVE entries is a time-consuming and resource- intensive task. In recent years, Natural Language Processing (NLP) techniques have emerged as a promising approach for automating CVE classification.**
This research focuses on developing an NLP-based approach for CVE classification. The proposed methodology utilizes state-of-the-art NLP techniques, including text preprocessing, feature extraction, and machine learning algorithms. The CVE textual descriptions and associated metadata are processed and transformed into a suitable format for classification.

***Index Terms*—Common Vulnerabilities and Exposures (CVE), classification, Natural Language Processing (NLP), machine learning, text preprocessing**

I. Introduction

A vulnerability is a weakness or error in a system or device's code that, when exploited, can compromise the confidentiality, availability, and integrity of data stored in them through unauthorized access, elevation of privileges, or denial of service.

II. Problem Description

A. Problem Statement

Implement Natural Language Processing (NLP) algorithms on the CVE Description to identify and tag one or more Vulnerability Type(s) that can be assigned to the CVE.

B. Objectives:

The primary objective of this thesis is to develop a CVE classification system using NLP techniques that can effectively categorize vulnerabilities based on their severity, impact, and potential exploitability.
The specific goals include:

- Gathering and preprocessing CVE data: Collecting a comprehensive dataset of CVE entries and performing data preprocessing tasks such as cleaning, normalization, and feature extraction to prepare the data for analysis.
- Exploring NLP techniques: Investigating various NLP techniques, including text representation models (e.g., word embeddings, language models), feature engineering methods, and linguistic analysis tools, to extract meaningful information from CVE texts.
- Machine learning models: Designing and implementing machine learning models (e.g., classification algorithms, deep learning architectures) to train and evaluate the CVE classification system. This involves selecting appropriate training algorithms, optimizing model parameters, and evaluating the performance of the models.
- Evaluation and comparison: Assessing the effectiveness of the proposed CVE classification system by comparing its performance with existing manual or rule-based approaches. Conducting rigorous evaluation metrics, such as precision, recall, and F1 score, to measure the system's accuracy and reliability.

C. Significance and Contributions

This thesis aims to make several contributions to the field of cybersecurity and vulnerability management:

- Automation and efficiency: Developing an automated CVE classification system will significantly reduce the time and effort required for vulnerability triage and remediation, enabling security professionals to focus on critical issues and respond more effectively to potential threats.
- Accuracy and consistency: By leveraging NLP techniques, the proposed system seeks to improve the accuracy and consistency of CVE classification, reducing the potential for human errors and biases that may arise from manual analysis.
- Knowledge discovery: Through linguistic analysis and machine learning, this research intends to uncover valuable insights and patterns within CVE texts, providing a deeper understanding of vulnerabilities and their characteristics.
- Practical implications: The findings of this thesis can be applied to real-world vulnerability management practices, benefiting organizations and security teams

III. Literature Review

A. Natural Language Processing (NLP) Techniques

NLP techniques play a vital role in extracting meaningful information from textual descriptions and metadata associated with CVEs. The review will explore various NLP techniques used in CVE classification, such as text preprocessing, feature extraction, and semantic analysis.
Text preprocessing techniques, including tokenization, stopword removal, and stemming, are employed to clean and normalize the textual data.
Feature extraction methods, such as bag-of-words, word embeddings (e.g., Word2Vec, GloVe), and named entity recognition, are used to represent CVE descriptions in a format suitable for machine learning algorithms.
Additionally, semantic analysis techniques, such as topic modeling and sentiment analysis, can provide deeper insights into the CVE descriptions and aid in classification.

B. CVE Classification Approaches

The literature review will cover different approaches for CVE classification using NLP techniques.
Supervised learning algorithms, including Support Vector Machines (SVM), Naive Bayes, and Random Forests, have been widely used in CVE classification. These algorithms require labeled training data, where CVEs are manually annotated with their respective classes.
Unsupervised learning algorithms, such as clustering and dimensionality reduction techniques, offer alternative methods for CVE classification, particularly in scenarios where labeled data is scarce.
Deep learning approaches, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer models, have shown promising results in various NLP tasks and can be applied to CVE classification as well.

IV. Methodology

A. Dataset Preparation

Dataset is being generated with the knowledge of web scraping from the website www.cvedetails.com which stores all information on CVEs. Three features are extracted namely CVE IDs,CVE descrtiptions and Vulnerability type.THe total dataset comprises 1.8Lacs of such data.This form of dataset is raw and can be used directly after preprocessing to train models and analyze information

B. Preprocessing

- Tokenization: Split the text into individual words or tokens.
- Stemming/Lemmatization: Reduce words to their base or root form.
- Stop Word Removal: Eliminate common, non-informative words.
- Special Character Handling: Remove punctuations, symbols, and special characters.
- Numeric Value Handling: Replace numerical values with generic tokens.
- Lowercasing: Convert all words to lowercase for consistency.
- Spell Checking: Correct misspelled words to improve accuracy.
- Removal of HTML Tags or URLs: Eliminate HTML tags or URLs if present in the text.
- Removal of Irrelevant Text: Remove irrelevant sections like headers, footers, or boilerplate text.
- Text Normalization: Perform additional normalization techniques like removing excessive white spaces or standardizing abbreviations.

C. Feature Engineering

- Special Character Handling: Remove punctuations, symbols, and special characters.
- Numeric Value Handling: Replace numerical values with generic tokens.
- TF-IDF: Assign weights to words based on their frequency and inverse document frequency.
- Word Embeddings: Represent words as dense vectors capturing semantic relationships.
- Contextualized Word Embeddings: Capture contextual information by considering surrounding words.
- Feature Selection: Select relevant features based on importance or statistical measures.
- Normalization: Scale feature values to a standard range for better model performance.

D. Classification Algorithms

Decision Trees: A tree-based model that splits data based on features to create hierarchical decision rules.
Random Forest: An ensemble of decision trees that combines their predictions for improved accuracy and robustness.
LSTM with Finetuning: Use of bigrams over monograms provide better results and expected to provide better results here.

E. Data Visualisation

Important Keywords that are much relevant particular to a vulnerability occurs frequently under the same vulnerability.This analysis can help selecting features for each vulnerability type to classify.
Use of Wordcloud to identify variables under the same vulnerability type
Use of TF-IDF over normal vectorizers outperforms model outcomes in many cases.

F. Experimental Setup

- Split the dataset into training, validation, and testing sets using stratified sampling and applying cross-validation,
- Perform cross-validation to validate the robustness of the models.
- Train each model on the training set and evaluate them on the validation set.

- Select the best-performing model based on the evaluation metrics.
- Test the whole dataset using the model and generate outputs and aggregate them to multi-map each cve description to multi-vulnerabilty types
- Find out the working performance of the aggregated model.
- Use of LSTM model for vulnerability classification as the model uses sequential data.There are lot many cases in cyber-security terminologies that is affected by the ordering of the terms.

V. OBSERVATIONS AND RESULTS

In this section we will go through the following:
The dataset and preprocessing of data
We will see the results received from using Supervised models and it's performance.
We will also go through bigrams extracted in the form of Wordcloud.

A. Preprocessing

Our data contains three Variables namely CVE ID, summary text(CVE Description) and Vulnerability Type.
We do preprocessing on the data and we can see that processed text contains some logical and extracted text from the description.Since the data is large it takes some time to train.It is shown in figure below:

Index	CVE ID Number	Vulnerability Type	Summary Text	Processed Text
173959	CVE-2010-4845	Security Vulnerabilities	Multiple SQL injection vulnerabilities in MH Products Projekt Shop allow remote attackers to execute arbitrary SQL commands via the (1) to parameter to details.php and possibly the (2) lceler parameter to index.php.	multiple sql injection vulnerabilities mh products projekt shop allow remote attackers execute arbitrary sql commands via to parameter detailsphp possibly lceler parameter indexphp
156851	CVE-2014-5685	Gain Information	The Runtastic Heart Rate (aka com runtastic android.heartrate.lite) application 1.3 for Android does not verify X.509 certificates from SSL servers, which allows man-in-the-middle attackers to spoof servers and obtain sensitive information via a crafted certificate.	runtastic heart rate aka comruntasticandroidheartratelite application android verify x certificates ssl servers allows maninthemiddle attackers spoof servers obtain sensitive information via crafted certificate
40982	CVE-2018-17644	Code Execute	This vulnerability allows remote attackers to execute arbitrary code on vulnerable installations of Foxit Reader 9.2.0.9297. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file. The specific flaw exists within the handling of the addItem method of a TimeField. The issue results from the lack of validating the existence of an object prior to performing operations on the object. An attacker can leverage this vulnerability to execute code in the context of the current process. Was ZDI-CAN-6481	vulnerability allows remote attackers execute arbitrary code vulnerable installations foxit reader user interaction required exploit vulnerability target must visit malicious page open malicious file specific flaw exists within handling addItem method timefield issue results lack validating existence object prior performing operations object attacker leverage vulnerability execute code context current process zdican
6544	CVE-2018-6187	Denial of Service	In Artifex MuPDF 1.12.0, there is a heap-based buffer overflow vulnerability in the do_pdf_save_document function in the pdf\pdf-write.c file. Remote attackers could leverage the vulnerability to cause a denial of service via a crafted pdf file.	artifex mupdf heapbased buffer overflow vulnerability dopdfsavedocument function pdfpdfwritec file remote attackers could leverage vulnerability cause denial service via crafted pdf file
863	CVE-2022-20794	Denial of Service	Multiple vulnerabilities in the web engine of Cisco TelePresence Collaboration Endpoint (CE) Software and Cisco RoomOS Software could allow a remote attacker to cause a denial of service (DoS) condition. view sensitive data on an affected device, or redirect users to an attacker-controlled destination. For more information about these vulnerabilities, see the Details section of this advisory.	multiple vulnerabilities web engine cisco telepresence collaboration endpoint ce software cisco roomos software could allow remote attacker cause denial service dos condition view sensitive data affected device redirect users attackercontrolled destination information vulnerabilities see details section advisory

Fig. 1. Pre-processed Data

B. Supervised Learning Models

Random Forest produced the best score for a test result on a sample of the data.

Performance of Supervised Models

	Accuracy Scores
Logistic regression TF-IDF	0.8395
Gaussian NB TF-IDF	0.8666
Decision Tree TF-IDF	0.8395
Random Forest TF-IDF	0.9633
K Nearest Neighbour TF-IDF	0.8395

Stratified Sampling and testing on Random Forest gave a decent accuracy of 71 percent.

As you can see the results are promising and using this supervised model we proceed to produce multi-labelled outcomes for every description as a description can have many possible vulnerabilities associated with it.

	CVE ID Number	Vulnerability Type	Summary Text	Processed Text	TF-IDF Predicted
80083	CVE-2017-8748	Overflow	Internet Explorer in Microsoft Windows 7 SP1, ...	internet explorer microsoft windows sp window...	Overflow
9343	CVE-2017-2531	Denial of Service	An issue was discovered in certain Apple produ...	issue discovered certain apple products ios a...	Denial of Service
39271	CVE-2019-7797	Code Execute	Adobe Acrobat and Reader versions 2019.010.201...	adobe acrobat reader versions earlier earlie...	Code Execute
71082	CVE-1999-0935	Code Execute	classifieds.cgi allows remote attackers to exe...	classifiedscgi allows remote attackers execute...	Code Execute
54799	CVE-2011-3411	Code Execute	Microsoft Publisher 2003 SP3 allows remote att...	microsoft publisher sp allows remote attacker...	Code Execute
85845	CVE-2013-0030	Overflow	The Vector Markup Language (VML) implementatio...	vector markup language vml implementation micr...	Overflow
129780	CVE-2008-3574	Cross Site Scripting (XSS)	Multiple cross-site scripting (XSS) vulnerabil...	multiple crosssite scripting xss vulnerabiliti...	Cross Site Scripting (XSS)
172766	CVE-2013-6767	Security Vulnerabilities	Stack-based buffer overflow in pepolyll in Q...	stackbased buffer overflow pepolyll quick hea...	Code Execute
141683	CVE-2020-0092	Bypass	In setHideSensitive of NotificationStackScroll...	sethidesensitive notificationstackscrolllayout...	Bypass
44366	CVE-2017-11274	Code Execute	Adobe Digital Editions 4.5.4 and earlier has a...	adobe digital editions earlier exploitable us...	Code Execute

Fig. 2. Random Forest Predicted results

Using this random forest as our model we predict if the description belongs to a particular vulnerability type or not, and do this for all the vulnerability types. Then all whichever it belongs to are aggregated and that is how we multi-label it!

This figure below shows working results:

1 to 5 of 5 entries Filter

Index	CVE ID Number	Vulnerability Type	Processed Text	DoS	XSS	CSRF	Mem C	Dir T	Code Exec	Bypass	HTTP	File Inc	OverF	SQLI	Gain Priv	Sec Vul	Gain Inf	all	null
95482	CVE-2018-4998	Memory Corruption	adobe acrobat reader versions earlier earlier earlier memory corruption vulnerability successful exploitation could lead arbitrary code execution context current user	0	0	0	1	0	1	0	0	0	1	0	0	0	0	Mem C,Code Exec,OverF	Memory Corruption,Code Execute,Overflow
127253	CVE-2012-1561	Cross Site Scripting (XSS)	crosssite scripting xss vulnerability finder module xx x xx xx xalpha drupal allows remote attackers inject arbitrary web script html via unspecified vectors related checkbox radio button functionalities	0	1	0	0	0	0	0	0	0	0	0	0	0	0	XSS	Cross Site Scripting (XSS)
97705	CVE-2015-3674	Memory Corruption	allpservser apple os x allows remote attackers execute arbitrary code cause denial service memory corruption via unspecified vectors	1	0	0	1	0	0	0	0	0	0	0	0	0	0	DoS,Mem C	Denial of Service,Memory Corruption
125856	CVE-2013-6991	Cross Site Scripting (XSS)	crosssite scripting xss vulnerability wpcron dashboard plugin earlier wordpress allows remote attackers inject arbitrary web script html via procname parameter wpadmintoolsphp	0	1	0	0	0	0	0	0	0	0	0	0	0	0	XSS	Cross Site Scripting (XSS)
27235	CVE-2002-0122	Denial of Service	siemens sx phone allows remote attackers cause denial service crash via sms message containing unusual characters	1	0	0	0	0	0	0	0	0	0	0	0	0	0	DoS	Denial of Service

Show 25 per page

Fig. 3. Multi-Labelled Predicted Outcomes of different vulnerabilities

C. Wordclouds

Bigrams are more preferred for this project as it is more logical and precise to a vulnerability type. Below list of figures contains all different wordclouds for all vulnerability type:

Fig. 4. Wordcloud representation of bigrams for every Vulnerability Type

VI. Conclusion and Future Work

A. Summary of Findings

- Use of Bigrams over monograms gives more clear and logical outcomes and expected results.
- Supervised Learning Models on it's best produces ceiling accuracy of 80 percent.
- These models produce decent results for some range of data but gives very poor outcome for other data.
- LSTM model without fine-tuning(using Monogram) gives very poor results.
- With fine-tuning and feature selections LSTM is expected to give best results.
- LSTM is a very computationally expensive as well as formations of bi-grams.

B. Contributions

Detailed analysis of each vulnerability type giving clear idea on each term affecting the classification of that description.
Use of Bigrams and list of all such important bigrmas for each vulnerbility type is a breakthrough in classifying vulnerabilties with high accuracy.
Use of TF-IDF to remove stopwords related to the particular document is what is implemented.This method is called bottom up approach.
This project produces a step by step procedure from pre-processing,TF-IDF vectorizing,visualization,feautre selection(bigrmas) and fine-tuning to LSTM neural networks.This can be used in any such problem statement that uses NLP as a tool.

C. Limitations

- One major issue faced is on hardware limitation of RAM for computation for large dataset.
- Other issue is time consumption as LSTM model has many hidden layer and so it takes lot of time to run.
- Also extracting the dataset using web scrapping takes more than 6 hours long!!.

D. Implications and Applications

Procedures and Concepts used in this project can be similar and used as a tool that uses NLP and can be a good tool and reference for such projects.

E. Future Work

Production of bi-grams using high configured systems and with the fine-tuned data it can be feeded to LSTM model to get a good working trained model.

F. Conclusion

Feature Engineering and Fine-tuning are two very important procedures that decides how good a model will work.
It is important to note that the quality and size of the training dataset significantly impact the performance of the classification models.
A large, diverse, and well-labeled dataset is crucial for training robust and accurate models. Additionally, fine-tuning the models and optimizing hyper-parameters can lead to further improvements in performance.

In summary, CVE classification using NLP techniques holds great promise for bolstering cybersecurity practices.

References

[1] Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). "Semantically Equivalent Adversarial Rules for Debugging NLP Models." arXiv preprint arXiv:1804.01174.

[2] Liu, Y., Li, Y., Zhao, H., Xie, S. (2020). "A Survey of Natural Language Processing Techniques for Information Security." IEEE Access, 8, 181591-181611.

[3] Liu, Y., Zhao, H., Li, Y., Wang, S., Xie, S. (2020). "A Survey of Natural Language Processing for Cybersecurity." IEEE Transactions on Information Forensics and Security, 15, 2256-2274.

[4] Rashid, F., Alazab, M., Alazab, M. (2019). "Detecting and Classifying Zero-day Attacks Using Natural Language Processing Techniques." Future Generation Computer Systems, 92, 114-127.