DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
CHENNAI – 600036

Classification of Vulnerability using NLP

A Thesis

Submitted by

DIBYENDU MANDAL

For the award of the degree

Of

DUAL DEGREE

June 2023

THESIS CERTIFICATE

This is to undertake that the Thesis titled **CLASSIFICATION OF VULNERABILITY USING NLP**, submitted by me to the Indian Institute of Technology Madras, for the award of **Dual Degree**, is a bona fide record of the research work done by me under the supervision of **Dr. Gaurav Raina**. The contents of this Thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Chennai 600036 **Dibyendu Mandal**

Date: June 2023

Dr. Gaurav Raina
Research advisor
Professor
Department of Electrical Engineering
IIT Madras

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this thesis report. Without their support, guidance, and encouragement, this endeavor would not have been possible.

First and foremost, I am immensely grateful to my supervisor, **Dr. Gaurav Raina**, for his invaluable guidance, expertise, and constant support throughout this research. His insightful suggestions and unwavering belief in my abilities have been instrumental in shaping this work.

I would like to extend my heartfelt appreciation to my mentors Aviral Verma, MD Aksam, Sandeep Challa and Prashant Bharadwaaj for their knowledge, expertise, and dedication in providing a stimulating academic environment. Their commitment to teaching and research has enriched my learning experience and broadened my perspective.

I am grateful to my friends and family for their unwavering support, understanding, and encouragement during this challenging journey.

In conclusion, I express my deepest appreciation to everyone who has contributed to this thesis report. Each person mentioned above has played a unique role in shaping my academic journey, and their support has been crucial in the successful completion of this work. Thank you all for your invaluable contributions and unwavering support.

ABSTRACT

KEYWORDS Common Vulnerabilities and Exposures (CVE), classification, Natural Language Processing (NLP), machine learning, text preprocessing, feature extraction, evaluation metrics.

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.

A comprehensive dataset comprising labeled CVE entries is collected and used for training and evaluation purposes. Various NLP algorithms, such as Bag-of-Words, word embeddings, and deep learning models, are employed to extract meaningful features from the CVE texts. These features are then used as input to train and evaluate classification models, including support vector machines, random forests, and neural networks.

This research contributes to the field of cybersecurity by providing an automated approach for CVE classification using NLP techniques. The developed system has the potential to aid security analysts, vulnerability researchers, and organizations in efficiently categorizing and prioritizing CVE entries, ultimately enhancing the overall security posture of software and hardware products.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

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.

The growing interconnectedness of our digital infrastructure has given rise to a critical need for effective vulnerability management. Cybersecurity professionals and organizations face the daunting task of dealing with a large number of vulnerabilities reported in the form of Common Vulnerabilities and Exposures (CVE) entries. These entries provide detailed information about identified security vulnerabilities and serve as a vital resource for prioritizing and addressing potential threats.

To effectively manage the ever-increasing volume of CVE entries, automated techniques are being explored to assist in their classification and prioritization. Natural Language Processing (NLP), a branch of artificial intelligence (AI) concerned with the interaction between computers and human language, has emerged as a powerful tool for analyzing and understanding textual data.

The purpose of this thesis is to explore the application of NLP techniques for CVE classification, aiming to develop a robust and efficient automated system that can accurately categorize vulnerabilities. By leveraging the power of machine learning algorithms and linguistic analysis, this research intends to provide valuable insights and solutions to enhance vulnerability management practices.

1.1 BACKGROUND AND MOTIVATION

The identification and classification of vulnerabilities play a crucial role in ensuring the security of computer systems, networks, and software applications. CVEs are widely recognized as a standardized way of reporting and sharing vulnerability information. However, the sheer volume of CVE entries, combined with the continuous evolution of cyber threats, presents a significant challenge for security professionals to triage and prioritize remediation efforts effectively.

Traditional approaches to CVE classification rely on manual analysis and expert knowledge, which are time-consuming, labor-intensive, and often prone to human error. In recent years, there has been a growing interest in leveraging AI and NLP techniques to automate the process of CVE classification, offering the potential for more efficient and accurate vulnerability management.

1.2 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.

1.3 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

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Common Vulnerabilities and Exposures (CVEs) are widely used to identify and track software vulnerabilities. CVE classification plays a crucial role in organizing and prioritizing these vulnerabilities, enabling security professionals to efficiently allocate resources and mitigate potential risks. Natural Language Processing (NLP) techniques have emerged as a powerful tool in the field of CVE classification, leveraging the textual descriptions and metadata associated with CVEs to automate the classification process. This literature review aims to explore the existing research and methodologies related to CVE classification using NLP, providing a comprehensive understanding of the current state-of-the-art methods, challenges, and opportunities in this domain.

2.2 COMMON VULNERABILITIES AND EXPOSURES (CVE)

The Common Vulnerabilities and Exposures (CVE) framework is a standardized system for identifying, naming, and classifying software vulnerabilities. CVEs provide a unique identifier for each vulnerability and include detailed descriptions, impact assessments, and relevant metadata. Understanding the structure and characteristics of CVEs is crucial for effective classification. This literature review will analyze various aspects of CVEs, including their framework, different types of vulnerabilities, and the challenges associated with CVE classification.

2.3 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, stop-word 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.

2.4 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.

2.5 EVALUATION MEASURES FOR CVE CLASSIFICATION

To assess the performance of CVE classification models, evaluation measures are employed. The literature review will discuss commonly used performance metrics, such as precision, recall, F1-score, and accuracy. Additionally, evaluation frameworks and datasets specifically designed for CVE classification will be examined. These frameworks and datasets provide benchmarks for researchers to compare the effectiveness of different classification methods and contribute to the reproducibility and comparability of results.

2.6 EXISTING RESEARCH IN CVE CLASSIFICATION USING NLP

The review will delve into the existing research on CVE classification using NLP techniques. It will analyze relevant studies and methodologies, highlighting the strengths and limitations of different approaches. Comparative analysis of these methods will be conducted, examining their performance, scalability, and applicability to real-world scenarios. The review will also discuss the challenges encountered in CVE classification, such as imbalanced datasets, domain-specific terminology, and evolving threat landscapes.

2.7 RESEARCH GAPS AND OPPORTUNITIES

Identifying research gaps and opportunities is essential for advancing the field of CVE classification using NLP. The review will highlight the current challenges faced in CVE classification and propose

potential areas for improvement. These may include addressing the limitations of existing methods, exploring novel techniques such as transfer learning and multi-modal approaches, developing more comprehensive datasets, and investigating the integration of external knowledge sources, such as vulnerability databases and security advisories.

2.8 CONCLUSION

The literature review will conclude by summarizing the key findings from the reviewed research papers and methodologies. It will provide recommendations for the thesis research on CVE classification using NLP, suggesting potential research directions and areas of focus. By consolidating the existing knowledge and identifying research gaps, this literature review will contribute to the advancement of CVE classification using NLP techniques, facilitating the development of more accurate and efficient classification models for software vulnerabilities.

CHAPTER 3

METHODOLOGY

3.1 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

3.2 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.

3.3 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.

3.4 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.

3.5 CLASSIFICATION ALGORITHMS

Naive Bayes: A probabilistic classifier based on Bayes' theorem that assumes independence among features.

Support Vector Machines (SVM): A discriminative model that finds the best hyperplane to separate different classes.

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.

Neural Networks: Deep learning models with multiple layers of interconnected nodes (neurons) that learn complex patterns.

LSTM with Finetuning: Use of bigrams over monograms provide better results and expected to provide better results here.

3.6 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.

CHAPTER 4

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.

4.1 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) ts parameter to details.php and possibly the (2) ilceler parameter to index.php.	multiple sql injection vulnerabilities mh products projekt shop allow remote attackers execute arbitrary sql commands via ts parameter detailsphp possibly ilceler 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

Figure 4.1: Pre-processed Data

4.2 SUPERVISED LEARNING MODELS

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

Table 4.1: 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.

	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 pepoly.dll in Q...	stackbased buffer overflow pepolydll 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

Figure 4.2: Random Forest Predicted results

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.

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	nall
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	afpserver 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 wap mobile phones 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

Figure 4.3: Multi-Labelled Predicted Outcomes of different vulnerabilities

4.3 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:

(a) Code Execution (b) Denial of Service (c) Overflow (d) SQL Injection

(e) Cross Site Scripting (f) Memory Corruption (g) Directory Traversal (h) CSRF

(i) Bypass (j) Security Vulnerability (k) Gain Privilege (l) File Inclusion

(m) HTTP (n) Gain Information

Figure 4.4: Wordcloud representation of bigrams for every Vulnerability Type

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 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 finetuning(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 bigrams.

5.2 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 preprocessing,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.

5.3 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!!.

5.4 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.

5.5 FUTURE WORK

Production of bigrams using high configured systems and with the finetuned data it can be feeded to LSTM model to get a good working trained model.

5.6 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 hyperparameters 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., 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.

[5] Huang, J., Li, Z., Hu, J., Qian, Y. (2019). "Automatically Classifying CVE Documents." In International Conference on Neural Information Processing (pp. 427-437). Springer, Cham.

[6] Al-Harbi, R., Alhadlaq, A., Alshehri, M., Raggad, M. (2021). "Deep Learning Approaches for Cybersecurity Threat Detection and Classification: A Survey." Computers Security, 104198.

[7] Schumacher, A., Hosseini, S., Poovendran, R. (2020). "Natural Language Processing for Cybersecurity: Methods, Tools, and Resources." IEEE Transactions on Information Forensics and Security, 15, 3616-3630.

[8] Lu, Z., Zhang, Y., Zhang, X. (2021). "A Comprehensive Survey of Natural Language Processing Techniques for Cybersecurity." IEEE Access, 9, 2976-2990.

[9] Bhattacharya, P., Srinivasan, D. (2020). "Natural Language Processing for Cybersecurity: A Comprehensive Survey." In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.

[10] Sharma, P., Saini, N., Bhatia, V., Jain, A. (2020). "Deep Learning Based Cyber Threat Intelligence: A Review and Analysis." In 2020 International Conference on Smart Electronics and Communication (ICOSEC) (pp. 395-399). IEEE.

[11] Padala, P. R., Rao, G. N., Meher, M. S. (2020). "Machine Learning and Deep Learning Techniques for Cybersecurity: A Comprehensive Review." Journal of Information Security and Applications, 54, 102577.

[12] Zhang, Y., Zhang, X., Huang, Z. (2020). "A Survey on Natural Language Processing

for Cybersecurity." In 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 55-62). IEEE.

[13] Chen, Y., Wang, S., Xu, H. (2021). "Research Progress on Natural Language Processing in the Field of Network Security." Journal of Information Security Research, 2, 132-138.

[14] Lee, J., Lee, J., Lee, K. (2018). "A Survey of Deep Learning-based Network Anomaly Detection." Cluster Computing, 21, 1693-1707.

[15] Noor, T. H., Anwar, M. W., Orgun, M. A. (2020). "A Survey on Machine Learning Techniques for Intrusion Detection Systems." ACM Computing Surveys (CSUR), 53(6), 1-35.

[16] Mohammadzadeh, S., Shamsinejadbabaki, P., Sharifi, M. (2020). "A Comprehensive Study on Natural Language Processing Techniques for Cybersecurity." In 2020 15th International Conference on Computer Science Education (ICCSE) (pp. 49-54). IEEE.

[17] Basu, S., Mehta, M. (2020). "Deep Learning for Cybersecurity Threat Detection in IoT Networks." In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[18] Imran, M., Vasterlund, M. (2020). "Deep Learning-Based Intrusion Detection System for Secure Industrial Internet of Things." In 2020 IEEE Industrial Cyber-Physical Systems (ICPS) (pp. 360-365). IEEE.

[19] Ruan, S., Feng, T., Zhang, J. (2020). "Research on Cyber Threat Intelligence Fusion Based on Natural Language Processing and Machine Learning." In 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA) (pp. 259-263). IEEE.

[20] Thapa, B., Shrestha, A. (2021). "A Systematic Review of Machine Learning Techniques for Detecting Insider Threats in Cybersecurity." In 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 1-6). IEEE.