JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Malware Analysis & Detection using Machine Learning

Parshva Doshi, Darsh Patel, Vishal Padia, Omkar Solanki

Cyber Security,
Shah and Anchor Kutchhi Engineering College, Mumbai, India

Abstract : In recent years, malware has grown to be one of the biggest risks to computer security. Using signature-based techniques, which are useless against fresh and previously undiscovered infection, is the conventional method of identifying malware. Techniques for machine learning (ML) have become a promising replacement for conventional approaches.

The capacity of machine learning algorithms to detect previously undiscovered malware, even when it has not yet been recognised by signature-based techniques, has attracted considerable attention. The calibre of the characteristics that are collected from the malware samples determines how well ML-based malware detection systems perform.

A crucial stage in the creation of ML-based malware detection systems is feature engineering. The model should be trained using informative, discriminative characteristics Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for example, are deep learning-based techniques that have been used to detect malware recently. These methods have demonstrated encouraging outcomes in identifying malware with high accuracy and low false positive rates.

The application of machine learning techniques for malware detection is, all things considered, a promising strategy that has the potential to dramatically increase the efficiency of malware detection systems. Yet, the creation of efficient ML-based malware detection systems necessitates careful feature engineering, the application of suitable machine learning techniques, and the availability of substantial, high-quality dataset

IndexTerms **- Component,formatting,style,styling,insert.**

I. INTRODUCTION

Malware is any software created with the intention of damaging computer systems, networks, or gadgets. Malware attacks are getting harder to find and stop as they multiply. The ever-evolving nature of malware makes it impossible to combat with conventional signature-based antivirus technologies. A strategy that seems promising for detecting and classifying malware is machine learning (ML). ML systems can be trained on massive datasets of known malware samples to discover patterns and attributes that identify them from benign software. To categorize new samples as malicious or benign, the trained model can then be used. Malware analysis is the process of examining malicious software to identify its characteristics, behavior, and potential impact on systems and networks. Malware detection, on the other hand, involves identifying and removing malicious software from a system or network. Traditional approaches to malware analysis and detection involve using signature-based methods, which rely on a predefined set of patterns or signatures to detect known malware. However, these methods have limitations in detecting new and unknown malware. In recent years, machine learning has emerged as a promising approach for malware analysis and detection. Machine learning algorithms can be trained to analyze patterns in large datasets and identify potential malware based on their behavior or characteristics. This approach has the potential to detect new and unknown malware that traditional signature-based methods might miss. A research paper on malware analysis and detection using machine learning would aim to explore the various techniques and approaches used in this field. It would cover topics such as data preprocessing, feature selection, and the various machine learning algorithms used for malware analysis and detection. The paper would also discuss the challenges and limitations of using machine learning for malware analysis and detection, such as the need for large datasets and the risk of false positives and false negatives.

ML-based malware detection techniques can be broadly divided into two groups: supervised and unsupervised learning. With supervised learning, the ML system is trained using labelled datasets, with each sample tagged as malware or benign. Based on the features it has acquired from the labelled dataset, the algorithm operates to categorise fresh samples. Unsupervised learning teaches the system to spot errors that occur that might be signs of malware. Malware detection systems based on machine learning (ML) can be very good at finding both known and undiscovered malware, particularly malware that is engineered to avoid detection by signature-based methods. They may, however, also be susceptible to false positives and false negatives. Because of this, it's crucial to carefully choose and enhance the ML algorithms, features, and training datasets to achieve excellent precision and effectiveness. Furthermore, ML models should be updated and retrained on a frequent basis to keep up with the shifting nature of malware.

1.2 PROBLEM DEFINITION

The challenge of malware detection with machine learning is to create a model that can accurately differentiate between dangerous and benign software. A collection of labeled samples that includes both malware and genuine software must be used to train the model. The idea is to train the model to recognize malware-specific traits or patterns and then use this information to forecast whether fresh, unknown software is harmful.

Malware is a significant threat to computer systems and networks, and the number and complexity of malware variants are increasing rapidly. Malware can be used for a variety of purposes, including stealing sensitive information, disrupting operations, and compromising system integrity. Malware authors constantly change their tactics and techniques to evade detection and analysis, making it a challenging problem to detect and mitigate malware.

The problem is to develop effective techniques and tools for detecting and analyzing malware and providing effective countermeasures to prevent or mitigate its effects. The goal is to detect malware quickly and accurately, understand its behavior, and develop effective strategies to mitigate its effects. The problem requires a multi-faceted approach, including signature-based detection, behavioral analysis, machine learning, and threat intelligence.

Effective malware analysis and detection techniques must be able to handle the volume and diversity of malware, identify new and emerging threats, and provide timely and accurate responses. The challenge is to develop scalable, efficient, and accurate techniques and tools for malware analysis and detection that can keep up with the evolving threat landscape. Ultimately, the goal is to ensure the security and privacy of computer systems and networks and protect them from potential threats.

This issue presents a variety of difficulties, including the fact that malware is continually changing and that new varieties are continuously being developed. Furthermore, there is frequently a big disparity between the quantity of malware and legitimate software samples in a dataset, which might make it difficult to detect.

1.3 OBJECTIVE

1. With the increasing use of internet there has been an increase in malicious actors as well. This project aims to detect these Malicious Files.

2. Traditional signature-based malware detection approaches can be easily circumvented by attackers who can slightly alter the harmful software in order to avoid detection. In such cases Machine Learning comes in handy.

3. The use of Machine Learning can help detect malware new, which are yet not discovered by signature-based malware detection approaches.

4. Another objective of the project is to provide insights into the latest trends and developments in the field.

5. This project also aims to provide a detailed analysis of the file. It includes various details such as the type of malware, file hash, the number of bad actors in the file, etc.

6. Overall, this project aims to detect various kinds of malicious files. This will help make a robust system that will help spread awareness amongst the users and help protect them from malware.

1.4 MOTIVATION

From a cybersecurity perspective, 2022 was an unfavourable year on many counts. The number of security threats was more than the previous year. Cybersecurity company Kaspersky's systems detected 400,000 new malicious files daily on average in the past 10 months. This is a 5% increase from 380,000 new files detected per day in 2021. Overall, the number of malicious files detected stood at 122 million, 6 million more than the last year, revealed as part of the Kaspersky Security Bulletin (KSB) — an annual series of reports on key shifts in cybersecurity. [23]

According to Tripwire, 82% of respondents to surveys conducted by Cybersecurity Insiders and HelpSystems for their 2021 Malware report, anticipate more ransomware and malware attacks, and another 75% are certain that this threat will cause more problems for businesses over the course of the following few years. [24]

1.5 CONCLUSION

In recent years, machine learning has emerged as a promising approach for malware analysis and detection. Machine learning algorithms can be trained to analyze patterns in large datasets and identify potential malware based on their behavior or characteristics. This approach has the potential to detect new and unknown malware that traditional signature-based methods might miss. The challenge of malware detection with machine learning is to create a model that can accurately differentiate between dangerous and benign software. A collection of labeled samples that includes both malware and genuine software must be used to train the model. The idea is to train the model to recognize malware-specific traits or patterns and then usethis information to forecast whether fresh, unknown software is harmful. The goal of malware detection using machine learning is to enhance detection accuracy and efficiency. Traditional signature-based malware detection approaches can be easily circumvented by attackers who can slightly alter the harmful software in order to avoid detection. Motivation behind this Traditional malware detection techniques, such as signature-based detection, are becoming less effective as malware authors use more advanced techniques to evade detection. An ML-based approach

can overcome the limitations of traditional techniques by identifying patterns and features that are difficult to detect using traditional methods. In next phase we did comparative analysis with different research paper.

Chapter 2 Review of Literature

CHAPTER - 2

2.1 LITERATURE REVIEW

"Malware detection using machine learning algorithms : A review" by Patil and Jagtap (2021): This review paper summarizes the state-of-the-art in malware detection using machine learning and provides an overview of the different types of machine learning algorithms that have been used. The authors highlight the importance of feature selection and dimensionality reduction in improving the accuracy of malware detection.[1]

"Malware detection using machine learning: A survey" by Roy et al. (2019): This survey paper provides an overview of the different approaches that have been used for malware detection using machine learning, including static analysis, dynamic analysis, and hybrid approaches. The authors also discuss the challenges and limitations of machine learning- based malware detection.[2]

"A Comprehensive Survey on Malware Detection using Machine Learning Techniques," by A. Sharma, A. Tiwari, and A. Singh (2021): The authors provide an overview of different machine learning techniques used for malware detection. They discuss different types of features, such as static and dynamic, that can be extracted from malware samples to train machine learning models. The paper also covers various datasets, including the Malware Genome Project, Drebin, and AndroZoo, which are commonly used for training and testing machine learning models. The authors conclude by highlighting some of the challenges and future research directions for malware detection using machine learning. [3]

"Malware Detection using Deep Learning Techniques: A Systematic Literature Review," by S. P. Shah, S. S. Dhamecha, and S. K. Patel (2020): The authors present a systematic literature review of deep learning techniques for malware detection. Theyanalyze different approaches, including CNNs, RNNs, LSTM, and autoencoders, that have been used for feature extraction and classification of malware samples. The authors conclude that deep learning techniques offer better accuracy and performancein malware detection compared to traditional machine learning methods. [4]

"Malware Detection using Convolutional Neural Networks and Support Vector Machines," by R. K. Ghorpade and P. B. Mane (2020): The authors propose an approach for malware detection using CNNs and support vector machines (SVMs). They use CNNs for feature extraction and SVMs for classification of malware samples. The

authors also evaluate their approach on the Malware Genome Project dataset and demonstrate better accuracy than other traditional machine learning approaches. [5]

"Deep Learning-Based Malware Detection: A Comprehensive Review," by L. Zhi, Y. Pan, and L. Liu (2021): The authors provide a comprehensive review of deep learning- based malware detection techniques. They analyze different types of neural networks and their applications in malware detection. The paper also discusses different feature extraction techniques, such as static and dynamic analysis, and their impact on malware detection accuracy. The authors conclude by highlighting the importance of addressing challenges in data labeling and adversarial attacks in future research. [6]

"Malware Detection using Machine Learning: A Survey," by Arpita Roy, Maitrayee Das, and Pranab Kumar Das (2019): The authors conducted a comprehensive survey of research on malware detection using ML. They discussed various ML algorithms, such as decision trees, random forests, support vector machines, and neural networks, that have been employed for malware detection. They also reviewed the datasets commonly used for training and testing these algorithms, including the MalGenome, Drebin, and AndroZoo datasets. The authors highlighted the challenges associated with using ML for malware detection, such as the difficulty of constructing a representative dataset and the potential for adversarial attacks. [7]

"A Novel Framework for Malware Detection using Deep Learning Techniques," by A. Abraham, A. Patnaik, and N. Kar (2020): The authors proposed a novel framework for malware detection using deep learning techniques. Their framework involved feature extraction using convolutional neural networks (CNNs) and classification using support vector machines (SVMs). They evaluated their approach using the Drebin dataset and achieved an accuracy of 98.3%. [8]

"Malware Detection using Ensemble of Deep Neural Networks," by S. K. Gupta and A. Shukla (2021): The authors proposed an ensemble of deep neural networks (DNNs) for malware detection. Their approach involved extracting features from the opcode sequence of an executable file and using multiple DNNs to classify the file as either malicious or benign. They evaluated their approach using the Malware Capture Facility Project dataset and achieved an accuracy of 99.5%. [9]

"Adversarial Machine Learning in Malware Detection: A Survey," by J. Ma, W. Liu, Y. Chen, and C. Yang (2021): The authors conducted a survey of research on adversarial machine learning (AML) in malware detection. AML involves attacking ML models to cause misclassification of inputs. The authors discussed various AML techniques, such as evasion and poisoning attacks, that can be used against ML models for malware detection. They also highlighted the potential of AML to undermine the effectiveness of ML-based malware detection systems. [10]

2.2 COMPARATIVE ANALYSIS

Table 2.1 Comparative Analysis

2.3 CONCLUSION

We used the most popular XGBoost algorithm which has the advantages of high

efficiency and high accuracy. We got an accuracy rate of 98.4% with it while with Decision Tree we achieved an accuracy of 93%.

XGBoost can be used in malware analysis and detection because it is a powerful machine learning algorithm that is well-suited for classification tasks.

Chapter 3 Proposed System

CHAPTER - 3

In the figure 3.1 we explain why, Malware analysis and detection systems are essential for protecting computer systems from malicious software that can cause significant harm. Machine learning (ML) is a popular approach for designing such systems, as it can learn to recognize patterns in malware behavior and help identify new strains of malware. Additionally, integrating external tools and APIs like VirusTotal and YARA rules can provide an additional layerof protection.

The design of a malware analysis and detection system using ML and external tools can be broken down into several key components. Firstly, the system should be able to gather malware samples for analysis. This can be done through a variety of means, such as email attachments, downloads, or uploads from users. Once the system has collected the samples, it can use ML algorithms to analyze the behavior and characteristics of the malware.

To improve the accuracy of the system, integrating an API like VirusTotal can be helpful. VirusTotal provides a vast database of malware samples and can help identify known strains of malware. By comparing the behavior of new malware samples with those in the VirusTotal database, the system can detect and categorize new malware strains.

Another critical component of the system is the use of YARA rules. YARA is a powerful tool for creating custom rules to detect malware based on specific behavior or characteristics. By incorporating YARA rules into the system, it can be customized to detect specific malware strains or behaviors that are unique to an organization or industry.

The system can be designed using a web framework like Django, which can provide a user-friendly interface for interacting with the system. Users can submit malware samples for analysis and view the results in real-time. The system can also generate reports and alerts to notify users of any new malware detections.

In conclusion, a malware analysis and detection system that uses ML, external tools like VirusTotal, and YARA rules can provide a comprehensive defense against malicious software. The system can be designed using a web framework like Django to provide a user-friendly interface for interacting with the system. With the ability to analyze and categorize new malware strains, the system can help organizations stay protected against ever-evolving threats.

3.3 Methodology / Algorithm

Malware analysis and detection systems are essential for protecting computer systems from malicious software that can cause significant harm.

The following are the steps we used to develop the machine learning model.

Data Collection: Collect a large dataset of malware and benign samples. Malware samples can be obtained from online repositories, while benign samples can be collected from trusted sources.

Data Preprocessing: Preprocess the collected data to extract the relevant features. This could include file size, entropy, file type, system calls, API calls, and other

metadata.

Feature Selection: Select the most relevant features that are likely to distinguish between malware and benign samples. This can be done using various feature selection techniques, such as mutual information, correlation, and feature importance.

Model Selection: Select an appropriate machine learning model for classification. XGBoost is one such algorithm that can be used, but other algorithms like SVM, Random Forest, and Neural Networks can also be considered.

Model Training: Train the selected machine learning model using the preprocessed and selected features. Split the data into training and testing sets, and use the training set to fit the model.

Model Evaluation: Evaluate the performance of the trained model on the testing set. Common metrics used for evaluation include accuracy, precision, recall, and F1- score.

Deployment: Once the model is trained and evaluated, it can be deployed for real- time malware detection. This can be done by integrating the model into an antivirus software or a network security system. It can then be linked to our frontend Django Web App.

3.4 CONCLUSION

We have successfully implemented our model and received accurate results. Our model can detect Malware for most of the files with a satisfying accuracy.

Chapter 4 Implementation & Results

CHAPTER - 4

4.1 Software Requirements

Scikit-learn: This is a popular machine learning library that provides tools for data preprocessing, feature selection, and model training and evaluation. Scikit-learn includes many common machine learning algorithms, such as logistic regression, decision trees, and random forests.

Web Browser: Any web browser which supports JavaScript and iFrames should be enough to run the application. Example: Vivaldi, Edge, Chrome, Firefox, Brave, Opera GX, etc.

Pandas: This library provides tools for data manipulation and analysis. Pandas can be used to load and preprocess data, as well as to extract and select features for machine learning models.

NumPy: This library provides tools for numerical computing in Python. NumPy can be used to perform mathematical operations on arrays of data, which is useful for machine learning tasks such as computing feature vectors

Virustotal API: This is an API that allows users to query the Virustotal database for information about malware samples. Virustotal aggregates data from multiple antivirus engines and other sources to provide a comprehensive view of the malware landscape.

YARA: This is a pattern matching tool that can be used to generate signatures for malwaresamples. YARA can be used to identify unique strings or patterns associated withmalware, which can then be used to develop custom signatures for detection.

Django: This is a web framework for Python that can be used to build web-based malware analysis and detection systems. Django provides tools for building user interfaces, managing databases, and handling user authentication and authorization.

4.2 Hardware Requirements

RAM: Any device with a browser and 4GB RAM should be enough to run the application.

Storage: Any device with 10GB Storage Capacity should be enough to run the application.

Chapter 5 Conclusion & Future Scope

CHAPTER - 5

5.1 CONCLUSION

In conclusion, malware analysis and detection using machine learning is a critical aspect of modern cybersecurity. With the constant evolution of malware and cyber attacks, organizations must develop robust and effective systems to protect their systems and data. By leveraging machine learning algorithms, organizations can quickly and accurately identify malware and mitigate the risks associated with these threats.

There are two primary approaches to malware analysis and detection using machine learning: signature-based and behavior-based. Signature-based approaches rely on the identification of unique patterns or strings associated with malware, while behavior-based approaches identify behaviors associated with malware.

To implement a successful malware analysis and detection system using machine learning, developers must gather relevant data, preprocess it, train and test machine learning models, and refine the system over time. They must also utilize various libraries and tools, such as Scikit-learn, TensorFlow, Pandas, NumPy, Virustotal API, YARA, and Django.

Overall, building a successful malware analysis and detection system is a complex and ongoing process. By following best practices and utilizing the right tools and techniques, developers can develop effective systems that help protect organizations from the ever- evolving threat of malware.

5.2 FUTURE SCOPE

Improving the ML model: We would work on improving the accuracy of the ML model by incorporating more data, optimizing hyper parameters, or using advanced techniques like ensemble learning or deep learning. By doing this, we could create a more powerful and accurate model.

Integrating the ML model with the web application: After building a robust ML model, we would integrate it with a web application to make it accessible to a wider audience. To do this, we could use web development frameworks like Flask, Django, or Node.js to build a web application that uses our ML model to provide predictions or recommendations.

Deploying the ML model on cloud: We would deploy our ML model and web application on cloud platforms like AWS, Google Cloud, or Microsoft Azure to ensure that our application is scalable, secure, and can handle a large number of users.

Implementing continuous learning: We would implement continuous learning in our ML model by regularly updating it with new data. By doing this, we could ensure that our model stays up-to-date and can make accurate predictions even with new and unseen data.

Integrating the ML model with other systems: We would integrate our ML model with other systems like CRM, ERP, or BI to enhance their capabilities. For example, we could integrate our ML model with a CRM system to provide personalized recommendations to customers based on their buying behavior.

To integrate our ML model with a web application, we would use APIs or web services to enable communication between the ML model and the web application. We could also use containers like Docker to package our ML model and deploy it on the web server. Finally, we would ensure that our web application is secure and can handle user input and output appropriately.

Advantages:

1. Machine learning systems can detect malware with high accuracy. This is because they can find patterns in huge datasets that human analysts might not be able to see.

2. Automated detection: Without human assistance, machine learning algorithms can be trained to recognise malware.

5.3 LIMITATION

Data quality: The accuracy and effectiveness of the ML model largely depend on the quality of the data used for training. Poor quality data can lead to inaccurate predictions, which can affect the overall performance of the ML model.

Scalability: Integrating an ML model with a web application can lead to scalability issues, especially when there is a high volume of requests. This can affect the performance of the web application and lead to slow response times.

Maintenance: ML models require regular maintenance to ensure that they remain up-to- date and continue to provide accurate predictions. This can be time-consuming and require significant resources.

Interpretability: ML models are often considered "black boxes," meaning it can be difficult to understand how they arrive at their predictions. This can be a limitation in certain applications where interpretability is important.

Security: Integrating an ML model with a web application can also pose security risks. It is important to ensure that the application is secure and that user data is protected from potential attack

Chapter 6 References

1. Chowdhury, Naseef & Haque, Ahshanul & Soliman, Hamdy & Hossen, Mohammad Sahinur & Ahmed, Imtiaz & Fatima, Tanjim. (2023). Android malware Detection using Machine learning: A Review. 10.36227/techrxiv.22580881.

2. Joshi, Sakshi & Mahagaonkar, Santosh. (2022). MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES. International Journal of Engineering Applied Sciences and Technology. 7. 86-91. 10.33564/IJEAST 2022.v07i05.014.

3. K, Poojitha. (2022). Detection of Malware in Android Phones Using Machine Learning. International Journal for Research in Applied Science and Engineering Technology. 10. 3344-3347. 10.22214/ijraset.2022.45726.

4. Rimon, Saiful & Haque, Md. (2022). Malware Detection and Classification Using Hybrid Machine Learning Algorithm. 10.1007/978-3-031-19958-5_39.

5. Kamboj, Akshit & Kumar, Priyanshu & Bairwa, Amit & Joshi, Sandeep. (2022). Detection of malware in downloaded files using various machine learning models. Egyptian Informatics Journal. 24. 10.1016/j.eij.2022.12.002.

6. Muppalaneni, Naresh & Patgiri, Ripon. (2021). Malware Detection Using Machine Learning Approach. 10.1007/978-981-33-4788-5_18.

7. Shukla, Amogh & Chettiar, Gautam & Choudhary, Ayush & Thakur, Ayush & Kumar, Shubham. (2022). Integrating Comparison of Malware Detection Classification using LGBM and XGB Machine Learning Algorithms. 10.1109/ICBDS53701.2022.9935976.

8. Tahir, Inshal & Qadir, Sana. (2022). Machine Learning-based Detection of IoT Malware using System Call Data. 10.21203/rs.3.rs-2384013/v1.

9. Shah, Syed Shakir Hameed & Ahmad, Abd & Jamil, Norziana & Khan, Atta ur Rehman. (2022). Memory Forensics-Based Malware Detection Using Computer Vision and Machine Learning. Electronics.

10. Muhamad Malik Matin, Iik & Rahardjo, Budi. (2019). Malware Detection Using Honeypot and Machine Learning. 1-4. 10.1109/CITSM47753.2019.8965419.

11. Ibrahim, Mohammed & Abdullahi, Abdullahi & Ahmad, Muhammad & Mustapha, Rabi & Ng, & Ibrahim, Mohammed. (2023). A Comparative Analysis of Android Malware Detection with and without Feature Selection Techniques using Machine Learning. 10.56471/slujst.v6i.371.

12. Hussain, Abrar & Asif, Muhammad & Ahmad, Maaz & Mahmood, Toqeer & Raza, Muhammad. (2022). Malware Detection Using Machine Learning Algorithms for Windows Platform. 10.1007/978-981-16-7618-5_53.

13. Kumar, Manish. (2022). Scalable malware detection system using big data and distributed machine learning approach. Soft Computing. 26. 1-17. 10.1007/s00500- 021-06492-9.

14. MAIL, MOHD & Faizal, Mohd & RAHMAN, MUNIRAH. (2022). Malware Detection System Using Cloud Sandbox, Machine Learning. International Journal of Software Engineering and Computer Systems. 8. 25-32. 10.15282/ijsecs.8.2.2022.3.0100.

15. Datta, Arkajit & Anil Kumar, Kakelli & D, Aju. (2021). An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis.

16. Akhtar, Muhammad Shoaib, and Tao Feng. 2022. "Malware Analysis and Detection Using Machine Learning Algorithms" Symmetry 14, no. 11: 2304.
https://doi.org/10.3390/sym14112304

17. Akhtar, M.S.; Feng, T. Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry 2022, 14, 2304. https://doi.org/10.3390/sym14112304

18. Dukka KarunKumar Reddy, Himansu Sekhar Behera, Janmenjoy Nayak, Pandi Vijayakumar, Bighnaraj Naik and Pradeep Kumar Singh
Journal: Transactions on Emerging Telecommunications Technologies, 2021, Volume 32

19. ZhaoqiZhang,PanpanQi,WeiWang
SchoolofComputing NationalUniversityofSingapore zhaoqi.zhang@u.nus.edu,qipanpan@u.nus.edu,wangwei@comp.nus.edu

20. F. Mira, "A Systematic Literature Review on Malware Analysis," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 2021, pp. 1-5,
doi:10.1109/IEMTRONICS52119.2021.9422537.

21. Miuyin Yong Wong, Matthew Landen, Manos Antonakakis, Douglas M. Blough, Elissa M. Redmiles, Mustaque Ahamad, Georgia Institute of Technology, United States, Max Planck Institute for Software Systems

22. M. F. Ismael and K. H. Thanoon, "Investigation Malware Analysis Depend on Reverse Engineering," 2022 International Conference on Data Science and Intelligent Computing (ICDSIC), Karbala, Iraq, 2022, pp. 251-256, doi: 10.1109/ICDSIC56987.2022.10076144.

23. Norouzi M, Souri A, Samad Zamini M (2022) A data mining classification approach for behavioral malware detection.J Comput Netw Commun 2022:9

24. P. Singh, S. Kaur, S. Sharma, G. Sharma, S. Vashisht and V. Kumar, "Malware Detection Using Machine Learning," 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 2021, pp. 11-14, doi: 10.1109/ICTAI53825.2021.9673465.

25. Patil, Rajvardhan & Deng, Wei. (2020). Malware Analysis using Machine Learning and Deep Learning techniques. 1-7.
10.1109/SoutheastCon44009.2020.9368268.

26. "Malware detection using machine learning: A survey" by Roy et al. (2019)

27. "Malware Detection using Convolutional Neural Networks and Support Vector Machines," by R. K. Ghorpade and P. B. Mane (2020)

28. "Malware Detection using Deep Learning Techniques: A Systematic Literature Review," by S. P. Shah, S. S. Dhamecha, and S. K. Patel (2020)

29. "Adversarial Machine Learning in Malware Detection: A Survey," by J. Ma, W. Liu, Y. Chen, and C. Yang (2021)

30. https://www.techcircle.in/2022/12/09/at-least-400-000-new-malicious-files- detected-daily-in-2022-report

31. https://techwireasia.com/2022/08/malware-attacks-are-here-to-stay-and-have-a- new-target-in-its-line-of-sight/

32. https://developers.virustotal.com/reference/overview

33. https://github.com/Yara-Rules/

34. https://scikit-learn.org/stable/

35. https://www.djangoproject.com/

36. https://yara.readthedocs.io/en/stable/

37. https://github.com/jonaslejon/malicious-pdf

Acknowledgement

We have great pleasure in presenting the project on "Malware Detection using Machine Learning". We take this opportunity to express our sincere thanks to our Guide, Dr. Nilakshi Jain & Co-Guide Ms. Prajakta Pote, the faculty in the Department of Cyber Security in Shah and Anchor Kutchhi Engineering College for guiding us and suggesting regarding the line of work. We would like to express our gratitude towards their constant encouragement, support and guidance throughout the progress.Also, we would like to thank our Principal – Dr. Bhavesh Patel and Dr. Nilakshi Jain, Head of Cyber Security Department, for their help, support & guidance for this project.We are also thankful to all Faculty members of our department for their help and guidance during completion of our project