2023 World Conference on Communication & Computing (WCONF)
Raipur, India. July 14-16, 2023

A Robust Approach for Malware Attacks Detection in the Internet of Things Communications

Chhaya Negi*, Amit Kumar Mishra† *Student Member, IEEE*, Anshika Verma*,
Zoya Ganguli*, Siddhant Thapliyal* *Student Member, IEEE*,
Mohammad Wazid*, *Senior Member, IEEE*, and D. P. Singh*, *Member, IEEE*
* "Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India"
† "Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India,"
"Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun 248 002, India"
E-mails: *chhayanegi1224@gmail.com, †amitmishraddun@gmail.com, *28.anshikaverma@gmail.com,
*gangulizoya@gmail.com, *sthapliyal37@gmail.com,
*wazidkec2005@gmail.com, *devesh.geu@gmail.com

***Abstract*—The detection of malicious software is an essential activity for ensuring the ongoing security of the Internet of Things (IoT). Malware, often known as malicious software, is computer code designed to steal private information, corrupt data or files, or otherwise interfere with the functioning of a computer system. A number of different methods, including signature-based detection, heuristic analysis, and behaviour-based detection, are utilised in the process of identifying malicious software. The process of searching for specific patterns or signatures that are characteristic of known malware is what is referred to as signature-based detection. The success of this method is dependent on the upkeep of a database containing known malware signatures. Heuristic analysis, on the other hand, uses rules and algorithms to detect suspicious behaviour that may indicate the presence of malware. This approach does not rely on specific signatures but instead analyzes the code or behaviour of a file or program. If the behaviour exhibits characteristics commonly associated with malware, it is flagged as potentially malicious. Behaviour-based detection focuses on analyzing the behaviour of software programs to identify anomalous or malicious activity. Utilizing machine learning algorithms for malware detection helps to identify potential threats and prevent them from compromising valuable data and information. In this paper, we propose a robust approach for the detection and analysis of malware attacks in the communications of the IoT. In the proposed protocol various machine learning-based models, i.e., k-nearest neighbours (KNN), random forest and neural networks are used. The proposed protocol is also compared with the other similar existing protocols and it has achieved better accuracy as compared to the other existing protocols.**

***Index Terms*—Malware, malware detection, machine learning, Random Forest, Logistic Regression.**

I. Introduction

Data and information have become valuable assets in today's world, and ensuring their security is of important. Malware refers to malicious software designed to harm computer systems, servers, and networks by stealing data and causing disruptions [1]. Its primary objectives include causing damage, stealing information, and disrupting the normal functioning of a system. Malware can manifest in various forms, including viruses, worms, and spyware, each with its own specific characteristics and potential for harm. Individuals and businesses alike can suffer significant consequences from malware attacks [2]. As a result, it is crucial to take measures to protect against malware attacks and ensure the security of data and information. Malware can infiltrate a system through various methods, including phishing emails, infected attachments, and malicious links [3].

Depending on the way of spreading, the malware attacks can be classified as follows.

- **Ransomware:** The ransomware is hidden in web links or files attached to the mail, and it prevents humans from accessing data until the ransom is paid.
- **Spyware:** It is a hidden software that keeps an eye on the system from the background and collects data without the users' knowledge.
- **Worms:** It spreads in the system by replicating itself.
- **Adware:** This type of malware serves unwanted advertisements.
- **Trojans:** It creates a backdoor entry in the system which allows other malware to enter the system.
- **Botnets:** These malware attacks are the most dangerous due to their ability to perform brute force and intelligent attacks on a large number of hosts at once.
- **Virus:** It usually appears in executable files, affecting the files on the system after the executable file is activated.
- **RootKits:** Allows hackers to control a victim's device remotely.

Algorithm for machine learning comes in a variety of types that can be utilized for malware detection, comprising supervised learning, unsupervised learning, and deep learning, supervised learning involves training the model on labelled data where each sample is classified as malware or benign. Unsupervised learning, on the other hand, does not rely on labelled data and instead attempts to identify clusters of similar software. The process of machine learning-based malware detection involves several stages, including data pre-processing, feature selection, model training, and testing.

By incorporating machine learning into malware detection, the effectiveness of traditional antivirus systems can be significantly improved. While conventional antivirus systems

979-8-3503-1120-4/23/$31.00 ©2023 IEEE

typically have an accuracy rate of around 90%, implementing machine learning approaches can boost accuracy by 3% to 4%. This improvement is achieved by leveraging the ability of ML models to learn complex patterns and adapt to evolving malware threats. Overall, the objective of using machine learning for malware detection is to develop a system that can accurately and efficiently identify malicious content in files or network traffic, thereby enhancing the security of computer systems and networks.

II. Literature review

The development of computer malware has increased exponentially over the past decade. Nowadays, attacks on computer systems are often carried out by cybercriminals (attackers) using malware. Computer systems are typically attacked via email, malicious websites, malicious software downloads, and drive-by attacks via the Internet. Ransomware, viruses, trojan horses, worms, rootkits, and adware are all examples of malicious software. Analyses of malware and benign samples are performed using static or dynamic analysis methods. It is possible to distinguish malware from benign files after analyzing them using unique features. Through the analysis techniques, the effectiveness of malware detection depends on the ability to extract discriminative malware features.

An investigation on malware carried out by Kambar *et al.* [4]. They talked about numerous mobile applications that contain private data has emerged as a direct result of the proliferation of both mobile technology (smartphones) and high-speed internet access. These two factors have contributed to the broad availability of both of these factors. Research shows that the frequency of incursions and attacks using mobile applications is increasing, despite the fact that prominent mobile operating systems such as iOS and Android are regularly upgrading the protection methods they utilise.

According to Sharma *et al.* [5] the paper has proposed many anti-malware tools are used in today's digital world, but they depend on stamps for detection, which are not up to date to expose progressive unfamiliar malware. Malware that transforms into metamorphic forms.

Kumar *et al.* [6] the paper has proposed examined the application of deep learning methods in the detection and classification of malware, highlighting their distinctions from other approaches. The study aimed to provide a comprehensive overview of different machine learning paradigms utilized in malware detection and offer practical recommendations for selecting the most suitable paradigms. The research focused on analyzing studies that employed deep learning techniques such as deep neural networks, convolutional neural networks, and recurrent neural networks for malware detection. The study assessed various aspects, including the algorithms used, input data representing malware, sample sizes, evaluation metrics, achieved accuracy, and additional tools employed in the detection process. By evaluating these factors, the research aimed to derive valuable insights into the most effective approaches for detecting malware using machine learning paradigms.

TABLE I
Comparison of reviewed protcols

Author	Description	Accuracy
Darem *et al.* [9]	They presented a semi-supervised approach for obfuscated malware detection that used deep learning, feature engineering, image transformation, and image processing techniques. The proposed method was thoroughly validated through experiments, demonstrating superior performance compared to existing methods in the field.	99.37%
Sewak *et al.* [10]	They proposed a combination of different deep learning algorithms to create a deep learning-based malware detection system.	98.00%
Krithika and Vijaya [11]	They proposed an ML framework that leverages the power of deep learning algorithms to identify malicious software. The study aimed to improve the accuracy of malware detection while reducing false positive rates.	93.00%
Saxe and Berlin [12]	They proposed that deep neural networks have been used to detect malware at an extremely low positive rate, resulting in a usable detection rate.	95.00%

Later on Muzaffar *et al* [7] presented According to estimates, around 70% of mobile phone users own an Android handset. Because of its ubiquity, the Android operating system draws a large number of malware threats. Because of the sensitive nature of the data on cell phones, it is critical to safeguard against these assaults. When faced with a large number of users and apps, traditional signature-based detection algorithms fall short. Machine learning, on the other hand, appears to function effectively and aids in the detection of zero-day attacks since it does not require an existing collection of harmful signatures. Urooj *et al.* [8] presented a framework Android is one of the most popular smartphone operating systems. This is the primary reason why Android has become a popular target for hackers and attackers. Malicious algorithms are being hidden in Android apps in such sophisticated ways that detecting and classifying an app as malware has become the most difficult task for security companies. Android malware has advanced in terms of cleverness and cognition to the point where it is increasingly resistant to standard detection procedures. Machine learning-based approaches have evolved as a far more effective manner of dealing with the complexities and uniqueness of evolving Android threats.

III. Details of the proposed protocol

The proposed protocol is executed through various steps, which are described below.

The network model of the proposed scheme is given in Fig. 1. The various IoT devices are deployed in various remote areas as sensing units, which transmit their data to the connected servers, where the received data is used for various purposes, i.e., monitoring, prediction and analysis.

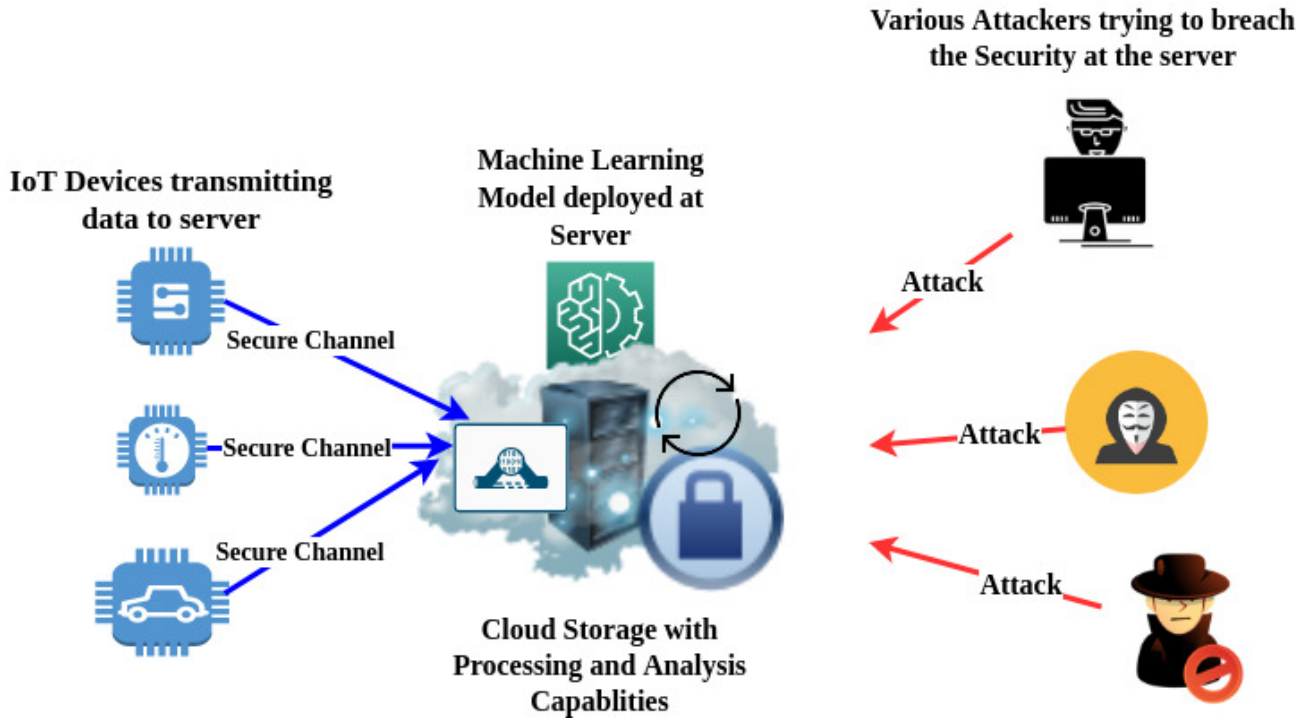

Fig. 1. Network model of the proposed protocol

As the servers are capable of processing and analyzing the data received and stored. Also, the machine learning models deployed at the servers, which help in the detection of malware attacks.

The process flow diagram of the proposed mechanism is given in Fig. 2.

Machine learning algorithms are designed to automatically determine optimal parameters from data in order to make accurate predictions. The proposed mechanism uses machine learning algorithms to classify binary files as either legitimate or malicious. The given model contains three main modules: a front-end module and two back-end modules. The front-end module gives a user interface for the user to input the dataset that needs to be analyzed for malicious activity. The two back-end modules are the train and test modules. The training module is used to train our model using various algorithms and the accuracy of the model is calculated. The test module is used to evaluate the model and generate the final results.

The phases of the proposed mechanism are given below.

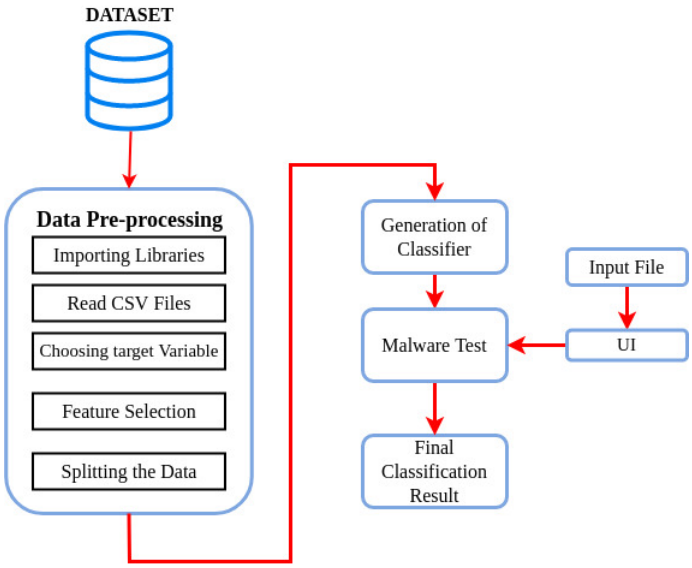

Fig. 2. Flowchart of the proposed mechanism

A. *Data collection*

The malware dataset is chosen and then worked upon.

B. *Data pre-processing*

It's an approach to generate the data applicable to a machine learning algorithm. It involves the following steps:

- Importing libraries.
- Reading the csv files.
- **Choosing the target variable** We will be selecting our target variable which is "legitimate," which has two values : 0 and 1. 1 represents original data and 0 represents malware.

  Fig. 3 shows the original and malware data in the used dataset for the target variable Legitimate.
- **Feature selection** It is a method to determine the most relevant and useful features and eliminate the irrelevant features from the dataset. In our dataset, since the Column "Name" only signifies the name of the file and Column

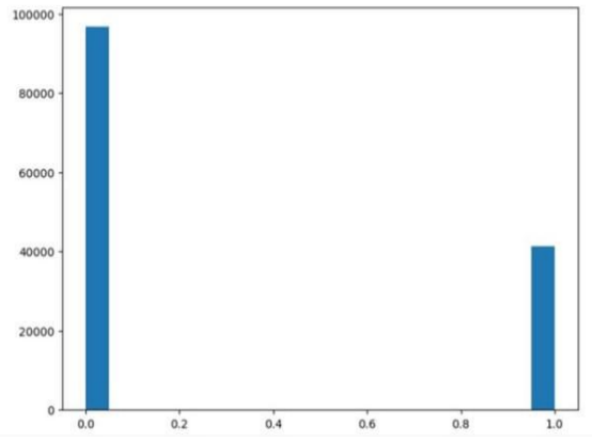

Fig. 3. Input variables 0 and 1 indicate the number of malware and original data respectively for the target variable 'legitimate'

"md" is a hash function so these values won't affect the training model and therefore we can drop them as follows. Df = df.drop(['Name', 'md5', axis=1)

- **Splitting the data into Test and Train** There are two sets of data in our dataset: a training dataset and attest dataset. Our machine learning model gives the best performance as a result of this step, a crucial part of pre-processing. The test dataset consisted of 20% of the dataset, whereas the training dataset consisted of 80% of the dataset.

C. Model building

The character selector outputs choose characters that are then used as input for modules. The modules apply detection approaches or classifiers to divide malicious and benign applications. The classifiers are enforced only after obtaining the relevant character, and each classifier is trained and tested using these character selections to operate the classification tasks. To detect unfamiliar malware in our dataset, we have appointed for machine learning classification algorithm, namely the random forest algorithm and the KNN algorithm.

1) K-nearest neighbour algorithm: The k-Nearest Neighbours (k-NN) algorithm is a popular and simple classification algorithm used in machine learning. It is a non-parametric algorithm that makes predictions based on the similarity between new input data and its neighbours in the training dataset. The algorithm operates on the principle that similar data points tend to belong to the same class or have similar properties. The k-NN algorithm starts by loading the training dataset, which consists of input feature vectors and their corresponding class labels. When given a new, unlabeled input data point, the algorithm calculates the distance between the input data point and all the training samples using a distance metric. The k-NN algorithm does not involve a training phase as it does not learn explicit models. Instead, it uses the entire training dataset for prediction. It requires calculating the distance for each test data point against all training samples.

2) Random forest: Random forest is generally used ML for supervised learning tasks, including classification and regression problems. To solve complex problems and enhance the performance of models, this method combines multiple classifiers using ensemble learning. Random forest concludes by building multiple decision trees trained on different subsets of the dataset and then taking the average of their predictions to make the final prediction. Through this, the Random forest goal is to reduce overfitting and increase the generalization of its models in order to improve prediction accuracy. As an alternative to answering on a single decision tree, Random forest creates prediction based on the majority of decision trees in the ensemble. The better can be achieved by using this rather than using a single decision tree since it captures more complex relationships and is less prone to outliers. Random forest models consist of the following steps:

- Take random samples from a set of data or training.
- Based on the decision tree, the vote will be averaged.
- Completely, choose the prediction result that obtains the max votes as the result latest prediction result.

3) Neural network: The neural network also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a type of machine learning algorithm commonly used in deep learning. They are composed of multiple layers of nodes arranged in input layers, one or more hidden layers, and an output layer. During the training phase, neural networks adjust their activation and threshold values to improve their performance on a specific task. If the output of the node surpasses its threshold value, it becomes activated and transmits data to the next layer in the network.

D. Evaluation criteria

We will be evaluating our model on the basis of the Confusion matrix for the attacker sample of 5%, 10% and 15% malware and find out the following values for each sample: Confusion matrix: In ML, the confusion matrix is a table that is used to calculate the accuracy of a classification model on an appropriate dataset. It helps to determine the kind of predictions built by the model by measuring them to the true value of the test data. To create a confusion matrix, we must have determined the actual or true values of the test data.

- True Negative: This value means that the model has concluded the value no and the original value is also no.
- True Positive: This value means that the model has concluded the value is yes and the original value was also yes.
- False Negative: This value means that the model has the conclusion the value is no but its original value is yes and it is also called a Type-II error.
- False Positive: This value means that the model has a conclusion the value is yes but its original value is no and it is also called a Type-I error.
- TPR (True Positives Rate) = number of True Positives/total number of positive $TPR = TP/(TP + FN)$.
- FPR (False Positives Rate) = number of False Positives/ total number of negatives $FPR = FP/(FP + TN)$.
- FNR (False Negatives Rate) = number of False Negatives/ total number of positives $FNR = FN/(TP + FN)$.

TABLE II
ACCURACY VALUES OF DIFFERENT MODELS IN THE PROPOSED PROTOCOL

S. No.	Model	Accuracy
1.	K nearest neighbor	98.82%
2.	Random forest	99.60%
3.	Neural networks	94.05%

TABLE III
COMPARISON OF ACCURACY VALUES THE PROPOSED PROTOCOL AND OTHER EXISTING PROTOCOLS

S.No.	Protocol	Accuracy
1.	Proposed protocol	99.60%
3.	Darem *et al.* [9]	99.37%
4.	Sewek *et al.* [10]	98.00%
5.	Krithika and Vijaya [11]	93.00%
6.	Saxe and Berlin [12]	95.00%

- Accuracy calculation $F1_{score} = 2 \times ((recall \times precision)/(recall + precision))$.

IV. PRACTICAL IMPLEMENTATION

In this section, we provide the details of the practical implementation of the proposed protocol. The details are given below.

A. Simulation settings

For the practical implementation of the proposed scheme, we have used Windows 11 operating system along with Intel(R) Core(TM) i5-8250U processor. The size of random access memory(RAM) was 8GB and Intel(R) UHD Graphics 620 was the graphics processor. Jupyter Notebook was the programming platform. Python was used for the programming part. We have used the "Mastering Machine for Penetration Testing: Malware Data" dataset [13]. The KNN, neural networks and random forest algorithms were used for the detection and analysis of the malware attacks. In the used dataset, there are 138047 rows and 57 columns.

B. Obtained results

The accuracy values of different machine learning models are given in Table II. These accuracy values were obtained using the 20% of the attacker sample. According to the above observations, the random forest algorithm is giving the best accuracy (i.e., 99.60%) out of all the algorithms.

C. Comparison of the proposed protocol with other existing protocols

The comparison of accuracy values of the proposed protocol and other existing protocols is given in Table III. The protocols of Darem *et al.* [9], Sewek *et al.* [10], Krithika and Vijaya [11], and Saxe and Berlin [12] achieved accuracy values of 99.37%, 98.00%, 93.00%, and 95.00%, respectively. Whereas the proposed scheme achieved an accuracy value of 99.60%, which is better than the other existing protocols.

TABLE IV
SIMULATION PARAMETERS

Parameter	Description
Operating system	Window 11
Processor used	Intel(R) Core(TM) i5-8250U
Random access memory (RAM)	8GB
Graphic Processor	Intel(R) UHD Graphics 620
Programming platform	Python, Jupyter notebook
Dataset Used	Mastering Machine for Penetration Testing: Malware Data [13]
Used algorithms	KNN, Neural Networks, Random Forest

V. CONCLUSION

A robust approach for the detection and analysis of malware attacks in the communications of the IoT was presented. In the proposed protocol various machine learning-based models, i.e., k-nearest neighbours (KNN), random forest and neural networks are used. It provided its best accuracy in the case of a random forest algorithm (i.e., 99.60%). The proposed protocol is also compared with the other similar existing protocols and it has achieved better accuracy as compared to the other existing protocols.

In future, we would like, to further improve the accuracy of the proposed protocol. Moreover, we want to add more functionality features to the proposed protocol.

REFERENCES

[1] S. Ali, O. Abusabha, F. Ali, M. Imran, and T. ABUHMED, "Effective Multitask Deep Learning for IoT Malware Detection and Identification Using Behavioral Traffic Analysis," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022, doi: 10.1109/TNSM.2022.3200741.

[2] D. K. A., V. P., S. Y. Yerima, A. Bashar, A. David, A. T., A. Antony, A. K. Shavanas, and G. K. T., "Obfuscated Malware Detection in IoT Android Applications Using Markov Images and CNN," *IEEE Systems Journal*, pp. 1–11, 2023, doi:10.1109/JSYST.2023.3238678.

[3] W. Niu, Y. Wang, X. Liu, R. Yan, X. Li, and X. Zhang, "GCDroid: Android Malware Detection Based on Graph Compression with Reachability Relationship Extraction for IoT Devices," *IEEE Internet of Things Journal*, pp. 1–1, 2023, doi:10.1109/JIOT.2023.3241697.

[4] M. E. Zadeh Nojoo Kambar, A. Esmaeilzadeh, Y. Kim, and K. Taghva, "A survey on mobile malware detection methods using machine learning," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 2022, pp. 0215–0221.

[5] S. Sharma, C. Rama Krishna, and S. K. Sahay, "Detection of advanced malware by machine learning techniques," in *Soft Computing: Theories and Applications: Proceedings of SoCTA 2017*. Springer, 2019, pp. 333–342.

[6] M. V. R. Kumar, A. Kumar, A. Bando, S. R. Gs, H. Shah, and S. C. Reddy, "A survey of deep learning techniques for malware analysis," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 6031–6042, 2020.

[7] A. Muzaffar, H. R. Hassen, M. A. Lones, and H. Zantout, "An in-depth review of machine learning based android malware detection," *Computers & Security*, p. 102833, 2022.

[8] B. Urooj, M. A. Shah, C. Maple, M. K. Abbasi, and S. Riasat, "Malware detection: A framework for reverse engineered android applications through machine learning algorithms," *IEEE Access*, vol. 10, pp. 89 031–89 050, 2022.

[9] A. A. Darem, F. A. Ghaleb, A. A. Al-Hashmi, J. H. Abawajy, S. M. Alanazi, and A. Y. Al-Rezami, "An Adaptive Behavioral-Based Incremental Batch Learning Malware Variants Detection Model Using Concept Drift Detection and Sequential Deep Learning," *IEEE Access*, vol. 9, pp. 97 180–97 196, 2021.

[10] M. Sewak, S. K. Sahay, and H. Rathore, "An Investigation of a Deep Learning Based Malware Detection System," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*. Hamburg, Germany: Association for Computing Machinery, 2018.

[11] V. Krithika and M. S. Vijaya, "Malware and Benign Detection Using Convolutional Neural Network," in *Data Engineering and Intelligent Computing*, V. Bhateja, S. C. Satapathy, C. M. Travieso-González, and V. N. M. Aradhya, Eds. Singapore: Springer Singapore, 2021, pp. 37–45.

[12] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, Fajardo, PR, USA, 2015, pp. 11–20.

[13] Chiheb Chebbi, "Mastering-Machine-Learning-for-Penetration-Testing:MalwareData," January 2022, https://github.com/PacktPublishing/Mastering-Machine-Learning-for-Penetration-Testing/blob/master/Chapter03/MalwareData.csv.gz. Accessed on May 2023.