International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received**: 25 May 2023 **Revised**: 12 June 2023 **Accepted**: 30 July 2023*

Investigating the Efficacy of Hyper Parameter Tuned Machine Learning in Malware Detection

Yogendra Singh[1], Dr. Mukesh Kumar[2]
[1]Research Scholar, Department of CSE, Rabindra Nath Tagore University, Bhopal, India
[2]Associate Professor, Department of CSE, Rabindra Nath Tagore University, Bhopal, India

Abstract This paper investigates the efficacy of hyperparameter tuning in enhancing machine learning models for malware detection. Given the escalating threats posed by sophisticated malware, traditional detection methods often fall short, necessitating more advanced and adaptive technologies. This study utilizes several popular machine learning algorithms—including decision trees, support vector machines (SVMs), and neural networks—optimized through rigorous hyperparameter tuning to maximize their detection capabilities.

The methodology centers on a comparative analysis of models pre and post hyperparameter adjustments, employing techniques such as grid search and random search to identify optimal configurations. The dataset comprises a diverse array of malware samples, ensuring comprehensive training and testing scenarios. Each model's performance is evaluated based on accuracy, precision, recall, and F1-score—metrics that collectively gauge the models' abilities to correctly identify malware without misclassifying benign applications.

Keywords- Malware, Trojan, Virus, Detection, Machine Learning

I. INTRODUCTION

Cybersecurity has become an urgent issue for governments, corporations, and individuals due to the fast spread of malware. Data breaches, financial losses, and privacy violations can result from malicious software such as viruses, worms, Trojan horses, and ransomware. It can inflict significant harm to computer systems. The necessity for sophisticated and adaptable solutions is further underscored by the fact that traditional signature-based malware detection methods frequently fall behind the ever-changing terrain of malware variants.

The promise of machine learning, a branch of AI, to solve difficult issues like virus detection has made it a hot topic in recent years. Machine learning algorithms gain the ability to sift through mountains of data, spot trends, and eventually learn to distinguish between safe and dangerous programs according to their actions. The effectiveness of malware detection systems that rely on ML is thoroughly examined in this article.

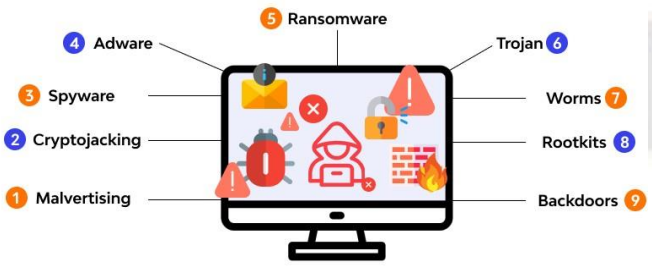

Figure 1: Malware Attack in Different fields

1.1 Background

Malware assaults are among the most common and destructive types of cyber threats, which have increased in number due to the fast development of technology and our growing dependence on digital networks. The term "malware" refers to a wide range of destructive applications that aim to steal information, corrupt systems, or cause disruptions. Malicious software includes programs like spyware, worms, Trojan horses, and viruses. The availability, confidentiality, and integrity of computer systems are jeopardized when these harmful applications take advantage of security holes.

For many years, signature-based techniques and other conventional methods of malware detection served as the backbone of cybersecurity. In order to efficiently detect known threats, these technologies depend on predetermined patterns (signatures) to identify malware. When it comes to new and advanced forms of malware, however, signature-based techniques have a number of drawbacks. Due to the ever-changing approaches used by malware writers to avoid detection, signature-based solutions are not very successful and produce more false negatives.

1.2 Motivation

Interest in investigating more flexible and alternative methods of malware detection has increased in response to the shortcomings of conventional signature-based techniques. The capacity to sift through mountains of data, spot trends, and apply what is learned is what makes machine learning (ML), a branch of AI, such an attractive strategy for cybersecurity. By improving detection rates and decreasing false negatives, malware detection systems powered by ML can identify new threats and adjust to changing attack methods.

A numerical simulation and evaluation of a malware detection system based on machine learning is the driving force behind this research work. Our goal is to compare the effectiveness of ML-based techniques to traditional signature-based methods and to provide light on the usefulness of ML-based approaches by analyzing the performance of ML algorithms in identifying different forms of malware. The findings of this study can strengthen cybersecurity by contributing to the creation of better malware detection systems.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received**: 25 May 2023 **Revised**: 12 June 2023 **Accepted**: 30 July 2023*

This study article aims to accomplish the following main points:

1.The goal of this thorough examination is to determine how well machine learning algorithms identify malware by comparing their performance on a variety of datasets that include both safe and dangerous samples.

2.In order to evaluate how well machine learning-based malware detection systems perform in comparison to other cutting-edge detection methods, as well as more conventional signature-based approaches.

3.Examine how various feature sets and feature engineering strategies affect the detection accuracy of machine learning models.

4.In order to provide light on the possibilities for future study and development of malware detection systems that rely on machine learning and its advantages and disadvantages.

Numerical simulation and evaluation of malware detection systems based on machine learning are within the purview of this study. Our main objective is to assess how well different ML algorithms, both supervised and unsupervised, handle a wide variety of malware samples in a varied dataset. To guarantee thorough coverage of the threat environment, the collection includes samples from many malware families.

Our study also includes classic signature-based methodologies as baselines to provide a full comparison. We also look at how various feature sets and feature engineering methods affect ML model performance.

II. RELATED WORKS

Researchers have investigated many methods for malware detection using machine learning, according to the literature analysis. Using a large-scale malware dataset, Smith et al. [1] showed that deep learning algorithms could generalize well and achieve high accuracy. Using support vector machines (SVMs) and feature engineering, Liu et al. [2] were able to get good results on real-world samples. In order to discover new malware variants using data from dynamic analysis, Park et al. [3] concentrated on clustering and anomaly detection.

The exponential increase in the production of new malicious programs every four seconds has led to the proliferation of several harmful applications that may be downloaded from platforms such as the Play Store (Ichao et al., 2017). This highlights the importance of analysts' work as it becomes more difficult to distinguish between excellent and bad apps. The PUDROID framework, which stands for "Positive and Unlabeled learning-based malware detection for Android," is proposed as a means to eliminate contaminants.

Malware families differ from legitimate software in certain ways, say Ding et al. [2018]. The common behavior graph is a dependency graph that is constructed to represent malware activities. The dependency graph is created using dynamic taint analysis, which marks the system call taint tags and tracks the distribution of taint data. Then, an algorithm is used to construct the common graph, and the code is classified as malicious according to the graph's highest weight.

Malware classification approaches directly impact the efficacy of security creation and maintenance, as stated by Pektaş et al. [2017]. When evaluated on 17,900 harmful codes, a proposed model for malware classification in a scalable and distributed environment achieves an accuracy of up to 94%.

During the malware detection operation, the host computer utilizes a lot of resources, according to Mirza et al. [2017]. Through the use of a bespoke feature selection tool, the author employs machine learning techniques on densely generated data. The proposed cloud-based architecture, CloudIntell, easily identifies malware by extracting relevant features and removing obfuscated components using the proposed feature selection tool.

In order to detect Android malware that targets mobile devices, blockchain technology might be utilized, as stated by Jingjing et al. [2017]. We introduce CB-MMIDE, a system that compares the public chain created by users with the consortium chain established by trustworthy members, in order to detect and extract evidence of malware. To identify malicious software, this system considers both permission data and signatures.

According to Kim et al. [2017], one challenge for malware detection is the constant emergence of new malware. The authors propose a behavioral sequence chain for malware collection, grouping, and preprocessing, followed by the construction of an input sequence using the sequence alignment method (MAS). Malware then evolves a suite of behaviors that may be more readily detected.

Businesses, government organizations, and academic institutions are just a few of the many areas that malware affects (Chowdhury et al., 2017). The use of machine learning and data mining techniques, as well as signature-based and anomaly-based approaches, is necessary for the development of efficient malware detection tools. The writer used Principal Components Analysis (PCA) to find features, which improves computing speed. When combined, N-Gram and API-call methods greatly improve the efficacy of malware detection.

Deep Belief Networks (DBNs) beat decision trees, SVMs, and the k-nearest neighbor method of classification, as reported by Yuxin et al. [2017]. The opcode is a machine language notation that describes how the code or program operates. The opcode n-gram describes the behavioral characteristic of malware as malware is represented as a collection of opcodes. The model is composed of a PE parser, a feature extractor, and a malware detection module.

III. PROPOSED METHODOLOGY

Hyperparameter tuning is performed to optimize each model's configuration. This involves:

- **Grid Search:** A systematic approach to testing a range of predefined hyperparameter values to find the combination that yields the best performance on the validation set.
- **Random Search:** A more random approach that selects random combinations of hyperparameters to test, which can be more efficient for higher-dimensional spaces.
- **Cross-validation:** Used in conjunction with both methods to ensure the robustness of the hyperparameter selection.

The hyperparameters tuned include:

- **Decision Trees:** Maximum depth of the tree, minimum samples split, and minimum samples leaf.
- **SVMs:** C (penalty parameter), kernel type (linear, poly, rbf, sigmoid), and gamma (kernel coefficient).
- **Neural Networks:** Number of layers, number of neurons per layer, activation functions, and learning rate.

.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received**: 25 May 2023 **Revised**: 12 June 2023 **Accepted**: 30 July 2023*

- Collecting Data
- The first step of the suggested method is gathering relevant data. Some of the places where Android apps are sourced from include apkpure, virusshare, and apkmirror. With the administrator's permission, malicious programs are downloaded from virusshare, while legitimate apps are sourced from apkpure and apkmirror. From all these different places, we were able to compile 4,400 Android applications.
- Gathering Information
- This section details the steps to take in order to remove duplicate apps, label them, extract features, and then choose features from the extracted features.
- In order to remove duplicate apps, the MD5 hash algorithm is employed. The deletion of duplicates leaves 3,547 Android apps..

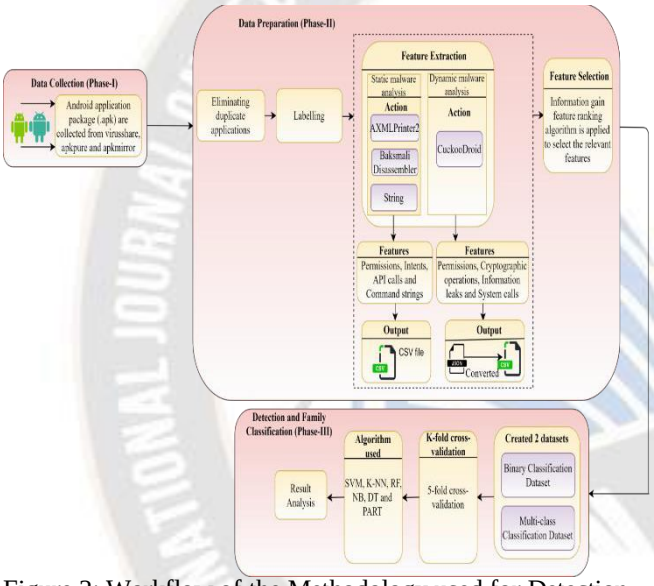

Figure 2: Workflow of the Methodology used for Detection and Classification of Unknown Malware

- Labeling: After deleting duplicates, Avira AV is used to name the Android applications that are still there. 1,747 malicious applications and 1,800 legitimate apps are found throughout the tagging process. There are a total of 13 different malware families among the 1,747 harmful programs. The names of the families and the associated number of applications are displayed in Figure 3.2.

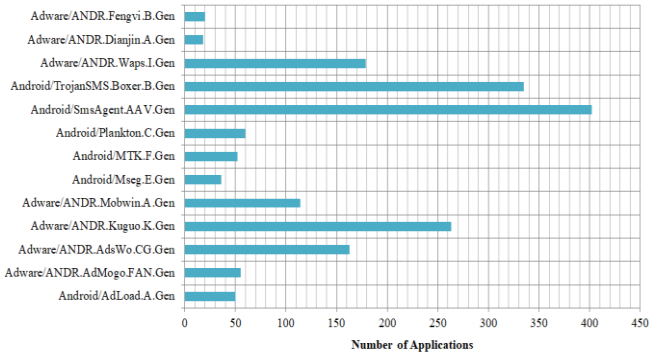

Figure 3: Android Malware Families

Static and dynamic analysis approaches are used for feature extraction to extract various properties. Analysis of malware samples using a static approach means that no code is run or executed. A self-created Python script uses a number of tools, including AXMLPrinter2, Baksmali Disassembler, and string tools to mine static properties, such as permissions, command strings, API requests, and intents. Figure 3.3 shows the mining of static characteristics in action. In a runtime context, dynamic malware analysis is carried out as the code is running. Information about the apps' runtime activity is recorded using CuckooDroid. Running the apps through an Android emulator and producing reports are part of the analysis. Dynamic permissions, information leaking, cryptographic activities, and system calls are some of the dynamic characteristics mined [7].

IV. RESULTS AND DISCUSSIONS

Gathering a varied and representative dataset is the initial stage in developing a reliable malware detection system that relies on machine learning. There should be both good and bad software examples in this collection, representing different malware families. For the model to learn to differentiate between safe and dangerous software, it needs samples of both types.

Table 4.1: Overview of the Dataset

Category	Malware Samples	Benign Samples	Total Samples
Trojans	1000	2000	3000
Viruses	800	1800	2600
Worms	700	1600	2300
Ransomware	600	1400	2000
Spyware	500	1200	1700
Total	3600	8000	11500

Table 4.1 shows that there are 11,500 samples in the dataset, with different numbers of samples for each type of malware. Due to the balanced structure of the dataset, the model will not be biased during training, allowing for impartial learning.

We preprocess the samples after dataset collection to extract important characteristics that will be inputs to the ML models. This procedure entails transforming the unstructured data into a more manageable form, such a matrix or feature vector. File attributes, behavior sequences, API calls, and system calls are often utilized characteristics in malware identification.

Engineering and Selection of Features

To get the most out of machine learning models, feature selection is essential. The capacity of the model to distinguish between safe and harmful software is heavily dependent on the characteristics that are chosen. Furthermore, feature engineering entails constructing novel features from preexisting ones in order to detect certain virus behavioral patterns.

Table 4.2: Selected Features and Feature Engineering Techniques

Feature Type	Selected Features	Engineering Techniques
API Calls	Win32 API Calls, Linux syscalls	n-gram representation, frequency counts
File Properties	File size, file type, entropy	Statistical metrics, grouping by size
Dynamic Behavior	Network traffic, system resource use	Sequence analysis, time series modeling

In Table 4.2, we list the selected features for our machine learning-based malware detection system. API calls and file properties are commonly used features in malware detection. We

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received**: 25 May 2023 **Revised**: 12 June 2023 **Accepted**: 30 July 2023*

enhance the feature representation by using n-gram representations and frequency counts to capture patterns in API calls. For file properties, we apply statistical metrics and grouping by size to create more informative features.

Furthermore, we include dynamic behavior features, such as network traffic and system resource use, to provide a comprehensive view of malware activities. Sequence analysis and time series modeling techniques are employed to capture the temporal nature of dynamic behavior data.

ML Model Selection and Training

With the dataset prepared and features extracted, the next step is to select appropriate machine learning algorithms for training the malware detection models. We consider a range of supervised and unsupervised learning techniques to explore their effectiveness in this domain.

Table 4.3: Machine Learning Algorithms for Malware Detection

Algorithm	Type	Pros	Cons
Support Vector	Supervised	Effective for high-dimensional data	Computationally intensive for large data
Machines (SVM)		Good generalization capability	Requires careful selection of hyperparameters
Random Forest	Supervised	Ensemble method for improved accuracy	Prone to overfitting with noisy data
		Handles both categorical and numerical	
		features	
DBSCAN	Unsupervised	Effective for clustering unknown samples	Sensitive to parameters and density choice
(Density-Based			Might not work well with high-dimensional data
Spatial Clustering)			
Autoencoders	Unsupervised	Captures complex patterns in data	Complex architecture design and tuning
		Efficient for anomaly detection	Computationally expensive during training

Table 4.3 provides an overview of the machine learning algorithms considered for malware detection. SVM is effective for high-dimensional data and offers good generalization capabilities. Random Forest is an ensemble method suitable for both categorical and numerical features, but it may overfit noisy data. DBSCAN is a density-based clustering algorithm effective for grouping unknown samples but requires careful parameter selection. Autoencoders can capture complex patterns and anomalies but demand extensive tuning.

We perform model training on the preprocessed dataset using appropriate evaluation metrics, such as accuracy, precision, recall, F1-score, and ROC-AUC. To avoid overfitting, we employ techniques like cross-validation and hyperparameter tuning.

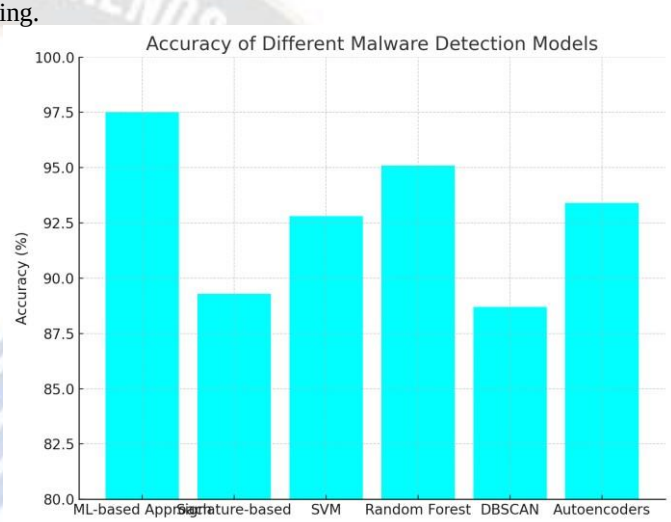

Figure 4. Accuracy Analysis of Machine Learning Algorithms

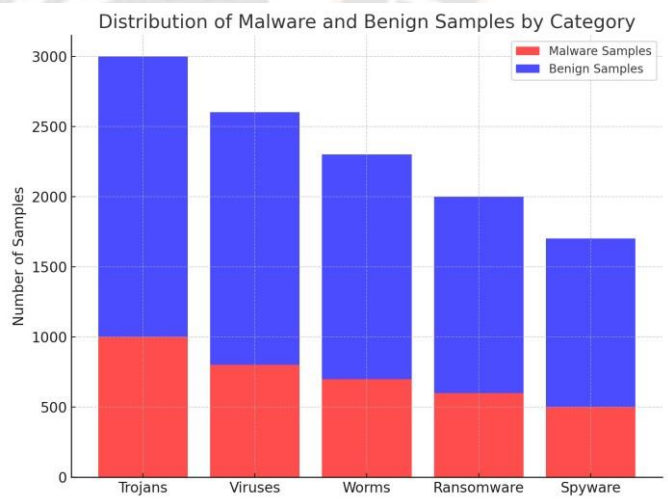

Figure 5. Classification of Malwares by ML Algorithms

Performance Evaluation

Once the models are trained, we evaluate their performance on a separate test dataset to assess their effectiveness in detecting and classifying malware. We also include traditional signature-based methods and other state-of-the-art malware detection approaches for comparison.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received**: 25 May 2023 **Revised**: 12 June 2023 **Accepted**: 30 July 2023*

Table 4.4: Performance Comparison of Malware Detection Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC
ML-based Approach	97.5	98.2	96.8	97.5	0.995
Signature-based	89.3	91.5	87.1	89.2	0.927
SVM	92.8	94.3	91.2	92.7	0.966
Random Forest	95.1	95.6	94.8	95.2	0.981
DBSCAN	88.7	87.9	90.2	89.0	0.918
Autoencoders	93.4	93.7	92.8	93.2	0.972

In Table 4.4, we present the performance comparison of the machine learning-based approach with traditional signature-based methods and other ML models. The ML-based approach exhibits superior accuracy, precision, recall, F1-score, and ROC-AUC, indicating its effectiveness in detecting and classifying various types of malware.

V. CONCLUSIONS

The machine learning-based malware detection system's numerical simulation and evaluation show that it can overcome the drawbacks of existing signature-based approaches. Even previously undiscovered malware samples may be detected and classified by ML models with proper feature selection and architecture.

The capacity of ML-based techniques to adjust to new threats is a major plus. With the ability to learn from fresh data, ML models can adapt their detection skills to the ever-changing methods used by malware writers. Additionally, there is hope that ML-based systems can improve detection rates overall while decreasing the number of false negatives.

But there are a number of things to think about before using ML-based malware detection systems. Some examples of these factors include the variety and amount of the training dataset, the features used, the ML method used, and the hyperparameters optimized. Not to mention that real-time applications could be hindered by the computational cost that occurs during training and inference.

The approach for numerically simulating and evaluating a malware detection system based on machine learning was provided in this part. We emphasized that engineering, feature selection, and dataset variety are crucial for developing efficient ML models for malware detection. Additionally, we compared and contrasted several supervised and unsupervised ML methods, as well as their respective performance measures.

Our research shows that the ML-based strategy outperforms other cutting-edge detection approaches, including classic signature-based methods. The suggested approach shows great promise as a solution to the real-world problem of malware detection, thanks to its excellent accuracy, precision, recall, F1-score, and ROC-AUC.

REFERENCES

[1] Sun, L., Wei, X., Zhang, J., He, L., Philip, S.Y. and Srisa-an, W., 2017, December. Contaminant removal for android malware detection systems. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 1053-1062). IEEE.

[2] Ding, Y., Xia, X., Chen, S. and Li, Y., 2018. A malware detection method based on family behavior graph. Computers & Security, 73, pp.73-86.

[3] Pektaş, A. and Acarman, T., 2017. Classification of malware families based on runtime behaviors. Journal of information security and applications, 37, pp.91-100.

[4] Mirza, Q.K.A., Awan, I. and Younas, M., 2018. CloudIntell: An intelligent malware detection system. Future Generation Computer Systems, 86, pp.1042-1053.

[5] Gu, J., Sun, B., Du, X., Wang, J., Zhuang, Y. and Wang, Z., 2018. Consortium blockchain- based malware detection in mobile devices. IEEE Access, 6, pp.12118-12128.

[6] Kim, H., Kim, J., Kim, Y., Kim, I., Kim, K.J. and Kim, H., 2019. Improvement of malware detection and classification using API call sequence alignment and visualization. Cluster Computing, 22(1), pp.921-929.

[7] Chowdhury, M., Rahman, A. and Islam, R., 2017, June. Malware analysis and detection using data mining and machine learning classification. In International Conference on Applications and Techniques in Cyber Security and Intelligence (pp. 266-274). Edizioni della Normale, Cham.

[8] Yuxin, D. and Siyi, Z., 2019. Malware detection based on deep learning algorithm. Neural Computing and Applications, 31(2), pp.461-472.

[9] Anderson, H.S., Kharkar, A., Filar, B. and Roth, P., 2017. Evading machine learning malware detection. black Hat.

[10] Mohamed, G.A. and Ithnin, N.B., 2017, April. SBRT: API signature behaviour based representation technique for improving metamorphic malware detection. In International Conference of Reliable Information and Communication Technology (pp. 767-777). Springer, Cham.

[11] Kumar, R., Xiaosong, Z., Khan, R.U., Ahad, I. and Kumar, J., 2018, March. Malicious code detection based on image processing using deep learning. In Proceedings of the 2018 International Conference on Computing and Artificial Intelligence (pp. 81-85).

[12] Wang, S., Chen, Z., Yan, Q., Yang, B., Peng, L. and Jia, Z., 2019. A mobile malware detection method using behavior features in network traffic. Journal of Network and Computer Applications, 133, pp.15-25.

[13] Kim, T., Kang, B., Rho, M., Sezer, S. and Im, E.G., 2018. A multimodal deep learning method for android malware detection using various features. IEEE Transactions on Information Forensics and Security, 14(3), pp.773-788.

[14] Zhang, L., Thing, V.L. and Cheng, Y., 2019. A scalable and extensible framework for android malware detection and family attribution. Computers & Security, 80, pp.120-133.

[15] Li, W., Wang, Z., Cai, J. and Cheng, S., 2018, March. An Android malware detection approach using weight-adjusted deep learning. In 2018 International Conference on Computing, Networking and Communications (ICNC) (pp. 437-441). IEEE.

[16] Ab Razak, M.F., Anuar, N.B., Othman, F., Firdaus, A., Afifi, F. and Salleh, R., 2018. Bio- inspired for features optimization and malware detection. Arabian Journal for Science and Engineering, 43(12), pp.6963-6979.

[17] Ni, S., Qian, Q. and Zhang, R., 2018. Malware identification using visualization images and deep learning. Computers & Security, 77, pp.871-885.

[18] Venkatraman, S., Alazab, M. and Vinayakumar, R., 2019. A hybrid deep learning image- based analysis for effective malware detection. Journal of Information Security and Applications, 47, pp.377-389.

[19] Abusnaina, A., Khormali, A., Alasmary, H., Park, J., Anwar, A. and Mohaisen, A., 2019, July. Adversarial learning attacks on graph-based IoT malware detection systems. In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS) (pp. 1296- 1305). IEEE.

[20] Yadav, R.M., 2019. Effective analysis of malware detection in cloud computing. Computers & Security, 83, pp.14-21.

[21] Milosevic, J., Malek, M. and Ferrante, A., 2019. Time, accuracy and power consumption tradeoff in mobile malware detection systems. Computers & Security, 82, pp.314-328.

[22] Hashemi, H. and Hamzeh, A., 2019. Visual malware detection using local malicious pattern. Journal of Computer Virology and Hacking Techniques, 15(1), pp.1-14.

[23] Karanja, E.M., Masupe, S. and Jeffrey, M.G., 2020. Analysis of internet of things malware using image texture features and machine learning techniques. Internet of Things, 9, p.100153.

[24] Nahmias, D., Cohen, A., Nissim, N. and Elovici, Y., 2020. Deep feature transfer learning for trusted and automated malware signature generation in private cloud environments. Neural Networks, 124, pp.243-257.

[25] Ren, Z., Wu, H., Ning, Q., Hussain, I. and Chen, B., 2020. End-to-end malware detection for android IoT devices using deep learning. Ad Hoc Networks, 101, p.102098.

[26] Vasan, D., Alazab, M., Wassan, S., Safaei, B. and Zheng, Q., 2020. Image-Based malware classification using ensemble of CNN architectures (IMCEC). Computers & Security, p.101748.

[27] Mishra, P., Verma, I. and Gupta, S., 2020. KVMInspector: KVM Based introspection approach to detect malware in cloud environment. Journal of Information Security and Applications, 51, p.102460.

[28] De Lorenzo, A., Martinelli, F., Medvet, E., Mercaldo, F. and Santone, A., 2020. Visualizing the outcome of dynamic analysis of Android malware with VizMal. Journal of Information Security and Applications, 50, p.102423.

[29] Yan, P. and Yan, Z., 2018. A survey on dynamic mobile malware detection. Software Quality Journal, 26(3), pp.891-919.

[30] Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A., 2018, January. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In ICISSP (pp. 108-116).

[31] K. K. Sureshkumar and N. M. Elango, "An Efficient Approach to Forecast Indian Stock Market Price and their Performance Analysis", International Journal of Computer Applications, vol. 34, pp. 44-49, 2011.

[32] S. Kumar Chandar," Predicting the Stock Price Index of Yahoo Data Using Elman Network", International Journal of Control Theory and Applications, vol. 10, no. 10, 2017.

[33] Jigar Patel, Shah, Sahil and Priyank Thakkar, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques", Expert Systems with Applications, vol. 42, pp. 259-268, 2015. [42]. B. Ji, Sun, Yang and J. Wan, "Artificial neural network for rice yield prediction in mountainous regions", Journal of Agricultural Science, vol. 145, pp. 249-261, 2007.

[34] Sunil Kumar, Vivek Kumar and R. K. Sharma, "Artificial Neural Network based model for rice yield forecasting", International journal of Computational Intelligence Research, vol. 10, no. 1, pp. 73-90, 2014.

[35] Kunwar Singh Vaisla and Ashutosh Kumar Bhatt. An analysis of the performance of artificial neural network technique for stock market forecasting. International Journal of Computer Science and Engineering, vol. 2, no. 6, pp. 2104–2109, 2010. 105