IJARASEM

ISSN: 2395-7852

International Journal of Advanced Research in Arts, Science, Engineering & Management

Volume 10, Issue 2, March 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 6.551

International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM)

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 6.551 |Bimonthly, Peer Reviewed & Referred Journal|

| Volume 10, Issue 2, March 2023 |

Machine Learning Methods for Malware Detection and Classification

V Aileen Emelda, Navya K S

Assistant Professor, Department of Computer Science and Engineering, MVJ College of Engineering, Bangalore, India

UG Scholar, Department of Computer Science and Engineering, MVJ College of Engineering, Bangalore, India

ABSTRACT: The main challenge for malware researchers is the large amount of data and files that need to be evaluated for potential threats. Researchers analyses many new malwares daily and classify them in order to extract common features. Therefore, a system that can ensure and improve the efficiency and accuracy of the classification is of great significance for the study of malware characteristics. A high-performance, high-efficiency automatic classification system based on multi-feature selection fusion of machine learning is proposed in this project. Its performance and efficiency, according to our experiments, have been greatly improved compared to single-featured systems. The increase of malware that are exploiting the Internet daily has become a serious threat. The manual heuristic inspection of malware analysis is no longer considered effective and efficient compared against the high spreading rate of malware. Hence, automated behaviour-based malware detection using machine learning techniques is considered a profound solution. The behaviour of each malware on an emulated (sandbox) environment will be automatically analysed and will generate behavior reports. These reports will be pre-processed into sparse vector models for further machine learning (classification). The classifiers used in this project are Random Forests, Support Vector Machine (SVM), Random Forest. In summary, it can be concluded that a proof of- concepts based on automatic behavior-based malware analysis and the use of machine learning techniques could detect and classify malwares quite effectively and efficiently. Since, many antiviruses have heavy impact on the user system and sometimes they are not able to detect new emerging harmful threats and on the other hand, light weight antiviruses are not so effective in detecting and preventing malwares. As the data security with no heavy impact on the user system is our main concern, our model is a cloud-based malware classification model. Therefore, without installing any third-party application on user system, we can test and predict whether any suspicious file is a malware or not.

KEYWORDS: Malware, Support Vector Machine (SVM), Random Forest.

I.INTRODUCTION

In recent years, continuous growing in malware technique has become a great threat to modern information technology and personal information security, which greatly inspires the need of anti- malware technique. many schools, personal enterprises and government infrastructures were hacked by a ransomware named WannaCry, while measures were made to minimize the loss by anti-malware solution providers. However, lots of variants were made in order to evade detection, and processes new large-scale infection with the help of new exploits. The emitting of WannaCry variants is just the real world in miniature, showing that malware is able to bypass normal protection mechanisms by current techniques. As an important factor of increasing number of malwares, modification and obfuscation are meant to avoid being detected by malware authors. According to McAfee's Q3 report, the amount of malware has increased about 10% compared with Q2. And researchers have to find a way to get rid of these new malwares by analyzing them for its families' classification. A malware family is a set of malware code that has similar intentions, regardless of their difference in code implementation. But modification and obfuscation have made it hard for researchers to analyses and classify samples with efficiency. So, it would be helpful to analyses samples with similar features, which can make their potential features easier to be found, as well as save researcher's time. Therefore, the classification of malware plays an important role in malware analysis. Manual methods with signature-based classification are often used to classify samples before the rapid development of machine learning, but obfuscation and polymorphic strategies influence the effectiveness of these techniques heavily.

The problem to be examined involves the high spreading rate of computer malware (viruses, worms, Trojan horses, rootkits, botnets, backdoors, and other malicious software) and conventional signature matching-based antivirus systems fail to detect polymorphic and new, previously unseen malicious executables. Malware are spreading all over the world through the Internet and are increasing day by day, thus becoming a serious threat.

The manual heuristic inspection of static malware analysis is no longer considered effective and efficient compared against the high spreading rate of malware. Nevertheless, researches are trying to develop various alternative approaches in combating and detecting malware. One proposed approach (solution) is by using automatic dynamic (behaviour) malware analysis combined with data mining tasks, such as, machine learning (classification) techniques to achieve effectiveness and efficiency in detecting malware.

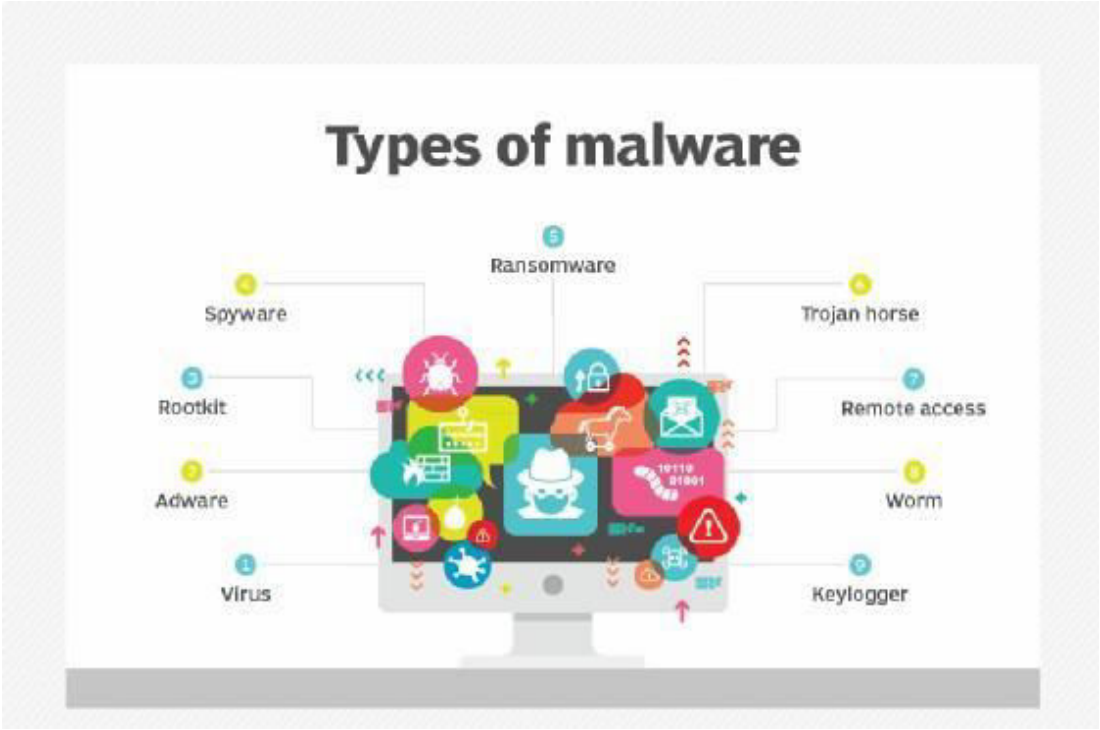

Figure 1:Types of Malwares

II.LITERATURE SURVEY

The file is executed and the information is collected about its properties like what actually it intends to do. What we can do is we can run our file by creating a virtual system such as Virtual box. While doing an analysis of this type, we can easily figure out all the behavioural based attributes example detecting a file, undo the file. We can define behaviour-based methods as both static and dynamic analysis. The Advantage of these dynamic methods is that we can make sure that what will happen when this type of malwares in an actual system.

Random Forest is an ensemble learning algorithm developed by Breiman. An ensemble learner method generates many individual learners and aggregates the results. Random Forest uses an extension to the bagging approach. In Bagging, each classifier is built individually by working with a bootstrap sample of the input data. In a regular decision tree classifier, a decision at a node split is made based on all the feature attributes. But in Random Forest, the best parameter at each node in a decision tree is made from a randomly selected number of features. This random selection of features helps Random Forest to not only scale well when there exists many features per feature vector, but also helps it in reducing the interdependence (correlation) between the feature attributes and is thus less vulnerable to inherent noise in the data.

As mentioned by the author, the number of random features m selected per decision node in a tree decides the error rate of the forest classification. The error rate of the Random Forest classifier depends on the correlation between any two trees, and the classification strength of each individual tree. Reducing the random features selected m causes reduction in both the correlation between classification trees and the strength of classification of each individual tree. Increasing m increases both the correlation between the trees and the strength of each tree. Breiman explains that the Out- of-Bag (OOB) error rate is an indication of how well a forest classifier performs on the dataset. The out-of-bag model leaves out one third of the input dataset for building the kth tree from the bootstrap sample for each tree. This

one-third sample is used to test the kth tree and the results of misclassification averaged over all trees. The author claims that for most cases OOB error estimate is a good estimate of the error and hence cross validation or separate test set is usually unnecessary when using the Random Forest algorithm.

SVM is a discriminative type of classification technique. The boundaries for each class are well defined by a separating hyperplane. In other words, it basically creates an optimal hyperplane that bounds the given training dataset and classifies the test sample based on the plane it falls under. Several types of kernel can be used for the final decision. In this research, we particularly focus on a Quadratic kernel (hence, QSVM).

The format of the file or metadata can be a good way to guess what actually the intended action is. For example, Portable executable files in Microsoft Windows is a good way to know about information like execute or compile time and other functionalities.

III.PROPOSED METHODOLOGY

A software requirements specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements, and can also consist of a use cases that describe interactions the users will have with the software. Software requirement specification establishes the ground work for an agreement between clients and contractors or suppliers (in market-driven projects, these roles may also be played by means of the marketing and improvement divisions) on what the software product has to perform as properly as what it is no longer anticipated to do. Software necessities specification permits a rigorous evaluation of requirements earlier than layout can commence and reduces later redesign; it additionally provides a practical groundwork for estimating product costs, risks, and schedules. The software necessities specification file enlists enough and quintessential necessities that are required for the assignment development. To derive the necessities, we need to have clear and thorough understanding of the products to be developed or being developed this is finished and sophisticated with targeted and continuous communications with the undertaking group and consumer until the completion of the software.

Non-functional requirements are constraints that must be adhered to during development. They limit what resources can be used and set bounds on aspects of the software's quality. One of the most important matters about non-functional requirements is to make them verifiable. The verification is generally finished by way of measuring a number of elements of the machine and seeing if the measurements confirm to the requirements.

Non-functional requirements contain different parameters such as:

Usability: The application is going to be used by the people of particular place and also the concerned authority. This is going to assist them in predicting disaster occurrence.

Efficiency: Our application takes less time to accomplish a particular task such as fetching current and past weather data which also reduces time complexity. It reduces the complications when an information has several functionalities thus increases the efficiency.

Reliability: The application that we are developing is designed to deliver set of services as expected by the user. The application provides many modules and each module is developed satisfy the non-functional requirements of the customers.

Maintainability: The application that we are developing is going to provide a high-performance measure such as the weather updates are done automatically without loss of data that already exists.

These requirements constrain the design to meet specified levels of quality. The second group of non-functional requirements categories constrains the environment and technology of the system.

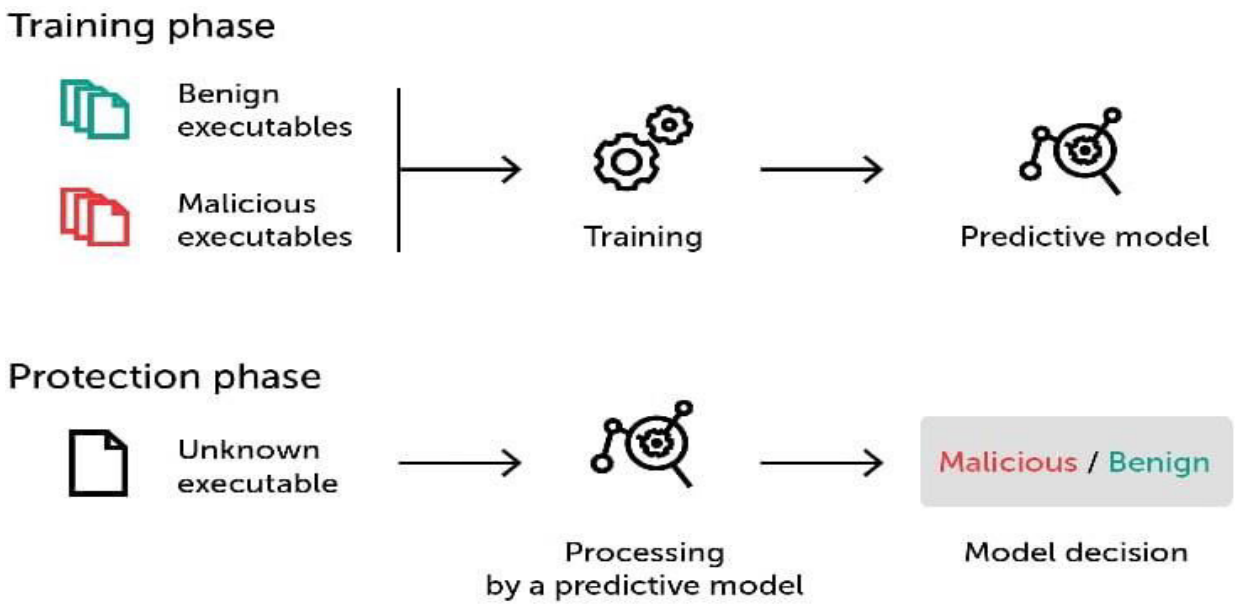

Figure 2 : Overall Architecture of the system

IV. RESULTS

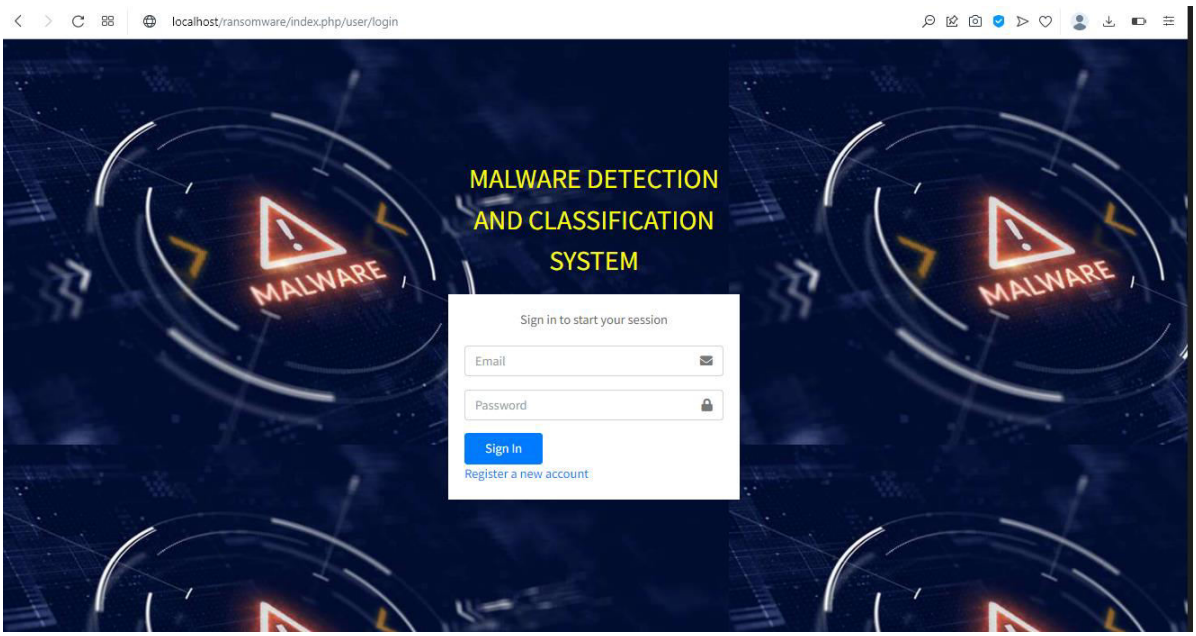

Figure 3: Results of Malware detection

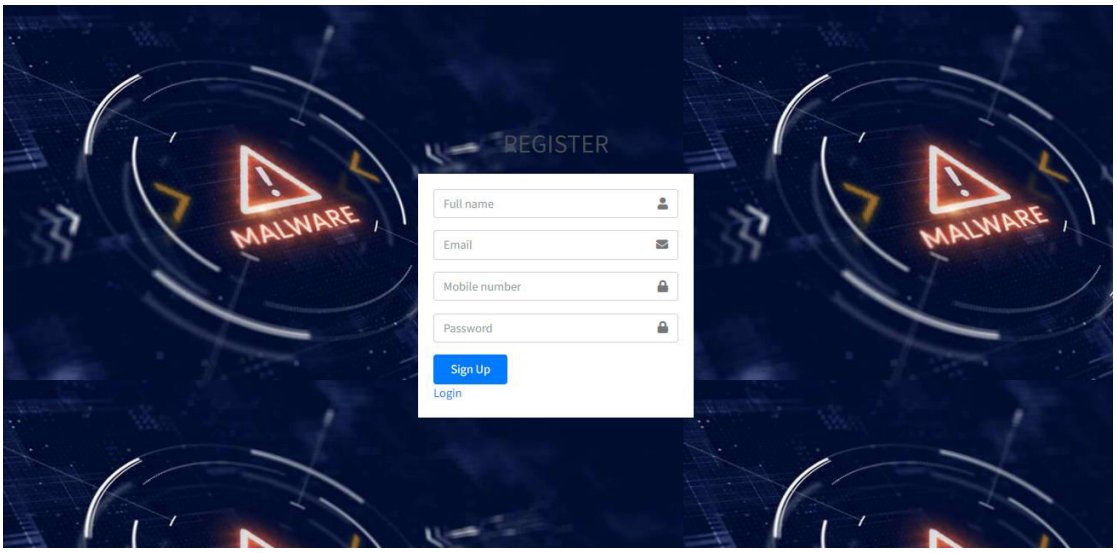

Figure 4: Results of Malware detection

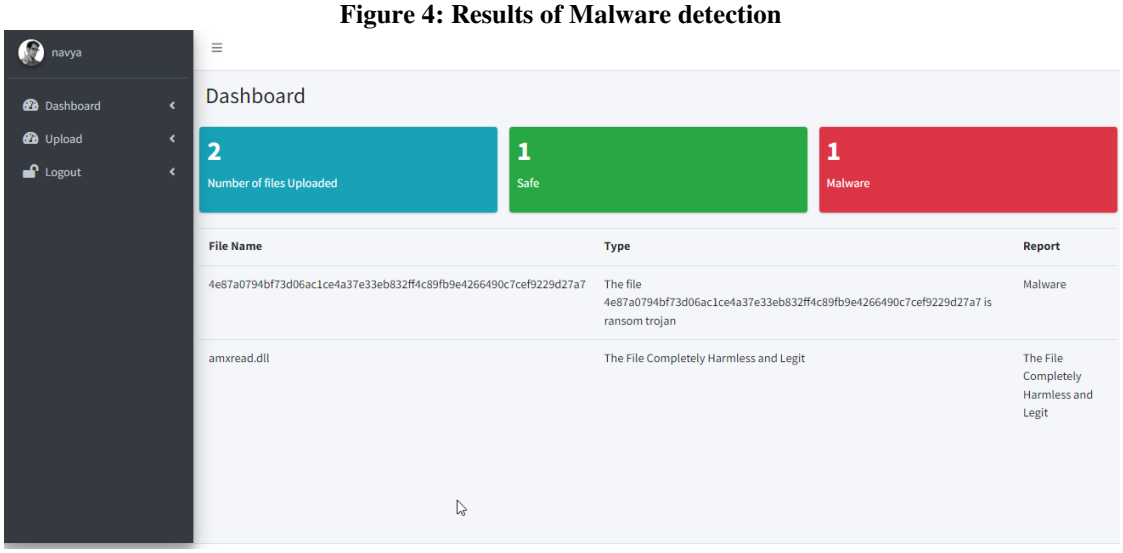

Figure 5: Results of Malware detection

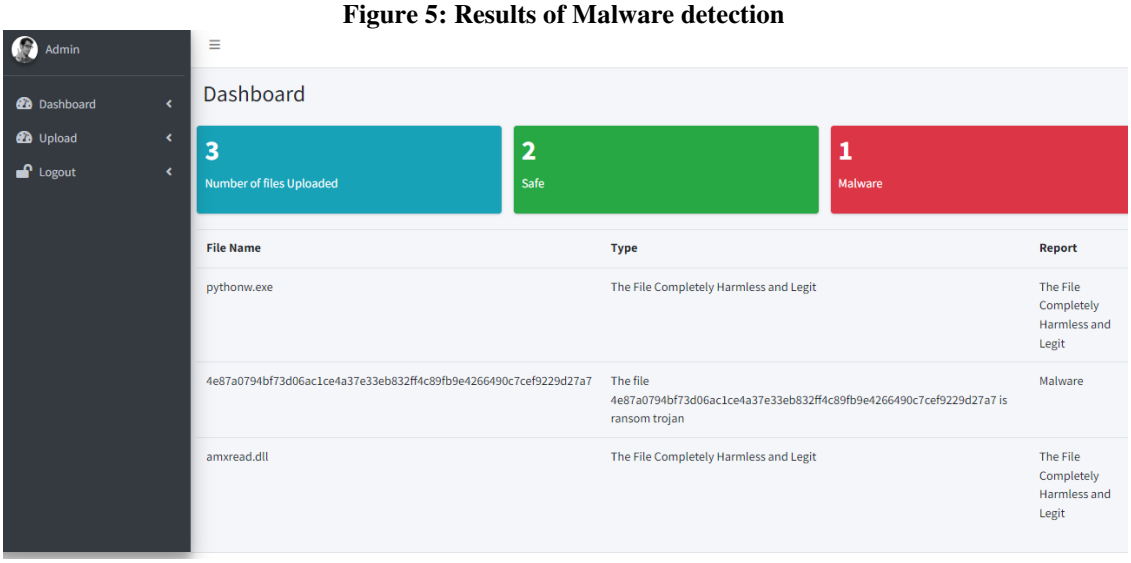

Figure 6: Results of Malware detection

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 6.551 |Bimonthly, Peer Reviewed & Referred Journal|

| Volume 10, Issue 2, March 2023 |

V.CONCLUSION

According to analysis we say that, when we apply different feature selection techniques, it results different no of features and for those features it will give different accuracy for different models. Somewhere it results like one model give the best accuracy is validation but same model cannot give the best accuracy for testing. But in recursively feature elimination technique same model gives the best accuracy in validation as well as testing and model is Random Forest. So we can conclude that this model is best for my analysis and particularly this dataset.

The work can be extended by creating better model for multiclass classifier. So that it can classify each malwares in a better way.

REFERENCES

[1] Nikam, U.V.; Deshmuh, V.M. Performance evaluation of machine learning classifiers in malware detection. In Proceedings of the 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 23–24 April 2022; pp. 1–5. [Google Scholar] [CrossRef]

[2] Akhtar, M.S.; Feng, T. IOTA based anomaly detection machine learning in mobile sensing. EAI Endorsed Trans. Create. Tech. 2022, 9, 172814. [Google Scholar] [CrossRef]

[3] Sethi, K.; Kumar, R.; Sethi, L.; Bera, P.; Patra, P.K. A novel machine learning based malware detection and classification framework. In Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 3–4 June 2019; pp. 1–13. [Google Scholar]

[4] Abdulbasit, A.; Darem, F.A.G.; Al-Hashmi, A.A.; Abawajy, J.H.; Alanazi, S.M.; Al-Rezami, A.Y. An adaptive behavioral-based increamental batch learning malware variants detection model using concept drift detection and sequential deep learning. IEEE Access 2021, 9, 97180–97196. [Google Scholar] [CrossRef]

[5] Feng, T.; Akhtar, M.S.; Zhang, J. The future of artificial intelligence in cybersecurity: A comprehensive survey. EAI Endorsed Trans. Create. Tech. 2021, 8, 170285. [Google Scholar] [CrossRef]

[6] Sharma, S.; Krishna, C.R.; Sahay, S.K. Detection of advanced malware by machine learning techniques. In Proceedings of the SoCTA 2017, Jhansi, India, 22–24 December 2017. [Google Scholar]

[7] Chandrakala, D.; Sait, A.; Kiruthika, J.; Nivetha, R. Detection and classification of malware. In Proceedings of the 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 8–9 October 2021; pp. 1–3. [Google Scholar] [CrossRef]

[8] Zhao, K.; Zhang, D.; Su, X.; Li, W. Fest: A feature extraction and selection tool for android malware detection. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 714–720. [Google Scholar]

[9] Akhtar, M.S.; Feng, T. Detection of sleep paralysis by using IoT based device and its relationship between sleep paralysis and sleep quality. EAI Endorsed Trans. Internet Things 2022, 8, e4. [Google Scholar] [CrossRef]

[10] Gibert, D.; Mateu, C.; Planes, J.; Vicens, R. Using convolutional neural networks for classification of malware represented as images. J. Comput. Virol. Hacking Tech. 2019, 15, 15–28. [Google Scholar] [CrossRef][Green Version]

[11] Firdaus, A.; Anuar, N.B.; Karim, A.; Faizal, M.; Razak, A. Discovering optimal features using static analysis and a genetic search based method for Android malware detection. Front. Inf. Technol. Electron. Eng. 2018, 19, 712–736. [Google Scholar] [CrossRef]

[12] Dahl, G.E.; Stokes, J.W.; Deng, L.; Yu, D.; Research, M. Large-scale Malware Classification Using Random Projections And Neural Networks. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing-1988, Vancouver, BC, Canada, 26–31 May 2013; pp. 3422–3426. [Google Scholar]

[13] Akhtar, M.S.; Feng, T. An overview of the applications of artificial intelligence in cybersecurity. EAI Endorsed Trans. Create. Tech. 2021, 8, e4. [Google Scholar] [CrossRef]

[14] Akhtar, M.S.; Feng, T. A systemic security and privacy review: Attacks and prevention mechanisms over IOT layers. EAI Endorsed Trans. Secur. Saf. 2022, 8, e5. [Google Scholar] [CrossRef]

[15] Anderson, B.; Storlie, C.; Lane, T. "Improving Malware Classification: Bridging the Static/Dynamic Gap. In Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence (AISec), Raleigh, NC, USA, 19 October 2012; pp. 3–14. [Google Scholar]

International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM)

| Mobile No: +91-9940572462 | Whatsapp: +91-9940572462 | ijarasem@gmail.com |

www.ijarasem.com