International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Machine Learning based Multi-Classification Approach for Enhancing Malware Detection for Risk Mitigation

Avinash Kumar, Dr. Meena Chaudhary

Submitted: 05/02/2024 **Revised**: 13/03/2024 **Accepted**: 20/03/2024

Abstract: The current digital era is having superfluous digital information transmitting over various applications. This creates huge number of data that are personal and sensitive in nature to be shared by the user. Hence, the cybercriminals are getting attracted towards these applications for stealing this information for bad intent. The current scenario has seen huge rise in special means of conquering these legal applications and that has come into picture in the form of malware which is a malicious code. Many work has been done for detection of malware for preventing entry of these malware in real time and that too in less time to reduce the mitigation duration that may hamper the business continuity. This paper focuses on the fourth pillar of End Point and Response (EDR) that is use of Machine Learning(ML) techniques to improve detection of malware in automation. The paper has used the concept of identification of API calls based detection which is one of the powerful concepts as, every malware even in obfuscation has to make Application Programming Interface (API) calls in order to initialize or import its library from its command and control server. Even a standalone malware need to make the API calls. The ML models has been used her to efficiently detect the malware by understanding the malicious API calls that a has been made by the malware. This work tries its best perform multi classification for detection of malware to safeguard the system. ML based detection of malware in automation would be further help in reducing cyber mitigation duration. Also, this work would help to detect those API calls that might be accidently ignored by manual inspection as during time of attack, as the duration of mitigation is very less.

Keywords- *Malware Detection, Machine Learning, Mitigation, API Calls, XGBoosting, SGD.*

1. Introduction

In the current digital era, where every filed is using internet or computation application to perform their task, it is very true that it will also increase the amount of data shared by individual while using these digital application with help of internet. The cybercriminal gets lured towards these data, and hence they use various cyber strategies to perform cyber-attacks on the assets to gain personal and sensitive information. Since, the current edge is witnessing various defensive device such as Firewall, Web Application Firewall, Intrusion Detection System (IDS), etc. the hackers have also evolved their techniques for performing attacks. Among various techniques, malware is one of the advanced strategies to perform attack on more secure targets. Malware is a type of malicious code which has all the capabilities of their predecessor like virus, worms, Trojans but what makes them more powerful is their encryption feature. The malware could be mainly categorized on two main types depending upon the nature they are encrypting the target. If the malware encrypts the files of the users to victimize them, then it is termed as Crypto malware and if the malware encrypts the main authentication section in order to lock the device, the it is termed as Locky-Malware. Malware has the capabilities to hide their real intension using obfuscation techniques. They mainly use two major concepts such as cryptor and packers. The malware has again further divided into many families depending upon their evading and lateral movement techniques.

The report form Palo Alto suggest that the malware is bourgeoning with greater pace [1]. The another feature of malware when they collect ransom from the victim is termed as ransomware which is also been faced by many users [2]. There are various classification approaches that has been proposed [3 -6]. The risk of organization getting hit by malware is becoming higher because of increase in digital communication. Since malware is very powerful concept that consist of various evading techniques, there is need for continuous evolution in the malware detection strategies to minimize the risk. Also, the malware being powerful crafted code, does

1Department of Computer Science Engineering, Mangalayatan University, Aligarh, UP, India
avinashkr338@gmail.com
2Department of Computer Science Engineering, Mangalayatan University, Aligarh, UP, India
meenachaudhary9350@gmail.com

very fast lateral movement from one infected unit to another. Hence, to counter that there is need for fast detection mechanism and that could be possible if the detection mechanism is not automated [7-10]. The use of Machine Learning (ML) is one of the vital concepts that can perform detection of malware in automation with minimal human manual intervention.

ML is a concept where a software is made to learn and consequently based upon the learning, it tries to give the best solution [11]. The use of ML has been proposed by various scholar [12 – 19]. There are mainly three major categories of the ML models. These are Supervised ML, Unsupervised ML and Semi-Supervised ML. The Supervised ML works upon the labeled data set for training various algorithms in order to yield efficient output. The Supervised ML is used in various applications but the most common use is to detect the spam from the email. Random Forest (RF), Support Vector Machine (SVM), naïve bayes are few methods that fall under Supervised ML. Another model that is Unsupervised ML uses unlabeled dataset in order to train the software for producing optimal results. The main aim behind these algorithms are to identify the hidden pattern. This includes k-mean clustering, probabilistic clustering, etc. Market segmentation and news clustering are some common use of Unsupervised ML model. Semi-Supervised ML model lies between the Supervised ML and Unsupervised ML models. It uses both labeled and unlabeled data set in proper sequence to arrive at efficient decision.

Application Programing Interface (API) plays a vital role in any software-based communication. API are set of rules that is used by any software to communicate with each other [20]. The use of API based detection of malware is one of the important factors because, every malware is a type of program that needs to communicate with other program or software in order to do its deceptive tasks [21]. Moreover, considering the feature of opcode is another way to detect the malware where the operation code is being used to identify if the sample is malware or not [22]. The use of API based detection is very useful because detection of malware using API calls helps in understanding the behavior of the malicious code, as the advanced malware may hide their real nature when detection mechanism is trying to detect using static malware concepts. The detection mechanism against malware could be made more powerful only when the behaviors is more accurately detected rather than relying only on the signature which is type of static malware analysis [23]. Therefore, this paper focuses on the API call as one of the major concept to discover the malware presence in the victim system or in the defensive deceives.

2. Prior Work

This section of the paper focuses on the reviews of various proposed work that has been achieved as part of the malware detection strategies. This section includes conference papers, journal papers, various articles that highlights the way the researchers has tried their techniques to achieve most optimal detection solution. The phishing threats [24], attack on IoT devices [25] and many others has been considered to detect the malware. The use of hybrid enabled malware detection mechanism has been adopted to detect the malware [26]. This work has been proposed to eliminate the intervention of human for malware detection. The paper lacks the addressing of advanced malware detection techniques in context to API calls.

Another important aspect of malware, when it asks for ransom is the ransomware. The use of decoy based mechanism has been proposed to identify the ransomware [27]. Also, the use of reverse engineering is one of the method to identify or detect the malware [28] but, it needs to be automated using the ML concepts so that the malware detection could be faster process rather than depending upon the manual one which is slower in comparison to the automation. Also, the use of indicators has been proposed [29] to detect the malware where the indicators consists of values or the signatures of the malware. The use of SVM has been proposed to detect windows based malware [30] which is yet another efficient way of detection but lacks the accuracy.

The use of API has been major area of interest because of the behavioral feature attached to it. Hence, when the API use class like CreateFileA over the Windows, it is the intension of the malware sample to drop malicious file over the victim and hence, understanding this API calls may reveal the real intention of the malicious program or the malware [31]. Also, sequential model has been proposed to detect the malware in a system or device where API calls is generated from the malware executable sample [32]. This is useful for the windows based malware detection.

Another method for malware detection has been proposed where the static and dynamic nature has been combined [33]. The use of sandbox has been adopted for the malware analysis. The hybrid model has been implemented to carry out detection. Here again, the use of API calls has been adopted to detect the malware over the infected operating system. Moreover, the dynamic analysis being more advance, needs more accuracy when considering different categories of malware that may arrive within one malware executable sample. Also, it is very important to automate the dynamic analysis of obfuscated malware when using ML concepts.

ML has been proposed for detecting malware [34], where the malware is exhibiting its behavior and that has been detected by the model. Also, the analysis has been done on different parameters to detect the malware [35]. Many has also proposed the use of Random Forest for detection of the malware where these malwares were collected by imaging the memory of the victim system [36]. Moreover, the use of intelligent algorithms has also been proposed for detection strategies of the malware but, still needs more development [37]. There is also, proposed mechanism for detecting malware over software defined network using ML models [38]. The use of Decision Tree using logistic regression has been adopted for malware detection [39].

The use of contextual understating has been proposed to detect and predict the malware on the system [40]. The API calls has been considered for detecting malware and in this case, the malware making API calls based on contextual parameters has been taken into consideration for anlysis [41]. When considering malware detection in categorization of various detection parameters, which may fall under static or dynamic, the API calls has been widely considered [42]. The main reason is the fundamental nature of the API calls that is software interaction which is one of the important feature that is also required by malware to perform its intended task.

The author proposed Unsupervised model of ML and also used the Radom Forest to detect the malware [43]. Here, the detection has been mainly focused using behavioral based analysis. Also, the use of SVM has been considered for identification of malware families.

The work has proposed the use of deep learning for detection of malware where few real devices that were infected were considered [44]. The malware here detected were those which were infecting the android operating system. The android malware is spreading fast mostly in the individual level whereas windows based malware main targets are the large number of the windows systems that could be corporate or other big organization. Also, the android malware needs users' intervention or triggering itself whereas the advanced windows malware can combine themselves with target machine and then gets executed by the victim when, victim is executing legal applications. So, detection of windows malware become more relevant as detection mechanism could be widely used to protect critical windows server.

3. Methodology

This section deals with the proposed models that has been developed using various algorithms of ML. These are elaborated in the below subsections. Fig 1 describes the generalized approach for multiclassification of various classes of malware [45][46].

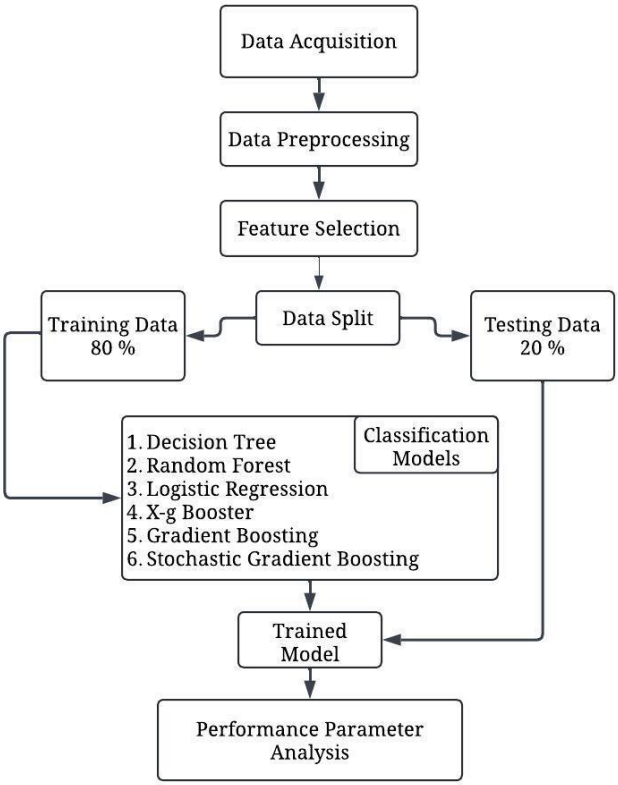

Fig 1. Generalized approach for Multiclassification of Malware

3.1 Data Acquisition

A dataset of Windows Portable Executable (PE) samples is collected. The feature set contains the API functions called by these malwares along with their SHA256 hash values and labels. The types of malwares in the dataset contains Redline Stealer, Downloader, Remote Access Trojan (RAT) and some Benign samples. The source of the malware can be found in https://www.kaggle.com/datasets/joebeachcapital/windows-malwares

3.2 Data Preprocessing

Data preprocessing is a crucial step in preparing a dataset for analysis, ensuring that the data is clean, consistent, and usable. The dataset imported contains null values as well as duplicate values. Null values, or missing values, can affect the analysis and model performance. So, these null values are replaced by categorical mean. The duplicated values are dropped [47][48].

3.3 Feature Selection

Feature selection involves selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection can improve the performance of the model by reducing overfitting, enhancing generalization, and speeding up the training process. Z-score Normalization is performed where scaling is performed on features to have a mean of 0 and a standard deviation of 1. "f_classif" feature selection is used which helps to identify the features that are most strongly associated with the target variable in a classification problem. This method is particularly useful for reducing the dimensionality of the dataset and improving the performance of classification models by focusing on the most significant features. Term Frequency-Inverse Document Frequency (TF-IDF) method is applied which helps in ranking and retrieving the significant features more effectively. The first 20000 features are considered using this TF-IDF method [49][50].

3.4 Data Split

It involves dividing the dataset into different subsets to evaluate the performance of a model. For training the classification models 80% of the malware samples are used and for testing the trained model 20% of the data samples are used [51].

3.5 Classification Models

For multiclassification purpose, several supervised learning algorithms are used that are described in the below sub sections

3.5.1 Decision Trees

Decision Trees (DT) is a concept where it is used in ML for the purpose of developing predictive model. The best part of the DT is that, it is suitable for both classification as well as regression. It follows tree like structure and the internal node is used to test attributes whereas every branch resembles value of the attributes. Finally, the leaf node resembles the prediction or the ultimate decision. The DT is formed based upon the recursive partition that is done on the data which rely on different attributes. Further, the finest attribute is considered for splitting it at every internal node. The split process is performed till the stop criteria is achieved [52].

3.5.2 Random Forest

Random Forest is an ensemble learning algorithm that uses both uses multiple decision trees to improve predictive accuracy and reduce overfitting. During training, Random Forest constructs many decision trees, using a random subset of the dataset and a random subset of features. The randomness ensures that each tree learns different parts of the data, reducing overfitting. For classification, the algorithm uses majority voting among the trees to determine the final class. For regression, it averages the predictions from each tree. There are various advantages of Random Forest. Random Forest deals with complex data well, including noisy or missing features. It provides reliable forecasts even when the environment changes. Also, by analyzing feature importance across trees, we can identify influential features. Random Forest is an example of bagging, where multiple weak models are trained independently and then they are combined. Unlike bagging, boosting trains models sequentially, with each model correcting errors made by previous one.

3.5.3 Logistics Regression

Logistic regression is a supervised machine learning algorithm used for binary classification as well multi- classification. The main purpose of the Logistic regression is to give the probability that if data belongs to a class e.g., spam or not spam, disease or no disease. Logistic regression uses the sigmoid function, also known as the logistic function to map predicted values to probabilities. The sigmoid function ensures that the output lies between 0 and 1. It classifies the non-binary values into binary ones like 0 and 1. If the probability exceeds a threshold value (usually 0.5), the instance is classified into one class and if it remains below 0.5, it belongs to the other class. There are three types of logistic regressions.

Binomial logistic regression is the one which is used for two possible dependent variables 0 and 1. while Multinomial logistic regression considered other parameters also. Finally, Ordinal logistic regression is another type used for three or more ordered types e.g.: low, medium, high.

3.5.4 Gradient Boosting

Gradient Boosting is very useful boosting algorithm which works by combining weaker learner together with the stronger one. Moreover, in each steps, the loss function is minimised and the gradient of the loss is calculated for the more accurate prediction. After, this prediction is done, this new model prediction goes for ensembling and it continues till the criteria for stopping is achieved. It also uses Gradient Boosted Tree which are used for training that are related to regression problems.

3.5.5 XGBoosting

It is like a more enhanced version of gradient boosting. It is designed to improve the machine learning performance. XGBoosting stands for Extreme Gradient Boosting. It's a powerful ensemble learning algorithm that combines the predictions of multiple individual models like decision trees to create an accurate predictive model. It is popular for structured data and dominates applied machine learning tasks. It extends traditional gradient boosting. In gradient boosting, weak learners like decision trees are trained sequentially, with each tree correcting the mistakes of its predecessors. Unlike traditional gradient boosting, XGBoosting constructs trees in parallel rather than sequentially. This significantly improves speed and performance. It includes regularization terms in its objective function. This reduces overfitting and enhances generalization. XGBoosting introduces a new parameter called the learning rate. It controls the contribution of each tree to the overall prediction. A lower learning rate makes the model more conservative and resilient. It builds trees level by level. At each level, it assesses whether adding a new node split improves the overall objective

function. If not, the split is trimmed. This approach makes the trees easier to understand and construct.

3.5.6 Stochastic Gradient Descent (SGD)

SGD is iteration based optimisation process that takes into account optimal value that is maxima and minima. The other important feature of the SGD is that; it provides higher accuracy for both training as well as testing. While using SGD, particular single random sample could be taken instead of selection complete data set. SGD is lesser in expense when taking computation as reference point.

3.6 Performance Parameter Analysis

Performance parameter analysis is crucial in machine learning to evaluate and compare models. It involves using various metrics to assess how well a model performs on different tasks, typically classification or regression. For comparison of the models the metrics used are accuracy, precision, recall and F1 score.

4. Results and Discussion

The malware samples used in this multi classification is categorized into 4 classes: Class 0 as Benign, Class 1 as Redline Stealer, Class 2 as Downloader, Class 3 as Remote Access Trojan (RAT). The samples undergo a series of preprocessing to handle the null and duplicate values in the dataset. The classification models are trained with two cases. In the first case, no feature selection of the data is performed. In the second case, the dataset undergoes feature selection techniques where features of 20000 rows are taken and trained. The below mention table 1 depicts the different accuracy of the classification models for the two cases.

Table 1. Comparative analysis of based on Accuracy.

Classification Models	Without Feature Selection	With Feature Selection
Decision Tree	0.86	0.9076
Random Forest	0.8688	0.9101
XGBoosting	0.8698	**0.9156**
Gradient Boosting Classifier	0.8512	0.8996
SGD Classifier	0.8628	**0.9151**

This shows that the accuracy of these classification models increase to a much satisfactory level when feature selection techniques are employed. Comparing the accuracy levels, the XGBoosting classifier and the SGD classifier outperform the rest of the classification models. The XGBoosting algorithm yields an accuracy of 91.56% while SGD classifier yields an accuracy of 91.51% with feature selection.

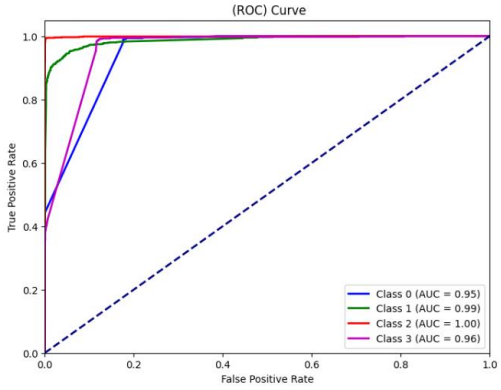

Fig 2 a) ROC curve Xg Boosting classifier (without Feature Selection)

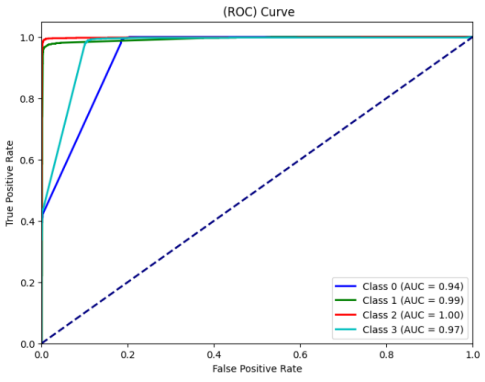

Fig 2 b) ROC curve Xg Boosting classifier (with Feature Selection)

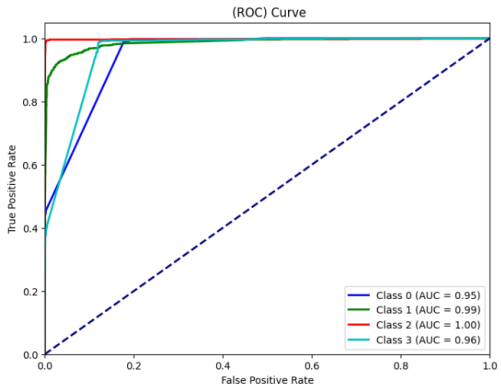

Fig 3 a) ROC curve SGD Classifier (without Feature Selection)

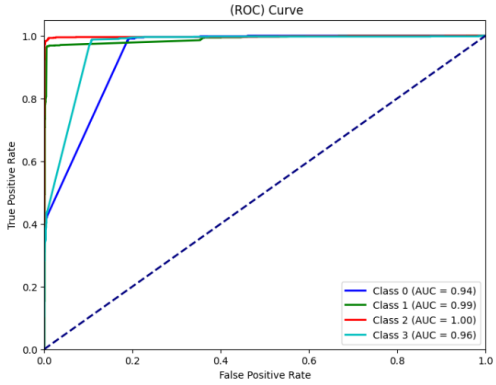

Fig 3 b) ROC curve SGD Classifier (with Feature Selection)

The ROC (Receiver Operating Characteristic) curve is plotted for the two best classification models that achieved higher accuracy for the malware classification. The performance metrics of these two best classification models i.e., XGBoosting classifier and SGD classifier are shown in table 2 and table 3 taking into consideration both the cases i.e., with feature selection and without feature selection.

Table 2 Performance Metrics of XGBoosting classifier and SGD classifier without Feature Selection

Classes of Malware	**XGBoosting classifier**			**SGD classifier**		
	Precision	**Recall**	**F1 score**	**Precision**	**Recall**	**F1 score**
Benign	0.95	0.38	0.61	0.42	0.95	0.61
Redline Stealer	0.93	0.91	0.92	0.92	0.92	0.92
Downloader	0.99	0.99	0.99	0.96	0.99	0.81
RAT	0.70	0.99	0.82	0.74	0.69	0.81

Table 3 Performance Metrics of XGBoosting classifier and SGD classifier with Feature Selection

Classes of Malware	**XGBoosting classifier**			**SGD classifier**		
	Precision	**Recall**	**F1 score**	**Precision**	**Recall**	**F1 score**
Benign	0.99	0.41	0.58	0.44	0.97	0.58
Redline Stealer	0.98	0.97	0.97	0.97	0.98	0.97
Downloader	0.99	0.99	0.99	0.98	0.99	0.99
RAT	0.80	0.98	0.88	0.99	0.79	0.88

Comparing the two tables, it is observed that the feature selection plays a crucial role in classification of malware. Moreover, there can be seen a significant rise in the values of performance metrics after feature selection is implemented.

5. Conclusion

The detection of malware is very crucial step to safeguard the system and also reduce the risk resulting in business continuity. This work has tried its best to not only detect malware but also classify them into different classes such as Class 0 as Benign,

Class 1 as Redline Stealer, Class 2 as Downloader, Class 3 as Remote Access Trojan (RAT) which would help to make more accurate defensive mechanism to counter malware attacks. The paper has created integrated model using Decision Trees, Random Forest, Logistics Regression, XGBoosting, Gradient Boosting and Stochastic Gradient Descent (SGD) to yield maximum efficiency.

References

[1] 2023 unit 42 ransomware and extortion report. (n.d.). https://start.paloaltonetworks.com/2023-unit42-ransomware-extortion-report

[2] Zakaria, W. Z., Abdollah, M. F., Mohd, O., Yassin, S. M., & Ariffin, A. (2022). Rentaka: A novel machine learning framework for crypto-ransomware pre-encryption detection. International Journal of Advanced Computer Science and Applications, 13(5). https://doi.org/10.14569/ijacsa.2022.0130545

[3] Pektaş, A., Pektaş, E. N., & Acarman, T. (2018). Mining patterns of sequential malicious apis to detect malware. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3247606

[4] Skovoroda, A., & Gamayunov, D. (2017). Automated Static Analysis and classification of Android malware using permission and API calls models. 2017 15th Annual Conference on Privacy, Security and Trust (PST). https://doi.org/10.1109/pst.2017.00036

[5] Mpanti, A., Nikolopoulos, S. D., & Polenakis, I. (2018). A graph-based model for malicious software detection exploiting domination relations between system-call groups. Proceedings of the 19th International Conference on Computer Systems and Technologies. https://doi.org/10.1145/3274005.3274028

[6] Balan, G., Simion, C.-A., Gavriluț, D. T., & Luchian, H. (2023). Feature mining and classifier selection for API calls-based malware detection. Applied Intelligence, 53(23), 29094–29108. https://doi.org/10.1007/s10489-023-05086-2

[7] Machine learning-based detection of smartphone malware: Challenges and solutions. (2023). Mesopotamian Journal of Cyber Security, 134–157. https://doi.org/10.58496/mjcs/2023/017

[8] Malware detection system using machine learning and data-mining techniques. (2019). International Journal of Engineering and Advanced Technology, 8(6), 2102–2109. https://doi.org/10.35940/ijeat.f8480.088619

[9] Souri, A., & Hosseini, R. (2018). A state-of-the-art survey of malware detection approaches using data mining techniques. Human-Centric Computing and Information Sciences, 8(1). https://doi.org/10.1186/s13673-018-0125-x

[10] Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. Computers & Security, 81, 123–147. https://doi.org/10.1016/j.cose.2018.11.001

[11] Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). Machine learning and deep learning approaches for cybersecurity: A Review. IEEE Access, 10, 19572–19585. https://doi.org/10.1109/access.2022.3151248

[12] Thangapandian, V. (2022). Machine learning in automated detection of ransomware: Scope, benefits and challenges. Illumination of Artificial Intelligence in Cybersecurity and Forensics, 345–372. https://doi.org/10.1007/978-3-030-93453-8_15

[13] Jung, S., & Won, Y. (2018). Ransomware detection method based on context-aware entropy analysis. Soft Computing, 22(20), 6731–6740. https://doi.org/10.1007/s00500-018-3257-z

[14] Maigida, A. M., Abdulhamid, S. M., Olalere, M., Alhassan, J. K., Chiroma, H., & Dada, E. G. (2019). Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms. Journal of Reliable Intelligent Environments, 5(2), 67–89. https://doi.org/10.1007/s40860-019-00080-3

[15] Kuswanto, D., Husni, & Anjad, M. R. (2021). Application of improved Random Forest Method and C4.5 algorithm as classifier to ransomware detection based on the frequency appearance of API calls. 2021 IEEE 7th Information Technology International Seminar (ITIS). https://doi.org/10.1109/itis53497.2021.9791836

[16] Alhawi, O. M., Baldwin, J., & Dehghantanha, A. (2018). Leveraging machine learning techniques for windows ransomware network traffic detection. Advances in Information Security, 93–106. https://doi.org/10.1007/978-3-319-73951-9_5

[17] Zhang, B., Xiao, W., Xiao, X., Sangaiah, A. K., Zhang, W., & Zhang, J. (2020). Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes. Future Generation Computer Systems, 110, 708–720. https://doi.org/10.1016/j.future.2019.09.025

[18] Hasan, M. M., & Rahman, Md. M. (2017). RansHunt: A support vector machines based Ransomware Analysis Framework with integrated

feature set. 2017 20th International Conference of Computer and Information Technology (ICCIT). https://doi.org/10.1109/iccitechn.2017.8281835

[19] Khan, F., Ncube, C., Ramasamy, L. K., Kadry, S., & Nam, Y. (2020). A digital DNA sequencing engine for ransomware detection using machine learning. IEEE Access, 8, 119710–119719. https://doi.org/10.1109/access.2020.3003785

[20] What is an API (application programming interface)?. IBM. (2021, September 16). https://www.ibm.com/topics/api

[21] Chen, Z.-G., Kang, H.-S., Yin, S.-N., & Kim, S.-R. (2017). Automatic ransomware detection and analysis based on dynamic API calls flow graph. Proceedings of the International Conference on Research in Adaptive and Convergent Systems. https://doi.org/10.1145/3129676.3129704

[22] Zou, C. (2023). Automatic protein sequences classification using machine learning methods based on N-Gram model. 2023 4th International Conference on Machine Learning and Computer Application. https://doi.org/10.1145/3650215.3650382

[23] Shaukat, S. K., & Ribeiro, V. J. (2018). Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning. 2018 10th International Conference on Communication Systems & Networks (COMSNETS). https://doi.org/10.1109/comsnets.2018.8328219

[24] Azeez, N. A., Salaudeen, B. B., Misra, S., Damaševičius, R., & Maskeliūnas, R. (2020). Identifying phishing attacks in communication networks using URL consistency features. International Journal of Electronic Security and Digital Forensics, 12(2), 200. https://doi.org/10.1504/ijesdf.2020.106318

[25] Narayan, V., Daniel, A. K., & Chaturvedi, P. (2023). E-FEERP: Enhanced fuzzy based energy efficient routing protocol for wireless sensor network. Wireless Personal Communications.

[26] Narayan, V., & Daniel, A. K. (2022). CHHP: coverage optimization and hole healing protocol using sleep and wake-up concept for wireless sensor network. International Journal of System Assurance Engineering and Management, 13(Suppl 1), 546-556.

[27] Narayan, V., & Daniel, A. K. (2022). Energy Efficient Protocol for Lifetime Prediction of Wireless Sensor Network using Multivariate Polynomial Regression Model.

[28] Narayan, V., & Daniel, A. K. (2021, October). IOT based sensor monitoring system for smart complex and shopping malls. In International conference on mobile networks and management (pp. 344-354). Cham: Springer International Publishing.

[29] Narayan, Vipul, et al. "A comparison between nonlinear mapping and high-resolution image." Computational Intelligence in the Industry 4.0. CRC Press, 2024. 153-160.

[30] Sandhu, Ramandeep, et al. "Enhancement in performance of cloud computing task scheduling using optimization strategies." Cluster Computing (2024): 1-24.

[31] kumar Mall, Pawan, et al. "Self-Attentive CNN+ BERT: An Approach for Analysis of Sentiment on Movie Reviews Using Word Embedding." International Journal of Intelligent Systems and Applications in Engineering 12.12s (2024): 612-623.

[32] Narayan, Vipul, et al. "7 Extracting business methodology: using artificial intelligence-based method." Semantic Intelligent Computing and Applications 16 (2023): 123.

[33] Yong, B., Wei, W., Li, K., Shen, J., Zhou, Q., Wozniak, M., Połap, D., & Damaševičius, R. (2020). Ensemble machine learning approaches for webshell detection in internet of things environments. Transactions on Emerging Telecommunications Technologies, 33(6). https://doi.org/10.1002/ett.4085

[34] Azeez, N. A., Odufuwa, O. E., Misra, S., Oluranti, J., & Damaševičius, R. (2021). Windows PE malware detection using ensemble learning. Informatics, 8(1), 10. https://doi.org/10.3390/informatics8010010

[35] Lee, J., Lee, J., & Hong, J. (2017). How to make efficient decoy files for ransomware detection? Proceedings of the International Conference on Research in Adaptive and Convergent Systems. https://doi.org/10.1145/3129676.3129713

[36] Khammas, B. M. (2022). Comparative analysis of various machine learning algorithms for ransomware detection. TELKOMNIKA (Telecommunication Computing Electronics and Control), 20(1), 43. https://doi.org/10.12928/telkomnika.v20i1.18812

[37] Scaife, N., Carter, H., Traynor, P., & Butler, K. R. (2016). Cryptolock (and drop it): Stopping ransomware attacks on User Data. 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). https://doi.org/10.1109/icdcs.2016.46

[38] Ngirande, H., Muduva, M., Chiwariro, R., & Makate, A. (2024). Detection and analysis of Android ransomware using the support Vector Machines. International Journal for Research in Applied Science and Engineering Technology, 12(1), 241–252. https://doi.org/10.22214/ijraset.2024.57885

[39] Ali, M., Shiaeles, S., Clarke, N., & Kontogeorgis, D. (2019). A proactive malicious software identification approach for digital forensic examiners. Journal of Information Security and Applications, 47, 139–155. https://doi.org/10.1016/j.jisa.2019.04.013

[40] Catak, F. O., Yazı, A. F., Elezaj, O., & Ahmed, J. (2020). Deep Learning based sequential model for malware analysis using Windows EXE API calls. PeerJ Computer Science, 6. https://doi.org/10.7717/peerj-cs.285

[41] Huang, X., Ma, L., Yang, W., & Zhong, Y. (2020). A method for windows malware detection based on Deep Learning. Journal of Signal Processing Systems, 93(2–3), 265–273. https://doi.org/10.1007/s11265-020-01588-1

[42] Herrera-Silva, J. A., & Hernández-Álvarez, M. (2023). Dynamic feature dataset for ransomware detection using machine learning algorithms. Sensors, 23(3), 1053. https://doi.org/10.3390/s23031053

[43] Maigida, A. M., Abdulhamid, S. M., Olalere, M., Alhassan, J. K., Chiroma, H., & Dada, E. G. (2019). Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms. Journal of Reliable Intelligent Environments, 5(2), 67–89. https://doi.org/10.1007/s40860-019-00080-3

[44] Khan, F., Ncube, C., Ramasamy, L. K., Kadry, S., & Nam, Y. (2020). A digital DNA sequencing engine for ransomware detection using machine learning. IEEE Access, 8, 119710–119719. https://doi.org/10.1109/access.2020.3003785

[45] Bello, I., Chiroma, H., Abdullahi, U. A., Gital, A. Y., Jauro, F., Khan, A., Okesola, J. O., & Abdulhamid, S. M. (2020). Detecting ransomware attacks using intelligent algorithms: Recent development and next direction from Deep Learning and Big Data Perspectives. Journal of Ambient Intelligence and Humanized Computing, 12(9), 8699–8717. https://doi.org/10.1007/s12652-020-02630-7

[46] Hirano, M., Hodota, R., & Kobayashi, R. (2022). RANSAP: An open dataset of ransomware storage access patterns for training machine learning models. Forensic Science International: Digital Investigation, 40, 301314. https://doi.org/10.1016/j.fsidi.2021.301314

[47] Urooj, U., Al-rimy, B. A., Zainal, A., Ghaleb, F. A., & Rassam, M. A. (2021). Ransomware detection using the dynamic analysis and Machine Learning: A Survey and Research Directions. Applied Sciences, 12(1), 172. https://doi.org/10.3390/app12010172

[48] Amer, E., & Zelinka, I. (2020). A dynamic windows malware detection and prediction method based on contextual understanding of API call sequence. Computers & Security, 92, 101760. https://doi.org/10.1016/j.cose.2020.101760

[49] Zhao, Y., Bo, B., Feng, Y., Xu, C., & Yu, B. (2019). A feature extraction method of hybrid gram for malicious behavior based on machine learning. Security and Communication Networks, 2019, 1–8. https://doi.org/10.1155/2019/2674684

[50] Shinde, O., Khobragade, A., & Agrawal, P. (2023). Static malware detection of ember windows-PE API call using machine learning. COMPUTATIONAL INTELLIGENCE AND NETWORK SECURITY. https://doi.org/10.1063/5.0130256

[51] Choudhary, S., & Sharma, A. (2020). Malware detection & classification using machine learning. 2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3). https://doi.org/10.1109/iconc345789.2020.9117547

[52] Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2020). DL-Droid: Deep Learning based Android malware detection using real devices. Computers & Security, 89, 101663. https://doi.org/10.1016/j.cose.2019.101663