IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN2347–3657

Volume13, Issue 1,2025

DETECTING AND CLASSIFYING RANSOMWARE WITH MACHINE LEARNING ALGORITHMS

[1]P SUBRAMANYAM VINODHINI, [2]DODDANNAGARI AKHILA, [3]KUCHANI VAMSHI, [4]SURYA JUGAL, [5]MISHE KASHINATH, [6]Mr. PAINDLA THIRUPATHI, [7]Mr. P VAMSI KRISHNA,

[12345]Student Department of DS, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

[6]Assistant Professor, Department of CSE, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

[7]Assistant Professor, Department of Mechanical Engineering, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

Abstract

Cybersecurity is always at risk from malicious assaults, malware, and ransomware families, which may wreak havoc on several industries' worth of computer networks, data centers, websites, and mobile apps. Conventional anti-ransomware technologies are rendered ineffective by more complex ransomware assaults. Therefore, cutting-edge methods, including traditional and neural network-based designs, may be invaluable to developing new ransomware remedies. In this paper, we provide a framework for detecting and preventing ransomware that uses a number of machine learning methods, such as designs based on neural networks and feature selection, to categorize security levels. A few of ransomware characteristics were subjected to several machine learning methods, including Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Logistic Regression (LR), and NN-based classifiers. For the purpose of thoroughly evaluating our suggested technique, we used a single ransomware dataset. Experimental data show that RF classifiers beat competing approaches on accuracy, F-beta, and precision measures.
Information security, AI, NN, ransomwarecategorization

INTRODUCTION

For instance, diverse industries' computer networks, data centers, websites, and mobile applications are vulnerable to ransomware and other forms of malicious software, which pose a serious risk to cybersecurity. the first – third. The fundamental objective of the majority of ransomware is to encrypt data in a manner that can only be deciphered by the attacker, hence denying the victim access to their own data. Ransomware demands payment from victims in order to decrypt their files, which is a permanent process [4]. If you do not comply with the demand of the attacker, all of your information will be gone forever. The use of modern technology by cybercriminals has resulted in the evolution of ransomware into new families, making it far more difficult to reverse an infection [5].

Ransomware is an ever-changing, sophisticated threat that encrypts user files or locks the screen, preventing access to data or the machine unless a ransom is paid [2]. Depending on the vector of attack, ransomware may assume one of two primary forms: locker ransomware, which encrypts all device or computer access, or crypto ransomware, which encrypts all file or data access [6]. It is quite difficult to reverse these attacks without paying the extortion. Ransomware detection strategies that rely on statistics, event-based analysis, or data-centric approaches are ineffective. Therefore, in order to resist these complex hostile attacks, the research community should focus on developing first-rate security and protection measures by using state-of-the-art technology.

One emerging area of study that shows great potential for the creation of novel anti-ransomware solutions is ransomware detection using machine learning [7]. Incorporating Machine Learning (ML) methods into security measures has the potential to enhance protection and enable the automatic identification of malicious software, such as ransomware, according to their behavior patterns [8]. A few neural network methods, including Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Logistic Regression (LR), and NN, might be beneficial in the detection and classification of ransomware [9]. We examine and assess machine learning techniques utilized for

ransomware categorization in this study. The primary areas that are improved by this research are:

We first explore ransomware classification in more depth, and then present a system that selects model building features using a combination of conventional ML classifiers and NN-based architectures.
Through thorough experimentation and comparison to other methods, we prove that the models' performance is generalizable.

What follows is an outline of the rest of the paper: We go into the connection between ML and ransomware detection in Section II. The procedures used to compile the information for this study are detailed in Section III. Results and experimental setup are detailed.

RELATED WORK

Ransomware and other forms of malware have traditionally been classified using conventional malware detection methods. The shared characteristics of ransomware families allow for their analysis within a well-defined behavioral framework. These families share characteristics including payload persistence, stealth tactics, and network activity. Most people still use the classic anti-malware method that A. M. Abiola and M. F. Marhusin [10] developed: a signature-based detection technique. It relied on n-gram signature decomposition after Brontok worm extraction. The foundation for malware detection is laid by the framework, which offers a dependable solution that gets rid of all threats. For this reason, [11] advocated a behavior-based strategy that uses both static and dynamic analysis to get over this issue. This framework monitors processes for suspicious activity and terminates them properly depending on what it finds. It also uses a static-based technique to check the application's code for dangerous activities. Finding unknown malware, code obfuscation, high variance output, or targeted assaults becomes a challenge when using static and dynamic-based analysis. Researchers F. Noorbehbahani and M. Saberi [8] found that ransomware may be detected using semi-supervised learning, which uses a combination of labeled and unlabeled data. Using the CICAndMal 2017 dataset, researchers used various feature selection and semi-supervised classification algorithms until they found that the technique using random forest as its fundamental classifier produced the best results for ransomware detection. Compared to older, less effective methods, ransomware detection and prevention employing state-of-the-art machine learning techniques may be preferable. In addition to a unique flow-oriented method for ransomware detection known as Biflow, researchers [12] suggested a network intrusion detection architecture that incorporates the Argus server and client programs. The dataset was classified using six feature selection procedures, and the detection module's accuracy and performance were improved using supervised machine learning. When it comes to detecting malware and ransomware, Random Forest is a prominent machine learning approach. The Digital DNA Sequencing Engine is the basis of the DNAact-Ran ransomware detection technology, according to F. Khan et al. [13]. The k-mer frequency vector and the constraints of the sequencing scheme are given precedence. To test how well the framework performed, we used 582 DNAact-Run ransomware and 942 goodware examples. Ransomware detection using machine learning was presented by S. Poudyalwe et al. [14]. Using multi-level analysis, this approach reveals the purpose of parts of malware code. The findings show that the model can identify ransomware with a performance level between 76% and 97%. A machine learning technique was suggested by V. G. Ganta et al. [15] as a substitute for the conventional way of ransomware detection. Ransomware in executable files may be detected using a number of classification methods. A few algorithms that fall under this category include ex-random forest, logistic regression, decision trees, and KNN.

As a way for dynamically assessing and categorizing ransomware, EldeRan was suggested by Daniele Sgandurra et al. [16]. In view of the expected distinctive symptoms of ransomware, it analyzes the infected software's actions. Feature selection and classification are two ML components that EldeRan employs in the Cuckoo Sandbox environment. Methods for dynamically accessing and analyzing datasets include Windows API calls, Strings, Registry Key Operations, File System Operations, Dropped Files, and File System Operations. The accuracy of the framework was shown using an area under the ROC curve of 0.995 using 582 ransomware files from 11 distinct families and 942 goodware programs. In order to successfully categorize ransomware based on its behavior, Sumith Maniath et al. [17] presented a technique for binary sequence classification of API calls using Long-Short Term Memory (LSTM) networks. The API calls were extracted from the modified log using a dynamic analysis approach in a sandbox setting. The results of the test demonstrate that the suggested LSTM-based framework successfully

International Journal of Information Technology & Computer Engineering
ISSN2347–3657
Volume13, Issue 1,2025

A success rate of 96.67% was achieved by automatically classifying ransomware activity using a huge collection of malicious code. Improved overall accuracy is possible, however, with a reinforced LSTM network. In order to find new kinds of ransomware, ML could be a good technology to utilize because to its accuracy. Thanks to their innovative dynamic detection mechanism, Deep Neural Networks (DNNs) can identify ransomware and other complicated threats. A new framework for automated hyperparameter tuning based on Bayesian optimization was suggested by researchers [18] for Deep Neural Network-Based Network Intrusion Detection. Recent work by Hadis Ghanei et al. [19] described research that offered a CNN and DNN-based solution for real-time malware identification. Building the ML model makes use of the LSTM. Something new was done to find potentially dangerous viruses by combining CNNs with the LSTM network. The assessment study found that when it comes to identifying new malware, a combination of DNN and LSTM had an accuracy rate of 91.63%. One area where Deep Learning has proven useful is in the fight against malware on Android devices. In order to classify Android malware, M. Masum and H. Shahriar presented a deep learning framework called Droid-NNet. The automatic deep learning capabilities of this framework are unmatched by even the most sophisticated machine learning algorithms. Malgenome-215 and Drebin-215 are two datasets used for testing Android applications; Droid-NNet exhibits robust and efficient malware detection on Android [20]. Testing with many ML classifiers, including neural network architecture, allowed for the fine-tuning of the suggested system using a small number of critical features. If the trials pan out, we may say that the suggested structure is solid and works as advertised.

METHODOLOGY

Neural network architecture and standard machine learning classifiers including decision tree, logistic regression, naïve bayes, and random forest classifiers were used for malware detection. You can see the model's structure in Figure 1. We started by checking that all of the scale variables were somewhat consistent before trying to normalize the ransomware data. In order to differentiate between ransomware and valid observations, we used a feature selection approach to extract a handful of essential properties from the data. These attributes were then fed into several classifiers. We used a 10-fold cross validation technique to enable the model to be applied to a wider range of scenarios. To conclude, we provided a variety of assessment measures for the models to be used in their evaluation, such as accuracy, F-beta score, recall, precision, and area under the ROC curve.

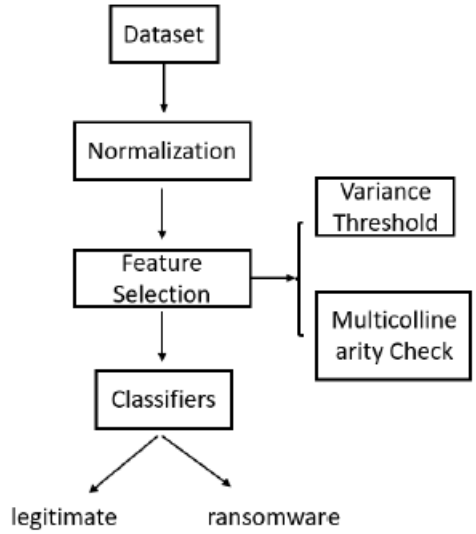

Figure 1: Framework to detect ransomware

EXPERIMENTS AND RESULTS

Dataset requirements

Seventy percent of the 138,047 samples with 54 characteristics were found to be ransomware, while the other thirty percent were genuine observations. The dataset was compiled from [21]. The distribution of the dataset is seen in Figure 2.

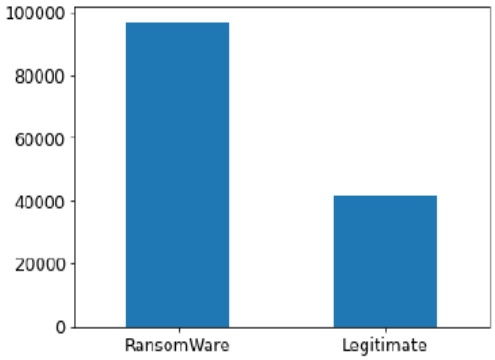

Figure 2 : Distribution of the dataset

Feature Selection

By centering them all at zero and giving them a standard deviation of one, the Z-score standardization procedure brought all of the variables into a consistent scale. For low variant

International Journal of Information Technology & Computer Engineering

ISSN2347–3657

Volume13, Issue 1,2025

features, we used variance inflation factor, and for strongly correlated features, we used variance threshold, as feature selection approaches. Since the number of features reduced significantly from 54 to 13 when the threshold was set to 1, we removed low variant features from the dataset by setting the variance threshold score to 1. A variety of characteristics with different variance threshold scores are shown in Figure

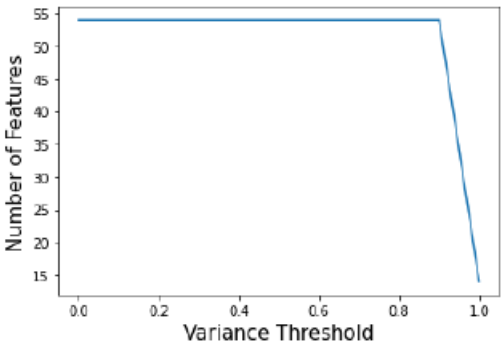

Figure 3: Variation in variance threshold and the number of characteristics

Using the variance inflation factor (VIF), we verified that the high variance features were not multicollinear in the second round of feature selection. Feature identification is based on a VIF score greater than 10, which was chosen to detect strongly associated features. Features: With VIF scores of 19.52 and 19.48, respectively, SectionMeanRawSize and SectionMaxRawSize demonstrate multicollinearity. One of these variables was dropped at random. All twelve of the high variant traits shown in Table 1 have VIF scores that are below the cutoff. In the end, we train the classifiers to identify ransomware using these 12 characteristics.
Features that were considered after using the variance threshold and the VIF criteria are shown in Table 1.

Feature	VIF
SizeOfOptionalHeader	1.24
MajorLinkerVersion	1.15
AddressOfEntryPoint	1.04
SectionAlignment	1.03
MinorOperatingSystemVersion	4.04
SizeOfHeaders	1.0
SizeOfStackReserve	1.19
LoaderFlags	4.04
SectionsMinEntropy	1.31
SectionsMaxEntropy	1.41
SectionMaxRawsize	1.0
SectionsMinVirtualsize	1.02
ResourcesMinEntropy	1.08

Evaluation metrics

Recall is the total number of positive samples that have accurate forecasts. Based on mathematical principles:

$$Recall = \frac{TP}{TP + FN}$$

Here, TP stands for True Positive, which is the number of accurate positive predictions, and FN for False Negative, which is the number of incorrectly categorized positive forecasts.
2. Accuracy: The ratio of expected positives to actually detected positives. Using mathematical methods

$$Precision = \frac{TP}{TP + FP}$$

The harmonic means of Precision and Recall are $F1\ bcofq$. In cases of unbalanced data, $F1\ bcofe$ outperforms the accuracy measure.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F_\beta = (1 + \beta^2)\frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall}$$

The value of F-beta is $F1\ ycore$ when β= 1. Precision and recall are weighted according to the β parameter. If we wish to prioritize accuracy, we may use β<1, while values greater than 1 indicate a higher priority on recall.

International Journal of Information Technology & Computer Engineering

ISSN2347–3657

Volume13, Issue 1,2025

Experimental setting

To gauge our model's efficacy, we contrasted its output with that of the LR, NB, RF, and DT methods. Both sets received the training and test datasets at random, with the same ratio of legitimate to malicious samples preserved. In every experiment, we used trained data for model training and test data for model assessment. In order to verify the consistency of each model, we used 10-fold cross-validation.
We compared the results of the two datasets using NN-based classifiers with those of RF, LR, NB, and DT. The algorithms were constructed using the scikit-learn package in Python, in conjunction with the available hyperparameter settings.

The design is based on a neural network and consists of one input layer, two hidden layers, and one output layer. We used the 'ReLu' activation function for the hidden layers and the'sigmoid' function for the output layer during training since we knew this was a problem with binary classification. Adam was the name of the optimizer, while binary cross-entropy was the name of the loss function. Using an early stopping technique, we terminate training when the model's performance on the test data stops improving. We set minimum delta to 1.3 to check if the monitored quantity changes by that much to be considered an improvement, and patience to 5 to check how many epochs pass without an improvement in the monitored quantity before training is terminated. We decided to track validation loss so that we could stop training early. Our initial learning rate was 0.01.

Results

We used DT, RF, NB, LR, and NN classifiers to differentiate between legitimate and ransomware samples. Recall, accuracy, precision, and F-beta score are shown in Table 3 along with the results of the models. The Random Forest classifier outperforms competing models with respect to accuracy, precision, and F-beta score. All performance metrics are severely underwhelmed by the NB classifier, despite its top recall. Both the DT and NN classifiers perform well when compared to RF. In comparison to other methods, LR fails to achieve acceptable F-beta and recall scores, even if it outperforms DT, RF, and NN classifiers in terms of accuracy. For each classifier, Figure 4-8 displays the ROC curves, which include both 10-fold and mean curves. While NN, RF, and LR all reached maximum Area Under Curve (AUC) ratings of 0.99, NB had the lowest result at 0.73. Experiment Results Shown in Table 2 for Classifier Assessment

Classifiers	*Accuracy*	*F-beta*	*Recall*	*Precision*
DT	0.98±0.01	0.94±0.05	0.94±0.05	0.98±0.00
RF	**0.99±0.01**	**0.97±0.03**	0.97±0.03	**0.99±0.00**
NB	0.35±0.03	0.97±0.03	**0.99±0.00**	0.31±0.01
LR	0.96±0.02	0.89±0.07	0.89±0.07	0.96±0.00
NN	0.97±0.01	0.95±0.05	0.95±0.05	0.97±0.00

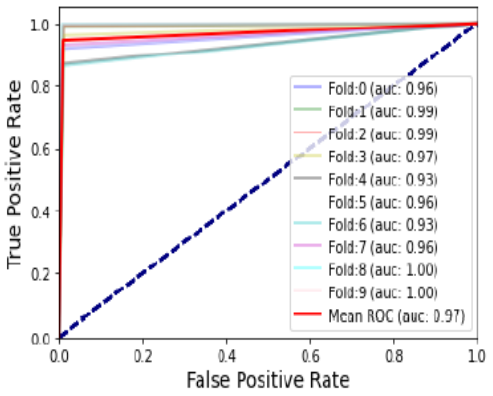

Figure 4: ROC curve for Decision Tree classifier

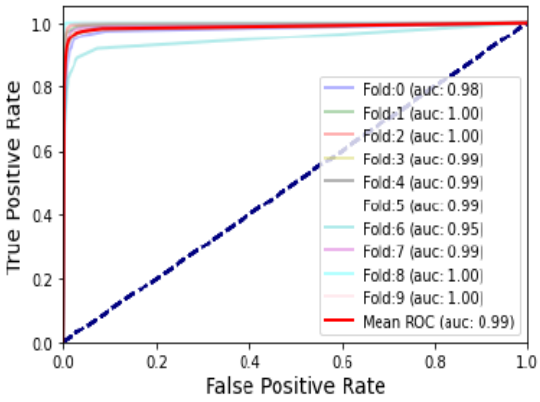

Figure 5: ROC curve for Random Forest classifier

International Journal of Information Technology & Computer Engineering

ISSN2347–3657

Volume13, Issue 1,2025

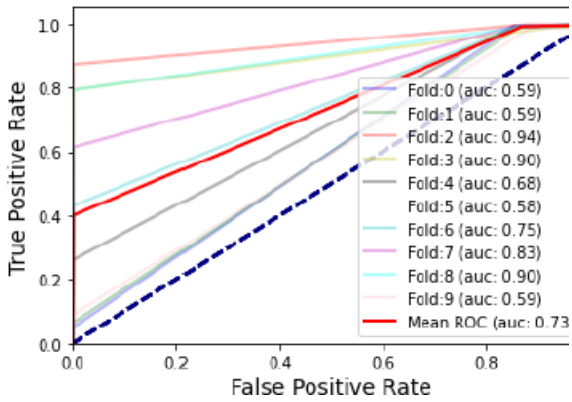

Figure 6: ROC curve for Naïve Bayes classifier

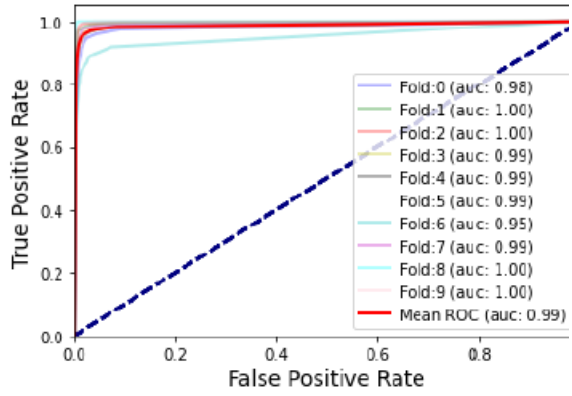

Figure 7: ROC curve for Logistic Regression classifier

CONCLUSION

Malware threats, particularly ransomware, are increasing and impact all sectors of society, including banks, consumers, and enterprises. We must create an automated system that can identify and categorize ransomware if we want to lessen the likelihood of harmful actions. Our innovative feature selection-based framework and neural network classifiers were among the several machine learning approaches we employed to successfully identify and categorize malware. Using the architecture, we conducted exhaustive performance comparisons of the DT, RF, NB, LR, and NN classifiers on a ransomware dataset. By consistently attaining the best accuracy, F-beta, and precision scores during all 10 rounds of cross-validation, the experimental findings show that the Random Forest classifier beats the others.

REFERENCES

[1]. K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade," IEEE Access, vol. 8, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.

[2]. N. Shah and M. Farik, "Ransomware-Threats, Vulnerabilities And Recommendations," Int. J. Sci. Technol. Res., 2017, [Online]. Available: https://www.ijstr.org/final-print/june2017/Ransomware-Threats-Vulnerabilities-And-Recommendations.pdf.

[3]. M. J. Hossain Faruk et al., "Malware Detection and Prevention using Artificial Intelligence Techniques," Proc. - 2021 IEEE Int. Conf. Big Data, Big Data 2021, 2021, [Online]. Available: https://www.researchgate.net/publication/356163392_Malware_Detection_and_Prevention_using_Artificial_Intelligence_Techniques.

[4]. e_Techniques.

[5]. F. Noorbehbahani, F. Rasouli, and M. Saberi, "Analysis of machine learning techniques for ransomware detection," Proc. 16th Int. ISC Conf. Inf. Secur. Cryptology, Isc. 2019, pp. 128–133, 2019, doi: 10.1109/ISCISC48546.2019.8985139.

[6]. [5] U. Adamu and I. Awan, "Ransomware prediction using supervised learning algorithms," Proc. - 2019 Int. Conf. Futur. Internet Things Cloud, FiCloud 2019, pp. 57–63, 2019, doi: 10.1109/FiCloud.2019.00016.

[7]. [6] K. Savage, P. Coogan, and H. Lau, "The Evolution of Ransomware," Res. Manag., vol. 54, no. 5, pp. 59–63, 2015, [Online]. Available: http://openurl.ingenta.com/content/xref?genre=article&issn=0895-6308&volume=54&issue=5&spage=59.

[8]. D. W. Fernando, N. Komninos, and T. Chen, "A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques," IoT, vol. 1, no. 2, pp. 551–604, 2020, doi: 10.3390/iot1020030.

[9]. [8] F. Noorbehbahani and M. Saberi, "Ransomware Detection with Semi-Supervised Learning," 2020 10h Int. Conf. Comput. Knowl. Eng. ICCKE 2020, pp. 24–29, 2020, doi: 10.1109/ICCKE50421.2020.9303689.

[10]. L. Chen, C.-Y. Yang, A. Paul, and R. Sahita, "Towards resilient machine learning for ransomware detection," 2018, [Online]. Available: http://arxiv.org/abs/1812.09400.

[11]. A. M. Abiola and M. F. Marhusin, "Signature-based malware detection using sequences of N-grams," Int. J. Eng. Technol., vol. 7, no. 4, pp. 120–125, 2018, doi: 10.14419/ijet.v7i4.15.21432.

[12]. D. Nieuwenhuizen, "A behavioural-based approach to ransomware detection," MWR Labs, 2017, [Online]. Available: https://labs.f-secure.com/assets/resourceFiles/mwri-behavioural-ransomware-detection-2017-04-5.pdf.

[13]. Y. L. Wan, J. C. Chang, R. J. Chen, and S. J. Wang, "Feature-Selection-Based Ransomware Detection with Machine Learning of Data Analysis," 2018 3rd Int. Conf. Comput. Commun. Syst. ICCCS 2018, pp. 392–396, 2018, doi: 10.1109/CCOMS.2018.8463300.

[14]. F. Khan, C. Ncube, L. K. Ramasamy, S. Kadry, and Y. Nam, "A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning," IEEE Access, vol. 8, pp. 119710–119719, 2020, doi: 10.1109/ACCESS.2020.3003785.

[15]. S. Poudyal, K. P. Subedi, and D. Dasgupta, "A Framework for Analyzing Ransomware using Machine Learning," Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018, pp. 1692–1699, 2019, doi: 10.1109/SSCI.2018.8628743.

[16]. V. G. Ganta, G. Venkata Harish, V. Prem Kumar, and G. Rama Koteswar Rao, "Ransomware Detection in Executable Files Using Machine Learning," Proc. - 5th IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. RTEICT 2020, pp. 282–286, 2020, doi: 10.1109/RTEICT49044.2020.9315672.

[17]. D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection," 2016, [Online]. Available: http://arxiv.org/abs/1609.03020.

[18]. S. Maniath, A. Ashok, P. Poornachandran, V. G. Sujadevi, A. U. P. Sankar, and S. Jan, "Deep learning LSTM based ransomware detection," 2017 Recent Dev. Control. Autom. Power Eng. RDCAPE 2017, pp. 442–446, 2018, doi: 10.1109/RDCAPE.2017.8358312.

[19]. M. Masum et al., "Bayesian Hyperparameter Optimization for Deep Neural Network-Based Network Intrusion Detection," Prc. - 2021 IEEE Int. Conf. Big Data, Big Data 2021, 2021, [Online]. Available: https://www.researchgate.net/publication/357164131_Bayesian_Hyperparameter_Optimization_for_Deep_Neural_Network-Based_Network_Intrusion_Detection/citation/download.

[20]. H. Ghanei, F. Manavi, and A. Hamzeh, "A novel method for malware detection based on hardware events using deep neural networks," J. Comput. Virol. Hacking Tech., vol. 17, no. 4, pp. 319–331, 2021, doi: 10.1007/s11416-021-00386-y.

[21]. M. Masum and H. Shahriar, "Droid-NNet: Deep Learning Neural Network for Android Malware Detection," Proc. - 2019 IEEE Int. Conf. Big Data, Big Data 2019, pp. 5789–5793, 2019, doi: 10.1109/BigData47090.2019.9006053