Vol.15, Issue No 2, 2025

# Identification & Categorization of Ransomware using Machine Learning

[1] Kumpatla Nandini, [2] Gullipalli Kavitha, [3] Koyyana Nageswari Priyanka, [4] Ponnamanda Jayanth, [5] Mrs. J. Priyanka,

[1,2,3,4,] Students, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

[5] Assistant Professor, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

## *Abstract—*

Malware infections have been rising daily in tandem with the expansion of computer networks and the Internet. Ransomware is one of the most recent forms of cyberattack and a major concern in the field nowadays. While several studies have investigated the efficacy of using machine learning approaches for malware detection, very few have concentrated on ransomware detection using machine learning. Using the CICAndMal2017 dataset, this study conducts two experiments to assess the efficacy of ransomware detection using machine learning approaches. The first step is to train the classifiers on a single dataset that includes several ransomware kinds. Secondly, separate datasets for each of the ten ransomware families are used to train the various classifiers. We found that random forest outperformed other classifiers in both studies, and that training the classifiers separately on each family had no discernible effect on their performance. As a result, ransomware detection using the random forest classification approach is very successful. Terms such as malware detection, classification techniques, machine learning, and ransomware

## INTRODUCTION

Ransomware encrypts or locks a victim's data and demands payment to unlock it again; it is a major security concern for people, corporations, and governments. "The year of the ransomware" was 2016, and today, ransomware remains a major security concern. Ransomware is distinct from other forms of malware in that it is very difficult to eradicate and the damage it does is permanent. New families of ransomware are becoming harder to detect, and their proliferation is driven by the lucrative nature of ransomware. Cryptowall, Locky, Cerber, and others fall into this category, and CryptXXX2.0 and CryptXXX3.0 are just a couple of versions among many. Ransomware broadly falls into two categories: locker-ransomware and crypto-ransomware. Locker ransomware prevents victims from accessing their devices by locking them. Encrypting data to prevent victims from accessing it, crypto-ransomware is the most frequent sort of ransomware. There are two main types of ransomware detection methods: those that look for abuse and those that look for anomalies. Anomaly detection approaches mimic the typical system behavior and trigger an alert in the event of a deviation, while abuse detection methods utilize ransomware signatures that are already known. Methods for detecting ransomware range from those based on events to those based on statistics, machine learning, and data.

A number of studies have used machine learning techniques to identify malware; however, detecting ransomware using the same methods is an emerging area of study.

This paper examines the efficacy of machine learning techniques in detecting ransomware. We put several supervised learning algorithms for ransomware detection to the test, including Decision Tree (DT) [1], Random Forest (RF) [2], Random Tree (RT) [3], K-Nearest Neighbors (KNN) [4], Naive Bayes (NB) [5], and Support Vector Machines (SVM) [6]. The remaining sections of the document are organized as follows: Part II reviews the related work on ransomware detection using machine learning approaches. The methodology for evaluating machine learning techniques for ransomware detection is detailed in Section III. In section IV, the evaluation results are presented and finally, section V concludes the paper.

RELATED WORK

The authors of [7] suggest a software-defined networking (SDN) based ransomware detection approach that uses characteristics retrieved from malware traffic. Cryptowall and Locky are two kinds of ransomware that the authors think may be detected

by examining HTTP communications. EldeRan, described in [8], is a machine learning approach for ransomware classification and detection. This approach may identify ransomware by dynamically checking the activities made by apps during installation. They used a dataset with 582 ransomware and 942 benign occurrences to test their strategy. The findings shown that ransomware and its novel variations may be effectively detected using machine learning. They trained and updated the model using regularized logistic regression and used mutual information for feature selection. It seems that the assessment dataset is somewhat little, hence fresh ransomware datasets are needed to evaluate their suggested method's efficacy in ransomware detection. To overcome the static nature of signature-based detection approaches, a data mining–based dynamic ransomware detection system is suggested in [9]. For the purpose of ransomware classification, they use a variety of data mining techniques, including Naive Bayes (NB), Support Vector Machine (SVM), Simple Logistic (SL), and Random Forest (RF). The foundation of their suggested approach is the creation of API Call Flow Graphs (CFG). With a detection rate of 0.976 and an accuracy of 0.982 when using the SL approach, their data suggest that their suggested ransomware detection system is successful. The absence of a standardized ransomware dataset for testing is the key issue with their approach. In their evaluation of their suggested technique, they only used a dataset of 168 cases. The authors used deep learning to identify malware in [10]. They used a dataset that included both malicious and benign samples taken from actual network traffic to train a deep neural network. Their technology may identify threats at an early stage of infection and is beneficial for ransomware detection, according to the findings. They are certain that their technology can be applied to real-world network topologies since it can be implemented on SDN switches. An SVM-based ransomware detection approach was suggested by Takeuchi et al. [11]. In order to train an SVM classifier, their approach used characteristics that were produced by API calls of ransomware. It is impossible to measure the success of their suggested technique due to the fact that the evaluation dataset only comprises 312 benign and 276 ransomware samples, which is a shortcoming of their experiment. The Windows ransomware detection system NetConverse, developed by Alhawi and colleagues [12], uses machine learning techniques. Logistic Model Tree, Bayes Network, Decision Tree, Multi-Layer Perceptron, Random Forest, and K-Nearest Neighbors were all put through their paces in order to identify ransomware. The decision tree emerged as the top classifier for detecting ransomware, according to the data. Once again, there is no conventional ransomware dataset to evaluate this study. A novel approach to ransomware detection is suggested in [13] using programmable forwarding engines (PFEs). Their approach used packet-filtering endpoints (PFEs) to inspect data sent between a compromised host and the C&C server. Based on unencrypted properties of HTTPS data, they used a random forest and a binary classifier to develop a model. There is a dearth of literature on ransomware detection on the Android platform since this is a young field of study [14]. One program that uses natural language processing (NLP) characteristics to identify ransomware is HelDroid [15]. A text classifier, which is crucial to HellDroid's performance, is vulnerable to attacks like string encryption. Another piece of work is R-PackDroid, a machine learning approach that Maiorca et al. [14] suggested for detecting android ransomware. In addition to detecting new ransomware, the authors demonstrated that their approach can efficiently distinguish between ransomware, generic malware, and innocuous files.

## ANALYSIS PROCESS

Here we detail the steps used to analyze machine learning approaches. As shown in Figure 1, this procedure consists of five stages.

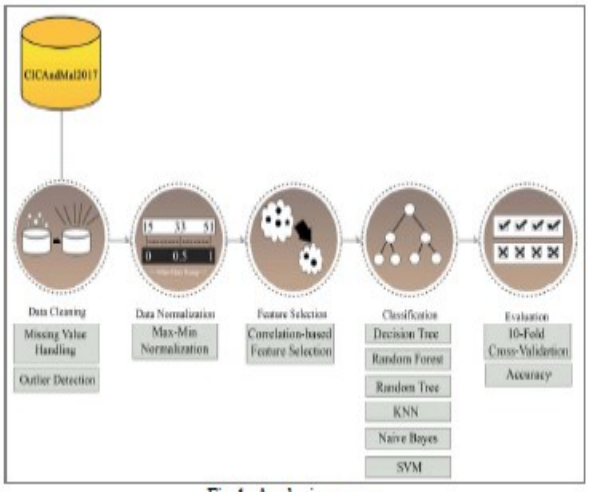



Fig.1. Analysis process

Before data can be classified, it must be cleaned. This stage involves addressing missing values and detecting outliers or noise in order to clean up the data and make it complete. The dataset used to evaluate the classifiers in this study is notable for not having any missing values. We also don't get rid of outliers since they could be useful in some scenarios,

including intrusion and virus detection. Step two involves normalizing the data using the Max-Min normalization technique, which scales the data from 0 to 1. In order to transform v from a range of values between 0 and 1, we use equation 1.

$$v' = \frac{v - min}{max - min} \quad (1)$$

Step three involves feature selection, which involves removing features that aren't relevant to the prediction process. Using the right features selection method before classification improves classifier performance. In this paper, we use the correlation-based feature selection (CFS) method [16], which employs the best first search for each ransomware family separately. Step four involves training seven popular classifiers on the dataset. Step five involves evaluating the classifiers using a 10-fold cross validation method and an accuracy evaluation measure, which is determined by equation 2.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

In this case, TP, FP, TN, and FN stand for the numbers of correct predictions, incorrect predictions, true negatives, and false positives, respectively.

## EVALUATION

We used the Weka program to examine machine learning methods for ransomware detection [17]. The CICAndMal2017 dataset, published in 2018 by Lashkari et.al., was used to assess several categorization algorithms [18]. Since its public release, this research is the only one that we are aware of to use this dataset for ransomware analysis. In order to create an Android malware dataset, Lashkari and colleagues suggested a methodical methodology. The CICAndMal2017 is built with actual cellphones, not emulators. There are 80 network-flow characteristics in this dataset that were extracted using CICFlowMeter [18]. These features include logs, API/SYS calls, phone, and memory statistics for 4 types of malware and 5 kinds of benign traffic. Adware, scareware, ransomware, and SMS malware make up 42 different kinds of malware. The ransomware category, which has ten families, was chosen exclusively as the assessment dataset for this study. Table I displays the details of each family of ransomware.

TABLE I. Details of Ransomware Families in CICAndMal2017

| | Family | Year | # of Records | # of Attributes | Total Captured |
|---|---|---|---|---|---|
| RANSOMWARE | Charger | 2017 | 79090 | 84 | 10 |
| | Jisut | 2017 | 51344 | 84 | 10 |
| | Koler | 2015 | 89410 | 84 | 10 |
| | LockerPin | 2015 | 50618 | 84 | 10 |
| | Pletor | 2014 | 9430 | 84 | 10 |
| | PornDroid | 2016 | 92167 | 84 | 10 |
| | RansomBO | 2017 | 79712 | 84 | 10 |
| | Simplocker | 2015 | 72682 | 84 | 10 |
| | Svpeng | 2014 | 108552 | 84 | 11 |
| | WannaLocker | 2017 | 65402 | 84 | 10 |
| BENIGN | Benign | 2017 | 409761 | 84 | 600 |

In order to examine the machine learning techniques for ransomware detection, we performed two trials. As a preliminary step, we train the classifiers on a single dataset that includes several ransomware samples. Separate datasets representing 10 unique ransomware families are used to train separate classifiers in the second experiment. Twenty percent of each ransomware family's dataset was used to create ten datasets in the first experiment. We then renamed the combined dataset "Ransomware" and combined all data sets into it. When we were satisfied with the ransomware dataset, we balanced it out by adding some benign samples labeled as "Benign" to it. Three tree-based classifiers and the other four classifiers (NN classifier stands for Nearest Neighbor) with feature selection are shown in Fig. 2 for accuracy. We omitted feature selection from the training process of tree-based classifiers because of the inherent feature selection they do.


Fig.2. Classifier accuracies in the first experiment

With an accuracy of 83.37, the Random Forest learning algorithm has the best detection accuracy. Decision Tree has an accuracy of 79.72 and Random Tree 79.68. Based on the findings, tree-based classifiers seem to be the most effective for binary ransomware detection. The second experiment uses separate datasets for training classifiers for each family of malware. We supplemented the collected dataset with an equal number of benign samples for every household. Figure 3 displays the average classifier accuracy for each family in the second experiment, showing results with and without feature selection. When a family's datasets are aggregated, the average accuracy is determined.Feature selection failed for tree-based classifiers in this experiment as it did in the previous one. The outcomes of the remaining four classifiers are shown in Figure 3.

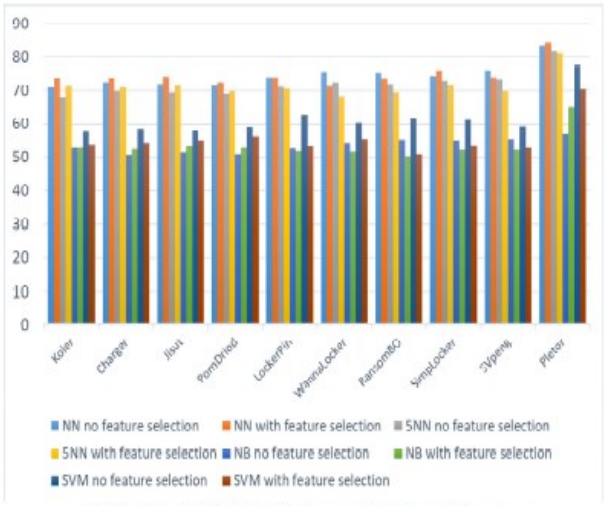
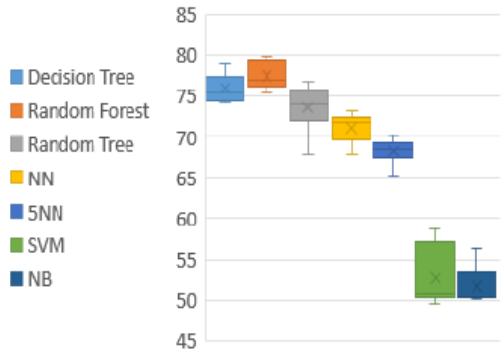
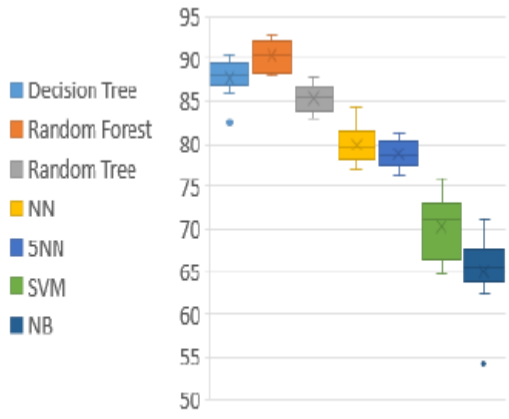



Fig.3. Effect of feature selection on classifiers performance

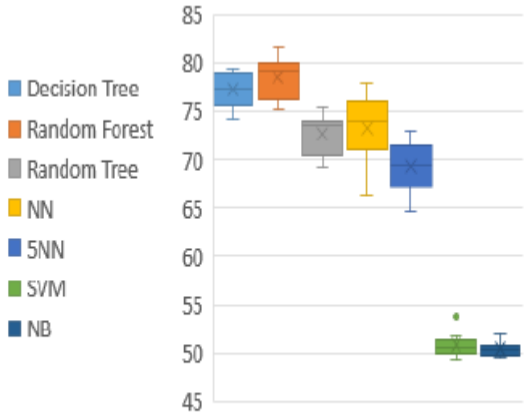
Figure 3 shows that, with the exception of support vector machine (SVM), feature selection increases classification accuracy across the board. For each family's recorded datasets, the box plot diagram of classifier accuracies is shown in Figures 4-a–4-j.



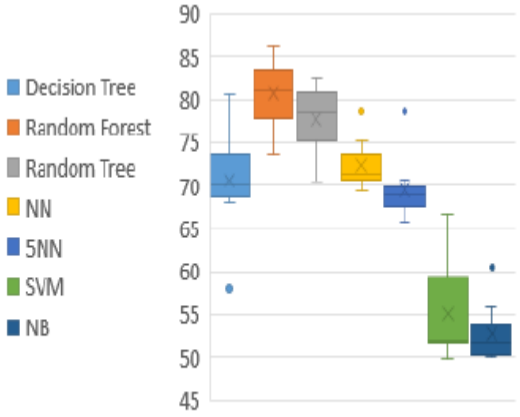

a- Classification accuracy for Charger family



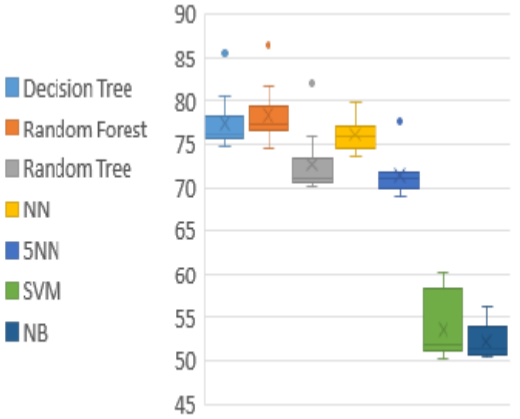

b- Classification accuracy for Pletor family



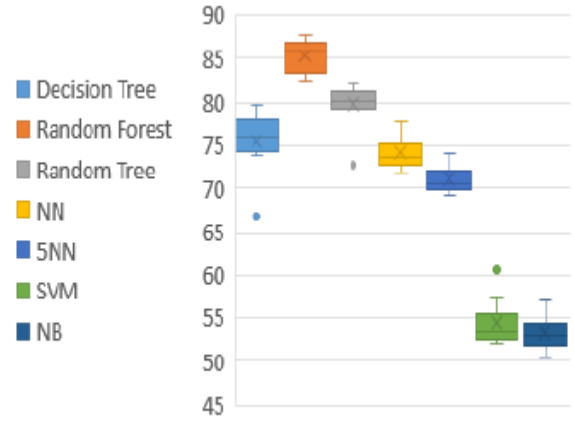

c- Classification accuracy for Jisut family



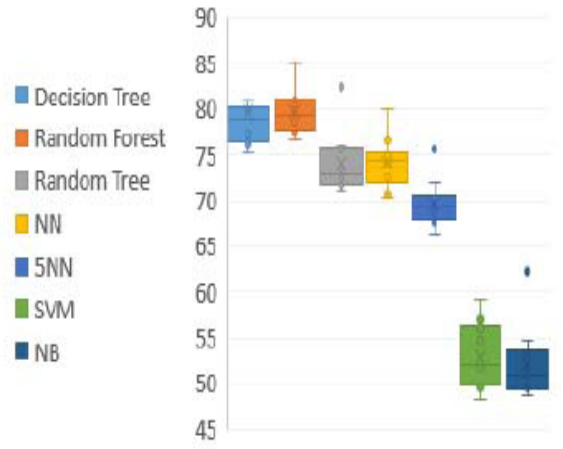

d- Classification accuracy for PornDriod family



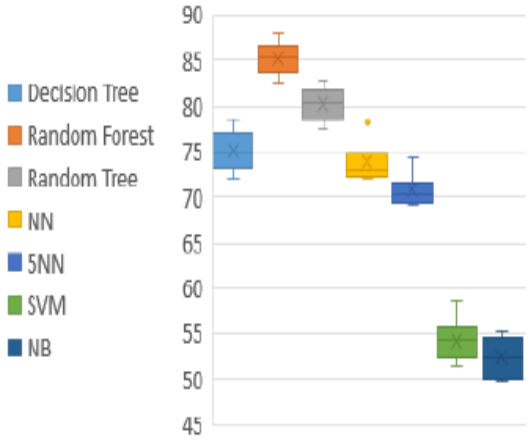

e- Classification accuracy for Koler family



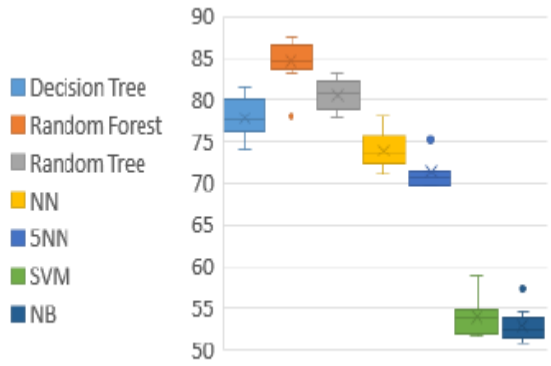

f- Classification accuracy for RansomBO family



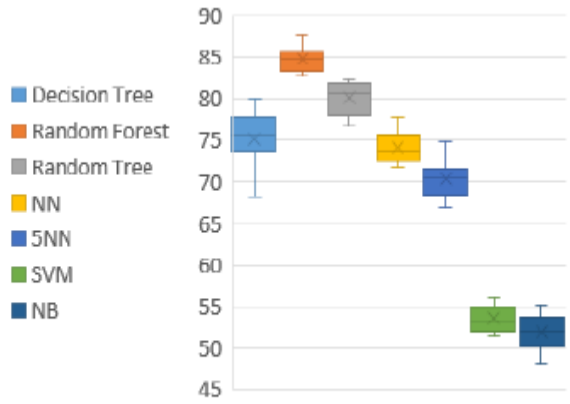

g- Classification accuracy for LockerPin family




h- Classification accuracy for SimpLocker family




i- Classification accuracy for Svpeng family




j- Classification accuracy for WannaLocker family

Fig. 4 . Classification accuracy for each family

Figure 5 shows that across all families, each classifier had an average accuracy. Among the classifiers, Random Forest has the highest average accuracy (0.82), as seen in Figure 5.

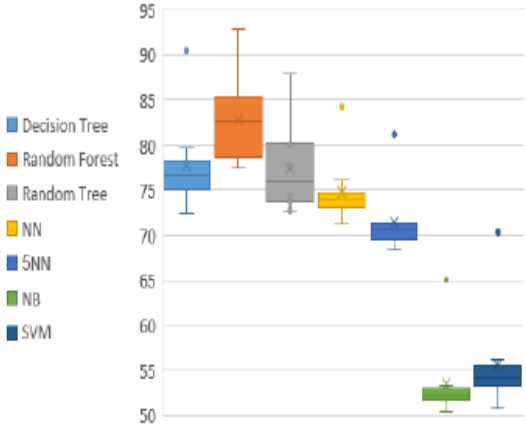



Fig. 5. Average classification accuracy for ransomware families

Table II. Provide an overview of the second experiment's findings. Table II and Figure 2 exhibit very similar findings, suggesting that the classifiers' performance was quite consistent throughout the two tests. At least according to the CICAndMal2017 dataset, ransomware families have some characteristics that may be used to identify them. Therefore, improving classification accuracies is not significantly achieved by dividing various families into separate datasets and then doing feature selection on each dataset separately.

TABLE II. Rank of classifiers for ransomware detection in the second experiment

| Classifier | Rank | Average accuracy |
|---|---|---|
| Random Forest | 1th | 82.80 |
| Decision Tree | 2th | 77.70 |
| Random Tree | 3th | 77.37 |
| NN | 4th | 74.76 |
| 5NN | 5th | 71.36 |
| SVM | 6th | 55.57 |
| NB | 7th | 53.49 |

## CONCLUSION

To determine how well machine learning algorithms work for ransomware detection, we undertook two tests, which are detailed in this article. On the CICAndMal2017 dataset, we used seven different categorization techniques for each trial. In all trials, Random Forest outperformed the other classifiers, proving that tree-based algorithms are effective against ransomware. Our results suggest that classifier performance is not considerably different when trained on each family separately. It would be interesting to explore other feature selection and classification approaches with other common ransomware datasets in future research.

## REFERENCES

[1] J. Ross, Q. Morgan, and K. Publishers, “Book Review : C4.5 : Programs for Machine Learning,” vol. 240, pp. 235–240, 1994.
[2] A. Verikas, A. Gelzinis, and M. Bacauskiene, “Mining data with random forests: A survey and results of new tests,” Pattern Recognit., vol. 44, no. 2, pp. 330–349, 2011.
[3] S. R. Kalmegh, “Comparative analysis of weka data mining algorithm randomforest, randomtree and ladtree for classification of indigenous news data,” Int. J. Emerg. Technol. Adv. Eng., vol. 5, no. 1, pp. 507– 517, 2015.
[4] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” Mach. Learn., vol. 6, no. 1, pp. 37–66, 1991.
[5] G. H. John and P. Langley, “Estimating continuous distributions in Bayesian classifiers,” in Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, 1995, pp. 338–345.
[6] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization, advances in kernel methods,” Support Vector Learn., pp. 185–208, 1999.

[7] K. Cabaj, M. Gregorczyk, W. Mazurczyk, P. Nowakowski, and P. Zorawski, "Network Threats Mitigation Using Software-Defined Networking for the 5G Internet of Radio Light System," Secur. Commun. Networks, vol. 2019, 2019.
[8] D. Sgandurra, L. Munoz-Gonzalez, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," arXiv Prepr. arXiv1609.03020, 2016.
[9] Z. G. Chen, H. S. Kang, S. N. Yin, and S.-R. Kim, "Automatic ransomware detection and analysis based on dynamic API calls flow graph," in Proceedings of the International Conference on Research in Adaptive and Convergent Systems, 2017, pp. 196–201.
[10] T. Aragorn, C. YunChun, K. YiHsiang, and L. Tsungnan. "Deep learning for ransomware detection," IEICE Technical Report; IEICE Tech. Rep., 2016, pp.87–92.
[11] Y. Takeuchi, K. Sakai, and S. Fukumoto, "Detecting ransomware using support vector machines," in Proceedings of the 47th International Conference on Parallel Processing Companion, 2018, pp. 1-6.
[12] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha, "Leveraging machine learning techniques for windows ransomware network traffic detection," in Cyber Threat Intelligence, Springer, 2018, pp. 93–106.
[13] G. Cusack, O. Michel, and E. Keller, "Machine learning-based detection of ransomware using sdn," in Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, 2018, pp. 1–6.
[14] D. Maiorca, F. Mercaldo, G. Giacinto, C. A. Visaggio, and F. Martinelli, "R-PackDroid: API package-based characterization and detection of mobile ransomware," in Proceedings of the symposium on applied computing, 2017, pp. 1718–1723.
[15] N. Andronio, "Heldroid: Fast and Efficient Linguistic-Based Ransomware Detection," M.Sc. thesis. University of Illinois at Chicago, 2015.
[16] M. A. Hall, "Correlation-based Feature Selection for Machine Learning," 1999.
[17] E. Frank et al., "Weka-a machine learning workbench for data mining," in Data mining and knowledge discovery handbook, Springer, 2009, pp. 1269–1277.
[18] A. H. Lashkari, A. F. A. Kadir, L. Taheri, and A. A. Ghorbani, "Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification," in 2018 International Carnahan Conference on Security Technology (ICCST), 2018, pp. 1–7.