Edge Computing Ransomware Detection in IoT using Machine Learning

Tejesh Radhakrishna, Nahid Ebrahimi Majd
Department of Computer Science and Information Systems
California State University San Marcos, United States
radha003@csusm.edu, nmajd@csusm.edu

***Abstract*— The resurgence of ransomware has emerged as a pressing security threat in computer networks and Internet connected machines and IoT devices. To address this challenge, accurate ransomware detectors are required to automatically detect and block the malicious traffic. Most ransomware detectors only detect whether the traffic is benign or ransomware. However, detecting the family of ransomware would be greatly useful to promptly eliminate or mitigate its destructive effects. To tackle this issue, we propose machine learning models that accurately detect each ransomware family. Our models aim to detect the ransomware network traffic and thwart it at the network edge before it enters the network. Considering that ransomwares directly work with the memory dump and file system, the information extracted from the operating system's functions on the memory dump is very useful to detect a ransomware attack. However, that information could be collected only when the ransomware has already infected the device and is actively disrupting the file system. In our research, we propose a framework that blocks the ransomware at the network edge. This restricts our research to using a dataset that extracts network traffic features with no access to the device's operating system's functionalities. An edge computing intrusion detection system is also beneficial for resource contained network devices, such as IoT, which have limited computational resources and cannot dynamically analyze the network traffic and run a strong intrusion detection system. We worked on CICAndMal2017 dataset and proposed a feature selection-based framework along with different machine learning models. We also applied a data augmentation technique to the training set to strengthen the data used to build our models. We extensively studied our proposed framework. Our experimental results demonstrated that chi-square feature selection with Random Forest and XGBoost models surpass other models and the state of the art in detecting ransomware classes.**

Keywords*— *Ransomware Detection; Network security; Feature Selection; Machine Learning;

I. INTRODUCTION

Detecting the malicious traffic emanated from smartphones infected by malwares is a pressing issue as the smartphone devices are widely being used on the Internet. In the first quarter of 2023, 92% of internet users accessed the internet using a mobile phone, and there were approximately 4.3 billion active mobile Internet users [1]. In this quarter, Android maintained its position as the leading mobile operating system worldwide with a market share of 71.4%, and the next one was Apple's iOS with a market share of 27.9% [2]. As the numbers of apps and their users are rapidly growing, the number and scope of smartphone malware infections also grow dramatically.

Ransomware is a type of malicious software (malware). The attacker first gets the control of the victim's filesystem and then encrypts the files and locks the screen. Then, they threaten the victim that they will disclose the victim's personal data and/or permanently block access to their files unless a ransomware is paid in the format of a cryptocurrency like Bitcoin. As the cryptocurrency accounts are in general anonymous, the victim cannot track or identify the attacker's real-life identity. In most cases, even when the victim pays the ransom, the attacker does not provide the crypto key to decrypt the files or recover access to the system file. Despite that, many victims tend to pay the ransom with the hope to recover access to their files, which makes ransomware very profitable for cyber-attackers [3].

There are different families of ransomwares. Most of them require a quick interaction with the user. They start with a phishing email or text message sent to the victim's machine including a link. Once the victim clicks on the link, it downloads the malicious attached file, which then installs the ransomware onto the victim's machine. However, some widespread ransomware families, like WannaCry worm, automatically spread in the network and infect other machines without any user interaction. The main purposes of a ransomware are to encrypt the files and block access to them. However, different ransomwares may also steal, harvest, modify, or upload information [4]. Thus, the network traffic patterns can help detecting ransomwares. Machine learning has been used to develop innovative solutions to combat ransomware.

In our research, we develop an edge computing ransomware detection framework suitable for resource constrained devices, like IoT. Since IoT devices are expected to be cheap, typically their computational powers are limited, which restricts their security solutions. Relocating the computationally expensive security solution from the network device to the network edge could save the entire IoT network from intrusions by detecting and throttling the malicious traffic at the edge.

The main contributions of our paper are as follows.

1. We used the CICAndMal2017 dataset [13]. This dataset has been created by analyzing the network traffic and extracting features on aggregated statistics of both benign and malicious network traffic. We used chi-square feature selection technique to find the features that are strongly associated with ransomware attacks. We analyzed these associations.

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

2. We applied Synthetic Minority Over-sampling Technique (SMOTE) to our training set. This technique generates synthetic samples, which augments the training set used to build the models. Then, we employed several machine learning algorithms and built models using our augmented training set. Our experimental results showed that our Extreme Gradient Boost (XGB) and Random Forest (RF) models outperform other models and surpass the state of the art.

We compare our results with [5] who applied machine learning algorithms to detect the ransomware. They applied correlation-based feature selection (CFS) method [6], which uses the best first search to classify each ransomware family distinctly. Their RF models demonstrated the best results with average accuracy of 82.8%.

Our research investigates the efficacy of various machine learning models to classify each ransomware attack distinctly. In this research, we use the network traffic generated by real Android smartphones network traffic and the features extracted from this network traffic using a standard tool. The metadata, including information on network packets in a flow, is fed to different algorithms to identify suspicious traffic.

The rest of this paper is organized as the following. Section II describes the related work. Section III explains the methodology, including the dataset, the preprocessing, the chi-square feature selection method, and the machine learning models. Section IV presents hyperparameter tunning and analysis. Section V presents the comparisons and discussion. Section VI draws the conclusion and the future plan.

II. Related Work

Ransomware has surged in 2023, threatening companies to either pay immense ransom or lose access to their files. More recent ransomwares threaten the targeted companies to even disclose their clients' private information, which will cause loss of trust among clients and the companies' bankruptcies. The main problems with ransomware are (1) the attacker cannot be traced, and (2) even if the attack target pays the demanded ransomware, there is no guarantee that they gain access to their files or the stolen information will never be disclosed. Thus, the best way to avoid these destructive consequences is to detect the ransomware at the network gateway and do not let the malicious traffic infect the network machines [7]. [8] surveyed recent research on ransomware detection using various techniques including machine learning.

A variety of datasets have been created that contain malware data, including ransomware data. Some of them are API based, extracting information on API calls from a ransomware installed on a machine [9]. Some datasets are image based, which convert the malware binary code to a binary image and classify it using Convolutional Neural Networks (CNN) [10]. Some datasets use manalyzer feature extractor tool, which extracts Portable Executable (PE) parameters of binary files. [11] used a dataset extracted by this tool and developed binary classification machine learning models to detect ransomware.

Some other datasets, which are the focus of our research, are network traffic based, which extract aggregated data from the traffic that the ransomware communicate with the network. Many of these datasets have used emulators to generate malware traffic. However, these datasets suffer from lack of essential information that the network anti-emulators automatically remove [12]. To address this issue, [13] created CICAndMal2017 dataset. They constructed a network including three Android smartphones and installed different malwares including 10 ransomware families on these devices. Then, at the network gateway, they captured the network traffic generated by malwares running on these smartphones. Then, they used the standard CICFlowMeter [14] tool to extract 80 network flow features from the captured traffic. A network flow is a sequence of packets in one communication session that share the same values for source and destination IP and port and protocol. They extracted data for both benign and malicious traffic. In our research, we use this dataset and investigate the ransomware traffic.

[13] that created CICAndMal2017 dataset, studied three machine learning models, RF, KNN, and DT with two feature selection algorithms of CfsSubsetEval with the Best First search method and Infogain with the Ranker search method in Weka data mining tool to classify each malware category. Their best models were RF binary classifiers, which classified each of the 4 malware categories with an average 85.8% precision. Their introduced dataset has been used in a variety of research studies to analyze malware network traffic. We review the research that specifically studied ransomware using this dataset.

[15] and [16] used the same dataset to study semi-supervised learning to classify ransomware families. In supervised learning, all instances are labeled, and the model learns the patterns of each class via the instances labeled to be in that class. In unsupervised learning, the instances are not labeled, and the model should find out the patterns in the data and label them. In semi-supervised learning, a small number of instances are labeled, and the model should learn the patterns of each class via that small number of labeled instances. [15] employed different feature selection methods, and among them they got the highest accuracies for chi-square and oneR. They created distinct models for each ransomware family. Their models achieved the highest accuracy of 88.5% for Pletor family and 76-81% accuracies for other families. [16] also studied this problem and improved accuracies via hyperparameter tunning.

Another study that used this dataset [17] exploited particle swarm optimization (PSO) for feature selection and then created DT and RF models. They studied two scenarios: (1) a binary classification to detect whether the traffic is benign or ransomware. (2) a family classification to detect the ransomware type. Their scenarios are different from ours, and we cannot compare our results with their research.

III. Methodology

A. Dataset description

We used the CICAndMal2017 dataset [18] created by Canadian Institute for Cybersecurity for our research because it is one of the only available network traffic based (rather than API-call based) ransomware detection datasets that has a substantial number of samples for each ransomware family as well as well-defined features extracted from network traffic using standard tools that clearly describe different aspects of a malicious or benign traffic flow that two machines communicate during a network session. This dataset consists of data for four malware categories, namely Adware, Ransomware, Scareware, and SMS Malware. The creators of this dataset released another version in 2019 as well. The 2019 version includes more malware categories but the ransomware part of that remained the same. They released another version in 2020 for dynamic behavioral analysis of malwares. For that type of analysis, they needed to use emulators rather than real smartphones. As we discussed earlier, that is not suitable for a real IoT network traffic analysis as the network anti-emulators will remove part of traffic that is essential for malware detection. Thus, the 2017 version is the best choice for our research.

Our selected dataset includes data for benign traffic and 10 ransomware families. For each family, they installed at least 10 ransomware samples on the smartphones and captured the malicious traffic communicated between the smartphone and the network. Then, they used a standard tool [14] to extract 84 standard network-flow traffic features from the captured traffic. The description of all flow features extracted by the tool is available at Canadian Institute For Cybersecurity GitHub website [19]. The numbers of samples and instances and the portions of ransomware instances in each family are given at Table 1. All instances have been collected in 2017.

Table 1: The Ransomware families and Benign datasets

	Family	Number of instances	Portion of ransomware instances
Ransomware	Charger	39,551	11%
	Jisut	25,672	7%
	Koler	44,555	13%
	LockerPin	25,307	7%
	Pletor	4,715	1%
	PornDroid	46,082	13%
	RansomBo	39,859	12%
	Simplocker	36,340	11%
	Svpeng	54,161	16%
	WannaLocker	32,701	9%
	Total Ransomware	348,943	100%
Benign	Benign	409,761	

They extracted 84 standard network-flow features from the captured traffic. A flow is a sequence of packets with the same values for 5 features consisting of source IP, destination IP, source port, destination port, and the protocol.

B. Data pre-processing

For each distinct ransomware family, we split the dataset to 80% training and validation sets and 20% test set. Then we randomly selected benign instances twice the number of training and validation instances and added them to the training set. We randomly selected benign instances the same number of test instances and added them to the test set. The benign instances of the test set were selected from the rest of benign dataset. For instance, for Charger family, we created a training and validation set of 0.8*39,551 Charger and 2*0.8*39,551 benign instances and a test set of 0.2*39,551 Charger and 0.2*39,551 benign instances. Then we applied SMOTE augmentation technique on the training and validation sets to get balanced training and validation sets. We created 10 datasets, each containing one ransomware family's instances and instances randomly selected from the benign dataset. We run binary classifications to detect the ransomware family.

The source IP, destination IP, Timestamp, and Flow ID do not include relevant data to predict the class, and we removed these features. At the end, there remained 68 features. [19] presents a complete list of features and their descriptions. We used min-max normalization to transform each value into a decimal between 0 and 1.

C. Feature selection

We applied different feature selection techniques to train the datasets and fed the transformed data to various machine learning algorithms. We got the best performances with chi-square technique. We present our results for chi-square feature selection. We selected 65 out of 69 features with highest chi-square scores. We tested both less and greater numbers of features, however, 65 presented the highest accuracies. The top 11 features with highest chi-square scores are listed in Table 2.

These scores indicate that a ransomware traffic is highly associated with the PSH and SYN flags. The PSH flag indicates that the receiving device should deliver the data to the receiving application asap, and the SYN flag is used to establish a TCP connection. These scores indicate that a ransomware attack frequently generates requests to establish connection with the victim machine and pushes the victim process to execute the ransomware code asap. The next features that are highly associated with the ransomware traffic are the time between two packets in a flow, the idle time in a flow, and the flow duration. The next feature with high chi-square is associated with URG flag, indicating that ransomware traffic urges the victim device to immediately process the ransomware code. The URG flag association is particularly interesting as this flag is rarely set in ordinary network traffic, but the results suggest that frequent URG flags is a strong signal that a ransomware attack is occurring.

Table 2: The top 11 features of the dataset with highest Chi-square values

#	Chi-square (rounded)	Feature	Description
1	151	Fwd PSH flag	Number of times the PSH flag was set in packets travelling in the forward direction
2	151	SYN Flag Count	Number of packets with SYN
3	78	Fwd IAT Max	Maximum time between two packets sent in the forward direction
4	78	Fwd IAT Total	Total time between two packets sent in the forward direction
5	50	Flow IAT Std	Standard deviation time between two packets sent in the flow
6	41	Idle Min	Minimum time a flow was idle before becoming active
7	38	Idle Mean	Mean time a flow was idle before becoming active
8	36	Idle Max	Maximum time a flow was idle before becoming active
9	33	Flow duration	Duration of the flow in Microsecond
10	28	Flow IAT Max	Maximum time between two packets sent in the flow
11	28	URG Flag Count	Number of packets with URG

D. Data Augmentation

To cope with the data imbalance issue of our dataset, we applied SMOTE data augmentation techniques to our training set. This technique generates synthetic samples between each sample of the minority class and its K nearest neighbors.

E. Machine Learning Models

We developed RF, XGB, and KNN and trained them with our datasets. We also trained Logistic Regression and Ensemble Learning models, but the resulted accuracies were low, and we do not present them here.

IV. Hyperparameter Tunning and Analysis

We used grid search method to fine tune the hyperparameters. Our hyper-tunning steps are described in Algorithm 1. Most models demonstrated higher accuracies after hyper-tunning. However, in a few cases, we observed less accuracies after hyper-tunning. The reason is the initial hyperparameters values resulted in overfitting. When the initial model demonstrates unexpected high accuracy, but the tuned model demonstrates much lower accuracy, it indicates that the initial model is overfitted. Thus, we should tune the model to get a reasonable accuracy despite a slight drop in the final tunned model's accuracy. Overfitting occurred in most of our RF models.

Algorithm 1: Hyper-tunning steps

1. Train the model using the train set.
2. Test the model using the test set.
3. Fine-tune the model's hyperparameters with grid search method using the validation set.
4. Retrain the model with the tuned hyperparameters using the combined train and validation sets.
5. Test the model using the test set. Compare the test results of steps 2 and 5. If overfitting has occurred, manually tune the hyperparameters and repeat steps 4-6 until highest accuracy with minimum overfitting is achieved.

Thus, after the initial grid search, we kept turning the hyperparameters for these models to reduce overfitting and get more efficient models. We observed some level of overfitting in our KNN and LR models as well. Our XGB models illustrated the best performance after hyper-tunning with minimum overfitting for almost all ransomware families.

V. Comparisons and Discussion

In this section, we compare the accuracies of our proposed KNN, RF, and XGB models with the best results of the related research [5], [15]. Table 3 illustrates the comparisons. Our experimental results demonstrated that our RF and XGB models outperform our other models and surpass the best models of [5], [15] for all Ransomware families. Our XGB models presented slightly higher accuracies than our RF model for most ransomware classes. Our GXB models present 6-18% higher accuracies than the best models of [5] and 11-21% higher accuracies than the best models of [15].

We conclude that appropriate feature selection, tunning hyper parameters, eliminating overfitting, balancing the training dataset using data augmentation, and selecting appropriate machine learning models improve the efficiency of ransomware classifiers. We presented the best classifiers that accurately detect the ransomware families.

VI. Conclusion and Future Work

In this research, we created machine learning models that effectively detect a ransomware family using the network traffic. We studied different feature selection techniques that choose the features that are strongly associated with the type of traffic to build the models. We got the best results with chi-square feature selection technique, which we presented in this paper. To cope with the data imbalance, we applied SMOTE data augmentation technique to our training set. This technique uses the current instances of a minority class and creates new instances. It provides a more uniform dataset, which could result in higher accuracy in an imbalanced dataset.

Table 3: Accuracy comparison of the proposed models and the best models of [15] and [5]

Class number	Ransomware class	Best model of [15]	Best model of [5]	Proposed KNN	Proposed RF	Proposed XGB
1	Charger	75	77	86	90	91
2	Jisut	76	79	86	92	92
3	Koler	77	77	84	89	91
4	Lockerpin	77	79	88	91	92
5	Pletor	88	90	97	99	99
6	Porndroid	76	81	84	89	91
7	RansomBo	82	86	84	91	91
8	Simplocker	80	85	83	91	92
9	Svpeng	81	84	85	90	91
10	Wannalocker	80	85	84	92	92

We created different machine learning models on the training dataset. We achieved the best results with XGB and Random Forest. We studied the performances of our models. To improve the models' performances, we fine-tuned their hyperparameters using grid search on the validation data. We evaluated the performances of our models for each ransomware family. We observed that in some cases the parameters suggested by grid search are not effective and cause overfitting, so we manually tuned the hyperparameters and chose the best performing combination. Our results indicate that XGB and appropriate hyper tunning along with Chi-square feature selection and SMOTE data augmentation present the best accuracies for almost all ransomware families.

One future direction for our research is creating deep learning models for this dataset. Another future direction is studying the impact of dimensionality reduction on ransomware classifiers. The dimensionality reduction technique reduces the number of features by combining the highly correlated features in one feature.

REFERENCES

[1] Available online, accessed on June 20, https://colorlib.com

[2] Available online, accessed on June 20, https://www.statista.com

[3] S. Kamil S, H.S. Norul, A. Firdaus, O.L. Usman, "The rise of ransomware: A review of attacks, detection techniques, and future challenges," International Conference on Business Analytics for Technology and Security, IEEE ICBATS, 2022, pp. 1-7, doi: 10.1109/ICBATS54253.2022.9759000.

[4] U. Urooj U, B.A. Al-rimy, A. Zainal, F.A. Ghaleb, M.A. Rassam, "Ransomware detection using the dynamic analysis and machine learning: A survey and research directions," Applied Sciences. 2022. doi: 10.3390/app12010172.

[5] F. Noorbehbahani, F. Rasouli, M. Saberi, "Analysis of machine learning techniques for ransomware detection" IEEE ISCISC, 2019. doi: 10.1109/ISCISC48546.2019.8985139.

[6] M.A. Hall, "Correlation-based Feature Selection for Machine Learning," 1999.

[7] M.E. Ahmed, H. Kim, S. Camtepe, S. Nepal, "Peeler: Profiling Kernel-Level Events to Detect Ransomware," European Symposium On Research In Computer Security (ESORICS). 2021. doi: 10.1007/978-3-030-88418-5_12.

[8] A. Alraizza, A. Algarni, "Ransomware detection using machine learning: A survey," Big Data and Cognitive Computing. 2023. doi: 10.3390/bdcc7030143.

[9] M.E. Ahmed, H. Kim, S. Camtepe, S. Nepal, "Peeler: Profiling kernel-level events to detect ransomware," European Symposium on Research in Computer Security (ESORICS), Springer, 2021. doi: 10.1007/978-3-030-88418-5_12.

[10] S. Freitas, R. Duggal, D.H. Chau, "MalNet: A Large-Scale Image Database of Malicious Software," ACM International Conference on Information and Knowledge Management (CIKM), 2022. doi: 10.1145/3511808.3557533.

[11] M. Masum, et al. "Ransomware classification and detection with machine learning algorithms," IEEE CCWC, 2022. doi: 10.1109/CCWC54503.2022.9720869.

[12] L. Xue, Y. Zhou, T. Chen, X. Luo, and G. Gu, "Malton: Towards ondevice non-invasive mobile malware analysis for art," ACM USENIX Security Symposium, 2017.

[13] A.H. Lashkari, A.F. Kadir, L. Taheri, and A.A. Ghorbani, "Toward developing a systematic approach to generate benchmark android malware datasets and classification," IEEE International Carnahan Conference on Security Technology (ICCST), 2018. doi: 10.1109/CCST.2018.8585560.

[14] G.D. Gil, A.H. Lashkari, M.Mamun, and A.A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related features," International Conference on Information Systems Security and Privacy (ICISSP), 2016.

[15] F. Noorbehbahani and M. Saberi, "Ransomware detection with semi-supervised learning," IEEE ICCKE, 2020. doi: 10.1109/ICCKE50421.2020.9303689.

[16] S. Sifat, M.S. Hossain, S.A. Tonny, B. Majumder, R. Mahajabin, H.M. Shakhawat, "Android ransomware attacks detection with optimized ensemble learning," International Conference on Cybersecurity, Cybercrimes, and Smart Emerging Technologies, Springer, 2022. doi: 10.1007/978-3-031-21101-0_4.

[17] M.S. Hossain, N. Hasan, M.A. Samad, H.M. Shakhawat, J. Karmoker, F. Ahmed, K.N. Fuad, K. Choi, "Android Ransomware Detection From Traffic Analysis Using Metaheuristic Feature Selection," IEEE Access, 2022. doi: 10.1109/ACCESS.2022.3227579.

[18] Available online, accessed on June 20, Android Malware Dataset at Canadian Institute for Cybersecurity, https://www.unb.ca/cic/datasets/ andmal2017.html

[19] Available online, accessed on June 20, CICFlowMeter, https://github.com/ CanadianInstituteForCybersecurity/CICFlowMeter/blob/master/ReadMe.txt