International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

A Comprehensive Framework for Early Detection and Mitigation of Ransomware in Enterprise Networks

Sai Charan Madugula
University of Central Missouri

Abstract

Ransomware has evolved as one of the most widespread and devastating cybersecurity threats to organisational networks. It frequently results in serious disruptions to business operations, the loss of data, and financial harm. When it comes to the continually developing strategies employed by ransomware perpetrators, traditional security techniques that are based on signatures and use reactive protection measures are usually insufficient. The purpose of this study is to present a complete architecture for the early identification and mitigation of ransomware attacks in business contexts. Real-time behavioural monitoring, anomaly detection based on machine learning, and deception technologies are all incorporated into the system in order to uncover early signs of penetration. In addition to this, it utilises a multi-layered response strategy, which includes automatic isolation, backup verification, and threat intelligence exchange, in order to limit ransomware and reduce its damage. Using modern ransomware samples, a prototype of the proposed system was tested on a simulated business network. The results showed that the system had a high detection accuracy and a low false positive rate. The findings provide further evidence that proactive and adaptive defence mechanisms are more effective than traditional reactive methods. Through the establishment of a solid basis for business cybersecurity architecture, this framework intends to strengthen the organization's resistance against potential ransomware attacks in the future.

Keywords: *Early Detection, Mitigation, Ransomware, cybersecurity, Networks*

1. Introduction

Ransomware has, in the last several years, grown into one of the most serious cybersecurity risks facing business networks throughout the globe. The sophistication and prevalence of ransomware attacks have grown in the past few years. The hallmark of these attacks is the use of ransomware, which encrypts critical data and then requests money to unlock the keys. Ransomware attacks have increased in frequency by almost 150 percent in the last five years, according to cybersecurity industry reports. Organisations have lost billions of dollars due to ransom demands, system outages, and damaged reputations. Risks to critical infrastructure, healthcare systems, government operations, and supply networks extend beyond the potential for monetary damage that ransomware presents. They have the ability to cause disruptions that go beyond just financial losses. The exploitation of zero-day vulnerabilities and social engineering techniques, the rise of Ransomware-as-a-Service (RaaS) platforms, and the anonymity of cryptocurrency payments are all factors contributing to ransomware's alarming increase in frequency. Conventional security measures like firewalls, intrusion detection systems (IDS), and signature-based antivirus programs sometimes fall short when it comes to early detection of novel or disguised ransomware strains. This emphasises the absolute necessity of a more thorough and proactive approach to detecting and preventing ransomware on corporate networks. We need this strategy right now. Most of the mitigation strategies out now focus on the aftermath of an infection, which can cause irreparable harm. By the time ransomware activity is detected, critical files have already been encrypted, and there are very few options for reacting to the assault. It follows that ransomware protection must include early detection, ideally prior to encryption starting. There are several advantages to using automated mitigation methods, including the possibility of significantly reducing reaction time, limiting the spread of malware, and protecting vital data.

The goal of this research is to put up a comprehensive, multi-layered architecture for detecting and preventing ransomware attacks in corporate settings as soon as they happen. Using a combination of real-time behavioural analysis, anomaly detection driven by machine learning, and deception tactics including honeypots and decoy systems, the proposed framework employs a hybrid detection strategy. The interplay between these parts makes

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

it possible to spot red flags like sudden file encryption, attempts to get unauthorised access, or unusual system activity well in advance of any full-scale attacks. The framework's mitigation module includes a suite of automated reaction techniques, including system quarantine, backup activation, and alert escalation. In addition, the architecture allows for the exchange of threat intelligence to strengthen resilience in all networked corporate environments.

1.1 Detection Based on Signatures

Signature-based detection is the primary method utilised by traditional anti-malware solutions. This method involves comparing files and processes to a database that contains known dangerous code patterns. This technique is fundamentally constrained since it is unable to detect zero-day and polymorphic ransomware strains, despite the fact that it is successful against ransomware variations that have been detected in the past. As an illustration, Symantec (2019) and Kaspersky (2020) have revealed that contemporary ransomware families regularly use obfuscation and code mutation techniques in order to circumvent signature-based technologies. Furthermore, by the time a ransomware signature is discovered and updated in databases, it is possible that the infection has already inflicted harm that cannot be repaired.

1.2 Heuristic and Behavioral Analysis

Heuristic-based detection, which employs established criteria to identify suspicious behaviour, has been adopted by a significant number of researchers as a means of overcoming the limits of signature-based authentication approaches. The quick renaming of files, bulk encryption, and abnormal utilisation of the CPU or disc are some examples of these. CryptoDrop is a system that was introduced by Scaife et al. (2016). It is designed to provide monitoring of behavioural indicators and to halt the process of file change once abnormalities are identified. On the other hand, heuristic techniques frequently experience large rates of false positives, which might make them less useful in contexts that are dynamic and enterprise-oriented.

1.3 Machine Learning-Based Approaches

Research efforts related to ransomware detection have been significantly influenced by recent advancements in machine learning (ML). Machine learning algorithms, both supervised and unsupervised, are trained on system call traces, network traffic, and user activity records to detect anomalies that might indicate ransomware. In their 2016 proposal, Sgandurra et al. used EldeRan to analyse data dynamically and use machine learning classifiers such as Random Forest and Support Vector Machines (SVMs) (2016). Its goal is to accurately distinguish malicious software from legitimate apps. Similarly, Kharraz et al. (2015) looked at how ransomware networks behaved and used that data to build models that could spot malicious messages as they happened.

ML models are susceptible to adversarial assaults and may require enormous datasets that have been labelled in order to be trained, despite the fact that they have produced promising outcomes. In addition, the implementation of these models in real-time production systems presents issues in terms of scalability and the amount of resources that are used.

1.4 Deception Technologies and Honeypots

Honeypots and honeyfiles are two examples of deception-based methods that have gained popularity in the field of ransomware research. An early detection and analysis may be accomplished through the use of these tactics, which entail the creation of phoney files or systems that are intended to attract ransomware. An implementation of Ransomware Trap was carried out by Moore et al. (2017). This system makes use of decoy documents and keeps an eye out for attempts to gain unauthorised access. Despite the fact that these systems are efficient at identifying certain varieties of ransomware, they are susceptible to being circumvented by more sophisticated variations that either postpone execution or check for the existence of monitoring tools.

1.5 Hybrid and Multi-Layered Frameworks

Researchers have been increasingly advocating for hybrid frameworks that incorporate various detection and mitigation strategies. This is because they are aware of the limitations that are associated with single-method approaches. A ransomware defence system that is cloud-based and integrates signature analysis, anomaly detection, and behaviour tracking was proposed by Huang et al. (2021). In the same vein, Al-Rimy et al. (2019) conducted an assessment of the models that were already in existence and emphasised the significance of incorporating dynamic and static analysis with real-time system monitoring. However, when used in big, distributed corporate networks, the majority of these frameworks do not possess the scalability and flexibility characteristics. Additionally, they have a tendency to place a greater emphasis on detection rather than proactive mitigation, which is essential for reducing the negative effects of an attack that is successful.

2. Research Methodology

The technique used to assess the computational metrics and performance of machine learning algorithms used for ransomware detection involves a number of steps, such as data collection, preprocessing, feature engineering, model selection, and experimental implementation. Some of the procedures covered in the method are these.

2.1 Algorithms Used

Logistic regression, or LR for short, is a statistical method for assessing the probability of a given outcome in a dataset that includes at least one independent variable. Its usage in machine learning models for classification applications, such fraud detection, is widespread. Using a logistic function, LR technique models the relationship between the independent factors and the target variable. E. Duman and Y. Sahin (2011). To create a structure like a tree, the Decision Tree (DT) method recursively splits the dataset according to the characteristics. A key formula in DT is the impurity measure, which might be the Gini index or the information gain (entropy). In order to solve classification and regression issues, DT aims to build a tree that is very good at categorising instances based on the values of their characteristics. The probabilistic ML method known as Naive Bayes (NB) is founded on the Bayes theorem. Not only is it well-known for its efficiency and usability, but it is particularly well-suited for applications that include classification. Y. Sahin and E. Duman(2011) It is an unfounded assumption to assume that the attributes are conditionally unrelated to the class label. After receiving an input vector x, the ΘB method uses the given information to determine the probability of each class and then produces a prediction about the class label y. The Random Forest (RF) approach is used for classification and regression applications. The method relies on random sampling and decision trees. With the goal of improving its overall accuracy, it builds many decision trees during training and then selects the one with the best forecast. Adaptive boosting, or AdaBoost as it's more often known, is a method used in ensemble learning that takes several inadequate classifiers and merges them into a single, more effective one. Instances and classifiers are given weights by the algorithm, which is then used to alter those weights depending on the classification errors. To forecast the value of the class label y, the AdaBoost method takes an input x and uses it. A large number of poor learners had their guesses aggregated to arrive at this forecast. The ultimate decision on classification is based on a weighted sum of the predictions produced by weak classifiers. In an iterative fashion, AdaBoost improves the performance of weak classifiers by giving more weight to misclassified instances. Weak data classifiers' predictions are weighted to get the final prediction. When it comes to handling statistical problems with regression and classification, Extreme Gradient Boosting (or XGBoost) is your go-to gradient-boosting method. It integrates regularisation and gradient boosting for the goal of attaining high performance. The approach builds decision trees iteratively while optimising the goal function through updates to the leaf scores. To further enhance its overall performance, XGBoost also adds concepts like feature importance and early pausing.

2.2 Dataset

For the purpose of this study, the UGRansome dataset, which is an important instrument for the detection of ransomware threats, was utilised. M. Tokmak,(2022). A number of ransomware varieties that had not been described in any other datasets before are included in this particular dataset, which makes it unique. It spans a spectrum of known ransomware families, including Locky and CryptoLocker, along with the legendary WannaCry, and expands to cover sophisticated persistent cyber threats. A total of 207,533 samples are included in the collection. As can be seen in Table I, each sample was distinguished by fourteen different characteristics, which together offered a comprehensive illustration of the characteristics of the file. After careful consideration, this dataset was chosen because of its large sample size, which enables machine learning models to be trained and tested in an efficient manner. In addition to this, the clearly defined characteristics made it easier to get relevant insights from the data. When compared to other datasets, this particular dataset offers a number of valuable advantages. B. A. S. Al-rimy et al. (2020) Firstly, it is superior to other ransomware datasets in terms of sample size, which guarantees that the training and assessment of the model will be accurate. Secondly, in contrast to collections of malware that are more general in nature, this particular collection is centred on ransomware, which is in line with the particular goals of this research. In the third place, as compared to datasets that were designed for signature-based detection or dynamic analysis, this particular dataset was more appropriate for the machine learning-based technique that was utilised in this research.

Table I. Dataset Features Description

Feature	Description
Time	The timing of network assaults is indicated via a quantitative column with integers.

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Protocol	The network protocol, such as TCP or UDP, is represented by a qualitative or categorical column.
Flag	SYN, ACK are examples of qualitative or categorical columns that show the status of the network connection.
Family	The network intrusion category is described in a qualitative/categorical column.
Clusters	Clusters or groupings of events are indicated by integers in the quantitative column.
Seed Address	Ransomware attack links structured in a qualitative/categorical column
Exp Addres	The initial links to ransomware attacks are shown in the qualitative/categorical column.
BTC	Values associated with Bitcoin transactions in assaults are stored in a numerical column.
USD	Column with numerical values representing attack-related monetary losses in USD.
Net flow Bytes	A numerical column displaying the number of bytes exchanged in the network flow.
IP address	The IP addresses linked to network events are presented in a qualitative column.
Threats	The type of intrusions or threats is represented by the qualitative column.
Port	The events table has a quantitative column that shows the number of the network port.
Prediction	This variable is meant to be a goal. The SS, Anomaly (A), and Signature (S) are the three possible results of the prediction model that this qualitative/categorical column indicates.

2.3 Data Preprocessing

After looking into it, we found that the dataset was free of duplicates and missing values. The research may then focus on obtaining the most relevant information from the data without having to worry about time-consuming preparation. Two attributes, "Name" and "md5", were found to have just a minimal quantity of data throughout the classification procedure. This led to the removal of these attributes from the final dataset. We didn't need to encode or change anything because the rest of the properties were just numbers.

2.4 Dataset Split

In order to create a training set and a test set, we used the train_test_split function from the Scikit-learn package. This method was used to randomly split the dataset into two halves while keeping the original class distribution intact. So, out of a total of 43,740 samples, 70 percent were used for the trained set. A total of 18,745 samples, representing the remaining 30%, made up the test set. No separate validation set was generated as a consequence of using 5-fold cross-validation. In addition to allowing for a more thorough assessment of the models' performance, this approach also served to mitigate the impact of any biases introduced by the data split.

2.5 Performance Metrics

It was shown that many quantitative measures may be used to assess the models' projection performance. The overall success rate of a model's predictions is a good indicator of its accuracy:

$$\text{Accuracy} = \frac{\text{Total Number of Predictions}}{\text{Number of Correct Predictions}} \qquad (1)$$

Looking at the percentage of accurate predictions that came true is one approach to gauge accuracy.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (2)$$

Recall is defined as the proportion of true positives that were correctly predicted.

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (3)$$

To find a happy medium between the two measures, we take the harmonic mean of the recall and accuracy scores and use it to get the F1-score. At one, it's the best, but at 0, it's the worst.

$$F1 = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \qquad (4)$$

One common performance metric for evaluating a classification model's discriminatory power across multiple thresholds is the Receiver Operating Characteristic Area Under the Curve, or ROC AUC. We calculate the ROC Area Under the Curve (AUC) based on the True Positive Rate (sensitivity) and the False Positive Rate. Based on the ROC AUC formula:

$$\text{ROC AUC} \int_0^1 \text{TPR d(FPR)} \qquad (5)$$

You may think of the area under the ROC curve as an approximation of the area under the curve for ROC analysis

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

(AUC). By plotting the True Positive Rate (TPR - sensitivity) vs the False Positive Rate (FPR) at different classification thresholds, the ROC curve is generated.

$$\text{ROC AUC} \approx$$
$$21 \sum_{i=1}^{N-1} (\text{TPR}_i + \text{TPR}_i + 1) \times (\text{FPR}_i + 1 - \text{FPR}_i) \qquad (6)$$

The True Positive Rate (TPRi) and False Positive Rate (FPRi) at each individual threshold are denoted by N and FPRi, respectively, in this equation, where N is the total number of thresholds.

In practice, the ROC AUC values could range between zero and one. The magnitude of the score demonstrates how well the positive and negative classifications are distinguished. This model's efficacy is shown by the ROC area under the curve (AUC). When it's 0.5, the model is acting randomly, and when it's 1.0, it's doing a fantastic job at classifying. When evaluating a classification model's efficacy, binary classification processes look at the Matthews Correlation Coefficient (MCC). It takes into account TP, TN, FP, and FN to make sure everyone is on equal ground, regardless of class imbalance. Regarding MCC:

$$\text{MCC} = \frac{\text{TP}\times\text{TN} - \text{FP}\times\text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \qquad (7)$$

A perfect forecast is represented by a value of +1, a prediction that is no better than a random one by a value of 0, and a total discrepancy between the two by a value of -1. Greater-than-average MCC values are often indicative of superior classification capacity.

3. RESULTS AND DISCUSSION

The models were evaluated using a method that is known as five-fold cross-validation, and the average performance across all folds is presented in the following table.

3.1 Performance Metrics

In order to provide a major insight into the capabilities of a range of machine learning algorithms to protect against ransomware attacks, the findings, which can be seen in Table II and Figure 1, are presented.

Table 2. Analysis Of Performance Score

Algorithm	Accuracy	Precision	Recall	ROCAUC	MCC	F1
LR	72.46%	72.55%	72.46%	86.31%	57.55%	72.46%
DT	99.42%	99.42%	99.42%	99.59%	99.10%	99.42%
NB	71.92%	73.56%	71.92%	86.83%	58.11%	71.46%
RF	99.37%	99.37%	99.37%	99.99%	99.02%	99.37%
Ada Boost	87.93%	87.96%	87.93%	85.15%	81.32%	87.94%
XG Boost	99.48%	99.48%	99.48%	100.00%	99.19%	99.48%

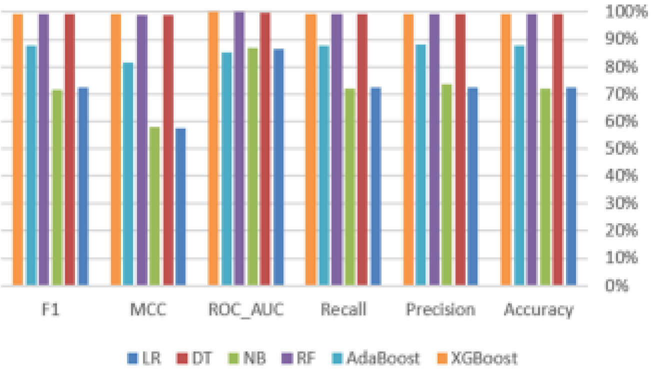
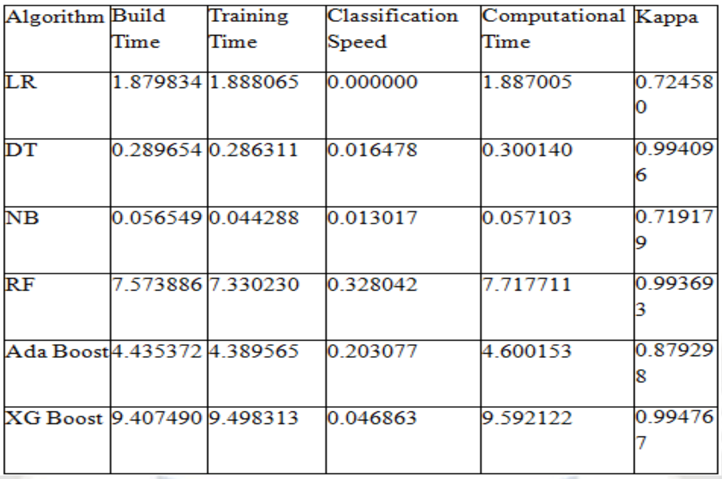


Fig. 1. The appraisal of performance.

The LR is able to distinguish between classes to a reasonable degree for the most part, as it displayed decent levels of accuracy, precision, recall, and F1 scores (around 72%), while having a relatively high ROC AUC score. Its minimal MCC also suggests that its quality estimates are reasonable. With near-perfect ROC AUC values, DT and XGBoost outperformed the competition across the board in terms of accuracy, precision, recall, F1, and performance. That they are able to correctly classify ransomware goods is a direct result of this. The fact that NB's performance metrics were lower than DT and XGBoost but similar to LR's might indicate that it struggles with this specific dataset or type of task. Aside from providing excellent performance, RF also showed performance on par with DT and XGBoost. Given that RF is an ensemble method that often does well in classification tasks, this should not come as a surprise. Compared to RF, DT, and XGBoost, AdaBoost's results were adequate but not exceptional. There were three types of ensembles: the high-performing AdaBoost, the moderate LR, and the more modest NB.

Table 3. Efficiency in Computerised Systems



Algorithm	Build Time	Training Time	Classification Speed	Computational Time	Kappa
LR	1.879834	1.888065	0.000000	1.887005	0.724580
DT	0.289654	0.286311	0.016478	0.300140	0.994096
NB	0.056549	0.044288	0.013017	0.057103	0.719179
RF	7.573886	7.330230	0.328042	7.717711	0.993693
Ada Boost	4.435372	4.389565	0.203077	4.600153	0.879298
XG Boost	9.407490	9.498313	0.046863	9.592122	0.994767

3.2 Computational Efficiency

This report includes a thorough evaluation of these algorithms' computational performance. Considerations like as construction time, training time, classification speed, total computing time, and the Kappa statistic are all part of this research. Table III includes further perspectives on the practical implementation of these algorithms. LR's computational time was the longest, taking into account both training and classification speeds, and its Kappa score, which measures how well the predicted and actual classes match, was small. In addition, LR achieved the best accuracy. For instance, DT was evidently fast because to its extremely short construction and computing time. Furthermore, it was highly congruent, as shown by its high Kappa score. Even though its Kappa value was moderate, NB showed the fastest build and computational speeds, indicating it is the fastest algorithm out of those evaluated. The fact that RF produced many decision trees suggests that it would demand a lot more time for building and calculations. It is possible that the iterative nature of the boosting techniques is the cause of the longer computing durations that AdaBoost and XGBoost experienced; yet, both of these approaches achieved extremely high Kappa values. For this ransomware detection challenge, the algorithms that performed the best were DT and XGBoost. These algorithms achieved a balance between excellent classification performance and great computational efficiency. However, the unique needs of the work might also play a role in determining the algorithm that is used. These criteria include the comparison between the need for speed and the need for accuracy.

4. Conclusion

On a worldwide scale, ransomware attacks continue to be a substantial and ever-evolving danger to company networks. These attacks can result in interruptions to business operations, breaches of data security, and financial losses. Considering that ransomware strains are becoming increasingly complex and frequently avoiding standard detection systems that are based on signatures, the requirement for proactive, real-time defence methods has grown more pressing than it has ever been. Behavioural analysis, machine learning-based anomaly detection, and deception technologies are all integrated into this comprehensive, multi-layered framework that was described in this study. The purpose of this framework is to allow early identification and effective mitigation of ransomware attacks in business contexts. The purpose of the system is to solve the shortcomings of existing detection algorithms, which include delayed responses, high false positives, and limited coverage against emerging ransomware strains. This will be accomplished by integrating real-time monitoring, automated incident response, and adaptive learning. The system's capacity to identify stealthy assaults, which are frequently missed by standard tools, is further strengthened by the use of deception tactics such as honeypots and decoy files. The experimental evaluation of the prototype in a simulated corporate environment revealed encouraging results in terms of the accuracy of detection, the reaction time, and the

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

durability of the system. In addition to this, it brought to light the significance of combining detection with prompt mitigation in order to reduce the amount of time needed for recovery and damage. It is possible that future work will involve the incorporation of sophisticated threat intelligence platforms, the implementation of continuous learning mechanisms to defend against adversarial assaults on machine learning models, and the deployment of the framework in a variety of corporate settings for the purpose of additional validation. Our defence systems need to develop in tandem with the ever-changing strategies employed by ransomware, moving in the direction of more intelligent, adaptable, and proactive cybersecurity infrastructures. In the end, the architecture that has been suggested represents a big step forward in the process of constructing enterprise-level resilience against one of the most severe types of cyber threats in the current digital world.

References

[1] Al-Rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2019). *Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions*. Computers & Security, 74, 144–166. https://doi.org/10.1016/j.cose.2017.11.004

[2] Huang, Y., Liu, H., Xu, K., & Liu, X. (2021). *A cloud-based multi-layered ransomware detection and prevention framework for enterprise networks*. Journal of Network and Computer Applications, 176, 102946. https://doi.org/10.1016/j.jnca.2020.102946

[3] Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). *Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks*. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 3–24). Springer. https://doi.org/10.1007/978-3-319-20550-2_1

[4] Moore, T., Clayton, R., & Anderson, R. (2017). *The economics of online crime*. Journal of Economic Perspectives, 23(3), 3–20. https://doi.org/10.1257/jep.23.3.3

[5] Scaife, N., Carter, H., Traynor, P., & Butler, K. R. (2016). *CryptoDrop: Stopping Ransomware Attacks on User Data*. In IEEE 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS) (pp. 303–312). https://doi.org/10.1109/ICDCS.2016.47

[6] Sgandurra, D., Muñoz-González, L., Mohsen, R., & Lupu, E. C. (2016). *Automated Dynamic Analysis of Ransomware: Benefits, Limitations and Use for Detection*. arXiv preprint arXiv:1609.03020. https://arxiv.org/abs/1609.03020

[7] Symantec Corporation. (2019). *Internet Security Threat Report 2019*. https://symantec-enterprise-blogs.security.com/

[8] Kaspersky Lab. (2020). *Kaspersky Security Bulletin: Statistics of the Year*. https://securelist.com/

[9] Y. Sahin and E. Duman, "Detecting credit card fraud by ANN and logistic regression," in 2011 International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, Turkey, Jun. 2011, pp. 315–319, https://doi.org/10.1109/INISTA.2011.5946108.

[10] A. S. Alraddadi, "A Survey and a Credit Card Fraud Detection and Prevention Model using the Decision Tree Algorithm," Engineering, Technology & Applied Science Research, vol. 13, no. 4, pp. 11505– 11510, Aug. 2022, https://doi.org/10.48084/etasr.6128.

[11] M. Wa Nkongolo, "UGRansome Dataset." Kaggle, https://doi.org/10.34740/KAGGLE/DSV/717254 3.

[12] M. Tokmak, "Deep Forest Approach for Zero-Day Attacks Detection," in Innovations and Technologies in Engineering, S. Tasdemir and I. Ali Ozkan, Eds. Istanbul, Turkey: Eğitim Yayinevi, 2022.

[13] B. A. S. Al-rimy et al., "Redundancy Coefficient Gradual Up-weightingbased Mutual Information Feature Selection technique for Cryptoransomware early detection," Future Generation Computer Systems, vol. 115, pp. 641–658, Feb. 2021, https://doi.org/10.1016/j.future.2020.10. 002.