P-ISSN: 2715-2448 | E-ISSN: 2715-7199
Vol.5 No.2 June 2024
Buana Information Technology and Computer Sciences (BIT and CS)

Ransomware Detection Using Machine Learning Algorithm

Olaniyi Abiodun Ayeni [1]*, Ibitola Elizabeth Adejumo [2]

Department of Cyber Security, School of Computing, Federal University of Technology, Akure Nigeria*
Department of Academic Planning, ICT, University of Medical Sciences, Ondo State Nigeria.
E-mail: oaayeni@futa.edu.ng[1]*, iadejumo@unimed.edu.ng [2]

Abstract

With the advent and subsequent explosion of the internet, global connectivity has been achieved, and is on the rise. This provides a host of advantages such as connectivity and communication, information broadcast and transmission, amongst others. This however introduces a new set of challenges: the safety and protection of these communication channels amongst them. Information has always been power, and the widespread mature of information only results in the widespread attempts to procure it, sometimes via illegal channels. In view of this, this research aims at detecting Crypto-ransomware and locker ransomware. Data was collected from an open repository and cleaned. The cleaned data was then split into tests, train sets and validation which was used to train a number of ML models based on the: Random Forest algorithm, Support Vector Machine (SVM) and Gradient boosting algorithm. Ransomware is one of the well-known ways and frequent use which cyber-attackers use in infecting their victims, either through phishing or drive download. Attackers will create an email pretending to be from a genuine resource and send it to their targeted victims. However, this research illustrated how to combat crypto-ransomware and locker ransomware. Implementing the machine learning algorithm, the system can detect ransomware under 30's, giving computer users over 90% assurance of their system for ransomware free.

Keywords: Gradient Boosting Algorithm, Machine Learning, Random Forest, Ransomware, Support Vector Machine.

I. Introduction

It is possible to access information via the internet and easily recover it for a cheaper cost in our digital world today where information's are stored digitally. Without stress, everything is completed effortlessly and efficiently. Digitalization has increased computer users' quality of life. But every pillar has two sides, as the saying goes. If used as a whole, digitization has reduced crime since technology has made tasks easier to complete and requires less paperwork. However, it creates a security concern for a person's private and sensitive data information. There are numerous thefts and cyberattacks that have occurred which include viruses, spyware, malware, Trojans, phishing, and intruders [1]. Ransomware is a theft which is a kind of infection that can be hard to recover from when being spread. Important files and data on the user's computer system are corrupted as a result of this ransomware.

Ransomware is a kind of malicious malware where the attackers encrypt your file and make it inaccessible to the owner, which spreads more widely and gets more sophisticated every day [10]. Every system in the network today is susceptible to attacks by online criminals. Now that automated technologies are more sophisticated, attackers have access to them, and new threats appear almost instantly. This makes it possibly challenging to maintain proper cybersecurity. Malicious software is one of the biggest concerns in the digital world, and sadly, the problem is getting worse day by day [4].

Manabu et al. (2019) stated that with the quick expansion in Internet of Things (IoT) devices, cyber-physical systems mobile devices, and the cloud services, there has been a surge in extensive cyberattacks on businesses and governmental sectors. Specifically, ransomware is a type of software that prevents victims from accessing or making use of their systems and files until a ransom is paid.

The current level of cyber security today is an ongoing process that entails gathering and comparing millions of data points across all of the personnel and infrastructure. It is fairly obvious that relying only on humans would not be sufficient, there is a need for machine learning support in order to identify trends and foresee potential security risks in massive data sets.

Ransomware can be classified into three categories.

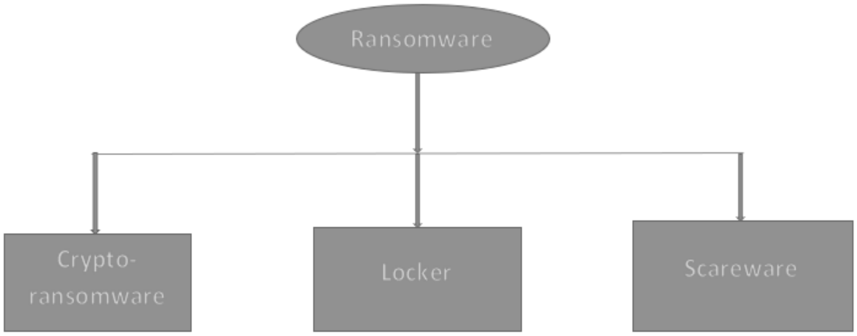


Fig 1. Types of ransomware

Crypto-ransomware is a type of ransomware that encrypts some vital files in the computer system making the user not able to access files or make use of the computer, for the user to retrieve the files in the system, then a ransom message will be passed to the computer user or the victims demanding payment before the user can retrieve the file or information back which can either be retrieved or not after payment and a short period of time will be given for the payment of the ransom. The attackers get their ransom by holding the vital files hostage and this ransom request is through a means like Bitcoin. An example of this ransomware is Wannacry [7]. Crypto-ransomware can be further being sub-divided into three which are:

a. Symmetrical Crypto-ransomware
b. Asymmetrical Crypto-ransomware
c. Hybrid Crypto-ransomware.

This is the type of ransomware that infects the user’s computer and blocked the user out completely, preventing the user from accessing their files on the computer. Even some parts of the computer can be blocked also the keyboard. The attacker thereafter demands a ransom to unblock the computer and limited access will be given to the user to communicate with the attacker until the ransom is paid, the computer will be unblocked. Despite the disruption caused by ransomware, the computer user can still easily be retrieved by removing the disk from the compromised system and placing it on a well-cleaned system [8]. The locker ransomware process is as follows.

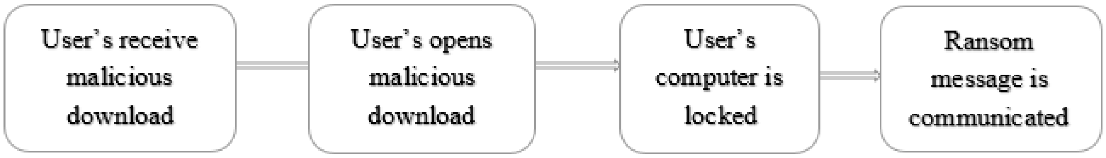

User’s receive malicious download
User’s opens malicious download
User’s computer is locked
Ransom message is communicated

Fig 2. Locker ransomware process

Scareware is ransomware with a tricky technique or a false message used to fake computer users in order to convince users to download harmful software or ransomware that can encrypt data and demand payment. This kind of ransomware post no danger to the victim. Attackers take advantage of the fear of the users to attack the victims [8].

II. Review of Related Works

The work of [6], Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms aims to conduct Some analyzing and selecting the most relevant and non-redundant dynamic features for identifying encryptor and locker ransomware from goodware, generating JSON files with dynamic parameters using a sandbox through experiments with encryptor and locker ransomware combined with goodware, and applying the dynamic feature dataset to obtain models with machine learning algorithms.. A dynamic features dataset is generated and made public. This method made use of machine learning methods, static and dynamic ransomware analysis, and a dataset derived from the created JSON files. At last, a dataset was created that included traits taken from decent software and the dynamic aspects of both locker and encryptor ransomware. However, a dataset was developed. A dataset was developed. The study's dataset contains relevant and lightly correlated features linked to ransomware that is created in runtime.

Eduardo et al. (2022), presented Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic. This focuses on offering an algorithm validation through an analysis of the false positive rate and the volume of user file data that the ransomware could encrypt before being discovered through deep learning and machine learning (Neural Network Model Optimization), as well as model validation through the use of various file-sharing protocols. While the malware is reading and writing files to a network-shared disk, the research finds crypto-ransomware. Provide a tool for detecting crypto-ransomware that is based on the examination of encrypted network traffic in situations involving file sharing. To do this, capabilities for extracting and filtering data that can differentiate between ransomware activity and innocuous, high-activity traffic must be used.

Meanwhile, this system detects crypto-ransomware while the malware is reading and writing files in a network-shared volume with a high false positive. Computers and mobile operating system were not considered.

Samah et al. (2019), presented Ransomware Detection System for Android Applications. This study suggested a static analysis mechanism for locating Android ransomware programs. Based on the calls made by API packages as a leading indicator of harmful behavior, API-RDS focuses on identifying ransomware with high accuracy before it damages the user's device. Analyze the most recent approaches to ransomware detection for Android devices by collecting information, suggesting an API-based system (API-RDS), evaluating API-RDS, and finally providing API-RDS services. This dataset includes information from API-Packages calls, Android ransomware, and innocuous Android datasets. For the purpose of identifying Android ransomware apps, the research offers a static analysis paradigm. It also creates a unique and current dataset that includes recent clean apps and most of the current Android ransomware families. This labeled reference might be applied by the research community. However, the system only focuses on android application detection which may not be applicable to other application.

SH Kok et al. (2019), in a study titled Ransomware, Threat and Detection Techniques: A Review This essay discusses the most recent methods for detection and offers a comprehensive overview of the threat posed by ransomware. provided a thorough description of the steps involved in a ransomware attack and their traits, which can be used as a foundation for further ransomware study. Data from static and dynamic assessments were combined to create a hybrid algorithm for approach. But the it is a review work and no model for a ransomware attack has been developed. The researcher has suggested that in the next research, a model to identify ransomware attacks be developed and that hybrid algorithms be used in place of a single one.

Subash et al. (2019), proposed A Multi-Level Ransomware Detection Framework using Natural Language Processing and Machine Learning. The researchers presented a multi-level big data mining system that combines methodologies from machine learning, natural language processing (NLP), and reverse engineering. Detector engine, Action engine, Passive analyzer, Function call tracker, Assembly instruction tracker, DLL tracker, and six other components are used in Natural Language using machine learning. The open source malware repository Zoo and Virus Total were two of the sources from which the dataset was gathered. This study created a framework for multi-level analysis using DLLs, function calls, and assembly instructions while taking advantage of machine learning classifiers and NLP schemes. It also investigated the differences in n-gram sequences for ransomware binary samples at the multi-level, which helped to create a useful feature database that increased the detection rate at various levels. 98.59% is the maximum detection accuracy for n-gram TF-IDF at N=3, and 97.13% is the second-highest at N=2. Nevertheless, the researcher admitted that performance testing between the research framework and NLP schemes and machine learning classifiers was not done. Instead, a framework of multi-level analysis was built using DLL function calls and assemble instructions.

III. Method

In order to improve the detecting performance of ransomware, the architecture of the suggested ransomware detection system was designed to improve detecting capabilities, the system incorporates machine learning techniques. The goal of this all-encompassing approach is to give enterprises a strong and flexible defense against the constantly changing threat landscape that ransomware attacks is present.

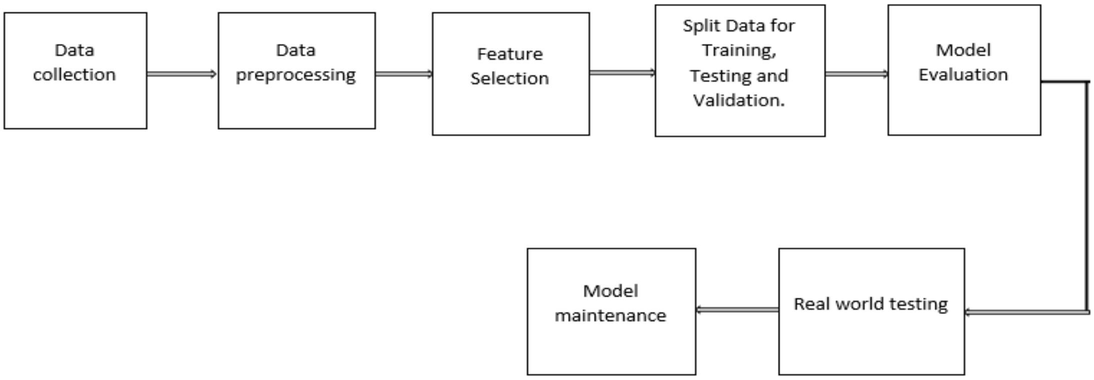
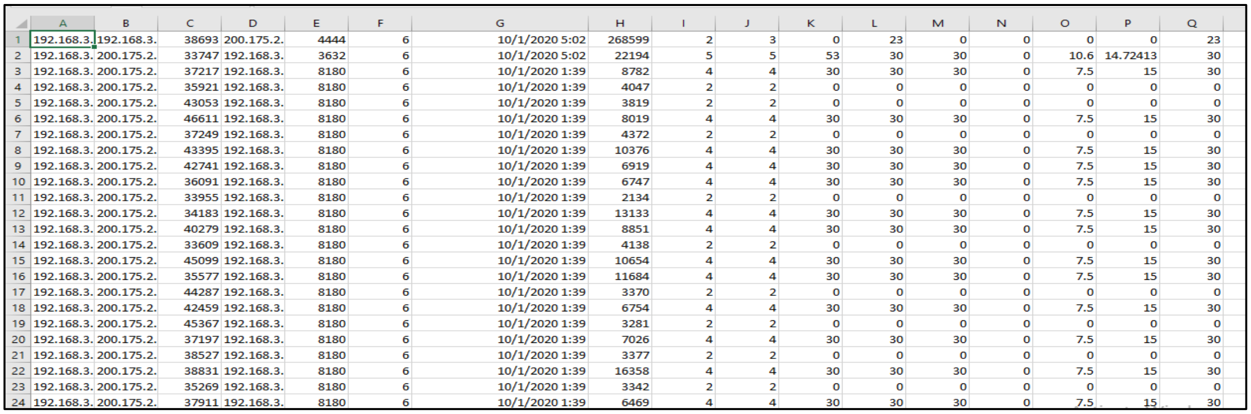


Fig 3. System Architecture

A. Data collection

The dataset of ransomware attack instances was obtained from Kaggle.com in Excel format. This dataset includes both benign and malicious samples, covering various types of ransomware. Initializing threshold settings, outliers will be checked by comparing the distance of the closest data point to the nearest cluster identification and identifying those that are outliers in our dataset.

The dataset represents real-world scenarios and contains features relevant to ransomware detection, such as file characteristics, network traffic patterns, and behavioral indicators. The dataset contains 143573 rows and 84 columns.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	192.168.3.	192.168.3.	38693	200.175.2.	4444	6	10/1/2020 5:02	268599	2	3	0	23	0	0	0	0	23
2	192.168.3.	200.175.2.	33747	192.168.3.	3632	6	10/1/2020 5:02	22194	5	5	53	30	30	0	10.6	14.72413	30
3	192.168.3.	200.175.2.	37217	192.168.3.	8180	6	10/1/2020 1:39	8782	4	4	30	30	30	0	7.5	15	30
4	192.168.3.	200.175.2.	35921	192.168.3.	8180	6	10/1/2020 1:39	4047	2	2	0	0	0	0	0	0	0
5	192.168.3.	200.175.2.	43053	192.168.3.	8180	6	10/1/2020 1:39	3819	2	2	0	0	0	0	0	0	0
6	192.168.3.	200.175.2.	46611	192.168.3.	8180	6	10/1/2020 1:39	8019	4	4	30	30	30	0	7.5	15	30
7	192.168.3.	200.175.2.	37249	192.168.3.	8180	6	10/1/2020 1:39	4372	2	2	0	0	0	0	0	0	0
8	192.168.3.	200.175.2.	43395	192.168.3.	8180	6	10/1/2020 1:39	10376	4	4	30	30	30	0	7.5	15	30
9	192.168.3.	200.175.2.	42741	192.168.3.	8180	6	10/1/2020 1:39	6919	4	4	30	30	30	0	7.5	15	30
10	192.168.3.	200.175.2.	36091	192.168.3.	8180	6	10/1/2020 1:39	6747	4	4	30	30	30	0	7.5	15	30
11	192.168.3.	200.175.2.	33955	192.168.3.	8180	6	10/1/2020 1:39	2134	2	2	0	0	0	0	0	0	0
12	192.168.3.	200.175.2.	34183	192.168.3.	8180	6	10/1/2020 1:39	13133	4	4	30	30	30	0	7.5	15	30
13	192.168.3.	200.175.2.	40279	192.168.3.	8180	6	10/1/2020 1:39	8851	4	4	30	30	30	0	7.5	15	30
14	192.168.3.	200.175.2.	33609	192.168.3.	8180	6	10/1/2020 1:39	4138	2	2	0	0	0	0	0	0	0
15	192.168.3.	200.175.2.	45099	192.168.3.	8180	6	10/1/2020 1:39	10654	4	4	30	30	30	0	7.5	15	30
16	192.168.3.	200.175.2.	35577	192.168.3.	8180	6	10/1/2020 1:39	11684	4	4	30	30	30	0	7.5	15	30
17	192.168.3.	200.175.2.	44287	192.168.3.	8180	6	10/1/2020 1:39	3370	2	2	0	0	0	0	0	0	0
18	192.168.3.	200.175.2.	42459	192.168.3.	8180	6	10/1/2020 1:39	6754	4	4	30	30	30	0	7.5	15	30
19	192.168.3.	200.175.2.	45367	192.168.3.	8180	6	10/1/2020 1:39	3281	2	2	0	0	0	0	0	0	0
20	192.168.3.	200.175.2.	37197	192.168.3.	8180	6	10/1/2020 1:39	7026	4	4	30	30	30	0	7.5	15	30
21	192.168.3.	200.175.2.	38527	192.168.3.	8180	6	10/1/2020 1:39	3377	2	2	0	0	0	0	0	0	0
22	192.168.3.	200.175.2.	38831	192.168.3.	8180	6	10/1/2020 1:39	16358	4	4	30	30	30	0	7.5	15	30
23	192.168.3.	200.175.2.	35269	192.168.3.	8180	6	10/1/2020 1:39	3342	2	2	0	0	0	0	0	0	0
24	192.168.3.	200.175.2.	37911	192.168.3.	8180	6	10/1/2020 1:39	6469	4	4	30	30	30	0	7.5	15	30

Fig 4. Sample of ransomware dataset

B. Data preprocessing

The data were cleaned and preprocess the collected data to remove any noise or inconsistencies handling missing data and identify and handle the missing values or data. By removing rows or columns with missing values or data and imputing the missing values. Outliers were identified and handled by removing them with the use of a robust statistical method.

Perform feature selection to extract meaningful features from the raw data. This may involve techniques like dimensionality reduction, feature selection, or transformation. The data was cleaned appropriately, after which particular features were selected.

C. Feature Selection

Feature selection was carried out for ease of modeling. The data was truncated for ease and speed of modeling. Perform required feature selection and dimensionality reduction. Perform model selection based on the algorithm choices outline (i.e., Random Forest, SVM and gradient boosting). Feature selection is a crucial step in machine learning where the goal is to choose the most relevant and significant features from a dataset to build a model. The process involves identifying and selecting a subset of features that contribute the most to the model's predictive power while disregarding irrelevant or redundant ones. This is done to improve model efficiency, reduce overfitting, and enhance generalization to new.

a. Random Forest

An ensemble learning method for applications like categorization and regression is called random forest. Additionally, during the phase, random forest constructs a large number of decision trees and produces a class that represents the mean of the classes, also known as classification, or mean prediction, also known as regression of each of the trees. It is common for random forests to correctly predict their training set. Random Forest is the go-to machine learning algorithm that uses a bagging approach to create a bunch of decision trees with a random subset of the data. A model is trained several times on a random sample of the dataset to achieve good prediction performance from the random forest algorithm. The output of every decision tree in the random forest is pooled to provide the final prediction in this ensemble learning technique. By analyzing the outcomes of each decision tree, the random forest method's final prediction is discovered or by selecting the forecast that emerges most frequently from the decision trees. The training set, the test set and validation are the three subsets that make up the random forest, and it selects some samples from the practice set [5].

Random forest aims at lowering the amount of time needed for learning and classification either to seek to increase accuracy, performance or both. The random first extracts subsamples from the original samples with the aid of the bootstrap resampling technique, then the algorithm categorizes the decision trees and implements a simple vote with the classification's largest vote serving as the prediction's outcome. There are three steps in the random forest algorithm which are:

1. Choose the training set by retrieving training sets from the original dataset employing the bootstrap random sampling technique, making sure that each training set has the same size as the first training set.
2. Develop the Random Forest model by making a classification regression tree for every bootstrap training set. These trees are left untrimmed to generate decision trees that make up the forest.
3. Create simple voting: Decision trees that have been trained in the same manner can be combined to generate the Random Forest. Because each decision tree's training procedure is autonomous, training for random forests can proceed simultaneously, greatly enhancing efficiency Xiang et al. (2019).

b. Support Vector Machine (SVM)

It permits the search for nonlinear decision boundaries using a variety of various kernels, Support Vector Machine can be used to categorize points from a data set in nonlinear decision boundary. Support vector machine basis has four possible values which are sigmoid, linear, polynomial and radial which is called kernel parameter [3].

c. Gradient Boosting algorithm

Gradient boosting is the method that enables gradual construction of an ensemble trees with the aim of reducing a target loss function. Boosting keeps the leaf node labels and the weights in a way that makes handling prediction interpretations simple.

XGBoost is one of the classification methods. The two improvements in XGBoosting over Gradient Descent are its improved periodicity technique and its increased level of sophistication. Gradient boosting retrieves the relative value scores of each attribute following that the boosted tree is built using an effective metric known as feature/importance [2].

D. Model Training and Evaluation

In this research, the model will be trained using 75% of the data, 15 for testing and the remaining 10% will be used for validation. Four criteria will be used to evaluate the trained model’s performance on the testing set, considering metrics like precision, recall, F-score and accuracy.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2}$$

$$\text{F-Score} = 2.\frac{(Precision*Recall)}{(Precision+Recall)} \tag{3}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

Where:
TP = True Positive values which stand for values that are correctly anticipated to be positive.
FP = False Positive values which stand for values that are incorrectly expected to be positive.
FN = False Negative values which stand for values that are incorrectly predicted to be negative.
TN = True Negative values which stand for values that are correctly predicted to be negative.

E. Mode Training

Hyperparameters such as `n_estimators`, `max_depth`, and grid search to optimize model performance were used for training the model, Numpy (Python library used for working with arrays). Imports sklearn, numpy, pandas, imbalanced-learn, cuml, matplotlib. Sets up rapids Python env with CUDA libraries for GPU acceleration.

IV. Experiment Result

The experiment was on the three-model used which are: Random Forest, Support vector machine on the dataset and are compare. The gradient boosting model looks most robust, maintaining high scores on the test set. The others overfit slightly more. The overall accuracy scores look quite good, all above 98-99% for train and validation. This suggests the models are fitting the training data very well. However, the test accuracy is a better evaluation of real-world performance. Here the scores drop slightly but are still strong at 98-100%. Precision and recall scores are also generally high, indicating the models are successfully learning the patterns in the data. The F1 scores are balanced, not favoring precision or recall heavily. The macro averages show there isn't a huge skew towards any particular class. The high accuracies and F1 scores indicate the models are learning the patterns and generalizing fairly well. Some overfitting is present but performance remains strong. Below is the comparison result for the models on training, testing and validation.

Table 1. Showing Training Classification

Model	Accuracy	Recall	Precision	F1
Random Forest	0.98	0.99	0.97	0.98
SVM	1.0	0.97	0.97	0.97
Gradient boosting	1.0	0.99	0.99	0.99

Table 2. Showing Testing Classification

Model	Accuracy	Recall	Precision	F1
Random Forest	0.98	0.99	0.97	0.98
SVM	1.0	1.0	0.99	0.99
Gradient boosting	1.0	1.0	1.0	1.0

Table 3. Showing Validation Classification

Model	Accuracy	Recall	Precision	F1
Random Forest	0.98	0.99	0.97	0.98
SVM	0.99	1.0	0.99	0.99
Gradient boosting	1.0	1.0	1.0	1.0

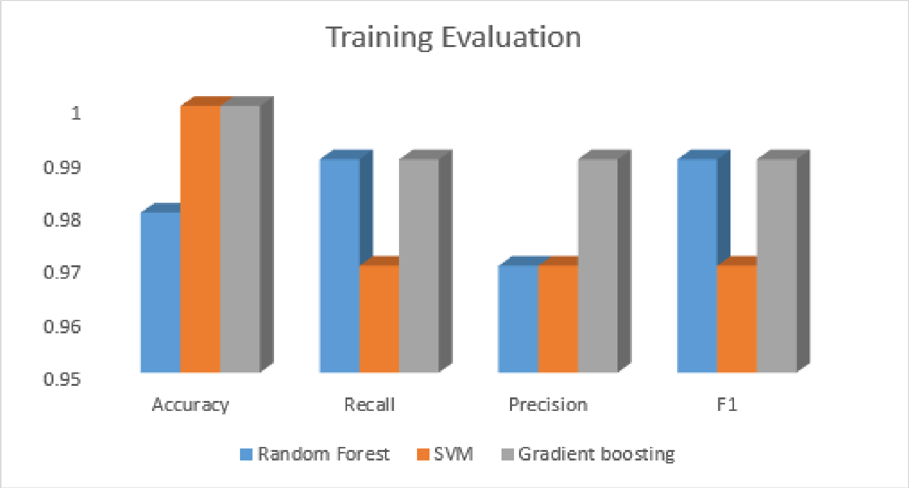


Fig 5. Graphical illustration of training classification

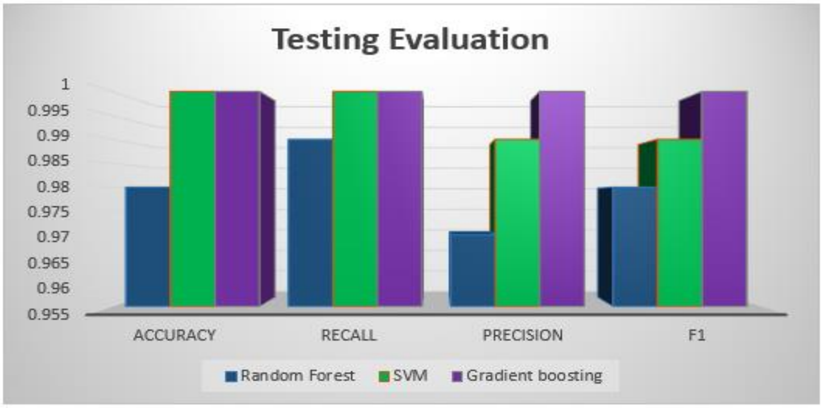


Fig 6. Graphical illustration of testing classification

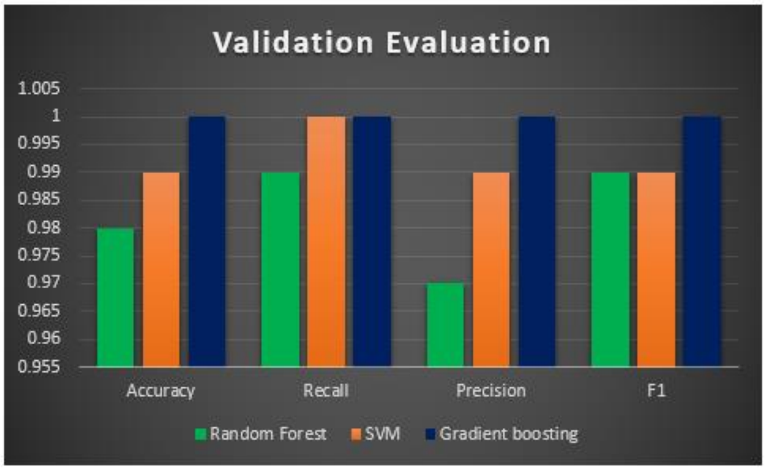
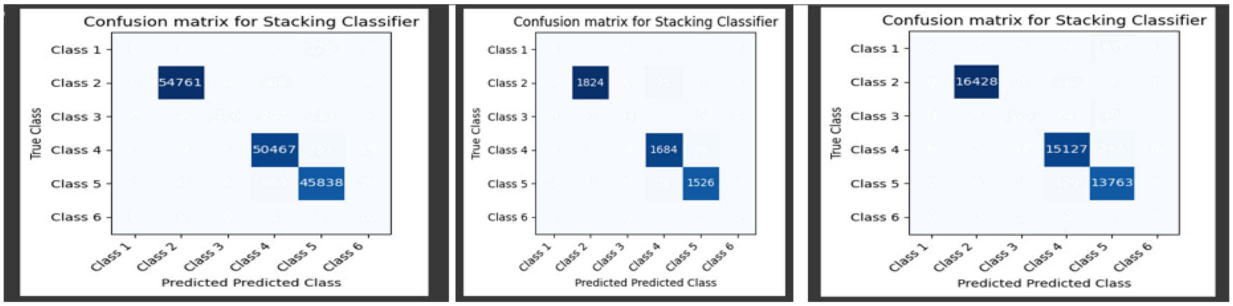


Fig7. Graphical illustration of validation classification

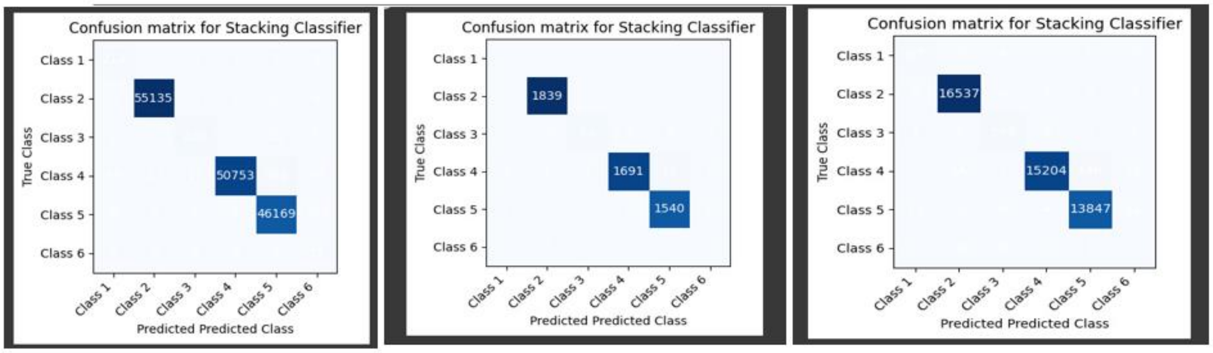
A. Confusion matrix

The selection of which metrics to prioritize is contingent upon the particular objectives and demands of the task at hand. These metrics offer distinct viewpoints on the model's performance.



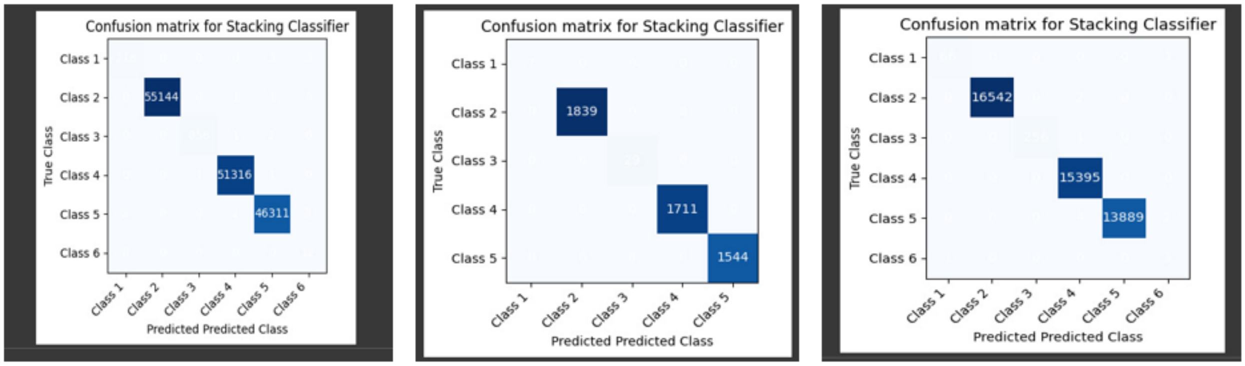
Random forest training matrix Random forest testing matrix Random forest validation matrix

Fig 8. Confusion matrix for Random forest



SVM training matrix SVM testing matrix SVM validation matrix

Fig 9. Confusion matrix for SVM



Gradient boosting training Gradient boosting testing Gradient boosting validation

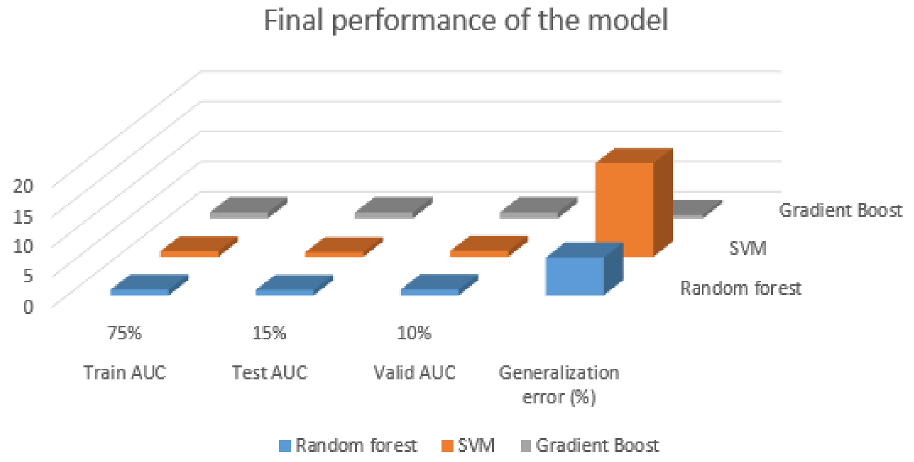
Fig10. Confusion matrix for Gradient boosting validation

B. Comparison on final performance on the model

The final performances of the models were compared over the ROC AUC score. An evaluation was carried out via the ROC AUC metric which is a tool for assessing and comparing the performance of classification models, particularly in situations where the balance between false positives and false negatives is important. These are tabulated as follows for the datasets.

Table 4. Showing Final Performance For The Models Of The Dataset

Model	Train AUC 75%	Test AUC 15%	Valid AUC 10%	Generalization error (%)
Random forest	0.962	0.899	0.98	6.3
SVM	0.923	0.767	0.99	15.6
Gradient Boosting	0.9998	0.9939	1.00	0.59



Figu11. Graphical illustration of the final performance

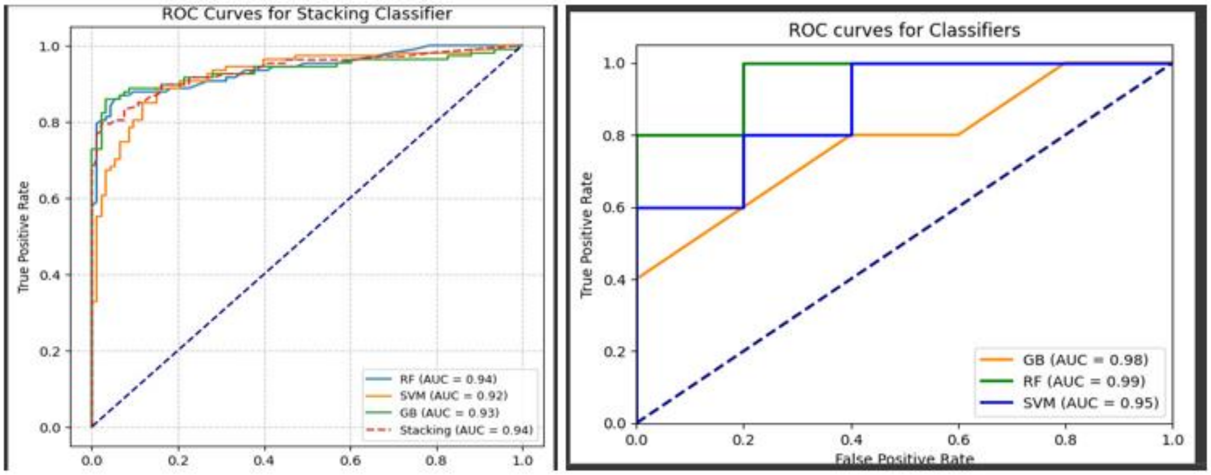


Fig 12. AUC Comparative analysis

V. Conclusion

Ransomware detection is an ongoing and multifaceted challenge that requires a combination of advanced technology, user awareness, and a proactive cybersecurity posture. Private users, commercial enterprises, and government networks must invest in modern detection techniques. The use of machine learning in ransomware detection has great potential to improve cybersecurity defenses. Nevertheless, by applying machine learning algorithms like gradient boosting, random forest, and support vector machines, this study has been able to offer advice on how to cope with both locker and crypto-ransomware. Users of computers can be more than 90% confident that their system is clear of ransomware. After the models were compared, it was found that the gradient boosting model (99%, 99%, and 100% AUC train, test, and validation, respectively) had the lowest generalization error, while the SVM model (92%, 76%, and 99% AUC train, test, and validation) performed the poorest. Despite this, the models were still overfit. In the middle of the pack (96%, 89%, and 98% AUC) was the Random Forest model. Hyperparametric optimization can be used to reduce the generalization error. The results, however, show that machine learning techniques have a lot of potential for use in cybersecurity and ransomware detection.

References

[1]. Abdullahi Arabo, Remi Dijoux,Timothee poulain,Gregoire Chevailer, (2020), Detecting Ransomware Using Process Behavior Analysis. Pp. 289 and 295

[2]. Darshana U., Jaume M., Marzia Z., and Srinivas S. (2019). Gradient Boosting Feature Selection with Machine Learning Classifiers for Intrusion Detection on Power Grids. IEEE Transactions on Network and Service Management. Pp. 3-5.

[3]. Drew Conway and John Myles White (2012) Machine Learning for Hackers. First edition *http://oreilly.com/catalog/errata.csp?isbn=9781449303716* O'Reilly Media, Inc. Pp. 275-278.

[4]. Eduardo Berrueta, Daniel Morato, Eduardo Magana, Mikel Izal (2022), Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic. Pp. 1-3

[5]. Fayez Tarsha Kurdi (2021), Random Forest Machine Learning Technique for Automatic Vegetation Detection and Modelling in LiDAR Data. International Journal of Environmental Sciences & Natural Resources. Pp. 001. (Fayez Tarsha

[6]. Juan A. Herrera-Silva and Myriam Hernández-Álvarez (2023), Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms. Pp. 1-21

[7]. SH Kok, Azween Abdullah, NZ Jhanjhi and Mahadevan Supramaniam (2019) Ransomware, Threat and Detection Techniques: A Review. ILCSNS International Journal of computer Science and Network Security, Vol. 19.2, Pp. 138-139.

[8]. Kok S.H. and Mahadevan (2019) Prevention of Crypto-Ransomware Using a Pre-Encryption Detection Algorithm. Articles www.mdpi.com/journal/computers. Pp. 2-5

[9]. Manabu Hirano and Ryotaro Kobayashi (2019) 'Machine Learning Based Ransomware Detection Using Storage Access Patterns Obtained from Live-forensic Hypervisor" Conference Paper · October 2019. Pp. 2-7.

[10]. Olaniyi Abiodun Ayeni, Otasowie Owolafe, Olabiyi Akinsola (2021), Malware Detection using Machine Learning, Conference paper. Pp. 86

[11]. Samah Alsoghyer and Iman Almomani (2019) Ransomware Detection System for Android Applications. Pp. 1-31.

[12]. SH Kok, Azween Abdullah, NZ Jhanjhi and Mahadevan Supramaniam (2019) Ransomware, Threat and Detection Techniques: A Review. ILCSNS International Journal of computer Science and Network Security, Vol. 19.2, Pp. 138-139.

[13]. SH Kok, Azween Abdullah, NZ Jhanjhi and Mahadeyan supramaniam. (2019), Ransomware, threat and Detection Techniques: A review. Pp. 1-11

[14]. Subash Poudyal, Dipankar Dasgupta, Zahid Akhtar, Kishor Datta Gupta., (2019), A Multi-Level Ransomware Detection Framework using Natural Language Processing and Machine Learning. Pp. 2-9

[15]. Xiang G., et al. (2019). An Improved Random Forest Algorithm for Predicting Employee Turnover. Research Article. Pp. 2-5