International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No.42, February 2024 – www.ijais.org

Detection of Ransomware using Random Forest, Support Vector Machine and Gradient Boosting Techniques

Adejumo Ibitola Elizabeth
University of Medical Science, Ondo. Nigeria
Department of Computer Science
Faculty of Science

Olaniyi Abiodun Ayeni
Federal University of Technology, Akure, Nigeria
Department of Cyber Security
School of computing

ABSTRACT

The internet's introduction and subsequent growth have made it possible to connect people worldwide, and this trend is continuing numerous benefits result from this, including connectivity and communication as well as the broadcast and transmission of information. The cyberspace, the concept of the space within which all internet and telecommunication activities take place has become an important resource. As it is shared across the world, all information transmitted within and through this space is fair game for any who is capable of intercepting it. The aim of this research is to detect crypto-ransomware and locker. There are many means of attempting this. However, one of the simpler ideas may be to neglect the cyberspace completely. Rather than attempt to intercept signals, or spam/overload servers, it is possible to intercept the information right on the computer system it originates on.

General Terms

Machine learning Algorithm, Ransomware.

Keywords

Random Forest, Support vector Machine, Gradient boosting Algorithm

1. INTRODUCTION

Ransomware is a type of malware that attacks your vital information and systems with the intention of demanding money or extortion [2]. Email spear phishing is a common method for distributing ransomware. A ransom demand is made by the cyber actor after the user has been locked out of the data or system. Following payment, the cyber attacker will allegedly provide the victim with a way to reclaim access to the network or data. Recent iterations target business end users, making education and training an essential preventive strategy. Ransomware targets private users, commercial enterprises, and networks run by the government. which may cause confidential or sensitive data to be lost temporarily or permanently, the disruption of routine company operations, costs associated with having to replace computers and files, as well as potentially serious harm to an organization's brand. The user may be instructed by ransomware to click on a link in order to pay a ransom; however, the link may be dangerous and could spread other malware [1].

[9] concluded that different techniques are being used by the malware to spread more widely. Cybercriminals are always looking for new ways to manipulate people through social engineering and target their victims. By using untraceable payment methods like bitcoins, it is now their main source of income. Cybercriminal's affect not just home users but also big enterprises and other establishments where there is a greater likelihood of receiving a ransomware attack. Implementing preventive steps, such as updating software's, using antivirus software, properly screening data obtained via a network, keeping backups, avoiding dubious links or emails, etc., is the only method to prevent being impacted by this dangerous program. Ransomware is being carryout by downloading a malicious file from the internet unknowingly to the victims from connecting to the internet and hackers mostly rely on social engineering techniques to spread their malicious mail to victims [14].

2. RELATED WORKS

[13], researched on dual Generative Adversarial Networks Based Unknown Encryption Ransomware Attack Detection. Aim at a detection method based on dual generative adversarial networks is used to identify unknown or variant ransomware attacks encrypted with the SSL protocol. Transfer learning mechanism is used to improve the generator's ability to generate adversarial samples and the discriminator's ability to detect normal samples in TGAN. A reconstruction loss function is introduced to further improve the discriminator's ability to detect normal sample. CNN (Convolutional Neural Network), DCGAN (Deep Convolutional Generative Adversarial Network), KDD99 DATASET, SWaT AND WADI DATASETS, and offered a dual generative adversarial networks detection framework based on DCGAN and TGAN is presented to find unknown or variant encrypted ransomware attack with high precision.

Notwithstanding, this paper presents a detection method based on dual generative adversarial networks, which is named TGAN-IDS, other techniques when deployed may enhance the performance of the unknown attack.

The work of [12], titled Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. To help researchers and developers who want to use machine learning or deep learning techniques to detect crypto ransomware, by providing a comprehensive list of future directions that will open up new avenues for research. To identify suitable solutions for each category that machine learning and deep learning can be applied to (hybrid and dynamic analysis) gave a thorough summary of research on ransomware detection using deep learning and machine learning, classifying the studies according to the timing of the encryption process, extracting a list of research topics that need to be pursued by other researchers, provided a brief taxonomy, described the studies on ransomware attack detection from 2019 to 2021, with an emphasis on dynamic analysis for various platforms, and displayed a number of datasets, in order to do dynamic analysis and to train and test ransomware detection systems created utilizing machine and deep learning techniques, their sources and analysis tools were used. Nevertheless, the researcher claims that ransomware that is created with its own encryption method may avoid detection. Static analysis's scope is limited due to its higher false alarm rate and lower accuracy. Additionally, ransomware that uses environmental fingerprinting can evade analysis. A few studies did not provide details about the dataset and analysis. Finally, analysis conducted for a set amount of time may have aided evasion techniques. The availability of a limited amount of data during the initial stage of the encryption process makes it difficult to identify ransomware that uses obfuscation and evasion techniques, Runtime detection algorithms may be compromised by malicious programs, and some detection studies are unable to identify system API calls or encrypted actions that occur during execution. These studies also fail to specify the source of the dataset and the quantity of samples used. A rogue application has the potential to corrupt hardware data.

[8], researched on RanSAP: An open dataset of ransomware storage access patterns for training machine learning models. It was suggested to provide a hypervisor-based storage access pattern monitoring system, which would be followed by the creation and use of a feature extractor and machine learning models for ransomware detection via dynamic analysis, machine learning methods, and dataset collection. Provide an open dataset with storage access patterns for five benign programs and seven well-known ransomware samples on a range of storage device kinds and situations. A variety of dynamic ransomware storage access pattern features. However, the author also discusses the limitations of the dataset, how it compares to other datasets and dynamic analysis techniques, use cases for the suggested dataset and detection method, and discussions on the most recent advancements in ransomware and malware detection, including evasion techniques and adversarial ML attacks.

The work of [3], titled ransomware Detection using Random Forest Technique offers a cutting-edge technique for ransomware detection that is based on static analysis.

Static analysis and the random forest classifier are two machine learning techniques used. Frequent pattern mining, normalization, and feature extraction from raw bytes are the three stages of preprocessing. High detection accuracy is achieved by employing 32-bit sliding windows (4-gram) features in the feature extraction procedure, which is carried out in a virtual machine.

offered a method for detecting ransomware attacks using a machine learning technique called a random forest classifier. The trials, in just 1.37 seconds, a tree size of 100 with a seed size of 1 achieved a high accuracy of 96.74%, a high ROC of around 99.6%, low FPR of approximately 0.04%, and low FNR of approximately 0.002%. The features ranging from 100 to 1000 displayed a poor detection rate. This was highlighted by the current analysis. Additionally, accuracy decreased as feature counts exceeded 1000, and tree counts between 200 and 1000 performed comparably to those of 100.

However, this system can detect Ransomware using random forest techniques with the accuracy of 97.74%. This can actually be improved on.

[10], presented Avoiding Future Digital Extortion Through Robust Protection Against Ransomware Threats Using Deep Learning Based Adaptive Approaches. Utilizing deep learning techniques with deep learning-based adaptive methodologies, calculate the underlying latent causes of the changing patterns in the new variants in an unsupervised manner. The gathering, preparing, and analyzing of data. The model can extract the ransomware attack patterns through a semi-supervised, deep learning approach, which involves the creation of a global feature collection and feature selection using FastICA. However, the study only extracts ransomware attack pattern using adaptive approaches and also developed an adaptive detection using deep learning based, there could be another algorithm that could perform better than the algorithm used.

The research of [7], titled A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning. to provide a novel and effective digital DNA sequencing engine that employs machine learning (ML) to identify ransomware before the first stage of an assault. AdaBoost, Decision Stump, and Naïve Bayes are three examples of machine learning algorithms that are used. a novel technique that detects and categorizes ransomware using an AI machine learning network and the Digital DNA Sequencing Engine. Utilizing the BCS and MOGWO algorithms, the k-mer frequency vector, digital DNA sequence restrictions, and computation of digital DNA sequences are produced based on the DNA sequences, the proposed concepts include a software product model, requirement model, compliance model, and ransomware detection methodology that uses a digital genome to classify and detect ransomware. Additionally, a concept demonstrator tool is provided to show the viability of the above concepts by successfully detecting and classifying ransomware using an active machine learning algorithm and real-world datasets. Meanwhile the system shows 78.5% accuracy using Naïve Bayes, 75.8% accuracy using Decision Stump and 87.9 % accuracy using AdaBoost during detection. This can actually be improved on.

[11], presented LooCipher Ransomware Detection Using Lightweight Packet Characteristics. In order to give a comprehensive packet analysis with a lightweight understanding of domain-specific ransomware analysis, this research suggested an approach that uses Lightweight Packet Characteristics to track online behaviors and comprehend the source/destination entities. However, the use of machine learning or deep learning techniques could offer better ransomware detection.

3. PROPOSED METHOD

The study's focus is to detect ransomware. Multiclass data on ransomware and associated features will be sources uci.com in Excel format. Initializing threshold settings, outliers will be checked by comparing how far the nearest data point is from the closest cluster identification and identifying those that are outliers in our dataset. The dataset for the ransomware will be cleaned up and outliers removed. Any missing values that data cleaning may produce will be fixed. Following the creation of new variables from a ransomware dataset, the variables will be filtered and duplicate values will be removed. To maximize the variance of the dataset, Gradient boosting algorithm, random forest and support vector machine will be utilized in supervised ML for zero-day attack detection. The most important stage in tackling controlled machine learning challenges is data collection. When challenged to recall objects from their training, machine learning models frequently produce excellent results.

In ransomware detection, a number of procedures are included in the detection methodology, including the data collection, data pre-processing of the collected data, feature selection techniques for choosing the necessary set of characteristics for recognizing ransomware, and machine learning classifiers in order to categorize the supplied data into predetermined groups.

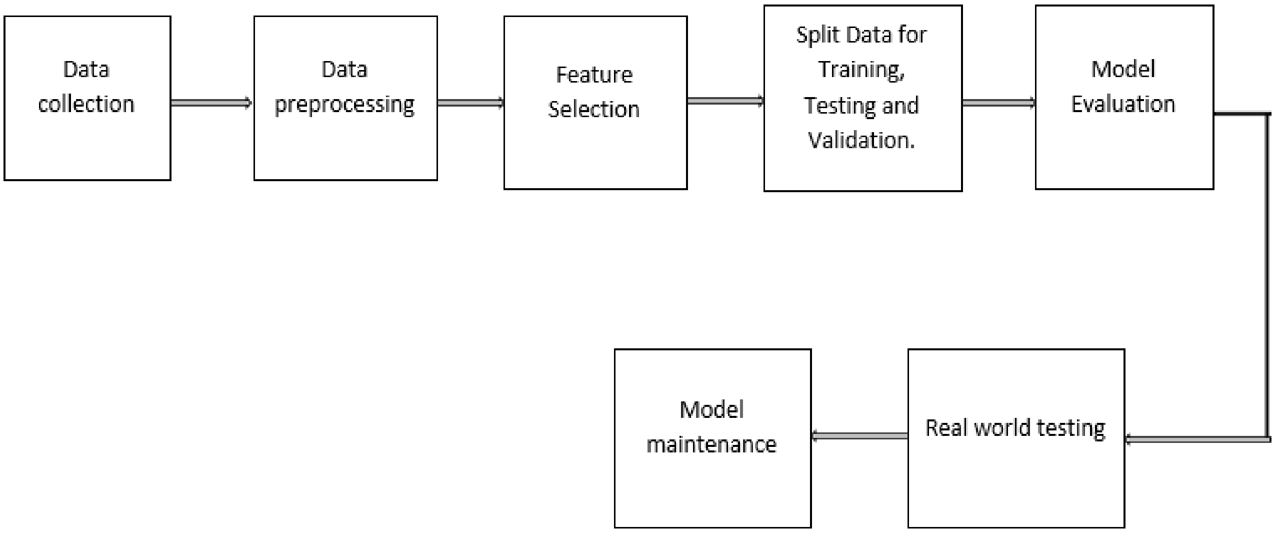


Fig 1: System Architecture

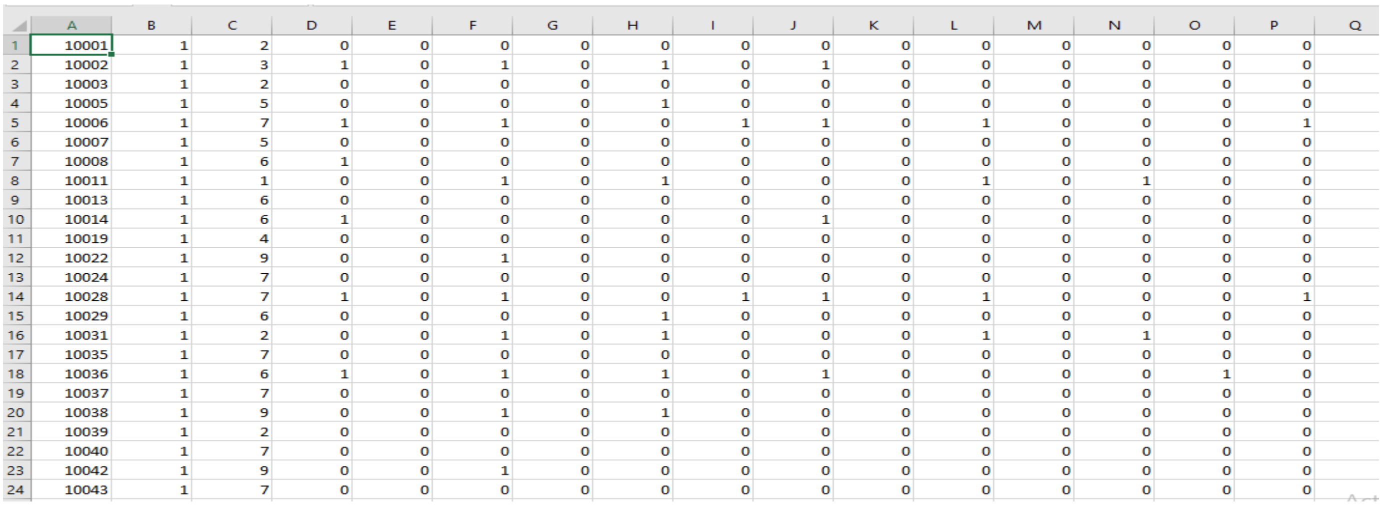
The proposed method will be divided into three modules as follows:

- Module 1: Dataset collection
- Module 2: Data preprocessing
- Module 3: Machine learning classifier

Modules

Module 1: dataset collection

Ransomware dataset sourced from uci.com in Excel format for training, testing and validating the dataset. A ratio of 75:15:10 would be adopted for splitting the data collected into training, testing and validating the dataset. The dataset contains total number of 62485 rows, the 75-split contained 46863 rows, 15-split contained 9372 rows while 10- split contains 6248 rows and 50 columns. The data was cleaned appropriately, after which particular features were selected.

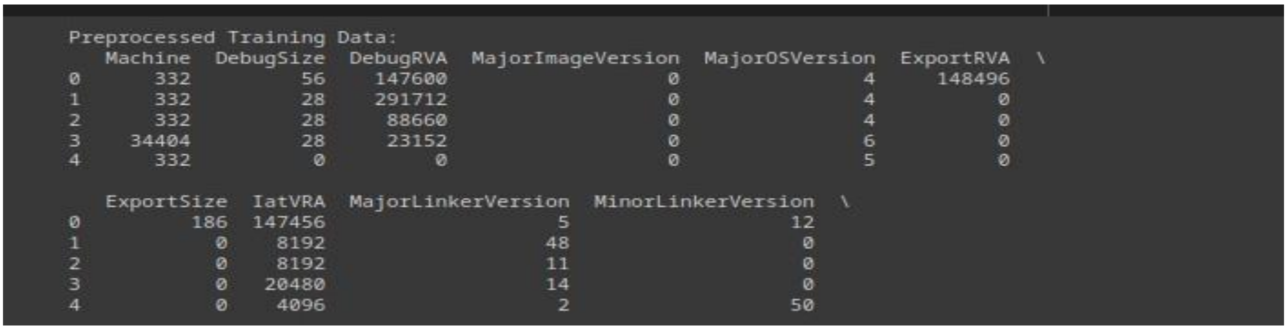


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	10001	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	10002	1	3	1	0	1	0	1	0	1	0	0	0	0	0	0	
3	10003	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	10005	1	5	0	0	0	0	1	0	0	0	0	0	0	0	0	
5	10006	1	7	1	0	1	0	0	1	1	0	1	0	0	0	1	
6	10007	1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	10008	1	6	1	0	0	0	0	0	0	0	0	0	0	0	0	
8	10011	1	1	0	0	1	0	1	0	0	0	1	0	1	0	0	
9	10013	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	10014	1	6	1	0	0	0	0	0	1	0	0	0	0	0	0	
11	10019	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	10022	1	9	0	0	1	0	0	0	0	0	0	0	0	0	0	
13	10024	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	10028	1	7	1	0	1	0	0	1	1	0	1	0	0	0	1	
15	10029	1	6	0	0	0	0	1	0	0	0	0	0	0	0	0	
16	10031	1	2	0	0	1	0	1	0	0	0	1	0	1	0	0	
17	10035	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	10036	1	6	1	0	1	0	1	0	1	0	0	0	0	1	0	
19	10037	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	
20	10038	1	9	0	0	1	0	1	0	0	0	0	0	0	0	0	
21	10039	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	10040	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	10042	1	9	0	0	1	0	0	0	0	0	0	0	0	0	0	
24	10043	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig 2: Sample of Dataset

Module 2: Data preprocessing

The ransomware dataset collected from uci.com will be cleaned to remove any noise or inconsistencies. Perform feature selection to extract meaningful features from the raw data. This will involve techniques like dimensionality reduction, feature selection, or transformation. Apply suitable feature selection techniques to determine which most informative and relevant features for detecting ransomware.



```
Preprocessed Training Data:
   Machine  DebugSize  DebugRVA  MajorImageVersion  MajorOSVersion  ExportRVA  \
0      332         56    147600                  0               4     148496
1      332         28    291712                  0               4          0
2      332         28     88660                  0               4          0
3    34404         28     23152                  0               6          0
4      332          0         0                  0               5          0

   ExportSize  IatVRA  MajorLinkerVersion  MinorLinkerVersion  \
0         186  147456                   5                  12
1           0    8192                  48                   0
2           0    8192                  11                   0
3           0   20480                  14                   0
4           0    4096                   2                  50
```

Fig 3: Dataset after preprocessing

Module 3: Machine learning classifier

Three machine learning algorithms which include Random Forest Algorithm, Gradient boosting algorithm and Support Vector Machine will be used as classifiers. These techniques will compare the model using a classifier.

The models are described below:

Base model 1: Random forest algorithm

Random forest aims at lowering the amount of time needed for learning and classification either to seek to increase accuracy, performance or both [6]. Mathematically expressed as:

$\hat{y}(x) = argmax_y(\sum_{i=1}^{n} \|(T_{i(X)=y}))$ [1]

Where:

$\hat{y}_{(x)}$ denotes the predicted output.

argmax is maximum number of the trees involved,

T is total number of the trees

y is a class label,

|| is the indicating functions that returns 1 if the condition or events inside is true and 0 otherwise.

Base model 2: Gradient boosting algorithm.

Boosting keeps the leaf node's labels and weights in a way that makes handling prediction interprets simple [4].

This can be illustrated mathematically as

$y_E = \sum_i a_i f_i(\bar{x})$ [2]

Where $a_i f_i(x)$ represents a function of a weak learner

a_i the weight or contribution of the leaf node.

$f_i(\bar{x})$ the prediction of the leaf nodes for the input $\bar{x}$

Base model 3: Support vector machine

Given a training dataset with labeled examples (x_i, y_i), where x_i stands for the input features and y_i for the matching class label (-1 or 1). SVM seeks to identify the ideal hyperplane that divides the data into the best possible two classes [5].

Model Representation:

The decision function for SVM is represented as the dot product of the input features (x) and a weight vector (w), plus a bias term (b):
f(x) = w · x + b [3]
Where:

f(x) is the decision function that predicts the class label (-1 or 1) for input x.

w is indicating the weight vector (coefficients) that ascertains the orientation of the hyperplane.

x is the input feature vector.

b is indicating the bias term (also known as the intercept).

Constraints:

The SVM aims to find the best hyperplane such that the following constraints are satisfied:

For positive samples ($y_i = 1$): $w \cdot x_i + b \geq 1$ [4]

For negative samples ($y_i = 0$): $w \cdot x_i + b \leq -1$ [5]

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.

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ [6]

Precision = $\frac{TP}{TP+FP}$ [7]

Recall = $\frac{TP}{TP+FN}$ [8]

F-Score = 2.$\frac{(Precision*Recall)}{(Precision+Recall)}$ [9]

Where:

TP = True Positive values are those that are accurately expected to be positive.

FP = False Positive values are values that are mistakenly predicted to be positive.

FN = False Negative values represent values that are mistakenly expected to be negative.

TN = True Negative values represent values that are accurately expected to be negative.

4. EXPERIMENT AND RESULT

The experiment is based on the three-model used which are Random Forest, Support vector machine on dataset. These models were carried out in training, testing and validation. In model Training Performance, the training accuracy is quite high for all models, ranging from 76-98% accuracy. This suggests the models are overfitting and memorizing the training data. Precision and recall on the training set are also very high, mostly over 0.9. This further indicates overfitting. The macro averages show there is not a large class imbalance issue. In model Testing Performance, test accuracy drops further for all models, now ranging from 65-72%. Overfitting is clearly occurring. Precision, recall and F1 are quite low, with many scores below 0.7 now. Performance degrades significantly. SVM generalizes slightly better than random forest and gradient boosting. But all models overfit. Finally, in model Validation Performance, Validation accuracy drops significantly compared to training for all models, ranging from 65-92%. This gap indicates overfitting. The SVM model generalizes best to the validation data with 92% accuracy. The random forest performs worst at 65%. Precision, recall and F1 scores also decline on the validation set. There is a noticeable drop in performance. Below is the comparison result for the models on training, testing and validation.

Table 1: Training result classification result

Model	Accuracy	Recall	Precision	F1
Random Forest	0.87	0.98	0.97	0.97
SVM	0.76	0.68	0.98	0.80
Gradient boosting	0.98	0.89	0.83	0.89

Table 2: Testing result classification result

Model	Accuracy	Recall	Precision	F1
Random Forest	0.65	0.75	0.78	0.77
SVM	0.72	0.78	0.83	0.80
Gradient boosting	0.70	0.77	0.81	0.79

Table 3: Validation result classification

Model	Accuracy	Recall	Precision	F1
Random Forest	0.88	0.82	0.87	0.84
SVM	0.92	0.88	0.92	0.90
Gradient boosting	0.92	0.76	0.86	0.80

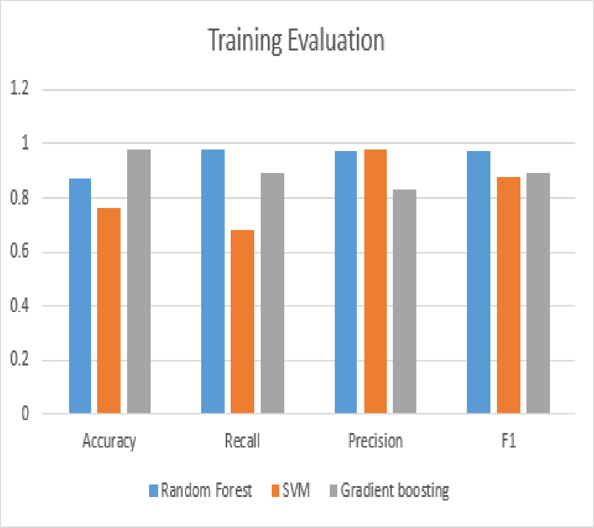


Fig 4: Graphical illustration of Training evaluation.

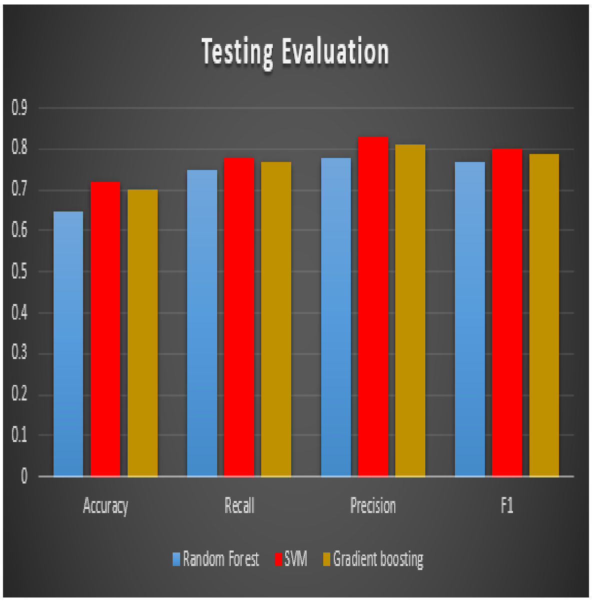


Fig 5: Graphical illustration of Testing evaluation.

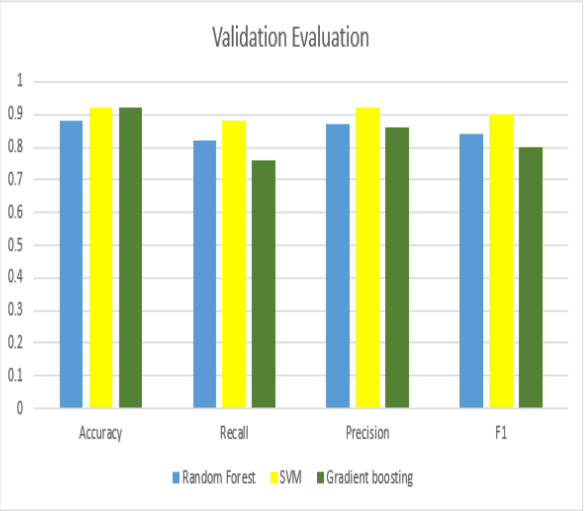
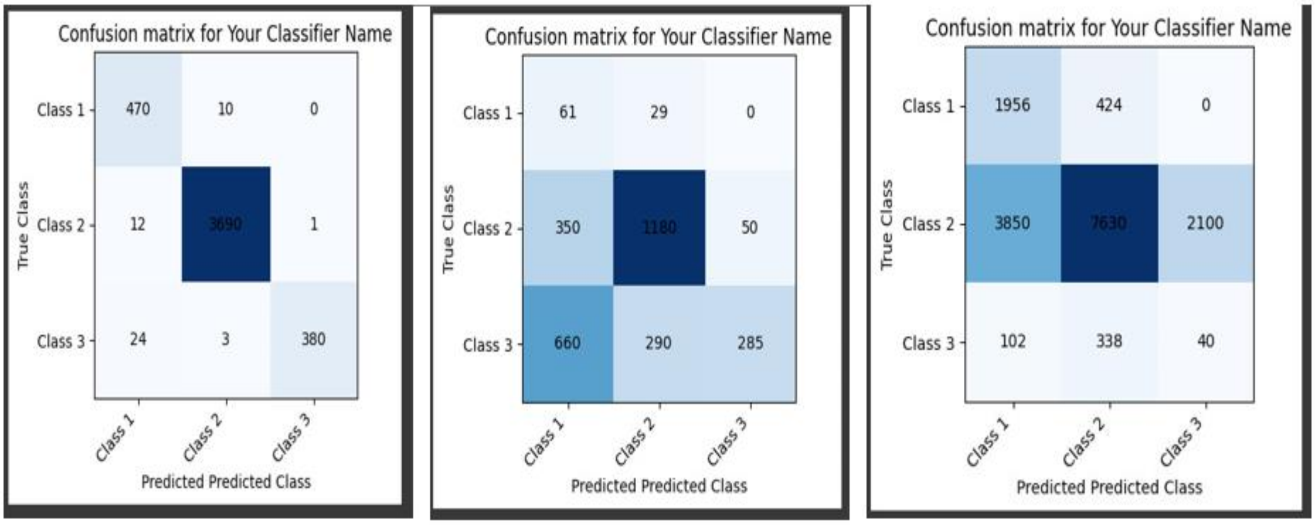


Fig 6: Graphical illustration of validation evaluation

4.1 Confusion matrix

The specific goals and requirements of the task at hand will determine which metrics should be prioritized. Different perspectives on the model's performance are provided by these metrics.

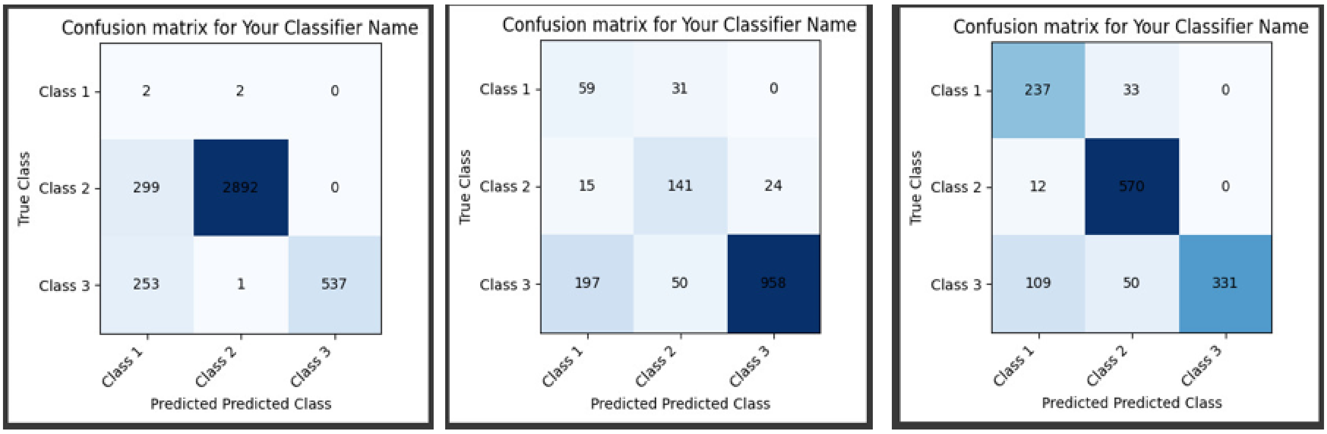


Random forest training

Random Forest testing

Random forest Validation

Fig 7: Confusion matrix for Random Forest

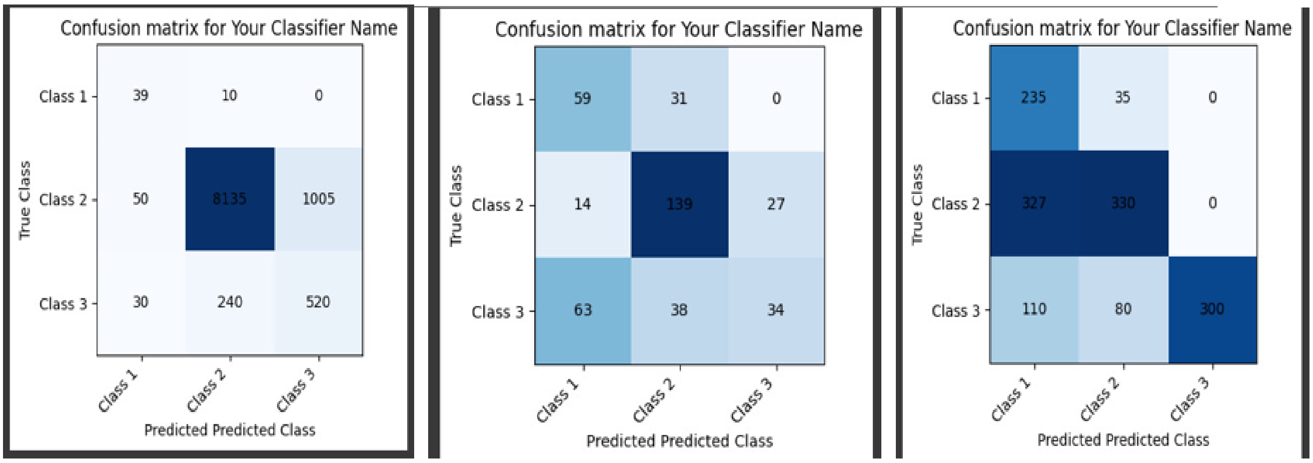


SVM training

SVM testing

SVM validation

Fig 8: Confusion matrix for SVM



Gradient boosting training

Gradient boosting testing

Gradient boosting validation

Fig 9: Confusion matrix for Random Forest

4.2. Comparison on final performance on the model

The ROC AUC score was used to compare the models' final performances. The ROC AUC metric was utilized to conduct an assessment and comparison of the performance of classification models, especially in scenarios where maintaining a balance between false positives and false negatives is crucial. These are tabulated for the datasets in the following ways.

Table 4: showing final performance for the models for the dataset

Model	Train AUC 75 %	Test AUC 15 %	Valid AUC 10 %	Generalization error (%)
SVM	0.94	0.78	0.99	-0.5
Random forest	0.98	0.75	0.92	0.06
Gradient Boosting	0.99	0.80	0.95	0.04

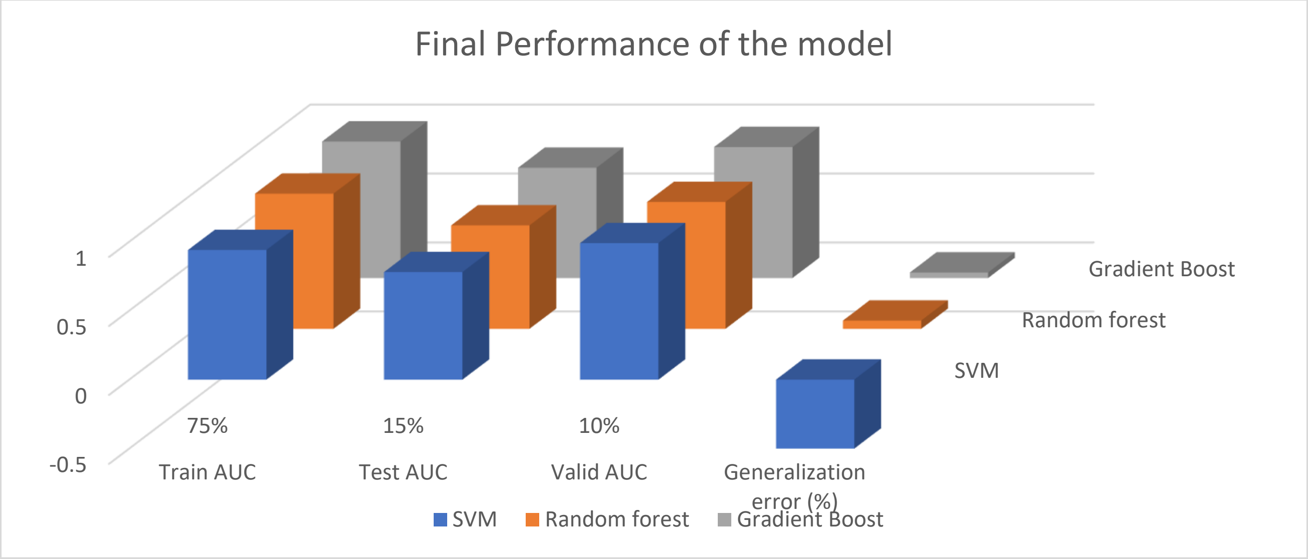


Fig 10: Graphical illustration of the final performance.

4.3. Comparative analysis

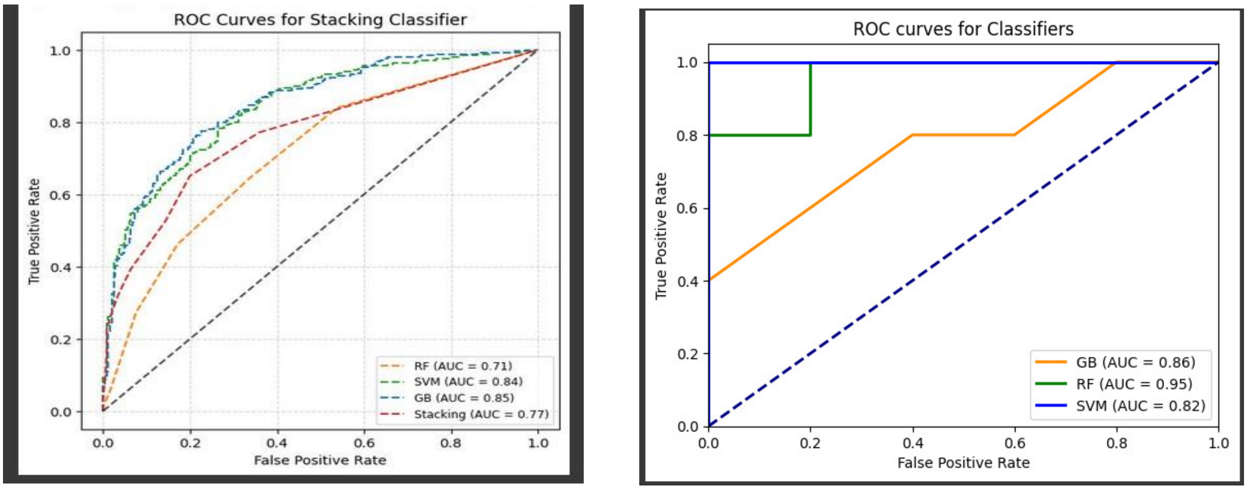


Fig 11: Comparative analysis

5. CONCLUSION

One well-known method that cybercriminals frequently employ to infect their victims either through drive-by downloads or phishing emails is ransomware. Perpetrators will craft an e-mail that appears to be from a reliable source and forward it to the intended recipients. Nonetheless, this study has been able to provide guidance on how to deal with both locker and crypto-ransomware. The system can detect ransomware under 30 seconds by implementing machine learning algorithms such as gradient boosting, random forest, and support vector machines. This gives computer users over 90% assurance that their system is free of ransomware. The models were compared and although they overfit, the SVM model (94%, 78%and 99% AUC train, testing and validation respectively) had the least generalization error, with the Random Forest model (98%,75% and 92% AUC train, test and testing) doing worst. The gradient boosting model was somewhere in between (99%, 80% and 95% AUC). The generalization error can be worked on via hyperparametric optimization. However, the results display great promise for the use of Machine Learning algorithms in cybersecurity for ransomware detection.

6. REFERENCES

[1] Ade Kurniawan and Imam Riadi (2018) “Detection and Analysis Cyber Ransomware Based on Network Forensics Behavior”, International Journal of Network Security. Pp. 2-3

[2] Ayeni Olaniyi Abiodun (2022), A Supervised Machine Learning Algorithm for Detecting Malware. Journal of Internet Technology and Secured Transactions (JITST). Pp. 765

[3] Ban Mohammed Khammas (2020), Ransomware Detection using Random Forest Technique. Pp. 325-330

[4] 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.

[5] 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.

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

[7] Firoz Khan, Cornelius Ncube, Lakshmana Kumar Ramasamy (2020), A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning. Pp. 119710 – 119718.

[8] 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.

[9] Sana Aurangzeb, Muhammad Aleem, Muhammad Azhar Iqbal and Muhammad Arshad Islam “Ransomware: A Survey and Trends” uploaded 2020. Pp. 2-9

[10] Shaila Sharmeen, Yahye Abukar Ahmed, Shamsul Huda, Bari. Koçer, and Mohammad Mehedi Hassan (2020), Avoiding Future Digital Extortion Through Robust Protection Against Ransomware Threats Using Deep Learning Based Adaptive Approaches. Pp. 24522 – 24524

[11] Te-Min Liu, Da-Yu Kao and Yun-Ya Chen, (2020), LooCipher Ransomware Detection Using Lightweight Packet Characteristics. Pp. 1677 – 1683

[12] Umara Urooj, Bander Ali Saleh Al-rimy, Anazida Zainal, Fuad A. Ghaleb and Murad A. Rassam. (2021), Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. Pp. 2

[13] Xueqin Zhang and Shinan Zhu. (2022), Dual Generative Adversarial Networks Based Unknown Encryption Ransomware Attack Detection. Pp. 901 and 912.

[14] Yagiz Y. (2022). Personality Types and Ransomware Victimisation. Pp 2.