The edge of Deep Learning over Traditional Machine Learning Methods in Ransomware Detection on Bitcoin Network

Project Report for CSE3501

Information Security Analysis and Audit

Akhil Srinivas Parimala	**20BCE0567**
Archit Sharma	**20BCE0956**
Soumya Kumar	**20BCE2378**
Ayushman Khuntia	**20BCE2615**

Submitted to

Prof Ramani S

School of Computer Science and Engineering

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Index

ABSTRACT

Ransomware is a form of Malware which encrypts data using some data encryption techniques and prevents the owner of the data accessing their own files, for which some credit is demanded in forms of money, cryptocurrencies and online transactions to release the key for decrypting the locked files. Online transactions used to leave traces behind and thus the ransomware activities/transactions were then moved to Cryptocurrency technology. This technology minimised the amount of traces which were left behind and were of an advantage to the attacker as the system is based on Blockchain Technology and is completely decentralised. For Transactions to take place, the attackers are left with bitcoin hash address, which have passed through traditional machine learning models to identify a pattern in ransomware transactions. From this observation, we are trying to study the effect of Neural Network Algorithms (Deep Learning) on the dataset and verify the classification of the address with the traditional ML Models and perform a comparative study.

***Keywords*: Ransomware, Bitcoin, Deep Neural Network, ransomware , malware , malicious, suspicious, crypto, dataset, detection, execute , addresses, algorithms, files, user, accuracy.**

INTRODUCTION

Ransomware is a type of malware that can spread inside a local network via a variety of vectors, including malicious email attachments, pay-per-install networks, and existing network vulnerabilities. When the ransomware is run on the host, it encrypts the files and documents and displays a ransom note on the screen, stating that the data will be unlocked if the ransom is paid in Bitcoins.

Bitcoin is a peer-to-peer internet transaction protocol that was created in the late 2000s by a group of developers. As it functions as a virtual currency, the framework is utilised to perform electronic payments. The entire network is decentralised, and transactions are recorded in a public ledger that is distributed among all nodes. As the transactions recorded in the ledger are irreversible and not controlled by a single authority. Hence, they fall outside of the governance control. A Bitcoin address is made up of alphanumeric characters received from the user's public and private keys. This renders user information pseudo-anonymous, safeguarding users' identities and making it suitable for illicit transactions.

When the targeted users receive fraudulent email attachments, or if any of their internet infrastructure have vulnerabilities that can be exploited, that’s when the attack begins.

The ransomware software begins to run on the host system and encrypts the user's important files and documents. After the encryption is complete, the attacker will email the victim a Bitcoin address to which the victim must send Bitcoins.

The ransomware displays a ransom notice alert on the user's screen, demanding payment in Bitcoins. The user's files are decrypted once the ransom is paid, and the user has access to their file The address which the attacker sends to the victim is a ransom address using which we can obtain clustering heuristics in the Bitcoin network.

The ransom message directs the victim to purchase Bitcoins from certain exchanges and internet services that allow Bitcoins to be converted to traditional currencies. These exchanges operate on a worldwide or regional scale, and some are centralised while others allow for direct exchange.

On the payment of ransom the victim’s files and documents are either automatically decrypted by the ransomware or the attacker sends the

In this project we tried out different machine learning approaches to detect the Bitcoin ransomware addresses and we found out that the neural network model performs slightly better compared to other models in metrics like Receiver Operating Characteristic (ROC), F1 score and accuracy. Neural network captures the information granularity at various layers which helps to detect ransomware addresses in a more generalised way.

PROBLEM STATEMENT & OBJECTIVES

2.1. Problem Statement:

Ransomware is a trailblazing malware that can spread inside a local network via a variety of vectors. When the ransomware is run on the host, it encrypts the files and documents and displays a ransom note on the screen, stating that the data will be unlocked if the ransom is paid in Bitcoins. In our project we tried to implement different machine learning approaches to detect Bitcoin ransomware addresses in order to detect attacks for the end user.

2.2. Objectives:

1. Detecting bitcoin ransomware attacks.
2. Preventing bitcoin ransomware attacks.
3. Improving effectiveness in detection and removal of malicious software.
4. Improving effectiveness in detection and removal of malicious software.
5. Spreading awareness from possible ransomware attacks.
6. Protecting the end user from possible redemption.
7. Preventing cybercrimes in society.

LITERATURE REVIEW

3.1. Hairil et al[1]

Approach:

- Authors have tried out different models in machine learning to classify bitcoin addresses as ransomware or genuine address.
- Emphasis on the dataset and discarding unused data headers. The author highlighted how the different conventional machine learning models have underperformed due their limitations.
- Even though these conventional methods have a performance score of just 5% less than the neural networks, various other mathematical factors in classification are looking in favor of the NN because of its ability to learn from past events.

Demerits:

- Training of the Model has to be done using a proper dataset after cleaning. Otherwise the classification may yield improper results.
- Model training may take longer times than conventional machine learning methods.

3.2. Samuel Egunjobi et al[2]

Approach:

- In this paper, researchers a classification technique of integrating both static and dynamic features to increase the accuracy of detection and classification of ransomware.
- In this work, the algorithms that will be used to carry out the machine learning procedure include: instance-based (IB1), Random Forest (RF), Naive Bayes and Support vector machine (SVM) .
- The classification of the dataset is done using the WEKA which is a dedicated tool for machine learning

Demerits:

- Using these techniques the program is identified as being malicious software based on sensitive APIs because of which there is increased chances of benign software invoking these API's.

3.3. Khammas, Ban Mohammed et al [3]

Approach:

- The strategy adopted is based on extracting the hierarchical features in the ransomware family as each ransomware family has common features.
- The pre-processing includes three steps namely feature extraction from raw bytes, frequent pattern mining, and normalization.
- The feature extraction process is performed in Virtual machine using 32-bit sliding windows (4-gram) features for high detection accuracy.
- Gain Ratio (GR) has been chosen as the feature selection method.

Demerits:

- Longer Training Period: Random Forest require much more time to train as compared to decision trees as it generates a lot of trees (instead of one tree in case of decision tree) and makes decision on the majority of votes.

3.4. Subash Poudyal et al[4]

Approach:

- A multi-level framework is run in an active mode so as to analyse the given binaries at three levels.
- This framework tracks the detection rate at each level going from DLL to the assembly instruction level so is named a multi-level framework. enervation and Machine learning prediction.
- Each phase conducts various operations such as reverse engineering, NLP schemes (N-grams), Action engine, Multi-Level Extractor, etc.

Demerits:

- The encryption process takes a huge amount of time.
- Multi-threaded attacks are not considered.

3.5. Sajad Homayoun [5]

Implementing a Deep Ransomware Threat Hunting and Intelligence System (DRTHIS) to distinguish ransomware from goodware and identify their families. DRTHIS utilizes LSTM and CNN, two deep learning techniques, for classification using the softmax algorithm.

220 Locky, 220 Cerber and 220 TeslaCrypt ransomware samples, and 219 goodware samples, are used to train DRTHIS. In the evaluations, DRTHIS achieves an F-measure of 99.6% with a true positive rate of 97.2% in the classification of ransomware instances.

Additionally, it is demonstrated that DRTHIS is capable of detecting previously unseen ransomware samples from new ransomware families in a timely and accurate manner using ransomware from the CryptoWall, TorrentLocker and Sage families.

The findings show that 99% of CryptoWall samples, 75% of TorrentLocker samples and 92% of Sage samples are correctly classified.

3.6. Adamović et al[6]

Ransomware was developed, based on SSH protocol for the distribution, using weaknesses produced by a human factor and encrypting files on attacked computers. After payment, the software sends an e-mail to the attacked by SMTP protocol. Therefore, targeted systems are Linux servers with opened SSH ports and weak passwords.

- The study proposes a method to develop ransomware in Python programming language while also analysing existing ransomware on Linux platforms.
- Malicious code exploits vulnerabilities resulting from the weak passwords on Linux servers.
- Ransomware employs Nmap to determine if the SSH port is open, and if it is, it enters the victim via SSH protocol.
- Files are encrypted using stream cipher based on a pseudorandom number generator.
- A simple yet inclusive approach.

Demerits:

- Besides techniques of the attack, techniques of encrypting are evolving too, this approach is very basic and might not be very efficient.
- Only suitable for Linux platforms.

3.7. Ashraf et al [9]

It can be difficult to identify and analyse suspected malware that is used for ransom. Intruders are now using sophisticated cryptographic methods to seize digital goods and then demand a

ransom. It is assumed that files typically contain certain properties, states, and patterns that a machine learning technique can identify. Thus, the focus of this work is on the detection of ransomware using feature engineering, which aids in the analysis of key characteristics and behaviours of the malware. This work's key contribution is the identification of significant and distinctive traits of ransomware that can aid in their detection. Finally, Deep Convolutional Neural Networks based on Transfer Learning have been utilised to detect Ransomware using both standard machine learning approaches and the selected features.
Two distinct datasets (static and dynamic) were constructed to undertake feature engineering and analysis. 3646 samples total (1700 ransomware and 1946 good ware) make up the static dataset.

3.8. Young et al [10]

Ransomware is malware that installs covertly on a victim's computer or smartphone, executes a crypto virology attack and demands a ransom payment to restore it. Ransomwares have been the most serious threat in 2016, and this situation continues to worsen. Because of high reward for Ransomwares, more and more Ransomware families appear, and it make us more difficultly to detect them. There are different signatures or behaviours among different families (i.e. Locky , Cerber, Cryptowall .....)or versions (i.e. CryptXXX2.0 ,CryptXXX3.0) of Ransomwares, it will be wonderful if there has a way that can detect potential Ransomware threats.
In this paper, we use deep-learning method to detect Ransomwares. At first we introduce how we label the data with different behaviours and what features we choose. And we present our model for detecting various Ransomwares and prevent them from encrypting victim's data. Experimental evaluation demonstrates that our deep-learning model can detect latest Ransomwares in high-speed network timely.

3.9. Alqahtani et al [11]

Crypto-ransomware is a malware category that targets user-related files to encrypt them and hold them to ransom. The irreversible effect of crypto-ransomware attacks entails early detection before it starts encrypting the files. Although several works have been proposed to detect such attacks at the pre-encryption phase before the encryption takes place, the main limitation of these works is the way in which they define the boundaries of the pre-encryption phase. That is, these studies determine the pre-encryption boundaries based on tracking the first call of any cryptography-related Application Programming Interface (API). However, relying on the first call of cryptography-related APIs to delineate the pre-encryption boundaries is not accurate as these APIs might be related to other (normal) tasks done by the crypto-ransomware, such as unpacking and/or decrypting the metamorphic payload, before the ransomware starts the malicious activities. In that case, the collected pre-encryption data lack many relevant pre-encryption attack patterns that come after the mistakenly-identified pre-encryption boundary. Such data insufficiency adversely affects the accuracy of the detection model and increases the rate of false alarms. To overcome such limitations, this paper proposes an early detection model (CRED) that can determine the pre-encryption boundaries and collect the data related to this phase more accurately. Unlike the extant research, the CRED model employs data-centric and process-centric detection approaches to combine both IRP and API data. These data will then be used to train a deep learning-based model. The CRED model will be evaluated using a data-benchmark collected by executing real-world crypto-ransomware samples downloaded from a widely-used repository. The performance of the detection model will be validated using the k-fold cross validation and compared against the models proposed by the existing works.

Gaps identified in existing Literature

	Dataset	Classification	Description	Demerits
Khammas, Ban Mohammed et al	-	Binary	Based on feature extraction from raw bytes, frequent pattern mining, and normalization	Much long Training period due to bigger tree branches
Samuel Egunjobi et al	-	Binary	- Classification technique of integrating both static and dynamic features - Instance-based (IB1), Random Forest (RF), Naive Bayes and Support vector machine (SVM)	Conventional Machine Learning Models used. Generally used for Experimentation purposes. Sensitive APIs
Subash Poudyal et al	-	Binary	The paper proposes a multi-level ransomware detection framework, which comprises of six major components	This approach is not based on Cryptocurrency transaction traces. Instead It focusses on the traditional file analysis method

Hairil et al	Bitcoin Heist Data	Binary	Proposal of an Artificial/Neural Network Model	Missing underlying mathematical/statistical approach. Can extend the study to achieve much higher accuracies

REQUIREMENTS

4.1. Software Requirements

S.No	Item	versions	Vendor	Description	Reference
1	Python	>= 3.7	Python Software Foundation	Python is a high-level, general-purpose programming language.	docs
2	Sklearn	0.23/0.24	New BSD License	Scikit-learn is a free machine learning library for Python featuring various algorithms.	docs
3	Windows OS	>= 7	Microsoft	Windows, is a group of several proprietary graphical operating system families.	-
4	Keras	2.7	Apache 2.0	Keras is an open-source software library that provides a Python interface for artificial neural networks.	docs
5	matplotlib	3.1	Matplotlib license	Matplotlib is a plotting library for Python and its numerical mathematics extension NumPy.	docs

Table 1: Summary of Software Components

Justification for the software usage

The project has been implemented on python programming language along with modules which support only python programming language with versions 3.7 and later. The Modules/Packages –

- Scikit Learn
- Keras
- Seaborn and
- Matplotlib

are used to run the various algorithm.

Scikit Learn supports the following algorithms –

- Logistic Regression

- K-Nearest Neighbours
- Support Vector Machine (SVC)
- Decision Tree
- Random forest
- Boosting Techniques

And Keras Modules is used to facilitate the TensorFlow Deep Neural Network Model.

P.T.O

DESIGN

All the algorithms mentioned below are being used for the classification problem.

- Logistic Regression
  - Logistic regression is a method of performing regression on a dataset that has categorical target values. It is used to transform linear combinations of the explanatory variables into probabilities.
- KNN
  - K-Nearest Neighbours - A Machine Learning algorithm which is classified under Supervised Learning. The algorithm assesses the dataset and creates new data points based on the given input(similarity). And the number of iterations for creating new data points is dependent on the value of ‘K’.

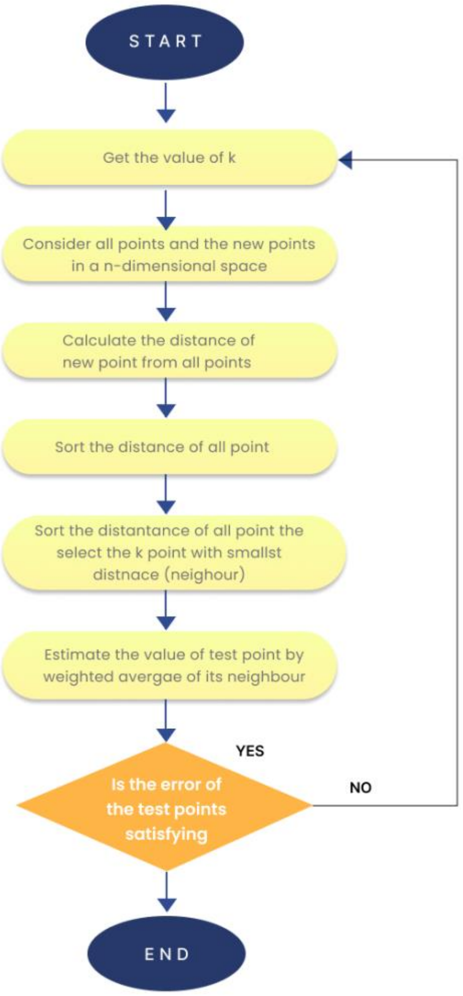
START
Get the value of k
Consider all points and the new points in a n-dimensional space
Calculate the distance of new point from all points
Sort the distance of all point
Sort the distantance of all point the select the k point with smallst distnace (neighour)
Estimate the value of test point by weighted avergae of its neighbour
YES
Is the error of the test points satisfying
NO
END

Fig no.1 Flow Diagram for K-Nearest Neighbours

- SVM
    - SVM is a technique for data classification; it can generate a nonlinear decision plane and classifies data which has non-regular distribution. It avoids attributes with greater numeric ranges dominating those with smaller numeric ranges and avoids numerical difficulties during the calculation as kernel values usually depend on the inner products of feature vectors.

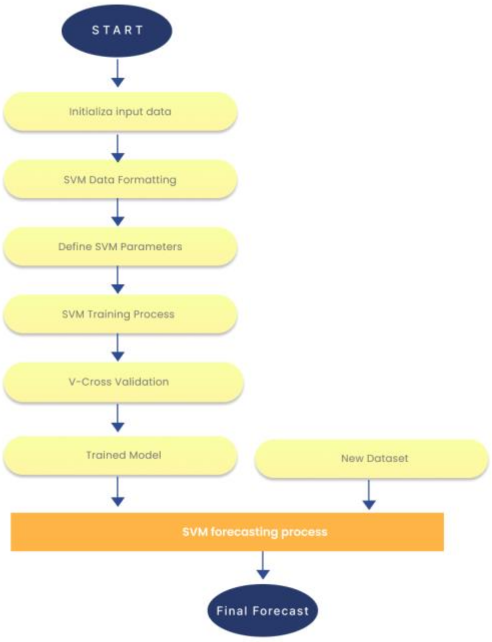

Fig no.2 Flow Diagram for a Support Vector Machine

- Decision Tree
    - Decision Tree is a supervised learning technique that may be used to solve both classification and regression problems, however it is most commonly employed to solve classification issues. Internal nodes represent dataset attributes, branches represent decision rules. Out of the various available algorithms we have implemented gini for our dataset.
- Random Forest
    - Random forest is a supervised machine learning algorithm that is commonly used to solve classification and regression problems. It creates decision trees from various samples, using the majority vote for classification and the average for regression.

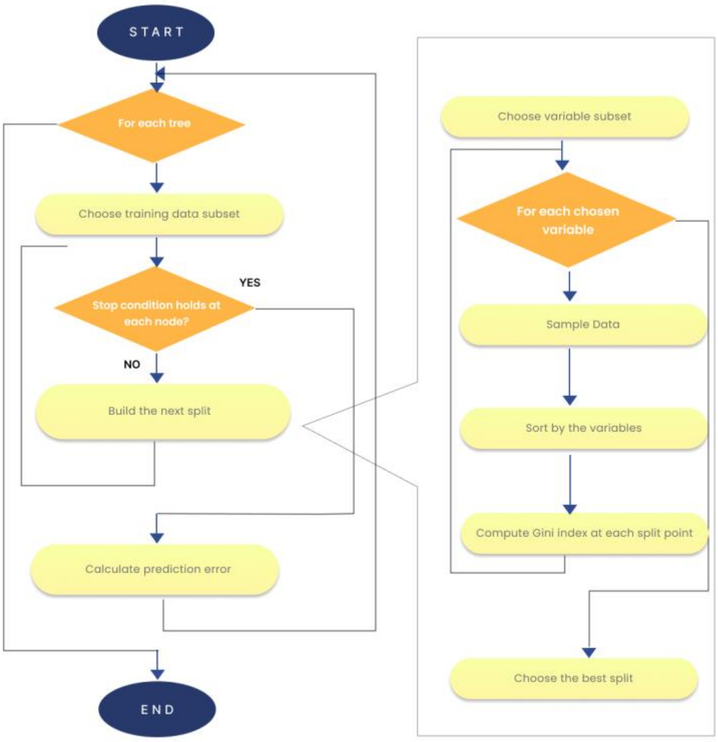

Fig no.3 Flow Diagram for Random Forest Algorithm

- AdaBoost
    - Adaboost is an ensemble learning algorithm which takes a collection of classifiers – called weak learners and using an iterative approach, combines them to produce a strong classifier.
    - Each subsequent model attempts to correct the predictions made by the model before it in the sequence.

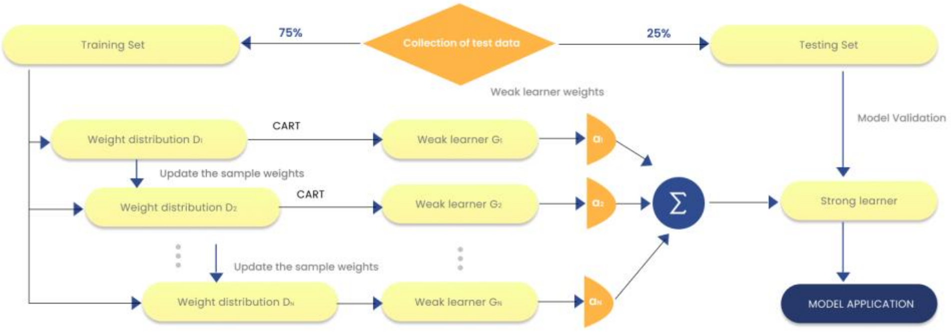

Fig no. 4 Flow Diagram for Boosting Algorithms

- XGBoost
  - Extreme Gradient Boosting, or XGBoost for short, is an open-source implementation of the gradient boosting method that is very efficient.
  - XGBoost anticipates having base learners who are consistently awful at the rest, so that when all of the predictions are added together, the bad ones cancel out and the better ones add up to make final positive predictions. In the project we have used a 100 tree estimator.
- Neural Network
  - This Algorithm is a part of Deep Learning Techniques. This Algorithm works based on different layers stacked next to each other. Each layer has a set of nodes which are otherwise called Neurons(derived from Neural Network).
  - Each layer will have its input from the previous layers and after processing through all the layers, for this project, we'll be getting a classifier as the final layer, i.e. binary classifier node.

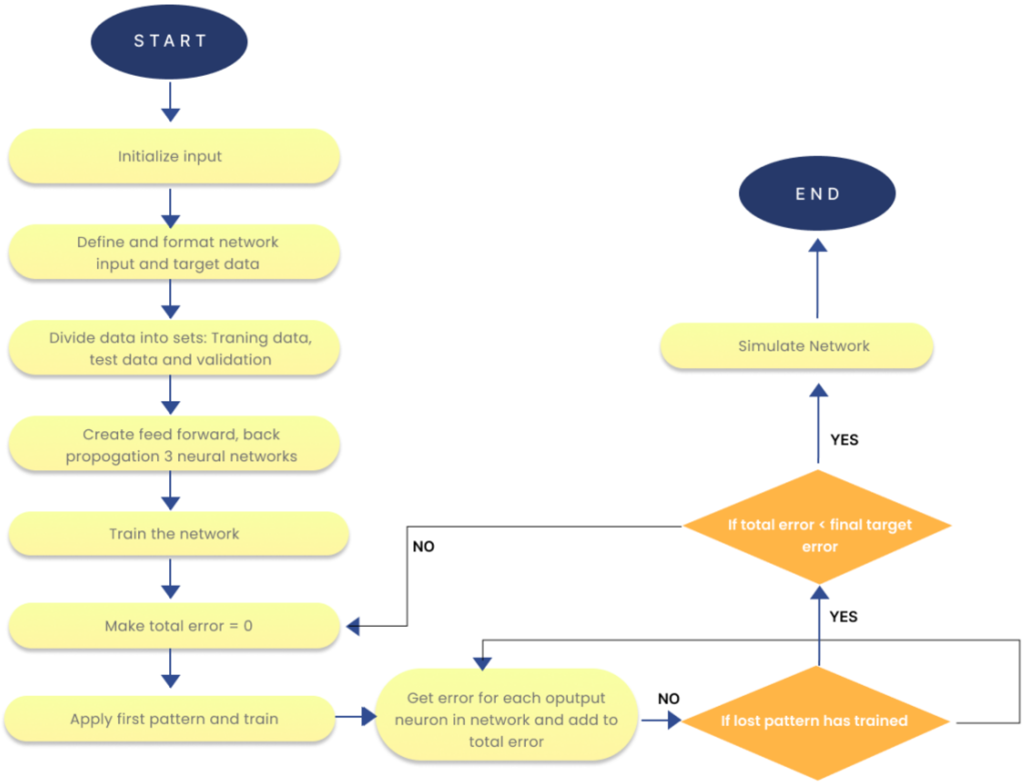
START
Initialize input
Define and format network input and target data
Divide data into sets: Traning data, test data and validation
Create feed forward, back propogation 3 neural networks
Train the network
Make total error = 0
Apply first pattern and train
Get error for each oputput neuron in network and add to total error
NO
If lost pattern has trained
YES
If total error < final target error
NO
YES
Simulate Network
END

Fig no.5 Neural Network Flow Diagram

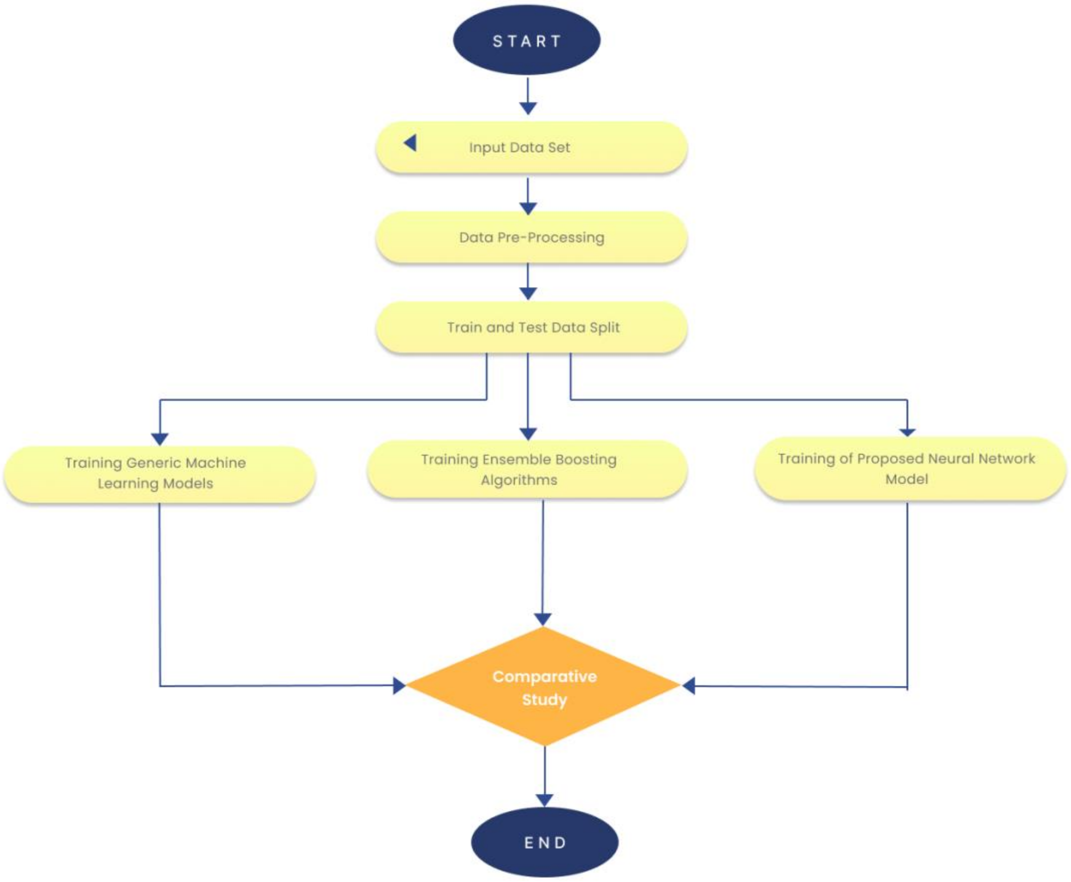
START
Input Data Set
Data Pre-Processing
Train and Test Data Split
Training Generic Machine Learning Models
Training Ensemble Boosting Algorithms
Training of Proposed Neural Network Model
Comparative Study
END

Fig no. 6 Block Diagram for Comparative Study

IMPLEMENTATION

Understanding the Dataset

We have used the dataset from the UCI Machine Learning Repository that contains parsed Bitcoin transaction graphs from 2009 January to 2018 December. The dataset has been curated from three widely adopted studies- Montreal, Princeton and Padua. Using a 24-hour time interval data network transactions have been extracted and the Bitcoin graph is formed. Network edges with less than the threshold minimum of 0.3 bitcoins have been filtered out.

This dataset consists of 3000000 observations and 10 attributes to express a specific ransomware transaction pattern. Length is designed to quantify mixing rounds on Bitcoin, wherein to hide the coin origin; transactions receive and distribute similar amounts of coins in multiple rounds with newly created addresses. Weight represents the information on the amount of transaction and expresses the merge behaviour where coins in multiple addresses are each passed through a succession of merging transactions and accumulated in a final address. The count is used to quantify the merging pattern and represents information on the number of transactions. Loop is an attribute that counts the number of transactions that split their coins, moved these coins in the network using different paths and finally merged them into a single address to be sold and converted to fiat currency. The label feature shows if a specific address of transaction belongs to ransomware class or not. The white label which indicates the address is not a ransomware address and rest of the label categories are grouped under black label.

Data Pre-processing

Features like bitcoin address, year and day were dropped off as they were irrelevant to the classification process. After analysing the data attributes via correlation analysis, we have observed that the features are independent of each other and possess minimal association; hence dimensionality reduction techniques can't be performed. After plotting the distribution of each feature, as shown in the figure below, we can observe that all the features are highly right skewed. Log transformations were applied to remove the skewness from the predictor and to convert them into normal distributions. On plotting the box plots there were no outliers in the dataset. Binary label encoding is implemented on the neighbours, length, count and looped feature attributes hence converting them into machine-readable form. Encoding as 0 indicates it as a ransomware address and 1 indicates it as a non-ransomware address.

- Neighbour values greater than log(2), it is encoded as 0 else 1
- Length values greater than log(8) are encoded as 0 else 1
- Count values greater than log(1) are encoded as 0 else 1
- Looped values greater than log(1) are encoded as 0 else 1

The label feature is observed to be imbalanced as the white label consists of more than 98.5% of the dataset. This means that a dataset is biased towards the white labelled data in the dataset; which is a typical problem. To balance the data, a new data frame of white and black labelled data was generated based on the encoded values. The number of rows to be removed and their indexes were computed and the rows were dropped.

Model Building

Further, the dataset was divided into 75% train and 25% test splits. Feature scaling was performed on the training and testing data using techniques like StandardScaler, MinMaxScaler and RobustScaler. Supervised classification machine learning models including logistic regression, KNN, SVM, Decision Tree, Random Forest, AdaBoost, XGBoost, Neural Networks were trained on this data and metrics like accuracy, precision, recall, F1 score and ROC were determined from the models for further comparative analysis.

Algorithms and their outcomes

1. **Logistic Regression**

   The input data is passed and an equation is generated based on the Multiple Regression Statistical Model. And the outcomes for the generated equation are noted.

2. **K-NN**

The input data is used as vector points and thus is used to generate new vector points. After generating K such Results, the algorithm yielded the result.

3. **SVM**

   We are training SVM with Radial Basis Function. We have two parameters for this function to follow -

   - C - Decides on the complexity of the Decision base
   - Gamma - factors the influence of a single training example
   - Gamma = 0.1, C = 1

Decision tree

- This ML Algorithm also inputs only data, similar to the Regression Algorithm mentioned before.
- After parsing a Decision tree. The following are the metrics for calculating the performance.

Random Forest

- The input data is passed to the algorithm and contains different parameters which have been set
  - n_estimators - 100
  - min_sample_leaf
  - Max_features
- After setting the parameters, the results are obtained.

AdaBoost

- As previously mentioned, this model is based on the weighted average of a combination of a set of classifiers
- Parameter n_estimator is set to 100
- After 100 iterations of multiple iterators, results are noted

Gradient Boost

- Similar to AdaBoost Algorithm's parameters, we have Gradient Boost (part of Ensemble methods)
- Parameters are being set to the default value

MLP Classifier - Neural Network

- This model optimises the log-loss function using LBFGS or stochastic gradient descent.
- Single Hidden Layer
  - i.e. parameter of no. of layers is set to 1
  - hidden_layer_sizes=(100,)
  - max_iter=500
- Multiple Hidden Layer
  - hidden_layer_sizes=(100, 100, 100),
  - max_iter=1000

- No. of layers set is 3

TensorFlow neural network

- 5 Layer Model
- Parameters set:
    - Epochs - 100 - Represents number of times the batches are getting trained
    - Batch_size = 10

Output Metrics – Evaluation

RESULTS ANALYSIS

From the implementation, we obtain the cleaned dataset and have applied it to run through various machine learning algorithms with the help of Scikit Learn.

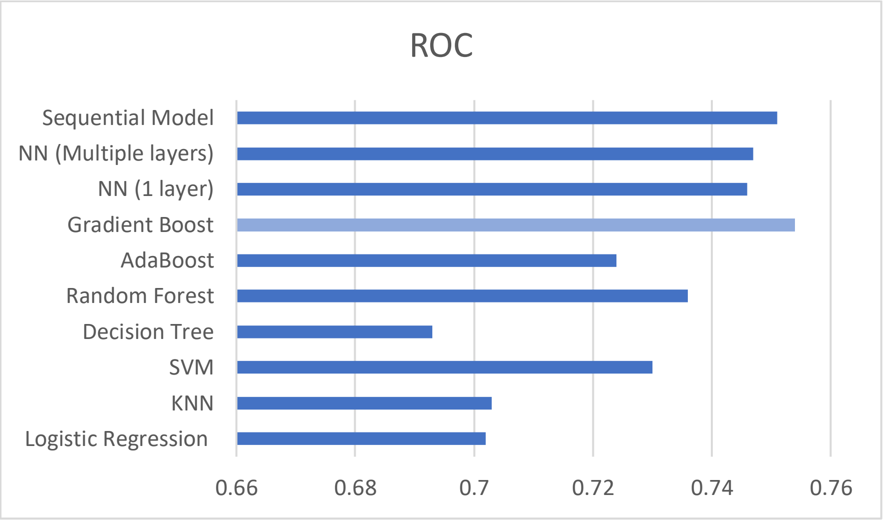

Fig no.7 ROC Values of Different Algorithms

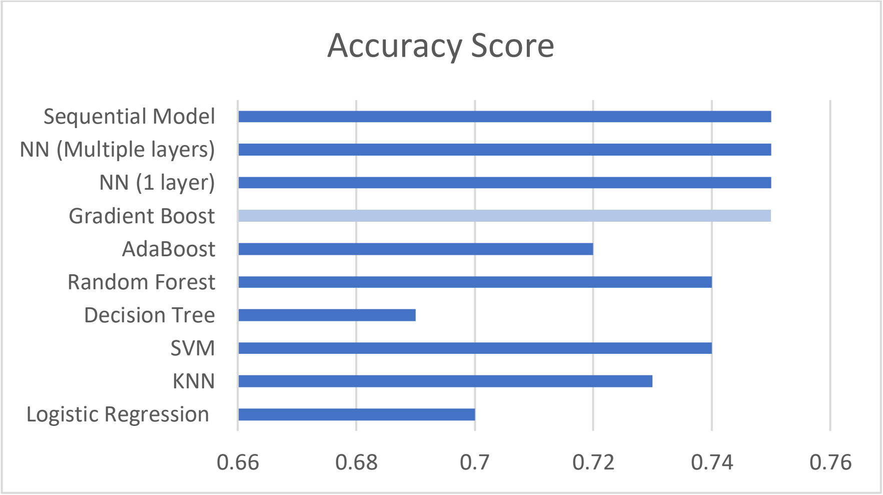

Fig no.8 Accuracy Values of Different Algorithms

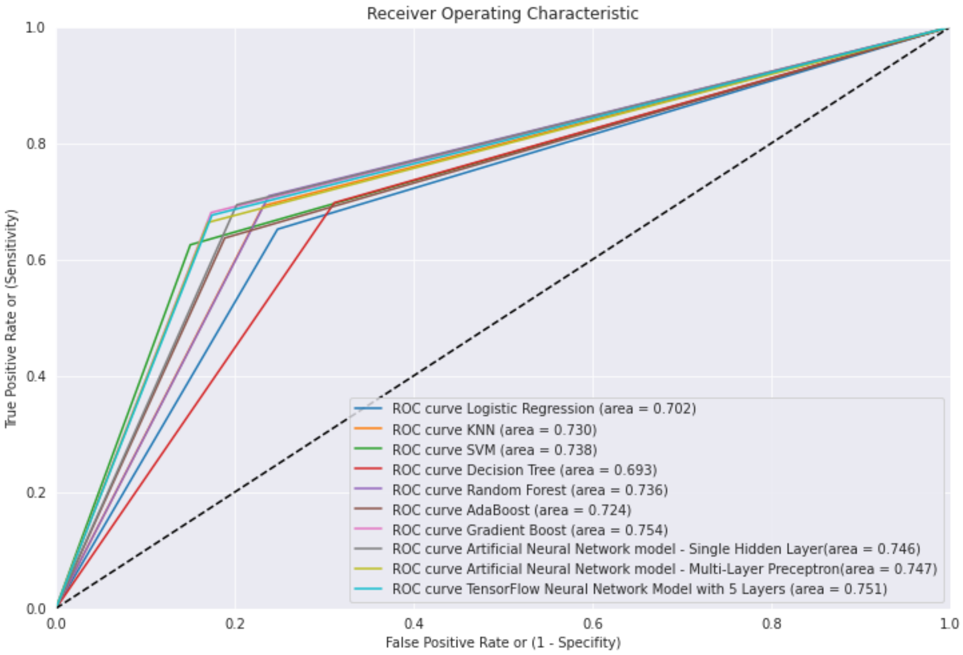

Fig no.9 ROC Curves Plotted using Matplotlib

Model	**Recall Score**	**Precision Score**	**F1 Score**	**Accuracy Score**	**ROC**
Logistic Regression	0.65	0.72	0.69	0.7	0.702
KNN	0.69	0.75	0.72	0.73	0.703
SVM	0.63	0.91	0.7	0.74	0.73
Decision Tree	0.7	0.69	0.69	0.69	0.693
Random Forest	0.71	0.75	0.73	0.74	0.736
AdaBoost	0.64	0.77	0.7	0.72	0.724
Gradient Boost	0.68	0.8	0.73	0.75	0.754
NN (1 layer)	0.69	0.77	0.73	0.75	0.746
NN (Multiple layers)	0.66	0.8	0.72	0.75	0.747
Sequential Model	0.68	0.8	0.73	0.75	0.751

Table 2. Evaluation Metrics of Various Algorithms used for Comparative study

APPLICABILITY CATEGORY

HealthCare - During a ransomware attack, information systems are shut down, and employees are denied access to critical information systems that they rely on to make decisions. Following a successful attack, clinicians are likely to detect a significant rise in CPOE-related medication mistakes.

Social Networking - Ransomware is a type of software that allows cybercriminals to extort money from you in exchange for your computer or files. Ransomware is far more dangerous than ordinary spyware or spam, because the criminals behind it can demand a large sum of money. Social media is being used by ransomware groups to publicise their attacks and increase the pressure on victims to pay the ransom.

Preventing bitcoin ransomware attacks that use hardware vulnerabilities to encrypt data of the system.

Govt Websites - Ransomware assaults on government entities are becoming more common. Cybercriminals are targeting everything from city government organisations to school districts to police departments in these attacks. Hackers are attempting to infect government computers with malware that renders them unusable, preventing users from accessing critical files. The hackers then demand a ransom from the government agency before removing the software and allowing victims to recover access to their data and devices.

Software development companies - Ransomware is usually disseminated by phishing emails that include links to malicious websites or attachments. Infection can also happen as a result of "drive-by" downloading, which occurs when a user visits an infected website and malware is downloaded and installed without their awareness. Employees and organisations are at a high risk of information leak which risks the organisation's sensitive information. For example, Kia Motors. Doppel Paymer demanded 404 Bitcoins from Kia Motors in February, equating to around $20 million in US dollars.

CONCLUSIONS

Through this paper we derive the conclusion that Ransomware can be detected and identified based on Bitcoin transactions with accuracies more than 50%. Though it is not 100% reliable, with scores nearing 75% accuracy, most of the ransom transactions can be identified. But, as the transactions are decentralised. There doesn't exist a supreme control over reverting the transaction. Thus the scope limits itself to just identification of the said transactions. From the Comparative Study we have gone through multiple algorithms and can be noticed that, Deep learning algorithms can be more effective in predicting hash addresses. From experimentation, we can find out that Gradient Boosting Algorithm provides better results than conventional algorithms along with the neural networks because of its better flexibility over ADABoost. But due to the quality of Neural Networks to learn from previous instances based on a layered approach, it is being preferred more.

REFERENCES

[1] Hairil, N. D. W. Cahyani and H. H. Nuha, "Ransomware Detection on Bitcoin Transactions Using Artificial Neural Network Methods," 2021 9th International Conference on Information and Communication Technology (ICoICT), 2021

[2] Samuel Egunjobi, Simon Parkinson, and Andrew Crampton University of Hudders eld. Classifying Ransomware Using Machine Learning Algorithms

[3] Khammas, Ban Mohammed. "Ransomware detection using random forest technique." ICT Express 6.4 (2020): 325-331.

[4] Subash Poudyal, Dipankar Dasgupta, Zahid Akhtar, Kishor Datta Gupta. A Multi-Level Ransomware Detection Framework using Natural Language Processing and Machine Learning.

[5] Homayoun, Sajad, et al. "DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer." Future Generation Computer Systems 90 (2019): 94-104.

[6] Adamović, Saša. "An Implementation of Ransomware Malicious Software in Python." Sinteza 2017-International Scientific Conference on Information Technology and Data Related Research. Singidunum University, 2017.

[7] Goldsmith, D., Grauer, K., & Shmalo, Y. (2020). Analyzing hack subnetworks in the bitcoin transaction graph. Applied Network Science, 5, 1-20.3

[8] Rivera-Castro, R., Pilyugina, P., & Burnaev, E. (2019, November). Topological Data Analysis for Portfolio Management of Cryptocurrencies. In 2019 International Conference on Data Mining Workshops (ICDMW) (pp. 238-243). IEEE.

[9] Ashraf, A., Aziz, A., Zahoora, U., Rajarajan, M., & Khan, A. (2019). Ransomware analysis using feature engineering and deep neural networks. arXiv preprint arXiv:1910.00286.

[10] A. Young and Moti Yung, "Cryptovirology: extortion-based security threats and countermeasures," Proceedings 1996 IEEE Symposium on Security and Privacy, 1996, pp. 129-140, doi: 10.1109/SECPRI.1996.502676.

[11] A. Alqahtani, M. Gazzan and F. T. Sheldon, "A proposed Crypto-Ransomware Early Detection(CRED) Model using an Integrated Deep Learning and Vector Space Model Approach," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0275-0279, doi: 10.1109/CCWC47524.2020.9031182.