*DOI: 10.15849/IJASCA.240730.04*

*Int. J. Advance Soft Compu. Appl, Vol. 16, No. 2, July 2024*
*Print ISSN: 2710-1274, Online ISSN: 2074-8523*

# Identification and Analysis of Ransomware Transactions in the Bitcoin Network

**Geethanjali Somasundaram[1], Sasikala Srinivasaga Perumal[1], Liliana Guran[2,3,]***

[1]Department of Computer Science, IDE, University of Madras Chennai, India
e-mails: sgeethanjali95@gmail.com; sasikalarams@gmail.com

[2]Hospitality Services Department, Babeș-Bolyai University, Cluj-Napoca, Romania
[3]Department of Computers, Technical University of Cluj-Napoca, Romania
email: liliana.guran@ubbcluj.ro

**Abstract**

***The advent of Blockchain and its subsequent application in creating Bitcoin has changed the world of finance. The peer-to-peer Blockchain networks, lack a third-party intermediary authority to regulate the transactions, making it vulnerable to various forms of stings. One of the most proliferate uses of crypto transactions is for the ransom payment made by victims of ransomware attacks. Owing to the varied nature of the ransomware attacks, coupled with the decentralized nature of Blockchain, tracking and guarding against such attacks is still a challenge. One way to prevent ransomware attackers from easily benefitting from such crypto transactions is to identify them and avert any payment to those attackers. In this paper, the impact of three ensemble classification algorithms – Random Forest, XGBoost and Balanced Bagging are studied to correctly classify ransomware payments from existing Bitcoin transaction data, to identify the attackers' addresses and possibly suspend them from taking part in any transactions. The outcomes of the three algorithms are compared with each other based on various indicators. From the experimental results, it could be concluded that Balanced Bagging Classifier demonstrated better performance with an accuracy of 98.41%.***

**Keywords:** ***Blockchain, Classification, XGBoost, Random Forest, Balanced Bagging, Cryptocurrency***

## 1 Introduction

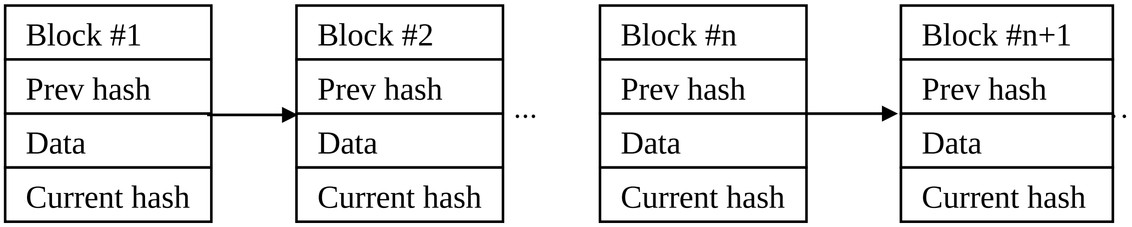
The term cryptocurrency has now become a household name. Bitcoin (2021), the forerunner of all cryptocurrencies, is a peer-to-peer digital currency that can be transmitted over the internet to facilitate any financial transactions that take place using hard cash. This has enabled two parties to directly transact funds without the need of a third party (Nakamoto 2008). Such transactions making use of digital, or cryptocurrencies are done in a decentralized manner, where there is no

***Received 10 April 2024; Accepted 15 June 2024***

regulating authority to maintain and manage neither the funds nor the corresponding transactions and are maintained in a public ledger, called as Blockchain.

Blockchain is maintained in a trustless environment and stores all transactions in a chronological manner marked by timestamps (Nakamoto 2008). Any participant of the network can verify the transactions that take place in the network via cryptographic Proof of Work (PoW) (Wu et al. 2008). The idea of Blockchain was first implemented in 2009. All the transaction details are stored in a block and a chain of blocks make up the Blockchain network. New blocks can be added to the chain of existing blocks, whereas, deleting or modifying any information present in any block is highly impossible due to the presence of linked hashes as depicted in Fig. 1. Blockchain network also houses several features such as decentralized nodes to store and manage the transaction data, information persistence on a public ledger, participant anonymity, and public auditability.

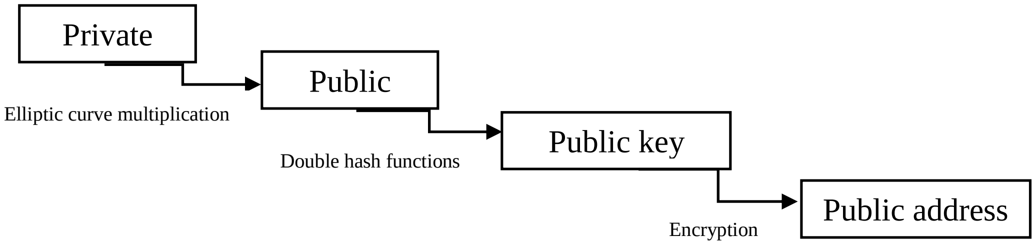
Fig. 1. Overview of the structure of blockchain network



Anonymity of the participants is well maintained in the Bitcoin network, and this basically makes it extremely difficult to trace the actual identity of the sender or the receiver (Reid and Harrigan 2013). The major advantage of the Blockchain ecosystem is the usage of digital addresses to provide total anonymity to the participants in the network. Each participant is identified by a unique Bitcoin address which is generated by a unidirectional function. This address includes a pair of keys – public and private (Herrera-Joancomartí 2017). The private key is only visible to the user. This generated key is used to authenticate transactions involving spending of the Bitcoins held by the user. An elliptic curve multiplication algorithm is applied on the private key to generate the public key. The public key upon applying a double hash function results in the public key hash. This hash is encrypted to generate the publicly visible address to which other users can send or from which they can receive Bitcoins.

Fig. 2 gives a diagrammatic representation of the address generation process. The users' personal data remains anonymous in the Blockchain ecosystem. Though there are various tools to de-anonymize the identity of a person to some extent, there is very little means to trace and recognize the identity of anyone participating in a Bitcoin transaction. Though this does have some advantages to it, the major drawback is that hackers and attackers also make use of this network to their advantage. Moreover, the absence of a cap on the number of addresses a user can create can give rise to situations where a single user can control multiple transactions and thereby engage in committing illegal activities on the network. But it is worth mentioning that tracing multiple activities to a user is still a possibility since all the transactions taking place in the network is made available publicly (Reid and Harrigan 2013).

Fig. 2. Bitcoin address generation from private key



One of the biggest challenges faced by governments and security firms across the globe is the usage of cryptocurrency to pay off ransomware attackers to retrieve encrypted information. Ransomware is a form of malicious software or malware that is premeditated to take control of the victim's machine to encrypt information and hold it at ransom (McAfee 2021). There will be a ransom demand for restoring the computer back to its previous usable state. It is transmitted over the internet to infect other systems. Upon exploiting the files in the system, the binary file is executed to encrypt all the sensitive files. Since it uses an asymmetric encryption, this results in public-private pair of keys being generated. Each victim would require the unique private key held by the attacker to reclaim their files. The key will only be shared upon the victim paying the demanded ransom via the specified online payment mechanism.

Currently, there are three types of ransomwares (Nieuwenhuizen 2017):

1. **Locker ransomware** – Work by blocking access to the computer.
2. **Crypto ransomware** – Works by making the victim's data unusable via means of encryption algorithms.
3. **Locker/Crypto ransomware** – A combination of blocking a user from using their computer while all their data is being encrypted by a malware.

Crypto ransomware is more prevalent due to the usage of strong encryption algorithms that make it nearly impossible to decrypt and retrieve the data without the availability of a key. These malwares are booming to success due to the presence of cryptocurrencies such as Bitcoin and Ethereum (2021). These greatly benefit the attackers since it is easy to quickly move the ransom money anywhere across the globe with very little to no ability to be tracked. By identifying these attackers in the network, one can report these addresses in various public portals such as Bitcoin Abuse Database (2021), Scam Alert (2021), where all the addresses used by hackers and criminals are made available to the public so as to check and prevent monetary transfers to these accounts. Users can also report cases to the local law enforcement agencies and government agencies such the FBI. In very few cases, government agencies have shown that they were able to seize the Bitcoins paid to these attackers as in the case of Colonial Pipeline Ransomware attack (Office of Public Affairs 2021).

Detecting these ransomware payments in the Bitcoin network is a challenge and one tool that can aid in this process is Machine Learning (ML). Machine Learning is believed to be a subset of Artificial Intelligence (AI) that involves building a mathematical model to ascertain and learn the diverse nature of the sample data, usually referred to as training data, which is a portion of the entire dataset (Zhang 2020). The model so developed is capable of accurately capturing the relationship between the data attributes and using that knowledge to make predictions or decisions on its own (Zhang 2020). The training phase of these models makes the algorithms adapt themselves based on experience or repetition so that their result is more accurate (El Naqa and Murphy 2015).

To establish an effective way to detect ransomware payments made to fraudsters' Bitcoin addresses and thereby blocking those addresses from receiving future payments, classification algorithms can be applied to identify and classify ransomware payments in the Bitcoin network. This will help mitigate financial losses to ransomware attacks. The central goal of this work is to discover and classify payments happening on the Bitcoin network based on transaction patterns. To this end, a publicly available dataset, from UCI, tagging several Bitcoin transactions as normal or ransomware payments was downloaded. The dataset was subject to various classification models to assess the performance and find a model which best classifies the data. Three ML algorithms were chosen namely, Random Forest, XGBoost and Balanced bagging classifier. These models were selected for this study due to their performance in our pilot comparative studies. Though the three algorithms are variations of the decision tree algorithm, their diverse functionality provide a range of results which are compared in this work. This paper also gives a detailed information as to how each of these algorithm work and enables easy comparison based on the pseudocode.

The rest of the paper is systematized as follows: Section 2 is dedicated to providing a review of related works in the identification and classification of ransomware and other attacks in the digital financial transaction platform. Section 3 deals with the methodology adopted in this research work. Dataset is given in Section 4 and the results and evaluation are discussed in Section 5. Lastly, the conclusion and future works are provided in Section 6.

## 2 Literature Review

Several researchers have proposed various ways to identify ransomware attacks and alleviate the impact it causes. Few related works are discussed, highlighting the problem statement, the mechanism to solve the problem and the limitations of these works. Table 1 shows a consolidated description of the reviewed papers.

Yin and Vatrapu (2017) tried to decode the proportion of cybercrime activities that take place in the Bitcoin ecosystem. The Bitcoin transaction data obtained was subject to various preprocessing techniques. Selected features were supplied to the following thirteen models namely, Linear Regression, Linear Discriminant Analysis, k-Nearest Neighbor Classifier (KNN), Classification And Regression Trees (CART), Support Vector Machine Classifier (SVM), Multi-Layer Perceptron Classifier (MLP), Naïve Bayes Classifier (NB), Random Forest Classifier (RFC), Extremely Randomized Forests Classifier (ETC), Bagging Classifier (BGC), Gradient Boosting Classifier (GBC), AdaBoost Classifier and Stochastic Gradient Descent Classifier (SGD) respectively (Yin and Vatrapu 2017). Among these thirteen classifiers the top four best performing classifiers and their Cross Validation (CV) accuracy were reported to be RFC with 77.38%, ETC with 76.47%, BGC with 78.46%, and GBC with 80.76% respectively. Finally, the proportion of cybercrimes to the total transactions taking place in the Bitcoin environment was predicted as 29.81% by BGC and as 10.95% by GBC. With respect to ransomware transaction, BGC classified 19.15% of transaction in the network as ransomware payments, while GBC classified the same as 5.28%.

Al-rimy et al. (2019) proposed an ensemble model, made up of three modules, to discover crypto ransomware attacks. They were iBagging module, Enhanced Semi-Random Subspace module and base classifiers. iBagging (Al-rimy et al. 2019) is an incremental bagging approach to build data subsets from the existing dataset. This helps build up the data that could be supplied to the classifier even when a previously unknown crypto ransomware attack tries to take place. Joint Mutual Information was utilized to rank and thereby select the primary features required to make predictions (Alshemmari, 2024). The Enhanced Semi-Random Subspace (ESRS) (Al-rimy et al. 2019) selection is a method that tries to extract informative features from the selected primary features, to enable the model to make accurate predictions. ESRS ensures that the variety of the data is maintained in the selected subspace. The output of this is fed to the ensemble classifier, which is made up of SVM Classifier, Logistic Regression, RFC, Decision Tree Classifier, AdaBoost Classifier, and MLP Classifier. The authors used Grid Search to select the prominent combination of classifiers. They also implemented a majority voting scheme to make the final decision. The overall average accuracy of this model was 96.8%. The only drawback in this model is the repetition of features across different subspaces which could have a solid impact on the detection accuracy.

Yazdinejad et al. (2020) proposed a model that utilizes Long Short-Term Memory (LSTM). LSTM is deliberated to be a form of Recurrent Neural Network (RNN). This deep learning model used opcodes of various cryptocurrency applications that can be executed on Windows systems to identify and classify ransomware payments. The acquired opcodes were filtered using tokenization and converted to corresponding numeric values using embedding. The reduced data - 448 hidden units and 512 unique opcodes were fed to the LSTM with Adam optimizer to update the weights of the neural network layers. Finally, a 10-fold CV was applied to evaluate the model. Among various configurations of the LSTM model, the optimum accuracy achieved was 98.25%. The result was also compared with traditional ML models such as Random Forest, SVM, NB, MLP, KNN, AdaBoost and Decision Tree and was found to be higher.

Dalal et al. (2021) came up with a model to identify miscreants involved in ransomware and gambling in the Bitcoin ecosystem. The authors devised a transaction graph modeling the address-to-address data. This is then converted to an entity graph which models actor-to-actor transactions by identifying all the addresses belonging to a particular actor by clustering local data. Supervised ML algorithms are then applied. The generated graph is broken into six sub-graphs and each sub-graph is fitted into six different classifiers. The result of each of these is then combined by using Stacking, an ensemble technique that results in a stacking probability. In the final stage, a stacking-bagging model called as meta classifier is created. This model uses Linear Regression and the output from stacking is fed into the meta classifier to get the final prediction. CV is used to determine the accuracy of this model which was reported to be around 96% and 99%.

In the work done by Agarwal et al. (2021), the authors have come up with measures to identify and classify malicious accounts in a permission-less Blockchain networks. Their study focused on the Ethereum main net Blockchain (Ethereum 2021) from which the authors collected the transaction data pertaining to gambling. They performed a time-series analysis on various features to identify the graph based temporal features that describe the behaviors of malicious agents. Primitive features like in-degree, out-degree, balance, neighbors are used in this process. As a result, they identified 28 features amongst the total 400. Random-under sampling was utilized to balance the highly imbalanced data. The data was then divided into multiple sub-datasets and were fed to TPOT, an autoML tool that was supplied with all the supervised ML algorithms. The tool identified as the best algorithm the one that gave the best balanced accuracy. From the experiment, they identified that the Extra Trees Classifier performed the best with 88.7% balanced accuracy. The classifier was validated on unseen data, for which it provided accuracy as low as 50%. This was identified to be due to the evolving characteristics of the malicious accounts. The sub-datasets were also tested on few unsupervised algorithms and k-means outperformed the others in correctly clustering malicious accounts. Moreover, it was able to cluster unseen data better than the supervised learning algorithm. Finally, the authors were also able to model the behavior changes of Ethereum accounts to identify malicious and benign actors.

Al-Haija and Alsulami (2021) presented a Bitcoin transaction predictive system that utilizes a Shallow Neural Network (SNN) and an Optimizable Decision Tree (ODT). To detect verified and anomaly transactions and perform a binary classification, an ensemble classifier for the ODT model or a Sigmoid classifier for the SNN model is used. When a multi-class classification task is at hand, an ensemble ODT (Al-Haija and Alsulami 2021) and Softmax classifier for the SNN (Al-Haija and Alsulami 2021) model are used. The accuracy for binary classifier was 99.9% and that of multiclass classifier was 99.4%.

Another notable work is the Host-based Intrusion Detection System (HIDS) built using Modified Vector Space Representation (MVSR) N-gram and Multilayer Perceptron (MLP) model for securing the Internet of Things (IoT), based on lightweight techniques and using Fog Computing devices (Khater et al., 2021). To maintain the lightweight criteria, the feature extraction stage considers a combination of 1-gram and 2-gram for the system call encoding. In addition, a Sparse Matrix is used to reduce the space by keeping only the weight of the features that appear in the trace, thus ignoring the zero weights. Subsequently, Linear Correlation Coefficient (LCC) is utilized to compensate for any missing N-gram in the test data. In the feature selection stage, the Mutual Information (MI) method and Principal Component Analysis (PCA) are utilized and then compared to reduce the number of input features. Following the feature selection stage, the modeling and performance evaluation of various Machine Learning classifiers are conducted using a Raspberry Pi IoT device.

A very recent work on an automated behavior-based detection model using Particle Swarm Optimization (PSO), a wrapper-based feature selection algorithm is analyzed (Abbasi et al., 2023). This model is used to efficiently classify ransomware transactions. The proposed method gave similar results in binary classification as that of the base work. However, it did show an improved performance in multiclass classification problems.

Table 1. Techniques and Algorithms utilized in the review papers

| Paper | Ransomware | Cryptocurrency | Data Mining Task | Technique(s) | Algorithm(s) | Efficiency (highest) |
|---|---|---|---|---|---|---|
| (Yin and Vatrapu 2017) | Yes | Yes | Classification | Multiple techniques | Multiple algorithms | 80.76% |
| (Al-rimy et al. 2019) | Yes | No | Classification | iBagging + ESRS + Ensemble classifier | SVM, Logistic Regression, RFC, Decision Tree, AdaBoost, KNN and MLP | 96.8% |
| (Yazdinejad et al. 2020) | Yes | Yes | Classification | Deep learning | LSTM | 98.25% |
| (Dalal et al. 2021) | Yes | Yes | Classification | Entity graph modeling + Stacking-Bagging model | Ensemble classifier and Meta classifier | 96% - 99% |
| (Agarwal et al. 2021) | No | Yes | Classification, Clustering, Behavior Analysis | Time series analysis + PCA + Cosine similarity | Extra Trees Classifier and others for classification | 88.7% in classification, silhouette score of |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | DBSCAN, HDBSCAN, OneClassSVM, K-Means for clustering | 0.356 in clustering |
| (Al-Haija and Alsulami 2021) | Yes | Yes | Classification | Binary and Multiclass classification | SNN and ODT | 99.9% and 99.4% |
| (Khater et al., 2021) | No | No | Classification | PCA, LCC | MVSR N-gram and MLP | 96% |
| (Abbasi et al., 2023) | Yes | No | Classification | Wrapper-based feature selection | PSO | 97.48% |

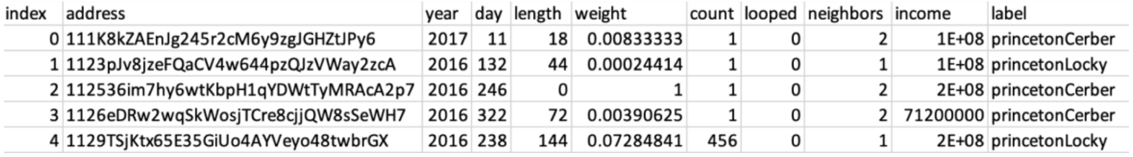
# 3 Research Methodology

This section discusses the methodology adopted in this work. The ransomware identification process consists of two stages, namely data acquisition and ML classifier modeling. The overall flow of work is depicted in Fig. 4.

## 3.1 Data acquisition

The dataset (UCI Machine Learning Repository 2020) used in this work was downloaded from UCI Machine Learning Repository. This dataset was developed from 10 years of payment transaction data in the Bitcoin network since its inception in 2009. A sample of 5 rows of data is listed in Fig. 3.

Fig. 3. Sample data from the Bitcoin Heist dataset



| index | address | year | day | length | weight | count | looped | neighbors | income | label |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 111K8kZAEnJg245r2cM6y9zgJGHZtJPy6 | 2017 | 11 | 18 | 0.00833333 | 1 | 0 | 2 | 1E+08 | princetonCerber |
| 1 | 1123pJv8jzeFQaCV4w644pzQJzVWay2zcA | 2016 | 132 | 44 | 0.00024414 | 1 | 0 | 1 | 1E+08 | princetonLocky |
| 2 | 112536im7hy6wtKbpH1qYDWtTyMRAcA2p7 | 2016 | 246 | 0 | 1 | 1 | 0 | 2 | 2E+08 | princetonCerber |
| 3 | 1126eDRw2wqSkWosjTCre8cjjQW8sSeWH7 | 2016 | 322 | 72 | 0.00390625 | 1 | 0 | 2 | 71200000 | princetonCerber |
| 4 | 1129TSjKtx65E35GiUo4AYVeyo48twbrGX | 2016 | 238 | 144 | 0.07284841 | 456 | 0 | 1 | 2E+08 | princetonLocky |

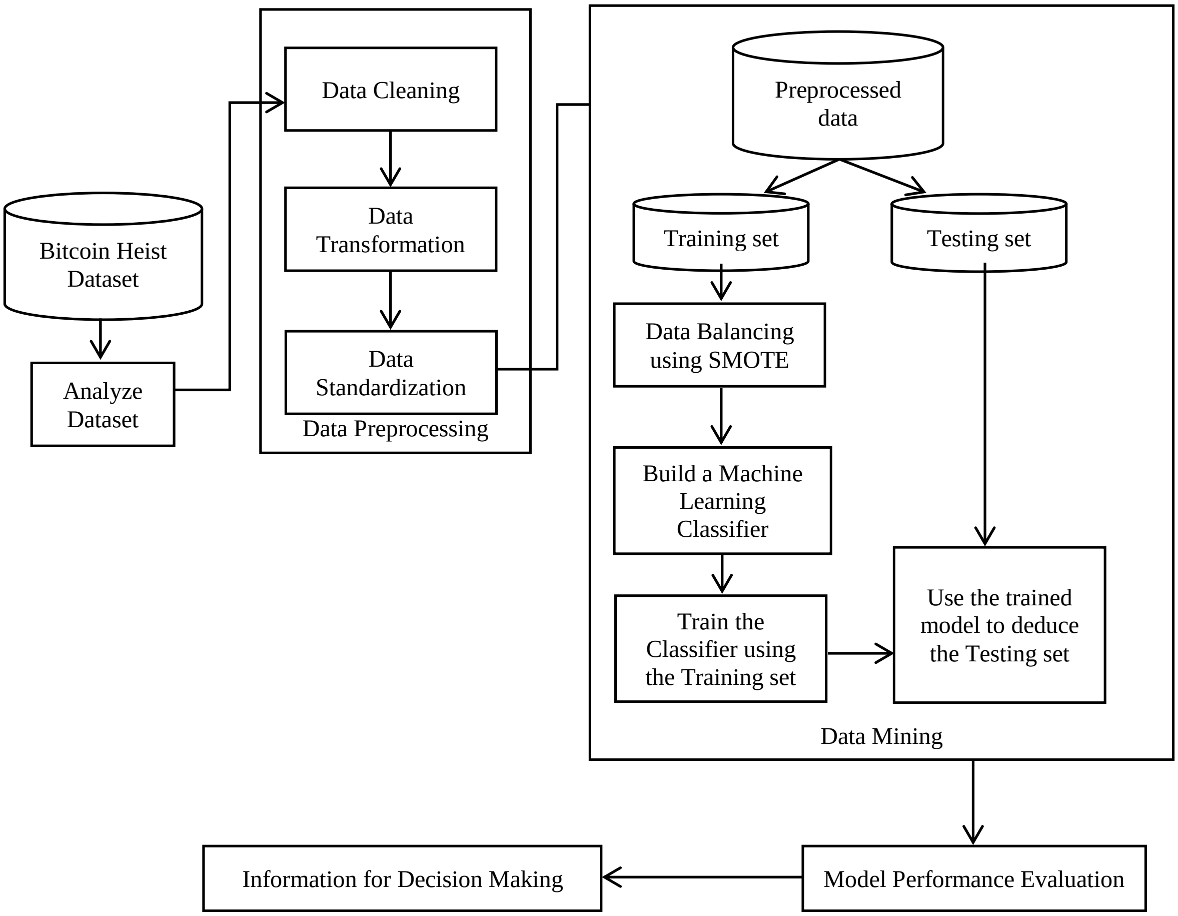
## 3.2 Dataset

The Bitcoin Heist dataset was downloaded from UCI Machine Learning Repository (2020) and analyzed. The dataset was constructed using addresses that were mined within a 24-hour window to better track the movement of the coins in the network. Six important features pertaining to an address '**u**' were estimated from the data. Each of these features was carefully chosen to explain the obscure behavior of ransomware payments. The six features are listed below (Akcora et al. 2019).

- **Length** – which indicates whether the specified address '**u**' is the output address of a starter transaction (length = 0) or not (length > 1). A length of 1 or more indicates how many non-starter (intermediate) transactions have taken place before the coin ended up in this address.
- **Count** – indicates the number of starter transactions which are associated to the address '**u**'.
- **Loop** – is indicative of how many starter transactions are connected to the specified address '**u**' in more than one directed path.
- **Weight** – is the sum of the fraction of coins that have originated from some starter transaction and ended up in '**u**'. This parameter is not concerned with the amount of coins being transacted.
- **Neighbors** – indicates the number of transactions which have '**u**' as their output address.
- **Income** – denotes the total number of coins '**u**' has received from various transactions.

Finally, the dataset also includes the dependent feature – label, which denotes what type of address '**u**' is, that is, whether the address is used by ransomware attackers or not. The downloaded dataset has 2,916,697 records with 10 columns. 8 of the 10 columns are numerical while 2 (address and label) are categorical in nature. The label column indicates the number of white (normal) and ransomware transactions (shown in Fig. 5).

Fig. 4. Overview of the methodology


Bitcoin Heist Dataset
Analyze Dataset
Data Cleaning
Data Transformation
Data Standardization
Data Preprocessing
Preprocessed data
Training set
Testing set
Data Balancing using SMOTE
Build a Machine Learning Classifier
Train the Classifier using the Training set
Use the trained model to deduce the Testing set
Data Mining
Model Performance Evaluation
Information for Decision Making

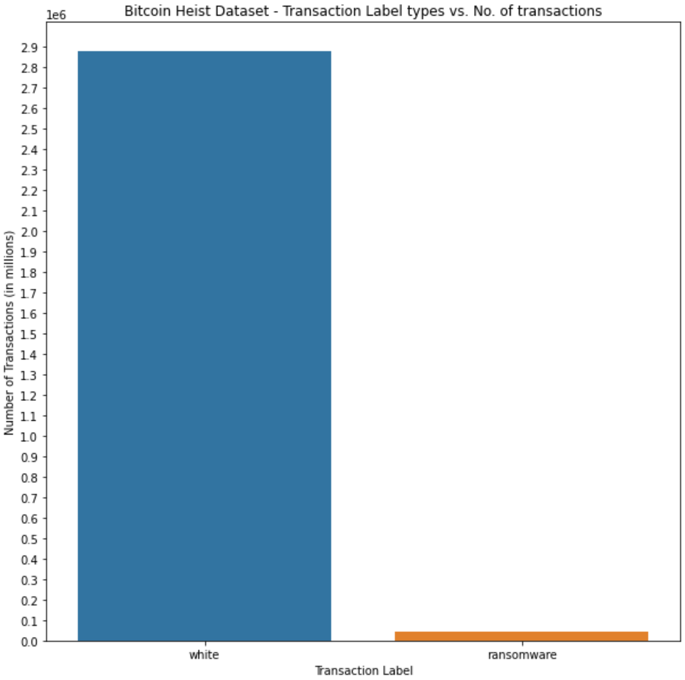
## 3.3 Modeling the Machine Learning Classifier

The CSV dataset was first analyzed and imported into the Jupyter Notebook environment and was subject to various preprocessing techniques.

- **Data cleaning**: There were no missing values in the dataset and hence no cleaning was required.
- **Data transformation**: In order for the classifier to work the data, the categorical 'label' column had to be converted into its numeric equivalent. Each ransomware label is represented as 1 while normal transactions are represented as 0 using Integer Encoding. For the address field, Label Encoder was used to encode the categorical address values.
- **Data balancing**: Due to the presence of a huge imbalance between the two class labels (0 or normal/white transaction and 1 or ransomware transaction), as shown in Figure 4, the training dataset had to be balanced before being fed to the classifier to ensure that the ML model doesn't over-fit the classifier based on the majority class. Synthetic Minority Oversampling Technique commonly known as SMOTE was utilized to oversample the minority class. SMOTE works by generating synthetic samples in the feature space which belong to the minority class (He and Ma 2013). This generates a balanced class distribution among the data instances.
- **Data standardization**: Standardized data typically rescales the data to have a mean 0 and unit variance, that is, a standard deviation of 1 (Standard Scaler Documentation 2021). Standard Scaler was used for this purpose. Although it is worth mentioning that some classifiers have shown to be unaffected by the scaling operation.

Three different classifiers were built to identify and classify ransomware transactions from the dataset, namely Random Forest classifier (RF), XGBoost classifier (XGB) and Balanced Bagging classifier (BB).

Fig. 5. Plot of Transaction label vs. No. of transactions



### 3.3.1 SMOTE method

Synthetic Minority Oversampling Technique (Chawla et al. 2002) is an approach where highly imbalanced data points can be balanced by oversampling the minority class. As the name suggests, synthetic data points are created, rather than the conventional oversampling method where data points are sampled with replacement.

The k-nearest neighbors of a single data point are identified, and lines are drawn from the point to any/all of its neighbors. Synthetic data points are introduced along these lines based on the extent of oversampling required. To generate these synthetic points along a line, the difference between the two points connected by the line is determined. This value is multiplied by a random integer between the range 0 and 1 and to this result the feature vector representing the data point is added. The final result is the value of a point, along the line.

**Algorithm for SMOTE:**

Given a dataset $\boldsymbol{D}$ with $\boldsymbol{T}$ minority samples, $\boldsymbol{N}$% amount of SMOTE, $\boldsymbol{k}$ nearest neighbors,

1. If $\boldsymbol{N} < \mathbf{100}$%, then randomize all the $\boldsymbol{T}$ minority class samples and set $N = 100$.
   $T = \left(\frac{N}{100}\right) x\, T$
2. Compute $\boldsymbol{N}$ as an integral multiple of 100.
   $N = (int)\left(\frac{N}{100}\right)$
3. For every sample $\boldsymbol{i}$ in $\boldsymbol{T}$, compute the $\boldsymbol{k}$ nearest neighbors and:
   a. Generate $\boldsymbol{N}$ synthetic data points as follows:
      i. From the $\boldsymbol{k}$ neighbors choose a random neighbor $\boldsymbol{nn}$
      ii. Compute $\boldsymbol{dif}$ the difference between the feature vectors of $\boldsymbol{nn}$ and that of $\boldsymbol{i}$.
      iii. Find a random number $\boldsymbol{g}$ between 1 and 0 and multiply it with $\boldsymbol{dif}$ and to the result add the feature vector of $\boldsymbol{i}$. This generates a single synthetic data point $\boldsymbol{S}$.
      $S = i + g\; x\; diff$

### 3.3.2 Random Forest Classifier

Random Forest Classifier (2021) is a supervised ML technique used in classification tasks that contains a collection of tree-structured classifiers defined as $\{\boldsymbol{h}(\boldsymbol{x},\ \boldsymbol{\theta_k}),\ \boldsymbol{k} = \mathbf{1}, \mathbf{2}, \ldots\}$ where $\{\theta_k\}$ are independent identically distributed random vectors (Breiman 2001). The final prediction $\boldsymbol{y}$ is determined by each tree $\boldsymbol{k}$ casting a vote and choosing the most popular class $\boldsymbol{C}$ at input $\boldsymbol{x}$.

$\boldsymbol{y} = \boldsymbol{majority\ vote}\ \{\boldsymbol{C_k}(\boldsymbol{x})\}, \boldsymbol{k} = \mathbf{1}, \mathbf{2}, \ldots$

RF is an ensemble method that utilizes multiple decision trees that operate in cohesion (Shah et al. 2020). A decision tree is a predictive model that represents the data graphically in the form of trees, where each node represents a predicate on an attribute that defines the nodes in the next level. Trees with low error rates act as strong classifiers.

RF overcomes the over-fitting problem of a single decision tree by having reduced variance. This is achieved by utilizing Bagging or Bootstrap aggregating technique on each tree (Breiman 2001). Bagging ensures that though the predictions of individual trees are subject to noisy data, the average of multiple trees is not affected by noise, provided these trees are uncorrelated. Another great feature in this classifier is the ability to use Out-Of-Bag score (OOB) to take data samples from the training set with replacement. Each tree will be tested on 1/3rd of the training dataset that was not used in building that tree. This basically makes the requirement of a separate test dataset unnecessary.

**Algorithm for Random Forest Classifier:**

Given a training set **D** with **n** instances and **d** attributes:

1. Construct **N** bootstrap samples $\mathbf{D_i}$ from the training set **D** by selecting **n** sample instances with replacement.
2. For each of these **N** samples $\mathbf{D_i}$ learn and construct a decision tree $\mathbf{T_i}$ as follows:
   a. Randomly select a subset of **p** attributes from **d**.
   b. At each internal node of $\mathbf{T_i}$ use an impurity measure and choose the best attribute from **p** to split the node.
   c. Repeat till all the leaf nodes of $\mathbf{T_i}$ are pure i.e. containing instances of some class.
3. Aggregate the result of each decision tree $\mathbf{T_i}$ through majority voting.

### 3.3.3 XGBoost Classifier

XGBoost or extreme gradient boosting (Chen et al. 2015) is an implementation of gradient boosting decision tree algorithm (Parsa et al. 2020). It makes use of Boosting, a technique which makes use of new models to correct the errors made by older models. Newer models are built and added until no more improvement is possible. It implements the ML algorithms belonging to the Gradient Boosting framework (Friedman 2001) which allows boosting operation and by adding the models together to make the final prediction (XGBoost Documentation 2021). The use of Gradient Descent algorithm minimizes any loss when new models are being added. It makes use of Classification and Regression Trees (CART) rather than decision trees as the base estimator.

**Algorithm for XGBoost:**

Given a dataset **D** with **n** instances and **m** features,

1. Define the initial CART tree $\mathbf{F_0(x)}$ to predict the target variable y.
2. Find the residual for the tree $(\boldsymbol{g_i})$ as the difference between the actual and the predicted value of the target.
   $$\boldsymbol{g_i = l(y_i, \hat{y}_i)}$$
3. Calculate the similarity score for $\mathbf{F_0(x)}$ as
   $$\boldsymbol{SS = \frac{\left(\sum_{i=0}^{n} g_i\right)^2}{h_i + \lambda}}$$
   where $\boldsymbol{h_i}$ is the number of residuals and $\boldsymbol{\lambda}$ is the regularization hyper-parameter.
4. Using the similarity score identify the information gain of the node. This acts as the loss reduction function.
   $$\boldsymbol{Gain = Left\ leaf_{similarity} + Right\ leaf_{similarity} - Root_{similarity}}$$
   $$\boldsymbol{L_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\,\epsilon\,I_{left}} g_i\right)^2}{\sum_{i\,\epsilon\,I_{left}} h_i + \lambda} + \frac{\left(\sum_{i\,\epsilon\,I_{right}} g_i\right)^2}{\sum_{i\,\epsilon\,I_{right}} h_i + \lambda} + \frac{\left(\sum_{i\,\epsilon\,I_{root}} g_i\right)^2}{\sum_{i\,\epsilon\,I_{root}} h_i + \lambda}\right] - \gamma}$$
   where $\boldsymbol{\gamma}$ is another regularization hyper-parameter.
5. Identify and split the node with the highest gain and continue to construct the tree $\mathbf{F_0(x)}$.
6. Calculate the new residual $\boldsymbol{r_i}$ as
   $$\boldsymbol{r_i = Old\ residual + \eta\ x \sum Predicted\ residuals}$$
   where $\boldsymbol{\eta}$ is the learning rate of the model
7. Build a model $\boldsymbol{f_1}\mathbf{(x)}$ to fit to the residual $\boldsymbol{r_i}$ from the previous step.

8. Build the next boosted CART tree $\mathbf{F_1(x)}$ in an additive manner by combining $\mathbf{F_0(x)}$ and $\boldsymbol{f_1}\mathbf{(x)}$.
9. Repeat the above steps until all the CART trees are learnt and residuals are minimized.
10. The tree boosting equation can be generalized as
    $\mathbf{F_m(x) = F_{m\text{-}1}(x)} + \boldsymbol{f_m}\mathbf{(x)}$

XGB is also considered to be faster than most boosting algorithms. It supports parallel tree building, tree pruning, efficient handling of missing data and it also supports regularization to avoid model over-fitting. Another great feature that makes XGB the go to algorithm is that it has in-built provisions to perform cross validations at the end of each iteration. The XGB classifier is constructed with tested parameters to improve the classification behavior.

### 3.3.4 Balanced Bagging Classifier

Balanced Bagging classifier is a Bagging classifier (Breiman 1996) with a balancing feature added to work with highly imbalanced data. A Bagging classifier is a bootstrap ensemble of meta-estimators that fits the base classifier on each random subset of the data and finally aggregates the individual predictions of each subset either through a majority voting mechanism or by averaging. The bootstrap samples are obtained by random uniform sampling with replacement. The base estimator can be any unstable model (Breiman 1996) such as a decision tree or a neural network. In this work the base estimator was set to a decision tree.

**Algorithm for Balanced Bagging Classifier:**

Given a training set **D** with **n** instances and **d** attributes:

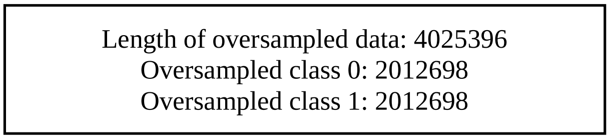
1. Construct **N** bootstrap samples $\mathbf{D_i}$ from the training set **D** by choosing **n** sample instances with replacement.
2. Obtain **N'**, the balanced bootstrap samples by applying SMOTE to each of the N samples and balance the majority and minority classes.
3. For each of these **N'** samples $\mathbf{D_i}$ learn and construct a decision tree $\mathbf{T_i}$**.**
4. Aggregate the result of each decision tree $\mathbf{T_i}$ through majority voting.

## 4. Data Analysis

The experiment is done on the obtained dataset which includes normal and ransomware transactions. Three ML classifiers were built using Python’s scikit-learn and imblearn libraries, namely random forest, extreme gradient boosting and balanced bagging. Binary classification was performed using all the three classifiers to classify the transaction in the dataset as normal or ransomware i.e., 0 or 1.

The entire dataset was employed in training and testing the three models. The data was subset in the ratio 70:30 to create training and testing sets. To ensure that the class distribution is maintained among the training and testing set, stratified splitting is done. SMOTE is applied on the training dataset to enable the classifier to learn from a balanced dataset (shown in Fig. 6). The classifier is then subject to the test dataset for classification. The results of the three classifiers’ performance are compared based on common model evaluation criteria specified in Section 4.1.

Fig. 6. Result of applying SMOTE on the training dataset



```
Length of oversampled data: 4025396
Oversampled class 0: 2012698
Oversampled class 1: 2012698
```

## 4.1 Model Evaluation

The developed models are trained and later tested on the remaining data. The performances of the models are estimated, and the following criteria listed below are used for evaluating them (Hossin and Sulaiman 2015). Table 2 provides an overview of a confusion matrix from which the performances are evaluated.

Table 2. Evaluation metrics from a confusion matrix

| **Confusion Matrix** | | **Actual Value** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Predicted Value** | **Positive** | True Positive (TP) | False Positive (FP) |
| | **Negative** | False Negative (FN) | True Negative (TN) |

- **Accuracy** – The proportion of correct predictions made by the classifier i.e., the number of correctly predicted positive and correctly predicted negative values among the entire predictions.

  Accuracy = $\frac{\text{No.of correct predictions made}}{Total\ no.of\ predictions\ made} = \frac{\text{TP+TN}}{TP+FP+FN+TN}$

- **Positive Predictive Value (PPV) or Precision or Confidence** – The proportion of positive predictions that were identified correctly i.e., the number of actual positive values out of all positive predictions.

  PPV/Precision/Confidence = $\frac{No.\ of\ correct\ positive\ predictions}{Total\ no.of\ positive\ predictions} = \frac{TP}{TP+FP}$

- **Sensitivity or Recall** – The proportion of actual positive predictions that were identified correctly i.e., the number of actual positive values that were correctly predicted out of all the actual positive values.

  Sensitivity/Recall = $\frac{No.\ of\ correct\ positive\ predictions}{Total\ no.of\ actual\ positive\ values} = \frac{TP}{TP+FN}$

- **F1 score** – The harmonic mean of precision and recall values.

  F1 score = $\left(\frac{recall^{-1}+precision^{-1}}{2}\right)^{-1} = 2\left(\frac{recall\ .\ precision}{recall+precision}\right)$

- **Balanced Accuracy Score –** In binary classification, it represents the average of the recall of both the classes when there is an imbalance in the dataset.

  Balanced Accuracy Score = $\frac{recall_{class1}\ x\ recall_{class2}}{recall_{class1} + recall_{class2}}$

- **Matthew's Correlation Coefficient (MCC) –** Used in binary classification to find the correlation between the true values and predicted values.

  Matthew's Correlation Coefficient = $\frac{TP\ x\ TN - FP\ x\ FN}{\sqrt{(TP+\ FP)(TP+FN)(TN+FP)(TN+FN)}}$

- **Geometric Mean Score (G-measure or G-mean) –** Gives the geometric mean between precision and recall.

  G-mean = $\sqrt{precision\ x\ recall}$

## 5. Results

This segment publishes the results of the three classifiers with their performance in accurately identifying and classifying Bitcoin transactions. The results are published in Table 3.

Table 3. Model performance

| **Model** | **Accuracy (%)** | **TP** | **FP** | **FN** | **TN** | **Precision** | **Recall** | **F1 score** | **AUC** |
|---|---|---|---|---|---|---|---|---|---|
| **RFC** | 69.488234 | 596960 | 265626 | **1355** | **11069** | 0.04 | **0.89** | 0.08 | 0.86 |
| **XGB** | 93.839957 | 813942 | 48644 | 5257 | 7167 | 0.13 | 0.58 | 0.21 | **0.91** |
| **BBC** | **98.406075** | **856326** | **6260** | 7687 | 4737 | **0.43** | 0.38 | **0.40** | 0.89 |

The accuracy, confusion matrix, precision, recall, F1 score are estimated by executing the model. The Area Under Curve (AUC) for all the three classifiers is also calculated and presented in Figs. 7, 8, 9.

The Receiver Operating Characteristic (ROC) Curve along with the Area Under Curve (AUC) for all the three classifiers is shown below. From the metrics in Table 4 and the ROC Curves shown in Figs. 7, 8, 9 it is evident that among the three classifiers, balanced bagging performs well since it has the highest accuracy as well as precision, indicating that this model got most of the predictions right as compared to the other two models. This is again palpable from the high value of F1-score.

Fig. 7. ROC-AUC Curve for Random Forest Classifier

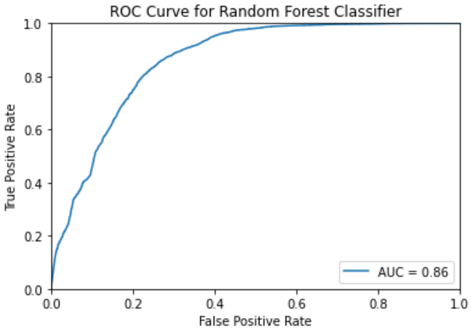


Fig. 8. ROC-AUC Curve for XGBoost Classifier

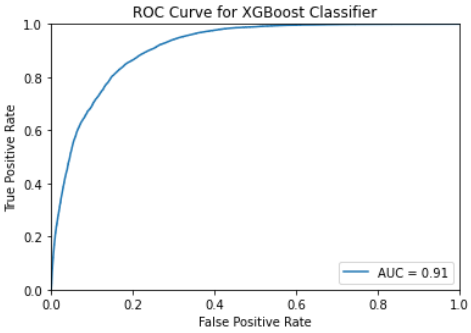
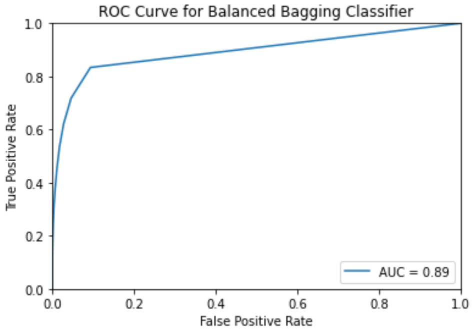


Fig. 9. ROC-AUC Curve for Balanced Bagging Classifier



Due to the imbalance of classes present in the dataset the above-mentioned evaluation metrics are alone not sufficient to determine the performance of these classifiers. Hence special evaluation metrics used exclusively for imbalanced datasets such as Balanced Accuracy, MCC and G-Mean are also calculated and compared in Table 4.

Table 4. Model performance considering the imbalanced nature of the Bitcoin Heist Ransomware Address Dataset

| Model | Balanced Accuracy | Matthews Correlation Coefficient (MCC) | G-Mean |
|---|---|---|---|
| **RFC** | **79.149771** | 0.148331 | **0.785226** |
| **XGB** | 76.023707 | 0.251986 | 0.737791 |
| **BBC** | 68.701046 | **0.397220** | 0.615233 |

## 6. Discussions

This section discusses the results and what they infer. The metrics in Table 3 and the ROC Curves shown in Figs. 7, 8, 9 show that among the three classifiers, balanced bagging performs well since it has the highest accuracy as well as precision, indicating that this model got most of the predictions right as compared to the other two models. This is again palpable from the high value of F1-score. When looking at the AUC curves, we can see that the curve of BBC has more closeness to the upper-left corner of the plot when compared to RFC and XGB. This area is typically preferred as this is where the sensitivity of the model turns 1 while the false positive rate is approaching 0 (Nahm, 2022). Yet the low recall value of BBC could be an indicator of quite a number of negative values being predicted as positive. This along with the straight line in ROC curve indicates that the model is probably subject to little overfitting.

As per the results obtained from balanced accuracy and G-mean, Random Forest classifier is said to perform better. But like F1-score, balanced accuracy and G-mean tend to ignore the impact of True Negative classifications and focuses only on the majority class which is labelled as positive class. This is overcome by MCC which uses all the four measures from the confusion matrix in determining the correlation coefficient (Chicco et al., 2021). Thus, in this work, we attribute more weightage to MCC than the other metrics.

### *Statistical Inferences*

Statistical inference is vital in research as it allows scientists to draw conclusions about a population based on sample data. By generalizing findings, testing hypotheses, and estimating parameters, it transforms raw data into meaningful insights. This process ensures that decisions and conclusions are backed by rigorous quantitative analysis, enhancing the reliability and validity of research outcomes.

In this study, we use the analysis of variance (ANOVA) technique. The following parameters are used for this computation: accuracy, precision, recall, F1-score, balanced accuracy, MCC and G-Mean. The three classifiers were also ranked in order of their performance as shown in Table 5. The alpha value (significance level) is fixed at 0.05 (95%). The following are the null ($H_0$) and alternate ($H_1$) hypotheses that were chosen for this analysis.

$H_0$: There is a significant difference in the performance of the classifiers.

$H_1$: There is no significant difference in the performance of the classifiers.

Table 5. Statistical analysis results of the performance of the classifiers

| Classifier | Mean Rank | Mean | Confidence Interval |
|---|---|---|---|
| **RFC** | 2.142857 | 0.489954 ± 0.380217 | [0.27598, 0.703929] |
| **XGB** | 2 | 0.515456 ± 0.317194 | [0.301481, 0.72943] |
| **BBC** | 1.857143 | 0.556209 ± 0.223721 | [0.342234, 0.342234] |

The estimated p-value for the ANOVA test was 0.7707. The confidence intervals are also analyzed to check for any significance. Since the p-value is greater than alpha, we reject $H_0$. Therefore, it can be safely said that the performance of the classifiers is not significantly different from each other. However, upon analyzing the mean ranking of the performance of each of these classifiers, it was observed that BBC edged the other two, though by a very small difference (as seen in Table 5). We can therefore conclude that Balanced Bagging Classifier does a comparatively better job in classifying the ransomware transactions.

The performance of the three classifiers cannot be easily determined by a single evaluation metric. This can be attributed to the highly imbalanced nature of the dataset. From the identified metrics, the best performing classifier amongst these three must be carefully chosen. Traditionally, accuracy, precision and recall are the commonly used metrics to evaluate a model where there is not much of imbalance between the classes in the dataset. But it has been corroborated that F1-score and ROC with AUC are much better metrics to ascertain the quality of ML models. But the unfair distribution of the classes in an imbalanced dataset makes it difficult to a gauge the performance of the model.

This is where MCC, Balanced accuracy and G-mean come into place. These are metrics well suited to assess the functioning of models built on imbalanced data. As per the results obtained from balanced accuracy and G-mean, Random Forest classifier is said to perform better. But like F1-score, balanced accuracy and G-mean tend to ignore the impact of True Negative classifications and focuses only on the majority class which is labelled as a positive class. This is overcome by MCC which uses all four measures from the confusion matrix in determining the correlation coefficient (Chicco et al. 2021). Thus, in this work, we attribute more weightage to MCC than the other metrics. We can therefore conclude that Balanced Bagging Classifier does a comparatively better job in classifying Bitcoin transactions as ransomware or normal based on F1-score and MCC.

## 7. Conclusions and Future Works

The growth of the Blockchain ecosystem has given rise to many uses and the financial sector has benefitted the most. The future of financial transactions now lies in the court of digital transactions, and cryptocurrencies, especially Bitcoin is becoming the forerunner in that. The primary reason for the immense popularity and success of cryptocurrencies is the integrity of the Blockchain network on which it is based. The last decade has seen a spurt of Bitcoin transactions owing to the various benefits it provides such as transparent transactions, absence of any intermediary authority, security, and user anonymity. All cyber criminals, especially ransomware attackers, are using this space to make the most without getting caught. Ransomware victims are required to transfer the ransom amount to the attacker’s address. Many a time, these transactions are routed through different addresses before reaching the final destination. This makes the money trail untraceable in most cases.

This paper aims to identify a classifier that can decode the Bitcoin addresses of ransomware attackers and prevent these addresses from receiving any payments in the future. A publicly available Bitcoin ransomware transaction dataset was employed in this study. Three different ML classifiers – Random Forest, XGBoost and Balanced Bagging were developed and trained using a part of the given dataset. These were then tested to check the efficacy of the models in correctly classifying transactions as normal or ransomware. The models were evaluated not only based on traditional ML metric such as accuracy but also using recall, precision, and F1-score. Evaluation metrics specially designed for imbalanced classes were also employed. Based on the results, it could be settled that the Balanced Bagging classifier outperformed the other two in terms of accuracy, F1-score, and MCC. Though this classifier had an MCC of almost 0.4, this can still be improved. Metrics like balanced accuracy and AUC are still low compared to the other classifiers and can be tuned better.

Moreover, the highly imbalanced nature of this dataset also makes accurate prediction a challenge, since the usage of a large amount of data could most probably result in model over-fitting the data and thereby providing inaccurate results. In the future, more ensemble models can be tested to classify the transactions. In order to improve the predictions, hyper-parameter tuning can be employed with Grid Search to identify and select the best hyper-parameters to attain maximum accuracy.

**Declaration of interest**

The authors declare no conflicts of interest.

**Acknowledgment**

The third author wants to tank Babes-Bolyai University, Cluj-Napoca, Romania for financial support.

# References

[1] Agarwal, R., Barve, S. & Shukla, S.K. Detecting malicious accounts in permissionless blockchains using temporal graph properties. *Appl Netw Sci* **6**, 9 (2021). https://doi.org/10.1007/s41109-020-00338-3

[2] Al-Haija, Qasem Abu, and Abdulaziz A. Alsulami. 2021. "High Performance Classification Model to Identify Ransomware Payments for Heterogeneous Bitcoin Networks" *Electronics* 10, no. 17: 2113. https://doi.org/10.3390/electronics10172113

[3] Al-rimy BAS, Mohd AM, Syed ZMS (2019) Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. *Future Generation Computer Systems*, Vol. 101, 476-491, DOI: https://doi.org/10.1016/j.future.2019.06.005

[4] Akcora CG, et al. (2019) BitcoinHeist: Topological data analysis for ransomware detection on the bitcoin blockchain. DOI: https://arxiv.org/abs/1906.07852v1

[5] Bitcoin (2021) Bitcoin - Open Source P2P Money. Available in: https://bitcoin.org/en/ Accessed on: December 31, 2021.

[6] Bitcoin Abuse Database (2021) Available in: https://www.bitcoinabuse.com/ Accessed on: December 28, 2021.

[7] Breiman, L. (1996) Bagging predictors. *Mach L..0arn* **24**, 123–140. https://doi.org/10.1007/BF00058655

[8] Breiman, L. (2001) Random Forests. *Machine Learning* **45**, 5–32. https://doi.org/10.1023/A:1010933404324

[9] Chawla NV et al. (2002) SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research.* DOI: https://doi.org/10.1613/jair.953

[10] Chen T et al. (2015) Xgboost: extreme gradient boosting.

DOI: https://mran.microsoft.com/

[11] Chicco, D., Tötsch, N. & Jurman, G. The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining* **14**, 13 (2021). https://doi.org/10.1186/s13040-021-00244-z.

[12] Dalal S, Zihe W, Siddhanth S (2021) Identifying Ransomware Actors in the Bitcoin Network. DOI: https://arxiv.org/abs/2108.13807v1

[13] El Naqa, I., Murphy, M.J. (2015). What Is Machine Learning?. In: El Naqa, I., Li, R., Murphy, M. (eds) *Machine Learning in Radiation Oncology*. Springer, Cham. https://doi.org/10.1007/978-3-319-18305-3_1

[14] Ethereum (2021) Available in: https://ethereum.org/en/ Accessed on: December 31, 2021.

[15] Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189–1232. http://www.jstor.org/stable/2699986

[16] He H, Yunqian M (2013) Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons.

[17] Herrera-Joancomartí, J. (2015). Research and Challenges on Bitcoin Anonymity. In: Garcia-Alfaro, J., *et al*. Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance. *Lecture Notes in Computer Science*, vol 8872. Springer, Cham.
https://doi.org/10.1007/978-3-319-17016-9_1

[18] Hossin M, Md NS (2015) A review on evaluation metrics for data classification evaluations. DOI: http://dx.doi.org/10.5121/ijdkp.2015.5201

[19] Khater, B.S.; Abdul Wahab, A.W.; Idris, M.Y.I.; Hussain, M.A.; Ibrahim, A.A.; Amin, M.A.; Shehadeh, H.A. (2021) Classifier Performance Evaluation for Lightweight IDS Using Fog Computing in IoT Security. *Electronics*, 10, 1633.
DOI: https://doi.org/10.3390/electronics10141633

[20] McAfee (2021) What is Ransomware? Available in:
https://www.mcafee.com/enterprise/en-in/security-awareness/ransomware.html Accessed on: December 21, 2021.

[21] Nakamoto S (2008) Bitcoin: A peer-to-peer electronic cash system.
DOI: https://bitcoin.org/bitcoin.pdf

[22] Nieuwenhuizen D (2017) A behavioural-based approach to ransomware detection.
DOI: https://labs.f-secure.com/

[23] Office of Public Affairs (2021) Department of Justice Seizes $2.3 Million in Cryptocurrency Paid to the Ransomware Extortionists Darkside. Available in:
https://www.justice.gov/opa/pr/department-justice-seizes-23-million-cryptocurrency-paid-ransomware-extortionists-darkside Accessed on: December 28, 2021.

[24] Parsa AB et al. (2020) Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. *Accident Analysis & Prevention*, 136, 2020, 105405, DOI: https://doi.org/10.1016/j.aap.2019.105405

[25] Random Forest Classifier (2021) Available in:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html Accessed on: November 26, 2021.

[26] Reid, F., Harrigan, M. (2013). An Analysis of Anonymity in the Bitcoin System. In: Altshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (eds) *Security and Privacy in Social Networks.* Springer, New York, NY. https://doi.org/10.1007/978-1-4614-4139-7_10

[27] Scam Alert (2021) Available in: https://scam-alert.io/ Accessed on: December 28, 2021.

[28] Shah K et al. (2020) A comparative analysis of logistic regression, random forest and KNN models for the text classification. DOI:https://doi.org/10.1007/s41133-020-00032-0.

[29] Shehadeh, H. A., Jebril, I. H., Jaradat, G. M., Ibrahim, D., Sihwail, R., Al Hamad, H., ... & Alia, M. A. (2023). Intelligent Diagnostic Prediction and Classification System for Parkinson's Disease by Incorporating Sperm Swarm Optimization (SSO) and Density-Based Feature Selection Methods. *J. Advance Soft Compu. Appl*, 15(1).

[30] Standard Scaler Documentation (2021) Available in: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html Accessed on: December 28, 2021.

[31] UCI Machine Learning Repository (2020) Bitcoin Heist Ransomware Address Dataset. Available in: https://archive.ics.uci.edu/ml/datasets/BitcoinHeistRansomwareAddressDataset Accessed on: November 19, 2021.

[32] Wu X, Kumar V, Quinlan J R, Ghosh J, Yang Q, Motoda H, Steinberg D. et al. (2008) Top 10 algorithms in data mining. DOI: https://doi.org/10.1007/s10115-007-0114-2

[33] XGBoost Documentation (2021) Available in: https://xgboost.readthedocs.io/en/stable/ Accessed on: December 1, 2021.

[34] Yazdinejad A et al. (2020) Cryptocurrency malware hunting: A deep recurrent neural network approach. *Applied Soft Computing*,Vol. 96, 106630, DOI: https://doi.org/10.1016/j.asoc.2020.106630

[35] H. Sun Yin and R. Vatrapu, "A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning," *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017, pp. 3690-3699, DOI: https://10.1109/BigData.2017.8258365

[36] Hang, XD. (2020). Machine Learning. In: A Matrix Algebra Approach to Artificial Intelligence. Springer, Singapore. https://doi.org/10.1007/978-981-15-2770-8_6

[37] Alshemmari, M. (2024). Semiotics of the Images on Social Media Signifying the Boycott of Western Products During Al- Aqsa Flood Crisis in 2023: An Analytical Study, Arab Journal for the Humanities: 167, 133-186. https://doi.org/10.34120/ajh.v42i167.619

## Notes on contributors

***Geethanjali Somasundaram*** is currently pursuing her Ph.D. in Data Analytics at the Department of Computer Science, Institute of Distance Education (IDE), University of Madras, Chennai, India. She received her M. Sc. in Information Technology (M. Sc. (I.T.)) and Bachelor of Computer Applications (B.C.A.) degrees from Stella Maris College. She is also currently holding the position of Assistant Professor in the Department of Computer Science, Stella

Maris College, Chennai, India. Her areas of interest include Data Analytics, Neural Networks and Deep Learning.

***Sasikala Srinivasaga Perumal*** holds the position of Professor in the Department of Computer Science at the Institute of Distance Education (IDE), University of Madras. Her academic qualifications include B.Sc. in Computer Science, M.C.A., SLET in Computer Science, M.Phil. in Computer Science, and Ph.D. in Computer Science. She has contributed significantly to her field with a focus on ML and Big Data Analytics. She has an impressive publication record, with 44 works featured in both Indian and international journals. Additionally, she holds two patents and has presented papers at various national and international seminars. Her active involvement in academia is evident through her organization and participation in 79 workshops and seminars. Professor Sasikala's expertise lies in the intersection of ML and Big Data Analytics.

***Liliana Guran*** is an Assistant Professor at the Faculty of Business, Department of Hospitality Services, Babes-Bolyai University, Cluj-Napoca, Romania. Her main teaching and research interests include Computer Science (Data Science, ML, IoB), and Applied Mathematics (Nonlinear Analysis, Fixed Points, Fractional Calculus, Mathematical Modeling). She has published around 80 research articles in international journals of mathematics and computer science and 6 books. Since 2016 she is Reviewer for American Society of Mathematics. She has more than 30 attendances at international conferences as a plenary, keynote, or invited speaker from around 70 international participations.