Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2023, Article ID 6274260, 10 pages
https://doi.org/10.1155/2023/6274260

Research Article

Machine Learning-Based Ransomware Classification of Bitcoin Transactions

Suleiman Ali Alsaif

Computer Department, Deanship of Preparatory Year and Supporting Studies, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia

Correspondence should be addressed to Suleiman Ali Alsaif; saalsaif@iau.edu.sa

Received 3 July 2022; Revised 11 December 2022; Accepted 12 January 2023; Published 23 January 2023

Academic Editor: Babek Erdebilli

Ransomware attacks are one of the most dangerous related crimes in the coin market. To increase the challenge of fighting the attack, early detection of ransomware seems necessary. In this article, we propose a high-performance Bitcoin transaction predictive system that investigates Bitcoin payment transactions to learn data patterns that can recognize and classify ransomware payments for heterogeneous bitcoin networks into malicious or benign transactions. The proposed approach makes use of three supervised machine learning methods to learn the distinctive patterns in Bitcoin payment transactions, namely, logistic regression (LR), random forest (RF), and Extreme Gradient Boosting (XGBoost). We evaluate these ML-based predictive models on the BitcoinHeist ransomware dataset in terms of classification accuracy and other evaluation measures such as confusion matrix, recall, and *F*1-score. It turned out that the experimental results recorded by the XGBoost model achieved an accuracy of 99.08%. As a result, the resulting model accuracy is higher than many recent state-of-the-art models developed to detect ransomware payments in Bitcoin transactions.

1. Introduction

Different forms of digital currency have been proposed and implemented over the decades. Blockchain technology allows the decentralized currency to exist and perform reliable transactions while avoiding double-spending by using a consensus algorithm based on proof of work [1, 2].

Bitcoin has now garnered much attention from researchers and investors due to its lack of regulation and criminals as well.

Recently, the Internet has been targeted by a wave of new types of malware, categorized as ransomware, which is a type of malware that once it successfully infects a target's computer system, it then encrypts files and private data so that the victim can no longer access them. The victim is then presented with a message explaining the situation and containing instructions on how to regain full access to their own system, and this usually involves paying an amount of money via Bitcoin.

There are several hacker groups and variants, also known as families, of this malware, but their defining feature is having to pay to release access to the captured data [3].

We propose to analyze several transactions collected from the Bitcoin blockchain to determine whether it is possible to easily classify each transaction as belonging to a ransomware family or not. Since cryptocurrency payments are anonymous, so finding out who sent them or when they were made is challenging. Therefore, in order to be able to recognize and categorize these transactions correctly, it is crucial that they are identified and labelled according to whether they are legitimate exchanges or trading operations.

To determine whether a transaction is malicious or benign, machine learning (ML) models are employed in this article [2, 4, 5]. To classify the different types of malicious transactions on Bitcoin, we used the BitcoinHeist ransomware dataset. Different transaction features are analyzed. Precision, accuracy, and recall of the results are evaluated.

The main contributions of this article can be summarized as follows:

(i) Showing how data balancing has an impact on ML-based predictive models for ransomware classification of Bitcoin transactions

(ii) Detecting anomalies before training the model

(iii) Testing and evaluating logistic regression (LR), random forest (RF), and Extreme Gradient Boosting (XGBoost) methods to learn the distinctive patterns in Bitcoin payment transactions

The rest of the article is organized as follows: Section 2 presents a literature review related to ransomware analysis and identification. Section 3 discusses our system modeling and specifications. The results and performance study are presented in Section 4. Finally, the conclusion and future directions of the research work are provided in Section 5.

2. Related Work

With the rapid increase in Bitcoin activity, several studies have analyzed blockchain technology from different angles [6].

The earliest developments aimed at finding the coins used in illegal activities while following the transaction network [7–9].

User identification is not required to join the network since Bitcoin provides pseudoanonymity. The authors in [10] proposed a mixing scheme to hide the coin flow in the network. Researchers revealed that some Bitcoin payments can be traced [11].

In ransomware analysis, several researchers have analyzed the networks of cryptocurrency ransomware [12, 13]. They found that hacker behavior can help us identify unknown ransomware payments.

Early studies in ransomware detection use decision rule on the amounts and time of known ransomware transactions to find undisclosed ransomware payments [14].

More recent studies are collaborative efforts between researchers and blockchain analytics companies. In [13], they identified shared hacker behavior and used heuristics to determine ransomware payments. The authors estimate that over 20,000 victims have made ransomware payments.

In [4], the authors studied the differences between ransomware families in Bitcoin trading behavior using descriptive statistical analysis. In [15], the authors used decision trees and ensemble learning to classify ransomware families. In [16], the authors proposed an approach, called NetConver, to analyze ransomware in network traffic using the decision tree (DT, J48) model. They achieved 97.1% accuracy rate. These research works are mainly targeted at detecting ransomware before it can contaminate a system and do not consider Bitcoin transactions.

3. Methodology

3.1. Bitcoin Transaction Dataset. The dataset used for this study was provided by [17] and is currently available in the UCI Machine Learning Repository. It is hosted by the University of California at Irvine [18]. Each row represents a Bitcoin blockchain transaction and contains the following set of attributes [17]:

(i) Address: string that describes the address of the transaction

(ii) Year: integer that describes the year of the transaction

(iii) Day: integer describing the day of the transaction

(iv) Length: the number of nonstarter transactions on its longest chain

(v) Count: the number of transaction starters that are linked to the address

(vi) Neighbors: the number of transactions that have this address as an output

(vii) Weight: the sum of the fraction of Bitcoin coins that come from a starter-start transaction and reach this address

(viii) Looped: the number of starter transactions that are connected to this address by more than one direct path

(ix) Income: integer that stores the amount of Satoshi (Bitcoin = 100 million Satoshi. Satoshi is the smallest unit of Bitcoin)

(x) Label: string that presents the nature of the transaction (Ransomware: 29 families and White: legitimate)

The dataset specifications are provided in Table 1. The record distribution between the ransomware families seems to be almost balanced, with 13,163 records for the first, 12,402 records for the second, and 15,848 records for the third family.

The features extracted from the dataset are as follows:

(i) The number of instances of each column

(ii) The average of each column

(iii) The standard deviation of each column

(iv) The minimum of each column

(v) The maximum of each column

According to Table 1, the transactions of the dataset constitute 28 families divided as follows:

(i) 3 ransomware categories are Princeton, Montreal, and Padua. They contain 27 families.

(ii) 1 white category which represents legitimate transactions

3.2. System Modeling. To produce a predictive model of ransomware and to classify bitcoin transactions, the engagement of supervised ML algorithms (MLAs) [19] seems to be necessary.

We are concerned with designing a learning-based self-reliant classification scheme starting by preprocessing, balancing, classifying, and evaluating.

The developed system is composed of four steps, as illustrated in Figure 1.

Table 1: Bitcoin transactions dataset statistics.

Ransomware family	Ransomware type	#Instances
Montreal	*montrealAPT*	11
	montrealComradeCircle	1
	montrealCryptConsole	7
	montrealCryptoLocker	9,315
	*montrealCryptoTorLocker*2015	55
	montrealCryptXXX	2,419
	montrealDMALocker	251
	*montrealDMALockerv*3	354
	*montrealEDA*2	6
	montrealFlyper	9
	montrealGlobe	32
	montrealGlobeImposter	55
	*montrealGlobev*3	34
	montrealJigSaw	4
	montrealNoobCrypt	483
	montrealRazy	13
	montrealSam	1
	montrealSamSam	62
	montrealVenusLocker	7
	montrealWannaCry	28
	montrealXLocker	1
	*montrealXLockerv*5.0	7
	montrealXTPLocker	8
	Total	13,163
Panuda	*paduaCryptoWall*	12,390
	paduaJigsaw	2
	paduaKeRanger	10
	Total	12,402
Princeton	*princetonCerber*	9,223
	princetonLocky	6,625
	Total	15,848
Total no. of ransomware	41,413	

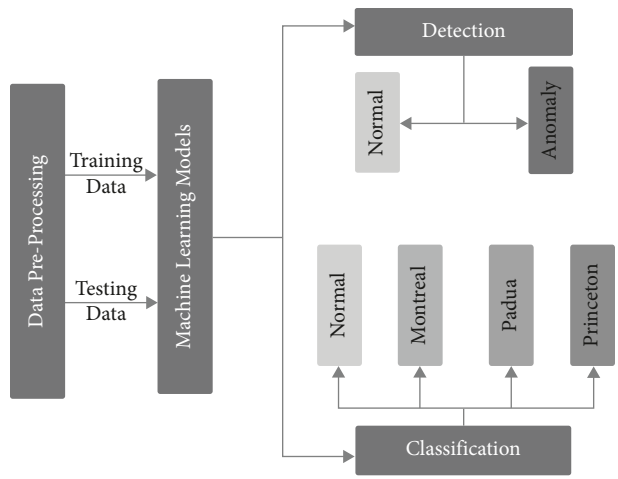

Figure 1: ML-based bitcoin transaction predictive model.

We took the following steps to make our prediction:

(i) Data preprocessing
(ii) Modeling
(iii) Model performance comparison
(iv) Choice of the best-performing model

3.3. Data Preprocessing. For any learning problem, the first step is to analyze the structure of the database and its different characteristics.

Then, we need to perform the necessary transformations to get the dataset ready to apply the prediction algorithms. To import our database, we used the Pandas library.

The goal of our study is to predict whether a new transaction is malicious or not. Thus, let us turn our problem into a binary classification problem. Table 2 presents the legitimate and ransomware Bitcoin transactions.

In this study, we will not take into account *year*, *day*, and *address* attributes because they do not affect the nature of the transactions [18].

The only attributes that will affect our prediction are *length*, *weight*, *count*, *looped*, *neighbors*, and *income*.

3.3.1. Data Sampling. The dataset used is unbalanced since we have 41,413 ransomware and 2 875,284 anomalies. To solve this problem, we used the sampling methods: undersampling and oversampling [20].

Typically, the use of sampling methods in imbalanced learning applications consists of the modification of an imbalanced dataset by some mechanism to provide a balanced distribution. Several research studies have shown that, for several classifiers, a balanced dataset provides enhanced classification performance compared to an imbalanced dataset.

Random undersampling (RUS) removes data from the original dataset. We randomly choose a set of majority class examples and remove the chosen samples from the dataset. Consequently, undersampling readily gives us a simple method for adjusting the balance of the original dataset [21, 22].

Figure 2 shows random undersampling and oversampling. The RUS consists of adding a sample set from the minority class: for a set of randomly selected minority samples, increase the original set by replicating the selected samples and adding them to the dataset. This way, the number of total samples in the minority class is increased and the class distribution balance is adjusted accordingly. This provides a mechanism for varying the class distribution balance degree to any desired level.

We used a method based on studies carried out in [18] to solve the problem of imbalanced data.

We considered a percentage of the total dataset since it contains ≃3 million instances, which will greatly affect the performance as well as prediction time.

We randomly took 200,000 instances (samples) from the total dataset, which will contain all 41,313 malicious transactions and 158,587 legitimate transactions. We, thus, obtained a subset containing a representative part of the original dataset.

Undersampling is the process of randomly selecting examples from the majority (legitimate) class to be removed from the dataset.

By using this method, we have reduced the number of legitimate transactions so that the dataset is more balanced. This approach may be appropriate since there are a sufficient number of examples in the minority class (41,413); therefore, it is possible to construct a useful model.

We finally got a dataset that contains 82,826 instances, so perfectly balanced.

3.3.2. Label Encoding. ML models require all input and output variables to be numeric. This means that if any of the data contain categorical data, it must be cipher-coded before the model can be fitted and evaluated. We will therefore use the *label encoder* method for the *MLabel* attribute which contains categorical data [23].

These encoding methods can be used with other data (nonbinary problems). There are other encoding techniques such as *One hot Encoding* and *Dummy Encoding* [24].

3.3.3. Anomaly Detection. The purpose of anomaly detection is to spot data that do not conform to what one would expect from other data.

These are, for example, data that do not follow the same pattern or that are atypical for the observed probability distribution.

The difficulty of the problem comes from the fact that we do not know beforehand the underlying distribution of the dataset. It is up to the algorithm to learn an appropriate metric to detect anomalies.

In our work, we used the Z score method to identify outliers in our dataset. This detection technique is one of the most used in the preprocessing phase of ML projects to improve model performance [23].

In the beginning, we proceeded to detect the outliers of each attribute. 178 attributes were detected for the *income* attribute. We repeated this procedure for all attributes to eliminate outliers.

We found that after this processing on the different columns of the dataset, we are left with 187,079 instances.

3.3.4. Data Standardization. Regarding Figure 3, we notice that there is a big difference between the column means (scaling issue). To solve this problem, we used the *scaling* method [23]. This will affect the models so that they converge quickly and the estimators become more efficient. Thus, the results obtained will be more significant.

3.3.5. Correlations. It is important to discover and quantify the degree of dependence of the variables in our dataset on each other. This knowledge will help us better prepare the data to meet the expectations of machine learning algorithms, such as linear regression, whose performance

Table 2: Legitimate and ransomware bitcoin transactions.

Class	#Instances
Normal (legitimate)	2875284
Anomaly (ransomware)	41413

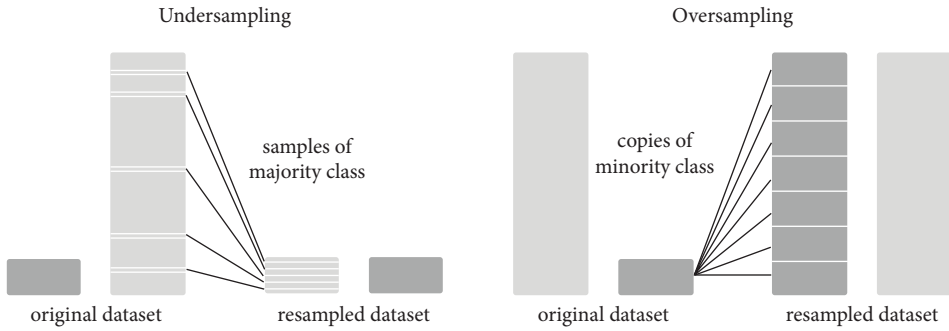

Figure 2: Random undersampling and oversampling.

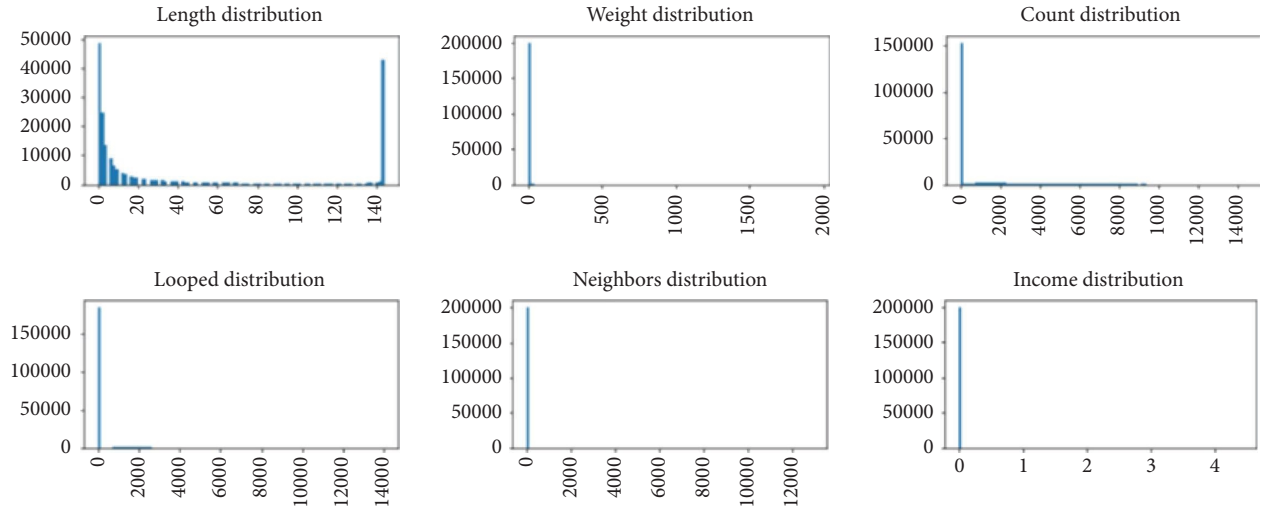

Figure 3: Attribute distribution.

degrades in the presence of these intercorrelations. Other classification models are not affected by collinearity.

From the correlation matrix shown in Figure 4, it is obvious that apart from the correlation between length and number (0.7), the other attributes have a very weak correlation between them. This result is very good for the application of the logistic regression (LR) model. It is also obvious that apart from the correlation between *length* and *count* (0.7), all other attributes have a very low correlation between them. This result is very good for the application of the logistic regression model. Other classification models are not affected by collinearity.

3.4. Classification Algorithms. LR is a method of statistical analysis that involves predicting a data value based on actual observations in a dataset. LR has become an important tool in the discipline of ML [25].

The general idea behind the RF method is as follows: instead of trying to obtain an optimized method at once, we generate several predictors before pooling their different predictions [26].

XGBoost is an optimized distributed gradient boosting method. It is developed with both deep consideration in terms of system optimization and principles of ML [27].

In spite of the fact that Gradient Boost methods are sequential algorithms, XGBoost uses multithread processing to search in parallel for the best split between the features. The use of multithreading helps XGBoost turn in a good performance when compared to other Gradient Boost methodimplementations [28].

3.5. Evaluation Metrics. In ML and statistics, there are a variety of methods for evaluating the performance of a classifier. The most common metrics are as follows:

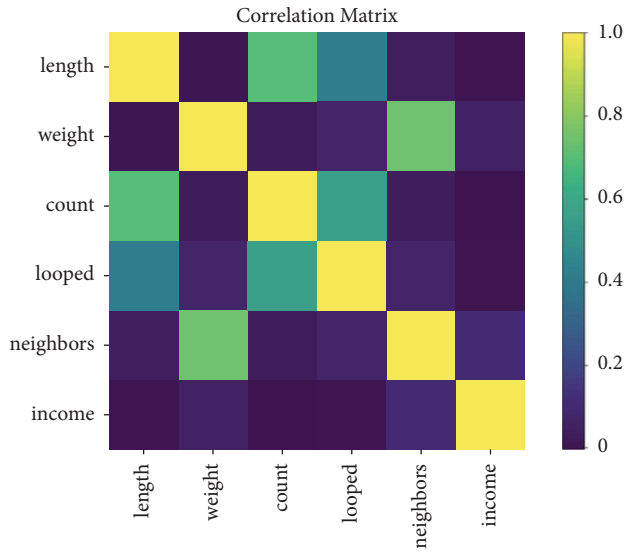

Figure 4: Correlation matrix.

(i) Confusion matrix is the most widely used metric (see Table 3) [29, 30].

Considering the confusion matrix in Table 1, we want the values of TN and TP as high as possible, and the values of FP and FN are as low as possible.

(ii) Precision is used to demonstrate the trade-off in a model between the sensitivity of detecting TP while balancing the number of FP [31, 32]. It is given by equation (1) as follows:

$$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}. \quad (1)$$

(iii) Furthermore, we can define the true positive rate (TPR), called recall, by equation (2) [31]. It is used to demonstrate the ability of a model to detect positive cases in a dataset as follows:

$$\text{Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}. \quad (2)$$

(iv) Accuracy is the fraction of predictions our model got right [33]. It is given by equation (3) as follows:

$$\text{Accuracy} = \frac{\sum \text{TN} + \text{TP}}{\sum \text{TN} + \text{TP} + \text{FN} + \text{FP}}. \quad (3)$$

(v) $F1$-score (or F-score) combines precision and recall [33]. It is given by equation (4) as follows:

$$F1 - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4)$$

4. Experiments and Results

4.1. Experimental Setup. After preparing the dataset, we proceeded to the modeling. This phase will consist of 3 steps as follows:

(i) Creation of training and testing sets

(ii) Construction of individual models

(iii) Model performance

In the first phase, we built two subsets: one contains a percentage of 33% of the data and the other 66%. Training data are ready to train any machine learning model.

In the second phase, we implemented the 3 classification algorithms used in our prediction: LR, RF, and XGBoost.

RF is longer than LR because of the complexity of its steps. RF gave us the opportunity to determine the importance of each attribute during the algorithm.

XGBoost provides better solutions than other machine learning algorithms in multiple types of applications.

All ML models mentioned previously have hyperparameters that must be defined to adapt the model to our dataset.

Hyperparameters are configuration points that allow model customization for a specific task or a set of data.

There is a difference between parameters and hyperparameters: parameters are learned automatically during fitting, while hyperparameters are set manually to help guide the learning process.

Thus, it is often necessary to search for a set of hyperparameters that achieves the best performance of a model on a dataset.

The Scikit-learn library [23] provides these techniques for tuning model hyperparameters. In this work, we used the *Random Search* method.

4.2. Result Analysis. After implementing the classification algorithms, we compared the performance of different prediction models to choose the most appropriate model.

Initially, data standardization and normalization have been performed to rescale the data values. Afterward, the class imbalance problem is resolved. Lastly, three classification algorithms, i.e., LR, RF, and XGBoost, are used to classify between abnormal transactions and normal ones (legitimate).

4.2.1. Performance Analysis of Classification Models without Handling Imbalanced Data. Table 4 presents the classification model results without handling imbalanced data, in which LR achieved the highest accuracy, precision, recall, and $F1$-score values approximately 90.39%, 77.12%, 73.7%, and 74.77%, respectively.

RF gave the lowest accuracy, precision, recall, and $F1$-score values of approximately 83.17%, 51.6%, 43.72%, and 47.33%, respectively.

XGBoost achieved accuracy, precision, recall, and $F1$-score values of approximately 89.09%, 71.6%, 73.88%, and 72.72, respectively.

4.2.2. Performance Analysis of Classification Models by Handling Imbalanced Data. The true positives (TPs) indicate the Bitcoin transactions where the predictions and the actual values are indeed positive. According to Figure 5, the

Table 3: Confusion matrix.

		Predicted class	
		Normal	Anomaly
Actual class	Normal	TN (true negatives)	FP (false positives)
	Anomaly	FN (false negatives)	TP (true positives)

Table 4: Classification models without handling imbalanced data.

Metric	LR	RF	XGBoost
Accuracy (%)	90.39	83.17	89.09
Precision (%)	77.12	51.6	71.6
Recall (%)	73.7	43.72	73.88
*F*1-score (%)	74.77	47.33	72.72

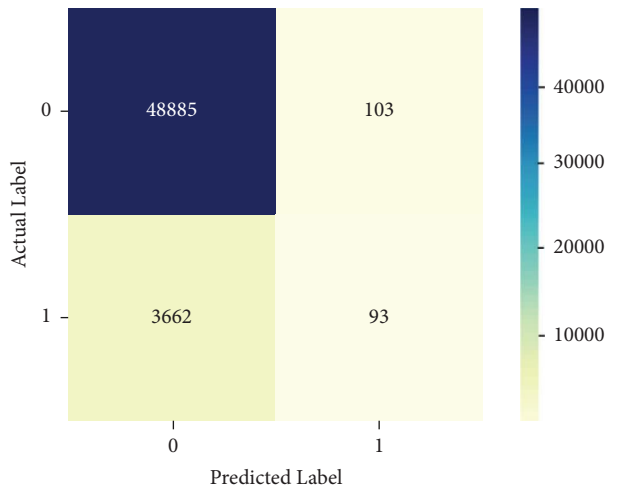

Figure 5: LR confusion matrix.

LR model recognized 48,885 Bitcoin transactions. True negatives (TNs = 93 Bitcoin transactions) given by LR indicate situations where both predictions and actual transactions are negative. False positives (FPs = 103 Bitcoin transactions) indicate a positive prediction contrary to the real value which is negative. They are also considered type-1 errors. In false negatives (FNs = 3662 Bitcoin transactions), the predicted transaction is negative, but the actual one is positive. They are also considered type-2 errors. In the case of Bitcoin transactions, this means relevant transactions that have been classified as ransomware.

The error rate is a metric that is calculated by summing all incorrect predictions over the total number of data (positive and negative). The lower it is, the better. The best possible error rate is 0, but it is rarely achieved by a model in practice. The error rate for the LR model is approximately 0.072, which represents the difference between the number of authentic transactions predicted as ransomware and the number of genuine transactions recognized as ransomware.

Figure 6 shows the LR confusion matrix. The number of positives and negatives are as follows:

(i) Positives: 55165 (TP = 46384 and TN = 8781)

(ii) Negatives: 3988 (FN = 1384 and FP = 2604)

The error rate for the RF model is approximately 0.06.

Figure 7 shows the XGBoot confusion matrix. The number of positives and negatives are as follows:

(i) Positives: 50 857 (TP = 48372 and TN = 2485)

(ii) Negatives: 472 (FN = 407 and FP = 65)

The error rate for the XGBoost model is approximately 0.01.

The error rate given by XGBoost is better than LR and RF.

Table 5 summarizes the results of classification models by handling imbalanced data, in which XGBoost achieved the highest accuracy, precision, and *F*1-score values approximately 99.08%, 99.86%, and 99.5%, respectively, whereas LR obtained the lowest accuracy and precision values of approximately 92.86% and 93.03%. RF achieved an accuracy of 94.45% with 97.1% precision, 94.68% recall, and 95.75% *F*1-score.

According to Table 5 and the accuracy metric, LR gives the smallest value (92.86%) compared to RF (94.45%) and XGBoost (99.08%).

Since accuracy is a metric used to measure the ability of the model to correctly predict positive classes (ransomware), XGBoost is considered the best model to correctly predict ransomware.

As we mentioned before, to get an appropriate trade-off and balance between precision and recall, we consider the *F*1-score. This metric does not really have any meaning or interpretation. It is simply a combination of recall and precision to find a compromise between them. It helps us to better decide which model to choose. We notice that XGBoost gives the best *F*1-score (99.5%) compared to LR (96.24%) and RF (95.75%).

According to the recall values, we highlight a little difference between LR (99.7%) and XGBoost (99.16%).

After analysis of the three classification models, it is observed that the XGBoost scheme provided the highest accuracy.

We have centered our comparison on the model's classification accuracies that are reported in the literature

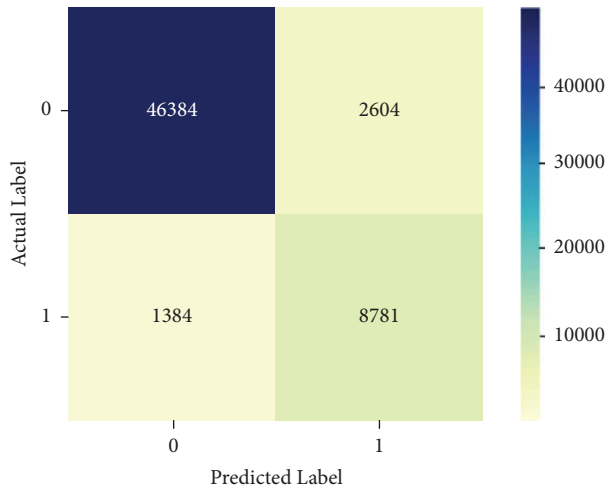

Figure 6: RF confusion matrix.

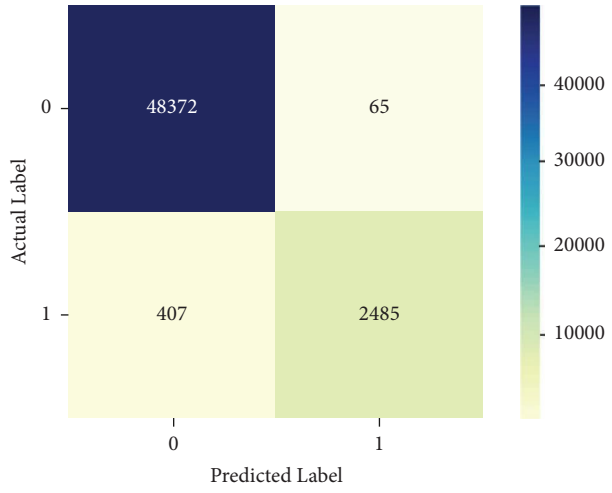

Figure 7: XGBoost confusion matrix.

Table 5: Classification models by handling imbalanced data.

	LR	RF	XGBoost
Accuracy (%)	92.86	94.45	99.08
Precision (%)	93.03	97.1	99.86
Recall (%)	99.7	94.68	99.16
*F*1-score (%)	96.24	95.75	99.5

work because accuracy is a vigorous performance evaluator to demonstrate the robustness of ML-based models.

Table 6 presents a comparison of accuracy values with existing ML-based predictive models.

According to the results presented in Table 6, the proposed model using XGBoost is competent and superior in providing identification for ransomware for Bitcoin transactions.

Table 6: Accuracy comparison.

Research	ML model	Accuracy (%)
Alhawi et al. [16]	Decision tree J48 classifier	97.1
Yazdinejad et al. [34]	Long short-term memory (LSTM)	98.0
Kolesnikova et al. [35]	Convolutional neural net (CNN)	97.1
Our model	Extreme gradient boosting (XGBoot)	**99.08**

5. Conclusions and Future Work

Nowadays, it may be difficult to detect zero-day ransomware attacks with statistical methods because they become dependent on data and are not sensitive to error costs. In this article, we developed, investigated, and evaluated a self-reliant ransomware prediction system for bitcoin transactions. The proposed system employs three supervised machine learning methods to recognize data patterns in bitcoin payment transactions, namely, LR, RF, and XGBoost.

Several performance evaluation metrics, such as classification accuracy, precision, and recall, have been used to evaluate the proposed predictive models using recent, up-to-date, and comprehensive Bitcoin transaction dataset. Consequently, the validation testing of model experimentation recorded 99.08% for the Bitcoin transaction detection accuracy (two-class classifier) as using XGBoost. In comparison with several existing bitcoin transaction prediction models, we have achieved the best accuracy results.

Based on empirical evidence, we show that Bitcoin transactions related to ransomware can be detected more accurately.

In future work, we will consider planning threat intelligence information to increase the accuracy of our prediction model and analyze the impact of hyperparameter tuning on the result of the predictive model using several MLAs.

Data Availability

The data used to support the findings of this study are available at https://archive.ics.uci.edu/ml/datasets/BitcoinHeistRansomwareAddressDataset.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] S. Ruoti, B. Kaiser, A. Yerukhimovich, J. Clark, and R. Cunningham, "Blockchain technology: what is it good for?" *Communications of the ACM*, vol. 63, no. 1, pp. 46–53, 2019.

[2] Z. Zhang, X. Song, L. Liu, J. Yin, Y. Wang, and D. Lan, "Recent advances in blockchain and artificial intelligence integration: feasibility analysis, research issues, applications, challenges, and future work," *Security and Communication Networks*, vol. 2021, Article ID 9991535, 2115 pages, 2021.

[3] S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Prevention of crypto-ransomware using a pre-encryption detection algorithm," *Computers*, vol. 8, no. 4, p. 79, 2019.

[4] S. Xu, *The Application of Machine Learning in Bitcoin Ransomware Family Prediction*, Association for Computing Machinery, New York, NY, USA, 2021.

[5] J. A. Abraham and S. M. George, "A survey on preventing crypto ransomware using machine learning," in *Proceedings of the 2nd International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, vol. 1, pp. 259–263, Kannur, India, July 2019.

[6] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: a technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 03 2016.

[7] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *Proceedings of the Financial Cryptography and Data Security*, pp. 34–51, Okinawa, Japan, April 2013.

[8] M. Ober, S. Katzenb/eisser, and K. Hamacher, "Structure and anonymity of the bitcoin transaction graph," *Future Internet*, vol. 5, no. 2, pp. 237–250, 2013.

[9] G. Di Battista, V. Di Donato, M. Patrignani, M. Pizzonia, V. Roselli, and R. Tamassia, "Bitconeview: visualization of flows in the bitcoin transaction graph," in *Proccedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)*, pp. 1–8, Chicago, IL, USA, October 2015.

[10] M. Moser and R. Böhme, "The price of anonymity: empirical evidence from a market for bitcoin anonymization," *Journal of Cybersecurity*, vol. 3, no. 2, pp. 127–135, 2017.

[11] S. Meiklejohn, M. Pomarole, G. Jordan et al., "A fistful of bitcoins: characterizing payments among men with no names," *Communications of the ACM*, vol. 59, no. 4, pp. 86–93, 2016.

[12] M. Paquet-Clouston, B. Haslhofer, and B. Dupont, "Ransomware payments in the bitcoin ecosystem," *Journal of Cybersecurity*, vol. 5, no. 1, 05 2019.

[13] D. Y. Huang, M. M. Aliapoulios, V. Guo et al., "Tracking ransomware end-to-end," in *Proccedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 618–631, San Francisco, CA, USA, May 2018.

[14] K. Liao, Z. Zhao, D. Adam, and G.-J. Ahn, "Behind closed doors: measurement and analysis of cryptolocker ransoms in bitcoin," in *Proceedings of the APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–13, Toronto, Canada, June 2016.

[15] M. Al Harrack, "The bitcoinheist: classifications of ransomware crime families," *International Journal of Computer Science and Information Technology*, vol. 13, no. 5, pp. 75–81, 2021.

[16] O. Alhawi, J. Baldwin, and D. Ali, *Leveraging machine learning techniques for windows ransomware network traffic detection*, pp. 93–106, Springer Nature, Berlin, Germany, 2018.

[17] G. Akcora Cuneyt, Y. Li, R. Gel Yulia, and M. Kantarcioglu, "Bitcoinheist: topological data analysis for ransomware detection on the bitcoin blockchain," 2019, https://arxiv.org/abs/1906.07852.

[18] D. Goldsmith, K. Grauer, and Y. Shmalo, "Analyzing hack subnetworks in the bitcoin transaction graph," *Applied Network Science*, vol. 5, no. 1, p. 22, 2020.

[19] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, p. 281, 2019.

[20] S. Deep, P. S. Jayadev, and N. Bhatt, "Aided selection of sampling methods for imbalanced data classification," in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (CODS-COMAD)*, pp. 198–202, New York, NY, USA, January 2021.

[21] F. Alharbi, L. Ouarbya, and J. A. Ward, "Comparing sampling strategies for tackling imbalanced data in human activity recognition," *Sensors*, vol. 22, no. 4, p. 1373, 2022.

[22] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[23] P. Fabian, V. Gael, and G. Alexandre, "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] A. H. Chowdhury, N. Mumenin, M. Taus, and M. A. Yousuf, "Detection of compatibility, proximity and expectancy of Bengali sentences using long short term memory," in *Proceedings of the 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pp. 233–237, Dhaka, Bangladesh, January 2021.

[25] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of Biomedical Informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[27] S. A. Alsaif, A. Hidri, and M. Sassi Hidri, "Stacking-based modelling for improved over-indebtedness predictions," *International Journal of Computer Applications in Technology*, vol. 69, no. 3, pp. 273–281, 2022.

[28] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 785–794, New York, NY, USA, August 2016.

[29] H. A. Ahmed, A. Hameed, and N. Z. Bawany, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Computer Science*, vol. 8, p. e820, 2022.

[30] S. A. Alsaif, M. Sassi Hidri, H. A. Eleraky, I. Ferjani, and R. Amami, "Learning-based matched representation system for job recommendation," *Computers*, vol. 11, no. 11, p. 161, 2022.

[31] S. A. Ludwig, "Intrusion detection of multiple attack classes using a deep neural net ensemble," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, Honolulu, HI, USA, December 2017.

[32] S. A. Alsaif, M. Sassi Hidri, I. Ferjani, H. A. Eleraky, and A. Hidri, "Nlp-based bi-directional recommendation system: towards recommending jobs to job seekers and resumes to recruiters," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 147, 2022, https://www.mdpi.com/2504-2289/6/4/147.

[33] P. Lin, K. Ye, and C.-Z. Xu, "Dynamic network anomaly detection system by using deep learning techniques," in *Proceedings of the 2th International Conference Cloud Computing*, pp. 161–176, Taichung, Taiwan, October 2019.

[34] A. Yazdinejad, H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, G. Srivastava, and M. Y. Chen, "Cryptocurrency malware hunting: a deep recurrent neural network approach," *Applied Soft Computing*, vol. 96, Article ID 106630, 2020.

[35] K. Kolesnikova, O. Mezentseva, and T. Mukatayev, "Analysis of bitcoin transactions to detect illegal transactions using convolutional neural networks," in *Proceedings of the IEEE International Conference on Smart Information Systems and Technologies*, pp. 1–6, Nur-Sultan, Kazakhstan, April 2021.