Fraudulent Behaviour Identification in Ethereum Blockchain

Karolis Lašas[a], Gabrielė Kasputytė[a], Rūta Užupytė[a] and Tomas Krilavičius[b]

[a]Baltic Institute of Advanced Technology, Department of Mathematics and Statistics, Vytautas Magnus, Kaunas, Lithuania
[b]Baltic Institute of Advanced Technology, Department of Applied Informatics, Vytautas Magnus, Kaunas, Lithuania

Abstract

The phenomenon of cryptocurrencies continues to draw a lot of attention from investors, innovators and the general public. There are over 1300 different cryptocurrencies, including Bitcoin, Ethereum and Litecoin. While the scope of blockchain technology and cryptocurrencies continues to increase, identification of unethical and fraudulent behaviour still remains an open issue. The absence of regulation of the cryptocurrencies ecosystem and the lack of transparency of the transactions may lead to an increased number of fraudulent cases. In this research, we have analyzed the possibility to identify fraudulent behaviour using different classification techniques. Based on Etherium transactional data, we constructed a transaction network which was analyzed using a graph traversal algorithm. Data clustering was performed using three machine learning algorithms: k-means clustering, Support Vector Machine and random forest classifier. The performance of the classifiers was evaluated using a few accuracy metrics that can be calculated from confusion matrix. Research results revealed that the best performance was achieved using a random forest classification model

Keywords

Cryptocurrency, Ethereum, Blockchain, Fraudulent Activity, K-Means Clustering, Support Vector Machine, Random Forest Classifier

1. Introduction

Cryptocurrencies are a viable alternative to traditional mediums of exchange for purchasing goods or services. The main idea behind such type of currency is that the exchange between two parties can occur without the involvement of a central authority. It is the network itself that manages and confirms each transaction. The overall history of transactions is controlled using the blockchain technology, which can be described as a growing list of records, that are linked together using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp and transaction data. Even though blockchain technology records information about each transaction, it also assures person anonymity, as long as there is no link between the wallet and its owner identity. Due to this reason, cryptocurrencies are more frequently used for fraudulent activities[1]. As collected by blockchain forensics company CipherTrace [2], the increasing amount of scams led to 4.5 billion dollars in losses in 2019. According to the blockchain monitoring company [3] Ethereum blockchain, which is the second largest cryptocurrency by market capitalization, is the top choice for fraudulent activity. The aim of this paper is to analyze the possibility to use machine learning techniques to identify wallets engaging in fraudulent activities in Ethereum blockchain.

The rest of the paper is organized as follows. Related work in this area is presented in section 2. Section 3 introduces the dataset used in the current study and the performed preprocessing steps. Section 4 presents the selected clustering techniques. Section 4.4 describes accuracy metrics that was used to evaluate computational results. Experimental results are provided in section 5. Finally, concluding remarks and future plans are discussed in Section 6.

2. Related Work

Fraudulent activity identification in cryptocurrency is discussed in [4]. The article aims to develop a Supervised Machine Learning based novel approach to de–anonymize the Bitcoin ecosystem and identify criminal activities in Bitcoin blockchain. The substantial number of Bitcoin addresses were already identified, clustered and categorized by the data provider. However, main part of clusters were uncategorized. In overall, the dataset contains around 395 million transactions related to 957 unique clusters.

The 957 observations which were labeled by the data provider were used for training and test sets. This dataset includes categories commonly associated with

IVUS 2020: Information Society and University Studies, 23 April 2020, KTU Santaka Valley, Kaunas, Lithuania
karolis.lasas@bpti.lt (K. Lašas); gabriele.kasputyte@bpti.lt (G. Kasputytė); ruta.uzupyte@bpti.lt (R. Užupytė); tomas.krilavicius@bpti.lt (T. Krilavičius)

illegal activities, including darknet market, mixing, ransomware, scam, stolen bitcoins, and gambling from the perspective of certain jurisdictions. The research method consisted of three iterations using three separate datasets: the initial dataset, the dataset with over-sampled minority classes, and the final, where all classes were over-sampled to achieve the same number of the most populated class observations.

Upon comparing the results of the three iterations the over-sampled datasets of the models were discarded. Moreover, the performance across seven algorithms: Decision Trees, Bagging, Random Forests, Extra Trees, AdaBoost, Gradient Boosting and k-Nearest Neighbors, was compared and the best four: Gradient Boosting, Random Forests, Extra Trees and Bagging Classifier, were chosen. Finally, Gradient Boosting was selected as the most accurate algorithm with an average cross–validation accuracy of 80.83%. Anomalies detection in Bitcoin network was analised in [5], where three unsupervised learning methods: k-means clustering, Mahalanobis distance method and Unsupervised Support Vector Machines, were applied.

In this research Bitcoin transaction network were transformed into two graphs: with nodes as users and with nodes as transactions. The dataset consists of more than 6 million unlabeled users with more than 37 million transactions and 30 revealed thieves in Bitcoin network. However, due to the long run-time, the dataset were limited to 100,000. Both Unsupervised SVM method and Mahalanobis distance based method suggested similar suspicious users. In this case two cases of theft and one case of loss out of the 30 known cases were detected.

The use of machine learning techniques for the identification of abnormal activities in the Ethereum network is discussed in [6]. In this case, decision tree classifier, k-nearest neighbors, Random forest, Support-vector Machine (SVM), Multi-layer perceptron (MLP) and Naive Bayes algorithms were compared. Using dataset consisting of 169,192,702 Ethereum transactions two evaluation models were analysed:

1. testing on 50 originally marked malicious addresses;
2. testing on randomly 50 malicious addresses out of possible 3830, under the assumption that the addresses are marked as malicious, if they have an outgoing transaction with the malicious marked addresses

Malicious addresses are considered to be the ones which perform unauthorized or illegal actions, such as: issues fake tokens, fake admin in ICOs (Initial coin Offering), scambot phishers, slackbot, fake etherscan site, fake site – asking for private keys or fake crowdsale site. 125 addresses were identified as malicious and later were split into 75 for training and 50 for testing as ground truth. After taking the previously mentioned assumption 3830 addresses were marked as malicious. The best results was achieved using second evaluation model were SVM, Decision Tree classifier and Random Forest classifier produced the result with the same accuracy of 99.66%. Moreover, 5-fold cross-validation was used to prevent the models from over-fitting.

A comprehensive identification model for detection of phishing scams in Ethereum is discussed in [7]. In this work, a large-scale Ethereum transaction network was built. Additionally, a novel network-embedding algorithm called *trans2vec* with biases of transaction amount and timestamp was designed to extract features from the Ethereum transaction network. Moreover, on account of data imbalance and network heterogeneity, the one–class SVM was adopted to classify the phishing and non–phishing addresses. Finally, the article concluded that after applying real information of Ethereum transactions, the results showed that proposed detection framework is effective and *trans2vec* is more superior than baseline methods in terms of feature extraction.

To sum up, some of these articles claim to have a high accuracy of fraudulent behavior identification results, while there are few low accuracy results in other articles. One of the article has detected that a new algorithm gives better results than the basic methods. The different types of data, its size and information have caused the differences between the results, while applying the same models. In order to analyze the accuracy while using our own data, we decided to use 3 very popular and the most common methods: K-Means clustering, Support Vector Machine and Random Forest classifier.

3. Data preprocessing and features‘ extraction

3.1. Initial data

A data set consists of two collections of Ethereum transactions. The first collection is composed of about 420 fraudulent wallets identified from *etherscamdb.info* database. A detailed information about their transactions was gathered from *etherscan.io.* The second data collection represents non-fraudulent activities and consists of 53 wallets and their transactional information gathered from *etherscan.io* database. Each data set includes:

- transaction hash code
- sender's address
- receiver's address
- transaction value
- time at which transaction was made
- Ethereum block number.

3.2. Features extraction

Transactional data was transformed into a graph, where the nodes represent wallets and edges indicate money transfers. Using a graph traversal algorithm, we identify parameters representing each wallets behaviour:

- total value in ETH sent by a wallet;
- total received value in ETH by a wallet;
- a number of transactions sent by a wallet;
- a number of transactions received by a wallet over a time period;
- average time between transactions performed by sending wallet;
- average time between transactions to a receiving wallet;
- standard deviation of time between transactions performed by a sending wallet;
- standard deviation of transaction time in seconds to receiving wallet - standard deviation of time between transactions to a receiving wallet;
- average value in ETH sent by a wallet;
- average value in ETH received by a wallet.

4. Methodology

4.1. K-Means Clustering

The first method that we considered was the k-means clustering algorithm as its computational times are considerably small comparing to other similar clustering techniques. Also k-means clustering may help to determine underlying patterns of fraudulent and non-fraudulent behaviour by grouping similar wallets' activities. K-means clustering algorithm works by allocating data points from given input vectors to a predefined number of clusters using similarity criteria, usually Euclidean distance:

$$\|x - \mu_k\|^2,$$

where x is a data point and μ_k is a k-th cluster's centroid. Each centroid is calculated by averaging given input vectors:

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i.$$

The objective of a k-means algorithm is to minimize total intra-cluster variance.
Among the many disadvantages of the k-means clustering algorithm, such as vulnerability to outliers or inability to cluster heavily overlapping data, there is a manual selection of clusters. Algorithm's inability to automatically select an optimal number of clusters in some cases makes it the unreliable solution to data partitioning as defining a number of clusters for unlabeled data leaves the user with uncertainty especially when working with large amounts of data. However, there is no need for guessing the number of clusters as there are a few methods that search for an optimal number of clusters. One of them is the elbow method. It is one of the oldest methods for defining an optimal number of clusters and works by calculating the sum of squared distances between every data point and its closest centroid [8]:

$$\sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2.$$

The optimal number of clusters can be identified by visible "elbow" on the curve (see fig. 1). The last number before curve flattens is an optimal count of clusters. The main drawback of this method occurs when there is no visible "elbow" on the curve or more than one "elbow" is visible.

4.2. Support Vector Machine

In order to find an optimal boundary between wallets with fraudulent and non-fraudulent behaviour, Support Vector Machine (SVM) is used. It offers high accuracy and requires less computational power than other machine learning algorithms. SVM aims to find a hyperplane in a n-dimensional space (where n is a number of factors used as input for the model) and separates given data points into new classes. SVM can be used both for regression and classification problems [9, 10]. Consider data set consisting of m pairs of records $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ as a training set, where $x_i \in \mathrm{R}^n$ and $y_i \in \{-1, 1\}$ [11]. In order to classify these pairs, we define a hyperplane that will separate them:

$$\{x : f(x) = x^T \beta + \beta_0 = 0\},$$

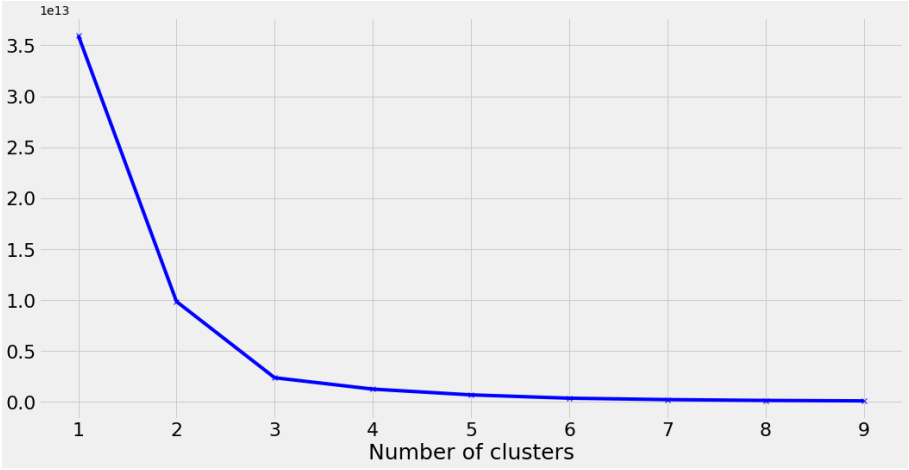

Figure 1: Elbow method to identify optimal number of clusters

where β is a unit vector ($\|\beta\| = 1$). Using defined hyperplane $f(x)$, a rule for data classification can be written as follows:

$$y(x) = sign[x^T\beta + \beta_0].$$

For a nonlinear SVM classification, kernel method is being used. Kernel method generates algorithms that maps given input data into a high-dimensional feature space. Popular kernel functions used in this method are:

- Polynomial:

  $$k(x_i, x_j) = (x_i \cdot x_j + 1)^d,$$

  where d is a degree of polynomial;

- Gaussian radial basis function (RBF):

  $$k(x_i, x_j) = \exp\{-\gamma\|x_i - x_j\|^2\}$$

  where $\gamma > 0$;

- Sigmoid:

  $$k(x, y) = tanh(\alpha x^T y + c)$$

4.3. Random Forest Classifier

Random Forest is a supervised machine learning algorithm that can be used to solve classification or regression problems and is more flexible with input data than SVM, especially working with large amounts of data. It is a decision tree–based algorithm that randomly selects various data samples and by calculating predictions for every tree makes decisions from which it partitions input data into new subsets. It uses averaging to improve the classification accuracy and controls the model to avoid over–fitting. For a n-dimensional input vector $X = (X_1, X_2, \dots, X_n)$ the goal of a random forest classifier is to find a prediction function $f(X)$ for predicting a response variable Y. The predictive function minimizes the expected value of the loss by using a loss function $L(Y, f(X))$ that usually is *zero-one loss* [12]:

$$L(Y, f(X)) = \begin{cases} 0, & \text{if } Y = f(X) \\ 1, & \text{otherwise} \end{cases}$$

Table 1
Confusion matrix structure

		Predicted values	
Actual values		TP	FN
		FP	TN

4.4. Accuracy evaluation

To estimate the accuracy of the proposed models, we use a few commonly used metrics [13, 14, 15, 16] that can be calculated from confusion matrix also known as contingency table (see table 1) :

- True Positive Rate:

  $$TPR = \frac{TP}{TP + FN}$$

  TPR also is known as sensitivity or recall, shows the amount of successfully predicted class‘ values compared to all class‘ values in a data set.

- True Negative Rate (Selectivity):

  $$TNR = \frac{TN}{TN + FP}$$

Table 2
Accuracy metrics for k-means clustering performance evaluation

Measure	Precision	Recall	F1-measure
Fraudulent wallets	0.89	0.98	0.93
Non-fraudulent wallets class	0	0	0
Model			0.87

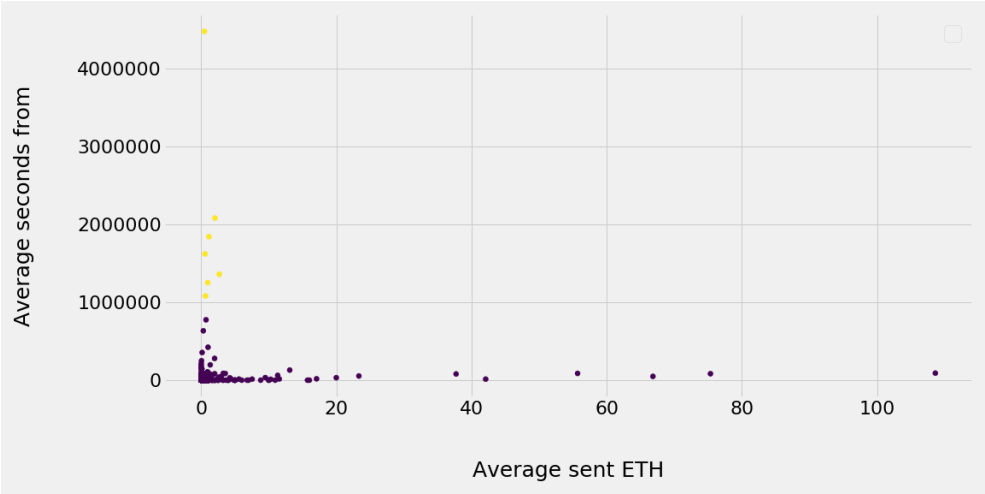

Figure 2: Calculated clusters

TNR also known as selectivity, is the amount of successfully predicted values for another class.

- Precision (Positive Predicted Value):

$$P = \frac{TP}{TP + FP}$$

- Negative Predicted Value:

$$NPV = \frac{TN}{TN + FN}$$

- F1-measure:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall},$$

F_1-measure is a harmonic mean of recall and precision [17] and refers to classification accuracy.

Here TP is true possitive (successfully predicted first class‘ values), TN is true negative (successfuly predicted second class‘ values), FP is false positive (faulty predicted second class‘ values also refered as type I error) and FN is false negative (faulty predicted first class‘ values also refered as type II error).

Table 3
Accuracy for different types of kernel

Kernel	Polynomial	Sigmoid	GRB	Linear
F1-measure	92	89	93	89

5. Results

5.1. K-Means Clustering

In this case, we decided to cluster the data into two groups referring to fraudulent and non-fraudulent wallets. We also performed an Elbow method to identify the optimal number of clusters (fig. 1), which confirmed that two clusters are an optimal choice. Using the actual data labels, we evaluated the accuracy of the k-means algorithm. Results revealed that overall clustering accuracy reaches 87% (see table 2). However, while fraudulent wallets were clustered with 93% accuracy, all non-fraudulent wallets were labeled as frauds (table 2). A more detailed study of clustering results was carried out using graphical analysis. For example, figure 2 represents the relationship between the average value in ETH sent by a wallet and the average time between outgoing transactions. Different colours

Table 4
Accuracy metrics for nonlinear SVM model with GRB kernel classification model

Measure	Precision	Recall	F1-measure
Fraudulent wallets	0.96	0.97	0.96
Non-fraudulent wallets class	0.50	0.43	0.46
Model			0.93

Table 5
Accuracy metrics for random forest classification model

Measure	Precision	Recall	F1-measure
Fraudulent wallets	0.98	0.97	0.97
Non-fraudulent wallets class	0.62	0.71	0.67
Model			0.95

represent separate clusters. By comparing clustering results with the labelled dataset (fig. 3), we can see that the algorithm identifies the most extreme cases (cases with the largest values). However, the model is unable to separate the rest of the data. Based on these results, we can conclude that k-means clustering provides unreliable results.

5.2. Support Vector Classifier

In order to achieve the best classification result, we have performed experiments using four support vector machine classification models:

- linear SVM;
- SVM with polynomial kernel;
- SVM with sigmoid kernel;
- SVM with Gaussian Radial Basis (GRB) kernel.

Labeled data set was split into training (80 percent of data) and testing (20 percent of data) sets. The highest accuracy (93%) was achieved by using nonlinear SVM model with Gaussian Radial Basis (GRB) kernel (table 4). However, although using nonlinear SVM with GRB kernel 96% of fraudulent wallets were classified correctly, 54% of non-fraudulent wallets were classified as frauds.

5.3. Random Forest Classifier

After performing classification with RFC with 90 trees, we extracted feature importances for model fine tuning (fig. 4). Parameters with importance level higher than 0.1 were selected as the most important:

- total sent value in ETH;
- the average value in ETH sent by a wallet;
- average time between outgoing transactions;
- standard deviation of time between outgoing transactions;
- frequency of outgoing transactions.

After defining the list of parameters that have the highest influence on classification results, random forest classification algorithm was performed. To evaluate model's accuracy we used accuracy metrics discussed in subsection 4.4. RFS model reaches 95% accuracy (see table 5). This method predicts fraudulent wallets with 97% accuracy and non-fraudulent wallets with 67%.

6. Conclusions

In this research, we investigated three machine learning techniques to identify fraudulent behaviour in the Ethereum blockchain data set. First of all, we suggested the data preprocessing framework for the extraction of individual behaviour patterns from a transactional dataset. Based on these patterns, the proposed models were trained and compared according to selected accuracy measures. Experimental results revealed that the random forest classification method is the most suitable for the identification of fraudulent behaviour. Furthermore, the model suggests that the most important factors for fraudulent behaviour identification are total value in ETH sent by a wallet, the average value in ETH sent by a wallet, the average time between outgoing transactions, the standard deviation of time between outgoing transactions and the frequency of outgoing transactions.

In the future, we are planning to improve the proposed model's reliability by increasing the number of

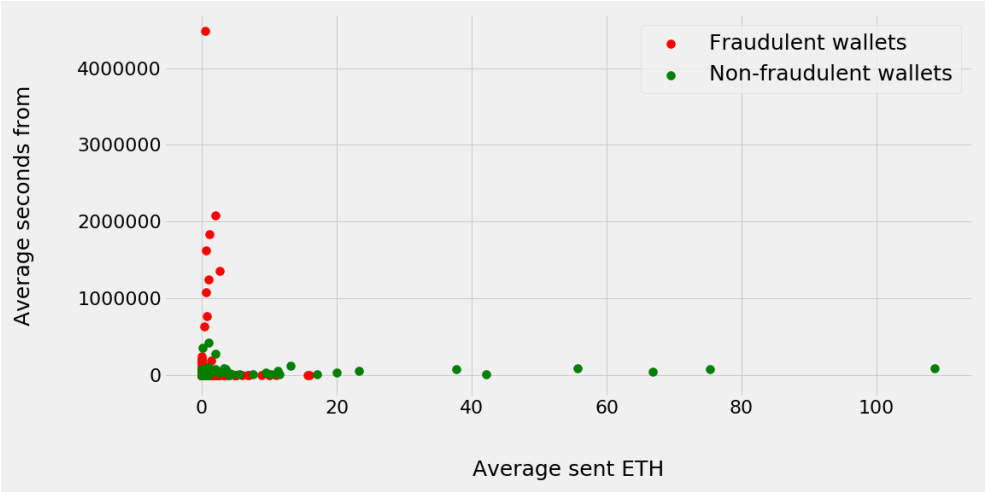

Figure 3: Labeled data set

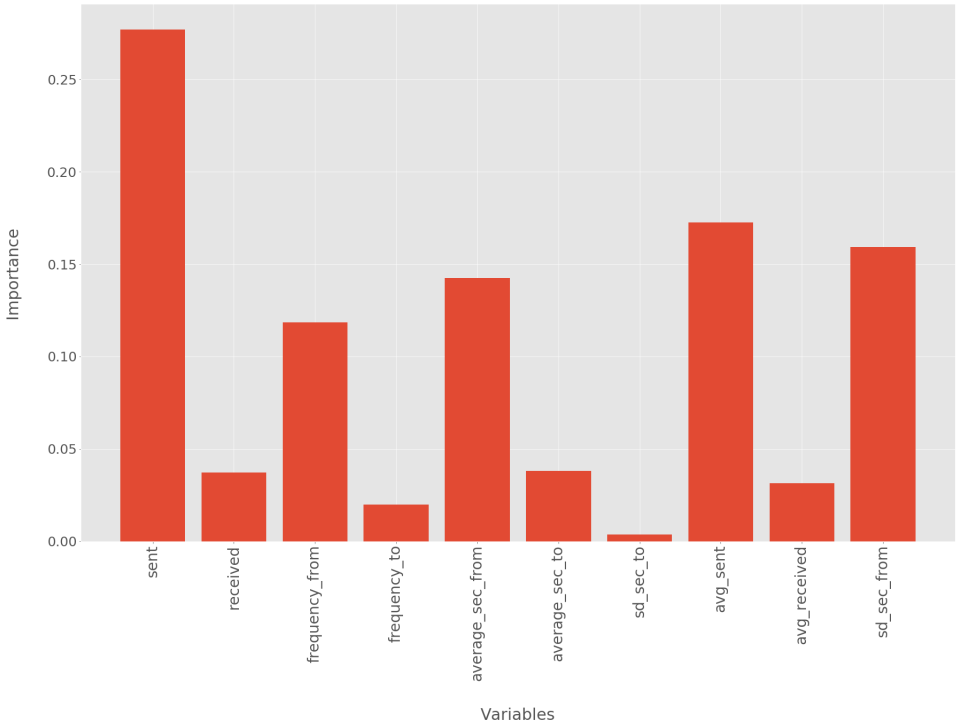

Figure 4: Random Forest Classifier feature importances.

both fraudulent and non-fraudulent wallets. Moreover, we are planning to analyse the possibility to use XGBoost method, as it was suggested to use for identification of abnormal activity in blockchain data [18]. Furthermore, we are planning to perform a statistical significance test in order to find out whether differences between results are statistically significant.

7. Acknowledgments

We thank Tadas Tamošiūnas, Pavel Sokolov and UAB Kevin EU [1] for cooperation and useful insights.

[1] https://getkevin.eu

References

[1] Baum, S. C., Cryptocurrency fraud: A look into the frontier of fraud, 2018.

[2] Ciphertrace, 2020. URL: https://ciphertrace.com.

[3] Chainalysis, 2020. URL: https://www.chainalysis.com/.

[4] H. H. Sun Yin, K. Langenheldt, M. Harlev, R. R. Mukkamala, R. Vatrapu, Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain, Journal of Management Information Systems 36 (2019) 37–73.

[5] T. Pham, S. Lee, Anomaly detection in bitcoin network using unsupervised learning methods, arXiv preprint arXiv:1611.03941 (2016).

[6] A. Sing, Anomaly Detection in the Etherum Network, Ph.D. thesis, Indian Institute of Technology Kanfur, 2019.

[7] J. Wu, Q. Yuan, D. Lin, W. You, W. Chen, C. Chen, Z. Zheng, Who are the phishers? phishing scam detection on ethereum via network embedding, arXiv preprint arXiv:1911.09259 (2019).

[8] T. M. Kodinariya, P. R. Makwana, Review on determining number of cluster in k-means clustering, International Journal 1 (2013) 90–95.

[9] M. Awad, R. Khanna, Support vector machines for classification, in: Efficient Learning Machines, Springer, 2015, pp. 39–66.

[10] F. Beritelli, G. Capizzi, G. Lo Sciuto, C. Napoli, F. Scaglione, Rainfall estimation based on the intensity of the received signal in a lte/4g mobile terminal by using a probabilistic neural network, IEEE Access 6 (2018) 30865–30873. doi:10.1109/ACCESS.2018.2839699.

[11] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer Science & Business Media, 2009.

[12] A. Cutler, D. R. Cutler, J. R. Stevens, Random forests, in: Ensemble machine learning, Springer, 2012, pp. 157–175.

[13] T. Fawcett, An introduction to roc analysis, Pattern recognition letters 27 (2006) 861–874.

[14] D. M. Powers, Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation (2011).

[15] G. Capizzi, G. Lo Sciuto, C. Napoli, D. Polap, M. Woźniak, Small lung nodules detection based on fuzzy-logic and probabilistic neural network with bio-inspired reinforcement learning, IEEE Transactions on Fuzzy Systems 6 (2020).

[16] F. Beritelli, G. Capizzi, G. Lo Sciuto, C. Napoli, M. Woźniak, A novel training method to preserve generalization of rbpnn classifiers applied to ecg signals diagnosis, Neural Networks 108 (2018) 331–338.

[17] Z. C. Lipton, C. Elkan, B. Naryanaswamy, Optimal thresholding of classifiers to maximize f1 measure, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2014, pp. 225–239.

[18] M. Ostapowicz, K. Żbikowski, Detecting fraudulent accounts on blockchain: A supervised approach, in: International Conference on Web Information Systems Engineering, Springer, 2019, pp. 18–31.