Record insights and model behaviors to capture fraudulent transactions in Ethereum ecosystem

Sanjay Nayak
Department of Computer Science
Texas A & M University
College Station, USA
sanjaynayak@tamu.edu

Piyush Nayak
Department of Computer Science
Texas A & M University
College Station, USA
kayan_piyush@tamu.edu

***Abstract*—Blockchain technology has become very popular owing to its distributed peer-to-peer network structure, anonymous nature, and robust applications. Cryptocurrency, one of the applications of blockchain technology, has changed the way financial transactions work owing to its cheaper, faster, and more secure methods. In the current times, it has the potential to disrupt the current financial transaction systems. Also, owing to it their anonymity, digital currencies have become very popular in the world of cyber crimes. Ethereum is one of the products of blockchain technology that has become popular and has been widely used as the base for digital money, payments, and contracts as well as in various applications. Frauds in financial transactions have always been a cause of concern for a long time and these frauds have also been reported in the digital currencies era. With the widespread fraud happening in the financial sector, it is very important to not only correctly identify fraudulent transactions but also have a mechanism to prevent them from happening in the future. This calls for efficient fraud detection techniques. With this view in mind, this paper focuses on Ethereum fraud and uses different algorithms that can be utilized to detect these frauds. An Ethereum-based financial transaction record containing fraudulent data has been used to train multiple models and find the best model among them. A combination of machine learning and deep learning models have been trained and compared based on parameters like accuracy, precision, recall, and the ROC curve. The results from the experiments demonstrate the excellent performance of models in detecting fraudulent Ethereum transactions.**

***Index Terms*—Blockchain, Ethereum, Cybersecurity, Fraud Detection, Neural Networks, Machine Learning**

I. Introduction

The concept of blockchain has been present for over 20 years. Blockchain is a peer-to-peer distributed system of networks consisting of a ledger of transactions that is shared over the network. The transactions are consistent chronologically as any block forming a chain in the network cannot be deleted or modified without some consensus mechanisms that span over the whole network. Bitcoin [1], introduced in 2009 by Satoshi Nakamoto, used the concept of blockchain for its payment system, kept its users anonymous, and used the proof-of-work consensus mechanism. The concept of blockchain technology has been utilized in various domains such as financial sectors [2], healthcare industries [3], and the Internet of Things [4]. Reference [5] provides information related to the usage of blockchain technology in the field of entertainment, advertising, copyright protection, and energy industries. It has become popular owing to its decentralized and tamper-proof properties.

Blockchain can be categorized into four types based on authority and permissions. They can be public blockchains, private blockchains, hybrid blockchains, and consortium blockchains. Public blockchains do not have any central body governing the system, while private and consortium blockchains have either a single organization or a group of organizations governing the blockchain network. Bitcoin and Ethereum are part of public blockchain networks.

With a market capitalization of $193 billion, which constitutes around 20% of the cryptocurrency market, Ethereum forms the second-largest cryptocurrency in the world. The main difference between Ethereum and Bitcoin is its programmable nature, which allows users to use Ethereum to build their own applications providing features of the blockchain. Ethereum provides the infrastructure and technology to build applications related to the financial domain, games, and social networks. It uses a proof-of-stake consensus method to add blocks to the blockchain system. One of the important features of Ethereum is the usage of smart contracts, small computer programs that behaves like traditional contracts in the digital world and also serve as the fundamental building block of Ethereum.

With the increase in technological advancements, the number of frauds and their mechanisms has also grown at an alarming rate. Of the frauds being reported and analyzed, frauds related to transactions have the highest percentage as transactions are an essential part of the communication between people. The digital currency has provided a new approach to the financial domain, but the feature of anonymity present in cryptocurrency is being used by fraudsters to commit crimes. Ethereum has also been part of these criminal activities where criminals are using Ethereum in black market deals, money laundering, and other illicit activities as they are not tracked or monitored by the authorities owing to its underlying features. As per Coindesk, even during the Ethereum Merge, where Ethereum transitioned from proof-of-work to proof-of-stake, roughly $1.2 million worth of Ether has been scammed. In 2021 Forbes reported that around $600 million worth of tokens were stolen from Poly, out of which

around $267 million worth of Ether were present.

All these frauds warrant a better fraud detection system that can detect and predict fraud in financial transactions. Research has been conducted related to fraud and other cybercrime activities to reduce the number of frauds and attacks. Ethereum-based transactional frauds can be detected using supervised algorithms that use the concept of machine learning. In the present time, graph-based algorithms [7] and data mining [6] have also been utilized to flag fraudulent transactions.

Blockchain has become one of the most secure and decentralized sources of communication and exchange today and is thus garnering a lot of audiences who want to utilize the platform to serve their personal needs. In the current generation, Ethereum has shown its potential because of its features like smart contracts but has been the center of many major cryptocurrency-related frauds. With the motivation to efficiently detect and flag fraudulent transactions in the Ethereum network in real-time, research has been carried out that utilizes Ethereum fraudulent data set. Multiple models related to both machine learning and deep neural networks were used to determine which algorithms can be efficiently used for Ethereum fraud detection in real-time as well as find out interesting analyses related to the same.

Below are some of the main contributions of this research:

- The paper provides an analysis of the Ethereum Fraud data set using 6 different machine learning and deep neural net models. The details about these methods can be found in Section III.
- Experiments were carried out using the Ethereum Data set and the model's performance was measured using accuracy, precision, and recall. The results show which models performed better in terms of all the 3 parameters mentioned above.

The research paper is organized as follows. Section II provides information about related work in the field of fraud detection using Bitcoin or Ethereum transaction data sets. Section III provides an overview of the six methods used in this research paper for the analysis. Section IV gives an overview regarding the acquisition, pre-processing of the data set, and the experiments performed. Section V discusses the results obtained and Section VI concludes with the work done in the research and provides some insights related to future work.

II. Related Work

Research work related to fraud detection in Ethereum and Bitcoin has used various techniques like machine learning models, graph-based algorithms, and data mining algorithms. Data Mining techniques [6] has been used to detect fraud in Ethereum-based transactions. The research focused on the detection of Ponzi schemes on the Ethereum network as these scams have increased over the years. The researchers built a data set comprising benign Ponzi schemes that use smart contracts and used data mining models in order to build a fraud detection model. Their 0-day model was able to detect the Ponzi schemes as soon as any smart contracts based on Ponzi schemes were uploaded. Their models showed an impressive 0.96 recall rate and 0.98 precision rate.

Research related to anomaly detection in Bitcoin network [9] was done that used three unsupervised learning methods that include k-means clustering, unsupervised Support Vector Machine, and Mahalanobis distance. The research utilized graphs that used the Bitcoin network for the generation. The researchers detected anomalies in the network in the form of transactions and users.

Graph Neural Networks have been used for the detection of phishing in transaction networks. Reference [8] uses the Graph Neural Network (GNN) models on the actual transaction network of Ethereum. They used the labeled phishing data to train the GNN models and did a study on the model accuracy and hyperparameters. They used a public data set of transactions from Kaggle containing 2,973,489 nodes and 13,551,303 edges, and having 1,165 nodes that are labeled as fraud. The researchers did a comprehensive study on homogeneous and heterogeneous GNN models and concluded that RGCN of the heterogeneous GNN model group performed the best in all parameters including F-1 score, precision, recall, and PR-AUC.

Another research utilizing Ethereum transaction records and graph neural network was also conducted in [7]. The researchers proposed their method to detect fraud in Ethereum transactions using the transaction data set of Ethereum. In order to capture addresses that were labeled as fraudulent, they took the help of a web crawler. The researchers designed an identification system that would be able to automatically detect the fraudulent address in Ethereum transactions. For this, they developed a network embedding based on the amount that was used to extract features from nodes. Then a graph convolutional network was used to detect and predict fraudulent or legit addresses. Their experiments achieved an impressive accuracy of 95%.

The research was conducted in [10] related to the performance of various machine learning models on the Bitcoin and Ethereum transaction networks to detect anomalies. The researchers made use of machine learning models including Random Forest, Support Vector Machines (SVM), and Logistic regression that was accelerated by GPU. A data set containing 30 million transactions over the network of Bitcoin as well as from the Ethereum network. The researchers verified and provided analysis regarding the performance of the machine learning models with respect to parameters such as confusion matrix, precision, recall, accuracy, and F-1 score. The data set used by the researchers used 30,294,698 Bitcoin transactions out of which over 30 million records were legit and 4,653 records were fraudulent. They were able to achieve a 0.987 recall, 0.987 accuracy, and 0.994 F-1 score for SVM for the Bitcoin network. They also achieved a recall of 0.834 accuracy, 0.835 recall, and an F-1 score of 0.909 for the Random Forest Classifier for the Ethereum network. The above was considered the best model after the analysis.

III. Methods

This section describes the different models utilized for experiments. For our research on the Ethereum Fraud data set, the following machine learning and deep neural network models are being used. The analysis is being conducted on the six different models using the parameters including precision, recall, accuracy, and the ROC curve.

A. Random Forest Classifier

This classifier is a meta-estimator that takes into account multiple decision trees and fits each of the decision tree classifiers on a random subset of the input data set. It uses the method of averaging to make the final prediction. This helps in making the accuracy better and also helps in controlling the over-fitting. This classifier was implemented with the help of the *sklearn* library of Python.

B. Support Vector Machines (SVM)

Support Vector Machines provides a set of supervised learning techniques with the objective to find a hyperplane to split the D-dimensional data, where D represents the number of features, with the largest margin. SVC kernels are utilized to map the non-linearly separable data to a linearly separable feature space. SVMs are versatile and can be used for the classification, regression, and in the detection of outliers. This implementation was done using the *sklearn* library of Python.

C. XGBoost Classifier

Extreme Gradient Boosting (XGBoost) is used in supervised learning having features such as high efficiency, flexibility, and portability. It uses the Gradient Boosting framework for its machine learning algorithms. The main idea behind the boosting algorithm is to reduce the loss with the addition of weak learners, i.e. it tries to create a strong learner from a set of weak learners. This classifier was implemented using the *xgboost* library of Python.

D. Gaussian Naive Bayes

This classifier is a supervised algorithm that is based on the Bayes theorem, where every feature is considered to be conditionally independent giving the name Naive Bayes. The assumption in the Gaussian Naive Bayes is that every class follows the Gaussian distribution. This classifier was implemented using the *sklearn* library of Python.

E. Logistic Regression

Logistic Regression is used as a classification algorithm that is used to estimate or predict the probability of the output classes or labels given a set of features or independent variables. It consists of one input layer and one output layer, and a sigmoid function is used to estimate the probability. For the implementation of logistic regression in the current research, Pytorch was used.

F. Advanced Neural Network

Neural Networks are a part of deep learning methods that resemble the functioning of neurons in the human brain. It consists of an input layer, an output layer, and one or more hidden layers that consist of interconnected nodes that process the input features using weights associated with each node. The weight values are adjusted based on the output in each iteration during the backpropagation technique. The implementation was done using the Pytroch library and consisted of two linear layers using ReLu and BatchNorm, and the sigmoid function. The dropout technique was also introduced before the output layer to increase the performance.

For analysis and comparison between different models, parameters including precision, recall, ROC curve, and accuracy were used.

- **Precision:** Ratio of true frauds to those classified as frauds.

$$precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

- **Recall:** Ratio of frauds recognized correctly to the total number of frauds in the data set.

$$recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

- **ROC curve:** It is used to depict a trade-off between True Positives or actual frauds and False Positives or legit transactions categorized as frauds.
- **Accuracy:** It measures how well the model has performed by correctly predicting the frauds and legit transactions.

$$\begin{aligned} accuracy &= \frac{TruePositive + TrueNegative}{P + N} \\ P &= TruePositive + FalseNegative \\ N &= FalsePositive + TrueNegative \end{aligned} \tag{3}$$

IV. Experiments

The following steps were taken to collect, prepare the data set, extract relevant features, train the models with the processed data set, and use the results to analyze and find the best model from among them.

A. Data Collection

The Ethereum transaction data was collected from Kaggle [11]. It had 9,841 rows and 51 columns containing transactions from the Ethereum network, with 7,662 legit transactions and 2,179 fraudulent transactions. Out of the 51 columns present in the data set, 50 columns describe the transaction features, and the FLAG column defined whether a transaction is fraudulent or legitimate.

B. Data Pre-processing

Data pre-processing is done to remove unwanted features and clean the data for model training. This helps in finding out the relation between different features and checking if the feature can be utilized for model training.

Columns 'Unnamed: 0', 'Index', and 'Address' are removed as it is not required in the data analysis and modeling. Two more columns having categorical data were removed because they did not have any relation with the current analysis. Correlation between features was used to find out the relationship and importance between features and features with correlation values greater than 0.7 are removed from the dataset. This led to a reduction in the number of features from 51 to 27. Fig. 1 depicts the correlation between the 27 features.

C. Feature Extraction

The number of features was further reduced using the box plot method, where three features having very low distribution and zero covariance are removed from the data set reducing the number of features to 24. Fig. 2 shows the box plot distribution used for four features used in the final training of models. Since the data set was imbalanced, the *SMOTE* method was used to balance it out and the *MIN-MAX Scaler* method was used to scale the features in the data set between 0 and 1. To further narrow down the number of features in the data set, *Recursive Feature Elimination* was used, and the top 15 ranked features were used for the final model training.

D. Model Selection and Training

The six models described in Section III were used to train on the processed data set. The data were divided into training and testing sets with a 4:1 ratio. It was very important to balance the training data before passing it into the models. The logistic Regression model was run for 50,000 epochs with a learning rate of 0.01, whereas the neural network was run for 20 epochs, with a learning rate of 0.01, and the two hidden layers consisted of 20 and 8 hidden nodes respectively.

V. Results

To analyze and assess the results, a confusion matrix was created and the metrics described in (1), (2), and (3) are derived. Out of these metrics, we are particularly interested in the recall metric as a low recall is undesirable owing to the fact that a low recall indicates a high number of false negatives. Each of the six models was trained with the same final processed data set and the Neural Network model was additionally trained with 10-fold cross-validation. In each of the metrics, the best value is indicated in bold.

It can be noticed from Table I that Random Forest does the best in terms of accuracy, while Neural Network and Random Forest perform equally in terms of precision, and XGBoost Classifier and Naive Bayes performs the best in terms of recall. The Random Forest Classifier was trained with depths=100, and both Logistic Regression and Neural Network were trained using Pytorch. Fig. 4 shows the confusion matrices generated for each of the six models used. From the result, it is evident that SVM and Naive Bayes performed poorly in terms of precision even though their recall is satisfactory. The confusion matrices show the exact number of true positives, false positives, false negatives, and true negatives which was useful for the models' analysis. It was expected for the neural network to perform well w.r.t all of the metrics because it is able to find the hidden information from the features of the data more efficiently than other models.

The ROC is a good measure to measure the performance of models for cybersecurity-related analysis as it is not affected by the imbalanced nature of the data set. We are more concerned with the ROC of the fraudulent classes. From Fig. 3, it can be noted that the *Area Under the Curve* (AUC) is maximum for Random Forest and XGBoost Classifier, followed by Neural Network. Logistic Regression and SVM have almost the same AUC. The Gaussian Naive Bayes performs the worst in terms of the AUC.

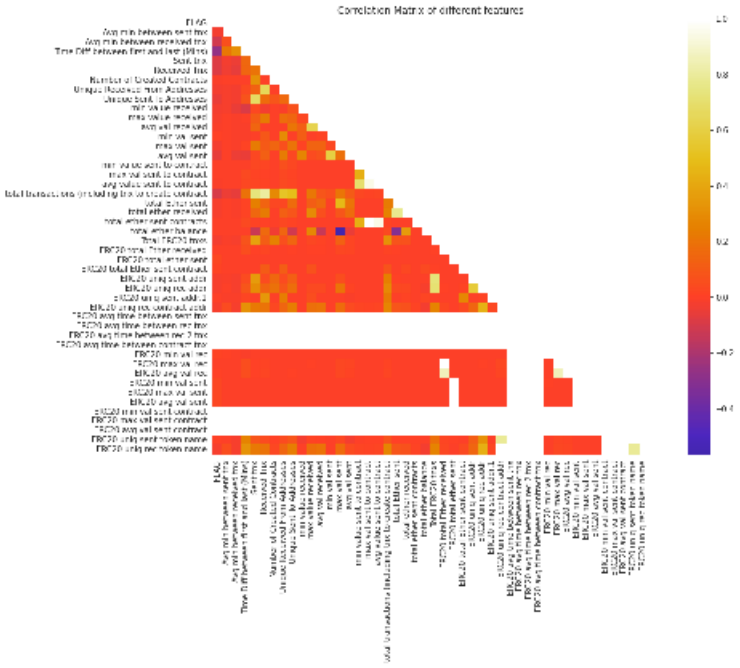

Fig. 1. Correlation matrix after feature removal

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT MODELS USED ON ETHEREUM FRAUD DETECTION DATA SET

Model	*Accuracy*	*Precision (Fraud*	*Recall (Fraud)*
Random Forest	**97.866**	**0.96**	0.94
SVM	74.301	0.45	0.89
XGBoost Classifier	97.105	0.91	**0.96**
Naive Bayes	41.34	0.27	**0.96**
Logistic Regression	82.843	0.86	0.56
Neural Network	95.752	**0.96**	0.95

VI. Conclusion

Even though blockchain is secure and tamper-proof in itself, it is correct to say that it is not crime-proof. Because of its anonymity feature, criminals are using it for various illicit purposes. By using the methods of phishing, cyber-attacks are done on the cryptocurrency wallet. These attacks show that there is a huge requirement for fraud detection, particularly in the case of blockchain and cryptocurrency.

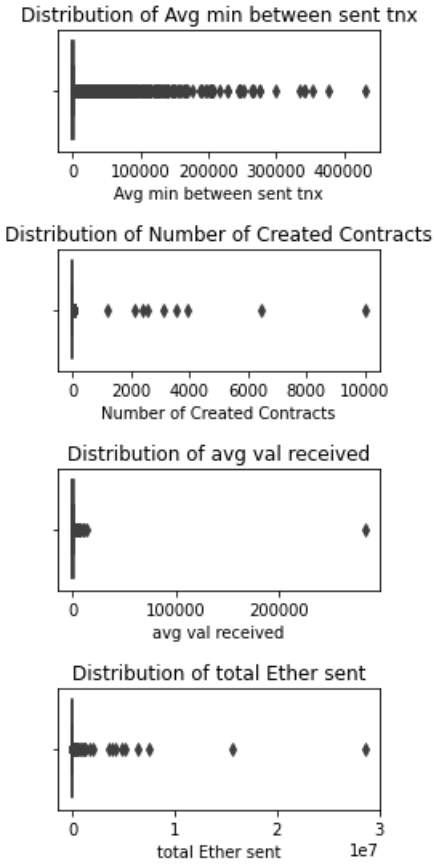

Fig. 2. Box plot representation of 4 features

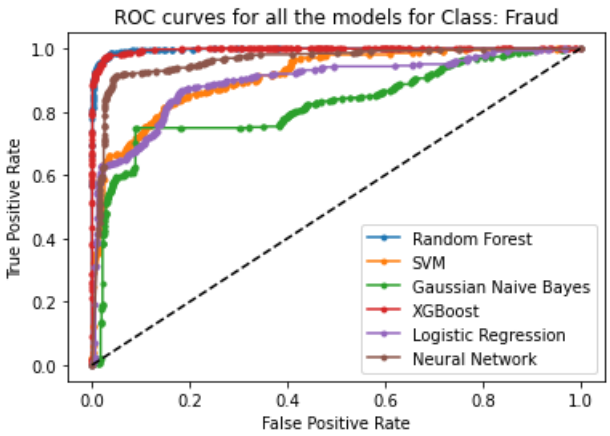

Fig. 3. ROC curve of different models

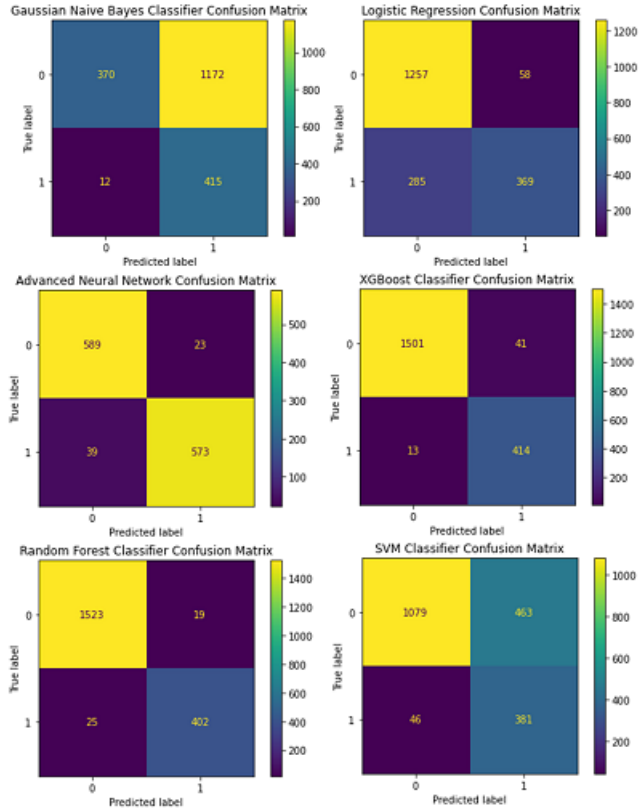

Fig. 4. Confusion Matrices of different models

Through this research, the behavior and performance of six different machine learning and neural network models were analyzed by using the Ethereum Fraud Detection data set.

It was concluded that Neural Networks and Random Forest Classifiers show promising results w.r.t. classification and detection of frauds. The high performance of neural networks can be attributed to their hidden layers, which use the features from their previous layers to learn more efficiently. The Neural Network implemented in the paper is a basic version that consists of linear layers and uses Batch Normalization, ReLu, dropout, and Sigmoid function to train its nodes. We also gave an overview of the performance of the models based on different metrics necessary to analyze the cyber attacks. In the future, further research will be carried out using different forms of deep neural networks like Graph Neural Networks, Autoencoders, and Convolutional Neural Networks, These networks are computationally heavy and will provide a different aspect to approach the fraud detection scenarios.

REFERENCES

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
[2] S. Singh and N. Singh, ”Blockchain: Future of financial and cyber security,” 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016, pp. 463-467, doi: 10.1109/IC3I.2016.7918009.
[3] M. Mettler, ”Blockchain technology in healthcare: The revolution starts here,” 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), 2016, pp. 1-3, doi: 10.1109/HealthCom.2016.7749510.

[4] P. Urien, "Blockchain IoT (BIoT): A New Direction for Solving Internet of Things Security and Trust Issues," 2018 3rd Cloudification of the Internet of Things (CIoT), 2018, pp. 1-4, doi: 10.1109CIOT.2018.8627112.

[5] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao. 2018. A Survey of Blockchain Applications in Different Domains. In Proceedings of the 2018 International Conference on Blockchain Technology and Application (ICBTA 2018). Association for Computing Machinery, New York, NY, USA, 17–21. https://doi.org/10.1145/3301403.3301407.

[6] E. Jung, M. Le Tilly, A. Gehani and Y. Ge, "Data Mining-Based Ethereum Fraud Detection," 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 266-273, doi: 10.1109/Blockchain.2019.00042.

[7] R. Tan, Q. Tan, P. Zhang and Z. Li, "Graph Neural Network for Ethereum Fraud Detection," 2021 IEEE International Conference on Big Knowledge (ICBK), 2021, pp. 78-85, doi: 10.1109/ICKG52313.2021.00020.

[8] H. Kanezashi, T. Suzumura, X. Liu and T. Hirofuchi, " Ethereum Fraud Detection with Heterogeneous Graph Neural Networks," https://doi.org/10.48550/arXiv.2203.12363.

[9] T. Pham, and S. Lee (2016). Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods. arXiv. https://doi.org/10.48550/arXiv.1611.03941.

[10] Y. Elmougy and O. Manzi, "Anomaly Detection on Bitcoin, Ethereum Networks Using GPU-accelerated Machine Learning Methods," 2021 31st International Conference on Computer Theory and Applications (ICCTA), 2021, pp. 166-171, doi: 10.1109/ICCTA54562.2021.9916625.

[11] V. Aliyef, 2020, Ethereum Fraud Detection Dataset, Version 1, https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset.