International Journal of Computing and Digital Systems
2024, VOL. 17, NO. 1, 1–10

http://dx.doi.org/10.12785/ijcds/XXXXXX

Predictive Analysis of Ethereum crypto-currency smart contract Transactions using Integrated Machine Learning Approach

Sheik Abdullah Abbas[1], **Priyadarshini Ramasubramanian**[2], **Utkarsh Mishra**[3] **and Samarth Sathyananda Prabhu**[4]

[1,2,3,4]*School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India*

Received 12 March 2024, Revised 15 June 2024, Accepted 10 July 2024

Abstract: Identification of Anomalies in Online Transactions [Anomaly detection] that can eliminate Risk such as Financial Frauds, Illegal Money Transaction and Anti-Money Laundering is very important. With the cryptocurrency market moving quickly, there is a rise in the need for preventing fraud especially with Ethereum (an open-source blockchain platform that enables developers to build and execute smart contracts). In 2020, Cryptocurrency frauds in the United States reached over 80,000 cases just that year and other countries such as Australia and UK being no different from having same challenges with numbers of his sort. This study proposed a hybrid analysis for detecting fraudulent transactions on the Ethereum network using machine learning and deep learning techniques. In particular, it utilizes Decision Trees, Neural Networks, Random Forest classifiers and SVM as well as Deep Convolutional Neural Network (CNN) models for echocardiogram classification with a focus on the Random Forest classifier. Following strict parametric evaluation and statistical analyses using Fisher's F-Test (p-value ¡ 0.001)) the Random Forest outperformed all other classifiers with an accuracy of 95.56%. This makes it effective in reducing the overfitting problem related to decision trees, and subsequently improving classification accuracy. Our results emphasize the need for extracting features from complex smart contracts and identifying anomalous transaction. The proposed model can serve as a secure way of validating cryptocurrency transactions, especially within the Ethereum ecosystem, which signals sustained and increased consumer adoption.

Keywords: Anomaly Detection, Financial Fraud, Cryptocurrency, Ethereum, Smart Contracts, Machine Learning, Deep Learning, Random Forest, Decision Trees, Neural Networks, SVM Classifiers, Deep Convolutional Neural Networks (CNN), Fraud Detection, Statistical Analysis, Fisher's F-Test, Model Accuracy, Blockchain Security, Transaction Monitoring

1. INTRODUCTION

This project is mainly about cryptocurrences. Our digital transactions has been able to done through the use of cryptocurrency. In the past decade, heightened awareness and understanding of cryptocurrency have led to a significant increase in its usage among people. The Global Cryptocurrency Market is projected to scale up nearly 24 billion in the Year 2023. In 2020, flagship cryptocurrencies such as Ethereum saw a daily transaction throughput of around one million. This really shows how the use has been skyrocketing. There are a high number of benefits in performing your transactions or business apps through cryptocurrency. Benefits of the existing network include transaction privacy with encrypted private transactions, scalability and performance nearly equal to that from a central authoritative source (a TPS rate between 800 – 2000) and large number advantageous greatly, most importantly there will not be government intervene because no one person own this code base. The cryptocurrency I will be looking at for this is Ethereum. Obviously, like any other cryptocurrency Ethereum works on a developing blockchain framework. Ethereum has the second highest market value, behind only Bitcoin. The thing is that Ethereum with its difference has some advantages over Bitcoin — it is more versatile and can handle faster transaction times than those of Bitcoin. Even if the blockchain and cryptocurrencies are still trying to develop, there is never ending problem of fraudulent transactions which needs to be put down.

The number of users using Ethereum alone has increased steeply in the last 3 years. This indicates the awareness about cryptocurrency that is reaching all types of users. The alarming rate at which these fraudulent transactions in cryptocurrency have been happening is a cause of concern. The growth of cryptocurrency could be hampered if immediate action is not taken. Users of poorer financial background

E-mail address: aa.sheikabdullah@gmail.com, utkarsh.mishra2021@vitstudent.ac.in, priyadarshini.r@vit.ac.in, samarthsp25@gmail.com

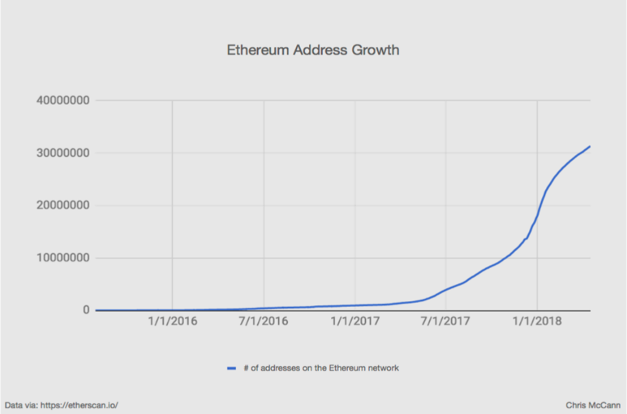


Figure 1. Ethereum Address Growth

may not be able to recover from such frauds. They should be well assured so that they can use cryptocurrency which will also aid their financial growth. Trust is in part what ultimately allows progress to be made with cryptocurrency. We want the entire Ethereum ecosystem to be open and inclusive for the largest number of people with different socioeconomic backgrounds as possible. It is important to identify both what makes a fraudulent or legitimate transaction. Furthermore, there is an important element — to add more understanding analyze these attributes and visualize the patterns/relationships within them.

Earlier we discussed various fraudulent activities that could be performed with respect to Ethereum and hence, the purpose of this project will remain to build such a model using machine learning algorithms which can detect fraud in these ethereum transactions so that they don't happen obviously on one side it reduces its value but also definitely provide very secured environment for its users.

2. Scope and Limitations

Ethereum, in particular, is growing and being adopted at an incredible rate across industries. We are going to highlight Ethereum which is one of the top cryptocurrencies in terms of market capitalization, and a relatively strong blockchain technology that has been featured across various applications such as decentralized finance (DeFi), smart contracts with NFT. Although technological developments and popularity have increased, the problem of cleptocamatic behavior is still a significant obstacle in Ethereum. Scams, phishing and double-spending are common methods to defraud a user in this space plaguing the ecosystem. Remarkably, this issue has not been paid much attention to (neither by the community nor researchers) at all which constitutes a crucial missing piece in Ethereum security. This situation was a good chance to dig deeper about this issue, research the sources and reasons of frauds in transactions, and also innovate ideas for falling down these risks. We are working on this because we want to help the Ethereum network achieve better security and stability.

As this is a project focused on Ethereum, the analysis we will conduct in our work cannot include all of cryptocurrency. There are many different cryptocurrencies out there, and by focusing solely on Ethereum, we could miss other things that might be driving the price of Bitcoin — ones such as digital currencies like Binance Coin or Ripple. The fact is that each cryptocurrency employs its own unique protocols and dynamic mechanisms, which can bring up a litany of both challenges as well as opportunities to counteract the constantly evolving criminal use cases powering fraud networks. Attempting the same tests without using multiple cryptocurrencies would reveal a smaller range of vulnerabilities and mitigation strategies than what is shown by testing on Ethereum alone. Furthermore, we do not know what were the time period of dataset used in our study and therefore cannot assess any trends over specific seasons or how fraudulent activities are evolving over a course. Knowing the timeframe of the data set could be useful for calling out time periods in which there has been a spike in fraud, or where particular events occur that are related to when fraudulent transactions skyrocket. Hence, it may be worthwhile to use a broader variety of cryptocurrencies and narrow down time frame considered in this dataset so that we can make the investigation more comprehensive and compelling.

3. Literature Review

Many studies have been conducted using machine learning algorithms to detect fraud and anomalies in the cryptocurrency space. In reference [1], Eunjin et al. applied Random Forest, Stochastic Gradient Descent (SGD), and J48 algorithms to identify Ponzi schemes on the Ethereum blockchain. The researchers gathered hyperlinks pointing to smart contract addresses and then used the Etherscan API to extract transaction data associated with these contracts. All addresses related to Ponzi schemes were sourced from previous research. In this paper we reported a data mining characterization for the discovery of deceptive contracts attaining precision 0.99 with SGD and recall 0.97 with J48. An obvious shortcoming in the work is that for a large dataset, especially when one consider using Etherscan API to collect data manually verify account relevance with ponzi scheme necessary. Farrugia et al. [2] have used psychological features to detect illegal accounts using supervised machine learning algorithms. The dataset utilized in this study was obtained from Etherscamdb and a local Geth client. After retrieving and preprocessing the data, we applied the XGBoost model.To improve the model, we conducted a grid search along with cross-validation to identify the best hyperparameters. The model achieved a recall of 0.963 with the optimized parameters. The authors have effectively attempted to fine-tune all parameters to achieve the best possible accuracy. It is surprisingly easy to develop simple computational techniques that can bypass the account detection system outlined in this paper. Reference [3] conducted a similar analysis employing decision trees, Random Forest, and K-Nearest Neighbors to identify fraudulent accounts. Ibrahim et al. conducted their research using a dataset available on Kaggle. This dataset consists of 42 features, and the goal of our study is to identify the six most important ones using feature selection algorithms. Our current work has achieved a higher F score, ranging from 31% to 52%, compared to previous results.

In [4], the authors employed both unsupervised and supervised methods to identify outliers in cryptocurrency transactions. The data was clustered using the expectation maximization algorithm, followed by the application of a Random Forest model to detect fraudulent transactions. The model exhibited remarkable performance, with metrics like precision, accuracy, and F1 scores all surpassing 90%, and even reaching as high as 99% in certain instances. The primary strength of this study lies in its remarkable accuracy.

In [5], the focus is on detecting malicious entities–Ethereum. Using various techniques like Logistic regression, SVM and random forest2 AdaBoost3 Stacking Classifier The methodology proposed in these cases. The best performing model after training on the different models, is employed for detecting malicious entities in Ethereum. Here, the highest F1 score is achieved with ensemble methods (99.6%) The benefits of such an approach are that it is capable to cope with the task in a scenario when dataset could be skewed and also should work well for analogous tasks.

In [6], Madhuparna et al. The implementations of the following study analyzed how machine learning algorithms perform when trained with fraudulent and legitimate transactions. They utilized the following algorithms for fraud detection: Logistic Regression, Multilayer Perceptron, Naïve Bayes, AdaBoost, and Decision Tree algorithms were applied using a threshold of 0.35. If a probability is 0.35 or higher, the instance is classified as class 1; if it is below 0.35, it is assigned to class 2. They also implemented Support Vector Machines and Random Forest Neural Networks, featuring four hidden layers. The maximum accuracy attained was 97%, using AdaBoost, Random Forest, and SVM. This method for detecting fraudulent transactions circumvents the limitations of account-based detection that can be evaded by existing computational algorithms.

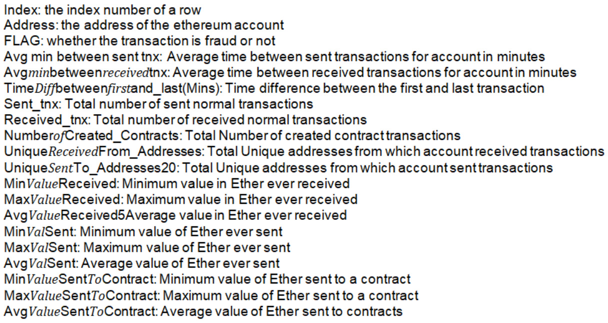
In [7], the authors classified nearly 400,000 accounts to identify fraudulent ones using only Random Forests, Support Vector Machines, and XGBoost classifiers. The results indicate that these methods can achieve recall and precision values that are sufficiently robust to function as an anti-fraud rule for the digital wallets or currency exchanges implemented in our system. They also performed a sensitivity analysis to illustrate the models' dependency on a specific feature and how the absence of this feature affects overall system performance.

[8] explores the untrusted users of cryptocurrency transaction-service nodes based on smartphones and computers. But as the technology is getting advance, transaction frauds are also increasing and Companies has to identify any loop holes in its systems. To identify suspicious users, we propose a methodology for classifying active users as either malicious or benign based on their reputation scores. This approach utilizes centrality measures combined with state-of-the-art machine learning techniques. We evaluated our results on two real-world cryptocurrency network datasets: Bitcoin-OTC and Bitcoin-Alpha, which include a trust score that reflects the system's characteristics as well as each user's individual status. Our testing indicated that the proposed solution offers enhanced robustness. Thus, integrating machine learning with centrality measures leads to a resilient system capable of adapting dynamically to safeguard the financial services of smart devices.

4. Proposed Methodology

A. Dataset

The dataset utilized for our analysis and predictions is sourced from Kaggle. It consists of 9,841 records and a total of 50 columns, with the "Flag" column serving as our target variable. This column includes two values: 0 (Legitimate Transaction) and 1 (ounterfeit Transaction). The other 48 columns will be preserved as independent variables for prediction purposes.



Index: the index number of a row
Address: the address of the ethereum account
FLAG: whether the transaction is fraud or not
Avg min between sent tnx: Average time between sent transactions for account in minutes
Avg*min*between*received*tnx: Average time between received transactions for account in minutes
Time*Diff*between*first*and_last(Mins): Time difference between the first and last transaction
Sent_tnx: Total number of sent normal transactions
Received_tnx: Total number of received normal transactions
Number*of*Created_Contracts: Total Number of created contract transactions
Unique*Received*From_Addresses: Total Unique addresses from which account received transactions
Unique*Sent*To_Addresses20: Total Unique addresses from which account sent transactions
Min*Value*Received: Minimum value in Ether ever received
Max*Value*Received: Maximum value in Ether ever received
Avg*Value*Received5Average value in Ether ever received
Min*Val*Sent: Minimum value of Ether ever sent
Max*Val*Sent: Maximum value of Ether ever sent
Avg*Val*Sent: Average value of Ether ever sent
Min*Value*Sent*To*Contract: Minimum value of Ether sent to a contract
Max*Value*Sent*To*Contract: Maximum value of Ether sent to a contract
Avg*Value*Sent*To*Contract: Average value of Ether sent to contracts

B. Description

There are several stages involved in the implementation of this project. All these steps have to be done sequentially to ensure the reliability and efficiency of the results to be obtained. The first stage involves the collection of the data. We went through many possible datasets before selecting a dataset in Kaggle with nearly 10000 records and 50 features. The next stage involves an extensive process of data preprocessing steps. They include processes such as:

1) Dropping unnecessary columns
2) Data Imputation
3) Removing features with 0 variance
4) Handling imbalanced dataset
5) Data Normalization
6) Feature Selection

After the data preprocessing stage, we will move onto the exploratory data analysis stage. This involves plotting various visualizations such as bar graphs, scatter plots, boxplots and obtaining valuable inferences from them. These insights will help us to understand the problem and underlying reasons for them.

The last stage the training of the various models. These is an important stage where we will determine the most efficient model using various metrics. A total of 5 algorithms will be implemented. They will be discussed in detail in the implementation stage. After deciding on the most efficient model, we will come to a conclusion for our problem statement. These are the step-by-step processes involved in our methodology.

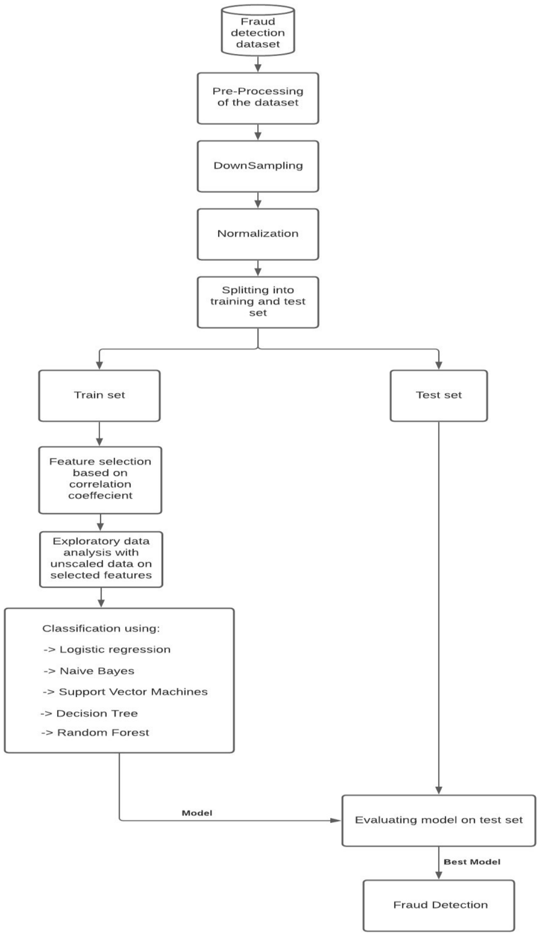

Fraud detection dataset
Pre-Processing of the dataset
DownSampling
Normalization
Splitting into training and test set
Train set
Test set
Feature selection based on correlation coeffecient
Exploratory data analysis with unscaled data on selected features
Classification using:
-> Logistic regression
-> Naive Bayes
-> Support Vector Machines
-> Decision Tree
-> Random Forest
Model
Evaluating model on test set
Best Model
Fraud Detection

Figure 2. Proposed Methodology

C. Data Preprocessing

Several steps were needed to be taken to overcome the limitations of the data taken. The data pre-processing stage was carried out in a step-by-step manner so as to bring it to a state where it can be used for the implementation of the models accurately.

The first step was to remove the index and character columns as they were not important to the results of the model to be implemented. Initially there were 50 columns. Now, 4 columns were dropped to reduce it to 46 columns.

1) Checking for Missing Values

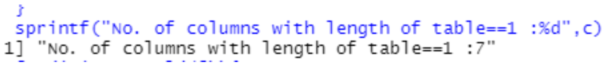
Carefully examine your data for missing values as it greatly impacts the results. The data was having 19067 missing values. We visualized the columns with missing values and found out how many of them were in those columns. There were just a little over 800 missing values for all those columns with NA.

2) Data Inputation

Next, we proceeded to implement data imputation. We then have to replace the missing values in these columns with mean value of that particular column. To use the mean function, the columns have to be of numeric type so they are first converted to numeric type then each of the missing values is replaced by the mean of their respective columns.

3) Single Value Columns

Next, the columns containing a single value were dropped. These columns have zero variance and thus are insignificant to the results we will obtain. By removing these columns, we reduce the feature space that we will be working on. 7 columns were removed through this process.



```
 }
 sprintf("No. of columns with length of table==1 :%d",c)
1] "No. of columns with length of table==1 :7"
```

4) Imbalanced Dataset

The next stage illustrated the distribution of classes, revealing a significant disparity between the two output categories: valid transactions and fraudulent transactions.

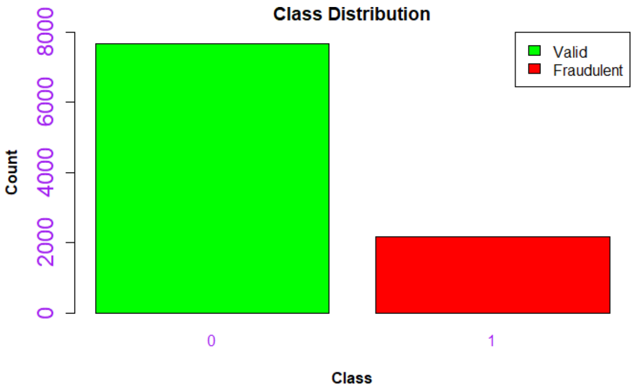


Figure 3. Class Distribution Indicating difference between classes

The valid transaction class contains over three times as many records as the fraudulent transaction class. Specifically, the number of records in the valid transaction class exceeds three times that of fraudulent transactions. It is a very bad scenario for the model, as having more diversity and not normalized values can bias towards majority class. Our goal is to flag the transactions as fraud and this imbalanced class will put a stronger impact on detecting the fraudulent transaction. This means this imbalanced data need to be treated. Then we down sample. Now the data is balanced between the 2 classes. So now Every class includes 2179 records.

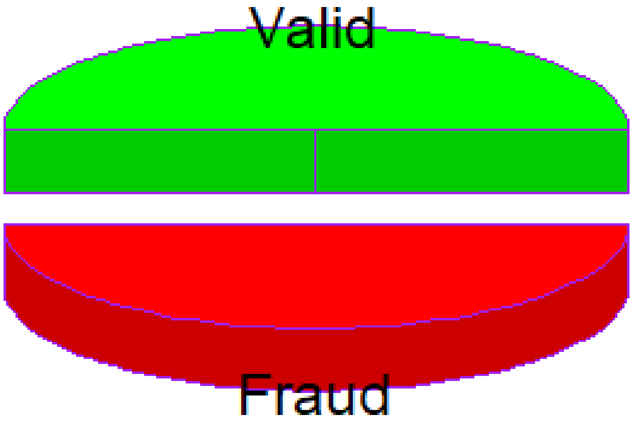


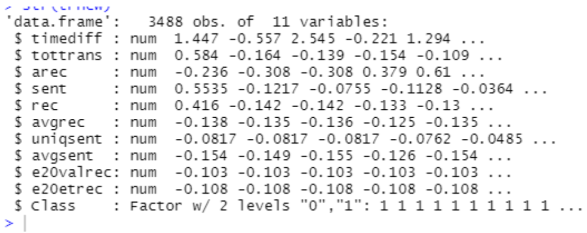
Figure 4. Data Balanced

D. Data Normalization

There is no target variable here included and 38 features Therefore, we have 38 features that can be from different units and scales. It's bad for our results because it hints that not all features contribute the same way to how we want labels (increase bias). So, to avoid this weakness the method of data normalization is performed. The scale() function in R is deployed to do this. This function is responsible for centering the data in a column. It gives us the average and standard deviation for this column. Each element is zero-centered and normalized by dividing it with the mean, standard deviation.

E. Feature Selection

The data is partitioned for the first time by randoming selecting 20% as the test set using a rule of thumb called an 80–20 rule. Feature Selection - Top 10 important features are selected which will be useful in further step of model fitting. Presently, we are having a dataset of 39 features. We can improve the power of our answers with trend analysis and increasing focus on redundancy. This uses a feature selection method to compute the correlation coefficient values between all features and target from this dataset as well, sorts in ascending order respective of correlation coefficients value and selects 10 best rows for more analysis.



```
'data.frame':   3488 obs. of  11 variables:
 $ timediff : num  1.447 -0.557 2.545 -0.221 1.294 ...
 $ tottrans : num  0.584 -0.164 -0.139 -0.154 -0.109 ...
 $ arec     : num  -0.236 -0.308 -0.308 0.379 0.61 ...
 $ sent     : num  0.5535 -0.1217 -0.0755 -0.1128 -0.0364 ...
 $ rec      : num  0.416 -0.142 -0.142 -0.133 -0.13 ...
 $ avgrec   : num  -0.138 -0.135 -0.136 -0.125 -0.135 ...
 $ uniqsent : num  -0.0817 -0.0817 -0.0817 -0.0762 -0.0485 ...
 $ avgsent  : num  -0.154 -0.149 -0.155 -0.126 -0.154 ...
 $ e20valrec: num  -0.103 -0.103 -0.103 -0.103 -0.103 ...
 $ e20etrec : num  -0.108 -0.108 -0.108 -0.108 -0.108 ...
 $ Class    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
> |
```

Figure 5. Selected Features

The correlation value between these features and the target variable is visualized.

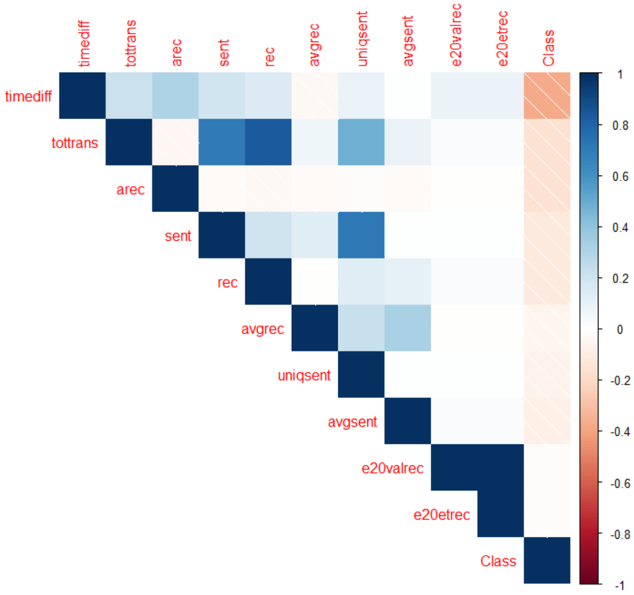


Figure 6. The correlation value between these features and the target variable

Correlation coefficient figure: for ease of comparison, you can view the numbers in this image:

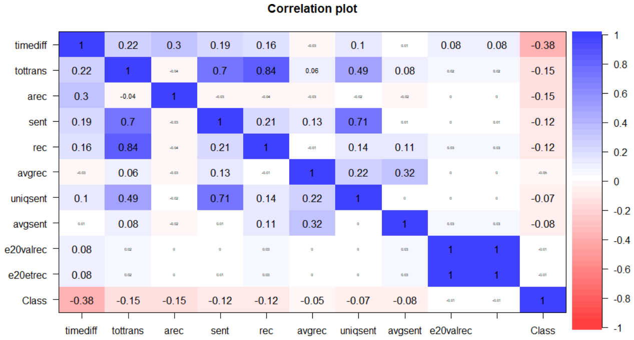
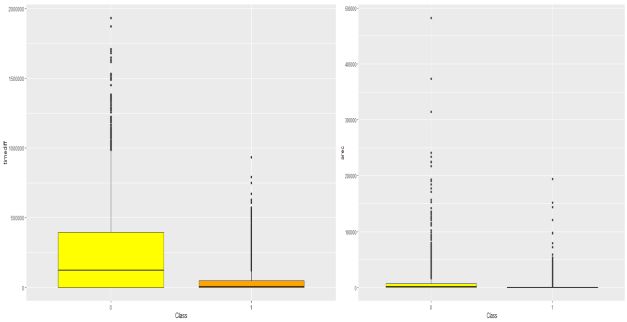


Figure 7. The numerical values of the correlation coefficient

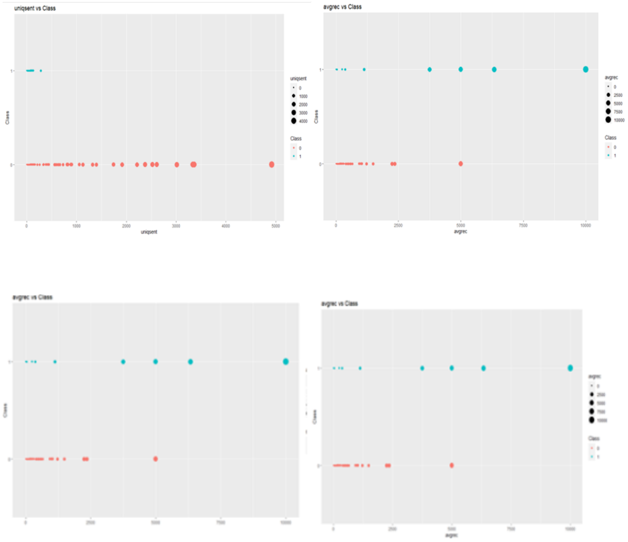
For class you can see timediff has the highest correlation with an absolute value of 0. Correlation between tottrans and areac 0.15 Again this is why there would be similarity between the higher value of correlation and it will show by dark color.

We have now reduced the feature space from 39 to 11. The class attribute is the dependent variable and the other extracted features are our independent variables. The analysis will be done on each of the independent variables. Analysis will also be done with respect to each class value (Valid or Fraud). Appropriate inferences will be drawn from them.

1) Boxplot



Boxplot for the Timediff and areac features was plotted. The boxplot was grouped by class label. It was found that many of the values lied outside the maximum range of the boxplot. The median is also squashed towards 0. This is due to the high concentration of records with value of 0. Because of this even values which are genuinely valuable seem like outliers according to the boxplot. This trend is followed for all other features as well.



Scatterplots were plotted to visualize the relationship between each column and the target variable. Furthermore, while plotting they were grouped based on the label that they belong to so that inferences could be drawn up for both the labels separately. Among all the 10 scatter plots plotted with each of the 10 columns, it can be noted that valid transactions could be separated from the fraudulent ones using the value of the respective column. For instance, in the case of the time diff column, the maximum in the case of fraudulent transactions does not exceed 1000000 but however, the time difference in the case of a valid transactions goes all the way up to 2000000. Therefore, if a transaction has a time diff value of 1500000 then it is highly likely to be a valid transaction. Similar inferences could be drawn from total transactions column, sent tnx column, unique addresses sent to column and avg value received column.

5. Summary Statistics

1) Valid Transactions

INSIGHTS

- When observing the summary statistics of valid transactions, it can be seen that the mean of every column is greater than its median. This indicates a positive skewness for each feature. Thus, definitely they do not follow a normal distribution.
- In most of the columns we can observe a very large maximum value, which indicates the stability of the account. For example, if we take the tottrans column a large number in this feature means that a user has carried out many transactions and has not had the need to change his account for any reasons.

2) Fraudulent Transactions

INSIGHTS

- The summary of the various statistical measure of every column was obtained using the summary() function in r
- It could be seen that fraudulent transactions could be separated from the valid ones with the help of various statistical summaries as well.
- In the case of time difference column, the first quartile value is 0 in the case of fraudulent transactions and in the case of valid transactions its more than 300. Furthermore, the mean and median values of the column also differ vastly for fraudulent and valid transactions. The inferences based on the min and max values for both the labels with respect to this particular column has already been explained with the help of scatterplots.

6. Predictive Analysis

This project aims to identify which Ethereum transactions are fraudulent and do so automatically and accurately. Prediction of the transactions will be done using 5 models. Models:(with algorithm Logistic Regression, Naïve Bayes, Decision Tree, Random Forest and Support Vector Machine)

The performance of these 5 models will be analyzed. Through this comparative analysis, the best model suitable for fraud detection will be identified.

A. Algorithms Implemented

1) Logistic Regression

It is a categorical data classification algorithm. We have binary outcomes in this case and so, I used Logistic Regression which is most suitable for the situation. It makes use of the sigmoid function as the activation function. It takes in the output of the hypothesis function as the input and squashes it and returns a value between 0 and 1. The output label is 1 if the value returned is greater than the threshold set and the output label is 0 if the returned value is less than the threshold set.

- Logistic Regression is implemented using the glm function in R. Logistic Regression is part of a larger class of techniques known as the generalized linear model.

- The model is trained using the training data, and a summary of the trained model is presented above.

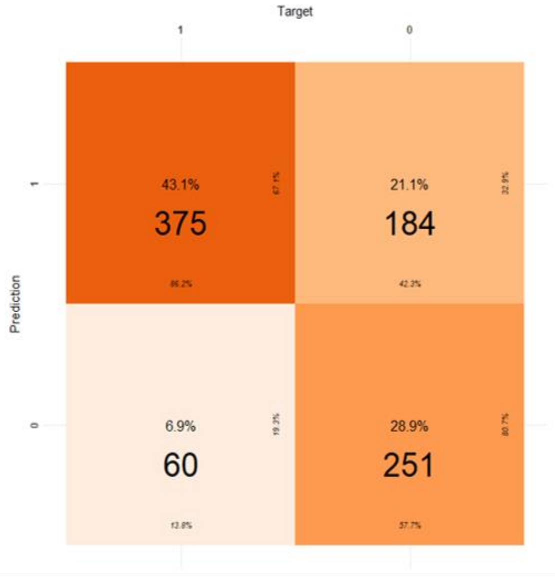


Figure 8. Confusion Matrix – Logistic Regression

2) *Naïve Bayes*

Naive Bayes is a supervised machine learning algorithm that utilizes probabilities to predict and classify labels or classes based on a given instance. Conditional Probability and Bayes Theorem Naïve Bayes is a family of algorithms for differentiating types of classes, such as Multinomial Naive Bayes, Gaussian naive bayesian and Bernoulli NB. Naive Bayes is easy to implement but a very strong classification algorithm.

- The accuracy of the Naïve Bayes model is 0.6, which is subpar compared to the previous model we implemented.

- Given either valid or fraudulent transaction, we find the probability that a test set records belongs to that particular class. Then we assign it to the class that generates a higher probability.

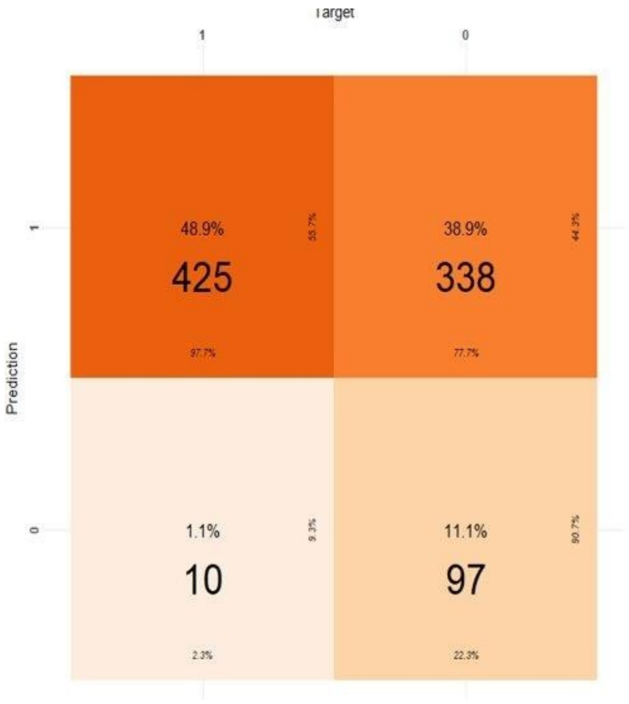


Figure 9. Confusion Matrix - Naïve Bayes

3) *Support Vector Machines*

It can also be used for Regression problems, hence called a support vector regressor. For this project we will use classification. The FLAG dependent variable will be represented in an n-dimensional space, with n corresponding to the number of included features. The next task is to find the right hyperplane so that we can label each point into 2 different classes of either valid or fraudulent. Project consists of using three Kernels: Radial Basis Function kernel, Linear Kernel and Poly kernel.

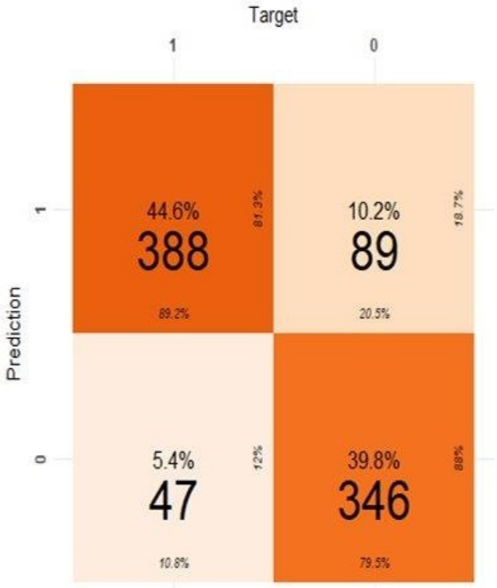


Figure 10. Confusion Matrix and Statistics

4) *Decision Tree*

In decision analysis, a Decision Tree is typically represented as a tree-like model. A binary tree is made up of the root, internal nodes and leaves. A Decision Tree is constructed based on entropy and information gain. At every node, we choose the feature with the maximum information gain among all features left for splitting. Internal nodes represent feature tests and leaf node suppression of labels Here we are going to discuss the one of the most powerful and widely used algorithm that is Decision Tree which can be applied for both classification as well regession problem, in this blog am covering its Classification capability.

- The performance of this model is excellent. It has generated an accuracy of approximately 93% which is the highest of all the models implemented so far.
- In addition to the accuracy, the precision and recall are also excellent with both greater than 91%. We can say this model has learnt its training data very well.

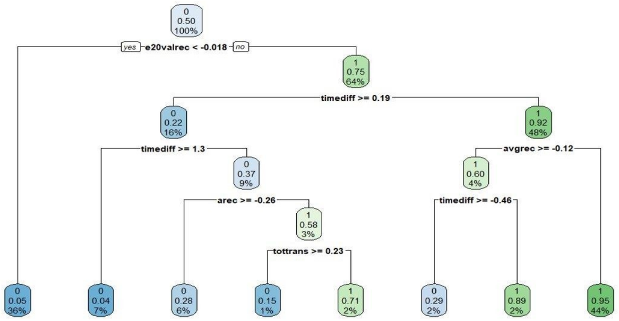


Figure 11. Decision Tree

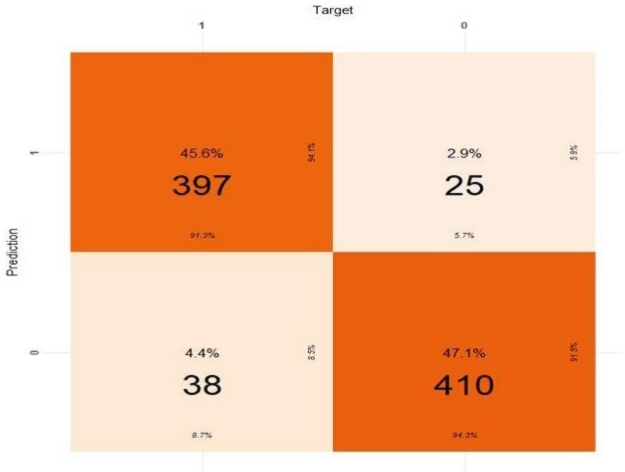


Figure 12. Confusion Matrix - Decision Tree

5) *Random Forest*

Random Forest is classification and regression algorithm. It is an ensemble method and it belongs to bagging methods. This method entails several decision trees working together to create one unified output. Every tree in the forest casts a vote (or some other measure of how much of an output that particular tree creates) to help make the prediction together. This structure lets every tree learn from the mistakes of others by working in unison like a single being.

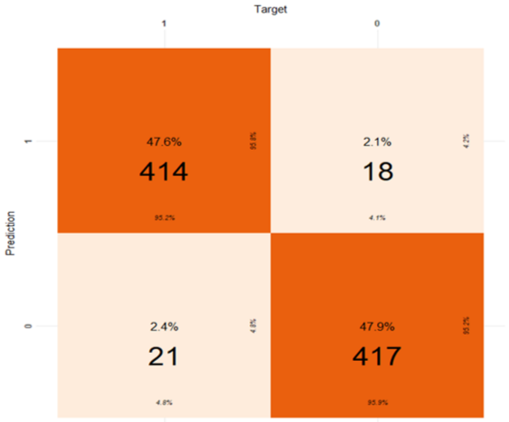


Figure 13. Confusion Matrix and Statistics

7. RESULT

TABLE I. Performance Metrics of Various Algorithms

Algorithm	Accuracy	Precision	Recall
Logistic Regression	71.95	57.70	80.71
Naïve Bayes	60.00	22.30	90.65
Decision Tree	92.76	91.52	94.25
Random Forest	95.52	95.51	95.86
SVM	84.37	88.04	79.54

8. CONCLUSION

The Random Forest algorithm reached the highest accuracy of 95.52%, as indicated by the table and graph provided above. The Naïve Bayes algorithm performed worst as a classifier with only 60% accuracy, could be due to the non-linear nature of this dataset. It can be expected that random forest would do better than the simple decision tree algorithm, since it is well-known to overfit. Random forest addresses the overfitting by constructing many treesAs mentioned earlier, Random Forest combats the problem of Over fitting while creating Trees using a technique which is known as Bagging. In this method, hundreds to thousands of trees are trained on different subsets of data. Finally, for predicting the label of a given example, each of the trees will spit out a label and the outcome chosen by most decision trees will be the final choice. This way all the trees are bagged together, correcting the mistakes of each other and hence acting as a strong classifier. This way the overfitting problem that is known to plague the decision tree is overcome, enabling the random forest model to perform better than the decision tree model. Thus, it can be concluded that Random Forest is the best performing model to fulfill our objective of detecting fraudulent transactions in Ethereum.

9. Bibliography

[1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [6], [16], [17], [18], [19], [20]

References

[1] E. Jung, M. Le Tilly, A. Gehani, and Y. Ge, "Data mining-based ethereum fraud detection," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 266–273.

[2] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the ethereum blockchain," *Expert Systems with Applications*, vol. 150, p. 113318, 2020.

[3] R. F. Ibrahim, A. M. Elian, and M. Ababneh, "Illicit account detection in the ethereum blockchain using machine learning," in *2021 International Conference on Information Technology (ICIT)*. IEEE, 2021, pp. 488–493.

[4] H. Baek, J. Oh, C. Y. Kim, and K. Lee, "A model for detecting cryptocurrency transactions with discernible purpose," in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2019, pp. 713–717.

[5] N. Kumar, A. Singh, A. Handa, and S. Shukla, "Detecting malicious accounts on the ethereum blockchain with supervised learning," in *Cyber Security Cryptography and Machine Learning*, ser. Lecture Notes in Computer Science, S. Dolev, V. Kolesnikov, S. Lodha, and G. Weiss, Eds. Springer, Cham, 2020, vol. 12161, pp. 112–126.

[6] M. Bhowmik, T. Sai Siri Chandana, and B. Rudra, "Comparative study of machine learning algorithms for fraud detection in blockchain," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2021, pp. 539–541.

[7] M. Ostapowicz and K. Żbikowski, "Detecting fraudulent accounts on blockchain: A supervised approach," 2019.

[8] R. Mittal and M. P. S. Bhatia, "Detection of suspicious or untrusted users in crypto-currency financial trading applications," *International Journal of Digital Crime and Forensics*, vol. 13, pp. 15–27, 2020.

[9] V. Patel, L. Pan, and S. Rajasegarar, "Graph deep learning based anomaly detection in ethereum blockchain network," in *Network and System Security*, ser. Lecture Notes in Computer Science, M. Kutyłowski, J. Zhang, and C. Chen, Eds. Springer, Cham, 2020, vol. 12570, pp. –.

[10] Z. Liu, H. Gao, H. Lei, Z. Liu, and C. Liu, "Blockchain anomaly transaction detection: An overview, challenges, and open issues," in *The 7th International Conference on Information Science, Communication and Computing. ISCC2023 2023*, ser. Smart Innovation, Systems and Technologies, X. Qiu, Y. Xiao, Z. Wu, Y. Zhang, Y. Tian, and B. Liu, Eds. Springer, Singapore, 2024, vol. 350, pp. –.

[11] I. Onu, A. Omolara, M. Alawida, O. Abiodun, and A. Alabdultif, "Detection of ponzi scheme on ethereum using machine learning algorithms," *Scientific Reports*, vol. 13, no. 1, p. 18403, 2023.

[12] R. Aziz, M. F. Baluch, S. Patel, and P. Kumar, "A machine learning based approach to detect the ethereum fraud transactions with limited attributes," *Karbala International Journal of Modern Science*, vol. 8, pp. 139–151, 2022.

[13] M. A. Ferrag and L. Maglaras, "Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Transactions on Engineering Management*, 2019.

[14] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine, "Credit card fraud detection in the era of disruptive technologies: A systematic review," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 1, pp. 145–174, 2023.

[15] M. Rabbani, Y. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, and P. Hu, "A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing," *Journal of Network and Computer Applications*, vol. 151, p. 102507, 2019.

[16] B. Kilic, A. Sen, and C. Ozturan, "Fraud detection in blockchains using machine learning," in *2022 6th International Conference on Big Data and Computing (BCCA)*, 2022, pp. 214–218.

[17] U. Sam, G. Moses, and T. Olajide, "Credit card fraud detection using machine learning algorithms," 2023.

[18] S. Siddamsetti, "Anomaly detection in blockchain using machine learning," *Journal of Electrical Systems*, vol. 20, pp. 619–634, 2024.

[19] P. Gupta, A. Varshney, M. Khan, R. Ahmed, M. Shuaib, and S. Alam, “Unbalanced credit card fraud detection data: A machine learning-oriented comparative study of balancing techniques,” in *Procedia Computer Science*, vol. 218, 2023, pp. 2575–2584.

[20] S. Taher, S. Ameen, and J. Ahmed, “Advanced fraud detection in blockchain transactions: An ensemble learning and explainable ai approach,” *Engineering, Technology & Applied Science Research*, vol. 14, pp. 12 822–12 830, 2024.