Fraudulent account recognition using supervised learning in Ethereum

A Project Report Submitted by

Prashant Kumar Choudhary

in fulfillment of the requirements for the award of the degree of

M.Tech. in Computer Science and Engineering

Indian Institute of Technology Jodhpur
Department of Computer Science and Engineering

July, 2021

Declaration

I hereby declare that the work presented in this Project Report titled "Fraudulent account recognition using supervised learning in Ethereum" submitted to the Indian Institute of Technology Jodhpur in partial fulfilment of the requirements for the award of the degree of M.Tech. in Computer Science and Engineering, is a bonafide record of the research work carried out under the supervision of Dr Debasis Das. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.

Signature
Prashant Kumar Choudhary
MT19CS008

Certificate

This is to certify that the Project Report titled Title of the Project Report, submitted by Prashant Kumar Choudhary(MT19CS008) to the Indian Institute of Technology Jodhpur for the award of the degree of M.Tech. in Computer Science and Engineering, is a bonafide record of the research work done by him under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Signature
Dr Debasis Das

Acknowledgements

I will like to show my gratitude towards my supervisor Dr. Debasis for his everlasting support and guidance. This project would not have been completed without his inputs and feedback. I will like to thank HOD of Computer Science for her extended encouragement and guidance. I take this opportunity to express a deep sense of gratitude towards all the professors and mentors who guided me throughout the project work. Last but not the least my classmates have been a great motivation for me especially during this tough pandemic situation.

Prashant Kumar Choudhary

IIT Jodhpur

MT19CS008

. . .

Abstract

Blockchain has gained significant popularity in the modern era. Almost all kinds of financial transactions are supported by the Blockchain platform. Ethereum is one of the most used Blockchain platforms. After Bitcoin, Ethereum is the contributor to the second-largest cryptocurrency. The number of transactions performed on Ethereum in a day exceeds 1 million. The security and ease of transactions make it ideal for all kinds of transactions. But despite all the security features provided by Ethereum, there is a significant quantity of illegal activities that are conducted on Ethereum. These illegal activities significantly harm the spread and usage of Ethreum by people and organizations. Thus, there is a need for a mechanism to detect the illegal activities on Ethereum Blockchain. The category of illegal activities is huge and the scope of this work is limited to the detection of illicit accounts on Ethereum using machine learning techniques. A novel convolution neural network architecture followed by an XGBoost classifier is proposed to segregate the accounts as illicit or normal based on transaction history. The XGBoost model is a tree-based ensemble classifier. XGBoost classifier has been used to improve the accuracy of the proposed model without overfitting the model too much. The implicit regularization supported by XGBoost helps the model to generalize well for the dataset. XGBoost provides another benefit in that the individual trees are parallelly created while training. This decreases the time required for training the model and makes it more scalable. The historical transactions of over 4000 Ethereum accounts are used as a dataset to train the model and perform prediction. The dataset is a balanced dataset as the normal accounts and fraudulent accounts both are in nearly equal proportion. The accuracy achieved is 98.39 % and an average AUC which is better than standard machine learning models.

Keywords: Ethereum, Blockchain, Fraud detection

Contents

List of Figures

List of Tables

Fraudulent account recognition using supervised learning in Ethereum

1 Introduction and background

This paper proposes a supervised approach to the detection of fraudulent accounts in Ethereum Blockchain. The proposed model is simple and effective in the sense that it uses a CNN as the base model. To further improve the performance of the proposed model, XGBoost is used for boosting. The proposed model is giving satisfactory results with an accuracy of 98.39% and an AUC of 0.998(std: 0.0008). The proposed method is thoroughly analyzed by plotting the confusion matrix and a comparison is done with other machine learning models like Logistic regression, SVM, Naive Bayes, Decision Tree, etc.

1.1 Problem Definition

The cryptocurrency platforms have gained popularity for performing secure and fast banking transactions. The transactions are not only limited to just sending or receiving funds over the blockchain platform but also smart contracts and trading of cryptocurrencies are very common. Ethereum is the second-largest cryptocurrency platform after Bitcoin with over 1 million transactions being performed daily. But like other cryptocurrency platforms, Ethereum too has its fair share of cyber frauds associated with it.

The transaction process in Blockchain is very robust and secure. Whenever a new transaction is submitted by one of the entities of the blockchain network, a consensus algorithm is run by all the entities of the Blockchain network. When the validity of the transaction is ascertained by the consensus algorithm, a block is added to the distributed ledger in encrypted form. Despite this internal security, a large number of scams, money laundering, Ponzi schemes, bribery, phishing are being performed on Ethereum. This is a serious issue since customers are getting targeted by scammers frequently. Certainly, a mechanism is needed to filter out the illegal accounts over Ethereum. But apart from centralized databases like Cryptoscamdb which records the illicit accounts, there is no concrete solution is provided by Ethereum to keep these activities in check. Currently, there is a record of over 6000 illicit accounts recorded on Cryptoscamdb. This large amount of account details can be beneficial for training machine learning models in supervised form. The resultant model can be run in the background to detect suspicious accounts online. So it is evident that there is a need for a mechanism to detect illegal activities on Ethereum.

1.2 Motivation

The motivation behind this work is summarised as below:

1. Ethereum being a large cryptocurrency and the Blockchain platform supports millions of transactions. The in-built security mechanism provides security and privacy to transactions. Still many attacks and illegal activities are reported on Ethereum Blockchain. The fraudulent accounts can be detected online if there is some mechanism to detect them online.

2. The transaction history of fraudulent accounts is available on the Cryptoscamdb platform. This can be used to create a dataset for training machine learning models to detect fraudulent accounts.

2 Literature survey

Inna and Valeriy(2018) et al. [1] has stated various factors which are acting as barriers to the use of blockchain in the financial sector. These barriers are authors' viewpoint and evidence are given in the form of various comments which are made by relevant technological leaders about the blockchain and its implementation. The major concerns have been stated as the architecture of pure Blockchain. Storing a complete chain on all devices makes it less feasible and complex. Alternative to Blockchain some of the variants are getting popular too which do not require storing the whole chain on all devices. Another issue is uncertainty. Adopting Blockchain would mean that the framework used earlier will not be used and migration of financial systems needs to be done. This requires both time and investment.

Sheetal, Kumkum, and Ruchika(2019) et al. [2] analyzed how blockchain can be incorporated in the financial sector. They have proposed a system containing a blockchain platform as a common distributed ledger that can be used for performing financial transactions. The proposed system claimed to be efficient in reducing delay, improving security, and decreasing processing overhead.

In 2019, Ostapowicz and Zbikowsk et al. [3] has worked on a supervised approach for detecting fraudulent accounts on Blockchain. The models such as random forest, SVM, and XGBoost are used to perform the detection of fraudulent accounts. The performance of these models is compared by using a Confusion matrix. Other parameters such as specificity, recall, precision, false-positive ratio(fpr), and F1 score are also recorded for each model.

In 2020, Arya Sastry et al.(2020) [4] performed classification of credit card fraudulent transactions. The dataset that is used contains around 2,84,000 transaction instances. Out of these only 492 fraudulent transaction instances exist in the dataset. Thus, the dataset is highly imbalanced. A novel model called 'DEAL' has been proposed to classify the transactions as genuine or fraudulent. The model consists of a CNN of 5 layers with 3 layers acting as dense layers and two layers acting as dropout layers. The deep layers are followed by ReLU activation functions. At the last layer, the sigmoid activation function has been used to get the classification labels and the Adam optimizer has been used to optimize the CNN. This CNN is taken as a base classifier and the AdaBoost classifier is used to enhance the accuracy of the proposed model. The feature importance as well as accuracy, AUC, etc. have been reported systematically to show that the model performs on par with existing mechanisms if not better.

To classify accounts as illicit and normal, Farruga et al(2020) [5] proposed a scheme that used a decision tree and XGBoost classifier. The dataset has been prepared as part of this work and consisted of over 4600 accounts. The importance of features w.r.t their weight has been reported with 'Time_diff_between_first_and_last(mins)' feature being the most important one. The model has been shown to give a very good accuracy of 96.3 % (std: 0.0006) with an average AUC of 0.994(std:0.0007).

3 Objectives

The objectives of this paper can be broadly summarised as below:

1. The Blockchain platform and especially cryptocurrency markets are very volatile. Any positive or negative feedback from some influential business leaders or organizations can simply affect the market growth of these platforms. The cases of illegal activities and malpractices can add to negative reviews from outsiders. Ethereum is no exception to this fact as well. Thus, it is highly essential to have mechanisms that can detect illegal activities on Ethereum. The primary goal of this paper is to detect illicit accounts on Ethereum Blockchain. To perform this task, previously reported illicit accounts' data hosted on Cryptoscamdb is used to train a machine learning model. The dataset contains features related to historical transactions of these Ethereum accounts as well as the historical transactions for normal accounts retrieved from test networks like MANET. The model trained on this dataset can identify the fraudulent accounts based on the transaction history of the account.

2. The proposed model should have high accuracy and scalability as this model can be deployed in an online manner to detect probable cases of fraudulent accounts. This could prove very helpful for government bodies and investigating agencies to find out traces of illegal activity being conducted on the Ethereum network.

3. The performance of the model should be compared with existing machine learning models and other proposed models related to fraud detection in Ethereum Blockchain. The metrics used for comparing performance should be generic and must be chosen carefully to ensure that the proposed model's ability to perform well in real use cases.

4. The common problem which machine learning models suffer from is overfitting. The proposed model should be designed carefully with the parameters of the model being selected after thorough testing and exploration.

4 Methodology

4.1 Dataset

The dataset is taken from the Farrugia et. al [5]. This dataset contains 4681 Ethereum accounts with their features. The dataset is not perfectly balanced with 2502 normal accounts and 2179 illicit accounts. The normal accounts have the feature 'FLAG' marked as whereas illicit accounts have 'FLAG' marked as 1.

Columns 'ERC20_most_sent_token_type' and 'ERC20_most_rec_token_type' have textual data. The textual data in these columns are mapped to numerical value as numerical data is required for training the proposed model. Two new features namely 'Ratio_min_rec_sent' and 'Ratio_time_min_rec_sent' have been added. 'Ratio_time_min_rec_sent' is defined as the ratio of columns of 'min_value_received' to 'min_value_sent'. 'Ratio_time_min_rec_sent' is defined as the ratio of columns 'Avg_min_between_received_tnx' and 'Avg_min_between_sent_tnx'. Infinity or NaN values for these two columns are filled with 0. For the rest of the columns NaN values are filled with mean of the column values.

4.2 Removing anomalous data

The dataset is filtered for anomalies. The anomalies are data points that show a great deviation from other data points. An isolation forest mechanism has been used to detect the anomalies. The account data containing 49 features have been passed as input to the isolation forest. The isolation forest tries to come up with a predictor to separate the data points. In this process of separation, the anomalies are separated further from normal data. Finally, The dataset is divided into two classes with one being normal and the other being anomalies. The classification label for anomalous data is -1 and the classification level is 0. 194 account data are marked as anomalous and the data is removed from the dataset. The contamination factor is defined as the percentage of outliers in the dataset. The contamination factor is 4.14%. The remaining 4487 accounts are used for training and prediction.

4.3 Visualization of dataset

The t-SNE visualization scheme has been used to visualize the refined dataset in two dimensions and three dimensions. As the features are large, PCA is used to convert the features to 2D and 3D before the t-SNE process. The plots are shown below.

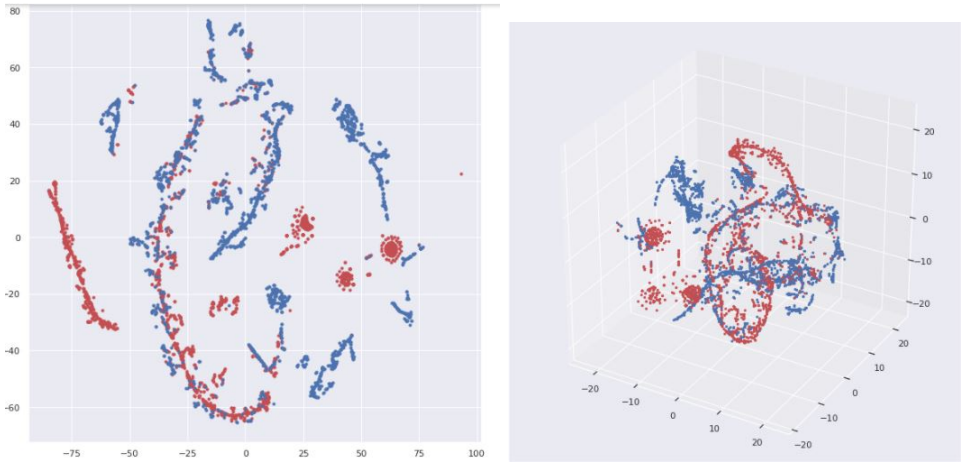

Figure 4.1: 2D plot of dataset (blue dots represent normal accounts and red dots represent illicit accounts)

The plots show that there the features of normal and illicit accounts are dispersed and they do not form two distinct clusters.

4.4 XGBoost Classifier

XGBoost is a tree ensemble-based boosting algorithm and it was developed by the University of Washington in 2016 [6]. XGBoost considers an ensemble of classifiers that are in the form of decision trees. Boosting in the case of XGBoost employs gradient-based optimization to minimize the errors of classifiers. The optimization is done only for classifiers that are committing errors and are weak. After the optimization is done and the weak classifiers are turned into strong classifiers, bagging is performed and the classification label is generated by taking a vote from all the classifiers.

XGBoost not only improves the accuracy of the model but also is fast as it performs building tree ensembles in a parallel manner. The algorithm is good for hardware optimization. Another benefit of this algorithm is that it prunes the decision trees to save computations and improve performance. It also supports regularization and cross-validation which are enhanced features for algorithms.

4.5 Model architecture

The proposed model consists of a CNN with 3 hidden dense layers and 1 hidden dropout layer. The input layer is dense with 16 filters. The hidden dense layers also have 16 filters. Each dense layer is followed by the ReLU activation function. The output layer is a dense layer with one filter. The sigmoid activation function follows the last dense layer. The optimizer used is Adam and the learning rate is 0.001. The dense layers are used to perform convolutions and transform the features. The dropout layer is added to decrease the need for computations required without affecting much accuracy of the classifier. This CNN classifier is used as a base classifier for XGBoost and the output is generated by XGBoost.

The complete hyper-parameters used in case of CNN classifier is shown in table below:

Table 4.1: Hyper-parameters used for benchmark and proposed classifier

Framework architecture	Hyper-parameters(proposed)
Input	49 features
Dense layer1	16 filters;RELU
Dense layer2	16 filters;RELU
Dense layer3	16 filters;RELU
Dropout layer1	dropout 50 %
Dense layer4	16 filters;RELU
Dense layer5	Nodes 1; Sigmoid
Loss function	Binary Cross Entropy
Optimizer	ADAM
Epocs	100

The proposed model architecture is shown below.

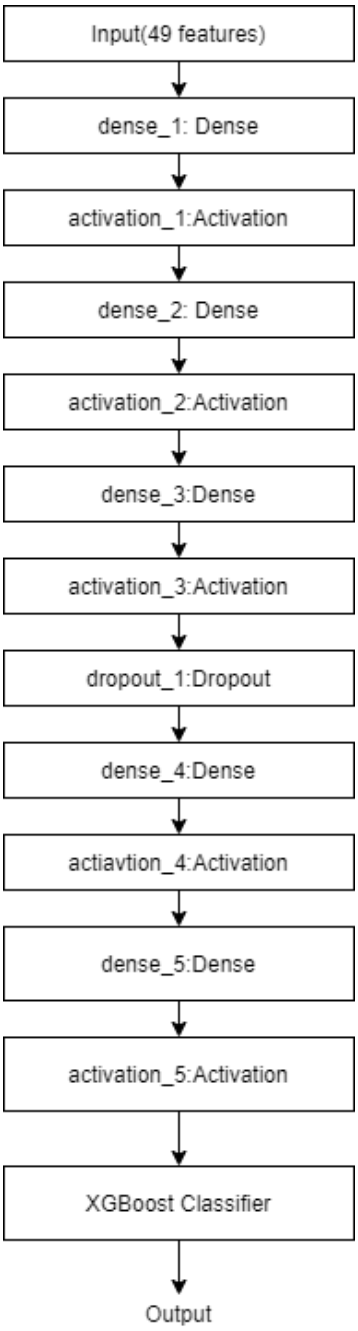

Figure 4.2: Proposed model architecture

5 Result and analysis

5.1 Environment, tools, and libraries

The implementation of the proposed model is done by using python. The libraries and systems used for implementation are shown in the table below:

Table 5.1: Environment, tools, and libraries for training and testing

Tools/Libraries/System	Version/Configuration
Processor	Intel(R) Xeon(R) CPU @ 2.30GHz
RAM	12.72 GB
Tensorflow	2.4.0
Numpy	Nodes 1.19.5
Pandas	1.1.5
Keras	2.4.3
Scikit-learn	0.22.2.post1
Seaborn	0.11.1
Matplotlib	3.2.2
Bioinfokit	1.0.5
Python	3.6.9
Platform	Google Collaboratory

5.2 Confusion matrix

The dataset is divided into 50-50 partition, 70-30 partition, 80-20 partition and 90-10 partition. The confusion matrix is plotted for each of these cases.

1. **True Positive(TP):** This corresponds to accounts that are normal and have been classified as normal by model.
2. **False Positive(FP):** This corresponds to accounts that are illicit and have been classified as normal by model.
3. **True Negative(TN):** This corresponds to accounts that are illicit and have been classified as illicit by model.
4. **False Negative(FN):** This corresponds to accounts that are normal and have been classified as illicit by model.

The confusion matrix for all these four partitions are shown below.

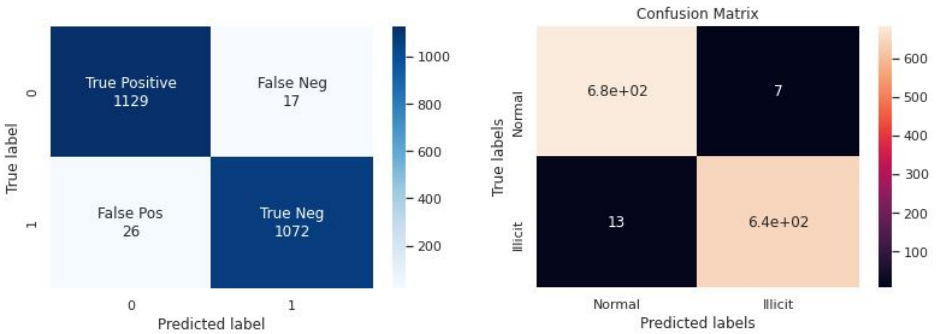

Figure 5.1: Confusion matrix for 50-50 and 70-30 split respectively

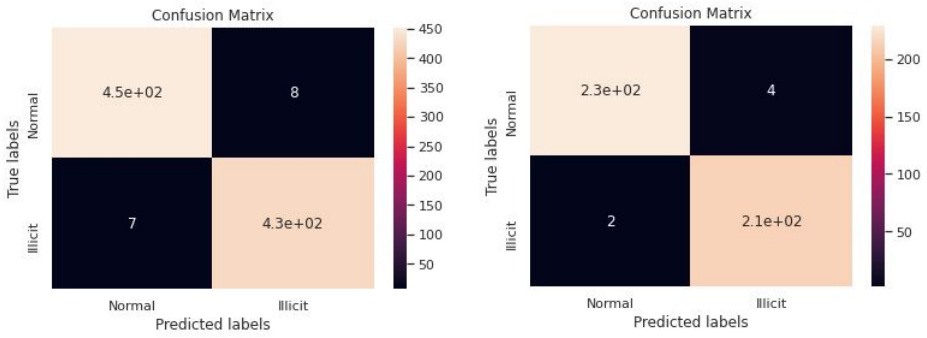

Figure 5.2: Confusion matrix for 80-20 and 90-10 split respectively

5.3 Accuracy, AUC and other parameters

1. **Accuracy**: It is defined as the percentage of test data samples that are correctly classified.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

2. **Precision**: It s defined as ratio of true positives and total number of test samples that are classified as positives.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

3. **Recall**: It is defined as ratio of true positives and sum of true positives and false negatives.

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

4. **AUC**: This is an approximation to the area under the curve for the precision-recall curve.

5. **F1 score**: This corresponds to the harmonic mean of the precision and recall.

6. **False Positive Ratio(FPR)**: This ratio gives us an idea of what fraction of negative class got incorrectly classified.

$$\text{FPR} = \text{FP}/(\text{TN}+\text{FP})$$

7. **True Positive Ratio(TPR)**: This ratio gives us an idea of what fraction of positive class was classified properly.

$$\text{TPR} = \text{TP}/(\text{TP}+\text{FN})$$

5.4 Stratified K-fold Validation

To remove the demerits of random sampling, 10-fold cross-validation is performed on the complete dataset. In this scheme, the dataset is divided into 10 parts and individually these parts are trained and tested based on the sample drawn in a stratified manner. The accuracy reported is the combination of accuracies achieved in individual partitions of data. The result is shown below:

Table 5.2: 10-fold cross validation results for proposed model

Optimal Number of estimators	learning rate	Accuracy	AUC
450	0.3	98.39	0.998(std:0.0008)

While performing boosting, various combinations of the number of estimators used and the corresponding AUC value is plotted. The plot shows that the optimal is 0.9990 for the number of estimators as 400 and depth of 4. The plot is shown below:

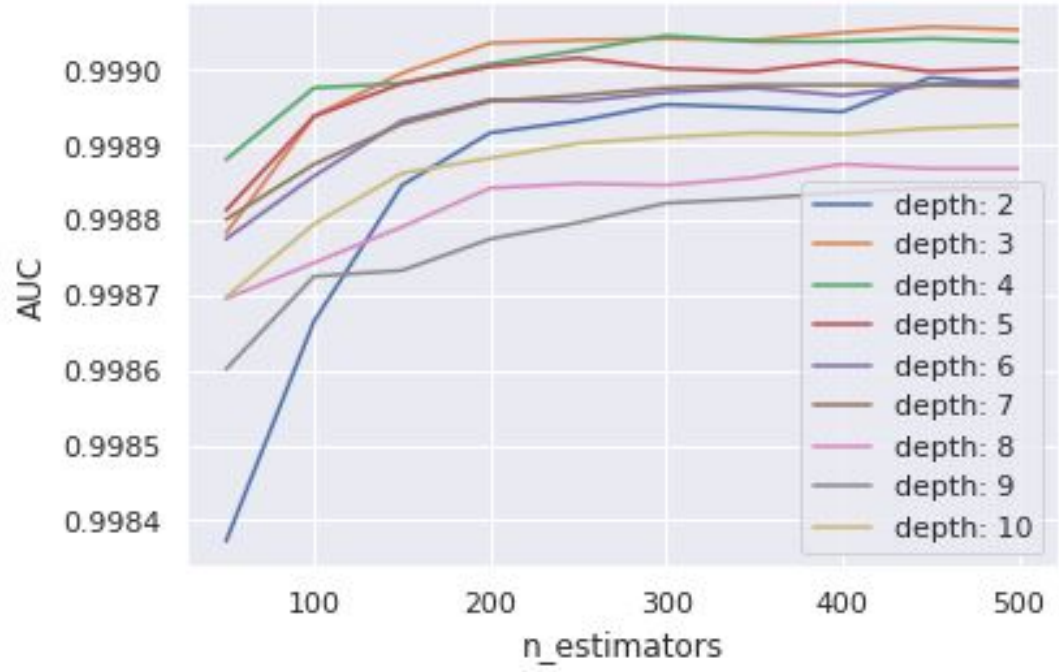

Figure 5.3: Number of estimators versus AUC for 10-fold cross validation

5.5 Performance metrics for various splits of dataset

The dataset is split into 50-50, 70-30, 80-20, and 90-10 fractions respectively. For each split, performance metrics such as the Optimal number of estimators, learning rate, accuracy, and other parameters are recorded. The performance matrix for various splits is shown in the table below.

Table 5.3: Performance matrix for various splits of dataset

Metric	Split 50-50	Split 70-30	Split 80-20	Split 90-10
Optimal number of estimators	50	100	250	50
Learning Rate	0.3	0.3	0.1	0.1
Optimal Depth	10	2	8	7
Accuracy	98.08	98.51	98.32	98.66
Precision	0.9774	0.981	0.984	0.991
Recall	0.985	0.989	0.982	0.982
F1-score	09803	0.9847	0.9828	0.9861

5.6 Comparison of the proposed model with standard machine learning models

The proposed model is compared with six other machine learning models which are commonly used for binary classification. 70-30 split of the dataset is chosen for training and testing the models. The comparison is done based on accuracy, AUC score, and F1 score. The comparison result is shown in the table below:

Table 5.4: Comparison of various models with 70-30 dataset split

Model	Accuracy	AUC	F1 score
Logistic Regression	70.52	0.6984	0.5816
KNN	89.01	0.8900	0.8873
Decision Tree	96.28	0.9630	0.9621
SVM	68.67	0.6788	0.5290
Naive Bayes	68.67	0.6789	0.5290
Stacking	96.36	0.9637	0.9629
Proposed model	98.51	0.9805	0.9801

Stacking or Stacked Generalization used here is an ensemble machine learning algorithm. In this model, Logistic Regression, KNN, Decision tree, SVM, and Naive Bayes all are taken as base models. In level 1, Logistic Regression is used to learn from the base models and come up with a strong classifier.

To compare further, the AUC-ROC curve is plotted for each model. This curve plots TPR against FPR at various thresholds. As the area under the curve(AUC) gets bigger, the model performs better classification of positive and negative classes. The AUC-ROC curve is shown below:

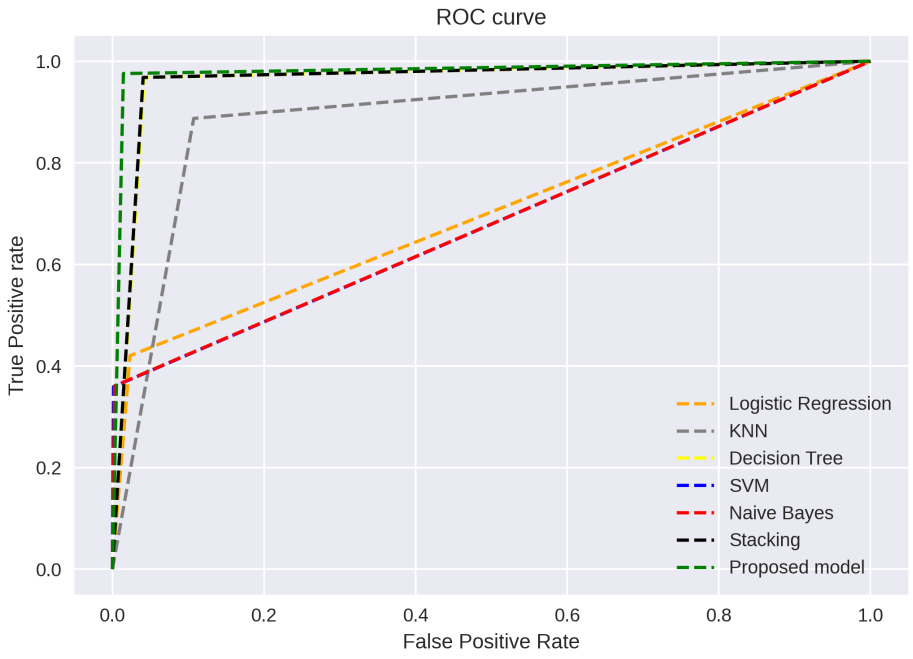

Figure 5.4: ROC-AUC curve for various models

It is quite evident that the proposed model exceeds the performance of these standard machine learning models.

6 Conclusion and future work

The proposed model is shown to be effective in classifying illicit accounts and normal accounts from the dataset that contains Ethreum account features. The proposed model is giving satisfactory results with an accuracy of 98.39% and an AUC of 0.998(std: 0.0008). This is on par with some of the models proposed earlier for a similar task. The proposed model is shown to be performing better than models like Logistic regression, KNN, Decision Tree, Naive Bayes, and Stacking.

In the future, this model can be taken as a base for other models to do similar classification. The concept of fraud detection is a widely researched area and similar analysis can be performed for other Blockchain platforms like Bitcoin, Litecoin, etc. The approach suggested above is supervised in nature, but it may require updating the model periodically as newer techniques are applied by scammers. A possible solution to this problem is the application of a reinforcement learning-based model for fraud detection.

References

[1] I. A. Kruglova and V. A. Dolbezhkin, "Objective barriers to the implementation of blockchain technology in the financial sector," in *2018 International Conference on Artificial Intelligence Applications and Innovations (IC-AIAI)*, 2018, pp. 47–50.

[2] S. Sinha, Kumkum, and R. Bathla, "Implementation of blockchain in financial sector to improve scalability," in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 2019, pp. 144–148.

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

[4] M. Arya and H. Sastry, "Deal – 'deep ensemble algorithm' framework for credit card fraud detection in real-time data stream with google tensorflow," *Smart Science*, vol. 8, pp. 71–83, 04 2020.

[5] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the ethereum blockchain," *Expert Systems with Applications*, p. 113318, 07 2020.

[6] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016. [Online]. Available: http://arxiv.org/abs/1603.02754