Journal of Global Research in Electronics and Communication
Volume 1, No. 7, July 2025
Available Online at: www.jgrec.info

RESEARCH PAPER

Transaction Analysis of Categorizing Ethereum Addresses Based on Advanced Supervised Machine Learning Approach for Predictive Modeling

Raja Lodhi
Mtech Scholar
Department of Computer Science and Engineering,
LNCT Group of Colleges
Bhopal, M.P., India
ndrajalodi@gmail.com

Raj Kumar Sharma
Assistant Professor
Department of Computer Science and Engineering,
LNCT Group of Colleges
Bhopal, M.P., India
Rajkumar.s@lnct.ac.in

***Abstract*—Ethereum has become one of the most significant cryptocurrencies in terms of transaction volume. Given Ethereum's recent rise, experts and the cryptocurrency community are eager to learn more about how Ethereum transactions behave. A machine learning system-based methodology exists for addressing Ethereum addresses to enable transaction classification. The preprocessing of CEAT dataset containing 4,371 entries with 15 features utilizes a systematic process that selects relevant features then handles missing data along with using SMOTE for class distribution balancing and converting categorical elements to numbers. The data is standardized with MinMax Scaler to improve model performance. Exploratory data analysis is done by visualizations such as Heat map, Histogram and Pair plot to compute the features which are correlated. The four models of machine learning algorithms include Decision Tree (DT), LightGBM, KNeighbors Classifier (KNN), and CatBoost Classifier, are trained with the best optimized hyperparameters. The classification reports, confusion matrices, and ROC curves are used for model evaluation for each model. Comparing these models, LightGBM has the highest accuracy of 91.99%, second is CatBoost Classifier with 91.23%, the Decision Tree is 82.43%, and the KNN model 78.91%. An important benefit from this study is that the results show that it is possible to use a machine learning approach for classifying Ethereum addresses to enhance transaction security and avoid fraud in decentralized financial systems.**

Keywords—Ethereum addresses, Machine Learning, Decision tree, K-neighbors, cat Boost and LightGBM.

I. Introduction

Ethereum stands as a well-known cryptocurrency. According to the quantity of recorded financial transactions, it has grown to be among the biggest cryptocurrencies at the moment [1]. Since its launch in February 2020, the network has processed over 470M transactions, or 9 per second, and the resulting market capitalization is over $27 billion USD. Furthermore, Ethereum's primary benefit over Bitcoin, the original cryptocurrency, has been thought to be its ability to handle programmed contracts, or smart contracts, which it adds to its financial transactions [2][3][4][5].

In a cryptocurrency system, a distributed consensus process can determine if a transaction was successful or unsuccessful after it was executed [6][7]. In this case, confirmation of transactions is vital because customers risk losing their funds if the network operators' or miners' fees for processing the transactions are not refunded[8][9]. Furthermore, prior to a transaction entering the network for processing, no failure risks are revealed. As a result, failures might negatively impact users' experience and discourage them from making future transactions [10][11]. In contrast, the distributed consensus process required to execute and verify each Ethereum transaction is complicated, and the tiny proportion of failed transactions, when taking into account millions of recorded transactions, makes it difficult to build models to forecast confirmation [12][13].

Ethereum addresses receive primary classification through transaction analysis methods that provide understanding about user behaviors and their transaction patterns [14]. Researchers utilize machine learning methods to parse meaningful data patterns from the Ethereum address transaction records for classification between individual users, exchanges, miners and malicious actors [15]. Finding any transactions that were thought to have unusual characteristics would require a laborious and time-consuming manual search of all of these transactions. An use of ML methods to aid in an identification of patterns associated with suspicious behavior is recommended by the fast development of such network blocks, especially smart contracts and transactions [16]. Building prediction models under supervised learning branches of ML requires training on extremely large datasets containing labeled samples with their actual outputs noted. Therefore, its usage of labeled datasets for model training is the primary differentiator from other ML types [17][18].

A. Motivation and Contribution

A motivation for this research stems by a growing complexity and security challenges in Ethereum transactions, where traditional fraud detection methods fall short against evolving threats. With the increasing adoption of blockchain technology, there is a pressing need for automated, scalable, and high-accuracy classification models to distinguish between legitimate users and malicious actors. By leveraging advanced supervised machine learning, this study aims to enhance blockchain security, improve fraud detection, and contribute to more robust financial forensics in decentralized ecosystems. Here are key research contributions from this study on categorizing Ethereum addresses using advanced supervised machine learning for predictive modeling:

- The research presents a strong supervised learning framework for Ethereum address classification using transactional patterns, which enhances the capacity to distinguish between real and fraudulent operations.

- A ML evaluation of Decision Tree, LightGBM, K Neighbors Classifier, and CatBoost Classifier is to identify the most effective model.
- The implementation of SMOTE effectively addresses class imbalance, ensuring fairer training and better generalization across different Ethereum address categories.
- The study demonstrates the impact of Min-Max Scaler for feature normalization, which enhances model performance and stability by reducing variance in Ethereum transaction data.
- The study systematically evaluates model performance using ROC curves, confusion matrices, and classification reports, offering a structured approach to selecting the best-performing classifier.

B. *Novelty and Justification*

The novelty of this research lies in its comprehensive integration of advanced supervised machine learning models to categorize Ethereum addresses with high accuracy, leveraging a structured pipeline that includes SMOTE-based data balancing, Min-Max Scaler normalization, and detailed exploratory data analysis (EDA). Unlike traditional blockchain classification methods, this study systematically evaluates multiple models Decision Tree, LightGBM, KNeighbors Classifier, and CatBoost Classifier using optimized hyperparameters, ROC curves, and confusion matrices for a robust comparison. The justification for this approach stems from the increasing complexity of Ethereum transactions and the need for scalable, automated, and high-precision classification techniques to enhance fraud detection, security, and regulatory compliance in decentralized financial systems.

C. *Structure of the paper*

The study is organized as follows: The current body of research on Ethereum categorization is examined in Section II. The approach used to gather the data for this investigation is described in section III. The findings and analysis of the text categorization are presented in Section IV. Finally, the conclusion is presented in Section V.

II. Literature Review

This section discusses the literature review on transaction analysis of categorizing Ethereum addresses based on advanced machine learning approach for predictive modeling. Table I also includes the abstracts of the following research reviews:

Bani-Hani, Shatnawi and Al-Yahya (2024) using deep learning methods, Ethereum transaction vulnerabilities may be categorized and detected. These transactions are transformed into RGB and greyscale pictures, which are subsequently analyzed by the binary and multi-label classification algorithms ResNet50, DenseNet201, VGG19, KNN, and RF. The best method for binary classification was RF, which had an accuracy rate of 86.66% and a score of 86.66% [19].

Yan and Kompalli (2023) determines whether a certain set of transactions follows the same course of execution on the current blockchain state. Their analysis of more than 1.3 billion Ethereum transactions successfully uncovers suspicious behaviors linked to an accuracy of 83.8 percent [20].

Saleem et al. (2023) have used the publicly accessible Ethereum blockchain's tagged dataset consisting of 300 million transactions. Using eleven feature vectors and 200 window widths, the XGBoost classifier achieved the greatest attainable accuracy of 73% in predicting the function of an unknown address, according to the test data. CNN model that was trained and evaluated on the dataset achieved an accuracy rate of 86% when predicting labels [21].

Aziz et al. (2023) proposed methods that were evaluated in relation to the performance and efficiency measures of other well-known methods for identifying fraudulent activity on Ethereum. These methods included KNN, LR, MLP, XGBoost, LGBM, RF, and SVC. With the maximum accuracy, the recommended approach and SVC models surpass all other models. When used in conjunction with the suggested optimization approach, DL achieves a performance of 91%, which is somewhat better than the RF model [22].

Pragasam et al. (2023) proves that the RF, GB, and XGBoost classifiers for address category prediction were trained and evaluated using a dataset of 4371 values. The XGBoost classifier outperformed all other models in this problem set, with a macro-averaged F1Score of 0.689 and an accuracy of 75.3%. The Random Forest classifier came in second, with a macro-averaged F1Score of 0.641 and an accuracy of 73.7%. With gradient boosting, the accuracy rate was 73% [23].

Dritsas and Trigka (2023) conducted experiments with various supervised ML models to identify early-stage symptoms of SARS-CoV-2 infection. The results demonstrated that the Stacking ensemble model achieved the best results, with accuracy, precision, re-call, and F-measure of 90.9% [24].

TABLE I. Summary of Literature Review Transaction Analysis of Ethereum Based on Machine Learning Approaches

Author	Dataset	Methods	Key Findings	Accuracy	Limitation/Gap
Bani-Hani, Shatnawi, and Al-Yahya (2024)	Ethereum transactions converted to RGB and Grayscale images	ResNet50, VGG19, DenseNet201, KNN, RF	RF achieved the best binary classification performance for RGB images	86.66%	Limited exploration of multi-label classification; potential improvements with ensemble techniques
Ivanov, Yan, and Kompalli (2023)	1.3 billion Ethereum transactions	Analysis of transaction Sequences to detect TOCTOU vulnerabilities	Identified suspicious behaviors and discrepancies in transaction sequences	83.8%	Focused solely on TOCTOU issues; broader vulnerabilities were not explored
Saleem et al. (2023)	300 million labeled Ethereum transactions	XGBoost, CNN	XGBoost achieved 73% accuracy; CNN outperformed with 86% accuracy	86% (CNN)	Limited feature set and window sizes; additional features and optimization may enhance accuracy
Pragasam et al. (2023)	Ethereum address profiles dataset (4371 samples) from Google BigQuery	RF, GB, XGBoost	XGBoost was the best-performing model with 75.3% accuracy and macro- averaged F1-score of 0.689	75.3%	Small dataset size and limited feature engineering; potential improvements with larger datasets and deep learning

Dritsas and Trigka (2023)	Dataset for early-stage SARS-CoV-2 symptoms	Supervised ML models (Stacking ensemble)	Stacking ensemble model achieved the best performance across accuracy, precision, recall, and F-measure	90.9%	Different domain from Ethereum; findings may not directly translate to blockchain-based vulnerabilities

III. Methodology

Figure 1 demonstrates the process flow of the proposed technique for classifying Ethereum addresses using supervised ML. It starts with data preparation and continues with visualization, model training, and performance assessment. The CEAT dataset, containing 4,371 entries and 15 features, undergoes cleaning by removing unnecessary columns, handling missing values, encoding categorical variables, and balancing imbalanced data using SMOTE. Min-Max Scaler standardizes the features to improve model performance. Exploratory Data Analysis (EDA) includes heatmaps, histograms, and pair plots to understand feature correlations. The dataset is split into an 80-20 train-test ratio, and four ML models, DT, LightGBM, KNeighbors Classifier, and CatBoost Classifier are implemented with optimized hyperparameters. Each model is evaluated based on classification reports, confusion matrices, and ROC curves to assess predictive accuracy. Finally, model comparison is performed to determine the best-performing algorithm for Ethereum address classification.

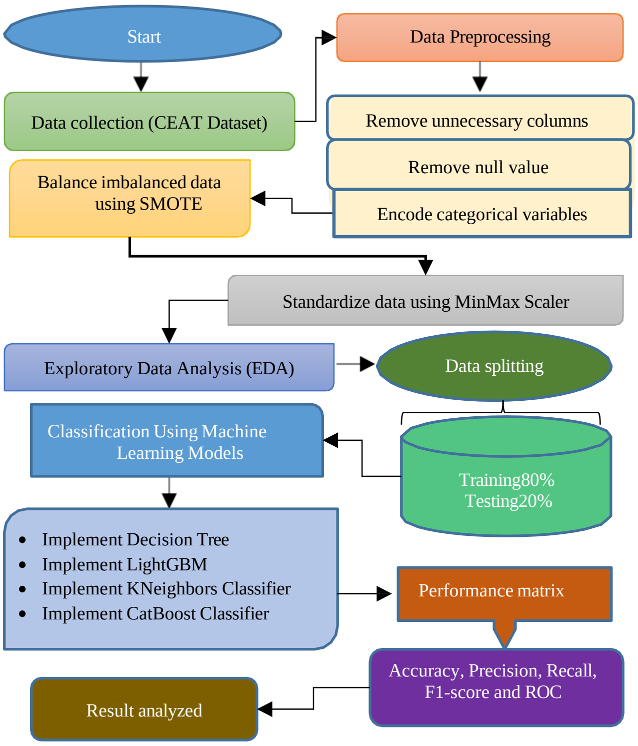

Start
Data Preprocessing
Data collection (CEAT Dataset)
Remove unnecessary columns
Remove null value
Balance imbalanced data using SMOTE
Encode categorical variables
Standardize data using MinMax Scaler
Exploratory Data Analysis (EDA)
Data splitting
Classification Using Machine Learning Models
Training80%
Testing20%
Implement Decision Tree
Implement LightGBM
Implement KNeighbors Classifier
Implement CatBoost Classifier
Performance matrix
Accuracy, Precision, Recall, F1-score and ROC
Result analyzed

Fig. 1. Proposed Flowchart Ethereum Transaction classification using machine learning techniques

A whole process of proposed methodology shows in Figure 1. also, each and every step is discussed below:

A. Data Collection

The "CEAT" project repository, created on June 30, 2023, appears to focus on data processing phase, contain 4371 rows of data and analysis. The structure suggests a workflow for preparing data, extracting features, and optimizing models, likely for machine learning or statistical analysis purposes.

B. Data Preprocessing

The primary first stage in data analysis is data preparation. It enables to convert unstructured data into a form that can be analyzed more effectively [25]. Data Preprocessing for Ethereum Classification on CEAT data:

- **Data Cleaning:** The dataset, consisting of 4,371 entries and 15 features, undergoes a cleaning process to ensure the removal of unnecessary or irrelevant columns [26]. The necessary features are identified for retention during this step to optimize the dataset for model training purposes.
- **Handling Missing Values:** The dataset is filled in with the missing values using the relevant procedures. It can involve operations like imputing the missing observations or altogether dropping the rows with missing data if the dataset needed to be cleansed before the machine learning techniques were applied on them.
- **Encoding Categorical Variables:** This transformative process refers to the conversion of categorical variables available within the data set into other types of data using the process of encoding. This step helps to avoid the ambiguity of categorical data, for example, regarding the type of transaction or the Ethereum address for the model input.

C. Data Normalization Using Min-Max Scaler

The dataset receives standardization treatment through normalization procedures. An essential part of data preparation, data scaling [27] seeks to standardize and make comparable all numerical properties [28]. A popular approach for this is Min-Max Scaler. For the purpose of normalization, the Min-Max Scaler method was used. Equation (1) shows the formula that was used to normalize the data

$$X' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x stands for the pre-normalization value, and x' refers to the value that remains after normalization; one may find the maximum and lowest values of the sample data by looking at the variables x max and x min, respectively.

D. Balancing an Imbalanced Dataset

The easiest way to increase the size of the minority class, while it may lead to overfitting, is to randomly increase the sample size. KNN is used in SMOTE (synthetic minority oversampling) to add duplicate instances to the training set in order to lessen the likelihood of overfitting [29]. Specifically, SMOTE makes use of Equation (2).

$$x_{syn} = x_i + (k_{nn} - x_i) \quad (2)$$

where t is a random integer between 0 and 1, and xi is a feature vector, which is known for the KNN.

E. Data Splitting

Data splitting in machine learning models is an essential step for evaluating the performance of model's dataset into 80% - 20% ratio, that is, 80% of the data they use for training purposes and 20% of the data use for testing purposes.

F. Classification of ML Models with Hyperparameter Tuning

In this section provide the classification machine learning models (Decision Tree (DT), KNeighbors Classifier, LightGBM (LGBM and CatBoost) for the Ethereum Transaction classification on CEAT data.

1) K-nearest neighbor (KNN) classifier

Regression and classification are two applications of the ML technique KNN. Using the labels or values of the KNN linked to the new data point in the training set, one may predict the new data point's label or value [30]. Simply said, KNN saves the complete training dataset, therefore training time is unnecessary. The "k" in k-nearest neighbors indicates that the method may consider any specified number of neighbors in the training set, rather than only the neighbor to the close-by data point. kNN is one of the algorithms that make up the instance-based learning family. Data is pushed to a neighboring class with a most immediate proximity. As an expansion, a quantity of closest neighbors, the value of k (with a common choice being n_neighbors=2), precision may increment [31]. KNN Euclidean, Manhattan and Minkowski calculate as Equations (3,4 and 5):

KNN Euclidean distance formula:

$$\sqrt{\sum_{i=0}^{k}(x_i - y_i)^2} \tag{3}$$

KNN Manhattan distance formula:

$$\sqrt{\sum_{i=1}^{k}|x_i - y_i|} \tag{4}$$

KNN Minkowski distance formula:

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}} \tag{5}$$

2) CatBoost classifier

When it comes to category data, the CatBoost classifier shines. It's a gradient-boosting method that use binary decision trees for prediction generation. It can accommodate both binary and numerical response variables. For the CatBoost Classifier model, it founds a happy medium between model complexity and training duration with 1000 iterations, 0.1 learning rate, and 6 depth. The loss function is set to 'Multiclass', suitable for multi- class classification tasks, while the evaluation metric is 'Accuracy', ensuring the model's performance is measured by its correct classification rate. The regularization parameter, l2_leaf_reg, is set to 3 to reduce overfitting by penalizing large leaf values. Lastly, to make sure the findings are reproducible between runs, the random state is set to 42. These settings were chosen to optimize model training, balancing performance and generalization.

3) Light Gradient Boosting Method (LGBM)

Light Gradient Boosting (or "Light GBM") is a tree-based gradient improvement approach that is both quick and effective [32]. The word "light" is used since the classifier employs a tree-based technique with vertical tree growth. it outperforms techniques based on horizontal trees in terms of efficiency [33]. Large dataset processing benefits from the time and resource efficiency of the Light gradient boosting technique. Light GBM is different from other techniques in that it grows tree leaf-wise, or vertically, as opposed to horizontally, such as most other methods do. The leaf with the greatest delta loss will be chosen for agricultural use.

Compared to a level-based technique, a leaf-wise approach to growing the same leaf may reduce waste more effectively [34]. The LightGBM model was configured with the following parameters: num_leaves set to 50, which aids in managing model complexity and overfitting by controlling the amount of leaves in each tree. A learning rate of 0.3 is used to balance the speed of learning and the potential for overfitting, ensuring faster convergence. The max_depth parameter is set to -1, allowing the model to grow trees without a predefined depth limit, promoting better fitting to the data. With 1000 estimators, the model utilizes a sufficient number of trees to improve predictive accuracy. These adjustments are made with the main objective of maintaining high training efficiency as well as enhancing the model’s ability for complex input patterns.

4) Decision Tree

Data can be divided into sub-sets based on feature values with a DT which is a type of supervised ML model implemented by a tree that utilizes nodes to account for features as well as leaves for the outcomes. It is used in classification and regression techniques for the purpose of creating the best split which maximizes information gain or minimizes variance. DT are easyto interpret and can accommodate both numerical and categorical data and its main disadvantage is overfitting when the trees are deep; this can be resolved by pruning or even aggregation techniques. The Decision Tree model was generated with the following settings of the Decision Tree Algorithm: criterion=log_loss, when making the decision on the best split. CNN splitting with splitter = ‘best’ guarantees that the chosen split at each node is the best possible with an outlook towards perfect decision boundaries. The min_samples_split=2 enables the splitting of a node as early as there are two samples which helps to capture more detailed patterns in the data. Finally, the max_depth=10 constrains the decision trees up to the depth of 10 in order to avoid overfitting but at the same model to have sufficient level of complexity. These settings were chosen because they should provide high accuracy on the problem while being as general as possible.

G. Performance Measures

To evaluate each model's effectiveness, four different performance criteria have been used: F1-score, precision, recall, and accuracy [35]. A confusion matrix is one of the most well-known academic performance measures used to analyze the outcomes. The matrix displays the outcome data using four main qualities; this data is the total of the outcomes from classifications. A result is considered true positive (TP) if the actual value of the classification equals the expected value. Similar in nature, true negative (TN) principles are centered on zero. False positives (FPs) occur when the predicted value is 1 but the actual value is 0, and false negatives (FNs) occur when the inverse is true.

1) Accuracy (Acc)

Accuracy is the ratio of correctly classified cases to the total error in class prediction. The accuracy Equation is (6):

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{6}$$

2) Precision

that it grows trees leaf-wise, or vertically, as opposed to horizontally, such as most other methods do. The leaf with the greatest delta loss will be chosen for agricultural use. Precision refers to the degree of accuracy in assigning instances to the correct class. Precision is formulated in Equation (7):

$$Precision = \frac{TP}{(TP+FR)} \tag{7}$$

3) *Recall*

This context calculates the percentage of all bearers of the ailment that the classifier correctly represents. Recall is formulated in Equation (8).

$$Recall = \frac{TP}{(TP+FN)} \tag{8}$$

4) *F1-score*

When recall and precision are weighted harmonically, the result is the F1-score, also called the F-measure Equation (9):

$$F1 - \text{score} = \frac{2 \times recall \times precision}{recall + precision} \tag{9}$$

5) *ROC Curve*

The ROC curve and AUC score are shown to evaluate the model's class-differentiation capabilities. The accuracy and efficacy of the model in forecasting the CEAT variable are shown by these indicators taken together.

IV. Result Analysis and Discussion

The experimental results for Ethereum Transaction classification using ML techniques on the CEAT dataset model are shown in this part. In addition to performing metrics such as recall, accuracy, precision, and F1-score, the classification report and ROC confusion matrix are also examined. A hardware platform was prepared to handle the computational demands of the proposed models by installing an NVIDIA GTX 1660i GPU with 16 GB of RAM and 8 GB of VRAM. This platform included the Python programming language, Jupyter Notebook, Google Colab, and Python Sk-learn, NumPy, seaborn, Pandas, and matplotlib, among other libraries and toolboxes. The results for classification of Ethereum transactions using approaches mentioned above are demonstrated further in this sections with the help of CEAT data visualization and analysis.

A. *Data Analysis and Visualization*

The CEAT dataset also contains number balance, transaction frequency and volume of transaction which is transformed into a balanced dataset for testing the address categorization systems. This dataset is categorized into three distinct classes of Ethereum addresses, enabling the development and evaluation of supervised ML models for predictive analytics. The goal is to enhance the detection and classification of address behaviors, ensuring robust and accurate categorization in the Ethereum blockchain ecosystem dataset.

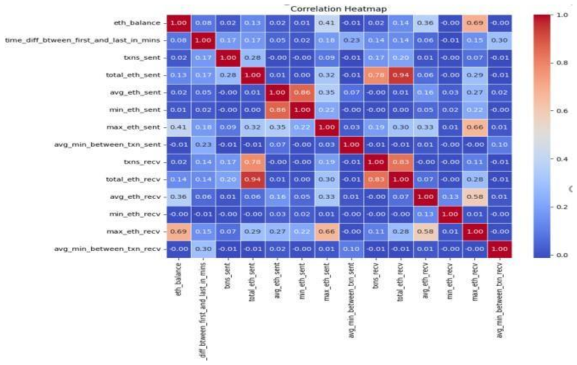


Fig. 2. Heatmap of CEAT dataset

The following Figure 2, the heatmap of the CEAT dataset image, depicts a correlation matrix, visualizing the pairwise correlations between multiple variables. The color intensity ranges from dark red (strong positive correlation) to dark shades (neutral or negative correlations). Closer numbers near 1 show a very positive connection, whereas those closer to -1 show a highly negative association.

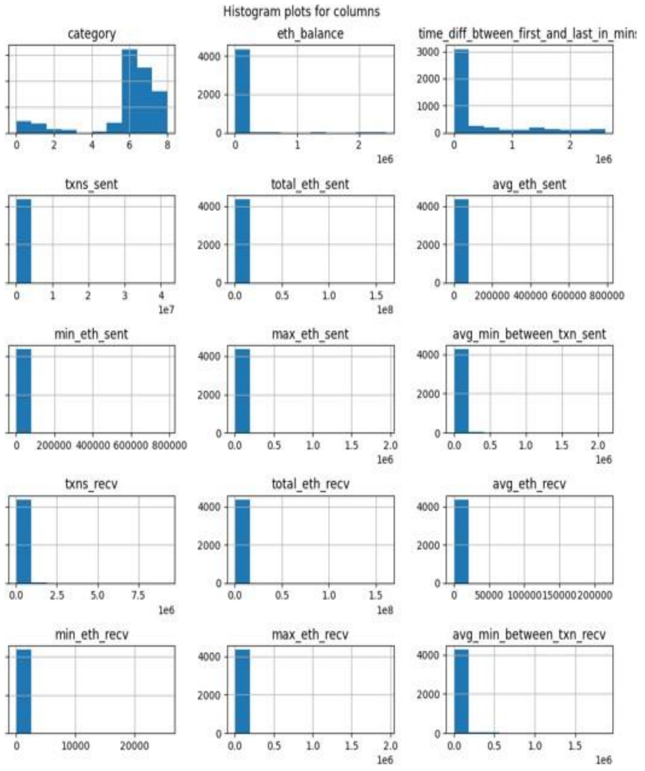


Fig. 3. Distribution in histogram CLEAT dataset

The following Figure 3 shows the distribution series of histogram plots representing the distribution of various features in a dataset. The features include transactional and balance-related metrics for Ethereum accounts. The first plot shows the "category" feature, which seems to be categorical with values distributed across a range. Other features, such as eth_balance, time_diff_between first and last in min, txns_sent, total_eth_sent, avg_eth_sent, and others related to Ethereum transactions, exhibit highly skewed distributions with most values concentrated near zero and a few extreme outliers. This pattern is consistent across features for both sent and received transactions, including the number, total, average, minimum, and maximum values, as well as the average time between transactions

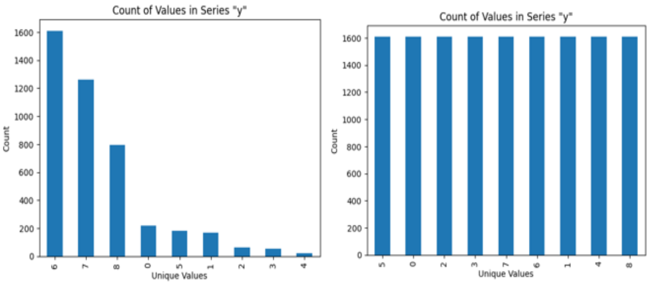


Fig. 4. Bar graphs count of values in series y after and before SMOTE

Figure 4 shows a bar graph comparing the count of values in series "y" before and after applying SMOTE. Before SMOTE, a significant class imbalance exists, with one dominant value (likely "5"). After SMOTE, the minority classes are augmented, and the distribution becomes more balanced, confirming SMOTE's effectiveness in addressing class imbalance for improved model performance.

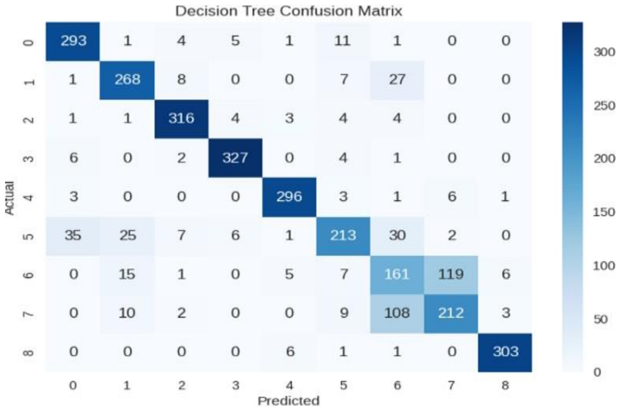
B. Experiment Results

The section presents the experimental results of the used ML models for Ethereum transaction categorization. The results are presented in table and graph style, based on performance metrics.

TABLE II. PROPOSED MODELS PERFORMANCE ON CEAT DATASET FOR ETHEREUM TRANSACTION CLASSIFICATION

Measure	DT	LightGBM	KNN	CatBoost
Accuracy	82.43	91.99	78.91	91.23
Precision	82.54	91.89	78.49	91.11
Recall	82.43	91.99	78.91	91.23
F1-score	82.38	91.93	79.06	91.13

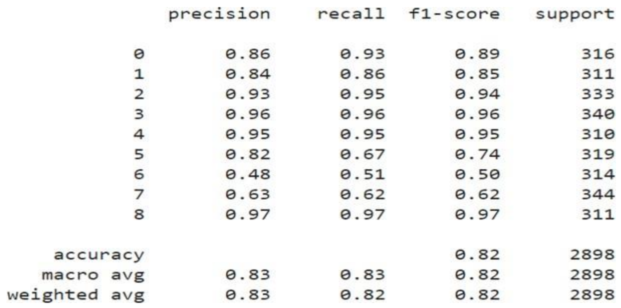
Table II presents the performance of four ML models—DT, LightGBM, KNN, and CatBooston the CEAT dataset for Ethereum Transaction classification, as measured by Rec, Acc, Prec, and F1score. LightGBM and CatBoost demonstrate superior performance, achieving accuracy scores above 91%, while DT achieves a moderate accuracy of 82.43%. KNN exhibits the lowest performance among the four, with an accuracy of 78.91%. F1score, which balances prec and rec, follows a similar trend, with LightGBM and CatBoost leading at around 91.93% and 91.13%, respectively, and KNN lagging at 79.06%. This indicates that ensemble methods like LightGBM and CatBoost are more effective for this classification task compared to traditional methods like DT and KNN.



	0	1	2	3	4	5	6	7	8
0	293	1	4	5	1	11	1	0	0
1	1	268	8	0	0	7	27	0	0
2	1	1	316	4	3	4	4	0	0
3	6	0	2	327	0	4	1	0	0
4	3	0	0	0	296	3	1	6	1
5	35	25	7	6	1	213	30	2	0
6	0	15	1	0	5	7	161	119	6
7	0	10	2	0	0	9	108	212	3
8	0	0	0	0	6	1	1	0	303

Fig. 5. Confusion matrix of Decision Tree

Figure 5 displayed a confusion matrix of a DT for Ethereum Transaction classification on the CEAT dataset that contains 9 classes 0 to 1. Class 0 achieved 293 correct predictions, with some minor misclassifications spread across classes 1-6. Class 1 had 268 correct predictions, with notable confusion with class 6 (27 instances). Class 2 performed well with 316 correct predictions. Class 3 showed strong performance with 327 correct predictions. Class 4 achieved 296 correct predictions, with minimal confusion with class 8 (6 instances). Class 5 had 213 correct predictions but showed some confusion with class 0 (35 instances) and class 1 (25 instances). Class 6 had 161 correct predictions but significant confusion with class 7 (119 instances). Class 7 had 212 correct predictions but showed confusion with class 6 (108 instances), indicating a notable bidirectional confusion between classes 6 and 7. Finally, class 8 performed very well, with 303 correct predictions and minimal confusion with other classes. The darker blue colors along the diagonal indicate strong classification performance for most classes, though the lighter blue sections, particularly between classes 6 and 7, highlight areas where the model had more difficulty distinguishing between certain transaction types.



```
              precision    recall  f1-score   support

           0       0.86      0.93      0.89       316
           1       0.84      0.86      0.85       311
           2       0.93      0.95      0.94       333
           3       0.96      0.96      0.96       340
           4       0.95      0.95      0.95       310
           5       0.82      0.67      0.74       319
           6       0.48      0.51      0.50       314
           7       0.63      0.62      0.62       344
           8       0.97      0.97      0.97       311

    accuracy                           0.82      2898
   macro avg       0.83      0.83      0.82      2898
weighted avg       0.83      0.82      0.82      2898
```

Fig. 6. Classification report of Decision tree

This classification report provides performance metrics for a Decision Tree model across nine classes (0 to 8), shown in Figure 6. The report shows detailed performance metrics across 9 classes (0-8). Class 0 achieved strong performance with prec of 0.86, rec of 0.93, and an F1score of 0.89 across 316 samples. Class 1 performed well with prec of 0.84, rec of 0.86, and F1score of 0.85 for 311 samples. Class 2 showed excellent metrics with 0.93 prec, 0.95 rec, and 0.94 F1score across 333 samples. Class 3 demonstrated the best performance with 0.96 across all metrics (prec, rec, and F1score) for 340 samples. Class 4 also performed very well, with 0.95 for all metrics across 310 samples. Class 5 showed decent performance with 0.82 prec but lower rec at 0.67, resulting in an F1score of 0.74 across 319 samples. Classes 6 and 7 showed the weakest performance - The results for Class 6 were as follows: prec 0.43, rec 0.57, and F1score 0.50 for 317 samples; for Class 7, the results were as follows: prec 0.63, rec 0.62, and F1score 0.62 for 344 samples. Class 8 showed excellent performance with 0.97 across all metrics for 311 samples. The overall model achieved acc of 0.83, with macro and weighted averages of around 0.82-0.83 across 2,898 total samples, indicating generally good but somewhat uneven performance across classes.

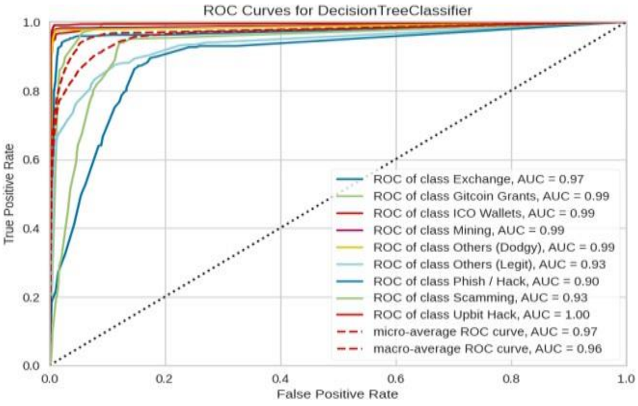
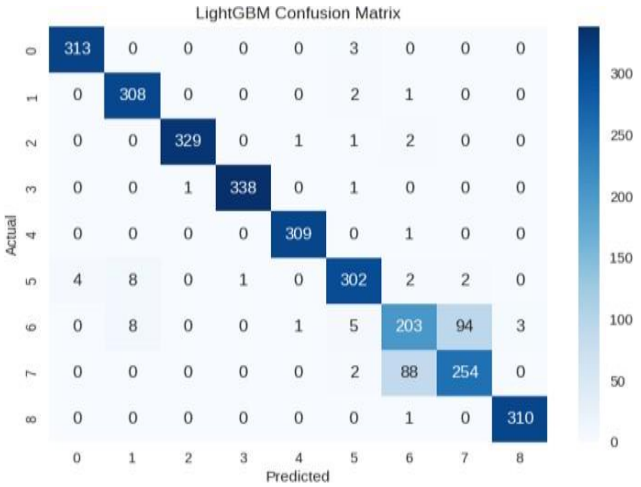


Fig. 7. Roc curve of Decision Tree

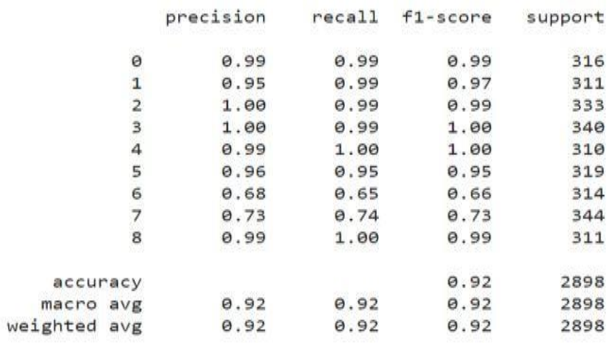
Figure 7 displays the ROC curves for a DTC across several classes. The ROC curves demonstrate the model's performance across different classes, with all classes showing strong AUC values above 0.90. The Upbit Hack class achieved perfect classification with an AUC of 1.00, represented by a curve reaching the top-left corner. The classes Gitcoin Grants, ICO Wallets, and Others (Dodgy) all performed excellently with AUC values of 0.99. Exchange transactions and Mining classes both achieved an AUC of 0.97, while the Others (Legal) class scored 0.93. Phishing/Hack activities showed good detection with an AUC of 0.90, and Scamming transactions achieved an AUC of 0.93. With an AUC of 0.97 on the micro-level and 0.96 on the macro-level, the model performs well as a whole. The Decision Tree classifier shows great effectiveness in differentiating between various kinds of Ethereum transactions, especially when it comes to identifying fraudulent activities like hacks and scams, as all class curves surpass the dotted diagonal line, which stands for random chance (AUC = 0.5).



	0	1	2	3	4	5	6	7	8
0	313	0	0	0	0	3	0	0	0
1	0	308	0	0	0	2	1	0	0
2	0	0	329	0	1	1	2	0	0
3	0	0	1	338	0	1	0	0	0
4	0	0	0	0	309	0	1	0	0
5	4	8	0	1	0	302	2	2	0
6	0	8	0	0	1	5	203	94	3
7	0	0	0	0	0	2	88	254	0
8	0	0	0	0	0	0	1	0	310

Fig. 8. Confusion matrix of LightGBM model

The LightGBM model demonstrates excellent classification across 9 Ethereum transaction classes, shown in Figure 8, with strong diagonal performance indicating accurate predictions. Most classes show minimal confusion, except for classes 6 and 7, which have mutual misclassifications (94 and 88 instances respectively). Overall, the model outperforms the previously analyzed Decision Tree model with clearer classification boundaries.



```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       316
           1       0.95      0.99      0.97       311
           2       1.00      0.99      0.99       333
           3       1.00      0.99      1.00       340
           4       0.99      1.00      1.00       310
           5       0.96      0.95      0.95       319
           6       0.68      0.65      0.66       314
           7       0.73      0.74      0.73       344
           8       0.99      1.00      0.99       311

    accuracy                           0.92      2898
   macro avg       0.92      0.92      0.92      2898
weighted avg       0.92      0.92      0.92      2898
```

Fig. 9. Classification report of LightGBM model

The LightGBM classification report shows exceptional performance across most classes, shown in Figure 9. Classes 0-5 and 8 achieve outstanding metrics with prec, rec, and F1scores ranging from 0.95 to 1.00. Class 3 achieves perfect precision and near-perfect recall, while Class 4 shows perfect recall. However, Classes 6 and 7 show relatively lower performance, with F1scores of 0.66 and 0.73 respectively, indicating some classification challenges. The model achieves a strong overall accuracy of 0.92, with consistent macro and weighted averages, demonstrating robust performance across the 2,898 total samples.

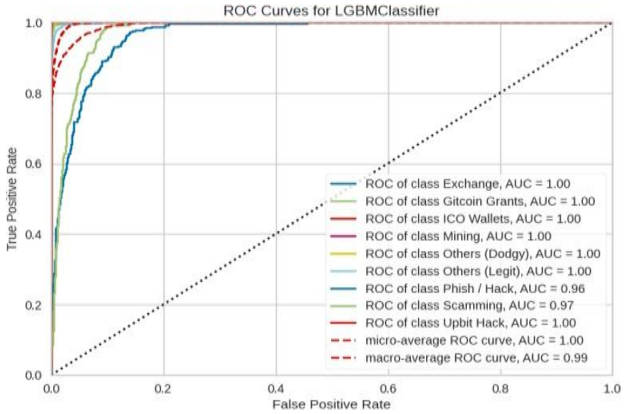
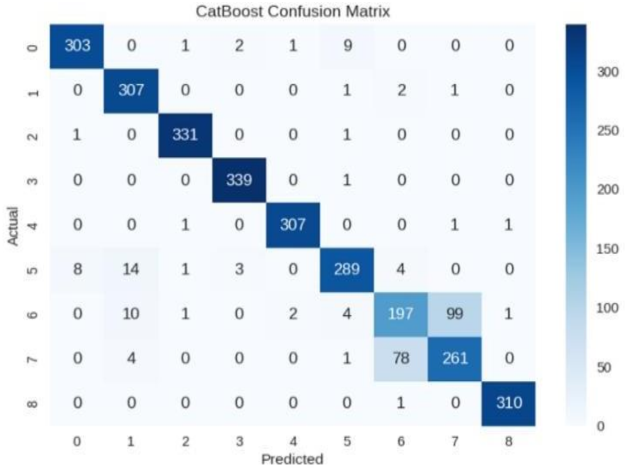


Fig. 10. Roc curve for LightGBM model

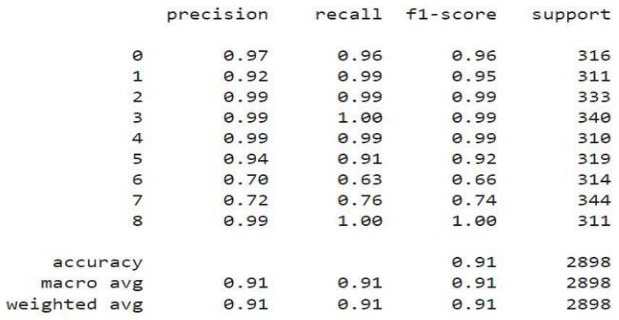
An ROC curves for a LightGBM model demonstrate exceptional classification performance across all transaction types, as shown in Figure 10. Most classes, including Exchange, Gitcoin Grants, ICO Wallets, Mining, Others (Dodgy), Others (Legal), and Upbit Hack, achieve perfect AUC scores of 1.00. The Phishing/Hack class shows strong performance with an AUC of 0.96, while Scamming transactions achieve an AUC of 0.97. Results from the Decision Tree model are greatly outperformed by the model, as shown by the micro-average ROC curve (AUC=1.00) and macro-average ROC curve (AUC=0.99).



	0	1	2	3	4	5	6	7	8
0	303	0	1	2	1	9	0	0	0
1	0	307	0	0	0	1	2	1	0
2	1	0	331	0	0	1	0	0	0
3	0	0	0	339	0	1	0	0	0
4	0	0	1	0	307	0	0	1	1
5	8	14	1	3	0	289	4	0	0
6	0	10	1	0	2	4	197	99	1
7	0	4	0	0	0	1	78	261	0
8	0	0	0	0	0	0	1	0	310

Fig. 11. Confusion matrix of Catboost model

The CatBoost model shows strong classification performance across 9 classes, with high accuracy along the diagonal, as shown in Figure 11. Classes 0-4 and 8 show excellent prediction accuracy. Notable confusion exists between classes 6 and 7, with 99 and 78 misclassifications, respectively. Class 5 shows some confusion with classes 0 and 1, but maintains good overall performance.



```
              precision    recall  f1-score   support

           0       0.97      0.96      0.96       316
           1       0.92      0.99      0.95       311
           2       0.99      0.99      0.99       333
           3       0.99      1.00      0.99       340
           4       0.99      0.99      0.99       310
           5       0.94      0.91      0.92       319
           6       0.70      0.63      0.66       314
           7       0.72      0.76      0.74       344
           8       0.99      1.00      1.00       311

    accuracy                           0.91      2898
   macro avg       0.91      0.91      0.91      2898
weighted avg       0.91      0.91      0.91      2898
```

Fig. 12. Classification report for catBoost model

The CatBoost model demonstrates excellent metrics across most classes shown in Figure 12, with particularly high performance in classes 2, 3, 4, and 8 (precision and recall ≥0.99). The F1-scores for Classes 6 and 7 are 0.66 and 0.74, respectively, indicating lesser performance. Consistent macro and weighted averages allow the model to reach a great overall accuracy of 0.91.

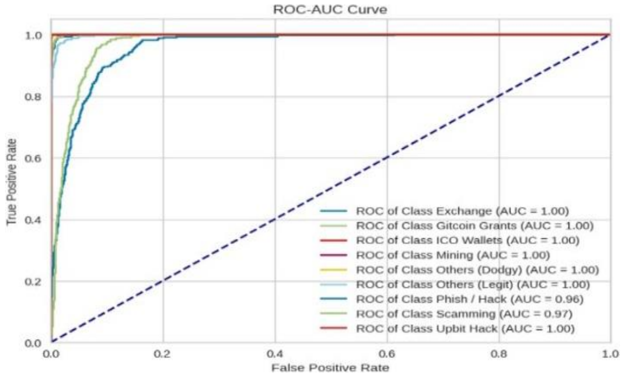
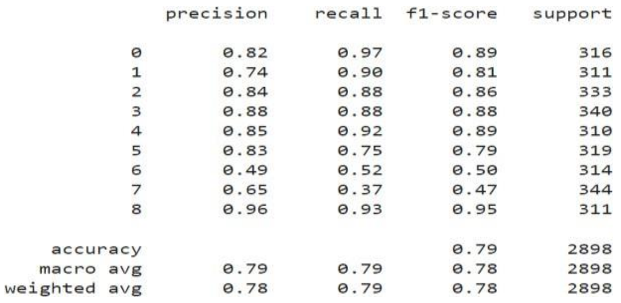


Fig. 13. Roc curve for cat Boost model

Figure 13 displayed the ROC-AUC curves for a CatBoost model classifying Ethereum transactions. Most classes achieve perfect or near-perfect AUC scores of 1.0, indicating excellent discriminatory power. Slightly lower AUCs of 0.96 and 0.97 for "Phish/Hack" and "Scamming," respectively, suggest slightly reduced performance in distinguishing these classes, though still remarkably high. Overall, the model demonstrates outstanding classification capabilities across all transaction types.



```
              precision    recall  f1-score   support

           0       0.82      0.97      0.89       316
           1       0.74      0.90      0.81       311
           2       0.84      0.88      0.86       333
           3       0.88      0.88      0.88       340
           4       0.85      0.92      0.89       310
           5       0.83      0.75      0.79       319
           6       0.49      0.52      0.50       314
           7       0.65      0.37      0.47       344
           8       0.96      0.93      0.95       311

    accuracy                           0.79      2898
   macro avg       0.79      0.79      0.78      2898
weighted avg       0.78      0.79      0.78      2898
```

Fig. 14. Classification report of K-neighbors

Figure 14 presents the classification report for K-neighbors transaction classification. It shows varying performance across classes, with classes 0, 2, 3, 4, 5, and 8 demonstrating relatively high precision, recall, and f1-scores. However, classes 6 and 7 exhibit significantly lower scores, indicating challenges in accurately classifying these transaction types, impacting the overall accuracy of 0.79, respectively.

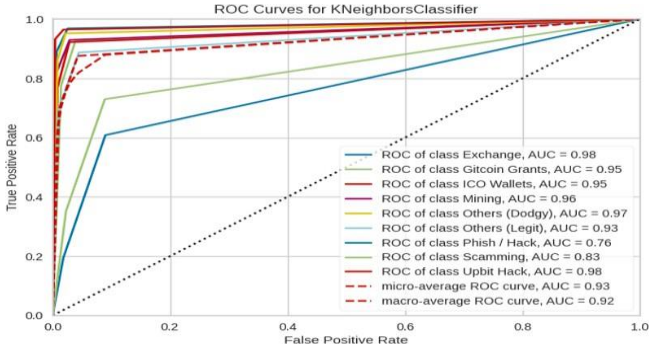


Fig. 15. ROC curve of K-Neighbors

Figure 15 shows the ROC curves for K-Neighbors transaction classification. Most classes achieve high AUCs, notably "Exchange" and "Upbit Hack" at 0.98. However, "Phish/Hack" (0.76) and "Scamming" (0.83) show lower performance, indicating reduced ability to distinguish these fraudulent classes. Overall strong but diverse performance is shown by the micro-average AUC of 0.93 and the macro-average AUC of 0.92.

C. Discussion

The performance comparison of various models reveals distinct differences in classification results shows in Table III. The DT model achieved an accuracy of 82.43%, with precision, recall, and F1-score all equal to 82.43%, indicating balanced performance but relatively lower compared to other models. LightGBM's superior accuracy (91.99%), high precision (91.89%), recall (91.99%), and F1-score (91.93%), all of which show that it effectively captures both genuine positives and negatives, surpasses all other models. KNN performed the least with an accuracy of 78.91%, precision of 78.49%, recall of 78.91%, and F1-score of 79.06%, showing lower performance. CatBoost was competitive with LightGBM, achieving 91.23% accuracy, 91.11% precision, 91.23% recall, and 91.13% F1 score. The Gaussian SVM model, with an accuracy of 78.47%, shows high precision (83.70%) but low recall (60.67%) and a poor F1-score (47.08%), indicating its failure in correctly identifying all positive instances. The AdaBoost model achieved the lowest accuracy of 67.7%, with a significantly high recall (98.8%), but low precision (39.3%) and F1-score (56.2%), highlighting its overfitting to positive class instances. Gradient Boosting showed moderate results with an accu of 76.8%, prec of 48%, and high rec (97.9%) but a lower F1score (64.4%), indicating it is better at detecting positive cases but suffers from poor precision. Overall, LightGBM and CatBoost provide the most balanced and highest performance for Ethereum address classification.

TABLE III. COMPARISON BETWEEN PROPOSED MODELS AND ANOTHER MODEL PERFORMANCE FOR ETHEREUM TRANSACTION CLASSIFICATION

Model	Accuracy	Precision	Recall	F1-score
DT	82.43	82.43	82.43	82.38
LightGBM	91.99	91.89	91.99	91.93
KNN	78.91	78.49	78.91	79.06
CatBoost	91.23	91.11	91.23	91.13
Gaussian SVM [15]	78.47	83.70	60.67	47.08
AdaBoost [36]	67.7	39.3	98.8	56.2
Gradient Boosting [36]	76.8	48	97.9	64.4

The proposed models for Ethereum address classification, including LightGBM, CatBoost, Decision Tree, and KNN, offer distinct advantages and implications. LightGBM and CatBoost have best accuracy, precision, recall, and F1 score making it even suitable for large-scale and real-time blockchain data especially when handling categorical data. Decision Tree is simple, clear and offers fairly balanced performance; however, it has comparatively low accuracy and can be effectively used only as a basic model. When implementing KNN, the results show lower efficiency, which implies it is suitable for the case of small data sets. However, issues such as high imbalance, high computational cost, overfitting and high model complexity still persist, and solutions include tuning the hyperparameters, combining shallow and deep learning, SMOTE method for data preprocessing, as well as adopting real-time implementation measures and Explainable AI for trustful model interpretation.

V. Conclusion And Future Work

This research effectively shows how supervised Machine learning can be used for classifying Ethereum addresses to improve the categorization of transactions. This means leaving a strong foundation for an effective data training since the methodology involved data preprocessing, feature selection, SMOTE-based data balancing and the Min-Max scaler normalization. DT, LightGBM, KNN, and CatBoost Classifier four classification models were developed and validated using several performance measures. In other words, the results indicated that LightGBM was the most accurate of the models with an accuracy of 91.99% and CatBoost Classifier had 91.23%. Accuracy was achieved as follows; Decision Tree was at 82.43 while the worst performer was KNN with 78.91 percent. LightGBM maintained its best performance metrics throughout all measurements of precision, recall and F1 score. These results indicate that LightGBM and CatBoost Classifier are the most effective models for Ethereum address classification in this context.

The study encounters limitations because its dataset is small and restricted in scope and does not account for all the aspects of Ethereum transactions. Supervised learning models restrict researchers from exploring alternative investigation methods during the analysis. Future studies may build a large database and consider incorporating from data that will enhance the accuracy of the model such as past addresses of the customer and their spending habits. It would also be beneficial to look into the possibility of improving the classification through classifying it as unsupervised learning methods, deep learning, and transfer learning. Real-time fraud detection systems using these models could also be developed to ensure better scalability and security for Ethereum transactions in dynamic environments.

References

[1] D. Chawla and P. S. Mehra, “A roadmap from classical cryptography to post-quantum resistant cryptography for 5G-enabled IoT: Challenges, opportunities and solutions,” *Internet of Things*, vol. 24, 2023, doi: 10.1016/j.iot.2023.100950.

[2] J. Kumar Chaudhary, S. Tyagi, H. Prapan Sharma, S. Vaseem Akram, D. R. Sisodia, and D. Kapila, “Machine Learning Model- Based Financial Market Sentiment Prediction and Application,” in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2023, pp. 1456–1459. doi: 10.1109/ICACITE57410.2023.10183344.

[3] W. Chan and A. Olmsted, “Ethereum transaction graph analysis,” in *2017 12th International Conference for Internet Technology and Secured Transactions, ICITST 2017*, 2018. doi: 10.23919/ICITST.2017.8356459.

[4] M. R. S. and P. K. Vishwakarma, “The Assessments of Financial Risk Based on Renewable Energy Industry,” *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 06, no. 09, pp. 758–770, 2024.

[5] S. Pandya, “Advanced Blockchain-Based Framework for Enhancing Security, Transparency, and Integrity in Decentralised Voting System,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 2, no. 1, pp. 865–876, Aug. 2022, doi: 10.48175/IJARSCT-12467H.

[6] S. Tyagi, “Analyzing Machine Learning Models for Credit Scoring with Explainable AI and Optimizing Investment Decisions,” *Am. Int. J. Bus. Manag.*, vol. 5, no. 01, pp. 5–19, 2022.

[7] R. Q. Majumder, “A Review of Anomaly Identification in Finance Frauds Using Machine Learning Systems,” *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 101–110, Apr. 2025, doi: 10.48175/IJARSCT-25619.

[8] S. Hisham, M. Makhtar, and A. A. Aziz, “A comprehensive review of significant learning for anomalous transaction detection using a machine learning method in a decentralized blockchain network,” *International Journal of Advanced Technology and Engineering Exploration*. 2022. doi: 10.19101/IJATEE.2021.876322.

[9] V. Prajapati, “Enhancing Supply Chain Resilience through Machine Learning-Based Predictive Analytics for Demand Forecasting,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 11, no. 3, pp. 345–354, 2025, doi: 10.32628/CSEIT25112857.

[10] A. Q. Md, S. M. S. S. Narayanan, H. Sabireen, A. K. Sivaraman, and K. F. Tee, “A novel approach to detect fraud in Ethereum transactions using stacking,” *Expert Syst.*, 2023, doi: 10.1111/exsy.13255.

[11] V. Singh, “Reinventing Business with Cloud Integration: The Cost - Effectiveness of Replacing Legacy Applications,” *Int. J. Sci. Res.*, vol. 13, no. 8, pp. 1882–1887, 2024.

[12] S. C. G. Varma and B. Chaudhari, “Federated Learning in Financial Data Privacy A Secure AI Framework for Banking Applications-edited,” 2025. doi: 10.63282/3050-9246.ICCSAIML25-112.

[13] R. M. Aziz, M. F. Baluch, S. Patel, and P. Kumar, “A Machine Learning Based Approach to Detect the Ethereum Fraud Transactions with Limited Attributes,” *Karbala Int. J. Mod. Sci.*, 2022, doi: 10.33640/2405-609X.3229.

[14] R. Q. Majumder, “Machine Learning for Predictive Analytics: Trends and Future Directions,” *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 4, pp. 3557–3564, 2025.

[15] J. Eduardo A. Sousa *et al.*, “Fighting Under-price DoS Attack in Ethereum with Machine Learning Techniques,” in *Performance Evaluation Review*, 2021. doi: 10.1145/3466826.3466835.

[16] V. C. Oliveira *et al.*, “Analyzing Transaction Confirmation in Ethereum Using Machine Learning Techniques,” in *Performance Evaluation Review*, 2021. doi: 10.1145/3466826.3466832.

[17] B. Mahesh, “Machine Learning Algorithms - A Review,” *Int. J. Sci. Res.*, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/art20203995.

[18] A. Said *et al.*, “Detailed analysis of Ethereum network on transaction behavior, community structure and link prediction,” *Peer J. Comput. Sci.*, 2021, doi: 10.7717/peerj-cs.815.

[19] R. M. Bani-Hani, A. S. Shatnawi, and L. Al-Yahya, “Vulnerability Detection and Classification of Ethereum Smart Contracts Using Deep Learning,” *Futur. Internet*, vol. 16, no. 9, 2024, doi: 10.3390/fi16090321.

[20] N. Ivanov, Q. Yan, and A. Kompalli, “TxT: Real-Time Transaction Encapsulation for Ethereum Smart Contracts,” *IEEE Trans. Inf. Forensics Secur.*, 2023 doi: 10.1109/TIFS.2023.3234895.

[21] T. Saleem *et al.*, “Predicting functional roles of Ethereum blockchain addresses,” *Peer-to-Peer Netw. Appl.*, vol. 16, 2023, doi: 10.1007/s12083-023-01553-2.

[22] R. M. Aziz, R. Mahto, K. Goel, A. Das, P. Kumar, and A. Saxena, “Modified Genetic Algorithm with Deep Learning for Fraud Transactions of Ethereum Smart Contract,” *Appl. Sci.*, 2023, doi: 10.3390/app13020697.

[23] T. T. N. Pragasam, J. V. J. Thomas, M. A. Vensuslaus, and S. Radhakrishnan, “CEAT: Categorising Ethereum Addresses’ Transaction Behaviour with Ensemble Machine Learning Algorithms,” *Computation*, 2023, doi: 10.3390/computation11080156.

[24] E. Dritsas and M. Trigka, “Supervised Machine Learning Models to Identify Early-Stage Symptoms of SARS-CoV-2,” *Sensors*, 2023, doi: 10.3390/s23010040.

[25] B. Boddu, “Serverless Databases Are the Future of Database Management,” vol. 6, no. 1, 2020.

[26] B. Boddu, “Challenges and Best Practices for Database Administration in Data Science and Machine Learning,” *IJIRMPS*, vol. 9, no. 2, p. 7, 2021.

[27] B. Boddu, “Scaling Data Processing with Amazon Redshift Dba Best Practices for Heavy Loads,” vol. 7, no. 7, 2023.

[28] M. Gopalsamy, “Identification and Classification of Phishing Emails Based on Machine Learning Techniques to Improvise Cyber security,” *IJSART*, vol. 10, no. 10, 2024.

[29] S. Mishra, “Handling Imbalanced Data: SMOTE vs. Random Undersampling",” vol. 04, 2017.

[30] H. Sinha, “An examination of machine learning-based credit card fraud detection systems,” *Int. J. Sci. Res. Arch.*, vol. 12, no. 01, pp. 2282–2294, 2024, doi: 10.30574/ijsra.2024.12.2.1456.

[31] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, “Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification,” in

Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, 2020. doi: 10.1109/ICSSIT48917.2020.9214160.

[32] S. Pandya, “Predictive Analytics in Smart Grids: Leveraging Machine Learning for Renewable Energy Sources,” *Int. J. Curr. Eng. Technol.*, vol. 11, no. 6, pp. 677–683, 2021.

[33] H. Sinha, “Predicting Employee Performance in Business Environments Using Effective Machine Learning Models,” *IJNRD-Int. J. Nov. Res. Dev.*, vol. 9, no. 9, pp. a875–a881, 2024.

[34] M. Fatima and M. Pasha, “Survey of Machine Learning Algorithms *for Disease Diagnostic,” J. Intell. Learn. Syst. Appl., 2017, doi: 10.4236/jilsa.2017.91001*.

[35] *F. Torres-Cruz, S. Tyagi, M. Sathe, S. S. C. Mary, K. Joshi, and S. K. Shukla, “Evaluation of Performance of Artificial Intelligence System during Voice Recognition in Social Conversation,” in 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), IEEE, Dec. 2022, pp. 117–122. doi: 10.1109/IC3I56241.2022.10072741*.

[36] C. Obi-Okoli, O. Jogunola, B. Adebisi, and M. Hammoudeh, “Machine Learning Algorithms to Detect Illicit Accounts on Ethereum Blockchain,” in *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, New York, NY, USA: ACM, Dec. 2023, pp. 747–752. doi: 10.1145/3644713.3644838.