Comp4801 – Interim Report

Project Title:

AI-Driven Blockchain Forensics

Member (UID):

Chow Pak Hang (3035787920)

Supervised by Dr. Chow, Kam Pui

Date of Submission:

2024-01-14

Abstract

Upon the rapid growth of blockchain technology, more criminal activities are presented with the use of cryptocurrency and related services, resulting in significant financial loss. While the forensics tool on blockchain is scarce in detection of malicious accounts and graphical visualization of transaction behaviour. Hence, this project is aims to develop application to investigate behaviour of anomaly accounts by leveraging the machine learning technique. It has explores the Logistic Regression (LR), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), and Light Gradient Boosting Machine (LightGBM), in addition, to visualize transactions of account in graphical approach, which is in form of node-link diagram. In the process of model training, a total of 13941 labelled accounts (5819 illicit and 8122 normal) are used for the implementation of the Classifier. To optimize the model performance, the sampling methods including Adaptive Synthetic Sampling (ADASYN), Edited Nearest Neighbor (ENN), and Synthetic Minority Oversampling Technique (SMOTE) are investigated and compared using the baseline model: LR. It has found that the original dataset having the highest performance of AUC value as 0.8499. The best model using LightGBM achieved an average accuracy of 96.96%, recall of 96.22% with average false positive rate (FPR) of 1.86% and false negative rate (FNR) of 3.78%. Currently, it is on the stage of Graphical Visualization of Transaction history, which involving of the research on the visualization library and unsupervised learning on account clustering. In the future, effort is required to combine the Classifier & Visualizer into single application.

Acknowledgment

Thank you for the support of the project supervisor, Dr. Chow, Kam Pui and the opportunity granted by the Department of Computer Science for doing this project.

Table of Contents

List of Figures

List of Tables

List of Equations

Abbreviations

Abbreviation	Definition
ADASYN	Adaptive Synthetic Sampling
API	Application Programming Interface
AUC	Area Under Curve
CA	Contract account
DBSCAN	Density-based spatial clustering of applications with noise
DeFi	Decentralized Finance
DT	Decision Tree
ENN	Edited Nearest Neighbor
EOA	Externally owned account
Ether	ETH
FNR	False negative rate
FPR	False positive rate
GBDT	Gradient Boosting Decison Tree
HDBSCAN	Hierarchical Density-based spatial clustering of applications with noise
ICO	Initial Coin Offering
IDS	Intrusion Detection System
KNN	K Nearest Neighbors
LR	Logistic Regression
NFT	Non-Fungible Token
LightGBM	Light Gradient Boosting Machine
RF	Random Forest
ROC	Receiver Operating Characteristic
SHAP	Shapley Additive Explanations
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TNR	True negative rate
TPR	True positive rate
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting

1 Introduction

In 2009, the Bitcoin platform launched based on the novel distributed ledger database, “blockchain", which is the transaction platform contains substantial properties such as decentralize, immutable, and transparent [1]. Recently, upon the rapid development of various applications on blockchain, in particularly, the Decentralized Finance (DeFi), Non-Fungible Token (NFT), Initial Coin Offering (ICO), and intermediary platform have shown that blockchain technology have provide a variety of benefits in financial industry and other areas [2]. For instance, cross-border payment allows minimized processing costs and efficient transactions in a more secure environment [2].

However, more cybercrimes are occurring on these blockchain platforms, especially for the public blockchains that is open-source and accessible to everyone, for instance, phishing, scam, fraudulent, fake ICO, and money laundering have cause considerable economic loss for the consumer [3]. It is recognized some of the properties of blockchain can favor the criminal activities. As an illustration, the property of decentralization, indicating no single party can control the platform, have enhanced the integrity on transaction, while it may increase the difficulty for the authorities to prohibit the illegal activities within the blockchain [4]. Moreover, the majority of public blockchains are having pseudo-anonymous identifiers for the entities involved in these blockchains, this makes the investigation towards suspicious and illicit accounts challenging [2], [4].

Presently, there are a variety of blockchain explorer websites that can query specific account data from multiple blockchains. In addition, certain intrusion detection systems (IDS) are established from scholars and experts for the recognition of anomalous activities in blockchain. Nonetheless, several blockchain explorers lacking the ability to visualize the transaction history [5] while the proposed IDSs are varying in quality and strategies [6]. As a result, it is motivated to provide a practical application that is publicly accessible for the detection of blockchain accounts and graphical visualization, concurrently, evaluating the existing anomalous classifier models for improvement of IDS in the future.

In the project, the Ethereum platform is the primary focus for forensics. Ethereum is widely adopted for the applications of smart contract, which allows diverse usage such as finance, logistics, and art in a simple and secure manner [2]. Likewise, IDS in Ethereum blockchain can be applicable to different area and targeting Ethereum is beneficial to further studies. The illicit account in this project is mainly referred as the Ethereum account undertaking several illegal activities such as phishing, scamming, fraud, and money laundering. It is proposed to have cross platform (web and mobile) application, and machine learning model for the detection of anomaly accounts.

1.1 Objectives

This project is mainly consisting of 3 objectives, which are explained in terms of the benefits of the project and rationale of related works in Section 3.

Objective 1: To Detect Anomalous Accounts on Ethereum Blockchain

Objective 2: To Visualize Transaction History in Graphical Approach

Objective 3: To Evaluate Existing Models on Anomalous Detection

1.2 Project Schedule and Status

For the status of the project, it is currently on schedule and have completed the first stage: Basic Implementation of the Illicit Account Classifier (Stage 1). Currently, it is working on the graphical visualization (Stage 2) that organize the account transaction data with graphical methods. The development of front-end web application (Stage 3) is start from January 2024 simultaneously. The details of proposed project schedule with status are stated in **Table 1.1**.

Stage	Deliverables	Details	Status
0	**Preliminary Research** Aug – Sep 2023	- Research and doing project plan	Done
-	**Phrase 1 milestones** Aug – Sep	- Complete Project Plan - Setup of Project Webpage	Done
1	**Data Preparation** Oct	- Collect blockchain dataset with normal and malicious activity	Done
1	**Illicit Account Classifier** Oct – Dec	- Data pre-processing, Algorithms Research - Model Training and Evaluation of Machine learning algorithms	Done
-	**Phrase 2 milestones** Oct – Jan	Interim Report & First Presentation - Preparation and Finalize - Preliminary Implementation	Done
2	**Data Analysis with Graphical Visualization** Dec – Mar 2024	- Organize account data with graphical method - Analyse account data for visualization	In Progress
3	**Front-End Development & Integration** Jan – Mar 2024	- Development of application and Integration of classifier model - Allow user input and graphic representations of transaction	Future
-	**Phrase 3 milestones** Mar – Apr 2024	- Preparation of Final Presentation and finalized Report - Finalized tested Implementation	Future
-	**Project Exhibition** Apr	Project Exhibition: - 3-min Video and Poster	Future

Table 1.1: Proposed project schedule with description and deadlines. Green text represents the completed stage; Red text represent the current tasks in progress; Grey background represent the current progress

1.3 Outline

In the remaining part of this report, Section 2 introduces the background of the Ethereum blockchain and analyze the related works in blockchain forensics. Section 3 describes and justifies the methodology for the implementation of the application, with the procedures and technical aspects. Sequentially, Section 4 discusses the preliminary results established, difficulties, and limitations, and potential solutions for this project. Moreover, Section 5 describes the current progress and future work. Conclusively, Section 6 concludes the main content of the report.

2 Background and Related Works

In this section, the background knowledge of the Ethereum blockchain platform is introduced (Section 2.1), the related works (Section 2.2) on illicit account detection and the visualization of the transaction in blockchain are described and compared.

2.1 Accounts and Transactions on Ethereum

In the Ethereum, all operation and the account states are maintained by the Ethereum Virtual Machine (EVM), which also ensure the valid state of the Ethereum environment [7]. There are two types of account in Ethereum, namely the Externally owned account (EOA) and the Contract account (CA) [8]. EOA is responsible for the transfer of tokens and Ether (ETH) (the native cryptocurrency in Ethereum) with another EOA. While CA is the smart contract deployed to the blockchain which is controlled by the self-executing code, only EOAs and other smart contracts can initiate the transaction from the CA [9]. In this report, the transactions between the EOAs are denoted as the *normal* transaction that recorded on the ledger, while the transactions executed from the CAs are denoted as *internal* transaction [10]. In 2022, the Ethereum have switched the consensus algorithms from Proof-of-Work (PoW) to Proof-of-Stake (PoS), which significantly reduce the energy consumption as compared to the computationally intensive process in PoW [11].

On the other hand, all the transactions are associated with the gas and gas price in Ethereum. Gas is defined as the amount of computation required for executing that transaction on the blockchain, gas (with value of 21000) are fixed for certain operation such as transferring Ethers [12]. The transaction fee is calculated from multiplies of unit of gas used and the gas price [12]. While the gas price is consisting of the base and optional priority fee, the amount of priority fee can be adjusted by the account initiated the transaction [12]. A higher priority fee increases the probability for including the transaction into the next block, indicating higher chance for faster transaction [12]. The basic structure of the transaction records in Ethereum is shown in **Table 2.1**, which summarize the important attributes that are analyzed in this project.

Field	**Description**
From	The field contains the sender's address of the transaction
To	The field contains the receiver's address of the transaction
Value	The field contains the amount of ETH in terms of in terms of wei (1 ETH = 10^{18} weis)
Data	The optional filed that is empty for ETH transfer, contains bytecode of contract at deployment
Gas Used	The field contains the gas used in the transaction
Gas Price	The field contains the gas price in the transaction
Gas Limit	The field contains the gas limit in the transaction, which set the maximum gas to be used for CA execution to avoid infinite loop of execution
Timestamp	The field contains the timestamp for the transaction being executed and included in the block

Table 2.1: The basic structure of an Ethereum transaction records, listing the important fields and their description

2.2 Related Works on Blockchain Forensics

2.2.1 Illicit Account Detection

In [13], Farrugia et al. utilized Extreme Gradient Boosting (XGBoost) to classify abnormal accounts from datasets of 4681 accounts that consisting of 2179 illicit accounts and 2502 ordinary accounts. It has achieved an average accuracy of 96.3%. However, the approach is believed to be rather simple and may need more experiment in detecting the malicious accounts in the large network of Ethereum [13]. Other studies on the detection of the illicit (or malicious) account have utilized the supervised and unsupervised learning respectively [10]. It has used the supervised algorithms such as

Random Forest (RF), K-nearest neighbor (KNN), and Extreme Gradient Boosting (XGBoost) methods, while using the unsupervised methods such as K-Means, Density-based spatial clustering of applications with noise (DBSCAN), and Hierarchical Density-based spatial clustering of applications with noise (HDBSCAN) for the analysis [10]. It has performed feature extraction from the temporal properties of the account.

On the other hand, there are several research on specific types of the illicit accounts. [14] proposed the recognition based on Network Embedding algorithm (utilized the proposed *trans2vec* algorithm) which detects the phishing accounts on Ethereum. The dataset has 1,259 addresses labeled as phishing out of 500 million address and another 1,259 normal addresses. The graph used in the model is involved of the second order transaction network, more than 60,000 nodes and 200,000 links on average in each subnetwork out of 50 random generated subnetwork [14]. It is believed the large amount of transaction data from accounts is complex and may not be efficient in distinguishing illicit accounts in real-time application without a reliable and efficient computation resources. Another work has analyzed the detection of specific types of illicit activities, the money laundering for the transactions associated with the accounts [15]. It has proposed the GTN2vec graph embeddings algorithms with an average accuracy of 95.7%, it is suggested to outweigh other related graph embedding methods [15]

Despite multiple research [10], [13] - [15] are conducted on the malicious account recognition, they are diverse in strategies and assumptions. In contrast, this project will consider several algorithms in detecting the illicit activates from accounts, moreover, providing the evaluation of the various models and an application that utilized the classifier model, which allow ordinary people and researcher to recognize the abnormal behaviour of accounts.

2.2.2 Graphical Visualization of blockchain accounts

As claimed by [5], most blockchain visualization tools are merely using simple chart and time series methods, where these approaches may not be effective in tracking money flow through transaction. For instance, the renowned blockchain explorer, Etherscan.io have certain features of analytics using the line chart, bar chart, and heatmap to represent the statistics of transactions, and account status in time series [16], however, without the transaction in form of the money-flow graph. In the use of money flow graph for the transaction history of the blockchain account, it is believed to be beneficial to certain fields including the tracing of the money flow, visualize the money flow among multiple addresses, and recognize the pattern and behaviours of the illicit accounts [17]. In the sight of lacking dedicated visualization platform for Ethereum blockchain [5], it is motivated to visualize the transaction history of account in this project, which is believed to be valuable for the future research on the illicit activities in blockchain forensics.

3 Methodology

In this section, the approaches for achieving the objectives in Section 1.1, including the implementation of anomaly account classifier (Section 3.1), graphical visualization of transaction history (Section 3.2), and integration of application and technical implication (Section 3.3) are introduced in general and technical level with justifications.

3.1 Anomaly Account Classifier

To differentiate malicious account in Ethereum blockchain, the anomaly account classifier is built with the leveraging of machine learning algorithms. It will take input from user for a specific Ethereum account address, then the address data is be collected and processed in feature extraction and cleansing procedures. Furthermore, the extracted data are analyzed with the use of the classifier, the predicted label for determining the illicit activities will be generated. To achieve the anomaly account recognizer for detection of illegal activities, several stages are divided and explained in in data collection, data pre-processing, and model training and comparison. The proposed framework is illustrated in **Figure 3.1**.

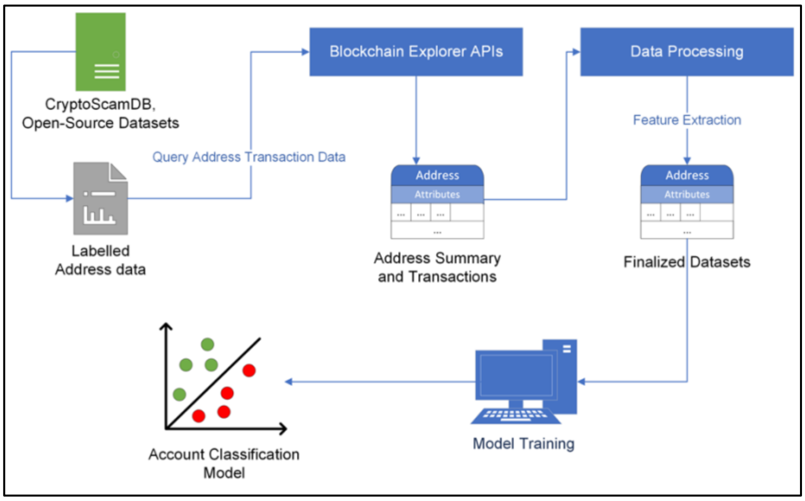

Figure 3.1: Proposed approaches on implementation of Anomaly Account Classifier

3.1.1 Data Collection

To build the classifier, labelled dataset for normal and illicit accounts is necessary for supervised and semi-supervised learning. For unsupervised learning, it is suitable for large dataset (i.e., blockchain), however, may require large number of accounts for the analysis and more difficult for evaluating its performance [18]. Thus, this project is focus on supervised learning in current stage, while unsupervised learning will be considered in next stage.

Since there is no labelling on the public Ethereum blockchain, the data is collected through several databases and open-source datasets. Labelled illicit Data are assembled through certain sources, including the academic open-source dataset [13], public cryptocurrency scam database, CryptoScamDB [19], and renowned data science resources website, Kaggle [20]. It is believed that these datasets have a high quality in correctness of the labels since various experts and research are involving in the validation. For technical tools, Python and JavaScript are utilized for web crawling purpose.

3.1.2 Data Pre-processing

On the other hand, all the collected account addresses (illicit and normal) are validated and pre-processed. For instance, they are input to a blockchain explorer called Etherscan.io [16] for checking the public name tag for “Phish/ Hack” label. For feature extraction, a selection of attributes from the accounts are extracted as the features for model training and prediction. The features are extracted based on academic evidence and heuristics for enhancing the effectiveness of the classifier model since the choice of features is vital for establishing model with precise detection of anomalies [6].

3.1.3 Model Training and Comparison

Moreover, the finalized data are processed for model training with various machine learning algorithms for comparison of the performance. To have a higher effectiveness for evaluating the models, the collected data is divided into 3 datasets, training data, validating data, and testing data. The data for training and validating consist of 90% of the collected data, which have used k-fold cross validation within the data. While testing consist of 10% of the collected data, which is mainly used for stimulating the performance of the final models as real world datasets.

It includes Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), XGBoost, and Light Gradient Boosting Machine (LightGBM). While LR is act as the baseline model for comparing the model performance. Simultaneously, the deep learning models may take into consideration.

Moreover, the significant features in affecting the performance of the models are determined by the utilization of Explainable Artificial Intelligence (XAI) methods, Shapley Additive Explanations (SHAP) is mainly used in this project. In addition, multiple models are compared and evaluated in terms of accuracy, precision, and other essential elements. Consequently, a machine learning model with highest effectiveness are employed into the application.

In technical aspects, Python and TypeScript are utilized for machine learning and deep learning, the models are either trained in the GPU farm from the Department of Computer Science or the local machine with MacOS system. This can facilitate accomplishment of the model comparison and the integration of prediction model. The final model is hosted as API for distinguishing the types of accounts (see Section 3.3

for details).

3.2 Graph Visualization of transaction history

To visualize the transaction history of an Ethereum account, several steps are depicted below, namely the real-time address data collection, transaction clustering, and visualization of processed data. Initially, the input account address data with selected features will be collected through the blockchain explorer APIs. Owing to considerable number of transactions for the account, certain important transactions will be selected exclusively for enhancing the quality and readability for further visualization. As an illustration, it will use the account transaction clustering to group similar transaction using unsupervised learning. The research of certain clustering techniques will be conducted in this stage.

Subsequent to the transaction clustering, the prior data will be visualized in the form of Node-link diagrams (demonstrated in **Figure 3.2**) and presented in the interface of the application. It is anticipated to explore the visualization libraries in JavaScript, and it is proposed to use Sigma.js [21], which provide interactive operation and customization of graph. In overall, it is expected the nodes are the account addresses for receive and send transaction associated with the input account, the links (or edges) are the amounts of transaction value and have predetermined pointing directions (send or receive from that account).

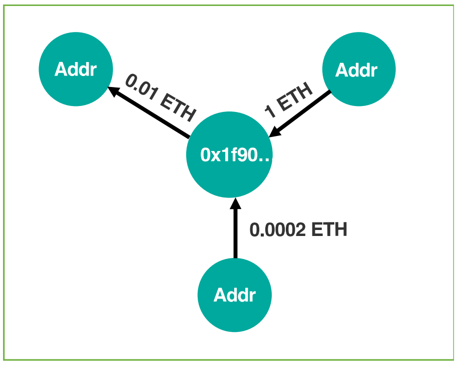

Figure 3.2: Demonstration of Node-link Diagram. The central node (green circle) is the input account, and the surrounding node are the account address that having transaction with the input account address. The arrow indicates the role of sender and recipients. The diagram is modified on demonstration purpose. Adapted from [22]

3.3 Integration of Application and Technical Implication

In this project, the application will have graphical interface for input account address and predict the risks associated with potential illicit activities. It will be primarily developed in Ionic framework and Angular as front-end. Since Ionic Framework provides cross platform application (in web and mobile), this allows convenient usage of forensics tools to inspect abnormal accounts in Ethereum. The application will be deployed on the Google Firebase for the web application. For the back-end services, it is mainly for the allocation of classifier model as API for the application, while it is developed by Python and Flask currently. Furthermore, the implemented API will generate the data required for the visualization of the graph in the application.

4 Result and Discussion

In this section, the outcomes in the implementation of the Illicit Account Classifier are described and analyzed. It includes the data collected for the illicit detection (Section 4.1), the results for the data pre-processing (Section 4.2), feature extraction (Section 4.3), and the evaluation of various machine learning algorithms in model training (Section 4.4). Further, it indicates the limitations of current results and implementations (Section 4.5).

4.1 Data Collection for illicit detection

In the collection of labelled account data in Ethereum blockchain, there are 20802 accounts (10662 illicit, 10139 normal accounts) collected from different sources (CryptoScamDB [19], Kaggle [20], academic journals [6] [13]) initially for illicit accounts. While the normal accounts are mainly collected form the Ethereum main blockchain randomly within different period of time, they are cross-validate from those illicit accounts. In overall, the account summary and latest 10,000 transaction records are collected with Blockchain explorer called the Etherscan.io [16] on or before the time of 10 November 2023. The last transaction time of the accounts is ranging from August 2015 to November 2023. The accounts without transaction records are denoted as invalid and are filtered in the view of the fact that no information is displayed to determine its nature (as normal or illicit). After filtering the duplicate and invalid accounts, a total of 13941 accounts (5819 illicit and 8122 normal accounts) are resulted, the distribution in percentage for the accounts is shown in **Figure 4.1**. However, due to the slight imbalance of datasets from number of illicit and normal accounts, this may cause inaccuracy and overfitting of models training, in specific, having lower predictive performance for the minority class (i.e., illicit account) [6] [23] [24].

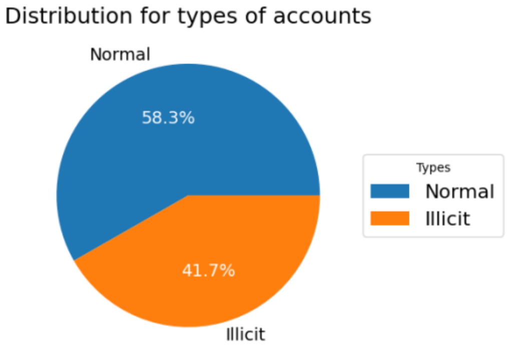

Figure 4.1: Distribution for types of collected accounts in form of pie chart

Consequently, to mitigate the issues, several strategies are considered and are experimented. In general, it has used the under-sampling, over-sampling, and combination of both, which could reduce the amount for minority class or increase the amount of the majority class. Logistic Regression is used as the baseline model for the model fitting used the data applying Adaptive Synthetic Sampling (ADASYN), Edited Nearest Neighbor (ENN), Synthetic Minority Oversampling Technique (SMOTE) with ENN. The comparison of Receiver Operating Characteristic (ROC) curves with the unsampled dataset and the various sampling methods is shown in **Figure 4.2**. It is indicated that Area Under Curve (AUC) for merely ENN having the lowest value of 0.8212, while unsampled dataset having the highest AUC of 0.8499, which is slightly higher than that SMOTE with ENN having AUC of 0.8496.

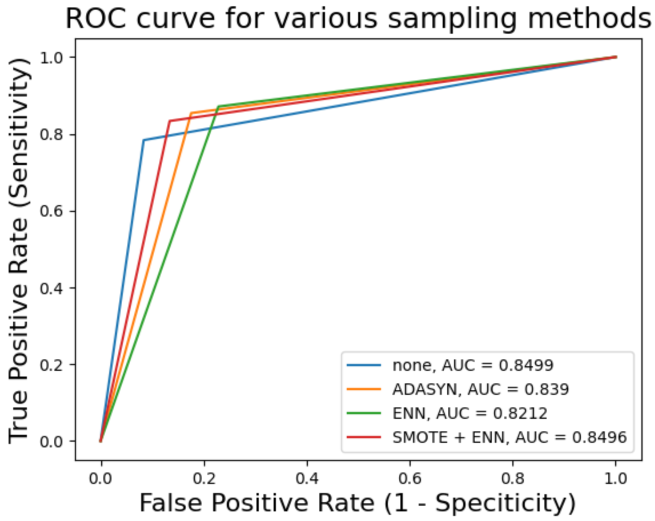

Figure 4.2: ROC curves for the various sampling methods including ADASYN, ENN, and SMOTE with ENN, and the data without sampling using Logistic Regression

Although the similar AUC value suggested that SMOTE with ENN have similar performance with the unsampled dataset, the ROC curve of SMOTE with ENN indicate a higher true positive rate (sensitivity) as compared to the unsampled dataset, which referring the higher performance in recognizing the accounts as illicit given that the accounts is illicit. Since the purpose of this project is to detect the malicious account, SMOTE with ENN may be more favorable. However, existing research shown the potential evidence of overfitting for applying certain sampling methods [25]. Hence, more research is required for adapting the sampling approaches.

4.2 Data Preprocessing

In data preprocessing, normalization of the data values is important for model fitting and classification [26]. The purpose of normalization is to transform the data into the narrow and similar scale, which favor several machine learning algorithms that compute the distances between or within different features [26]. In general, there are two types of widely used approaches used for testing: Min-Max Normalization and Standardization (or called Zero-value Normalization) [26].

The Min-Max Normalization uses the minimum and maximum value to transform the data into the fixed bound, mostly referred to range between 0 and 1, or between -1 and 1 [26]. It is effective when the data distribution is unknown, however, algorithms' performance would be affected by the value outside the minimum and maximum value (called the "outliers") used in model fitting [27] [28]. The equation of Min-Max Normalization is shown in **Equation 4.1**.

$$x_{new} = x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Equation 4.1: *Equation of the Min-Max Normalization for the calculation of the scaled value; x_{new} and x_{norm} are the scaled value; x is the original variable value; x_{min} is the minimum value among the data points, x_{max} is the maximum value among the data* [26]

Standardization uses the Z-score, consisting of the mean data value and the standard deviation. It does not scale the data in the fixed range as compared to the Min-Max Normalization. It is more effective when the data follow the normal (or gaussian) distribution [28]. The equation of the Standardization is shown in **Equation 4.2**.

$$x_{new} = Z = \frac{x - \bar{x}}{\sigma}$$

Equation 4.2: Equation of the Standardization (or Z-Score Normalization) for the calculation of the scaled value; x_{new} and Z are the scaled value; x is the original variable value; $\bar{x}$ is the variable mean value; σ is the variable standard deviation [26]

For the Ethereum account and transaction data, it is assumed majority of the features align in the gaussian distribution due to the substantial amount of transaction in the blockchain. For evaluating the effect between the two types of normalization, experiment is performed in different machine learning algorithms including logistic regression, KNN, SVM, and other tree-based algorithms. With the use of k-fold cross validation, all the non-tree-based algorithms used Standardization have a higher performance from 1% to 8% in accuracy, macro F1, and weighted F1 as compared to the Min-Max Normalization. While the tree-based algorithms: RF, XGBoost, and LightGBM are slightly influenced, about 1% higher F1 in Standardization as compared to the Min-Max Normalization, which clearly reflected that tree-based algorithms are less impacted by the normalization or feature scaling with existing research [29]. In the view of the higher performance of the standardization than the Min-Max Normalization, therefore, Standardization is being adapted in the phase of data pre-processing currently.

4.3 Feature Extraction

The features extraction is performed on the account data with selective attributes and information regarding its transaction history and status of the account. In the selection of the features for model training, the correlation between different attributes and the distribution of types of accounts are investigated. Moreover, the features for model fitting and prediction are determined with the several academic evidences [3], [6], [9], [13] - [15] that suggest the significance of certain attributes in the model predication.

The features extracted are generally divided by three types: time of transaction and between specific account status, fees related to transaction, and counting of specific occurrence of transaction. It is believed a comprehensive extraction of features is necessary for generalizing the behaviour of the account for the model fitting.

With the specific transaction mechanism in Ethereum (as stated in Section 2.1), it is suggested the criminals (owner of the illicit account) (e.g., related to scamming) may set a higher gas price to provide higher incentive for validators to include this transaction at a higher speed [10]. As a result, the features related to gas, gas price, and the transaction fee are extracted from the transactions of each account. A full list of 64 extracted features is presented in **Table 4.1**, however, some of the features may be highly correlated, which could impact the model in certain extent, hence, selection of several features may be conducted in future stage.

	Feature name	Description	Data type
1	balance	The current balance of the Ethereum account	Float
2	transaction_count	The total number of transactions made by the account	Integer
3	send_amount	The total amount sent by the account	Float
4	receive_amount	The total amount received by the account	Float
5	token_amount	The total amount of tokens held by the account	Float
6	total_token_value	The total value of tokens held by the account	Float
7	total_transaction_count	The total number of transactions made by the account, including internal transactions	Integer
8	num_of_normal_transaction	The number of regular transactions made by the account	Integer
9	out_transaction_percent	The percentage of outgoing transactions compared to total transactions	Float
10	in_transaction_percent	The percentage of incoming transactions compared to total transactions	Float
11	max_val_send	The largest amount sent by the account in a single transaction	Float
12	min_val_send	The smallest amount sent by the account in a single transaction	Float
13	mean_val_send	The average amount sent by the account in all transactions	Float
14	stdev_val_send	The standard deviation of the amounts sent by the account in all transactions	Float
15	max_val_recv	The largest amount received by the account in a single transaction	Float
16	min_val_recv	The smallest amount received by the account in a single transaction	Float
17	mean_val_recv	The average amount received by the account in all transactions	Float
18	stdev_val_recv	The standard deviation of the amounts received by the account in all transactions	Float
19	max_gas_price	The highest gas price paid by the account in a single transaction	Float
20	min_gas_price	The lowest gas price paid by the account in a single transaction	Float
21	mean_gas_price	The average gas price paid by the account in all transactions	Float
22	stdev_gas_price	The standard deviation of the gas prices paid by the account in all transactions	Float
23	mean_transaction_fee	The average transaction fee paid by the account in all transactions	Float
24	max_transaction_fee	The highest transaction fee paid by the account in a single transaction	Float
25	min_transaction_fee	The lowest transaction fee paid by the account in a single transaction	Float
26	stdev_transaction_price	The standard deviation of the transaction fees paid by the account in all transactions	Float

27	uniq_send_address_num	The number of unique addresses the account has sent transactions to	Integer
28	uniq_receive_address_num	The number of unique addresses the account has received transactions from	Integer
29	zero_val_tx_num	The number of transactions with a value of 0	Integer
30	zero_val_send_tx_num	The number of outgoing transactions with a value of 0	Integer
31	zero_val_recv_tx_num	The number of incoming transactions with a value of 0	Integer
32	mean_time_between_tx	The average time between transactions made by the account (in second)	Float
33	mean_time_between_send_tx	The average time between outgoing transactions made by the account (in second)	Float
34	mean_time_between_recv_tx	The average time between incoming transactions made by the account (in second)	Float
35	highestBalance	The highest balance the account has had	Float
36	lowestBalance	The lowest balance the account has had	Float
37	num_of_internal_transaction	The number of internal transactions made by the account	Float
38	internal_out_transaction_percent	The percentage of outgoing internal transactions compared to total internal transactions	Float
39	internal_in_transaction_percent	The percentage of incoming internal transactions compared to total internal transactions	Float
40	internal_max_val_send	The largest amount sent in a single internal transaction by the account	Float
41	internal_min_val_send	The smallest amount sent in a single internal transaction by the account	Float
42	internal_mean_val_send	The average amount sent in all internal transactions by the account	Float
43	internal_stdev_val_send	The standard deviation of the amounts sent in all internal transactions by the account	Float
44	internal_max_val_recv	The largest amount received in a single internal transaction by the account	Float
45	internal_min_val_recv	The smallest amount received in a single internal transaction by the account	Float
46	internal_mean_val_recv	The average amount received in all internal transactions by the account	Float
47	internal_stdev_val_recv	The standard deviation of the amounts received in all internal transactions by the account	Float
48	internal_max_gas	The highest gas price paid by the account in a single internal transaction	Float
49	internal_min_gas	The lowest gas price paid by the account in a single internal transaction	Float
50	internal_mean_gas	The average gas price paid by the account in all internal transactions	Float
51	internal_stdev_gas_price	The standard deviation of the gas prices paid by the account in all internal transactions	Integer
52	internal_uniq_send_address_num	The number of unique addresses the account has sent internal transactions to	Integer
53	internal_uniq_receive_address_num	The number of unique addresses the account has received internal transactions from	Integer
54	internal_zero_val_tx_num	The number of internal transactions with a value of 0	Integer
55	internal_zero_val_send_tx_num	The number of outgoing internal transactions with a value of 0	Integer
56	internal_zero_val_recv_tx_num	The number of incoming internal transactions with a value of 0	Integer
57	internal_mean_time_between_tx	The average time between internal transactions made by the account (in second)	Float

58	internal_mean_time_between_send_tx	The average time between outgoing internal transactions made by the account (in second)	Float
59	internal_mean_time_between_recv_tx	The average time between incoming internal transactions made by the account (in second)	Float
60	time_diff_between_min_balance_and_first_tx	The time difference between the first transaction made by the account and when the balance was at its lowest (in second)	Float
61	time_diff_between_max_balance_and_first_tx	The time difference between the first transaction made by the account and when the balance was at its highest (in second)	Float
62	time_diff_between_min_balance_and_last_tx	The time difference between the last transaction made by the account and when the balance was at its lowest (in second)	Float
63	time_diff_between_max_balance_and_last_tx	The time difference between the last transaction made by the account and when the balance was at its highest (in second)	Float
64	time_diff_between_first_and_last_tx	The time difference between the first transaction and last transaction from the account (in second)	Float

Table 4.1: Table listing the 64 extracted features of account data with descriptions for model fitting and prediction

With the usage of the SHAP techniques, it has found the top 10 features influence the model decisions on classifying illicit accounts to a large extent (see **Figure 4.3**), which are number of unique address that received from, the minimum gas price in transactions, mean gas price in transaction, total sending amount in transaction, time between different status and other transaction related attributes, the description of these features are indicated in **Table 4.1**. In **Figure 4.3**, it indicated that a higher number of unique address send into the account have a larger positive impact for the model to determine the account as illicit. From the gas price, the malicious account may pay higher gas price in the transaction for fast payment [10], however, the value of maximum gas price indicate a less impact on the model prediction. In summarize, the top 10 important features may show certain of the characteristics of those illicit account on Ethereum to some extent, while it may be biased owing to uneven distribution of the account.

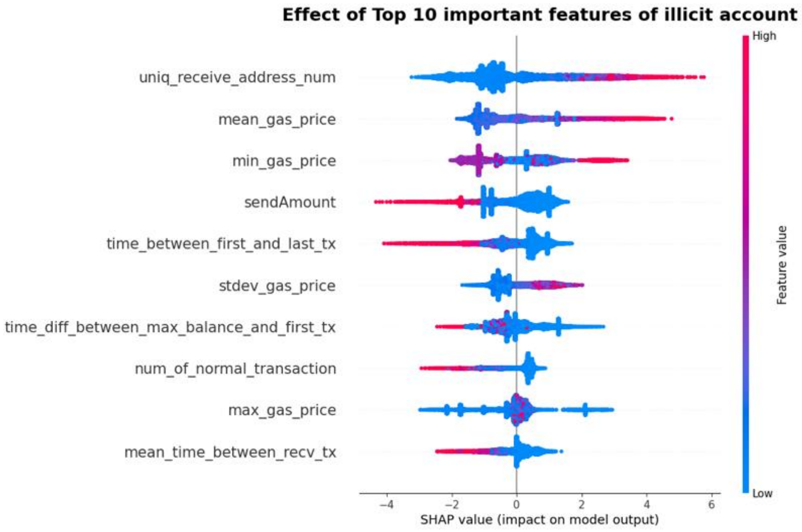

Figure 4.3: The effect of top 10 important features of the illicit account. The higher SHAP value indicate a favorable and positive influence for determine the account as illicit, and vice versa.

4.4 Model Evaluation

For the model evaluation, the positive class is the illicit account and negative class is the normal account in this project for calculation the scoring metrics. The evaluation is tested on Logistic Regression (LR), RF, KNN, XGBoost, LightGBM, SVM, and the stacking of five algorithms (SVM, RF, KNN, LR, and decision tree (DT)). The performance is evaluated using accuracy, precision, recall, specificity, F1, macro and weighted average of F1. Nevertheless, considering the biased accuracy resulting from the uneven distribution of the accounts, the recall, and specificity, and F1 would be more effective in evaluating the model performance. Since it is more important to recognize the illicit account as positive instead of negative, it should be lower the false negative rate (FNR), which indicates for a higher recall. While specificity is considered along with recall. The **Equation 4.3** and **Equation 4.4** show the equations for recall and specificity respectively.

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

Equation 4.3: The equation for calculating the recall, which is also referred to true positive rate (TPR). TP stands for true positive while FN stands for false negative

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

Equation 4.4: The equation for calculating the specificity, which is also referred to true negative rate (TNR). TN stands for true negative while FP stands for false positive

Table 4.2 summarize the performance of various machine learning algorithms in model training using the k-fold cross validation, with k equals to 10. In the evaluation, the baseline model, LR have an accuracy of 79.48%, while having a significantly lower performance in recall of 64.69%. Among all tested models, the models based on Gradient Boosting Decision Tree (GBDT) (i.e., LightGBM and XGBoost) have the highest performance. LightGBM achieved the highest performance with average accuracy of 96.96%, recall of 96.22% with average false positive rate (FPR) of 1.86% and false negative rate (FNR) of 3.78%. While XGBoost shared similar scoring as LightGBM, with slightly lower performance.

Method	**Accuracy**	**Precision**	**Recall (TPR)**	**Specificity (TNR)**	**F1**	**Macro_F1**	**Weighted_F1**
Logistic Regression	0.8466	0.8633	0.7518	0.9146	0.8036	0.8389	0.8447
Random Forest	0.9573	0.9587	0.9381	0.9710	0.9483	0.9559	0.9572
KNN	0.9085	0.9056	0.8719	0.9347	0.8884	0.9054	0.9082
XGBoost	0.9691	0.9689	0.9567	0.9780	0.9627	0.9681	0.9690
LightGBM	0.9734	0.9738	0.9622	0.9814	0.9679	0.9726	0.9734
SVM	0.9020	0.9203	0.8379	0.9480	0.8770	0.8978	0.9012
Stack (RF, KNN, LR, DT, SVM)	0.9586	0.9541	0.9465	0.9673	0.9502	0.9574	0.9586

Table 4.2: Evaluation of various machine learning algorithms using k-fold cross validation, using scoring metrics including accuracy, precision, recall, F1, macro F1, and weighted F1; The text in red indicate the highest scoring and method with highest performance

In the comparison among LightGBM and XGBoost, LightGBM have certain benefits such as faster speeds and less memory consumption in predication (or training) [30] [31]. Considering the performance and benefits of models, the LightGBM algorithms is selected as the base model for the illicit account classifier in the web application currently. With more feature engineering, other scoring metrics, and research on other algorithms such as deep learning, the main models may be altered due to the method's performance.

Furthermore, in testing the performance of LightGBM on the classification, it has utilized the testing data consisting of 813 normal accounts (positive class) and 582 illicit accounts (negative class). The results are displayed in **Table 4.3** showing the scores of precision, recall, F1-score, and accuracy for both types of accounts (normal and illicit) correspondingly. The overall accuracy is 97.85% and it achieved at least 97.5% among normal and illicit classes in all the scoring metrics (precision, recall, f1-score) in macro and weighted average respectively, with false positive rate of 1.23% and false negative rate of 2.75%. Nonetheless, all the metrics for illicit classes are lowered than the normal class, it is probably owing to the fewer data of illicit accounts in training phase, causing more incorrect prediction [23].

	Precision	**Recall**	**F1-Score**	**Support**
Normal Class	0.9804	0.9828	0.9816	813
Illicit Class	0.9759	0.9725	0.9742	582
Accuracy	-	-	0.9785	1395
Macro Average	0.9781	0.9776	0.9779	1395
Weighted Average	0.9785	0.9785	0.9785	1395

Table 4.3: Evaluation of LightGBM in testing data. The scoring metrics are precision, recall, F1-Score

among normal and illicit class; Support indicate the number of corresponding data class used for the calculate the scores. The use of hyphens indicates the value is same among the same row

The confusion matrix indicates the correct and incorrect prediction of the LightGBM using the testing data (see **Figure 4.4**). Although It shows a relative few number of false negative and positive respectively, resulting in about 2% incorrect predictions, could result in considerable amount of incorrect prediction in the real blockchain data. Moreover, there are more false negative predictions than false positive, which may result in inability to detect illicit behaviour if presence. Nevertheless, this approximation may be rather simplified since the data distribution and behaviours of the illicit accounts could be more complicated, resulting in lower performance. Consequently, a larger amount and coverage of testing data is required to have a more comprehensive analysis of the performance of the classifier.

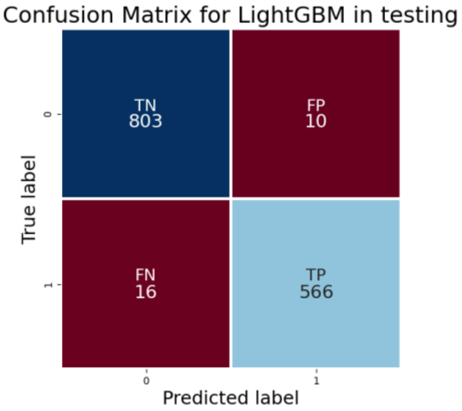

Figure 4.4: Confusion Matrix for the result of LightGBM on the testing data, where value 0 refer to normal account and value 1 refer to illicit account.

4.5 Limitations of the Classifier

In the collection of the transaction data for each account, the latest 10,000 transaction records for each of normal and internal transaction are collected and analyzed instead of all the historical transactions. These restrictions are established by the blockchain explorer APIs to limiting the computing resources [16]. Although there are certain strategies that can retrieve all the historical transaction records, majority of them are time consuming. For instance, the Google BigQuery [32] that having the public Ethereum blockchain data, would require considerable amount of time for retrieving the full transaction records, thus, may not be practical owing to the limited time, while extra expenditure may be induced for those querying. For the collected data, there are less than 1.09% of the collected account have more than 10,000 transaction records among the 13,941 accounts. Their transactions count is ranging from 10,269 to 46,335,494. Hence, the time for retrieving the transaction of these accounts may be significantly large. Instead of all the historical transaction, it is assumed that the latest 10,000 transaction is sufficient for generalizing the latest behaviour of that account, which may pose restriction on change of behaviour before those 10,000 transactions.

On the other hand, there are the restraints on evaluating the effectiveness of the classifier for the recognition of the malicious account, owing to the scarce and fixed labeled data available. amount of dataset is certainly not adequate to represent the majority of the existing Ethereum blockchain data, which these data merely account for less than 1% among the daily active Ethereum unique address [33]. Therefore, an extra data for testing the classifier in real world data is required for testing the final performance classifier, which should be explored in later stage. It is important that those extra dataset is not for evaluating the model, it is rather, to test the performance and stability of the classifier model on real time data not included in any training or test dataset.

In addition, the classifier that merely rely on the supervised machine learning algorithms may not be reliable without knowing the real types of the accounts, false prediction may be generated on the real-world scenarios that without the account labels. As a result, the greatest challenge is the validity of the unlabeled normal account, the accounts without illicit label are not necessary to be normal. In addressing these issues, unsupervised learning may be adapted for a more comprehensive classification without predetermined labels. In addition, To improve the quality of the classifier, it might need to adapt with the new data to recognize the uncovered pattern, for example, query from open-source database for that account address from user input and used for re-training in real-time. As a result, other methods must be adapted in addition with the ML models to prove a more reliable classification results.

5 Future Works

The current progress is on schedule with the completion of the basic implementation of the first milestones of the project: the Illicit Account Classifier. The present task is the further optimization of the classifier in two areas: utilize deep learning or unsupervised model, and further optimize in data pre-processing phase. Concurrently, the implementation of the Graphical Visualization for the accounts will be established. Specifically, it is involving of the visualizing the account transaction history in the form of money flow graph and the unsupervised learning techniques in address clustering.

For future work, the next stage will be the development of the front-end that combining the illicit account classifier and the visualization of the graph in the single application. It will be integrating the application to the classifier API for retrieving the classification results and the data required for the visualization, which are displayed into the application (in mobile and web) for the user.

6 Conclusion

In this report, it has introduced the proposing of the detection of anomalous accounts in Ethereum with the utilization of machine learning techniques, and the visualization of transaction history in order to mitigate and recognize the behaviour of the cybercrimes in blockchain. In the sight of increasing occurrences of cybercrimes on the blockchain such as phishing and fake ICO, this project will provide the practical application that increase the accessibility for the blockchain forensics which recognizing malicious accounts in the Ethereum blockchain. In addition, visualise the account transaction history would be beneficial to understand pattern and behaviour in a more intuitive way.

Previously, it is focused on the first stage of the project: the Illicit Account Classifier, involving of the data preparation, data preprocessing, and the evaluation of the various model. Currently, basic implementation of the first stage have been completed, the focus will be shift to the second stage: Graphical Visualization of the Transaction. In the first stage, it has experimented in the machine learning algorithms: LR, RF, KNN, LR, XGBoost, LightGBM, and stacking of five ML models (RF, KNN, LR, DT, SVM) using the k-fold cross validation in model training. The baseline model has reached the average accuracy of 84.66%, while LightGBM achieved the best average accuracy of 96.96%, recall of 96.22% with average false positive rate (FPR) of 1.86% and false negative rate (FNR) of 3.78%. In general, the tree-based algorithms (RF, LightGBM, XGBoost) are having higher performance among the tested algorithms. Although the result is fairly satisfactory comparing to the related studies [3], [9], [15], the amount of test data may not be sufficient to evaluate the performance of the model in the substantial amount of account in the public Ethereum blockchain. Therefore, an extra test data is required.

With the completion of the first stage, the next task is the implementation of the graphical visualization of the transaction, for instance, the selection of JavaScript visualization library will be conducted with several research and attempts according to the performance in application. Simultaneously, more approaches will be adapted for the optimization of the classifier. For instance, research and utilize on the various deep learning and unsupervised models. In the future, the web (and mobile) application will be developed with the integration of the Illicit Account Classifier and the Graphical visualization model.

References

[1] U. Hacioglu, *Blockchain Economics and Financial Market Innovation: Financial Innovations in the Digital Age,* Springer Nature, 2019.

[2] Ciphertrace, "Cryptocurrency crime and anti-money laundering," Ciphertrace, Mar 2023. [Online]. Available: https://ciphertrace.com/wp-content/uploads/2023/03/Ciphertrace-CAML-Report-Q3_FINAL.pdf. [Accessed 23 Oct 2023].

[3] H. Han, R. Wang, Y. Chen, K. Xie, and K. Zhang, "Research on abnormal transaction detection method for blockchain," *International Conference on Blockchain and Trustworthy Systems,* pp. 223-236, 2022.

[4] Chainalysis, "The 2023 Crypto Crime Report," Feb 2023. [Online]. Available: https://go.chainalysis.com/rs/503-FAP-074/images/Crypto_Crime_Report_2023.pdf. [Accessed 23 Oct 2023].

[5] N. Tovanich, N. Heulot, J.-D. Fekete and P. Isenberg, "Visualization of Blockchain Data: A Systematic Review," *IEEE Transactions on Visualization and Computer Graphics,* vol. 27, no. 7, p. 3135–3152, 2021.

[6] S. Al-Emari, M. Anbar, Y. K. Sanjalawe and S. Manickam, "A labeled Transactions-Based dataset on the Ethereum network," pp. 61–79. doi: 10.1007/978-981-33-6835-4_5, 2021.

[7] Ethereum.org, "INTRO TO ETHEREUM," 13 Apr 2023. [Online]. Available: https://ethereum.org/en/developers/docs/intro-to-ethereum/. [Accessed 9 Jan 2024].

[8] Ethereum.org, "ETHEREUM ACCOUNTS," 31 Jul 2023. [Online]. Available: https://ethereum.org/en/developers/docs/accounts/. [Accessed 9 Jan 2024].

[9] S. Dolev, V. Kolesnikov, S. Lodha and G. Weiss, "Detecting Malicious Accounts on the Ethereum Blockchain with Supervised Learning," *CSCML,* vol. 12161, p. 94–109, 2020.

[10] R. Agarwal, S. Barve and S. K. Shukla, "Detecting malicious accounts in permissionless blockchains using temporal graph properties," *Applied Network Science,* vol. 6, no. 9, 2021.

[11] Ethereum.org, "PROOF-OF-STAKE (POS)," 26 Sep 2023. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/. [Accessed 10 Jan 2024].

[12] Ethereum.org, "GAS AND FEES," 19 July 2023. [Online]. Available: https://ethereum.org/en/developers/docs/gas/. [Accessed 2 Jan 2024].

[13] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the Ethereum blockchain," *Expert Systems With Applications,* vol. 150, p. 113318.

[14] Wu, J. et al., "Who are the phishers? Phishing scam detection on Ethereum via network embedding," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 52, no. 2, p. 1156–1166, Feb 2022.

[15] J. Liu, C. Yin, H. Wang, X. Wu, D. Lan, L. Zhou and C. Ge, "Graph Embedding-Based Money Laundering Detection for Ethereum," *Electronics,* vol. 12, no. 14, pp. 3180-, 2023.

[16] Etherscan, "Etherscan," [Online]. Available: https://etherscan.io. [Accessed 2 Jan 2024].

[17] J. S. Tharani, E. Y. A. Charles, M. P. Z. Hóu and V. Muthukkumarasamy, "Graph Based Visualisation Techniques for Analysis of Blockchain Transactions," *IEEE 46th Conference on Local Computer Networks (LCN),* pp. 427-430, 2021.

[18] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain and A. J. Aljaaf, "A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science," *Supervised and Unsupervised Learning for Data Science,* pp. 3-21, 2020.

[19] CryptoScamDB LLC, "CryptoScamDB," [Online]. Available: https://cryptoscamdb.org. [Accessed 8 Nov 2023].

[20] Kaggle, "Ethereum Fraud Detection Dataset," 3 Jan 2021. [Online]. Available: https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset. [Accessed 8 Nov 2023].

[21] Sciences-Po médialab and OuestWare, "Sigma.js," [Online]. Available: https://www.sigmajs.org. [Accessed 20 Oct 2023].

[22] R. Severino, "The Data Visualisation Catalogue," [Online]. Available: https://datavizcatalogue.com/methods/network_diagram.html. [Accessed 23 Oct 2023].

[23] M. Bach, A. Werner and M. Palt, "The Proposal of Undersampling Method for Learning from Imbalanced Datasets Ima,∗balanced Datasetas,∗ Małgorzata Bach , Aleksandra Werner , Mateus," *Procedia Computer Science,* vol. 159, pp. 125-134, 2019.

[24] 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 International Journal of Modern Science,* p. 139–151, May 2022. doi: 10.33640/2405-609x.3229.

[25] D. Meng and Y. Li, "An imbalanced learning method by combining SMOTE

with Center Offset Factor," *Applied Soft Computing,* vol. 120, pp. 108618-, 2022.

[26] M. Butwall, "Data Normalization and Standardization: Impacting Classification Model Accuracy," *International Journal of Computer Applications,* vol. 183, no. 35, pp. 6-9, Nov 2021.

[27] H. A. Ahmed, P. J. Muhammad Ali, A. K. Faeq and S. M. Abdullah, "An Investigation on Disparity Responds of Machine Learning Algorithms to Data Normalization Method," *ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY,* vol. 10, no. 2, pp. 29-37, 2022.

[28] D. Singh and B. Singh, "Investigating the Impact of Data Normalization on Classification Performance," *Applied Soft Computing,* vol. 97, p. 105524, 2020.

[29] Ahsan, M., Mahmud, M., Saha, P., Gupta, K., & Siddique, Z., "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies (Basel),* vol. 9, no. 3, pp. 52-, 2021.

[30] R. Silhavy, P. Silhavy and Z. Prokopova, "The Comparison of Machine-Learning Methods XGBoost and LightGBM to Predict Energy Development," *Computational Statistics and Mathematical Modeling Methods in Intelligent Systems,* vol. 1047, p. 208–215, 2019.

[31] Y. Ju, G. Sun, Q. Chen, M. Z. H. Zhang and M. U. Rehman, "A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecastin," *IEEE Access,* vol. 7, p. 28309–28318, 2019.

[32] BigQuery, Google Cloud, "Ethereum Cryptocurrency," [Online]. Available: https://console.cloud.google.com/marketplace/product/ethereum/crypto-ethereum-blockchain. [Accessed 2 Jan 2024].

[33] Statista Research Department, "Daily active Ethereum (ETH) addresses up until November 9, 2022," 26 Sep 2023. [Online]. Available: https://www.statista.com/statistics/1278174/ethereum-active-addresses/. [Accessed 4 Jan 2024].

[34] S. Dyson, W. J. Buchanan, and L. Bell, "The challenges of investigating cryptocurrencies and blockchain related crime," *The Journal of the British Blockchain Association,* vol. 1, no. 2, pp. 1-6, Dec 2018.

[35] L. Yang, Q. Zhu, J. Huang and D. Cheng, "Adaptive edited natural neighbor algorithm," *Neurocomputing (Amsterdam),* vol. 230, p. 427–433, 2017.

[36] S. Pothuganti, "Review on over-fitting and under-fitting problems in Machine Learning and solutions," *International Journal of Advanced Research in*

Electrical, Electronics and Instrumentation Engineering, vol. 7, no. 9, pp. 3692-3695, Sep 2018.

[37] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper,* pp. 1-32, 2014.