Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19233-5

Blockchain transaction deanonymization using ensemble learning

Rohit Saxena[1,2] · Deepak Arora[1] · Vishal Nagar[2] · Brijesh Kumar Chaurasia[2]

Received: 26 September 2023 / Revised: 18 January 2024 / Accepted: 8 April 2024

Abstract
Bitcoin is a digital currency that provides a way to transact without any trusted intermediary; however, privacy is an issue. Numerous deanonymization endeavors have been proposed, in spite of the fact that Bitcoin addresses aren't linked with a specific identity. In this work, blockchain transactions are deanonymized using ensemble learning. An excess of four million labeled dataset samples comprising user activities such as pools, services, gambling, and exchanges have been gathered from various repositories and prepared for training and validation to perform the classification. The main aim is to deanonymize blockchain transactions via classification and separate legitimate ones from illegitimate ones. On the class imbalanced dataset, remarkable cross-validation accuracy was attained using the EXtreme Gradient Boosting with default parameters and hyperparameters. Using EXtreme Gradient Boosting, Random Forest, and Bagging on the class-balanced dataset produced the best cross-validation accuracy when using the default parameters and hyperparameters. The empirical findings indicate that the effectiveness of the proposed deanonymization using the proposed ensemble learning model has achieved up to 98.45% accuracy.

Keywords Blockchain · Supervised machine learning · Ensemble learning · Hyperparameter tuning · Randomized search · Grid search

1 Introduction

Cryptocurrency, digital payments, contactless payments, and *e*-commerce have become more popular as a result of the COVID-19 crisis. Cryptocurrencies such as Bitcoin, Tether, Ethereum, Dogecoin, etc. tend to be digital assets that offer secure and verified transactions and the creation of new assets through the use of a decentralized control system and encryption [1, 2]. The cryptocurrency known as Bitcoin first came to light in 2008 [3, 4]. Due to their distinctive features, such as the lack of centralized control, guarantee against

✉ Brijesh Kumar Chaurasia
bkchaurasia.itm@gmail.com

[1] Amity University Uttar Pradesh, Lucknow, India

[2] Pranveer Singh Institute of Technology, Kanpur, UP, India

ambiguity, and substantial level of anonymity, cryptocurrencies at large and specifically Bitcoin, have recently attracted greater interest from scientists from diversified disciplines of academia [5–7] as well as practitioners. Due to its relatively substantial degree of anonymity, Bitcoin has been termed the preferred method of payment for illegal activities. A well-known illustration in this context is the closure of the illicit drug marketplace Silk Road [8]. In addition, several publications [9, 10] allege that Bitcoin has previously been used for ransomware, theft, scams, and the funding of terrorism. Financial regulators, law enforcement organizations, intelligence agencies, and businesses that use the Bitcoin blockchain for transactions have developed a watchful eye toward the technical advancements, business challenges, and social acceptance of Bitcoin [6, 11]. In order to more effectively enlighten administrative and institutional aspects related to regulation and legal compliance, this work aims to provide a clearer understanding of the various Bitcoin transactions. We achieve this by utilizing supervised machine learning's potential to deanonymize the Bitcoin ecosystem to assist in the identification of high-risk counterparties and likely cybercriminal activities [12]. Organizations may encounter unfavorable consequences when communicating with high-risk counterparts on the Bitcoin blockchain due to legal restrictions (such as anti-money laundering procedures) or reputational risk factors. The illicit usage of Bitcoin for money laundering, financing terrorism, or cybercrime poses a serious problem for governments [9, 10]. In such circumstances, disclosing the true identities of the people in question would be ethically acceptable and permitted by law, but it could prove technically difficult, according to a common misconception about how resilient anonymity is in the Bitcoin ecosystem. Nevertheless, earlier studies [13, 14] have proven that it is possible to classify Bitcoin addresses based on user activities and connect these classes to real-world individuals. These research results contradict the generally believed perception that users' identities are protected when using Bitcoin. In this work, we examined the blockchain's transactions and deanonymized them using ensemble learning.

1.1 Motivation

The primary intent of this research is to enhance transparency in the Blockchain ecosystem and encourage consumers and businesses to accept Bitcoin as a legitimate payment method. This will contribute to the growth of the economy without resorting to illegal tactics. The results of this research will also be useful to lawmakers seeking data-driven sources for projections of the Bitcoin landscape, enterprises seeking compliance and effective risk evaluation of Bitcoin transactions, and law enforcement agencies looking to analyze and investigate Bitcoin addresses associated with illegal activities. Additionally, this study may assist in identifying illicit behaviors by linking Bitcoin addresses to suspicious activities.

1.2 Contribution

This work, therefore, aims at obtaining blockchain transaction deanonymization using the proposed ensemble learning model. The following are the significant contributions to this work:

- Preparation of labelled dataset samples obtained from various repositories, followed by cleaning, and normalizing.

- Moreover, we have also prepared a balanced dataset using oversampling and undersampling approaches.
- Determining the proportion of malicious and non-malicious activities in the Bitcoin Blockchain's transactions.
- Optimize the accuracy of classification using hyperparameters.
- Deanonymize Bitcoin transactions using the Ensemble approach.
- The results are also optimized to get the best-optimized model and achieve an accuracy of up to 98.45% for deanonymization.

1.3 Organisation of the paper

This article's remaining section is organized as follows: Related work is included in Section 2. A problem formulation is presented in Section 3. Section 4 covers the preliminaries for the proposed work. Section 5 presents the proposed work, and Section 6 presents the results and analysis. The conclusion and the future scope are illustrated in Section 7.

2 Related work

This section provides a literature review of different approaches. In relation to current research into deanonymizing blockchain transactions in the broader context of cryptocurrencies, we will assess the state of the art at the moment.

Starting with a quick assessment of related studies from the perspective of information systems, we'll go on to a summary of various legal groups' efforts to establish a regulatory framework for cryptocurrencies. We will also briefly outline the most recent advancements in de-anonymizing cryptocurrency entities.

2.1 Information systems & cyber threat intelligence

The current research that is pertinent to our work can be separated into two groups from the viewpoint of Information Systems (IS). The first section is empirical and relates to *Cyber Threat Intelligence*, and covers the literature that has been published on the subject; the second is conceptual and presents the literature in the context of *Blockchain and Cryptocurrencies*.

The studies on anonymity, identifying dishonest traders, identifying cybercrime activities, and identifying financial fraud concerning electronic markets and commerce channels are reviewed in [14–17]. Contextual data, such as participant messages, organization or industry-specific information, customer reviews, and stylometric analysis, has been utilized for carrying out a variety of sentiment analysis and to derive indicators that identify companies and individuals who engage in these types of malpractices. A meta-learning framework that improved financial fraud detection is presented in [18] using a design science approach. Abbasi et al. [19] established the Writeprint technique for identifying anonymous traders, and they proposed the use of stylometric analysis to detect traders on the internet based on the writing style traces present in the posted feedback comments. With the goal of discovering possible long-term and important members, Benjamin et al. [20]. suggested using a computational method to analyze the Internet relay chat (IRC) groups of cybercriminals. To examine IRC participation by hackers and better understand the key behaviors they display, the authors applied the extended Cox model. By utilizing an

automated and ethical web, data, and text mining approach for gathering and analyzing massive volumes of dangerous hacker tools from significant, global underground hacker networks, Samtani et al. [15]. made a significant contribution to the development of a cyber threat intelligence framework. The authors discovered numerous openly accessible harmful elements employing this framework, including keyloggers, crypters, web attacks, and database attacks. Recent breaches against companies like the Office of Personnel Management may have been brought on by some of these technologies. Abbasi et al. [19]. suggested a novel method for identifying phishing websites employing a design science approach. The suggested genre tree kernel technique uses fraud cues connected to differences in intent between genuine and phishing websites, displayed via genre composition and design structure, leading to increased anti-phishing capabilities through the use of a genre theoretic perspective. Several tests were run on a testbed made up of numerous genuine and phishing websites in order to assess the genre tree kernel approach. Abbasi et al. [20]. suggested the establishment of a new class of statistical learning theory-based fraudulent website identification systems in response to these shortcomings. They created a prototype system to show the potential utility of this class of technologies using a design-science approach. On a test bed of 900 websites, the authors [21] conducted a number of trials evaluating the suggested approach against several other fake website identification techniques.

2.2 Blockchain and cryptocurrencies

The research regarding blockchain-based technologies has been carried out by Beck et al. [18], who predicted that in the near future, distributed ledger technology would be made available to organizations, enabling them to adopt solutions. These technologies will make it easier for decentralized autonomous organizations to emerge because they will allow organizations to manage contracts and transactions independently of one another without the need for separate legal bodies [22].

Without having to provide their personal information, end users may generate pseudo-anonymous financial transactions using Bitcoin. This is accomplished by creating a user-generated pseudonym, often known as an “address”. On the one hand, users who respect their privacy were drawn to the seeming anonymity and convenience of setting up pseudo-anonymous financial transactions; on the other hand, hackers who wish to exploit it for ransomware and other illicit activities were drawn to it as well [23]. This study showed that mapping Bitcoin addresses to IP data allows for the identification of the address owners through real-time transaction relay traffic tracking. By simulating user actions and transactions on the Bitcoin Blockchain, simulation experiments were used to analyze the privacy guarantees of the cryptocurrency and reveal that, even when users take the privacy precautions that Bitcoin recommends, it is still possible to discover almost 40% of users’ profiles [24]. Numerous researchers also emphasized the shortcomings of the Bitcoin Blockchain and looked forward to a few of the alternative cryptocurrencies in addition to ideas for enhancements and/or brand-new approaches to provide users with anonymity. A protocol that allows for anonymous transactions in Bitcoin and other cryptocurrencies and depends on technology widely used by mixing services has been disclosed by some of the research’s in-depth examinations of Bitcoin’s technological workings. These analyses highlighted technical faults in the system and offered suggestions for how to repair them [25]. A noteworthy scientific effort in this field is the development of Zerocash, a Bitcoin substitute with zero-knowledge proofs, and other ZKP uses for IMoT [26], as well as theoretically feasible privacy-enhancing overlays for Bitcoin [27].

Existing literature has a few studies to attack the anonymity of blockchain transactions using machine learning and deep learning. By utilizing supervised ML to forecast the characteristics of as-yet-unidentified entities, Harlev et al. [28] developed a way to decrease the anonymity of the Bitcoin blockchain. They developed classifiers that could distinguish between ten categories using a training set of 434 entities with 200 million transactions whose identity and kind had been made public. Their main finding was an estimation of the type of unknown creature.

An ML-based approach to attacking blockchain bitcoin transactions is addressed by Zola et al. [29]. The authors employed an entity characterization strategy to challenge Bitcoin anonymity using an ML model with a suitable number of input attributes that were directly extracted from Bitcoin blockchain data, such as entity and address data, as well as developed *via* first motif and second motif principles. Yin et al. [30] introduced a method by utilizing Supervised ML to predict the nature of entities that are still unknown, thereby deanonymizing the Bitcoin Blockchain. After creating classifiers that distinguished between 12 categories using a sample of 957 entities with approximately 385 million transactions—whose identity and type had been disclosed—the authors evaluated the machine learning models using cross-validation accuracy, precision, recall, and F1-score. Together with features that have been widely used in past studies to develop a classification model for identifying abnormalities in Bitcoin network addresses, Lin et al. [31] put forward new features that include different high orders of moments of transaction time that efficiently summarize the transaction history. Supervised machine learning techniques such as Neural Network, Adaptive Boosting with Decision Tree, Random Forest, Perceptron, Support Vector Machine, Extreme Gradient Boosting, Light Gradient Boosting Machine, and Regression are used to train the extracted features. Micro-F1 and Macro-F1 were used in the evaluation of these ML models. Lee et al. [32]. developed a method for classifying Bitcoin transactions employing ML approaches in order to identify transactions that are legitimate or illegitimate. The four steps that the authors used were transaction gathering, extraction of features and labelling, model training, and testing. The gathered dataset comprises hash values from both legal and illicit transactions that were retrieved from forum websites, Blockchain Explorer, and WalletExplorer through the use of a Python script that made JSON-RPC calls. Subsequently, essential transaction features were collected, and an artificial neural network and random forest classifier were trained using the labels that were assigned. The F1-score was used to assess the performance. Li et al. [33] obtained a dataset of illicit addresses by crawling user profiles, public forums, and darknet markets via keywords like "drug," "arms," "Ponzi," "investment," "ransomware," "blackmail scam," "sextortion," "bitcoin tumbler," "darknet market," and so on in order to identify the illicit addresses in the Bitcoin network. Subsequently, the authors designed a feature set for illegitimate addresses that comprised topological, temporal, and reference-based elements from previous publications. They next evaluated features using Random Forest, Support Vector machines, eXtreme Gradient Boosting, and Artificial Neural Networks. In addition, the study suggests a model that creates temporal characteristics by integrating LSTM into an auto-encoder. A system for tracing addresses in Bitcoin systems is provided by Liu et al. [34]. The authors claim that it can be employed to alert pertinent organizations to the need to look into more private encryption schemes. To classify two Bitcoin addresses, the study first collects their properties and then integrates them into a binary classification. Subsequently, a two-level learner model was constructed to determine whether two Bitcoin addresses belong to the same user, and if so, to cluster the addresses correspondingly. The performance of the models is assessed using precision, recall, and the F1 score. In their study, Farrugia et al. [35] employed the XGBoost classifier to identify illicit accounts on

the Ethereum network by analyzing transaction histories of 2502 regular accounts and 2179 accounts that the Ethereum community had reported for illicit conduct. The authors utilized area under the curve (AUC) and cross-validation to evaluate the model after ten-fold cross-validation. In order to determine if blockchain-based systems are vulnerable to deanonymization, Michalski et al. [36] looked at whether it would be feasible to disclose the characteristics of nodes in a blockchain-based network using a set of attributes built upon transaction history and supervised ML algorithms. To ensure this, they created a dataset with around 9,000 Bitcoin node addresses and labeled them with terms like exchanges or miners. Next, using Decision Trees, Random Forests, Extra Trees, Neural Networks, Support Vector Machines, Logistic Regression, and K-Nearest Neighbors, they created a set of attributes that determine the behavior of nodes in the network. They then assessed the classification results and used the Macro-F1 score and confusion matrix to measure quality. A methodology for identifying malicious entities in the Ethereum blockchain network has been offered by Poursafaei et al. [37]. It involves extracting a collection of attributes from the Ethereum blockchain data to characterize the transactional behavior of the entities. The authors obtained a dataset from the Ethereum blockchain explorer and utilized various ensemble methods, including Stacking and AdaBoost Classifier, Random Forest, Support Vector Machine, and Logistic Regression, to identify malicious entities. The classifiers were assessed through cross-validation accuracy, precision, recall, and f1-score. In an attempt to de-anonymize Bitcoin, Kang et al. [38] used P2P network traffic to identify the IP address of each Bitcoin address's owner. To step ahead in their analysis, the authors included a procedure for clustering Bitcoin addresses, allowing them to assume that each cluster's addresses are held by a single entity. After assigning an IP address to every cluster, they demonstrated how this modification raised the proportion of trustworthy mappings they were able to locate. They employed network packet capture and analysis software Pyshark to capture inbound transaction message packets and gather transaction data along with the sender's IP address. They were able to obtain the sender's IP address as well as the raw transaction data from packets by utilizing Pyshark. They stored the gathered data in a database we created using the open-source MariaDB database. They gathered the transaction ID, the transaction's raw data, the sender's IP address, and the arrival time for every transaction that was recorded in the database. Using ML models, namely decision tree (j48), random forest, and K-nearest neighbors, Ibrahim et al. [39] analyzed illicit accounts on the Ethereum blockchain and suggested a fraud detection methodology. They used a dataset of 42 features that they downloaded from Kaggle and trained these models. The most useful features were determined by the authors using the correlation coefficient, and they then used those six attributes to create a new dataset. Elbaghdadi et al. [40] put out a model that identifies unlawful transactions by utilizing the KNN technique in conjunction with the elliptic dataset. They used the elliptic dataset for training to classify into three categories: unknown, licit, and illicit. The authors assessed the suggested model using precision, recall, and F1-score after testing it with random values for neighborhood and Euclidian distance. Nerurkar et al. [41] suggested employing an ensemble of decision trees for supervised learning as a method to deal with the problem of recognizing these illegal entities. 1216 actual Bitcoin entities were taken from the Blockchain as a dataset to assess the model. For the purpose of training the model to distinguish between 16 distinct licit and illegal user groups, nine features were designed. Support Vector Machines, Random Forest, EXtreme Gradient Boosting, and Logistic Regression were all trained and validated by the authors. Jatoth et al. [42] conducted research on identifying transactions on a blockchain that are risky and those that are not. The authors employed correlation-based feature selection in their ensemble learning, either with or without feature selection.

Accuracy, F1-score, recall, and precision have all been used to assess the learning models. Nerurkar et al. [43] identified illegitimate transactions on Bitcoin by taking nineteen features collected from the Bitcoin network and combining spectral graph convolutions and transaction features to develop a deep learning-based graph neural network model. The dataset that the authors gathered included 13,310,125 transactions from 2059 entities with 3,152,202 Bitcoin account addresses and 28 user types. They classified the transactions as either legitimate or illegitimate and specified which of the twenty-eight different categories of entities the transaction originated from. To classify illicit transactions on Bitcoin, they contrasted the model with an ensemble of decision trees and decision trees trained on convoluted features. Fidalgo et al. [44] offered approaches to generating and classifying illegitimate Bitcoin transactions. Initially, upon identifying a shortage of ground truth Bitcoin data sets—which are crucial for training supervised ML algorithms—they found about 25,000 illegitimate transactions connected to over 13,500 Bitcoin addresses associated with Ponzi schemes and ransomware payments. They looked at how Generative Adversarial Networks, particularly when working with time-series data, can accurately drive the generation of synthetic samples, which can assist in closing the gap between unbalanced data sets. As they continued to create state-of-the-art ML models for binary classification, they classified the transactions into two groups. They used LSTM Networks as an illustration of how deep learning—in particular, RNNs—can also be a great choice for tasks of such a nature. IoT-blockchain fusion for advanced data protection in Industry 4.0 is discussed in [45]. The security issues and future scope of quantum computing and blockchain are also discussed. A model for identifying suspicious activity in the Bitcoin network was put forward by Al-Hashedi et al. [46]. They built a labeled dataset by gathering a set of illicit transactions from available online Bitcoin forums, then verified and filtered the collected unlawful transactions with the original dataset and manually labeled them as legal or illegal. They also extracted a new set of time-slice-based features and balanced the skewed dataset. Three supervised classifiers—Logistic Regression, Naïve Bayes, and Artificial Neural Network—were utilized to evaluate the proposed model, with the classifiers evaluated on Precision, Recall, F1 scores, and AUC. Through the integration of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN), Umer et al. [47] suggested a method of ensemble learning for fraudulent cryptocurrency transactions. They compared the accuracy and losses from training and test datasets of off-the-shelf CNN and LSTM, ensemble CNN, and ensemble LSTM using the bagged and boosted method. Additionally, the suggested strategy was evaluated using the 10-fold cross-validation method.

3 Problem formulation

Significant challenges arise due to the illicit utilization of Bitcoin for money laundering, funding of terrorism, and cybercrime. A widely held belief about the resilience of anonymity in the Bitcoin ecosystem states that while it would be ethically and legally right to publicly disclose the identities of the participants in these circumstances, nevertheless might be practically impossible. Additionally, the great majority of clusters across the Bitcoin Blockchain are still unclassified. Hence, it's required to classify and deanonymize the Bitcoin addresses used in illegitimate transactions based on user activities.

4 Preliminaries

This section outlines the primary concepts that motivate our research on utilizing supervised machine learning to deanonymize the Bitcoin blockchain. In the context of decentralized networks like blockchain, we first provide fundamental principles and a brief discussion on the anonymity and deanonymity of blockchain transactions. We first briefly go over the fundamental ideas driving blockchain technologies before talking about how cryptocurrencies work technologically. Finally, the fundamental ideas behind the many supervised machine-learning methods employed in this research will be discussed.

4.1 Anonymity

One of the attributes that has likely been essential to the success of the cryptocurrency deployment is anonymity [48]. The ability for users to generate a limitless number of anonymous Bitcoin addresses for use in their Bitcoin transactions provides the foundation for anonymity on the Bitcoin network. The Bitcoin ecosystem is under surveillance, and the adversary can have access to the transactions coming from that address or its pseudonym. The Bitcoin ecosystem is an anonymity zone $\mathcal{B}$.

The level of anonymity of a transaction is the inability of the adversary to pinpoint the source address of the transaction $\mathcal{T}$ (anonymity set).

This anonymity set $\mathcal{T} \subseteq \mathcal{T}_{total}$ with $\mathcal{T}_{total}$ being the total number of transactions in $\mathcal{B}$. The entropy of the anonymity set is the measure of the anonymity of a transaction in the set.

If all the addresses can be the source of transactions with equal probability, then the probability p_i that the address $\mathbb{A}_i$ under observation is the target,

$$p_i = P_r(\mathbb{A}_i = \mathbb{A}), \forall\ i\ \in \mathcal{B}\ i\ and\ \sum p_i$$

The entropy [49, 50] of the distribution of the anonymity set is:

$$H(p) = -\sum p_i \log_2 p_i$$

For a transaction to be completely anonymous, all addresses involved should have an equal chance of being identified as the source. This means that the probability of any specific address, $\mathbb{A}_i$, being the source should be the same, represented by the variable p_i

$$p_i = \frac{1}{\mathbb{A}}$$

Following the definition of the level of anonymity given by Wu and Bertino [51], we have

$$\mathbb{A}_i = 1 - {}^{1}\!/_{|\mathbb{A}|}$$

With entropy $H(p) = -\log_2 \mathbb{A}$

In the context of the Internet and electronic communication, Froomkin [52, 53] proposed traceable anonymity, untraceable anonymity, traceable pseudonymity, and untraceable pseudonymity as the four unique kinds of anonymity/pseudonymity that may be used. Blockchain transaction deanonymity may also be referred to as traceable anonymity. When

information is conveyed via traceable anonymity, the sender's identity is hidden from the recipient. The sender's information is only accessible to the agent or system acting as the communication's intermediary. Anonymity networks—such as the Tor—are a vital component of the technology that has allowed the Internet to grow anonymously. Nonetheless, the number of attack techniques aimed at anonymous communication networks has been steadily rising [54]. The Sybil attack, for example, Zhang et al. [55], entails hiding high-performance nodes to obstruct node selection during path creation. According to Schnitzler et al. [56], another illustration is the denial-of-service assault, which prevents anonymous communication and permits traffic analysis.

4.2 Ensemble learning

Ensemble learning is a method for combining two or more ML algorithms to get better performance than when the individual algorithms are used alone [57]. Using a combination rule, the predictions from several learners are merged to create a single, more accurate prediction rather than relying solely on a single model. Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. In contrast to ordinary machine learning approaches which try to learn one hypothesis from training data, ensemble methods try to construct a set of hypotheses and combine them to use [58].

To arrive at weak predictive outcomes using features extracted by means of a variety of projections on data, ensemble learning methods combine results with different voting strategies to accomplish better performances than those obtained from any individual constituent algorithm alone [59]. Figure 1 illustrates that as the complexity of the model grows, the learning model's overall error consistently decreases until it crests at the bottom before rapidly climbing.

A typical ensemble classification model, as shown in Fig. 2, consists of two steps: (1) producing classification results using numerous weak classifiers, and (2) combining multiple outcomes into a consistency function to produce an outcome with voting schemes. *Adaptive Boosting(AdaBoost)*, an ensemble learning technique, uses an

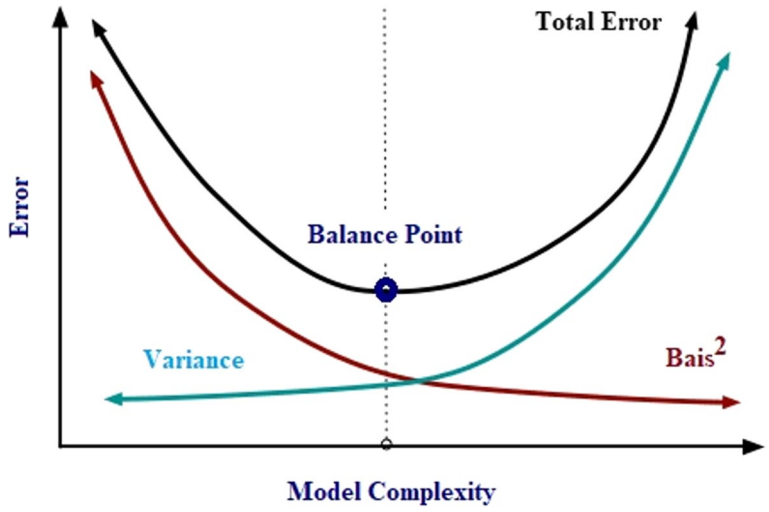


Fig. 1 The relationship between the Learning Curve and Model Complexity [59]

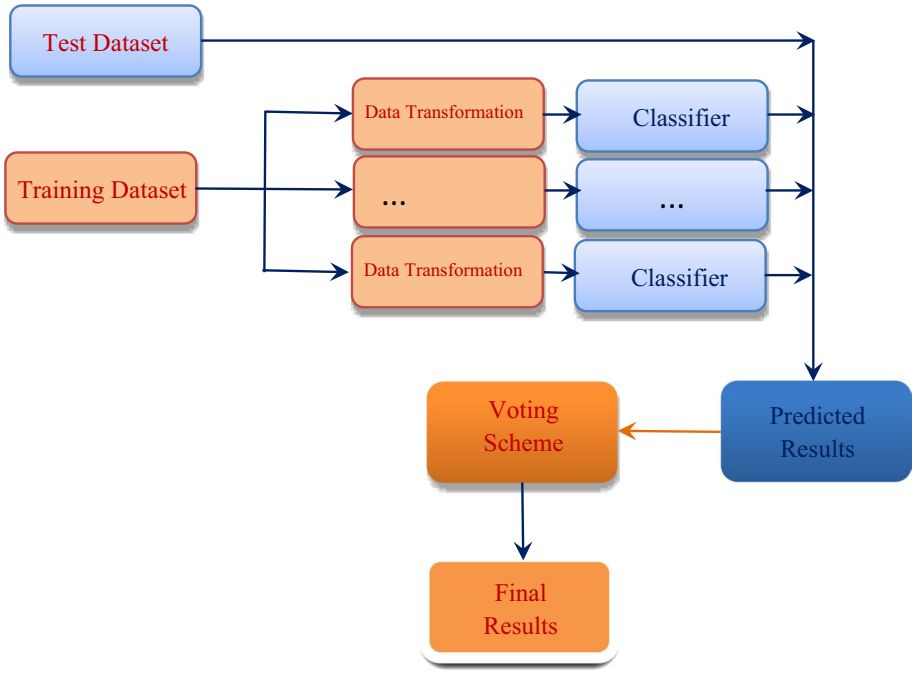


Fig. 2 Framework for Ensemble Classification

iterative procedure to help weak classifiers get better by learning from their errors. It was created by Yoav Freund et al. [60] and is also referred to as "meta-learning". *Adaboost* employs "sequential ensembling," which is different from the "parallel ensembling" used by Random Forest. It combines a number of ineffective classifiers to produce a strong classifier with a high level of accuracy. AdaBoost is referred to as an adaptive classifier in this respect since it considerably boosts the classifier's effectiveness yet occasionally results in overfitting. Despite being sensitive to noisy data and outliers, *AdaBoost* performs best when enhancing the performance of decision trees and base estimators [61], which focus on binary classification problems. Machine learning and data science both make extensive use of the ensemble classification technique known as the random forest classifier.

The parallel ensembling method utilized in this method fits numerous decision tree classifiers simultaneously on different data. Gradient Boosting is also an ensemble learning technique that, like Random Forests, builds a final model from a set of individual models, typically decision trees. A variation of gradient boosting known as *EXtreme Gradient Boosting (XGBoost)* selects the best model by taking more precise approximations into account [61]. Second-order gradients of the loss function are computed to reduce loss, and advanced regularisation (L1 and L2) is applied to enhance model performance and generalization [61]. Huge datasets can be handled with *XGBoost* with efficiency, and it is simple to use.

4.3 Blockchain transactions

Digital currencies were first created in order to establish peer-to-peer electronic assets [4]. The Bitcoin Blockchain is a distributed database that contains records of Bitcoin transactions and is the first known application of blockchain technology. A chain of digital signatures is created on a blockchain using public key cryptography protocols [62]. To send Bitcoin from one peer to another, you must first create a transaction [63]. Virtually speaking, a transaction consists of two parts: (*a*)*input*, and (*b*)*output*. This work provides an illustration of deanonymization of blockchain transactions using ensemble learning.

5 Proposed methodology

This section presents blockchain transaction deanonymization using the proposed ensemble learning model. The proposed model has five phases: data collection, data preparation and data processing, class balancing, obtaining hyperparameters using randomized search and grid search, and classification using default parameters and hyperparameters as shown in Fig. 3.

5.1 Dataset collection

The dataset employed for the research has been collected from the *Blockchair* [64] and *WalletExplorer* [65] repositories. *Blockchair* is a blockchain explorer that serves as a search engine for numerous different blockchains, such as Bitcoin, Ethereum, Litecoin, Ripple, etc. In addition to carrying out exhaustive searches on the blockchains, one can also filter and arrange blocks, transactions, and data within them using a variety of other parameters. There are a total of 22 features in every transaction obtained from this repository. These features are: *block_id, hash, time, size, weight, version, lock_time, is_coinbase, has_witness, input_count*, *output_count, input_total, input_total_usd, output_total, output_total_usd, fee, fee_usd, fee_per_kb, fee_per_kb_usd, fee_per_kwu, fee_per_kwu_usd*, and *cdd_total.* Some of them are transaction hash, block_id, timestamp, size, weight, version, fee_used, etc. *WalletExplorer* is a Bitcoin blockchain explorer that offers an easy way to view public blockchain data, i.e., Bitcoin transactions corresponding to wallets. This repository provides dataset samples of the wallets mapped to the transactions for *exchanges*, *pools*, *services*, and *gambling*. It has a total of 7 features which are *date, received from, received amount, sent amount, sent to, balance, and transaction*. We targeted to download the Bitcoin blockchain's transaction history for January, February, and March of the year 2023 from the Blockchair [64] and WalletExplorer [65] repositories. We had an option of manually downloading the transaction history from both repositories. This task could have been time-consuming so we preferred to automate this task by writing a script in Python using the *Beautiful Soup* [66] library that uses the *Selenium Webdriver protocol*.

5.2 Data preparation and preprocessing

A feature-rich labeled dataset has been prepared in this phase, and it has then been preprocessed. The dataset that is available at *Blockchair* [64] is unlabelled; however, those available at *WalletExplorer* [65] are labeled. The transaction hash (feature 'hash' from Blockchair and feature 'transaction' from WalletExplorer) is the common feature in both

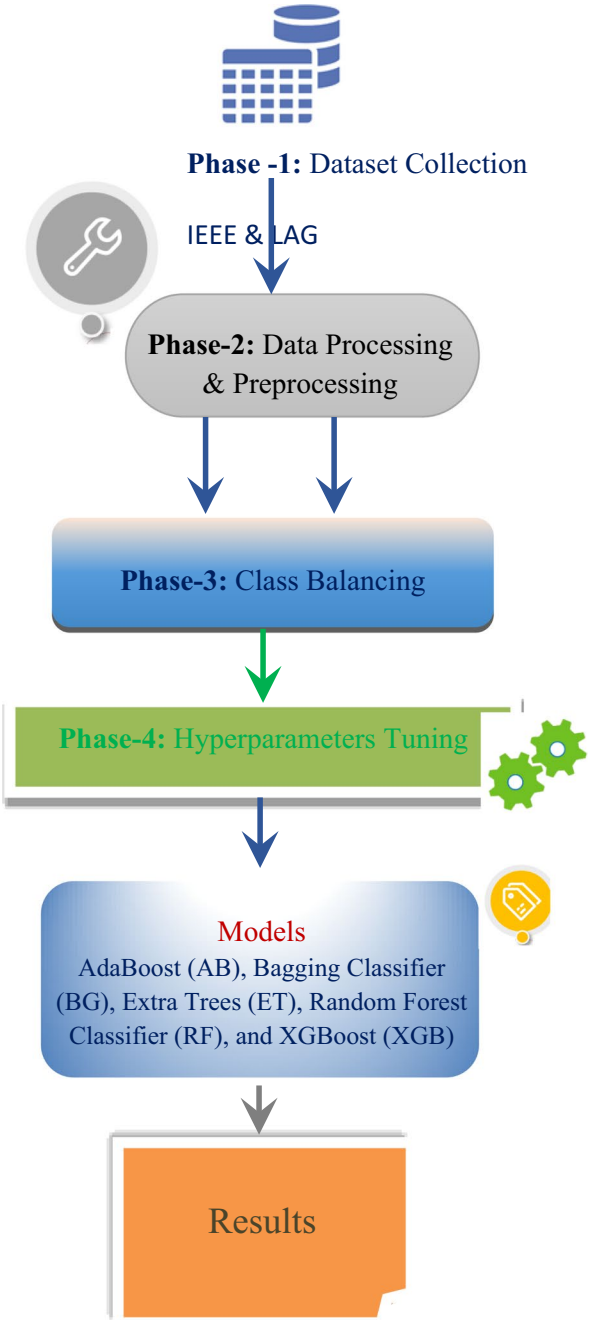


Fig. 3 The Proposed Ensemble Learning Model

datasets. Therefore, it has been used to merge the two datasets. As a result, a feature-enriched, labeled dataset with 29 features is obtained. Fig. 4 shows the data preparation and preprocessing.

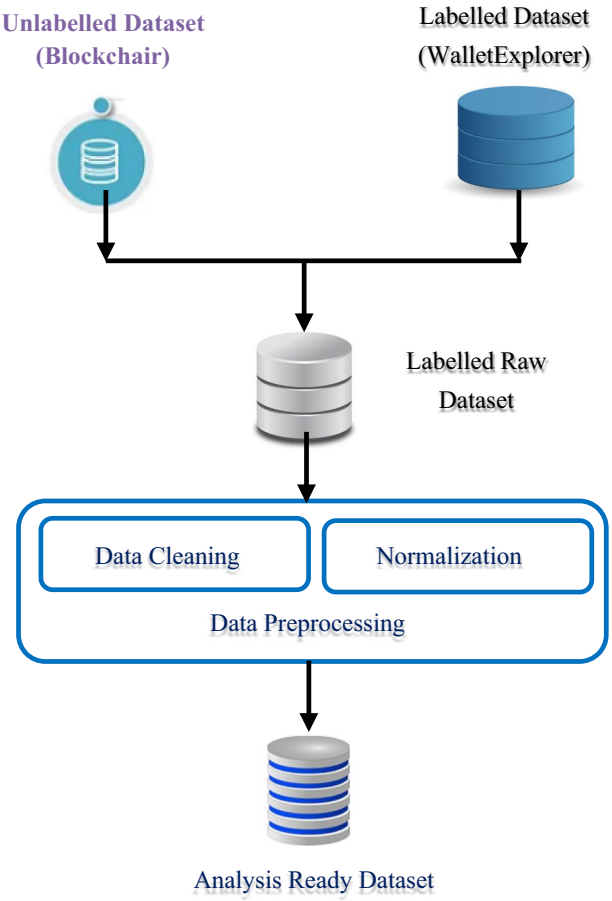


Fig. 4 The Data Preparation and Preprocessing Phase

In data preprocessing, the dataset is cleaned to eliminate the samples comprising *null* and *inf* values because the ML models cannot handle such values. String values are transformed into integer values by applying LabelEncoder [67] from Scikit-Learn to make the data suitable for ML models. Data normalization is a crucial pre-processing step that entails transforming characteristics into a common range to prevent larger numeric feature values from predominating over smaller numeric feature values [68]. The primary goal is to reduce the bias of features that have a larger numerical contribution when it comes to pattern class discrimination. When predicting the output class of an unknown instance, features in the data have been given identical weight even if their relative relevance is unknown [69]. Given that every feature in the data contributes equally to the learning process, it is highly helpful for statistical learning approaches. The dataset samples are then normalized to the same scale using the feature-based min-max scaler. *Min-max* normalization is one of the most frequently used data normalization strategies. According to predetermined lower and upper boundaries, the approach scales the unnormalized data linearly [70]. Usually, the data is rescaled to fall between 0

and 1 or -1 and 1. The minimum and maximum values for each feature are both set to 0, and all other values are set to a fractional value between 0 and 1. Here's the Eq. (1) [68]

$$X'_{i,n} = \frac{X_{i,n} - \min(X_i)}{\max(X_i) - \min(X_i)}(nMax - nMin) + nMin \tag{1}$$

where *min* and *max* represent, respectively, the i^{th} feature's minimum and maximum value. Using *nMin* and *nMax*, respectively, indicate the lowest and upper bounds to rescale the data.

5.3 Class balancing

The data preparation method produced 427,625 transactions grouped among four categories, including *exchange, pool, services*, and *gaming*, which are included in the analysis-ready dataset samples were obtained. It is evident, from Table 1, that the dataset samples' proportion of user activity is unbalanced. The dataset samples of user activities, *pool, services*, and *gambling* account for 12.95%, 7.98%, and 0.53% of the total dataset samples, respectively, while *exchange* accounts for 78.53%. A class imbalance issue resulted from the dataset samples where *gambling, services*, and *pools* remained unidentified and undersampled. Due to the unexplained nature of their behaviour, which encourages the deployment of privacy-enhancing measures, many classes continue to be undersampled. They use peeling chain mixing, which combines a customer's payments into a single address, to hide transactions. The remaining coins (change) are then sent to an address for recent changes, and the system begins sending extremely small amounts of money from that address to various other services. This procedure is repeated unless the last coin is expended. This generates a large number of change addresses, which makes it nearly impossible to identify and group addresses and hides the true source of a transaction. Performance can be attained by the prediction model; it has been demonstrated by Chawla et al. [71]. by improving the sensitivity of the classifiers to the minority classes by increasing the sample size. The categories of the dataset samples must be balanced to a great extent. ML algorithms frequently create subpar classifications if faced with unbalanced datasets. The classification result is unexpected if the event that was predicted falls to the majority class or the minority class in any unbalanced data set. Samples from the training and testing datasets are distributed at 60% and 40%, respectively. Only the samples from the training dataset were used for class balancing, and the samples from the testing dataset were retained separately. The class balancing issue is resolved using Weighted mean [72, 73] and the Synthetic Minority Oversampling Technique (SMOTE) [71].

Table 1 Categorization of Samples

User Activities	No. of Transactions per User Activities	Percentage-wise share of Activities (%)
Exchanges	335,847	78.53
Pool	55,390	12.95
Services	34,124	7.98
Gambling	2,254	0.53
Total	**427,625**	

- *Synthetic Minority Oversampling Technique (SMOTE).*

The fundamental concept is to interpolate between a number of nearby minority class examples to create new minority class samples. As a result, the overfitting issue is avoided, and the boundaries of decision-making over the minority class are expanded into the space of the majority class [68]. This method operates in "feature space" rather than "data space," generating synthetic instances of samples in a less application-specific manner. By taking each minority class sample and inserting synthetic samples along the line segments connecting any or all of the k minority class nearest neighbours, the minority class is oversampled. Randomly selected neighbours from the k-nearest neighbours are determined by the volume of oversampling necessary [71]. The *SMOTE* [74] module from the imblearn.over_sampling library of Scikit Learn has been used to generate synthetic dataset samples to balance the classes.

- *Weighted Mean*

The Weight of user activities is taken into account when calculating the Weighted Mean [72]. The Weighted Mean is utilized to decide whether to over- or under-sample the dataset as needed. For oversampling and undersampling, *RandomOverSampler* [75] and *RandomUnderSampler* [76] modules from imblearn.over_sampling library and imblearn.under_sampling library, respectively from Scikit Learn have been used. The dataset samples obtained have an even distribution for all user activities.

5.4 Hyperparameter tunning

Non-parametric models' corresponding hyperparameters need to be optimized in order to achieve stable performance results. Additional focus on this crucial stage should be given because default hyperparameter settings cannot provide the best performance of machine learning techniques [77]. Grid search is one of the most basic techniques since it evaluates each potential combination of the discrete parameter spaces that are provided. Continuous parameters have to first be discretized. Another method is randomized search, which selects hyperparameter values at random (for example, from a uniform distribution) from a predetermined hyperparameter space [78].

- *Randomized Search*

The randomized search approach evaluates the hyperparameters while selecting the best results [75, 79]. It takes precedence over every combination's random selection in its entirety. While the method can be easily extended to continuous and mixed spaces, it may be used with ease for discrete applications as well. In situations where the ML algorithm's performance is influenced by a limited number of hyperparameters [80], random search may yield better results than grid search [81]. Randomized search is effective and effectively handles data with several dimensions [82]. Randomized search was implemented using *RandomizedSearchCV* module of the Scikit learn library [83].

- *Grid Search*

A grid search is a conventional approach to optimize hyperparameters; it entails conducting a thorough search over a specific area of the training algorithm's hyperparameter space [81]. Grid search, in actuality, is an in-depth search based on subsets, whose hyperparameters are established by employing a lower limit, an upper limit, and the number of steps [84]. The grid technique thoroughly investigates all alternatives by creating a grid, which will then be assessed to determine which grid offers the best value [85]. Data execution correctness is a benefit of the grid search approach [86]. To use a grid search, we might need to define a boundary because the parameter space of the ML method might contain spaces with actual or infinite values for some parameters. Grid search suffers from high dimensional spaces, although it can frequently be parallelized with ease because the algorithm's hyperparameter values are typically unrelated to one another [81]. Grid search was implemented using *GridSearchCV* module of the Scikit learn library [87].

5.5 Classification using default parameters & hyperparameters

In this last phase, the proposed ensemble learning model, which includes adaptive boosting (commonly known as *AdaBoost*), bagging, extra trees, random forests, and extreme gradient boosting (widely known *as XGBoost*), has been employed for training and validation using the default parameters and hyperparameters obtained using randomized search and grid search techniques. We preferred to employ ensemble learning models since these models offer slightly more reliable results than other supervised ML classification and prediction techniques [88]. The proposed ensemble learning model is trained and tested using Scikit-learn libraries. The user activities are used as the y-axis and the remaining features as the x-axis for training the model and predicting the user activities, i.e., y.

6 Experimental evaluation

This section evaluates the efficacy of the proposed ensemble model used to deanonymize Bitcoin blockchain transactions. The experiments were conducted in Python 9.10.2 and Visual Studio Code 1.79.2. The processor employed for this work is the Intel(R) Core(TM) I9-10900, 2.81 GHz, with 32 GB of RAM. The user activity exchange dominated, with the highest share among the available classes in the dataset samples, followed by the pool, services, and gambling. A total of 4.0 million of data are used for the experiment. Figure 5 depicts the proportion of various user activities from the dataset gathered.

This uneven proportion of dataset samples led to the issue of class imbalance. To overcome this issue, we employed SMOTE [71], and Weighted Mean [72]. The dataset samples after the balancing of classes of user activities are given in Table 2.

6.1 Results

We have evaluated five ML models that were initially trained and tested using a 60:40 ratio, respectively. The models are *AdaBoost (AB), Bagging Classifier (BG), Extra Trees (ET), Random Forest Classifier (RF)*, and *XGBoost (XGB)*. We have evaluated the proposed model in three scenarios: the first is an imbalanced dataset, the second is a balanced dataset using SMOTE, and the third is a balanced dataset using Weighted Mean. The metrics considered are precision, recall, F1-score, and accuracy with non-parameteric classification,

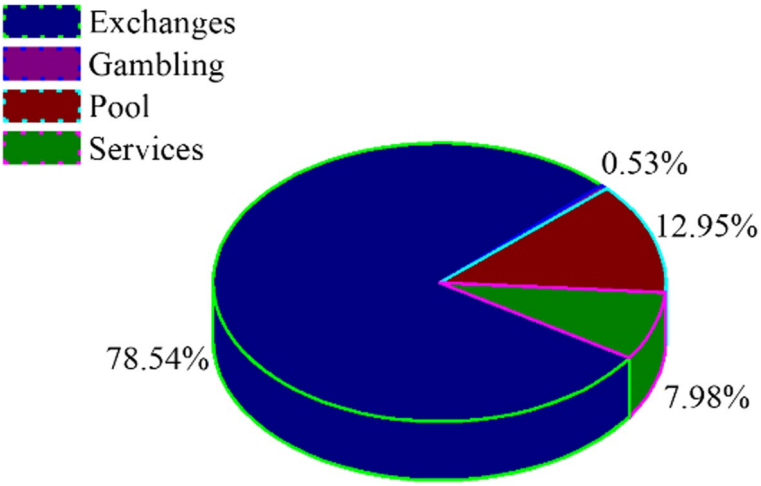
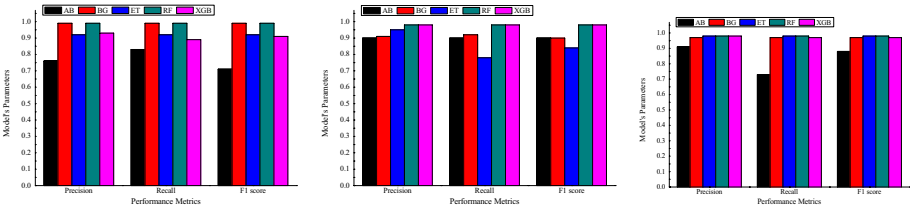


Fig. 5 Dataset of Blockchain Transactions

Table 2 Balanced Dataset

User Activities	Generated Dataset Samples for Training		
	Unbalanced	SMOTE	Weighted Mean
Exchanges	335,847	201,580	64,143
Gambling	55,390	201,580	64,143
Pool	34,124	201,580	64,143
Services	2254	201,580	64,143

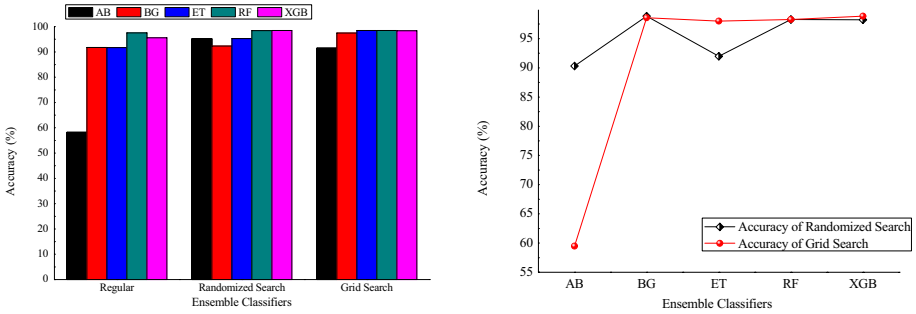


(a) Regular (b) Randomized Search (c) Grid Search

Fig. 6 Performance of Proposed Ensemble Model over Imbalanced Dataset

i.e., regular (REG) and hyperparameteric classification, i.e., randomized search (RS), and grid search (GS).

Scenario 1: Performance over Dataset Samples with Imbalanced Classes



(a) Comparative accuracy over REG, RS, GS

(b) Comparative accuracy over RS, GS

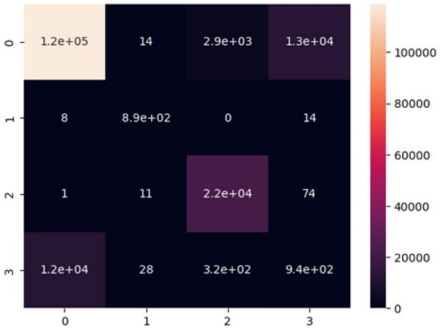
Fig. 7 Accuracy of the Proposed Ensemble Model over Ensemble Classifiers

Figure 6 shows the performance of the proposed ensemble learning model. The performance of the *XGB* model is better for REG and RS, i.e., up to 98.73%. However, the performance of *RF* is constant in all three scenarios. Moreover, it is also observed that the performance of the *AB* model in terms of recall is increased by 76–83% in the case of randomized search; however, in the case of grid search, it is reduced by 76–71%. In addition, the tuning of hyperparameters has improved the model parameters to a great extent. Although it can be visualized that tuning has a mixed effect on precision, recall, and F1-score, for *BG* and *ET*, grid search has better outcomes for precision, recall, and F1-score, while for *RF*, randomized search has produced better outcomes. It is evident that the performance of all the models except the *AB* model is either increased or constant over randomized search and grid search, respectively.

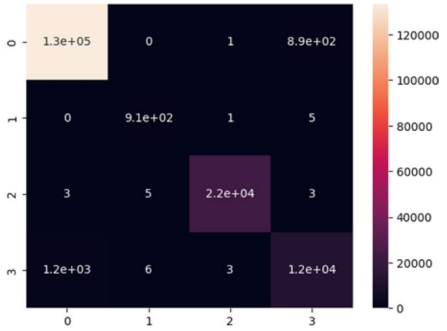
Figure 7 shows the accuracy of the proposed ensemble model over an imbalanced class dataset. Figure 7a shows the accuracy of the proposed model over REG, RS, and GS; however, Fig. 7b is a comparative analysis between RS and GS of our proposed model.

A detailed description of accuracy is as follows:

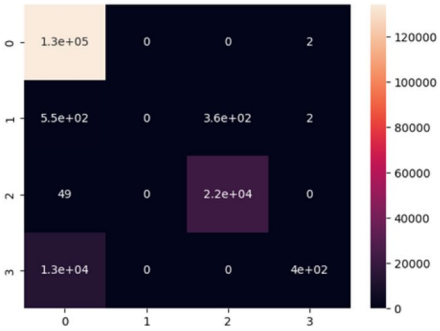
- *Regular vs. Randomized Search*: There is a high spike made by the *AB* from 58.3 to 95.18%, but the time taken to obtain the hyperparameters is relatively high (18 *secs*., and 960 *secs*. in the case of REG and RS, respectively). The *RF* and *XGB* can be seen competing with CV accuracy of 98.73% and 98.48%, respectively, but the *RF* has an edge in the time taken to obtain the hyperparameters.
- *Regular vs. Grid Search*: The CV accuracy of *AB*, *BG*, and *ET* has been optimized from 58.3 to 91.52%, 91.8–97.48%, and 91.7–98.5%, respectively. A slight improvement can be noticed for *RF* and *XGB*.
- *Randomized Search vs. Grid Search*: The randomized search has given better CV accuracy for *AB* in almost half of the time taken to obtain the hyperparameters in comparison to the grid search. The CV accuracy of *RF* and *XGB* is almost the same, but considering the time taken to obtain the hyperparameters, the grid search has outperformed the randomized search. For *ET*, grid search has better accuracy and takes less time to obtain the hyperparameters.



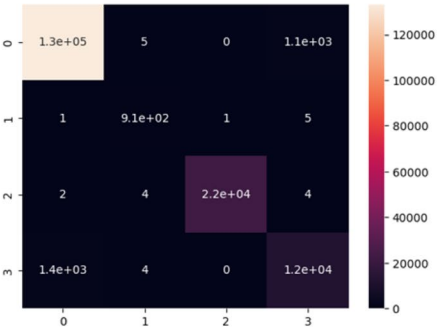
a Confusion Matrix for AB



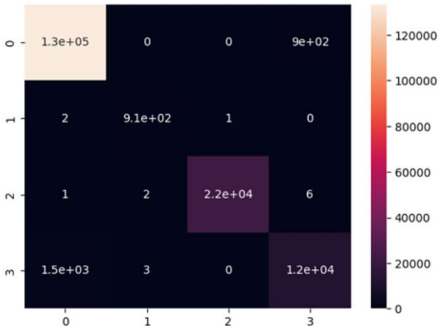
b Confusion Matrix for BG



c Confusion Matrix for ET



d Confusion Matrix for RF

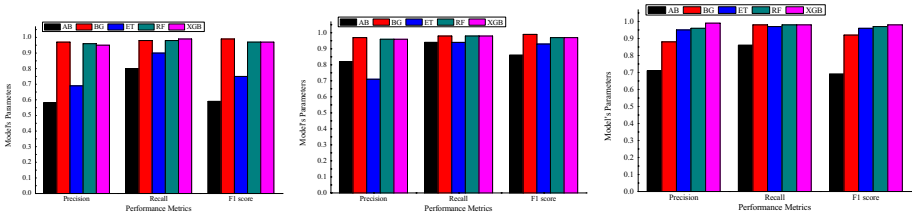


e Confusion Matrix for XGB

Fig. 8 Confusion Matrices of the Proposed Model Using Regular Classification

Figure 7 clearly shows that the accuracy of the proposed ensemble model of all the cases is increased over either randomized search or grid search, respectively.

Figure 8a–e are the confusion matrices of the proposed ensemble models for blockchain transaction deanonymization without using hyperparameters. The classes are 0, 1, 2, and 3 for exchanges, gambling, pools, and services, respectively. The confusion matrix shown in Fig. 8a shows the classification using the AdaBoost (AB) model. It is



(a) Regular (b) Randomized Search (c) Grid Search

Fig. 9 Performance of Proposed Ensemble Model over Balanced Dataset using SMOTE

observed that the correctly classified transactions are 1.188 million, 892, 22,003, and 941 for exchanges, gambling, pools, and services classes. On the other hand, the highest misclassified transactions are 12,645, 14, 74, and 12,331 for classes 0 to 3.

Figure 8b shows the classification using the Bagging Classifier (BG) model. It is observed that the correctly classified transactions are 1.335 million, 908, 22,078, and 12,379 for exchanges, gambling, pools, and services classes. On the other hand, the highest misclassified transactions are 894, 5, 5, and 1234 for classes 0 to 3.

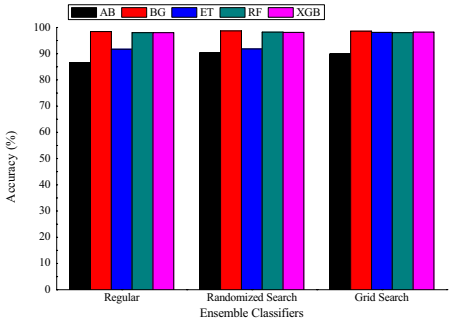
The classification using the Extra Trees (ET) model is displayed in Fig. 8c. It has been noted that the transactions for exchanges, gaming, pools, and services classes—1.344 million, 0, 22,040, and 398—have been appropriately classified. However, for classes 0 through 3, the most misclassified transactions are 2, 360, 49, and 13,224. It is clearly shown that the ET model is not working well for gaming class.

Figure 8d shows the classification using the Random Forest Classifier (RF). It has been observed that the transactions for the classifications of services, games, exchanges, and pools—1.332 million, 907, 22,079, and 12,247—have been correctly categorised. Nonetheless, the most often misclassified transactions for classes 0 through 3 are 1136, 5, 4, and 1371. The RF classifiers's significant performance in all classes is evident.

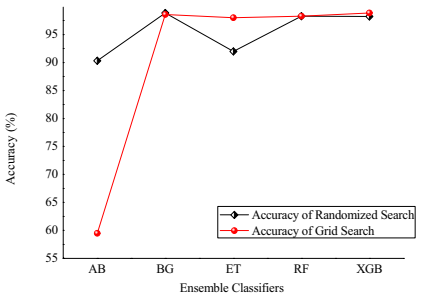
The XGBoost (XGB) model appears in Fig. 8e. The transactions for the classifies of services, games, exchanges, and pools —1.335 million, 911, 22,080, and 12,138—have all been found to be appropriately classified. However, 1136, 5, 4, and 1371 are the most frequently misclassified transactions for classes 0 through 3. It is clear that the XGB model's performance is also satisfactory in all classes.

Scenario 2: Performance over Class Balanced Dataset using SMOTE

Figure 9 shows the performance of the proposed ensemble learning model over a balanced dataset using the SMOTE technique. The performance of the XGB model is better for REG and RS, i.e., up to 98.45%. However, the performance of *RF* is constant in all three scenarios. Moreover, it is also observed that the performance of the *AB* model in terms of recall is increased by 58–71% in the case of randomized search; however, *BG's* performance in the case of grid search has reduced from 97 to 88%. In addition, the tuning of hyperparameters has improved the model parameters to a great extent. Although it can be visualised that the values of precision, recall, and F1-score for *AB* and *ET* have been optimized by randomized and grid search. The precision, recall, and F1-score of *BG, XGB*, and *RF* have been found to be the best for randomized search, while those of *XGB*, *RF*, and *ET* are best for grid search.



(a) Comparative accuracy over REG, RS, GS



(b) Comparative accuracy over RS, GS

Fig. 10 Accuracy of the Proposed Ensemble Model over Balanced Dataset using SMOTE

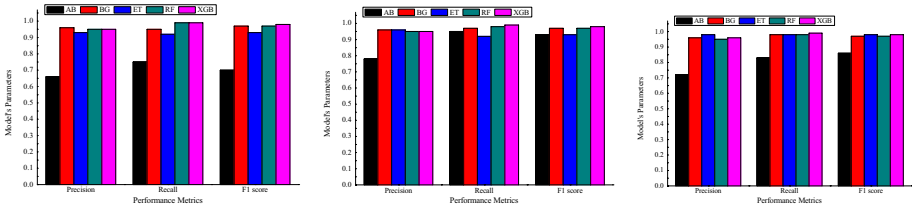
Figure 10 shows the accuracy of the proposed ensemble model over a balanced class dataset using SMOTE. Figure 10a shows the accuracy of the proposed model overall REG, RS, and GS; however, Fig. 10b is a comparative analysis between RS and GS of our proposed model.

A detailed description of accuracy is as follows:

- *Regular vs. Randomized Search*: Although *AB*'s accuracy has substantially improved, going from 55.27 to 90.31%, obtaining the hyperparameters has taken a while. Due to the fact that the difference in the respective CV accuracy is very small and the time required to collect the hyperparameters is extremely large, it can be seen that applying randomized search to *XGB*, *RF*, and *ET* has no change or very little effect on accuracy.
- *Regular vs. Grid Search*: From Fig. 10, grid search appears to have a moderate effect on *AB* and *ET*, as can be seen. The CV accuracy of *AB* remains fairly low in comparison to other models even after the model is trained using the hyperparameters discovered by grid search. There has been a modest improvement in the CV accuracy for *XGB*, *RF*, and *BG*.
- *Randomized Search vs. Grid Search*: It is evident from Fig. 10 that, *BG* and *ET* have given better accuracy on the hyperparameters obtained using grid search in comparison to randomized search. In comparison to randomized search, grid search takes much longer to find the hyperparameters for *BG*, whereas for *ET* it works the other way around. The accuracy of *RF* is nearly identical for both hyperparameter tuning techniques; however, grid search takes much less time than randomized search to obtain the hyperparameters. Similarly, the *XGB* accuracy is about the same for both procedures, although randomized search has an advantage in terms of time consumption.

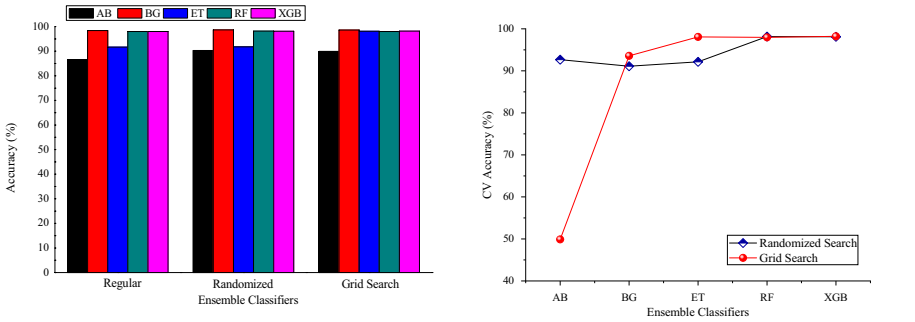
Figure 10a and b clearly show that the accuracy of the proposed ensemble model in all cases is either increased over randomized search or grid search, respectively. However, in Fig. 10b, the accuracy of the *AB* model is lower than that of the *RS*, and computation takes up to 5800 s. Moreover, *BG* also consumes more time to execute, up to 24,000 s; however, in this case, accuracy is 98% achieved.

Scenario 3: Performance over Balanced Classes using Weighted Mean



(a) Regular (b) Randomized Search (c) Grid Search

Fig. 11 Performance of Proposed Ensemble Model over Balanced Dataset using Weighted Mean



(a) Comparative accuracy over REG, RS, GS (b) Comparative accuracy over RS, GS

Fig. 12 Accuracy of the Proposed Ensemble Model Balanced Dataset using Weighted Mean

Figure 11 shows the performance of the proposed ensemble learning model over a class balanced dataset using the Weighted Mean approach. The performance of the *XGB* model is better for REG, up to 98.22%. However, the performance of *RF* and *XGB* is constant in all three scenarios. Moreover, it is also observed that the performance of the *AB* model in terms of recall is increased by 66–75% in the case of a randomized search. In addition, the tuning of hyperparameters has improved the model parameters to a great extent. The precision, recall, and F1-score of all models have been found to be the best for randomized search. The results of the proposed model are also better in grid search compared to REG; comparative performance is better in all cases except the *AB* model.

Figure 12 shows the accuracy of the proposed ensemble model over a balanced class dataset using the weighted mean. Figure 12a shows the accuracy of the proposed model over all the REG, RS, and GS; however, Fig. 12b is a comparative analysis between RS and GS of our proposed model.

A detailed description of accuracy is as follows:

- *Regular vs. Randomized Search*: The CV accuracy of *AB* is improved by the utilization of hyperparameters. A slight impact can be observed on the rest of the classification models.
- *Regular vs. Grid search*: A moderate effect of hyperparameters can be noticed on *AB*, however, had a very small impact on the performance of other models.

- *Randomized Search vs. Grid Search*: It is evident from Fig. 12 that *AB* and *ET* have given better accuracy on the hyperparameters obtained using grid search in comparison to randomized search. In comparison to randomized search, grid search takes much longer to find the hyperparameters for *AB*, whereas for *ET* it works the other way around. The accuracy of *BG* and *XGB* is nearly identical for both hyperparameter tuning techniques; however, randomized search takes much less time than grid search to obtain the hyperparameters. Similarly, the *XGB* accuracy is about the same for both procedures, although randomized search has an advantage in terms of time consumption.

Figures 13, 14 and 15 provide an illustration of the final result of the proposed approach in terms of CV accuracy. The results demonstrate that for the balanced dataset, the proposed ensemble model has significantly improved accuracy, but in some instances, time consumption is higher. The effectiveness of the proposed approach has also been assessed using training to test ratios of 70:30 and 80:20. The studies depicted in Figs. 13, 14 and 15 demonstrates that accuracy is approximately the same, i.e., up to 98.22%, with the exception of the *AB* model. In contrast to unbalanced dataset samples, the CV accuracy has significantly improved for class-balanced dataset samples.

The comparative analysis of blockchain transaction deanonymization is carried out using ensemble learning with Nerurkar et al. [41], as depicted by Fig. 16. In this result, Wt. Mean is assumed to be Weighted Mean. It is evident from the result that all the variations of the proposed ensemble model have achieved more accuracy than [41]. It is also clearly shown that the proposed model with the SMOTE achieved an accuracy of up to 98.45%; however, the Weighted Mean achieved an accuracy of up to 97.9%; Nerurkar et al. [41]

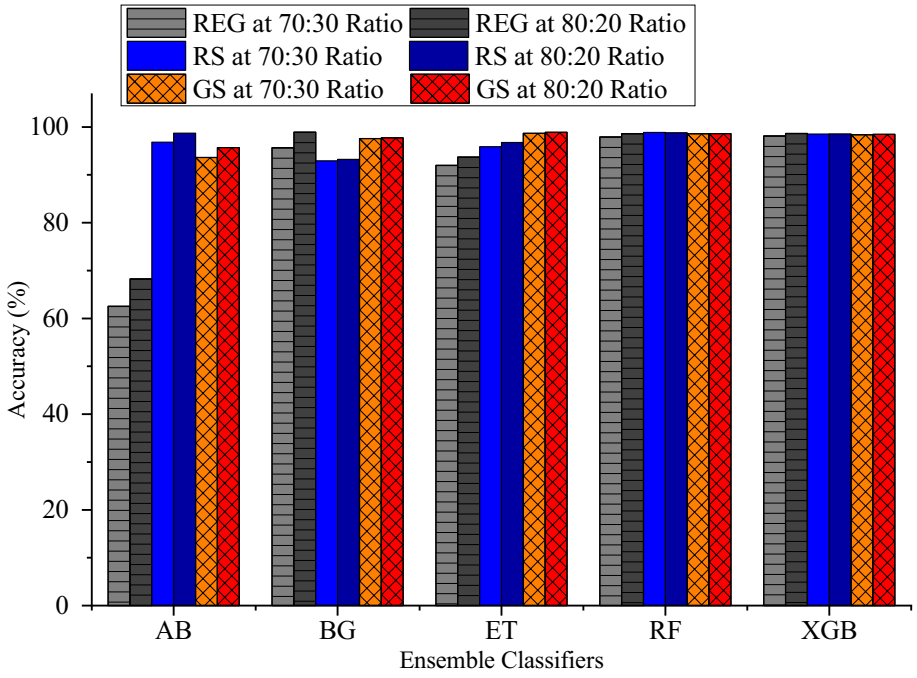


Fig. 13 Comparison of CV accuracy on Unbalanced Class Dataset

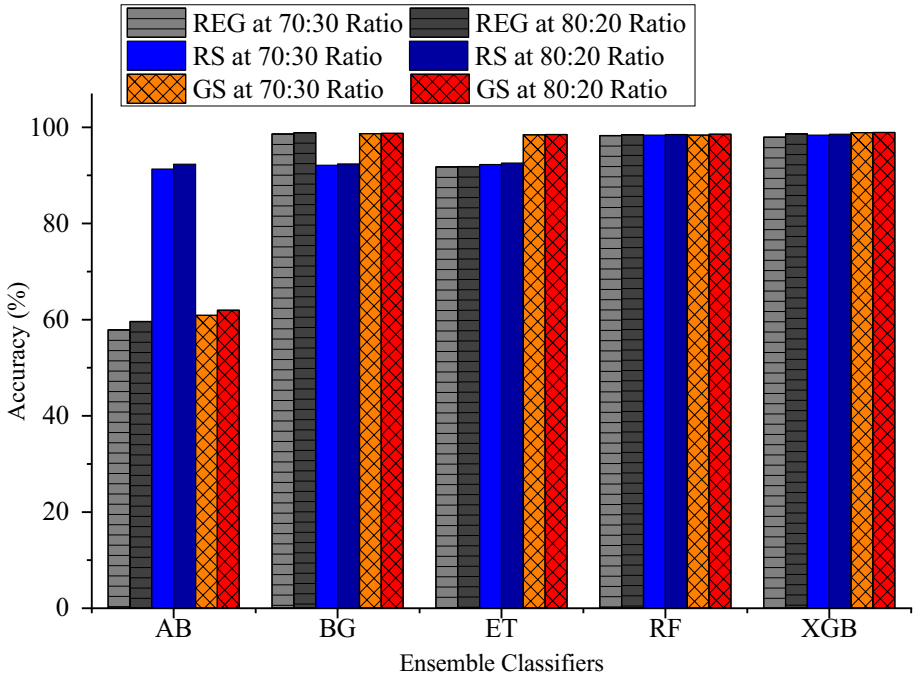


Fig. 14 Comparison of CV accuracy on Dataset Balanced using SMOTE

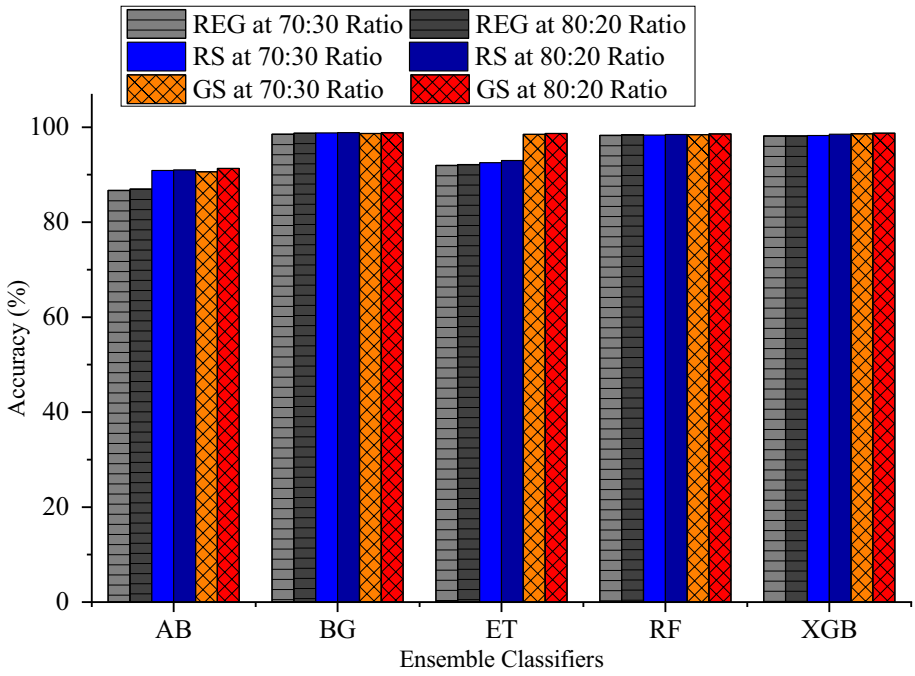


Fig. 15 Comparison of CV accuracy on Dataset Balanced using Weighted mean

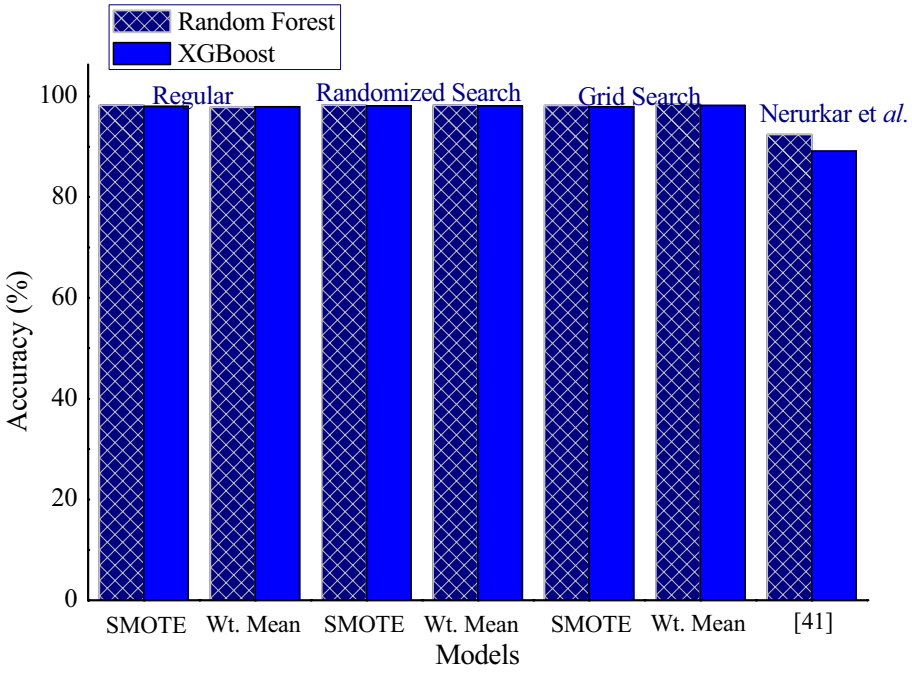


Fig. 16 Comparative analysis with the proposed ensemble model

achieved an accuracy of up to 92% and 89% using the same model with *Random Forest* and *XGBoost*, respectively. At the outset of the comparative analysis, it was found that the proposed ensemble learning approach is suitable for blockchain transaction deanonymization. We have also evaluated our proposed model in terms of the confusion matrix, which also proved the efficacy of our proposed ensemble learning for blockchain transaction deanonymization. In addition, it was also found that a heat map in the case of a randomised search is better than a grid search.

7 Conclusion and futures work

Ensemble learning is used in this paper to carry out the deanonymization of blockchain transactions. An extensive multi-class classification has been carried out to deanonymize transactions carried out over the Bitcoin blockchain. The dataset samples were scraped from the Blockchair and WalletExplorer repositories. An average cross-validation accuracy of 97.8% and an F1-score of 91% were achieved by using the proposed ensemble approach. Furthermore, the work is employed weighted mean and SMOTE with hyperparameter tuning, which has enhanced the classification performance, especially for adaptive boosting, regardless of the imbalanced or balanced nature of the classes in the dataset samples. The results also proved the efficacy of the proposed ensemble learning for deanonymizing blockchain transactions in terms of accuracy, up to 98.45%. The unavailability of labeled datasets is also addressed with different searches, SMOTE, and Weighted Mean.

In the future, in addition to gathering datasets from various sources and producing the labeled dataset, which can contain imbalanced classes, we will also recommend hybrid models and other strategies to enhance results. Federated learning may also have a future scope of privacy-preserving machine learning approaches in this application.

Author contributions The idea and problem formulation along with proposed solution, result analysis, and by corresponding author & supervisor, and verified by all other authors.

Funding The authors have not received funding from any of the sources.

Data availability The data set generated and/or analyzed during the current study is available upon reasonable request from the corresponding author. However, data sets are available as open source.

Declarations

Conflict of interest The work is not submitted in any other journal. There is no conflict of interest.

References

1. Nayyer N, Javaid N, Akbar Ma, Aldegheıshem A, Alrajeh N, Jamil M (2023) A new framework for fraud detection in Bitcoin transactions through Ensemble Stacking Model in Smart cities. IEEE Access 11:90916–90938. https://doi.org/10.1109/ACCESS.2023.3308298
2. Mundhe P, Phad P, Yuvaraj R et al (2023) Blockchain-based conditional privacy-preserving authentication scheme in VANETs. Multimed Tools Appl 82:24155–24179. https://doi.org/10.1007/s11042-022-14288-8
3. Nicholls J, Kuppa A, Le-Khac NA (2023) SoK: The next phase of identifying illicit activity in Bitcoin. In: Proc IEEE Int Conf Blockchain Cryptocurrency (ICBC), pp 1–10. https://doi.org/10.1109/ICBC56567.2023.10174963
4. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Available at SSRN 3440802. Accessed 11 Sept 2023
5. Bohme R, Christin N, Edelman B, Moore T (2015) Bitcoin: Economics, technology, and governance. J Economic Perspect 29(2):213–238. https://doi.org/10.1257/jep.29.2.213
6. Rahouti M, Xiong K, Ghani N (2018) Bitcoin concepts, threats, and machine-learning security solutions. IEEE Access 6:67189–67205. https://doi.org/10.1109/ACCESS.2018.2874539
7. Panda SK, Sathya AR, Das S (2023) Bitcoin: beginning of the Cryptocurrency era. In: Panda SK, Mishra V, Dash SP, Pani AK (eds) Recent advances in Blockchain Technology. Intelligent systems Reference Library, vol 237. Springer, Cham. https://doi.org/10.1007/978-3-031-22835-3_2
8. Christin N (2013) Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace. In: Proceedings of the 22nd International Conference on World Wide Web, pp 213–224. https://doi.org/10.1145/2488388.2488408
9. Hout MCV, Bingham T (2013) Silk Road', the virtual drug marketplace: a single case study of user experiences. Int J Drug Policy 24(5):385–391. https://doi.org/10.1016/j.drugpo.2013.01.005
10. Martin J (2014) Lost on the Silk Road: online drug distribution and the 'cryptomarket.' Criminol Criminal Justice 14(3):351–367. https://doi.org/10.1177/1748895813505234
11. Karlstrøm H (2014) Do libertarians dream of electric coins? The material embeddedness of Bitcoin. Distinktion: Scandinavian J Social Theory 15(1):23–36. https://doi.org/10.1080/1600910X.2013.870083
12. Nouman M, Qasim U, Nasir H, Almasoud A, Imran M, Javaid N (2023) Malicious Node Detection Using Machine Learning and Distributed Data Storage Using Blockchain in WSNs. IEEE Access 11:6106–6121. https://doi.org/10.1109/ACCESS.2023.3236983
13. Meiklejohn S, Pomarole M, Jordan G, Levchenko K, McCoy D, Voelker GM, Savage S (2016) A fistful of bitcoins: characterizing payments among men with no names. Commun ACM 59(4):86–93. https://doi.org/10.1145/2896384
14. Chaurasia BK, Verma S (2010) Maximising Anonymity of a Vehicle. In: International Journal of Autonomous and Adaptive Communications Systems (IJAACS), Special Issue on: Security, Trust, and

Privacy in DTN and Vehicular Communications, Inderscience 3(2):198–216. https://doi.org/10.1504/IJAACS.2010.031091https://doi.org/10.1080/07421222.2016.1205918
15. Samtani S, Chinn R, Chen H, Nunamaker JF Jr (2017) Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence. J Manage Inform Syst 34(4):1023–1053. https://doi.org/10.1080/07421222.2017.1394049
16. Andola N, Yadav VK, Venkatesan S, Verma S (2021) Anonymity on blockchain based e-cash protocols—A survey. Comput Sci Rev 40:100394–100411. https://doi.org/10.1016/j.cosrev.2021.100394
17. Andola N, Raghav, Yadav VK et al (2021) SpyChain: a Lightweight Blockchain for Authentication and Anonymous authorization in IoD. Wirel Pers Commun 119:343–362. https://doi.org/10.1007/s11277-021-08214-8
18. Beck R (2018) Beyond bitcoin: The rise of blockchain world. Computer 51(2):54–58
19. Abbasi A, Zahedi FM, Zeng D, Chen Y, Chen H, Nunamaker JF Jr (2015) Enhancing predictive analytics for anti-phishing by exploiting website genre information. J Manage Inform Syst 31(4):109–157. https://doi.org/10.1080/07421222.2014.1001260
20. Benjamin V, Zhang B, Nunamaker JF Jr, Chen H (2016) Examining hacker participation length in cybercriminal internet-relay-chat communities. J Manage Inform Syst 33(2):482–510
21. Abbasi A, Hsinchun C (2005) Applying authorship analysis to extremist-group web forum messages. IEEE Intell Syst 20(5):67–75. https://doi.org/10.1109/mis.2005.81
22. Beck R, Czepluch JS, Lollike N, Malone S (2016) Blockchain–the gateway to trust-free cryptographic transactions. In: Twenty-Fourth European Conference on Information Systems (ECIS), pp 1–14
23. Koshy P, Koshy D, McDaniel P (2014) An analysis of anonymity in bitcoin using p2p network traffic. In: Christin N, Safavi-Naini R (eds) Financial Cryptography and Data Security. FC 2014. Lecture Notes in Computer Science() 8437:469–485. https://doi.org/10.1007/978-3-662-45472-5_30
24. Androulaki E, Karame GO, Roeschlin M, Scherer T, Capkun S (2013) Evaluating user privacy in bitcoin In: Sadeghi AR (eds) Financial Cryptography and Data Security 7859: 34–51. https://doi.org/10.1007/978-3-642-39884-1_4
25. Bonneau J, Narayanan A, Miller A, Clark J, Kroll JA, Felten EW (2014) Mixcoin: Anonymity for bitcoin with accountable mixes. In: Christin, N., Safavi-Naini, R. (eds) Financial Cryptography and Data Security. FC 2014. Lecture Notes in Computer Science 8437: 486–504. https://doi.org/10.1007/978-3-662-45472-5_31
26. Misra G, Hazela B, Chaurasia BK (2013) Zero knowledge based authentication for internet of medical things. In: 14th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp 1–6
27. Meiklejohn S, Orlandi C (2015) Privacy-enhancing overlays in bitcoin. In: International Conference on Financial Cryptography and Data Security, pp 127–141.https://doi.org/10.1007/978-3-662-48051-9_10
28. Harlev MA, Sun Yin H, Langenheldt KC, Mukkamala R, Vatrapu R (2018) Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning. In: Proceedings of the 51st Hawaii International Conference on System Sciences, pp 3497–3506
29. Zola F, Eguimendia M, Bruse JL, Urrutia RO (2019) Cascading machine learning to attack bitcoin anonymity. In: IEEE International Conference on Blockchain (Blockchain), pp 10–17. https://doi.org/10.1109/Blockchain.2019.00011
30. Yin HHS, Langenheldt K, Harlev M, Mukkamala RR, Vatrapu R (2019) Regulating cryptocurrencies: a supervised machine Learning Approach to de-anonymizing the Bitcoin Blockchain. J Manage Inform Syst 36(1):37–73. https://doi.org/10.1080/07421222.2018.1550550
31. Lin YJ, Wu PW, Hsu CH, Tu IP, Liao SW (2019) An evaluation of bitcoin address classification based on transaction history summarization. In: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp 302–310. https://doi.org/10.1109/BLOC.2019.8751410
32. Lee C, Maharjan S, Ko K, Hong JWK (2020) Toward detecting illegal transactions on Bitcoin using machine-learning methods. In: Zheng Z, Dai HN, Tang M, Chen X (eds) Blockchain and Trustworthy systems. BlockSys 2019. Communications in Computer and Information Science, vol 1156. Springer, Singapore.https://doi.org/10.1007/978-981-15-2777-7_42
33. Li Y, Cai Y, Tian H, Xue G, Zheng Z (2020) Identifying Illicit addresses in Bitcoin Network. In: Zheng Z, Dai HN, Fu X, Chen B (eds) Blockchain and Trustworthy systems. BlockSys 2020, vol 1267. Springer, Singapore. https://doi.org/10.1007/978-981-15-9213-3_8
34. Liu T et al (2020) A new Bitcoin address Association Method using a two-level learner model. In: Wen S, Zomaya A, Yang LT et al (eds) Algorithms and architectures for parallel Processing. ICA3PP 2019, vol 11945. Springer, Cham. https://doi.org/10.1007/978-3-030-38961-1_31
35. Farrugia S, Ellul J, Azzopardi G (2020) Detection of illicit accounts over the Ethereum blockchain. Expert Syst Appl 150:113318. https://doi.org/10.1016/j.eswa.2020.113318

36. Michalski R, Dziubałtowska D, Macek P (2020) Revealing the character of nodes in a blockchain with supervised learning. IEEE Access 8:109639–109647. https://doi.org/10.1109/ACCESS.2020.3001676
37. Poursafaei F, Hamad GB, Zilic Z (2020) Detecting malicious Ethereum entities via application of machine learning classification. In: 2nd IEEE Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), pp 120–127
38. Kang C, Lee C, Ko K, Woo J, Hong JWK (2020) De-anonymization of the Bitcoin Network using address clustering. In: Zheng Z, Dai HN, Fu X, Chen B (eds) Blockchain and Trustworthy systems. BlockSys 2020, vol 1267. Springer, Singapore. https://doi.org/10.1007/978-981-15-9213-3_38
39. Ibrahim RF, Elian AM, Ababneh M (2021) Illicit account detection in the ethereum blockchain using machine learning. In: 2021 InternationaL Conference on Information Technology (ICIT), pp 488–493. https://doi.org/10.1109/ICIT52682.2021.9491653
40. Elbaghdadi A, Mezroui S, El Oualkadi A (2021) K-Nearest Neighbors Algorithm (KNN): An approach to detect illicit transaction in the bitcoin network. In: Azevedo A, Santos M (eds) Integration Challenges for Analytics, Business Intelligence, and Data Mining, (pp 161–178). IGI Global. https://doi.org/10.4018/978-1-7998-5781-5.ch008
41. Nerurkar P, Bhirud S, Patel D, Ludinard R, Busnel Y, Kumari S (2021) Supervised learning model for identifying illegal activities in Bitcoin. Appl Intell 51:3824–3843. https://doi.org/10.1007/s10489-020-02048-w
42. Jatoth C, Jain R, Fiore U, Chatharasupalli S (2022) Improved classification of Blockchain transactions using feature Engineering and Ensemble Learning. Future Internet 14(1):16. https://doi.org/10.3390/fi14010016
43. Nerurkar P (2023) Illegal activity detection on bitcoin transaction using deep learning. Soft Comput 27:5503–5520. https://doi.org/10.1007/s00500-022-07779-1
44. De Juan Fidalgo P, Cámara C, Peris-Lopez P (2023) Generation and Classification of Illicit Bitcoin Transactions. In: Bravo J, Ochoa S, Favela J (eds) Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022). UCAmI 2022. Lecture Notes in Networks and Systems, vol 594. Springer, Cham. https://doi.org/10.1007/978-3-031-21333-5_108
45. Sharma AK, Peelam MS, Chaurasia BK, Chamola V (2023) QIoTChain: Quantum IoT-blockchain fusion for advanced data protection in Industry 4.0. IET Blockchain published by John Wiley & Sons Ltd, pp 1–11.https://doi.org/10.1049/blc2.12059
46. Al-Hashedi KG et al (2023) A supervised model to detect suspicious activities in the bitcoin network. In: Al-Sharafi MA, Al-Emran M, Al-Kabi MN, Shaalan K (eds) Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems. ICETIS 2022. Lecture Notes in Networks and Systems, vol 584. Springer, Cham. https://doi.org/10.1007/978-3-031-25274-7_53
47. Umer Q, Li JW, Ashraf MR, Bashir RN, Ghous H (2023) Ensemble deep learning based prediction of fraudulent Cryptocurrency transactions. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3310576
48. Khalilov MCK, Levi A (2018) A survey on anonymity and privacy in bitcoin-like digital cash systems. IEEE Commun Surv Tutorials 20(3):2543–2585. https://doi.org/10.1109/COMST.2018.2818623
49. Reiter MK, Rubin AD (1998) Crowds: anonymity for Web transactions. ACM Transactions on Information and System Security (TISSEC) 1(1):66–92. https://doi.org/10.1145/290163.290168
50. Chaurasia BK, Verma S, Tomar GS (2013) Intersection attack on anonymity in VANET. In: Gavrilova ML, Tan CJK (eds) Transactions on Computational Science XVII, Springer-Verlag Berlin Heidelberg 7420:133–149. https://doi.org/10.1007/978-3-642-35840-1_7
51. Wu X, Bertino E (2007) An analysis study on Zone-based Anonymous Communication in Mobile Ad Hoc Networks. IEEE Trans Dependable Secure Comput 4(4):252–264. https://doi.org/10.1109/TDSC.2007.70213
52. Froomkin AM (1995) Anonymity and its enmities. Journal of Online Law, Online available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2715621. Accessed 15 May 2023
53. Froomkin AM (1999) Legal issues in anonymity and pseudonymity. Inform Soc 15(2):113–127. https://doi.org/10.1080/019722499128574
54. Cui J, Huang C, Meng H, Wei R (2023) Tor network anonymity evaluation based on node anonymity. Cybersecurity 6(55):1–16. https://doi.org/10.1186/s42400-023-00191-8
55. Zhang W, Lu T, Du Z (2021) TNRAS: Tor nodes reliability analysis scheme. In: The 11th International Conference on Communication and Network Security, pp 21–26
56. Schnitzler T, Pöpper C, Dürmuth M, Kohls K (2021) We built this circuit: Exploring threat vectors in circuit establishment in Tor. In: 2021 IEEE European Symposium on Security and Privacy (EuroS&P), pp 319–336
57. Mienye ID, Sun Y (2022) A survey of ensemble learning: concepts, algorithms, applications, and prospects. IEEE Access 10:99129–99149. https://doi.org/10.1109/ACCESS.2022.3207287

58. Chaurasia BK, Raj H, Rathour SS, Singh PB (2023) Transfer learning driven ensemble model for detection of diabetic retinopathy disease. In Medical, Biological Engineering and Computing, Springer 61:2033–2049. https://doi.org/10.1007/s11517-023-02863-6
59. Zhou ZH (2012) Ensemble methods: foundations and algorithms, 1st edn. Chapman and Hall/CRC. https://doi.org/10.1201/b12207
60. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: ICML 96:148–156. Online Available at: https://cseweb.ucsd.edu/~yfreund/papers/boostingexperiments.pdf. Accessed 17 Sept 2023
61. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J (2011) Scikit-learn: Machine learning in Python. In: The Journal of machine Learning research 12:2825–2830. Online Available at: https://jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf. Accessed 17 Sept 2023
62. Merkle RC (2019) Protocols for public key cryptosystems. IEEE Symposium on Security and Privacy, pp 122–134. https://doi.org/10.1109/SP.1980.10006
63. Andrychowicz M, Dziembowski S, Malinowski D, Mazurek Ł (2015) On the Malleability of Bitcoin Transactions. In: Brenner M, Christin N, Johnson B, Rohloff K (eds) Financial Cryptography and Data Security. Lecture Notes in Computer Science 8976: 1–18. https://doi.org/10.1007/978-3-662-48051-9_1
64. Blockchair Database, Online available at: https://gz.blockchair.com/bitcoin/transactions. Accessed 29 Mar 2023
65. WalletExplorer, Online available at: https://www.walletexplorer.com/. Accessed 30 Mar 2023
66. Beautiful Soup, Online available at: https://www.browserstack.com/guide/web-scraping-using-beautiful-soup. Accessed 30 Mar 2023
67. LabelEncoder, Online available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html. Accessed 23 Apr 2023
68. Singh D, Singh B (2020) Investigating the impact of data normalization on classification performance. Appl Soft Comput 97:105524
69. García S, Luengo J, Herrera F (2015) Data preprocessing in data mining, vol 72. Springer International Publishing, Cham, Switzerland, pp 59–139
70. Han J, Pei, J, Tong, H (2012) Data mining: concepts and techniques. Morgan Kaufmann. https://doi.org/10.1016/C2009-0-61819-5
71. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 1(16):321–357. https://doi.org/10.1613/jair.953
72. Saxena R, Arora D, Nagar V (2023) Classifying blockchain cybercriminal transactions using hyperparameter tuned supervised machine learning models. Int J Comput Sci Eng 26(6):615–626. https://doi.org/10.1504/IJCSE.2022.10056854
73. Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsl 6(1):20–29. https://doi.org/10.1145/1007730.1007735
74. SMOTE Module, Online available at: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html. Accessed on 23/04/2023
75. RandomOverSampler, Online available at: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html. Accessed 23/04/2023
76. RandomUnderSampler, Online available at https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html. Accessed 23/04/2023
77. Schratz P, Muenchow J, Iturritxa E, Richter J, Brenning A (2019) Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. Ecol Model 406:109–120. https://doi.org/10.1016/j.ecolmodel.2019.06.002
78. Probst P, Wright MN, Boulesteix AL (2019) Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: Data Min Knowl Discovery 9(3):1–15. https://doi.org/10.1002/widm.1301
79. Zhang L, Zhan C (2017) Machine learning in rock facies classification: an application of XGBoost. In: International Geophysical Conference on Society of Exploration Geophysicists and Chinese Petroleum Society, pp 1371–1374. https://doi.org/10.1190/IGC2017-351
80. Bajpai S, Sharma K, Chaurasia BK (2023) Intrusion detection Framework in IoT Networks. Springer Nature Computer Science Journal. Special Issue Mach Learn Smart Syst 4(350):1–17. https://doi.org/10.1007/s42979-023-01770-9
81. Liashchynskyi P, Liashchynskyi P (2019) Grid search, random search, genetic algorithm: a big comparison for NAS. Online available at https://arxiv.org/pdf/1912.06059.pdf. Accessed 12/02/2023
82. Putatunda S, Rama K (2018) A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost. In: Proceedings of the 2018 International Conference on Signal Processing and Machine Learning, pp 6–10. https://doi.org/10.1145/3297067.3297080

83. RandomizedSearchCV Online available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html. Accessed 23/04/2023
84. Syarif I, Prugel-Bennett A, Wills G (2016) SVM parameter optimization using grid search and genetic algorithm to improve classification performance. (TELKOMNIKA) Telecommun Comput Electronics Control 14(4): 1502–1509. https://doi.org/10.12928/TELKOMNIKA.v14i4.3956
85. Ataei M, Osanloo M (2004) Using a combination of genetic algorithm and the Grid Search Method to Determine Optimum Cutoff grades of multiple metal deposits. Int J Surf Min Reclam Environ 18(1):60–78. https://doi.org/10.1076/ijsm.18.1.60.23543
86. Xiao T, Ren D, Lei S, Zhang J, Liu X (2014) Based on grid-search and PSO parameter optimization for Support Vector Machine. In: Proceeding of the 11th World Congress on Intelligent Control and Automation, pp 1529–1533. https://doi.org/10.1109/WCICA.2014.7052946
87. GridSearchCV Online available at https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. Accessed 23/04/2023
88. Aversana PD (2019) Comparison of different machine learning algorithms for lithofacies classification from well logs. Bollettino Di Geofis Teorica Ed Appl 60(1):69–80. https://doi.org/10.4430/bgta0256