future internet

MDPI

Article

Fraud Detection in Cryptocurrency Networks—An Exploration Using Anomaly Detection and Heterogeneous Graph Transformers

Víctor Pérez-Cano and Francisco Jurado *

Department of Computer Engineering, Universidad Autónoma de Madrid, 28049 Madrid, Spain; victor.perezcano@estudiante.uam.es

* Correspondence: francisco.jurado@uam.es

Abstract: Blockchains are the backbone behind cryptocurrency networks, which have developed rapidly in the last two decades. However, this growth has brought several challenges due to the features of these networks, specifically anonymity and decentralization. One of these challenges is the fight against fraudulent activities performed in these networks, which, among other things, involve financial schemes, phishing attacks or money laundering. This article will address the problem of identifying fraud cases among a large set of transactions extracted from the Bitcoin network. More specifically, our study's goal was to find reliable techniques to label Bitcoin transactions, taking into account their features. The approach followed involved two kinds of Machine Learning methods. On the one hand, anomaly detection algorithms were applied to determine whether fraudulent activities tend to show anomalous behaviour without resorting to manually obtained labels. On the other hand, Heterogeneous Graph Transformers were used to leverage the heterogeneous relational nature of the cryptocurrency information. As a result, the article will provide reasonable conclusions to acknowledge that unsupervised approaches can be useful for fraud detection on blockchain networks. Furthermore, the effectiveness of supervised graph methods was revalidated, emphasizing the importance of data heterogeneity.

Keywords: blockchain; cryptocurrency; fraudulent activity detection; machine learning

Academic Editors: Christoph Stach, Clémentine Gritti and Iouliana Litou

Received: 20 November 2024
Revised: 10 January 2025
Accepted: 17 January 2025
Published: 19 January 2025

Citation: Pérez-Cano, V.; Jurado, F. Fraud Detection in Cryptocurrency Networks—An Exploration Using Anomaly Detection and Heterogeneous Graph Transformers. *Future Internet* **2025**, *17*, 44. https://doi.org/10.3390/fi17010044

1. Introduction

Cryptocurrency networks such as Bitcoin or Ethereum are extensively used due to their features and differences from traditional money. Although these networks are considerably robust from the perspective of a distributed system thanks to being supported by the blockchain algorithm, cryptocurrencies face a challenge: the commission of fraudulent activities in these networks, mainly because of the lack of a regulatory entity [1,2].

Significant progress has already been made in the context of fraud detection, especially featuring the use of Machine Learning (ML) methods that can process large amounts of transaction data produced by cryptocurrency networks and use them to make predictions about their illicitness. To cite a few of the most recent works, we can highlight the use of traditional ML approaches like Naive Bayes, SVMs, Logistic Regression, Gradient Boosting, Ada-Boost, Random Forests and other Ensemble Learning models [3–8], but also Federation Learning [9], neural network approaches [10–12], anomaly detection [13] and more recent graph-based approaches like those that consider the relational nature of transactional data, like the Hybrid Graph Neural Network [14], graphs based on a Transformer Network [15],

the subgraph-based contrastive learning algorithm for heterogeneous graphs [16] and graph anomaly detection [17].

Creating better and bigger datasets and developing more sophisticated deep learning models opens up new possibilities in the context of cryptocurrency fraud detection. However, all these supervised learning methods share a common problem: the need for a large enough labelled dataset to train them. From another perspective, in the absence of a manually labelled dataset, using unsupervised learning methods that can extract information about the data is another research field that, as far as we know, has not been extensively explored for the case of cryptocurrency fraud detection. In this context, this work aimed to revalidate the effectiveness of supervised learning methods on a representative Bitcoin transaction dataset by testing the Heterogeneous Graph Transformer (HGT) [18], a novel framework aimed at leveraging relational data with a heterogeneous nature. In addition, this study also explored the possibilities of different unsupervised learning methods based on anomaly detection through the use of k-means clustering and several feature-based outlier detection algorithms, such as the Local Outlier Factor (LOF) [19]. The obtained results revealed that the performance achieved with non-supervised methods was still far from that of one of the supervised approaches, even though they represent a considerable alternative due to not needing a manually labelled dataset.

Hence, the main contributions of our research can be summarized in the following points: (1) we provide a comprehensive evaluation of unsupervised and supervised methods using real-world datasets, discussing their limitations and potential improvements; (2) explore the potential of feature-based anomaly detection methods in the context of cryptocurrency networks and, as a result, propose a hybrid approach combining structural and intrinsic data features; and (3) put to the test a novel graph framework that leverages the heterogeneous nature of transactional data.

The structure of the rest of this article will be as follows: Section 2 will introduce the background on cryptocurrency fraud and describe the current literature about the topic. Then, Section 3 will present the related works. Later, Section 4 will formally describe the problem, the datasets and the methodology. Following this, Section 5 will detail the different experiments carried out, grouped into four categories: exploratory data analysis, anomaly detection, clustering and graph deep learning. Finally, Section 6 will draw conclusions, summarize the results and provide suggestions for future works.

2. Background on Cryptocurrency Fraud

In brief, cryptocurrency technologies are supported by the blockchain algorithm, which permits the creation of a distributed ledger. This ledger is structured as a sequence of blocks that are cryptographically linked, ensuring that the information stored on them remains immutable, as shown in Figure 1. Each block in the chain contains a set of transactions (tx_n), a timestamp and a cryptographic hash of the previous block ($block_{i-1}$), creating a secure and chronological record. This structure not only protects the integrity of the data but also facilitates trust among users, as they can independently verify transactions without relying on a central authority.

The blockchain operates on a decentralized network, meaning that no single entity has control over the entire system. This decentralization enhances security and transparency, as every participant in the network has access to the same information, making it exceedingly difficult for any malicious actor to alter the data without consensus from the majority.

However, the lack of a regulatory entity and the nature of these distributed systems, which involve a range of properties such as anonymity and decentralization, make these networks an ideal context for cybercriminals to act in.

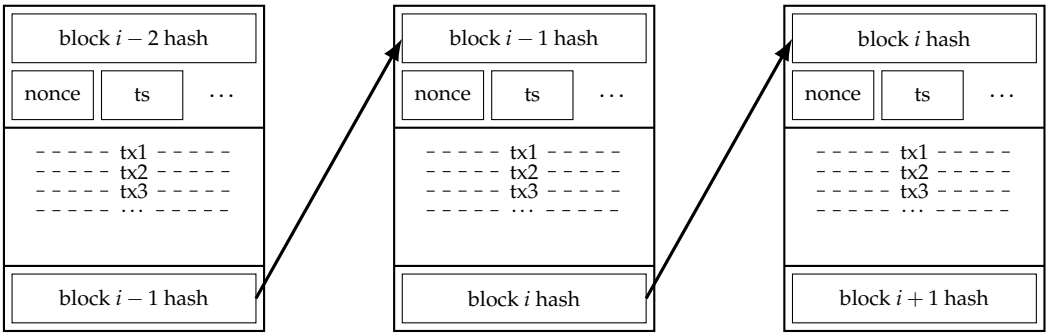


Figure 1. Example blockchain sequence.

According to recent studies conducted by the European Parliament [20] and Europol [21], some of the most prevalent types of fraud occurring within cryptocurrency networks include the following:

- Financial schemes. There are a variety of types of fraud that can be categorized in this group. Some examples include the following:
  - Pump and Dump. This type of fraud consists of a market manipulation technique involving speculation with a cryptocurrency derivative, in which the fraudster "pumps" up the price of the product with the intention of eventually selling all their holdings, consequently "dumping" the price of the derivative.
  - Ponzi Scheme. This is a pyramid scheme in which earlier investors are rewarded with the funds of the newer ones, typically attracted by the potential growth promised by the fraudsters.
  - Rug Pulls. This is a type of fraud in which victims are encouraged to invest in a promising Non-Fungible Token (NFT) or other token-like project, which is then abandoned by the developers, who quickly disappear, leaving the victims with useless assets.
- Phishing attacks. This is a type of fraud where the fraudster fakes their identity, for example, by impersonating a prestigious crypto-investor or creating a fake landing page for a well-known exchange portal.
- As a means of payment. Although this does not involve fraud by itself, cryptocurrencies tend to be the way to go for purchasing illegal goods or services on the dark net, given the lack of oversight and control of state authorities on this part of the web. In this manner, cryptocurrencies are used in markets associated with drugs, arms, sexual abuse or hitman services. Additionally, cryptocurrencies are often requested as payment for multiple kinds of cyberattacks, such as those using ransomware, due to their complicated traceability. As shown in the next section, remarkable progress has been achieved on this specific topic.
- Money laundering. Although this may not be considered fraud, it typically follows some criminal activity. The anonymity that cryptocurrency wallets offer is ideal for those who want the traceability of their money to be lost. However, it is important to note that cryptocurrency wallets are not always completely anonymous, but offer so-called pseudo-anonymity. When cryptocurrencies are purchased with traditional, traceable money, the cash flow can be traced back to its origin, therefore retrieving the purchaser's identity. To avoid this, special services known as mixers are available online, where users can send their money to be mixed with that of one of the other users before being sent to an untraceable output wallet. These services tend to have a sophisticated infrastructure and charge fees to their clients. Some popular examples available on the Bitcoin network are CoinJoin and JoinMarket. Some modern mixers are so-called non-custodial mixers, which means no entity has custody of the users'

money at any time, with the logic of the mixer being handled by a smart contract on the correspondent network. An example is Tornado Cash on the Ethereum network.

When compared to normal transaction traffic on blockchain networks, different types of anomalies can be found on these networks beyond fraud that might occur because of network issues or delays without malicious intent, as detailed in works like [22–26]. As those works reveal, some of the main anomalies are the following:

- Consensus problems. These include situations that might affect the blockchain consensus mechanism, such as network partitioning, in which the blockchain is temporarily split into different forks, or vulnerabilities in the Proof of Work (PoW) algorithm.
- Double-spending attempts. These occur when the same digital asset is spent more than once.
- Smart contract vulnerabilities. These are flaws in the smart contract code that could lead to unintended behaviours or security risks.
- Atypical transaction patterns. These include sudden spikes in the transaction volume or irregular transaction sizes, which might indicate underlying issues.

Nevertheless, to the best of our knowledge, there is no approach that identifies fraud as an anomaly, and this is the issue we would like to address in this article.

Now that the main concepts of cryptocurrency fraud and blockchain anomalies have been briefly illustrated, the next section will detail related works to address the problem.

3. Related Works

As the cryptocurrency community has grown in recent years, so too has the number of fraud cases and types occurring on these networks. Several research works have attempted to extract useful information to identify the patterns of them. The most relevant investigations on this topic have proposed frameworks for modelling a cryptocurrency network and its transaction flow to capture important information such as temporal or structural features. These data can then be used along with different methods for a variety of tasks within the fraud detection context.

Earlier works showed the performance of traditional ML models applied to a classification task on transaction datasets extracted from the blockchain and manually labelled, such as Naive Bayes, SVMs, Logistic Regression, Gradient Boosting, Ada-Boost, Random Forests [3–8], Federation Learning [9], Long Short-Term Memory [10], Recurrent Neural Networks [11] or anomaly detection [13].

However, recent investigations have attempted to outperform the metrics obtained with these models by involving additional features that represent the structural topology of the transaction graph. Most of these works have made use of a variation of a Graph Neural Network [12,27], which is a deep learning model that can encode the underlying structure of a graph by combining a node's representation with that of one of its neighbours in a process known as message passing, which can consist of multiple layers, therefore combining the information of multiple degrees of the neighbourhood.

In this context, Di et al. [28] suggested a framework for modelling Bitcoin transactions as a random graph to exploit their structural properties and analyze them from the perspective of Graph Theory. This idea can be fitted in different ways to the available data and the models to be used. For example, wallets could be represented as graph nodes, with the edges being the transactions between them, as proposed by Tharani et al. [29]. For their part, Weber et al. [30] proposed a different framework in which transactions are taken as nodes themselves while edges represent the money flow between transactions. This approach turned out to be more suitable for the goal of mining data and especially for the task of identifying fraud patterns. More sophisticated works, like those performed

by Lin et al. [31], suggested advanced dynamic models that also take into account the temporal features of the data.

Some of the most popular types of message-passing layers are the ones implemented in Graph Convolutional Networks (GCNs) [32] and Graph Attention Networks (GATs) [33]. Recent investigations have applied this kind of framework specifically to classify transactions as licit and illicit ones and, in most cases, no improvement has been obtained by the graph networks themselves. However, these message-passing networks can be used to build high-dimensional embeddings that, concatenated to the nodes' original features, can improve the metrics obtained with more traditional methods such as Random Forests, as shown in [30]. Furthermore, newer studies have proposed novel frameworks, such as the Heterogeneous Graph Transformer (HGT) [18], that have not been tested in this specific context yet.

Leaving aside general fraud detection, numerous studies have been carried out with the objective of identifying specific kinds of fraud on cryptocurrency networks. This is the case in [14,15], which made use of novel data analytics techniques to identify phishing scams on different networks. In [14], a Data Augmentation Method and Hybrid Graph Neural Network Model was proposed to address sample imbalance and feature extraction issues, achieving impressive performance metrics on a real dataset. For their part, Choi and Buu [15] used a Deep Graph Traversal based on a Transformer for Scam Detection to address the complexities of Ethereum's transaction networks to detect phishing scams, outperforming traditional methods.

Other type-specific fraud detection studies have included ransomware activity tracking, mostly on the Bitcoin network, where these types of scams are commonly performed. Among these studies, the work of Akcora et al. [34] stands out due to their successful application of advanced topological blockchain graph analysis on a finely constructed dataset with different ransomware families. Other studies related to ransomware detection include [35,36].

From another perspective, a frequently unexplored approach to this topic is the use of unsupervised methods, which rely on the idea of making predictions based exclusively on the features of the data and not on the labels. The advantage of not using labels is especially important in this case since the labelling process is a complicated, costly and time-consuming process that can be skipped when following this strategy. Some of these approaches include clustering techniques such as the k-means or outlier detection methods such as the Local Outlier Factor (LOF) [19]. Most of them are traditional and well-established algorithms that have been proven to perform well in a wide range of applications and so could perform similarly in the context of this study. In fact, various works have already employed some of these techniques for anomaly detection, such as [25,37,38] with the use of an Unsupervised SVM or the Mahalanobis Distance, among others. In addition, the study by Shayegan et al. [39] proposed methods for detecting user-level anomalies by aggregating the data of multiple digital wallets owned by the same user, with the aim of identifying if their overall activity is anomalous instead of assessing individual addresses.

In addition to feature-based approaches, researchers have recently proposed models for structural anomaly detection, specifically adapted to graph data. One example is the DOMINANT [40] framework, which makes use of autoencoders [41] to combine both structural and feature-related anomalies. Applied to cryptocurrency fraud detection, the work detailed in [17] proposed the use of the Graph Convolutional Autoencoder model to enhance the detection of anomalies in cryptocurrency transactions by treating transactions as edges and accounts as nodes within a graph neural network.

4. Problem Statement and Methodology Definition

This section outlines the experiments conducted in this research, which focused on a binary classification task involving cryptocurrency transactions, using a dataset with a fixed number of features and evaluating the model's performance.

4.1. Problem Statement

In general terms, the experiments conducted in this research aimed to carry out a binary classification task on a cryptocurrency transaction (or wallet) set. More specifically, given a set of transactions (or wallets), V, with a fixed number of features each, the objective was to build a model to decide whether a given element, $v \in V$, belonged to the licit subset of transactions or the illicit one. To compute the performance of each model, it was tested against a partially labelled dataset, which will be described in the next subsection. The approaches used in this research to build such a model followed different strategies.

On the one hand, a series of experiments using non-supervised methods were conducted to contrast them and explore the following hypothesis:

Hypothesis 1. *Generally, fraudulent activities tend to show anomalous behaviour in contrast to normal licit activities; therefore, illicit transactions could be detected by identifying unusual and abnormal patterns in the data, without resorting to manually obtained labels.*

The objective was to test several traditional and established feature-based outlier detection and clustering algorithms and to deepen the concept of anomalies in the context of fraud detection to find aspects that might improve the understanding of the relation between abnormal transactions and illicit ones.

On the other hand, a more traditional supervised approach was used. For this phase, transaction data were modelled as a graph, following the ideas of related research with state-of-the-art results. With the aim of outperforming those results, the novel HGT framework was tested, trying to leverage the heterogeneous nature of transaction data.

4.2. The Datasets and the Graph Nature of the Data

The data-gathering step is a particularly complicated task in the context of cryptocurrency transactions, even though raw transaction data are publicly available through different blockchain APIs. One of the main reasons for this is the difficulty of manually labelling transaction data, since this task requires exhaustive research to determine whether a given transaction is licit or not. Because of this, most available datasets are partially labelled. Another important factor is the great volume of transactions that are continuously being processed and the volatility of these data since they depend on different variables that are constantly changing, such as the price of the cryptocurrency on the market. Consequently, throughout this section, the different datasets used in the experimentation process of this research will be described, along with the handling of the data modelling.

As mentioned previously, transaction data have the peculiarity of carrying underlying relational information that can be extracted and utilized by classification models to make more accurate predictions. To achieve this, the dataset needs to be modelled as a graph, and the most suitable way of doing this is by modelling the dataset as a homogeneous graph, $\mathcal{G} = (V, E)$, where V is the transaction set and $E \subseteq V \times V$ is the set of edges so that $e = (s, t)$ represents the flow of cryptocurrencies or other assets from transaction $s \in V$ to transaction $t \in V$. This approach was utilized by researchers in [30] and applied to the Elliptic Bitcoin Dataset, a dataset with more than 200,000 Bitcoin transactions (approximately USD 6 billion) published by the Elliptic company in the same research work. Each of the transactions in this dataset is labelled as either licit (0), illicit (1) or unknown (2) and has 166 features. The first 94 features represent local information about the transaction, such

as the number of inputs and outputs, the transaction fee and the output volume, while the other 72 features are obtained by aggregating the local ones. Further details on the feature extraction process are not available due to the intellectual property policies of the Elliptic company. Following the previously explained mechanism, the dataset also contains a list of approximately 230,000 edges representing the Bitcoin Unspent Transaction Output (UTXO) flow between transactions. The dataset is described as being constructed from a set of transactions explicitly known to be licit and illicit and then extended with random transactions broadcast to the blockchain in evenly spaced time windows. Thus, one of the features of the transactions is a discrete temporal component ranging from 1 to 49, referred to as the timestep. For this reason, we believe that the Elliptic dataset constitutes a reliable representation of the Bitcoin transaction flow, and including different fraud categories in the illicit label made it ideal for the purpose of this study. Therefore, this dataset was widely used in a good part of the experiments in our work.

Additionally, some popular datasets might also include features for both transactions and wallets, as well as the relations between them. In these cases, the transaction graph can be modelled as a heterogeneous graph, $\mathcal{G} = (V_{tx}, V_w, E_{tx}^w, E_w^{tx})$, in which V_{tx} is the set of transactions (with its transaction-specific features), V_w is the set of wallets (with its wallet-specific features) and $E_{tx}^w \subseteq V_{tx} \times V_w$ and $E_w^{tx} \subseteq V_w \times V_{tx}$ are the sets of the `Transaction`→`Wallet` and `Wallet`→`Transaction` edges, respectively. This is the case for the Elliptic++ [12] dataset, presented in 2023 as an improved extension of the Elliptic dataset, containing a set of around 800,000 wallets with 55 features each and the relations between them as actors in the transactions of the original dataset. Given the heterogeneous nature of this dataset, it was also used for another part of the experiments in this research.

A relevant aspect of both datasets is that the number of transactions (and wallets) labelled as illicit in relation to the licit ones is highly imbalanced, which is a fact that had to be taken into account during the experimentation process.

4.3. Methodology

The experimentation phase of this research was carried out using popular Python libraries such as Scikit-Learn for common ML algorithms and Pytorch for tensor handling and more advanced deep learning algorithms, in conjunction with Pytorch Geometric (PyG) for the implementation of GNNs and other graph-related applications.

Due to the high volume of the data and the complexity of the utilized models, specifically the GNNs, it was necessary to apply a mini-batch training technique to avoid Out Of Memory (OOM) errors. To split the transactions graph into separate batches without losing structural information, it was utilized as a unit for batching the subgraph corresponding to each timestep, since the transactions from different timesteps were not linked to each other. Formally, each subgraph can be defined in the following manner:

$$G_t = (V_t, E_t) \quad \text{where} \quad \begin{cases} V_t = \{v \in V \mid \text{TIMESTEP}(v) = t\} \\ \text{and} \\ E_t = \{(s,t) \in E \mid s,t \in V_t\} \end{cases} \qquad t = 1, \ldots, 49 \qquad (1)$$

Then, the G_ts could be grouped to form training batches of a specific size.

Concerning benchmarking and the evaluation of the models' performance, selected metrics were chosen to account for the imbalance of the data. Thus, the Accuracy metric was discarded since it only measures the number of successful predictions, which can be very high for the licit class but low for the illicit class. Instead, other evaluation functions such as the Precision, Recall and F1 were selected, since they penalize the amount of false positives, false negatives and both, respectively. Additionally, for the sake of displaying an

accurate picture of the predictions performed by the models, a confusion matrix is provided for each experiment.

Finally, as a complementary measure, the ROC curve graphic is also provided for each experiment, showing the proportion of true positives to false positives as the decision threshold grew. This threshold could be a probability in the case of supervised learning models or an outlier factor in the case of anomaly detection experiments. The Area Under the Curve (AUC) is also provided as another metric.

5. Experimentation Process

The collection of experiments carried out in this research can be grouped into four categories:

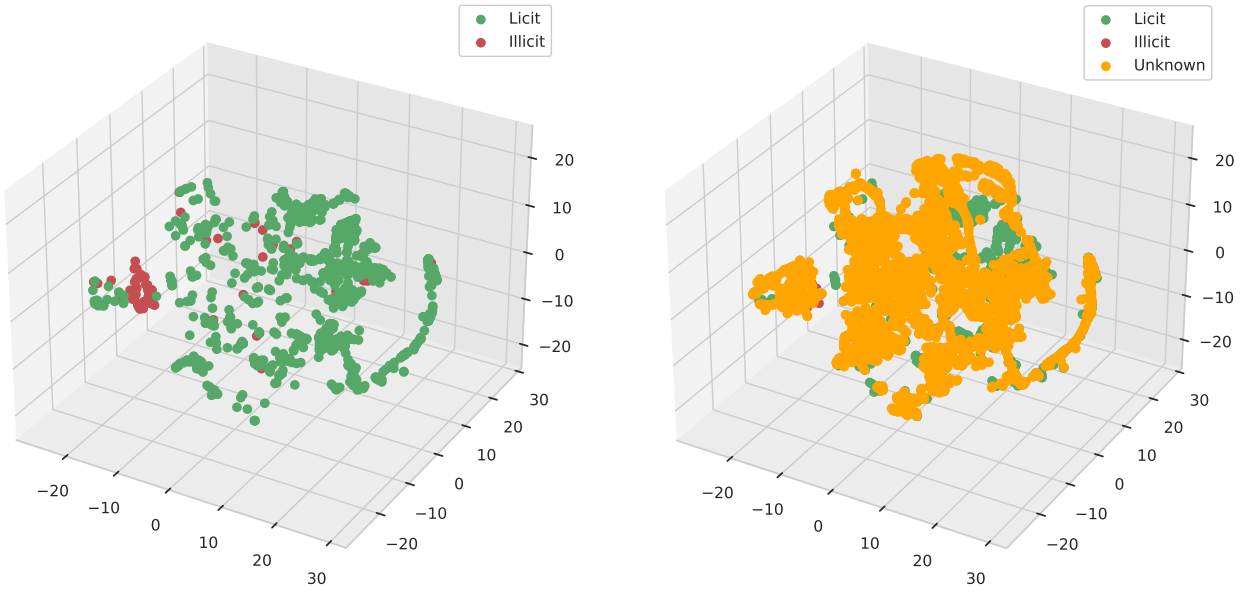
1. First, an exploratory data analysis (EDA) was conducted, including several tasks such as a reduced component visualization of the dataset, a class imbalance analysis and a graph subsample visualization.
2. Second, a set of anomaly detection experiments were carried out using different popular algorithms to find the relation between the fraudulent transactions and the anomalies, as stated in the hypothesis that was described in detail at the beginning of this section.
3. Third, clustering was conducted using the k-means algorithm on the transactions dataset, employing a non-supervised approach to identify correlations between the transaction labels and the resulting clusters.
4. Lastly, a modern GNN algorithm was tested on the full Elliptic++ dataset, this time following a supervised methodology, which has traditionally been used more in this context. Specifically, the deep learning framework used was the Heterogeneous Graph Transformer (HGT) and aimed to leverage the heterogeneous nature of the Elliptic++ dataset to reach or even outperform the benchmarks obtained with other deep learning algorithms already tested on this dataset.

5.1. Exploratory Data Analysis

This subsection will introduce several graphs and diagrams that illustrate the nature of the Elliptic and Elliptic++ datasets to extract useful information about their features, the class imbalance and the relational structure.

An important thing to consider before diving into the experiments is the high dimensionality of the data being worked with. Specifically, the transactions dataset had a total of 183 features. Dealing effectively with this number of features was crucial for both the experiments and the visualization of the data, due to the noise they can produce. In later sections, dimensionality reduction methods oriented to experimentation, such as PCA, will be treated in detail. However, for the sake of a clear visualization of the data, other methods tend to be more suitable, such as t-Distributed Stochastic Neighbour Embedding (t-SNE). In this manner, a precise representation of the original data can be obtained in a reduced two- or three-dimensional space that can be represented on a concise plot.

Accordingly, Figure 2 shows the output of the t-SNE algorithm applied in a three-dimensional space to the subset of transactions that occurred at the seventh timestep. The first subfigure includes the transactions with a known label, namely licit (in green) or illicit (in red). The second one also includes the transactions with an unknown label (in orange). Both of them give a hint about the imbalance that exists between the different labels of the dataset.



(**a**) (**b**)

Figure 2. Reduced visualization of transactions at timestep 7. (**a**) Known label transactions; (**b**) known and unknown label transactions.

To take a closer look at the imbalance, Figure 3 shows the number of transactions of every label (licit, illicit and unknown) at each timestep of the dataset, from 1 to 49. A high imbalance between different classes can be observed in the figure, and it is also remarkable that this imbalance varied from timestep to timestep with high volatility. Additionally, Figure 4 shows the proportion of licit and illicit transactions at each timestep, confirming this tendency.

Equivalently, Figures 5 and 6 show that the set of wallets showed similar behaviour to that of the transactions set.

Using a probability theory approach, the mutual information shared between the label and a feature, treated as two random variables, could be estimated by sampling both from the dataset. This provided a measure of their dependency, allowing us to draw conclusions about the extent to which a feature influenced the class. In this manner, Figures 7 and 8 show the amount of estimated mutual information of each feature concerning the class (licit or illicit) for both the 183 features of the transactions and the 55 features of the wallets, respectively.

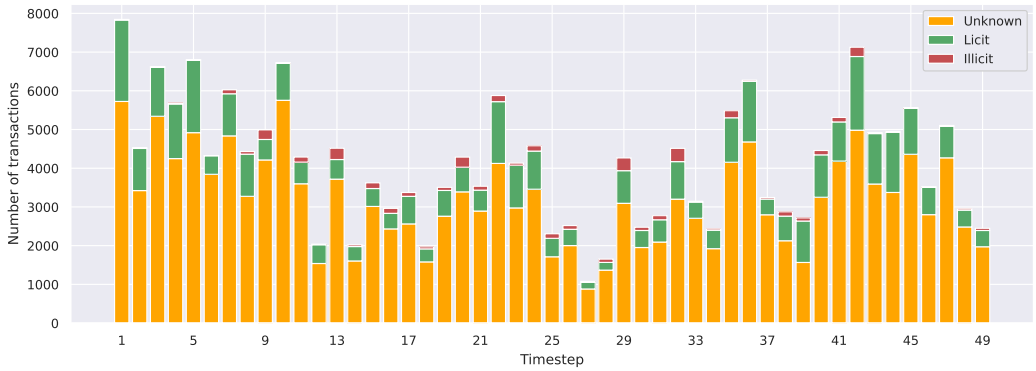


Figure 3. Number of transactions in each class per timestep.

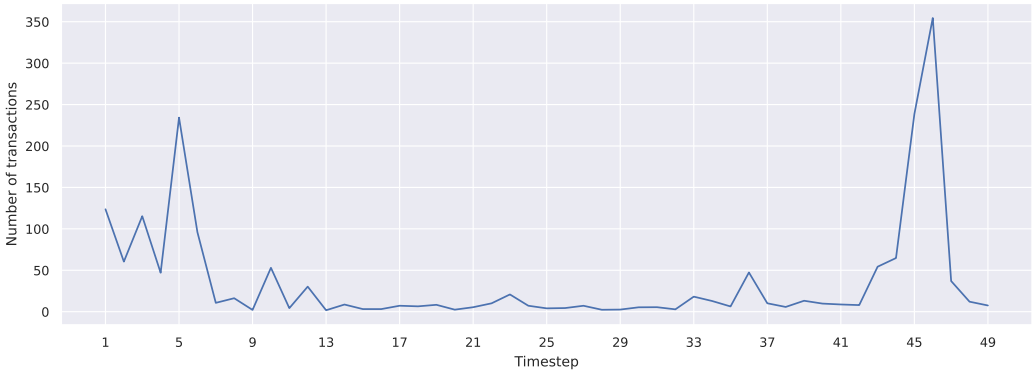


Figure 4. Imbalance between licit and illicit transactions per timestep.

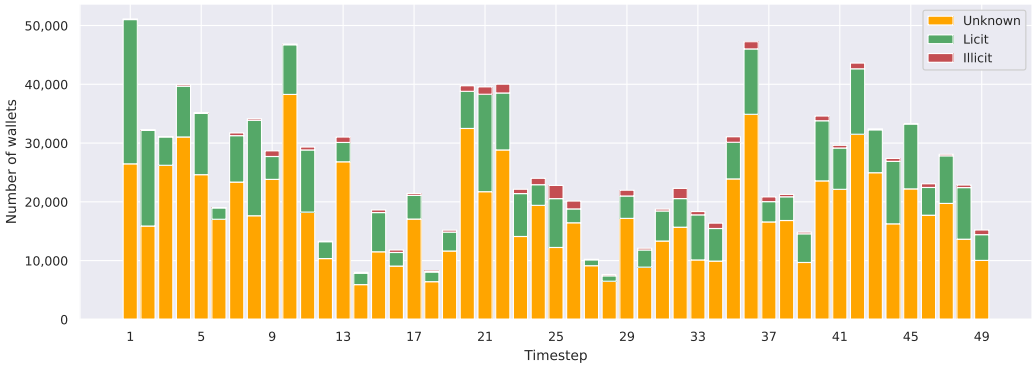


Figure 5. Number of wallets in each class per timestep.

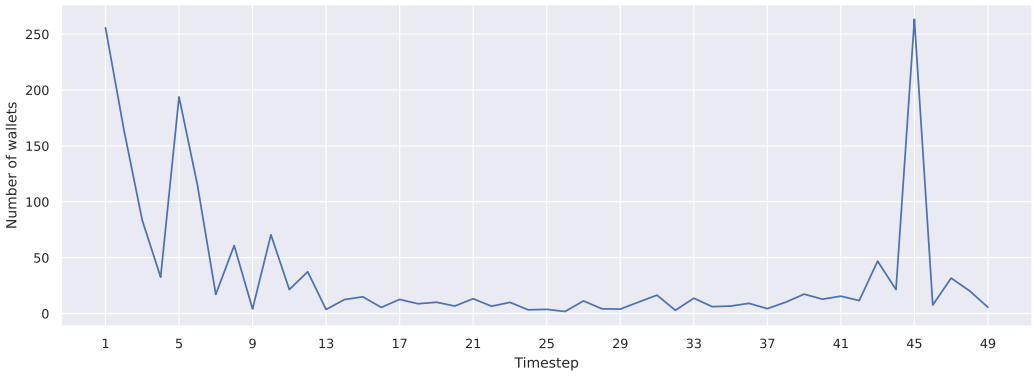


Figure 6. Imbalance between licit and illicit wallets per timestep.

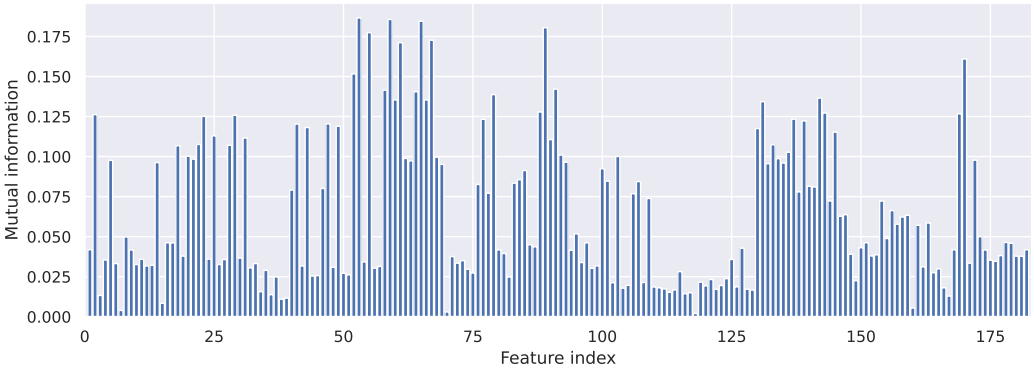


Figure 7. Mutual information of transaction features.

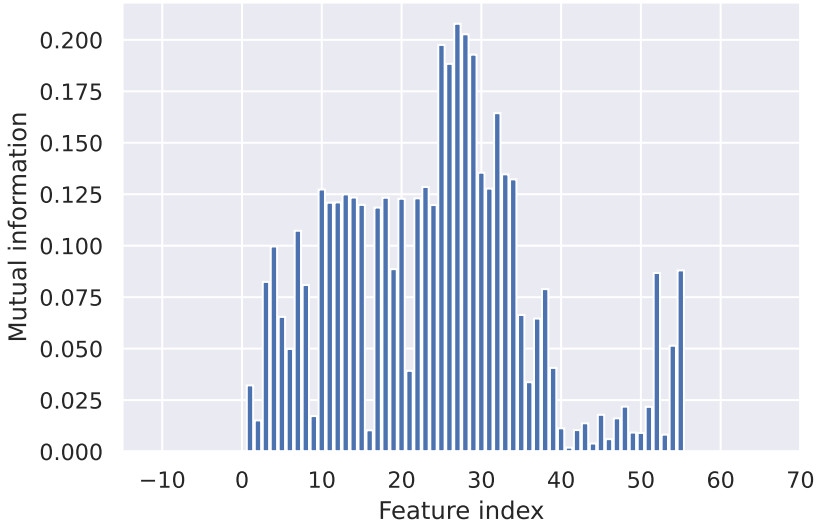
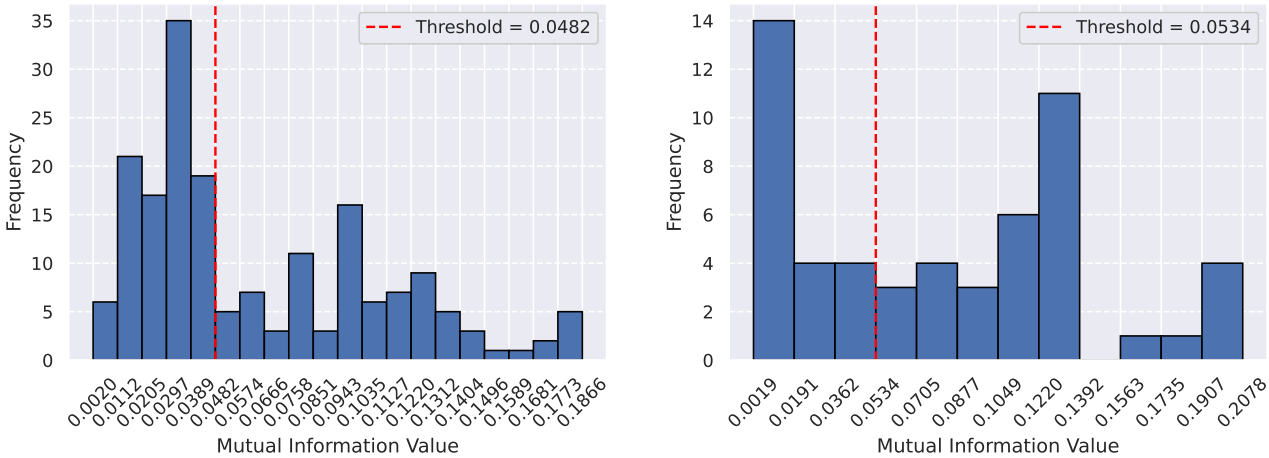


Figure 8. Mutual information of wallet features

As can be observed in the figures, especially for the transactions, there were slight patterns in some groups of features. This could be attributed to the fact that some features may have been obtained by aggregating other features. However, as stated in Section 4.2, the authors of the dataset do not provide such information. Although in both cases, the amount of mutual information was generally low, there existed slight differences between the features, which suggested that a component reduction algorithm could be beneficial for some models. On the first iteration, a feature ablation analysis was carried out by removing all the features below a threshold in the mutual information quantity. In order to select this threshold, a histogram was plotted for both transaction and wallet features to analyze the distribution of the mutual information values in both cases. Figure 9a,b show these histograms with the chosen threshold for both transactions and wallets, respectively.



(**a**) (**b**)

Figure 9. Distribution analysis of MI of features. (**a**) MI histogram for transaction features; (**b**) MI histogram for wallet features.

Figures 10 and 11 show the final result of the feature ablation compared to the previous mutual information plots. Unfortunately, this component reduction technique showed no improvement in subsequent experiments compared to other feature reduction algorithms such as Principal Component Analysis (PCA).

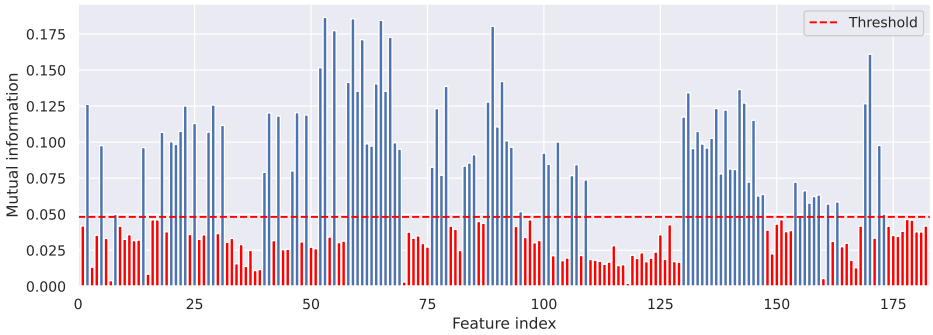


Figure 10. Ablation of transaction features. The red bars represent transaction features that have been removed because their mutual information is below the threshold line.

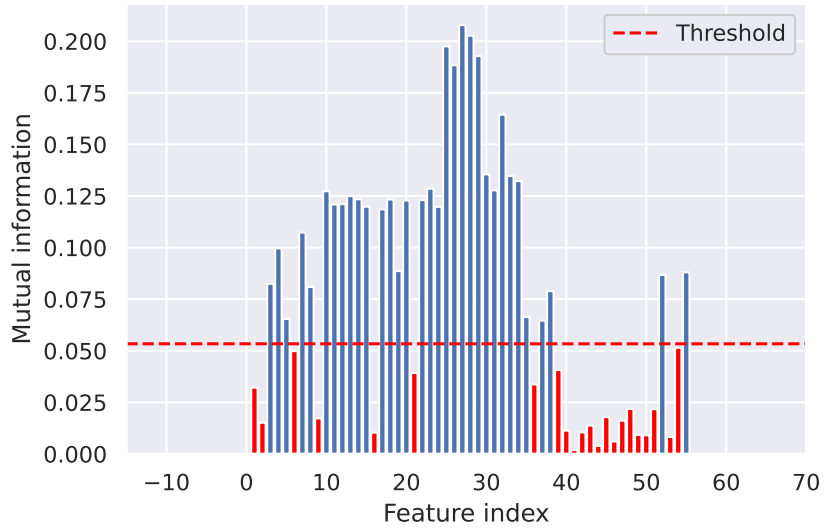


Figure 11. Ablation of wallet features. The red bars represent wallet features that have been removed because their mutual information is below the threshold line.

Unlike this technique, PCA is a more mathematical approach based on a linear dimensionality reduction aimed at maximizing the variance captured by the principal components. In order to find a suitable number of components, the cumulative explained variance was plotted and is shown in Figure 12. The elbow at 75 components indicates that this number of reduced features is enough to be input into the outlier detection algorithms since it balances dimensionality reduction and information retention.

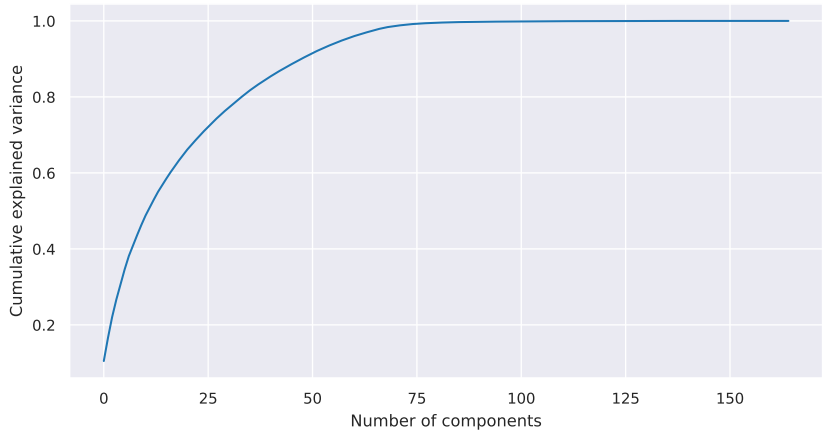


Figure 12. Cumulative explained variance (PCA).

After analyzing the features, visualizing a sample of the transactions graph can be useful for understanding the Bitcoin flow between wallets, but its construction requires the selection of a node and the careful exploration of the graph obtained from it. More specifically, an illicit wallet was chosen as a starting node to ensure the existence of fraudulent entities in the subsample, and a Breadth-First Search-like algorithm was applied to it with a depth of three levels. Figure 13 shows the resulting subgraph.

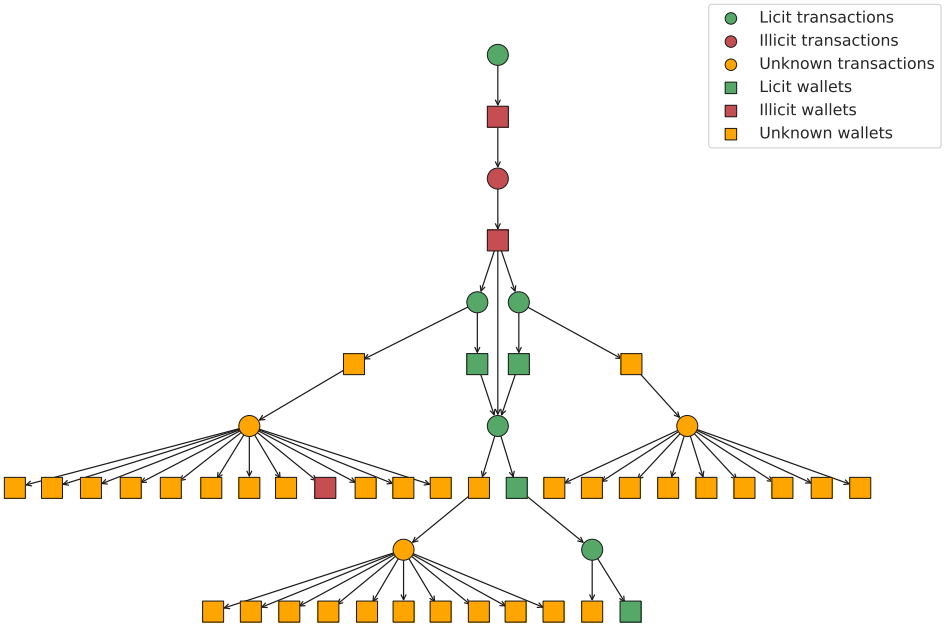


Figure 13. A visualization of a sampled subgraph.

5.2. Anomaly Detection

Throughout this section, we describe how the Elliptic transaction dataset was tested against a series of well-established outlier detection algorithms for the sake of testing Hypothesis 1. The objective was to find a correlation, at least a partial one, between the transactions classified as outliers by the algorithms and those labelled as illicit.

The choice of the anomaly detection algorithm needed to take into consideration different aspects intrinsic to the anomaly concept. For this experiment, a set of four common outlier detection algorithms was selected, as they represent some of the main techniques that are widely used in feature-based outlier detection.

- Local Outlier Factor (LOF) [19]: This algorithm evaluates the local outlier factor for an element based on the local density of its K nearest neighbours, considering their features for computing the distance. It is particularly robust when applied to datasets with varying densities or clusters. Therefore, it can be effective in detecting some types of fraud since they can often appear as localized anomalies, for example, a transaction with unusual patterns within a cluster of transactions.
- Isolation Forest (ISOF) [42]: This method uses several random binary trees with different decision thresholds for randomly selected features to isolate anomalous samples on the leaves. It is particularly optimal for high-dimensional data, which were analyzed in this study, and has a very efficient time complexity, which could help scale the algorithm to large datasets such as the entire blockchain.
- SGD One-Class SVM: This approach adapts the classical Support Vector Machine to a non-supervised approach by using a Stochastic Gradient Descent to efficiently fit a boundary that includes inlier data and keeps outliers on the outside. The ability to

model complex boundaries can make it effective in fraud detection, since anomalous patterns can be distinctly separated from normal behaviour.

- Elliptic Envelope: This method assumes that the data follow a Gaussian distribution and fits an ellipse that serves as a boundary between inliers and outliers. Many types of financial fraud often appear as deviations from standard Gaussian-like behaviour [43] (for example, the distributions of transaction amounts). Under these assumptions, the algorithm can efficiently identify these deviations.

All the algorithms were applied to the 75 reduced dimensions obtained after PCA, described in the preceding subsection. While both labelled and unlabelled data from the transactions dataset were fed to the outlier detection algorithms, only labelled and known data were taken into account for measuring the performance of these models. Since all these algorithms return an outlier score for each transaction, a threshold had to be fixed to establish the boundary between inliers and outliers. Therefore, and to deal with the high class imbalance of the dataset, this threshold was set so that 8% of the predictions were classified as outliers, following a licit/illicit proportion as near as possible to the real labels of the dataset. As a result, Table 1 shows the metrics of the four aforementioned algorithms, taking as positive predictions (illicit) those transactions classified as outliers and negative predictions (licit) those classified as inliers.

Table 1. Outlier detection methods' performance. I: inlier. O: outlier.

Algorithm	Precision		Recall		F1 Score		Accuracy	ROC-AUC
	I	O	I	O	I	O		
LOF ($K = 2$)	0.91	0.22	0.95	0.13	0.93	0.17	0.87	0.495
ISOF ($N = 1000$)	0.87	0.00	0.71	0.01	0.78	0.00	0.64	0.096
SGD One-Class SVM	0.93	0.13	0.49	0.68	0.65	0.21	0.51	0.567
Elliptic Envelope	0.87	0.01	0.72	0.03	0.79	0.02	0.66	0.173

Some of the best metrics were obtained with the LOF method using $K = 2$ neighbours, even though experiments conducted with other values of K obtained a very similar performance. This suggests that the local density was not that relevant for anomaly detection on this dataset. Figure 14 shows the confusion matrix of the results obtained in this specific experiment.

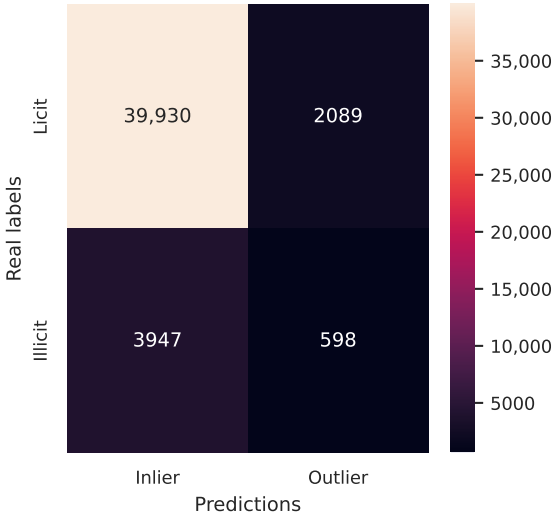


Figure 14. Confusion matrix for Local Outlier Factor ($K = 2$).

Although the table shows that most algorithms seemed to obtain poor performance, particularly when predicting illicit labels, it is worth noting that these predictions were based only on the features of the data and did not take into account their labels at any moment. Thus, they were expected not to reach the performance of traditional supervised learning models, but on the other hand, there was the advantage of not needing manually obtained labels, avoiding a tedious and complicated process. Moreover, these methods are noticeably more performant than others based on deep learning frameworks or graph-related solutions and therefore are more suitable to be applied on massive datasets or even the entire blockchain.

Still, these results show that Hypothesis 1 barely holds true when using strictly feature-based anomalies. Thus, the experiments suggested that the anomaly concept might be extended by taking into account other aspects such as the relational information of the transactions, which indeed, has been proven to be crucial in the context of fraud detection, as shown in other investigations. This could be achieved by making use of more sophisticated graph anomaly detection frameworks such as Graph Autoencoders [44], which identify structural anomalies based on high link reconstruction errors, or hybrid models such as DOMINANT [40], which combine both feature and structural anomalies.

5.3. Clustering

In this phase of the experimentation, another non-supervised learning approach was tested on the transactions' data: the k-means clustering algorithm. A clustering algorithm aims to group elements of the dataset into different clusters based on the similarity of their features.

For this specific case, splitting the set of transactions into different clusters can help distinguish between fraudulent and legal transactions depending on the cluster they belong to. In this manner, the results of different experiments were analyzed by contrasting each transaction's cluster with its label (licit or illicit). This analysis could be summarized in the corresponding contingency matrix. Additionally, to evaluate the quality of the generated clusters, a silhouette score analysis was carried out for each experiment to measure the cohesion of each cluster.

To handle the large number of points in the dataset, the experiments were carried out using a specific implementation of the algorithm called the mini-batch k-means, which carries out the creation of batches with a minimal impact on the results. After tweaking several parameters, including the number of clusters (K), some interesting results were obtained with the values of $K = 7$, $K = 33$ and $K = 49$. Figures 15–17 show their respective contingency matrices. In addition, Figure 18 shows the silhouette diagram for the $K = 33$ case.

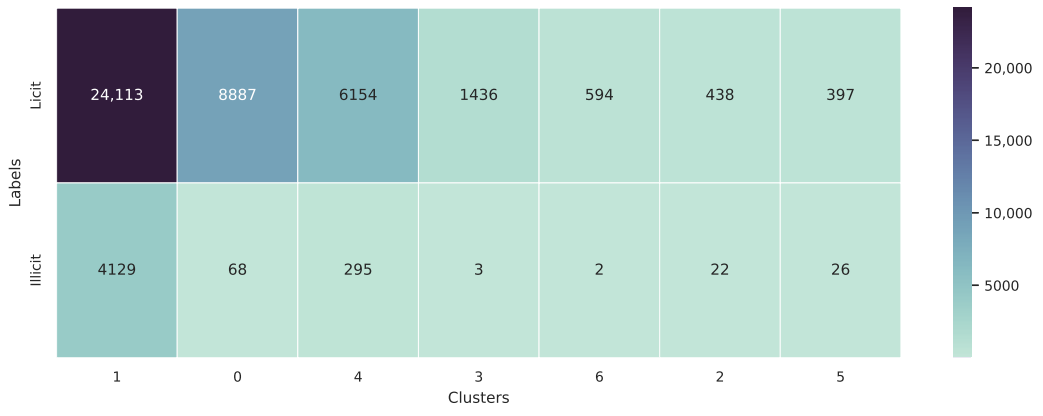


Figure 15. Contingency matrix of k-means $(K = 7)$.

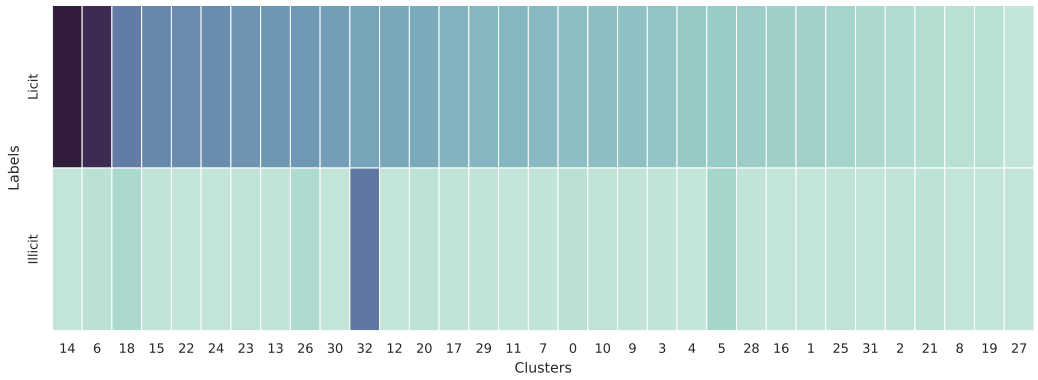


Figure 16. Contingency matrix of k-means ($K = 33$).

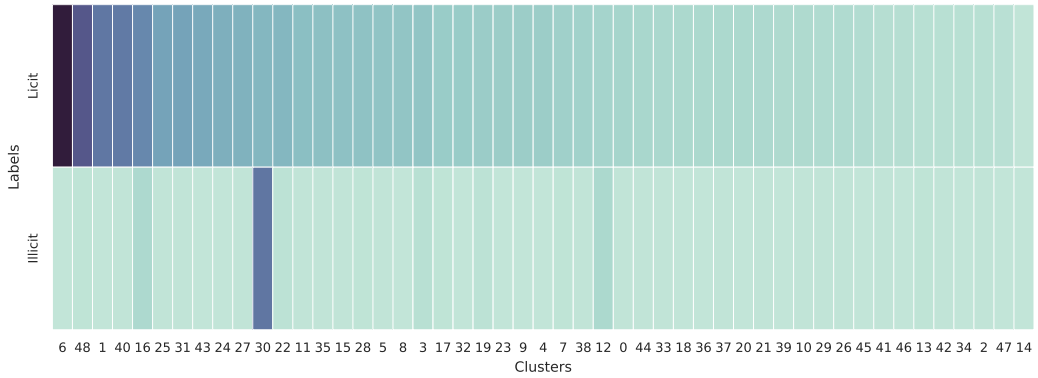


Figure 17. Contingency matrix of k-means ($K = 49$).

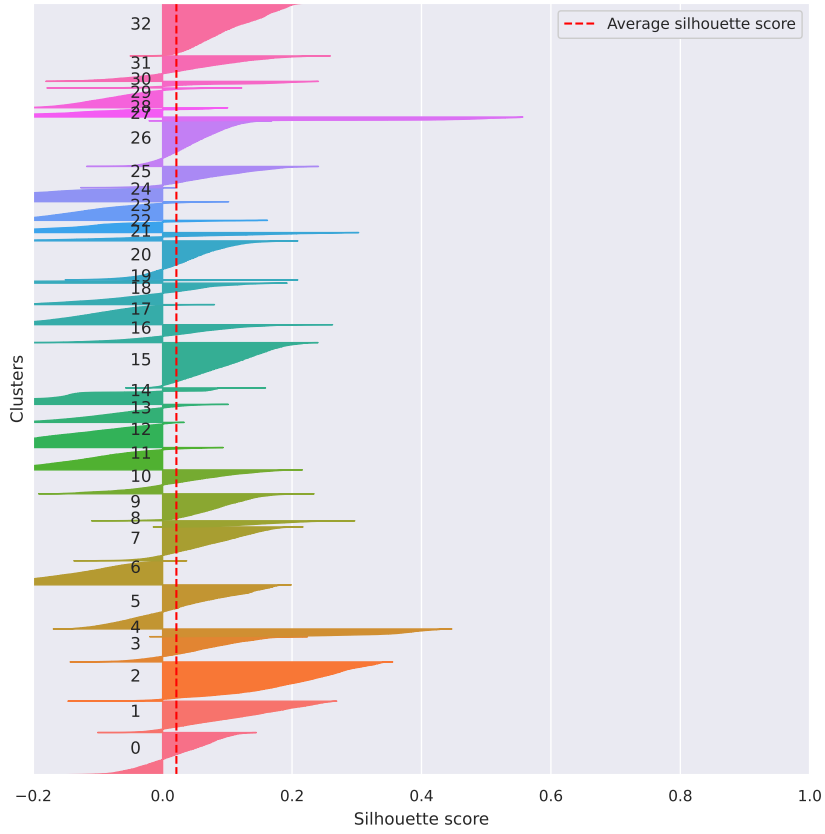


Figure 18. Silhouette score diagram of k-means ($k = 33$). Each color represents a different cluster.

While the results obtained with $K = 7$ showed a cluster with a remarkable number of illicit transactions (4129), that same cluster also contained the majority of licit transactions (24,113), making it useless for the task of distinguishing between both labels. Conversely, both experiments conducted with $K = 33$ and $K = 49$ revealed a cluster with a large number of illicit transactions while keeping the number of licit ones low, as illustrated by clusters 10 and 30 in Figures 16 and 17, respectively. The outcome of these experiments is significantly more relevant for the proposed task, as features shared by the transactions in those clusters might indicate a fraudulent tendency.

On the other hand, Figure 18 shows a very poor performance in terms of the silhouette analysis, with an average silhouette score close to 0. This indicates a low cohesion of the generated clusters and a generally bad clustering quality, suggesting that the features of the data, and especially the high dimensionality, hindered their clusterization.

5.4. Graph Deep Learning

Both Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs) have shown excellent performance when applied to traditional supervised learning tasks on cryptocurrency transactions and more specifically on the Elliptic dataset, as stated in previous related works. Following this approach, this section of the experiments aimed to test a recent state-of-the-art model that can leverage the heterogeneous nature of the Elliptic++ dataset.

The Heterogeneous Graph Transformer (HGT) [18] is a novel deep learning framework designed for heterogeneous graphs that combines the ideas of conventional GNNs with the Transformer [45] architecture, using different mutual attention mechanisms depending on the graph node type. This complex architecture can be used to manage different attention coefficients and weights for each node type in the Elliptic++ dataset, namely, transactions and wallets.

In order to carry out the experiments with this framework, an HGT Binary Classifier model was developed specifically for this case, using a linear transformation that projected the feature vectors into the embedding space and a series of HGT convolutional layers, followed by Leaky ReLU activation functions. Finally, another linear transformation was applied to obtain an output logit. The parameters for this model that were used in the experiment are displayed in Table 2.

Table 2. The parameters for the HGT classifier.

Model Parameter	Value
Embedding dimension	128
Number of message-passing layers	2
Number of attention heads	4

As described in Section 4.3, a mini-batch training strategy was applied to fit the complex architecture of the HGT model into the CUDA device memory. The subgraphs corresponding to each timestep in the dataset were used as individual units for batching. In addition, the feature matrices of both transactions and wallets were normalized in advance. For the training phase, an Adam optimizer was utilized with a custom loss function that aimed to minimize the loss of both transactions and wallets, taking into account a custom parameter, β, used to balance both loss functions, as shown in the following formula:

$$L = \beta \cdot L_{wallet} + (1 - \beta) \cdot L_{tnx} \quad (2)$$

Both transaction and wallet loss functions consisted of a binary cross entropy loss with a custom positive weight that helped with the class imbalance. Additionally, to deal with the class imbalance differences between timesteps previously analyzed in Section 5.1, a stratified train/test split strategy was applied to the dataset. All the parameters used during the training phase can be found in Table 3.

Table 3. Training parameters.

Training Parameter	Value
Batch size	8
Positive weight for wallets	5
Positive weight for transactions	5
Number of epochs	600
Learning rate	0.01
Weight decay	0.001
Wallet/tnx loss balance (β)	0.4

Figure 19 illustrates the evolution of the binary cross entropy loss of both wallets and transactions and the combined loss function throughout the 600 epochs of training. This plot shows an irregular decreasing trend with numerous fluctuations in most loss functions, typical of batch training in which gradients are computed for each batch.

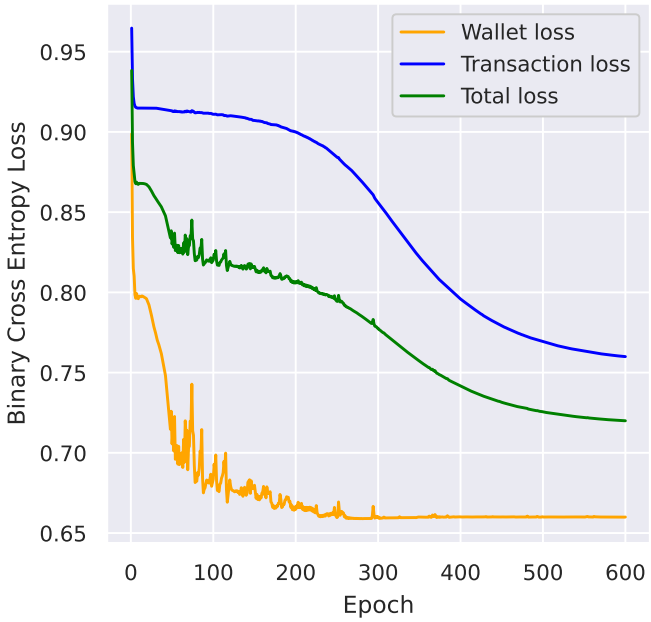


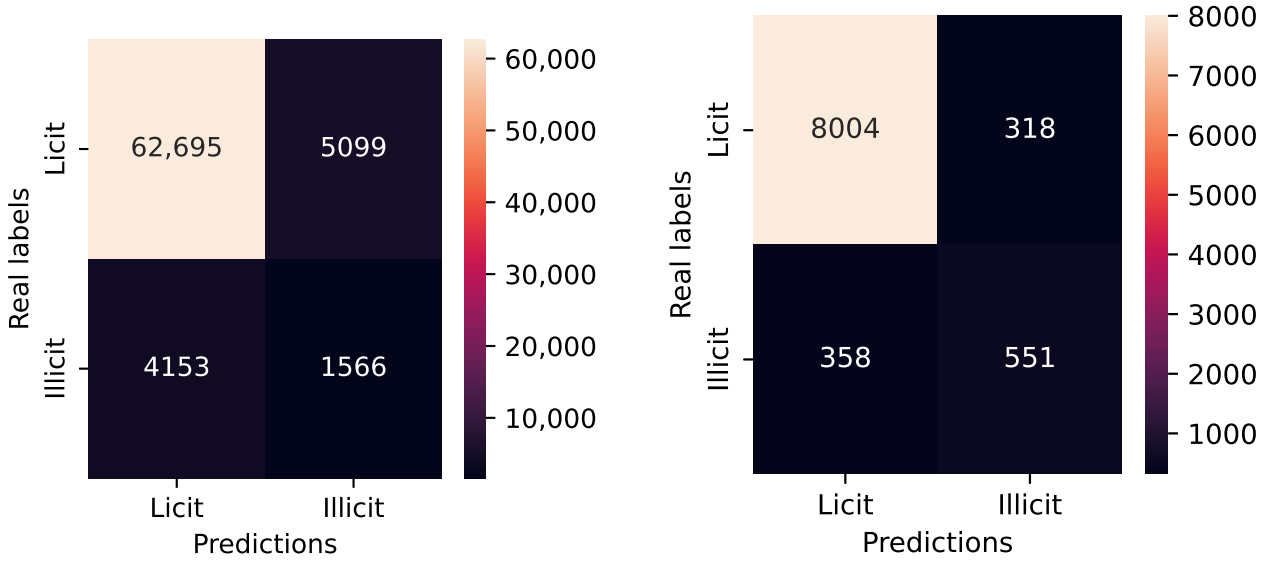
Figure 19. Loss functions during HGT model training.

Finally, the results of the experiments are displayed in Figures 20 and 21 with the confusion matrices and ROC curves of both wallets and transactions. Additionally, Table 4 summarizes the evaluation metrics of this experiment for both node types in the graph.

Table 4. The performance metrics of the HGT classifier experiment.

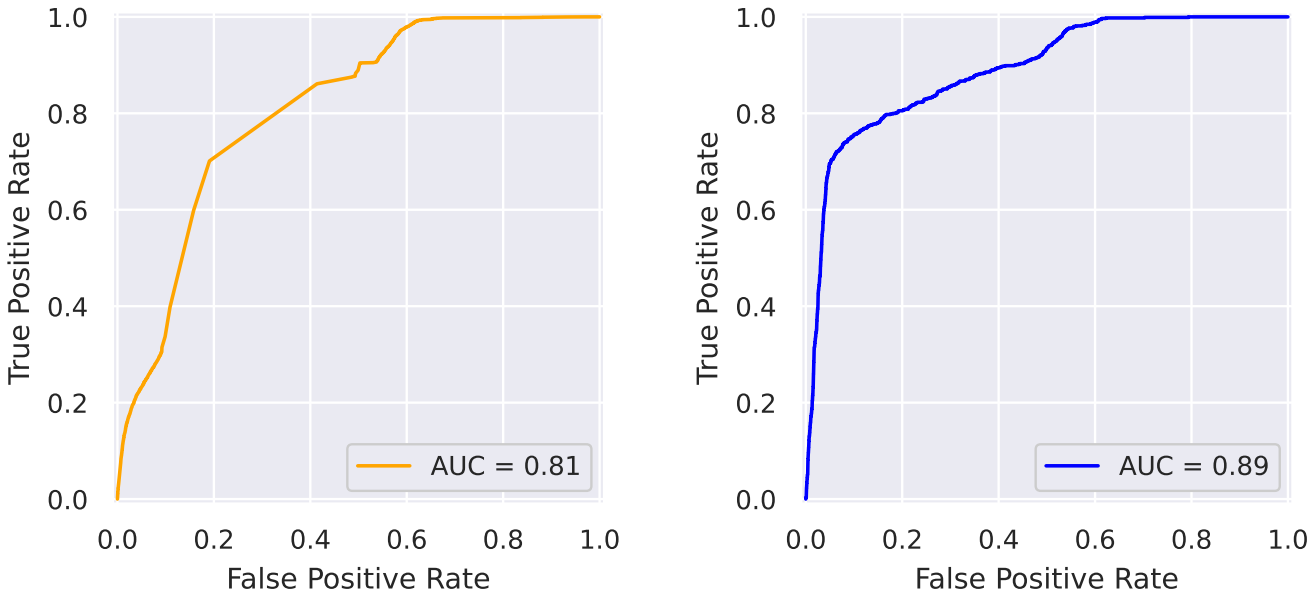
Node Type	Precision		Recall		F1 Score		Accuracy	ROC-AUC
	Licit	Illicit	Licit	Illicit	Licit	Illicit		
Wallets	0.94	0.23	0.92	0.27	0.93	0.25	0.87	0.812
Transactions	0.96	0.63	0.96	0.61	0.96	0.62	0.93	0.892

Although both confusion matrices show results that are not diagonal, it is important to take into account the class imbalance present in both sets. Therefore, it was expected to have a considerably large number of elements in the upper left quadrant of both matrices compared to the other quadrants. On the other hand, both ROC curves indicate correct behaviour, with an area under the curve (AUC) close to 1 in both cases.



(**a**) (**b**)

Figure 20. Confusion matrices of results of HGT model. (**a**) Confusion matrix for wallets; (**b**) confusion matrix for transactions.



(**a**) (**b**)

Figure 21. ROC curves of results of HGT model. (**a**) ROC curve for wallets; (**b**) ROC curve for transactions.

In addition, Table 5 provides a comparison of the HGT results for transaction nodes with other experiments from the literature that applied graph techniques to the standard Elliptic transaction dataset. Although the HGT did not directly outperform the other models, it should be noted that our experiments were carried out on a larger dataset with two different types of nodes (wallets and transactions). No direct comparisons with

other experiments on the Elliptic++ dataset were carried out since no works that apply heterogeneous graph approaches to this dataset were found in the current literature.

Table 5. A comparison of the HGT with other models. The first two were trained and evaluated using the standard Elliptic dataset, while the HGT was applied to the entire heterogeneous Elliptic++ dataset, and only the metrics for illicit transactions are shown.

Method	Illicit Transactions		
	Precision	Recall	F1 Score
Graph Convolutional Network (GCN) *Elliptic* [30]	0.81	0.51	0.63
Graph Attention Network (GAT) *Elliptic* [46]	0.92	0.86	0.89
Heterogeneous Graph Transformer (HGT) *Elliptic++ TX*	0.63	0.61	0.62

Overall, the HGT has proven to be an excellent framework for heterogeneous graph deep learning, achieving state-of-the-art results in the proposed task and revalidating the effectiveness of the classical supervised learning approach to fraud detection, despite requiring a large set of manually labelled samples.

6. Conclusions

With the growth of cryptocurrency, the fight against fraudulent activities has become a challenge. After introducing the background on cryptocurrency fraud and analyzing the related works, this article addressed the problem of identifying fraud in a large set of transactions extracted from the Bitcoin network. The goal was to find reliable methods to label Bitcoin transactions, taking into account their features.

This study analyzed the effectiveness of a variety of ML methods with different approaches to the task of identifying fraud on the Bitcoin cryptocurrency network. Although the work could be generalized to other currencies such as Ethereum by following a similar method, a large enough dataset is needed. Moreover, the transaction graph for each network should be adapted to the intrinsic features of that specific currency. For example, not all cryptocurrencies follow the UTXO model used by Bitcoin, which is the mechanism that was used to build the edges between transactions in the Elliptic dataset. At the same time, the Elliptic dataset also has some limitations, such as the fact that its authors do not provide detailed information about the feature selection, the transaction gathering or the graph construction processes and if graph-specific properties like the density or node degree distribution were taken into account.

On one hand, the results obtained with unsupervised learning methods have shown the usefulness of these approaches for detecting illicit transactions in the absence of a manually labelled dataset, despite still being far from the performance of supervised methods. Unfortunately, these methods also involve a series of potential drawbacks and limitations, such as the absence of labelled data for validation, difficulty in distinguishing between fraudulent patterns and other anomalous but licit activities, and sensitivity to noise and irrelevant data.

On the other hand, the experiments carried out using the HGT framework revalidated the capability of deep learning methods in supervised approaches, particularly graph-based methods. The importance of data heterogeneity was shown in the results, although no further experiments were performed to isolate which specific components of the HGT model were responsible for this.

Combining both unsupervised and supervised approaches in a more ambitious experiment could bring several benefits, such as enhancing the anomaly detection process, filtering out outliers before the use of deep learning models, handling newer and unknown fraud patterns or improving the explainability of the results.

The results emphasize the importance of integrating both the structural information and the intrinsic features of the data, particularly in the context of transactional data due to their relational nature. Aggregating the information of a given transaction with that of its neighbouring transactions is a crucial step to decide whether it might be illicit or not, since the commission of the fraud itself can occur several hops before or after the given transaction. This aligns with related works that applied graph frameworks to supervised learning methods, obtaining remarkable improvements. With this work, we believe that unsupervised methods can also benefit from the use of these frameworks through the detection of structural anomalies or the aggregation of neighbouring features.

As threats to the validity of our work, due to the dataset's high imbalance, it is necessary to consider additional evaluation metrics for the supervised approach, such as the Matthews correlation coefficient or the area under the Precision–Recall curve, because the Precision, Recall, F1 Score and ROC-AUC may not capture all the nuances. Also, for the unsupervised approaches, alternative clustering methods and pre-processing techniques must be explored to improve the cluster cohesion.

Although considerable progress has been made in recent years, several research lines remain open in this field. Therefore, in future works, one significant challenge will be to improve the quality, size and availability of labelled cryptocurrency datasets. On the other hand, tuning models and enhancing their adaptability to other types of data is another major challenge. Furthermore, incorporating hybrid approaches that can leverage the potential of graph deep learning frameworks in a non-supervised fashion is another interesting direction for future research.

Author Contributions: Conceptualization, V.P.-C. and F.J.; data curation, V.P.-C. and F.J.; investigation, V.P.-C. and F.J.; methodology, V.P.-C. and F.J.; software, V.P.-C.; validation, V.P.-C. and F.J.; writing—original draft, V.P.-C. and F.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the Center for Research in Forensic and Security Sciences of the Universidad Autónoma de Madrid (ICFS-UAM) (in Spanish *Centro de Investigación en Ciencias Forenses y de la Seguridad de la Universidad Autónoma de Madrid*, ICFS-UAM).

Data Availability Statement: The datasets used in this study can be found at the following links: Elliptic dataset, available at https://www.kaggle.com/datasets/ellipticco/elliptic-data-set accessed on 15 January 2025; and Elliptic++ available at https://github.com/git-disl/EllipticPlusPlus, accessed on 16 January 2025. The original contributions presented in this study are included in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Burgess, T. A multi-jurisdictional perspective: To what extent can cryptocurrency be regulated? And if so, who should regulate cryptocurrency? *J. Econ. Criminol.* **2024**, *5*, 100086. [CrossRef]
2. Perdana, A.; Jhee Jiow, H. Crypto-Cognitive Exploitation: Integrating Cognitive, Social, and Technological perspectives on cryptocurrency fraud. *Telemat. Inform.* **2024**, *95*, 102191. [CrossRef]
3. Chauhan, R.; Mehtar, K.; Kaur, H.; Alankar, B. Evaluating Cyber-Crime Using Machine Learning and AI Approach for Environmental Sustainability. In Proceedings of the Sustainable Development through Machine Learning, AI and IoT, Virtual Event, 27–28 April 2024; pp. 37–49. [CrossRef]
4. Taher, S.S.; Ameen, S.Y.; Ahmed, J.A. Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 12822–12830. [CrossRef]
5. Snigdha, K.; Reddy, P.S.N.; Hema, D.; Gayathri, S. BitPredict: End-to-End Context-Aware Detection of Anomalies in Bitcoin Transactions using Stack Model Network. In Proceedings of the 3rd International Conference on Advances in Computing, Communication and Applied Informatics, ACCAI 2024, Chennai, India, 9–10 May 2024. [CrossRef]

6. Dutta, S.; Sharma, A.; Rajgor, J. Ethereum Fraud Prevention: A Supervised Learning Approach for Fraudulent Account Recognition. In Proceedings of the 2024 1st International Conference on Trends in Engineering Systems and Technologies, ICTEST 2024, Kochi, India, 11–13 April 2024. [CrossRef]
7. Md, A.Q.; Narayanan, S.M.S.S.; Sabireen, H.; Sivaraman, A.K.; Tee, K.F. A novel approach to detect fraud in Ethereum transactions using stacking. *Expert Syst.* **2023**, *40*, e13255. [CrossRef]
8. Cunha, L.L.; Brito, M.A.; Oliveira, D.F.; Martins, A.P. Active Learning in the Detection of Anomalies in Cryptocurrency Transactions. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1717–1745. [CrossRef]
9. Ahmed, A.A.; Alabi, O.O. Secure and Scalable Blockchain-Based Federated Learning for Cryptocurrency Fraud Detection: A Systematic Review. *IEEE Access* **2024**, *12*, 102219–102241. [CrossRef]
10. Gürfidan, R. Suspicious transaction alert and blocking system for cryptocurrency exchanges in metaverse's social media universes: RG-guard. *Neural Comput. Appl.* **2024**, *36*, 18825–18840. [CrossRef]
11. Qasim Abdulkadhim, R.; Abdullah, H.S.; Hadi, M.J. Surveying the prediction of risks in cryptocurrency investments using recurrent neural networks. *Open Eng.* **2024**, *14*. [CrossRef]
12. Elmougy, Y.; Liu, L. Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 3979–3990. [CrossRef]
13. Hisham, S.; Makhtar, M.; Aziz, A.A. Anomaly detection in smart contracts based on optimal relevance hybrid features analysis in the Ethereum blockchain employing ensemble learning. *Int. J. Adv. Technol. Eng. Explor.* **2023**, *10*, 1552–1579. [CrossRef]
14. Chen, Z.; Liu, S.Z.; Huang, J.; Xiu, Y.H.; Zhang, H.; Long, H.X. Ethereum Phishing Scam Detection Based on Data Augmentation Method and Hybrid Graph Neural Network Model. *Sensors* **2024**, *24*, 4022. [CrossRef]
15. Choi, S.H.; Buu, S.J. Learning to Traverse Cryptocurrency Transaction Graphs Based on Transformer Network for Phishing Scam Detection. *Electronics* **2024**, *13*, 1298. [CrossRef]
16. Ouyang, S.; Bai, Q.; Feng, H.; Hu, B. Bitcoin Money Laundering Detection via Subgraph Contrastive Learning. *Entropy* **2024**, *26*, 211. [CrossRef] [PubMed]
17. Kang, J.; Buu, S.J. Graph Anomaly Detection with Disentangled Prototypical Autoencoder for Phishing Scam Detection in Cryptocurrency Transactions. *IEEE Access* **2024**, *12*, 91075–91088. [CrossRef]
18. Hu, Z.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous Graph Transformer. *arXiv* **2020**, arXiv:2003.01332. [CrossRef]
19. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the SIGMOD '00, 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 15–18 May 2000; Association for Computing Machinery: New York, NY, USA, 2000; pp. 93–104. [CrossRef]
20. Directorate-General for Internal Policies of the Union (European Parliament); Snyers, A.; Houben, R. *Cryptocurrencies and Blockchain—Legal Context and Implications for Financial Crime, Money Laundering and Tax Evasion*; Technical Report; EU Publications: Luxembourg, 2018. [CrossRef]
21. Europol. *Cryptocurrencies—Tracing the Evolution of Criminal Finances*; Technical Report; Europol: The Hague, The Netherlands, 2021. [CrossRef]
22. Cholevas, C.; Angeli, E.; Sereti, Z.; Mavrikos, E.; Tsekouras, G.E. Anomaly Detection in Blockchain Networks Using Unsupervised Learning: A Survey. *Algorithms* **2024**, *17*, 201. [CrossRef]
23. Natoli, C.; Gramoli, V. The Blockchain Anomaly. In Proceedings of the 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 31 October–2 November 2016; pp. 310–317. [CrossRef]
24. Kircanski, A.; Tarvis, T. Coinbugs: Enumerating Common Blockchain Implementation-Level Vulnerabilities. *arXiv* **2021**, arXiv:2104.06540. [CrossRef]
25. Siddamsetti, S. Anomaly Detection in Blockchain Using Machine Learning. *J. Electr. Syst.* **2024**, *20*, 619–634. [CrossRef]
26. Apiecionek, L.; Karbowski, P. Fuzzy Neural Network for Detecting Anomalies in Blockchain Transactions. *Electronics* **2024**, *13*, 4646. [CrossRef]
27. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [CrossRef] [PubMed]
28. Di, Z.; Wang, G.; Jia, L.; Chen, Z. Bitcoin transactions as a graph. *IET Blockchain* **2022**, *2*, 57–66. [CrossRef]
29. Tharani, J.S.; Charles, E.Y.A.; Hóu, Z.; Palaniswami, M.; Muthukkumarasamy, V. Graph Based Visualisation Techniques for Analysis of Blockchain Transactions. In Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, 4–7 October 2021; pp. 427–430. [CrossRef]
30. Weber, M.; Domeniconi, G.; Chen, J.; Weidele, D.K.I.; Bellei, C.; Robinson, T.; Leiserson, C.E. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. *arXiv* **2019**, arXiv:1908.02591. [CrossRef]
31. Lin, D.; Wu, J.; Yuan, Q.; Zheng, Z. Modeling and Understanding Ethereum Transaction Records via a Complex Network Approach. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2737–2741. [CrossRef]

32. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907. [CrossRef]
33. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903. [CrossRef]
34. Akcora, C.G.; Li, Y.; Gel, Y.R.; Kantarcioglu, M. BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain. *arXiv* **2019**, arXiv:1906.07852. [CrossRef]
35. Wang, K.; Pang, J.; Chen, D.; Zhao, Y.; Huang, D.; Chen, C.; Han, W. A Large-scale Empirical Analysis of Ransomware Activities in Bitcoin. *ACM Trans. Web* **2021**, *16*, 1–29. [CrossRef]
36. Huang, D.Y.; Aliapoulios, M.M.; Li, V.G.; Invernizzi, L.; Bursztein, E.; McRoberts, K.; Levin, J.; Levchenko, K.; Snoeren, A.C.; McCoy, D. Tracking Ransomware End-to-end. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 618–631. [CrossRef]
37. Hasan, M.; Rahman, M.S.; Janicke, H.; Sarker, I.H. Detecting anomalies in blockchain transactions using machine learning classifiers and explainability analysis. *Blockchain Res. Appl.* **2024**, *5*, 100207. [CrossRef]
38. Pham, T.; Lee, S. Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods. *arXiv* **2017**, arXiv:1611.03941. [CrossRef]
39. Shayegan, M.J.; Sabor, H.R.; Uddin, M.; Chen, C.L. A Collective Anomaly Detection Technique to Detect Crypto Wallet Frauds on Bitcoin Network. *Symmetry* **2022**, *14*, 328. [CrossRef]
40. Ding, K.; Li, J.; Bhanushali, R.; Liu, H. Deep Anomaly Detection on Attributed Networks. In Proceedings of the SIAM International Conference on Data Mining (SDM), Calgary, AB, Canada, 2–4 May 2019.
41. Bank, D.; Koenigstein, N.; Giryes, R. Autoencoders. *arXiv* **2021**, arXiv:2003.05991. [CrossRef]
42. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation Forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; ICDM'08; pp. 413–422. [CrossRef]
43. West, J.; Bhattacharya, M. Intelligent financial fraud detection: A comprehensive review. *Comput. Secur.* **2016**, *57*, 47–66. [CrossRef]
44. Zhu, D.; Ma, Y.; Liu, Y. Anomaly Detection with Deep Graph Autoencoders on Attributed Networks. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–6. [CrossRef]
45. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
46. K1SHIN. GAT Fraud Detection on Elliptic Dataset. 2021. Available online: https://www.kaggle.com/code/k1shin/gat-fraud-detection-illicit-f1-0-89 (accessed on 15 January 2025).