Anomaly Detection in Blockchain Networks

ALEXANDRU-TUDOR NECHITI, University of Twente, The Netherlands

Blockchain systems have risen in popularity in recent years, and many industries started adopting this technology. This can be attributed to their features, such as decentralization, consistency, anonymity, and transparency. Despite these advantages, as with other technologies, they are not immune to anomalous activities. The distinct characteristics of the blockchain make it more difficult to detect anomalies in such networks than the traditional ones. This research paper focuses on detecting anomalies within a dataset of Bitcoin transactions. It aims to improve understanding of anomaly behaviours within blockchain networks and explore how these anomalies can be effectively identified. Moreover, it tries to enhance existing methods for static anomaly detection and provide a comprehensive theoretical analysis of dynamic anomaly detection techniques.

Additional Key Words and Phrases: Blockchain, Anomaly detection, Network analysis, Transaction metrics

1 INTRODUCTION

Blockchain technology represents a distributed database that records transactions across a network of computers in a decentralized manner. This entails that each transaction is validated and maintained by a peer-to-peer network through specific protocols rather than a central authority, as is usually the case. The data within the network is made of linked blocks; each has a timestamp, a link to the previous block, and a list of transactions. These blocks are encrypted; therefore, altering any entry once added to the chain is nearly impossible. [21].

Most cryptocurrencies rely on this technology to ensure fast and accurate execution. Their transparency and decentralization allow for high trust between parties without needing a third party to mediate transactions [6]. These virtual currencies' evolution and underlying blockchain technology have been rapid and dynamic [12], and with their sudden increase in popularity, the networks supporting them have also increased in complexity. There is still much to learn about the fundamental properties of these networks and their evolution due to their novelty. This is especially true when detecting anomalies in these networks. Anomalies are data points that deviate from normal behaviour. In the context of blockchain and cryptocurrency, anomalies may indicate malicious activities that require further investigation and mitigation. Even though the protocols, anonymity, cryptography, and numerous other features of blockchain technology highlight its potential for securing transactions and preventing attacks, it is essential to note that the blockchain is not entirely immune to all sorts of fraud and other malicious activities [19].

The current literature on anomaly detection methods for blockchain-based networks is relatively new. This research project aims to examine the problem of anomalies in the context of such networks. Significant improvements in blockchain technology and cryptocurrencies have increased efficiency and transparency for all transaction parties. However, the immutability property means that anomalous transactions cannot be reversed. Therefore, quick detection of anomalies is of utmost importance for preventing damage or developing mitigation methods in due time. Anomaly detection in blockchain networks is more complex than in conventional networks due to the complexity of the former. All in all, the following research question is used as the basis for this research.

- Research Question: "How can we integrate data science methods to effectively identify anomalies in both static and dynamic cases within blockchain networks?"

The research question needs to be answered through an in-depth Bitcoin network analysis. Anomalies in the network can be identified using several data science techniques, such as machine learning, deep learning and statistical techniques. Furthermore, the main goals of this research can be categorized as follows:

Firstly, to investigate the evolution of leading cryptocurrencies and blockchain-based networks over time. Detecting abnormal data in these networks is a challenge, which requires understanding the fundamental properties and structures of these networks. Benchmarks for normal conditions need to be established.

Secondly, to examine the static case of anomaly detection using existing statistical and data mining methods, as a single method often results in a high rate of false positives. Therefore, existing methods are expanded and combined to improve detection accuracy.

Last, to analyze the theoretical foundations of dynamic methods for anomaly detection in blockchain-based networks. As these networks continually generate new data, real-time anomaly detection is crucial. A deeper understanding is gained by delving into the theory, potential applications and limitations of the dynamic anomaly detection methods.

2 RELATED WORK

In order to gain more insight into the topic at hand and gather related literature to the research domain, Google Scholar, Scopus, and IEEE were used. Many research documents related to blockchain technology, network graph analysis, and anomaly detection were collected using these search terms.

The research conducted by Signorini et al.[17] explore a machine-learning approach for detecting anomalies in blockchain networks. It works by extracting several characteristics from the blockchain network data and then using these features to train a machine learning model. In this way, the model is able to identify deviations from the norm within the blockchain network. The solution is scalable and can be applied to blockchain networks. However, it requires a significant amount of computational power.

In his research, A. Bogner leverages an unsupervised learning method with the aim of enhancing the anomaly detection process. His approach involves several key features, including the average count of transactions within a block over a specific period, the

TScIT 39, July 7, 2023, Enschede, The Netherlands

frequency distribution of block times, the volume of transactions gauged by gas expenditure, and the gas usage per block tracked over a time interval. These selected features contribute to the model's training, subsequently augmenting the precision of anomaly detection. In this context, 'gas' is the measuring unit for the computational exertion needed to execute a specific blockchain operation.

In his work, A. Bogner [9] uses an unsupervised learning algorithm to enhance the anomaly detection process. His approach is specifically designed for the Ethereum blockchain network, and it involves several features: the average number of transactions per block over time, the number of transactions based on gas consumed, and the gas consumption per block over time. All these features are used to train the algorithm and improve anomaly detection accuracy. In this context, 'gas' is defined as the measuring unit for the computational effort required to execute a specific blockchain operation.

The paper presented by T.Ide [13] proposes a method for collaborative anomaly detection on blockchain that combines data from multiple sensors. He argues that sensor data can be noisy and quite challenging to interpret. Therefore he comes up with an approach that combines data from multiple sensors to improve anomaly detection accuracy.

Thai T. Pham and colleagues [15] directed their attention towards anomaly detection, specifically within Bitcoin transaction networks. They employed k-means clustering, Mahalanobis distance, and unsupervised support vector machines to identify suspicious users and transactions. The study utilized a dataset containing two graphs, with one graph representing users as nodes and another representing transactions as nodes.

Despite several studies investigating anomalies in blockchain networks, the existing literature has significant potential for further development, as it is still in its early stages since blockchain itself as a technology is relatively novel.

3 METHODOLOGY

The steps taken to address the research question are described in depth in this section. The methodology enables the speedy detection of anomalous data, which is helpful for blockchain networks. The analysis used data from an extensive dataset of 2011 Bitcoin transactions made available by the Harvard Institute, accessible in the *Appendix* section. Pandas, Scikit-learn, TensorFlow, Keras, and other data science and machine learning libraries are used to implement the analysis in Python. The three components of the suggested research design are as follows:

I. Analysis of the evolution of blockchain-based networks and their properties. The initial phase involves investigating and analyzing the key properties of blockchain networks and their evolution. This crucial investigation aspect must be finished before we analyze current anomaly detection techniques and create new ones. Because normal and anomalous data concepts are not as well-defined and well-understood in this context, many assumptions concerning anomalies in other network types may not apply to such networks.

II. Analysis of static anomaly detection methods. The second phase focuses on enhancing the existing anomaly detection methods within blockchain networks. When attempting to detect anomalies in these networks, using a single method frequently results in a high rate of false positives due to the scarcity of labelled data. Therefore, the objective is to expand and combine existing methods to improve basic detection accuracy.

III. Analysis of dynamic anomaly detection methods. In the final phase of the research, the primary focus is on the theoretical analysis of dynamic anomaly detection within blockchain networks. Due to the rather limited time of this research, this phase consists only of theoretical analysis of such detection methods, without trying to enhance or apply them to some data, as is the case with the second phase.

4 RESULTS

The findings of this research project are divided and categorized based on the three phases that were also reflected in the methodology.

4.1 First phase

The first phase of the research comprises the analysis of the evolution of blockchain networks along with their key properties and the establishment of benchmarks to be used during the implementation.

4.1.1 Evolution over time.

In recent years, blockchain networks significantly impacted by creating a secure way to conduct transactions without an intermediary. These networks are *decentralized* since a network of nodes validates transactions within the network instead of a central authority. Each transaction is recorded on the blockchain and cannot be altered from that point. This highlights the *immutability* property and ensures that the network is *tamper-proof*, providing high security and transparency.

As these networks quickly evolved, the need for updated architectures and mechanisms that improve the network's performance and scalability also increased. For example, Bitcoin uses an algorithm called *proof-of-work* to verify transactions. This algorithm involves having miners solve complex mathematical problems using computational power. Other nodes in the network later assess the validity of the transactions based on the algorithm's output[6]. However, it comes with limitations. It can limit the transaction throughput of the network due to its slow speed. To address these issues, other consensus mechanisms generally considered faster were developed, such as *proof-of-stake*, used by networks like Ethereum, and *delegated proof-of-stake*, used by networks like EOS. These mechanisms use more efficient alternatives and no longer rely on miners. [18].

Besides various mechanisms, other factors, such as the size of the network and the transaction volumes they can handle, have also changed throughout time. The Bitcoin network has grown significantly and currently has over 16000 active nodes, according to [4]. Similarly, other cryptocurrencies, such as Ethereum, showcased similar growth. Despite this, these networks have scalability problems because of their low throughput, high transactional latency, and rising resource requirements. For example, in September 2017, the blockchain size of Bitcoin was about 158GB, with a bootstrap time of around four days for a new node to take part. As of 2021, its size has roughly reached 350GB. Ethereum seems to suffer from similar limitations.[20].

4.1.2 *Benchmarks.*

Benchmarks are essential during this research since they represent a baseline for detecting anomalies or unusual behaviour within the network. The focus needs to shift toward distinguishing between normal and abnormal data. Determining "normal conditions" benchmarks for network statistics can later be used as input for anomaly detection methods. According to [2], which is a reliable website that offers real-time data on the current status of the Bitcoin network, the most popular metrics/benchmarks used include average block size, market price, mining hash rate, average transaction fee, total confirmed transactions per day and so on. These benchmarks can provide a baseline for what is considered normal behaviour in the Bitcoin network, and deviations from these metrics may indicate anomalous activity.

However, the dataset of Bitcoin transactions tested consists of over 6 million transactions from 2011. Each transaction entry has the following attributes available: *timestamp*, *source address*, *destination address* and *size in Satoshi*, which is transformed in Bitcoin (BTC) for the sake of this research. Since it is made solely of unlabeled data, it is quite challenging to understand what constitutes anomalous data based on the available information alone. Thus, given the limited data attributes available, some benchmarks to establish normal conditions for Bitcoin transactions include:

- transaction volume: The number of transactions per day/week over time is used as a benchmark. This can help establish the average number of transactions during a specific period and identify any unusual spikes in the transaction activity.
- transaction amount: Each transaction amount can be used as a benchmark. This can help establish the average transaction size during a specific period and identify any unusual patterns in transaction amounts.
- transaction frequency: The frequency of transactions between specific addresses can be used as a benchmark. This can help establish the typical frequency of transactions between specific addresses and identify any unusual activity.
- time between transactions: This is the interval between consecutive transactions made by a specific source. This feature can help in identifying irregularities in transaction patterns.
- transaction ratio: This is the ratio of each transaction amount to the average transaction amount of a specific source. This feature can reveal if a transaction significantly differs from a source's usual amount.

4.2 Second phase

4.2.1 *Existing static detection methods.*

In recent years, the rise of blockchain technology and cryptocurrencies has increased interest in anomaly detection methods for these systems. Various statistical, machine, and deep learning techniques were used to detect anomalies in blockchain networks, as presented in the *Related Work* section.

Machine learning approaches use complex tools to find unusual patterns in data. Instead of using statistics to set a normal behaviour, they learn from the data, finding things that do not fit the usual patterns. It is divided into *supervised learning* and *unsupervised learning* methods, but due to a lack of labelled data, only the latter is used during this research. The following techniques are employed:

- Clustering: This technique groups similar objects into clusters. Clustering algorithms can detect anomalies without prior knowledge.
- Local Outlier Factor: is a density-based anomaly detection method that can identify anomalies based on local density deviation in a dataset [11]. In blockchain networks, based on benchmarks such as transaction volume, amount, frequency, and time between transactions, it calculates a score reflecting the degree of abnormality of data points, which can point out potential anomalies.
- Isolation Forest: This is a tree-based detection algorithm that isolates anomalies by randomly selecting a feature and a split value between the min and max values of the selected feature. Then, the points that do not fit with the rest represent anomalies/outliers. These points will stand out early on, making them easy to spot and separate from the rest of the data [1].

Deep learning can be viewed as a subset of machine learning. While machine and deep learning involve learning from data, the main difference lies in their complexity. Machine learning can use more straightforward, linear models to make predictions without requiring specific programming to execute a particular task. On the other hand, deep learning uses artificial neural networks that contain several layers. While a neural network with a single layer can still make approximate predictions, additional hidden layers can significantly increase the performance [5]. Deep learning uses a lot of architectures and techniques that can help to detect anomalies, but for the relevance of this research project, only the following is explained:

- Long Short-Term Memory Networks: These networks are particularly effective for sequential data [10]. In anomaly detection, they can identify anomalies in temporal data by learning what a 'normal' sequence looks like and then identifying any sequences that deviate from this. In blockchain transactions, their use can provide valuable insights into anomalous behaviour.

I) The first approach entails the use of unsupervised learning methods to analyze the case of anomalies within the data. The first technique is clustering based on transaction-level metrics, such as the benchmarks defined in the *4.1.2 Benchmarks* section. By grouping similar transactions into clusters, transactions that deviate significantly from their group's typical behaviour can be identified. Next, data is scaled, and algorithms such as Isolation Forest and Local Outlier Factor are applied to the identified clusters. The reason for using both methods is that they can provide a more comprehensive and diverse anomaly detection approach. Since they were identified by two complementary procedures, anomalies found by both methods are likely to be quite robust. These anomalies require further investigation. However, one crucial fact should be kept in mind; when working with unlabeled data, as is the case here, the interpretations are indeed hypothetical and based on the assumptions of the models used.

II) The second approach involves creating an anomaly detection model using an LSTM recurrent neural network (RNN). Using

LSTM layers, these networks can retain information for long periods, which is useful when dealing with sequential data[10]. This model can recognize transactions that differ from the norm by learning temporal patterns within the data. These transactions are then flagged as potential anomalies. However, it needs further refining with additional layers for better detection accuracy. More details are given in the implementation section below.

Nevertheless, it should be kept in mind that the cryptocurrency market is constantly fluctuating, with new currencies arising and trying to make an impact quite often.[12]. Therefore, static anomaly detection models may need help adapting to these changing conditions and require practical improvements. The previously mentioned approaches offer some alternatives that may improve the effectiveness of static methods.

4.2.2 *Analysis of the first method and implementation in Python.*

After exploring the key characteristics of blockchain networks and investigating the available static anomaly methods that can be used to detect anomalous data in such networks, it is time to use these findings through Python implementation, which is accessible in the *Appendix* section. The following steps were conducted:

I) Firstly, the focus was on a descriptive analysis before delving too deep into complex algorithms. This involved summarizing the main characteristics of the data, gaining some distribution insights, and identifying patterns by plotting graphs. All the graphs can be visualized in *Appendix C*. The graphs in figure 1 and 2 indicate a steady increase in the amount of Bitcoin transferred and the total number of transactions conducted. This increase makes sense as the popularity and use of cryptocurrencies such as Bitcoin have increased since their initiation. Additionally, the histogram depicted in figure 3 represents the distribution of logarithmic values of Bitcoin. In this chart, the most frequent logarithmic Bitcoin values are on the left, while the least frequent ones are on the right, suggesting a right-skewed distribution. This means that most data points have lower values and fewer high-value ones. Regarding anomaly detection, the transactions corresponding to the distribution's far right could be considered anomalous since they differ significantly from the majority. Nevertheless, just because an anomaly is considered anomalous according to the distribution, further investigation is needed to assess that.

II) For the second step, clustering was done through K-means, an unsupervised machine learning algorithm aiming to partition a given dataset into K clusters. Since there is no definitive rule for choosing the correct number of clusters, one proper technique to determine the optimal number is *elbow technique*. It is a heuristic technique that tries to balance between having too few clusters, which may result in an oversimplified representation of data, or too many clusters, which may result in overfitting [8]. Using this technique suggested six as the optimal number of clusters.

The K-means algorithm was then used with the number mentioned above. Figure 5 depicts the clustering done based on daily data, however, since we want to analyze transactions individually and not the aggregate daily transaction metrics, clustering was applied differently. As a result, Figure 4 represents the output of the clustering algorithm that uses transaction volume (x-axis) and average transaction amount (y-axis) as features for visualization.

Due to the enormous number of transactions, which is almost seven million, it is almost impossible to point out some cluster points, such as the ones contained by clusters three or four. To get a better view, figure 6 plotted the clusters without cluster 0, while figure 7 went even further and only plotted values from Cluster 1. Even though there are a lot of data points in this cluster, almost all of them overlap, indicating that they might have almost identical values.

One solution would be to use different features for visualization, but this is not the main focus of this research. Nonetheless, clustering aims to discover meaningful patterns in the data rather than produce clusters that can be easily visible. The identified clusters, alongside their statistics, are expanded upon in the following section.

Each cluster represents a group of transactions with similar characteristics in terms of transaction volume, average transaction amount, transaction frequency, time between transactions, and transaction ratio. These enumerated features have been explained in the *Benchmarks* section. To avoid formatting errors in generated LaTeX tables, the features will be labelled as follows:

- *F0 -> transaction volume
- *F1 -> average transaction amount
- *F2 -> transaction frequency
- *F3 -> time between transactions (in seconds)
- *F4 -> transaction ratio

-	F0	F1	F2	F3	F4
mean	3756.304	88.25	40.117	20916.961	0.990
std	17406.6	1064.5	452.13	159538.2	2.856
min	1	0	1	0	0
max	122710	122710	8425	3166045	695.57

Table 1. Cluster 0 statistics

Cluster 0: This is the largest group, with 6,318,031 transactions. The transactions in this cluster have a small average amount (88.25), also a relatively small transaction volume (3756.3), and their frequency is also low (40.117 on average). The time between transactions is relatively high, indicating that these transactions are happening infrequently. This cluster could represent regular, low-value transactions.

-	F0	F1	F2	F3	F4
mean	422834	3.4	210.817	45.189	0.653
std	0	0	316.932	4184	8.214
min	422834	3.4	1	0	0
max	422834	3.4	1956	2688635	817.79

Table 2. Cluster 1 statistics

Cluster 1: This cluster contains 422,751 transactions. The transactions in this cluster have a small average amount transferred (3.4) and identical transaction volume. The latter indicates a high degree of regularity, which could represent automated payments, a fact highlighted by the relatively high transaction frequency. However, the maximum transaction ratio of 817.790 is significantly higher than the mean, suggesting there may be a few highly anomalous transactions within this cluster.

Cluster 2: This cluster, with 61,102 transactions, also shows an identical volume and average transaction amount for all its transactions. The higher ratio suggests these transactions are more significant than the user's average amount.

-	F0	F1	F2	F3	F4
mean	122710	17.49	61102	58.95	1.868
std	0	0	0	315.1	2.369
min	122710	17.5	0	0	0
max	122710	17.5	61102	8322	36.31

Table 3. Cluster 2 statistics

-	F0	F1	F2	F3	F4
mean	365400	8.5	71.27	6827.9	1665.189
std	132101	53.876	27.10	62827.57	233.8
min	1002	0	1	0	921.31
max	422834	539	83	623453	1766.37

Table 4. Cluster 3 statistics

Cluster 3: This is the smallest cluster, with only 99 transactions. However, these transactions have the highest volume (365,400 on average). Given the minimal size of this cluster, it means that the sources inside the clusters are involved in many transactions. Moreover, the transaction ratio in this cluster is very high compared to other clusters. Such a high ratio indicates that the transaction amounts in this cluster are quite large compared to the average transaction amount made by the corresponding sources. It could be indicative of outliers or anomalous transactions.

-	F0	F1	F2	F3	F4
mean	37.528	70.649	5.730	6296197	0.555
std	145.11	881.17	101.49	3439394	1.5
min	2	0	1	3439394	0
max	8492	79193	8425	31120628	126.8

Table 5. Cluster 4 statistics

Cluster 4: This cluster contains 20,420 transactions and has a high degree of feature variability, indicating diverse transaction behaviour within this cluster. For example, the high time between transactions, with a mean of approximately 73 days, may indicate infrequent transactions. Even though great values do not represent the mean of transaction volume and average transaction amount, their relatively high standard deviation shows greater variability in their values.

-	F0	F1	F2	F3	F4
mean	1.8	254125.34	1.008	5056.6	1231
std	0.588	72616.5	0.09	132053.9	0.866
min	1	129948.5	1	0	1
max	6	499246.5	2	4467637	3.33

Table 6. Cluster 5 statistics

Cluster 5: This cluster has 1,231 transactions with very high average transaction amounts (254,125.34), the highest among all clusters. The transaction volume is relatively low, with a mean value of 1.8 and a maximum of 6. This suggests that sources in this cluster are involved in a few transactions but with high Bitcoin amounts. Furthermore, the time between transactions within this cluster varies significantly, with some entities showing a significant time gap between their transactions. This cluster's behaviour, especially the low volume and frequency combined with very high amounts transferred, stand out the most among all clusters and suggests the presence of anomalies.

We observe distinct behaviours for each in interpreting the transaction patterns across the clusters. Specifically, Cluster 3 and 5 are potentially anomalous due to their distinct traits. Cluster 3, while the smallest cluster, has an exceptionally high transaction volume and transaction ratio, making it stand out from the other clusters. Its high transaction ratio could be a flag for anomalies, given that it is way more prominent than in other clusters. Meanwhile, Cluster 5 is characterized by infrequent transactions but significantly high value. The average transaction amount for this cluster is also substantial, hinting at potential outlier behaviour. Furthermore, a wide gap in the time between transactions is observed, adding to this cluster's unusual pattern.

III) The last step involved applying anomaly detection algorithms such as Isolation Forest and Local Outlier Factor post-clustering, which allowed the identification of outliers within each cluster separately. As these clusters represent groups of similar transactions, the benefit of applying anomaly detection post-clustering could be the identification of anomalies within each cluster. Separating the data into clusters can establish each group's "normal" behaviour. Anomalies can then be detected as deviations from the norm within their respective groups.

Both algorithms used on the identified clusters yielded the following output: *68234* anomalies found by Local Outlier Factor and *68141* anomalies found by Isolation Forest. Running both algorithms on the same data can provide a more comprehensive view of the anomalies. This statement is reinforced by the fact that these algorithms approach anomaly detection from different angles and may identify anomalies that the other is incapable of. Local Outlier Factor is based on density, whereas Isolation Forest is based on data partitioning. In both cases, the anomalies appear to be characterized by unusually high values for several variables.

In scenarios where a single method is utilized, the chance of data points being incorrectly flagged as anomalies may increase. These can often be false positives, data points incorrectly identified as anomalies due to certain limitations or biases of the model used. For this reason, anomalies detected by both methods are likely to be highly robust as they are identified by two separate techniques that employ different approaches to anomaly detection. The total number of common anomalies - that is, anomalies identified by both Isolation Forest and Local Outlier Factor is: *1181*. The values from Table 7 represent the summary statistics of the data points identified (common anomalies).

The statistical differences between normal and anomalous data can offer valuable insights regarding the diverse characteristics of anomalous behaviour in Bitcoin transactions. The results show that anomalies frequently involve significant amounts of Bitcoin. This implies that anomalies are more likely to occur in high-value transactions. This can indicate possible criminal activities, such as money laundering or scams, usually involving larger amounts.

The attributes' labelling and what they stand for have already been explained in the previous section.

-	F0	F1	F2	F3	F4	Type
mean	120028.4	9065.9	1902.6	386130.7	20.2	Anomalous
std	178678.3	17074.4	8318.2	1950503.6	100.7	
mean	30477.4	99.4	572.771	49841.1	0.94	Normal
std	102724.6	3582	5755.5	48340	6.75	

Table 7. Comparison of Anomalous and Normal Data

- *transaction volume*, this benchmark shows the number of transactions made by each source and is significantly higher for anomalies.
- *transaction frequency*, between each unique combination of source and destination addresses, is also higher in anomalous cases. This implies that anomalous transactions occur more frequently between certain addresses, hinting at unusual trading behaviours.
- *average time between transactions*, is also higher for anomalies. This may suggest that anomalous transactions are not uniformly distributed over time but rather occur in rapid succession.
- *transaction ratio*, for this benchmark as well, the value is more substantial for anomalies. Transactions with a notably high ratio could represent illicit activities like money laundering, usually characterized by more incoming transactions than outgoing ones.

These metrics showcase the patterns that characterize anomalous Bitcoin transactions within the data. These unsupervised learning algorithms use a variable called the "contamination factor" that reflects the percentage of anomalies in the data. The actual percentage of anomalies is unknown due to the lack of labelled data. However, by considering the domain and the fact that anomalies are usually rare, we can assume a small percentage of 1%.

4.2.3 Analysis of the second method and implementation in Python. The second approach used a Long Short-Term Memory (LSTM) neural network. Each Bitcoin transaction is not isolated but part of a larger chain of events influenced by various interactions between addresses within the network. Thus, for the sake of the analysis, the transactions were modelled as a directed graph using *NetworkX* Python library [3]. Each node represents a Bitcoin address, and each edge is a transaction between two addresses. In contrast to the previous approach, which mainly relied on transaction-level features, this e different employs features. Initially, it runs using *network-related* features. Subsequently, in a separate training session, the model is fed with *temporal features*.

The model can learn to predict patterns within these features. This knowledge can be used to detect anomalous data. The model is tested using different configurations to ensure a comprehensive view of the results.

The model uses bidirectional LSTM layers, which enable it to analyze data in both forward and backward directions. This feature increases its efficiency in recognizing patterns in the data. Besides the LSTM ones, it also incorporates Dropout layers to prevent overfitting and the 'Adam' optimizer, one of the most widely used optimization algorithms nowadays due to its effectiveness.

The model is trained using either the Mean Squared Error (MSE) or Mean Absolute Error (MAE) loss function. After training, the model makes predictions on the test set. Each transaction's error is then calculated by comparing the predicted and actual values. Transactions with errors exceeding the 95th percentile threshold are classified as anomalies. This threshold ensures that the model focuses on identifying the most extreme inconsistencies, representing transactions that deviate significantly from the expected patterns captured by the model. This threshold was chosen since it balances the need to detect anomalies, which are rare by definition, with the need to limit false positives.

The test set is made of *1705909* transactions, which represents 25% out of the total number from the original dataset.

Experiment No.	**Feature Type**	**Number of LSTM Units**	**Dropout Rate**	**Anomalies identified**
1	Network	50	0.1	85827 (MAE) 85245 (MSE)
2	Network	150	0.2	85287 (MAE) 85257 (MSE)
3	Network	50	0.3	85287 (MAE) 85251 (MSE)
4	Network	100	0.3	85287 (MAE) 85256 (MSE)
5	Temporal	50	0.1	85296 (MAE) 85253 (MSE)
6	Temporal	100	0.2	85296 (MAE) 85296 (MSE)
7	Temporal	150	0.1	85296 (MAE) 85254 (MSE)
8	Temporal	100	0.3	85296 (MAE) 85255 (MSE)

Table 8. Experiments with different model configurations.

The model ran with different configurations to check whether the change in LSTM units (neurons) or the Dropout rate and loss function would heavily impact the outcome. Table 8 displays the results, which are almost identical in terms of the number of anomalies identified. This suggests the model is quite robust to configuration changes. The fact that the number of anomalies identified is very similar across different configurations suggests that these anomalies are quite distinct and can be identified by the model regardless of these hyperparameters.

The **network features** are more suited to identify anomalous nodes (addresses) rather than anomalous transactions. However, it can be justified that transactions that involve anomalous nodes might be considered suspicious and require further investigation. The following were used:

- degree centrality - the in-degree and out-degree metrics represent the fraction of nodes' incoming and outgoing edges connecting a node and are calculated for source and destination addresses.

- eigenvector - measures the importance of a node within the network based on its centrality.

-	in-degree	out-degree	eigenvector	Type
mean	1.12	1.56	7.1	Anomalous
std	8.6	5.67	7.87	
mean	1.25	1.38	7.9	Normal
std	7.72	4.84	6.67	

Table 9. Anomalous and Normal Data using Network features

According to table 9, the average centrality Measures (in-degree, out-degree, eigenvector) are lower for anomalies than those for normal transactions. This implies that anomalous transactions usually involve nodes less central to the network. This idea makes sense when considering malicious activities since individuals engaging in them may attempt to evade detection by minimizing their interactions with the main network. Additionally, the standard deviation of most features for anomalous transactions is higher than that of normal transactions. This indicates that anomalous transactions are those that are varied and less predictable.

On the other hand, the **temporal features** provide a chronologically-oriented perspective to the transactions. This can help identify patterns or anomalies that are not just based on the transaction values or the network structure but also on the timing and regularity of transactions. The following features were used:

- time since the last transaction (F0)
- transaction count in last 24h (F1)
- average transaction value last 24h (F2)

-	F0	F1	F2	Type
mean	3.89	26.43	1.49	Anomalous
std	2.52	106.2	44.7	
mean	5.22	3.08	1.28	Normal
std	4.21	9.09	4.44	

Table 10. Anomalous and Normal Data using Network features

Based on the information in Table 10, transactions classified as anomalous display distinctive characteristics. The average time since the last transaction is shorter for anomalous transactions, suggesting a more frequent occurrence. Furthermore, the transaction count for anomalous transactions in the last 24 hours is significantly lower, implying less frequent activity on a day-to-day basis. Despite this, these transactions involve larger Bitcoin amounts, as evidenced by anomalous transactions' slightly higher average transaction value. These patterns could indicate potentially suspicious behaviour within the network.

It is worth reminding that, due to the lack of labelled data, unsupervised learning models aim to identify potential data points that may require further investigation rather than drawing definitive conclusions about the nature of these data points.

4.3 Third phase

In the final phase of the research, the main focus was on the theoretical analysis of dynamic anomaly detection in blockchain networks. As these networks continue to generate more data with each new transaction, the need for reliable real-time anomaly detection also increases. The main challenge is represented by blockchain systems' evolving characteristics, making relying only on static anomaly detection models incredibly difficult. The dynamic nature of these networks means that the data distribution can change over time, making it challenging for models to adapt.

Methods for dynamic anomaly detection in these systems usually involve graph-based techniques. These techniques exploit the graph structure of blockchain transactions to detect anomalous patterns. For instance, a sudden increase in the transaction volume between two entities may indicate an anomaly. However, these methods often face scalability issues due to the large size and complexity of blockchain networks, as noted by [7]. The authors introduce an end-to-end unsupervised framework for detecting edge anomalies in dynamic multiplex networks in the same paper. The framework, named "ANOMULY", employs Graph Neural Networks (GNNs) and Gated Recurrent Unit (GRU) cells to exploit the network's temporal properties. Theoretically, this framework can also be used in blockchain networks due to structural similarities between the two types of networks, particularly in the representation and processing of data. Both can be modelled as graphs, with entities as nodes and relationships as edges. This graph-like structure allows for applying similar analytical techniques, especially those from graph theory.

Moreover, it can be particularly useful in such networks, where anomalous transactions may only represent a small fraction of the total network activity. The authors found that the model outperformed all baselines in dynamic and static multiplex networks, improving the best baseline results by an average of 8.18%. This suggests that integrating GNNs, GRUs, and attention mechanisms may provide an efficient solution to dynamic anomaly detection challenges in blockchain networks.

Additionally, [16] offers a thorough review of anomaly detection methods in dynamic networks, including those applicable to the blockchain. It categorizes these methods into four types: anomalous vertices, edges, subgraphs, and events. Methods include graph-based techniques, similar to the previously discussed paper. However, these methods also face scalability issues due to blockchain networks' large size and complexity. The paper also discusses using scan statistics for vertex detection, Bayesian discrete-time counting processes for edge detection, fixed subgraphs, and other alternatives. Each method can play a part in detecting anomalies within blockchain networks.

Nevertheless, despite these advancements, there are still limitations to dynamic anomaly detection in blockchain networks. Besides scalability issues, another significant challenge is the lack of labelled data for training and evaluating models. This is particularly problematic in the blockchain domain, where privacy is essential and obtaining labelled data for fraudulent transactions takes much work. The paper of Lorenz et al. [14] describes this issue regarding money laundering identification in blockchain networks, where it is complicated to get access to labelled data to train machine learning models on. Money laundering is a malicious activity that may be reflected by some of the anomalous transactions identified in the previous sections. Furthermore, blockchain networks' complex and evolving nature can lead to new protocols, interactions and, thus,

new anomalous behaviour not encountered. This aspect poses challenges for existing detection methods since models are trained on past data.

5 DISCUSSION

The results have various implications for anomaly detection in blockchain networks. The first approach used clustering on transaction-level metrics and then applied Isolation Forest and Local Outlier Factor algorithms on the identified clusters. The main idea was to analyze the statistics of the common anomalies identified by both complementary methods to provide a more comprehensive view. This could be particularly beneficial for identifying potentially fraudulent transactions or other malicious activities within the network, improving its security and reliability. This approach is somehow limited by the need to assume the percentage of anomalies within the data. In this case, the 1% percentage was chosen since it is a conservative estimate based on the assumption that anomalies are usually rare, prioritizing the minimization of false positives.

The second approach involved a more complex model with LSTM layers that uses network metrics and temporal features to assess the state of a transaction. The model is trained with either a Mean Squared Error (MSE) or Mean Absolute Error (MAE) loss function, but the choice is negligible since they more or less identify the same anomalies. The model identifies anomalies that exceed the 95th percentile threshold. The results after running ten experiments with different model configurations. Although various configurations have been tested, using only the 'Adam' optimizer could be a limiting factor. Different optimizer alternatives should be explored to yield different outcomes.

Moreover, the findings are mainly hypothetical due to the lack of labelled data. In the preliminary stages of this research, one idea was to categorize these anomalies based on the type of malicious activity they represent, whether it be money laundering, scams and other fraudulent activities. However, the idea was dropped due to the scarcity of labelled data in this field since using supervised learning methods to train the models was outside reach. While the models have identified potential anomalies within the dataset, confirming their validity as true anomalies is impossible. This uncertainty emphasized the need for more research and development in this area.

Lastly, delving into the theoretical aspects and potential of dynamic anomaly detection methods in blockchain networks resulted in few significant findings. The current literature could is limited, and most graph-based methods highlighted are meant to be used in multiplex networks. This can also be applied to blockchain networks since they share structural similarities. However, gauging their effectiveness at the current stage is complicated.

6 CONCLUSION

In conclusion, the research conducted in this paper delves into the phenomenon of anomalies in blockchain networks, with an emphasis on Bitcoin transactions. Despite their unique features highlighting robust privacy, these networks are far from perfect. The anomalies identified, which could be signs of malicious activities, are more challenging to detect in these networks compared to traditional ones.

Reflecting upon the research question established in the beginning, this paper has investigated and tried to enhance the existing methods for static anomaly detection. The analysis and implementation were made exclusively using unsupervised learning methods. Initial descriptive analysis revealed patterns and distributions in the data, followed by two unsupervised learning approaches. In the final stage, the research delved into the current status of dynamic anomaly detection methods in blockchain networks, showing that the current advancements are quite limited.

7 LIMITATIONS AND FUTURE RESEARCH

Further research should continue exploring and developing innovative anomaly detection methods in blockchain networks that address the ongoing challenges. The biggest challenge is the lack of labelled data, mainly due to the blockchain's nature. While transaction data is public, the parties involved and their identities are not disclosed or attached to the transactions. In addition, what constitutes an anomaly may vary depending on the context. For example, a transaction involving a large amount of Bitcoin might be considered normal in business transactions but could be seen as anomalous in other contexts. This makes it challenging to label data as 'normal' or 'anomalous' based on the transaction data alone, and more domain knowledge is required.

Another challenge is the dynamic nature of blockchain networks. These networks continually generate new data with each transaction. This makes it challenging for static anomaly detection models to adapt and detect anomalies efficiently. A lot more focus should be on dynamic anomaly detection methods.

The scalability of anomaly detection methods is also an issue. As blockchain networks grow in size and complexity, it becomes increasingly difficult to process and analyze vast amounts of data promptly.

Ultimately, it is essential to acknowledge that this field is still in its early stages and has considerable potential for further development. As blockchain technology evolves, so must the techniques for detecting anomalies within its networks to mitigate malicious actions.

REFERENCES

[1] 2022. Isolation Forest — Auto Anomaly Detection with Python | by Andy McDonald | Towards Data Science. https://towardsdatascience.com/isolation-forest-auto-anomaly-detection-with-python-e7a8559d4562

[2] 2023. Blockchain Charts. https://www.blockchain.com/explorer/charts

[3] 2023. Centrality — NetworkX 3.1 documentation. (2023). https://networkx.org/documentation/stable/reference/algorithms/centrality.html

[4] 2023. Coin Dance | Bitcoin Nodes Summary. (2023). https://coin.dance/nodes

[5] Charu C Aggarwal. 2018. *Neural Networks and Deep Learning*. Vol. 1. 300–301 pages. https://doi.org/10.1007/978-3-319-94463-0

[6] Imran Bashir. 2017. *Mastering Blockchain, Second Edition.* Technical Report.

[7] Ali Behrouz and Margo Seltzer. 2022. Anomaly Detection in Multiplex Dynamic Networks: from Blockchain Security to Brain Disease Prediction. (2022).

[8] Purnima Bholowalia and Arvind Kumar. 2014. EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. *International Journal of Computer Applications* 105, 9 (2014), 975–8887.

[9] Andreas Bogner. 2017. Seeing is understanding - Anomaly detection in blockchains with visualized features. *International Joint Conference on Pervasive and Ubiquitous Computing and the ACM International Symposium on Wearable Computers* (9 2017). https://doi.org/10.1145/3123024.3123157

[10] Raghavendra Chalapathy and Sanjay Chawla. 2019. DEEP LEARNING FOR ANOMALY DETECTION: A SURVEY A PREPRINT. (2019).

[11] Zhangyu Cheng, Chengming Zou, and Jianwei Dong. 2019. Outlier Detection using Isolation Forest and Local Outlier Factor. *Proceedings of International*

Conference on Research in Adaptive and Convergent Systems (2019). https://doi.org/10.1145/3338840.3355641
[12] Abeer Elbahrawy, Laura Alessandretti, Anne Kandler, Romualdo Pastor-Satorras, and Andrea Baronchelli. 2017. Evolutionary dynamics of the cryptocurrency market. *Royal Society Open Science* 4, 11 (11 2017). https://doi.org/10.1098/rsos.170623
[13] Tsuyoshi Idé. 2018. Collaborative Anomaly Detection on Blockchain from Noisy Sensor Data; Collaborative Anomaly Detection on Blockchain from Noisy Sensor Data. (2018). https://doi.org/10.1109/ICDMW.2018.00024
[14] Joana Lorenz, Maria Inês Silva, David Aparício, Feedzai João Tiago Ascensão, Feedzai Pedro Bizarro, and João Tiago Ascensão. 2020. Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity. (2020). https://doi.org/10.1145/3383455
[15] Thai T Pham and Steven Lee. 2016. Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods. *Pham, Thai and Lee, Steven* (2016). https://doi.org/10.48550/arXiv.1611.03941
[16] Stephen Ranshous, Shitian Shen, Danai Koutra, Christos Faloutsos, and Nagiza F Samatova. 2015. Anomaly Detection in Dynamic Networks: A Survey. *WIREs Comput Stat* 7 (2015), 223–247. https://doi.org/10.1002/wics.1347
[17] Matteo Signorini, Matteo Pontecorvi, Wael Kanoun Thales UAE, Abu Dhabi, and Roberto Di Pietro. 2020. BAD: A BLOCKCHAIN ANOMALY DETECTION SOLUTION ARXIV E-PRINT. (2020). https://coinmarketcap.com/
[18] Yang Xiao, Ning Zhang, Wenjing Lou, and Y Thomas Hou. 2020. A Survey of Distributed Consensus Protocols for Blockchain Networks. *IEEE Communications Surveys & Tutorials* 22, 2 (2020), 1432–1465. https://doi.org/10.1109/COMST.2020.2969706
[19] Jennifer J Xu. 2016. Are blockchains immune to all malicious attacks? *Financial Innovation* 2, 25 (2016). https://doi.org/10.1186/s40854-016-0046-5
[20] Ruizhe Yang, F Richard Yu, Pengbo Si, Senior Member, Zhaoxin Yang, and Yanhua Zhang. 2019. Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS* 21, 2 (2019). https://doi.org/10.1109/COMST.2019.2894727
[21] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. (2017). https://doi.org/10.1109/BigDataCongress.2017.85

A APPENDIX

A.1 Python code

The link with the complete Python implementation on Google Colab can be here: https://colab.research.google.com/drive/1vTozweLs7JvK.

A.2 Bitcoin transactions data

The database with transactions from different years (2011 in our case) is accessible through this link: https://dataverse.harvard.edu/dataset

A.3 Plots

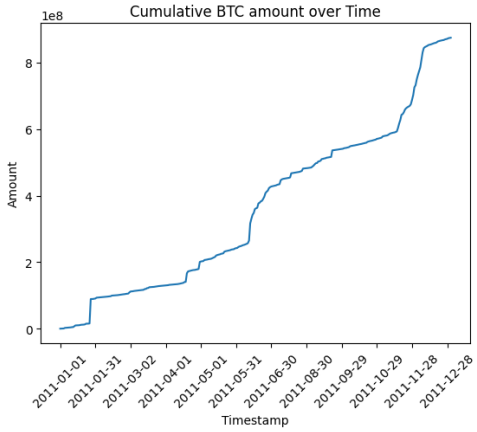

Fig. 1. The increase in the amount of Bitcoin transferred over a year

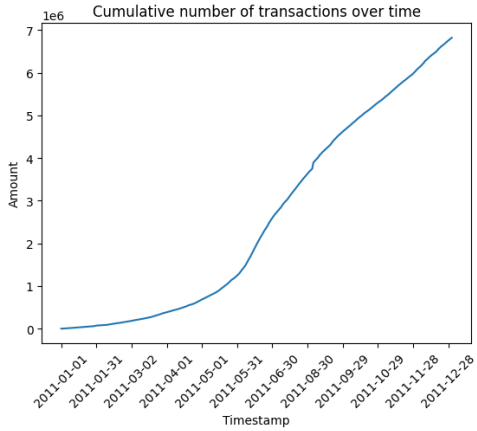

Fig. 2. The increase in the number of Bitcoin transactions over a year

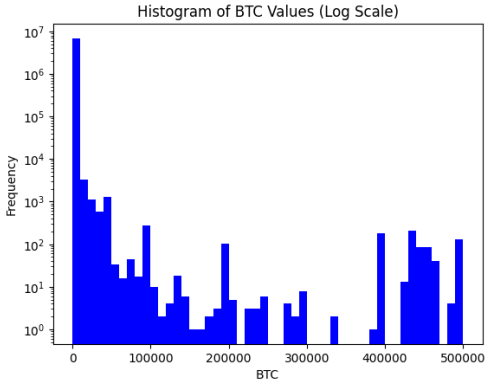

Fig. 3. The distribution of (logarithmic) Bitcoin values within the data

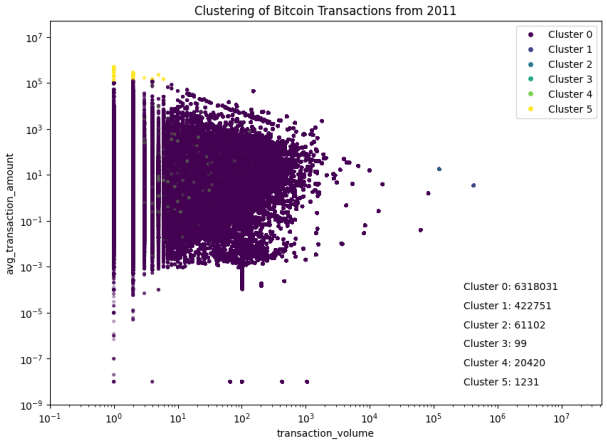

Fig. 4. K-means clustering applied to all transactions from 2011 is not that suggestive due to the enormous amount of data points

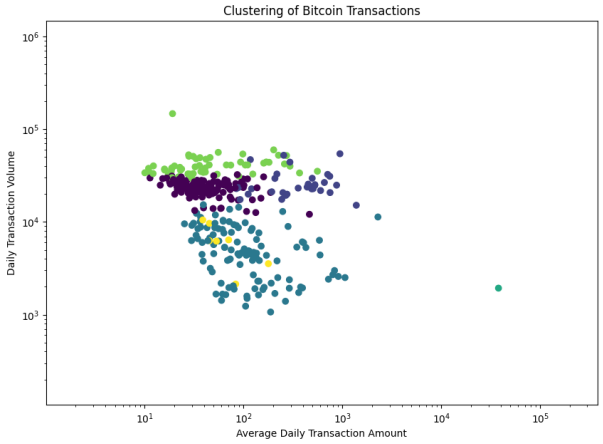

Fig. 5. Clusters of daily metrics

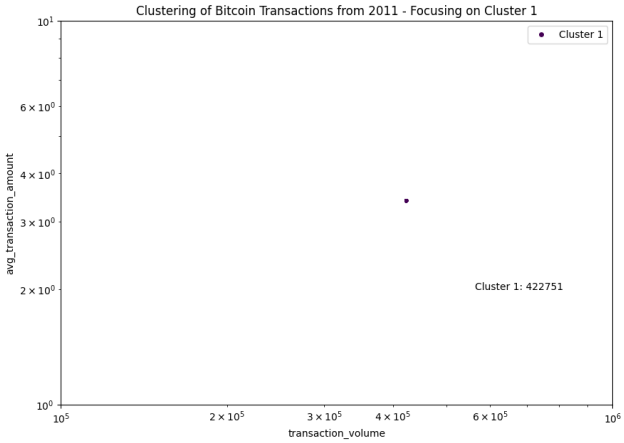

Fig. 7. Clustering with only Cluster 1 data points shows the overlap in this cluster

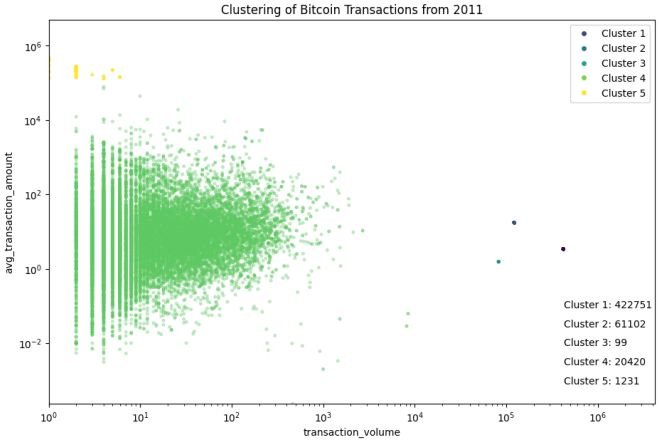

Fig. 6. Clustering without Cluster 0 data points