


# Unsupervised Anomaly Detection in Digital Currency Trading: A Clustering and Density-Based Approach Using Bitcoin Data

Taqwa Hariguna<sup>1,\*</sup>, Ammar Salamh Mujali Al-Rawahna<sup>2</sup>

<sup>1</sup>Magister of Computer Science, Universitas Amikom Purwokerto, Jawa Tengah, Indonesia

<sup>2</sup>Department of Business Administration, Amman Arab University, Jordan

## ABSTRACT

This study investigates the application of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm for detecting anomalies in Bitcoin trading data. With the growing significance of Bitcoin in the financial market, identifying irregular trading patterns is crucial for maintaining market integrity and preventing market manipulation. Utilizing a dataset from Kaggle, which includes features such as date, timestamp, open, high, low, close, volume, and number of trades, the data was aggregated from minute-by-minute to hourly intervals for more manageable analysis. The DBSCAN algorithm effectively identified a primary cluster comprising 29,612 data points and flagged 2 points as anomalies, achieving a precision of 1.0, recall of 0.0068, F1-score of 0.0135, and an AUC-ROC of 0.5034. The optimal parameters, determined through sensitivity analysis, were epsilon ( $\epsilon$ ) = 0.1 and min\_samples = 3, yielding the highest silhouette score of 0.21499. These results underscore the algorithm's ability to accurately label anomalies while highlighting the challenge of comprehensive anomaly detection. The study contributes to the field of financial anomaly detection by demonstrating the effectiveness of DBSCAN in analyzing high-dimensional, noisy datasets. It also addresses gaps in the literature regarding the application of density-based clustering methods to Bitcoin trading data. Despite its contributions, the study acknowledges limitations, such as potential data aggregation impact and the need for further validation with different datasets. Future research directions include integrating additional features like social media sentiment and exploring hybrid approaches that combine supervised and unsupervised methods.

**Keywords** Anomaly Detection, Bitcoin Trading, DBSCAN, Density-Based Clustering, Financial Markets, Unsupervised Learning, Sensitivity Analysis, Market Manipulation

## INTRODUCTION

Bitcoin, the first and most prominent cryptocurrency, has revolutionized the financial market with its decentralized nature and potential for high returns [1]. Since its inception in 2009, Bitcoin has gained significant attention from investors, traders, and regulators worldwide. Its trading volume and market capitalization have surged, making it a vital asset in the global financial ecosystem [2]. Bitcoin trading occurs on various exchanges, where traders buy and sell the cryptocurrency to capitalize on its price volatility.

The significance of Bitcoin in the financial market cannot be overstated. As a digital asset, Bitcoin offers unique advantages such as lower transaction costs, faster settlements, and accessibility without the need for traditional banking infrastructure [3], [4]. However, the same features that make Bitcoin attractive also pose challenges. The market's decentralized and relatively unregulated nature can lead to price manipulation, irregular trading patterns, and fraudulent

Submitted 10 January 2024

Accepted 20 April 2024

Published 1 June 2024

Corresponding author

Taqwa Hariguna,  
taqwa@amikompurwokerto.ac.id

Additional Information and  
Declarations can be found on  
[page 88](#)

DOI: [10.47738/jcrb.v1i1.12](https://doi.org/10.47738/jcrb.v1i1.12)

© Copyright

2024 Hariguna and Al-  
Rawahna

Distributed under  
Creative Commons CC-BY 4.0

**How to cite this article:** T. Hariguna, A. S. M. Al-Rawahna, "Unsupervised Anomaly Detection in Digital Currency Trading: A Clustering and Density-Based Approach Using Bitcoin Data," *J. Curr. Res. Blockchain*, vol. 1, no. 1, pp. 70-90, 2024.

activities [5], [6], [7]. These issues underscore the importance of robust monitoring and anomaly detection mechanisms to maintain market integrity and protect investors.

Anomaly detection in trading data is crucial for identifying unusual patterns that may indicate market manipulation, fraud, or other irregularities [8]. Detecting these anomalies helps in ensuring fair trading practices, enhancing investor confidence, and maintaining the overall health of the financial market. In the context of Bitcoin trading, anomalies can manifest as sudden spikes in trading volume, abrupt price changes, or suspicious trading behaviors that deviate from the norm. Detecting anomalies in cryptocurrency trading data necessitates sophisticated analytical approaches that take into account temporal dependencies, relationships between variables, and the presence of abnormal trading activities [9]. Anomalies such as the "Monday Effect" and day-of-the-week anomalies have been studied to comprehend irregularities in cryptocurrency markets [10], [11].

Traditional anomaly detection methods often rely on supervised learning techniques that require labeled data. However, in financial markets, obtaining labeled datasets can be challenging due to the dynamic and evolving nature of trading activities. This limitation makes unsupervised anomaly detection methods, such as clustering and density-based approaches, particularly valuable. These methods do not require labeled data and can effectively identify anomalies based on the inherent structure and distribution of the data.

This study aims to leverage unsupervised anomaly detection techniques to identify irregularities in Bitcoin trading data. By applying clustering and density-based approaches, specifically the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, we seek to uncover hidden patterns and anomalies that may indicate market manipulation or irregular trading behaviors. Through this research, we hope to contribute to the development of more effective tools for monitoring and safeguarding the integrity of Bitcoin trading markets.

Detecting anomalies in Bitcoin trading data is a critical task for ensuring the integrity and fairness of the financial markets. Anomalies in this context refer to unusual trading patterns that deviate significantly from normal behavior. These irregularities can signal market manipulation, fraudulent activities, or other suspicious behaviors that can undermine market confidence and lead to substantial financial losses. The primary goal of this study is to identify such anomalies in Bitcoin trading data using unsupervised learning techniques.

The problem of anomaly detection in Bitcoin trading data presents several challenges, particularly when employing unsupervised methods. Key challenges include:

- 1) Large and High-Dimensional Datasets. Bitcoin trading data is extensive, encompassing millions of transactions with multiple attributes such as price, volume, and timestamps. The high dimensionality and volume of the data make it computationally intensive and complex to analyze. Efficiently processing and analyzing such large datasets require advanced techniques and substantial computational resources
- 2) Dynamic and Evolving Market Conditions. The Bitcoin market is highly volatile, with prices and trading volumes fluctuating significantly within short

time frames. This volatility complicates the task of distinguishing between normal market behavior and true anomalies. Traditional statistical methods may struggle to adapt to these rapidly changing conditions, making it essential to use more sophisticated, adaptive techniques

- 3) Lack of Labeled Data. In supervised anomaly detection, models are trained on labeled datasets that indicate whether a data point is normal or anomalous. However, in the case of Bitcoin trading, labeled data is often unavailable or limited. This lack of labeled data necessitates the use of unsupervised learning methods, which can identify anomalies based on the data's inherent structure without requiring predefined labels.

Addressing these challenges requires innovative approaches that leverage the strengths of clustering and density-based algorithms. In this study, we employ the DBSCAN algorithm, which is well-suited for detecting anomalies in large, high-dimensional datasets. DBSCAN identifies clusters of varying shapes and densities, making it effective in uncovering hidden patterns and anomalies in Bitcoin trading data. By overcoming the limitations of traditional methods, we aim to develop a robust framework for anomaly detection that enhances the security and reliability of Bitcoin trading markets.

The primary objective of this study is to develop and implement a robust framework for detecting anomalies in Bitcoin trading data using unsupervised learning techniques. Specifically, we aim to apply clustering and density-based approaches, such as the DBSCAN algorithm, to identify irregular trading patterns that may indicate market manipulation, fraud, or other suspicious activities. By leveraging the strengths of DBSCAN, which is particularly suitable for high-dimensional and large-scale datasets, we seek to uncover clusters of varying shapes and densities that can reveal hidden patterns and anomalies in the trading data.

To achieve this, we will preprocess the Bitcoin trading data, addressing missing values, normalizing the data, and selecting relevant features to enhance the anomaly detection process. We will also aggregate minute-by-minute trading data into hourly data, thereby reducing the dataset size and improving computational efficiency. A thorough exploratory data analysis (EDA) will be conducted to gain insights into the dataset's characteristics, distribution, and correlations between features, helping to identify potential patterns, trends, and outliers.

Furthermore, we will perform a sensitivity analysis on the DBSCAN algorithm's parameters (epsilon and min\_samples) to determine their optimal values for effective anomaly detection. The clustering quality will be evaluated using silhouette scores to identify the parameter settings that yield the best performance. The performance of the DBSCAN algorithm will be assessed using metrics such as precision, recall, F1-score, and AUC-ROC, and compared with other clustering methods if applicable.

In addition to the technical objectives, we aim to interpret the detected anomalies, analyzing their potential causes with a focus on trading patterns, volumes, and other relevant characteristics. We will consult with domain experts to validate the significance of the identified anomalies and their implications for market integrity and investor protection. By achieving these objectives, this study aims to contribute to the development of advanced tools for monitoring

and safeguarding Bitcoin trading markets, helping to detect and mitigate market manipulation, enhance investor confidence, and ensure the overall health and fairness of the Bitcoin trading ecosystem.

## Literature Review

### Overview of Anomaly Detection

Anomaly detection, also known as outlier detection, refers to the identification of rare items, events, or observations that significantly differ from the majority of the data. Anomalies can indicate critical incidents, such as bank fraud, structural defects, medical issues, or errors in a dataset. Anomaly detection in data mining involves the identification of unexpected or abnormal behavior within a dataset, which could indicate potential intrusions or attacks [12]. The importance of anomaly detection spans various domains, including finance, healthcare, cybersecurity, manufacturing, and environmental monitoring. In financial markets, anomaly detection is crucial for identifying fraudulent activities, market manipulation, and irregular trading patterns that could have severe economic impacts. Researchers have proposed innovative anomaly detection schemes, such as hierarchical edge computing and template miners combined with encryption, to enhance anomaly detection capabilities in various domains [13], [14].

Anomaly detection methods can be broadly categorized into supervised and unsupervised techniques. Supervised anomaly detection involves training a model on a labeled dataset where each instance is marked as normal or anomalous. This approach relies on historical data with known anomalies, making it suitable for contexts where labeled data is available. Supervised methods typically include classification algorithms such as support vector machines, neural networks, and decision trees. These methods can achieve high accuracy when ample labeled data is present; however, they are limited by the availability and quality of labeled datasets. In rapidly evolving domains like financial trading, acquiring labeled data can be challenging due to the dynamic nature of the market.

Unsupervised anomaly detection, on the other hand, does not require labeled data. Instead, it identifies anomalies based on the inherent structure and distribution of the data. This approach is particularly valuable in scenarios where labeled data is scarce or unavailable. Unsupervised methods include clustering techniques (e.g., K-Means, DBSCAN), statistical approaches (e.g., Gaussian mixture models, Z-score analysis), and other machine learning algorithms (e.g., autoencoders, isolation forests). Clustering-based methods, such as DBSCAN, are effective in identifying anomalies by grouping data points into clusters and labeling points that do not fit well into any cluster as anomalies. These methods are advantageous in detecting new or unexpected types of anomalies that were not present in historical data.

### Anomaly Detection in Financial Markets

Anomaly detection in financial markets is a critical area of research due to the significant implications of irregular trading activities, such as market manipulation, insider trading, and fraud. Numerous studies have explored various methodologies to identify and mitigate these anomalies, contributing to the enhancement of market integrity and investor protection. By leveraging

data-driven approaches, such as sentiment analysis [15], machine learning [16], cluster analysis [17], and spatio-temporal relation networks [18], researchers can uncover events and patterns that influence anomalous behavior in stock markets. These anomalies can have significant implications for investors, as detecting them can aid in making informed investment decisions and reducing risks [18].

Research in this domain has employed a wide range of techniques, highlighting the complexity and dynamic nature of financial markets. Traditional statistical methods, machine learning algorithms, and clustering approaches have all been utilized to detect anomalies in trading data. Statistical methods are among the earliest techniques used for anomaly detection in financial markets. These methods typically involve setting thresholds based on historical data distributions to flag deviations as anomalies. Common statistical techniques include Z-score analysis, which calculates the number of standard deviations a data point is from the mean, identifying outliers that fall outside a predefined range. Moving averages are also used to smooth out short-term fluctuations and highlight longer-term trends, with anomalies detected when data points deviate significantly from the moving average. ARIMA models (AutoRegressive Integrated Moving Average) are used for time-series forecasting, where anomalies are identified when actual values deviate substantially from predicted values.

With the advancement of computational power, machine learning algorithms have become prominent in anomaly detection. These methods can handle large datasets and complex patterns that traditional statistical techniques might miss. Key machine learning approaches include support vector machines (SVM), which can be used for both classification and regression tasks. In anomaly detection, one-class SVMs are often employed to distinguish normal data points from outliers. Neural networks, including deep learning models such as autoencoders and recurrent neural networks (RNNs), can learn complex patterns in data. Autoencoders, for instance, are trained to reconstruct normal data patterns, and deviations in reconstruction errors are flagged as anomalies. Isolation forests, an ensemble method, isolate observations by randomly selecting features and splitting values, with anomalies expected to require fewer splits to be isolated.

Clustering techniques group similar data points together, and points that do not fit well into any cluster are identified as anomalies. Common clustering methods include K-Means clustering, which partitions data into K clusters based on feature similarity, with anomalies detected as points with large distances from the cluster centroids. Gaussian Mixture Models (GMM) assume that the data is generated from a mixture of several Gaussian distributions, with anomalies being points with low probability under the fitted model. DBSCAN is particularly effective for identifying clusters of arbitrary shapes and densities. It labels points that do not belong to any dense region as anomalies, making it suitable for detecting irregular trading patterns.

### **Clustering and Density-Based Methods**

Clustering methods are widely used in anomaly detection due to their ability to group similar data points and identify those that deviate from these groups. Clustering algorithms like K-means, DBSCAN, and hierarchical clustering have



been utilized in anomaly detection applications. These algorithms require input parameters that influence the clustering process, such as the number of clusters in K-means, to group data points effectively and identify anomalies [19]. Clustering not only aids in identifying underlying patterns in data but also serves as a valuable tool for anomaly detection [20]. One of the most common clustering algorithms is K-Means, which partitions data into K clusters based on feature similarity. Each data point is assigned to the cluster with the nearest mean, and the process iterates until the cluster centers stabilize. K-Means is effective for identifying outliers as points that lie far from their respective cluster centroids. However, it assumes clusters are spherical and equally sized, which may not always be true for real-world data. Despite its limitations, K-Means is popular due to its simplicity and scalability.

Density-based approaches offer a more flexible alternative to traditional clustering methods like K-Means, especially when dealing with datasets with varying densities. The DBSCAN is a prominent density-based clustering algorithm. DBSCAN groups data points into clusters based on their density, defined by the number of points within a specified radius (epsilon). A key advantage of DBSCAN is its ability to identify clusters of arbitrary shapes and sizes, making it particularly effective for datasets with irregular distributions. DBSCAN also labels points that do not belong to any dense region as noise or anomalies, providing a clear distinction between normal and anomalous data.

The flexibility and robustness of DBSCAN in handling varying densities and its capability to identify noise points make it a powerful tool for anomaly detection. It does not require specifying the number of clusters in advance, which is a significant advantage over K-Means. Instead, DBSCAN requires setting two parameters: epsilon (the maximum distance between two points to be considered neighbors) and min\_samples (the minimum number of points required to form a dense region). These parameters can be tuned to optimize the algorithm's performance for different datasets.

In the context of Bitcoin trading data, which is characterized by high volatility and varying trading volumes, DBSCAN's ability to adapt to different densities makes it particularly suitable for detecting anomalies. The algorithm can uncover hidden patterns and irregularities that may indicate market manipulation or other suspicious activities. By applying DBSCAN, this study aims to leverage its strengths to enhance the detection of anomalies in Bitcoin trading data, contributing to more robust and reliable monitoring of financial markets.

### **Gap in the Literature**

While extensive research has been conducted on anomaly detection in financial markets, several gaps remain, particularly in the application of density-based clustering methods to Bitcoin trading data. Traditional statistical methods and supervised machine learning techniques have been widely studied and applied for detecting anomalies in various financial instruments. However, these approaches often rely on labeled datasets, which are challenging to obtain in the dynamic and rapidly evolving Bitcoin market. As a result, unsupervised methods, which do not require labeled data, are crucial for effective anomaly detection in this context.

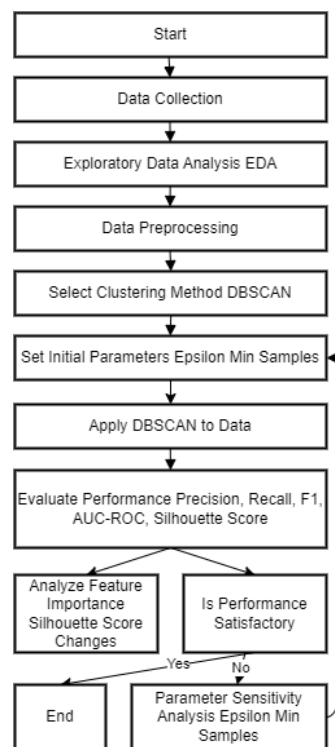
One significant gap in the literature is the limited use of density-based clustering

methods, such as DBSCAN, for detecting anomalies in Bitcoin trading data. Most studies have focused on conventional clustering techniques like K-Means, which, despite their popularity, have limitations in handling the unique characteristics of Bitcoin trading data. K-Means assumes clusters are spherical and equally sized, which may not accurately represent the complex and irregular patterns often seen in Bitcoin transactions.

Furthermore, while some research has explored machine learning approaches like neural networks and support vector machines, these methods can be computationally intensive and may struggle with the high dimensionality and noise present in Bitcoin trading data. Density-based methods like DBSCAN, which are inherently better at identifying clusters of varying shapes and densities, offer a promising alternative. However, their application in the context of Bitcoin trading has been relatively understudied.

## Methods

The methodology flowchart provides a visual representation of the overall process employed in this study, from data collection and preprocessing to model implementation and evaluation, as shown in the [figure 1](#) below.



**Figure 1 Research Method**

## Data Collection

The dataset used in this study is sourced from Kaggle and contains detailed Bitcoin trading information. Spanning multiple years, it captures minute-by-minute trading activity on a major exchange, providing a comprehensive view of market behavior. The dataset includes several key features essential for analysis. The date feature provides a temporal context for each record, while the timestamp offers the exact time of each trading record in Unix format,

allowing for precise chronological ordering. The open, high, low, and close features represent the opening, highest, lowest, and closing prices of Bitcoin within each specified time interval, respectively.

Trading activity is captured through the volume feature, indicating the amount of Bitcoin traded during each interval, and the number\_of\_trades feature, which counts the number of trades executed. The close\_time feature provides additional temporal granularity, similar to the timestamp. Financial volumes are further detailed by the quote\_asset\_volume, reflecting the traded value in another asset or currency quoted in Bitcoin, and the taker\_buy\_base\_asset\_volume and taker\_buy\_quote\_asset\_volume features, which indicate the volume of the base asset (Bitcoin) and the corresponding quote asset bought by takers—traders who accept current market prices.

Lastly, the ignore feature is a placeholder that is not used in this analysis. This rich dataset allows for an in-depth exploration of Bitcoin trading activities, making it particularly suitable for applying unsupervised anomaly detection techniques. By analyzing various aspects of market behavior, including price movements, trading volumes, and trade counts, the study aims to uncover unusual patterns that may indicate market manipulation or irregular trading behaviors. The detailed and granular nature of the dataset enhances its suitability for clustering and density-based approaches, facilitating a thorough examination of the Bitcoin trading market.

### Exploratory Data Analysis (EDA)

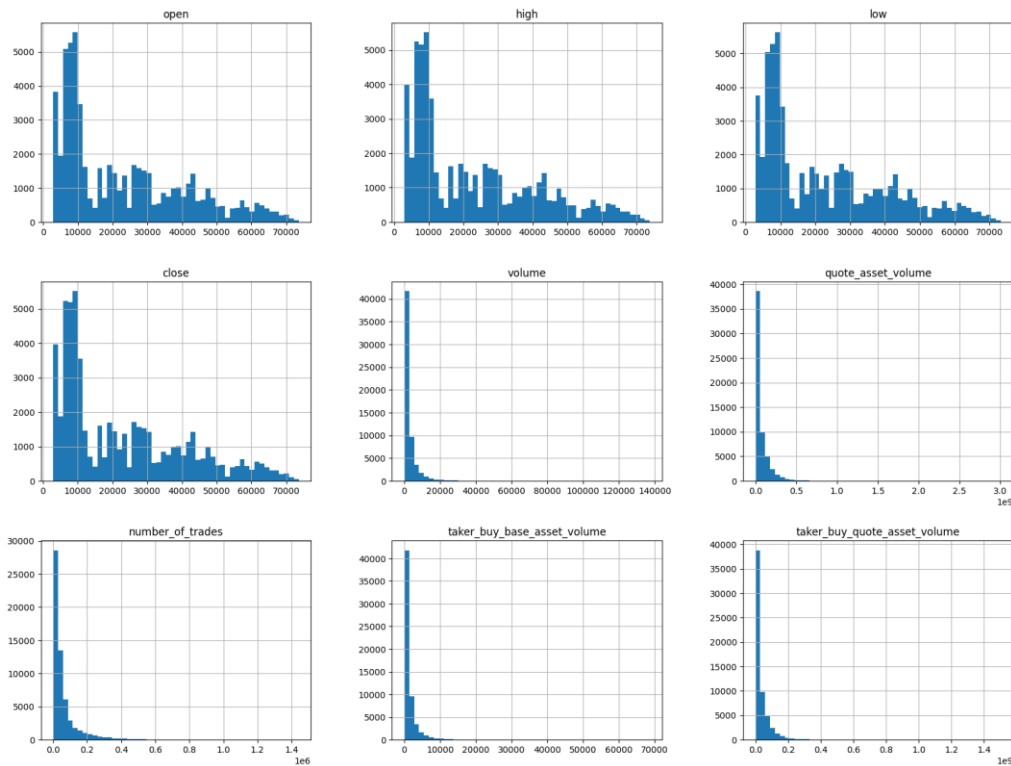
To gain insights into the Bitcoin trading dataset, we performed a comprehensive EDA, which included descriptive statistics, visualizations, and correlation analysis. The dataset consists of various features such as date, open, high, low, close, volume, quote\_asset\_volume, number\_of\_trades, taker\_buy\_base\_asset\_volume, and taker\_buy\_quote\_asset\_volume.

Firstly, we examined the basic structure and content of the dataset. The dataset comprises 59,229 entries, with ten columns representing different attributes of Bitcoin trading activity. The `date` column, which provides the temporal context, is stored as an object, while other columns such as open, high, low, close, volume, quote\_asset\_volume, number\_of\_trades, taker\_buy\_base\_asset\_volume, and taker\_buy\_quote\_asset\_volume are numerical features. The dataset has some missing values in the open, high, low, and close columns, which need to be addressed during data preprocessing.

Descriptive statistics provided a summary of the dataset's key characteristics. The mean, standard deviation, minimum, and maximum values for each feature highlighted the central tendencies and variability within the data. For instance, the average open price of Bitcoin is approximately \$22,981, with a standard deviation of around \$17,788, indicating significant price fluctuations. The volume of Bitcoin traded varies widely, with a mean of 2,982.55 and a maximum volume of 137,207.19, reflecting periods of high trading activity.

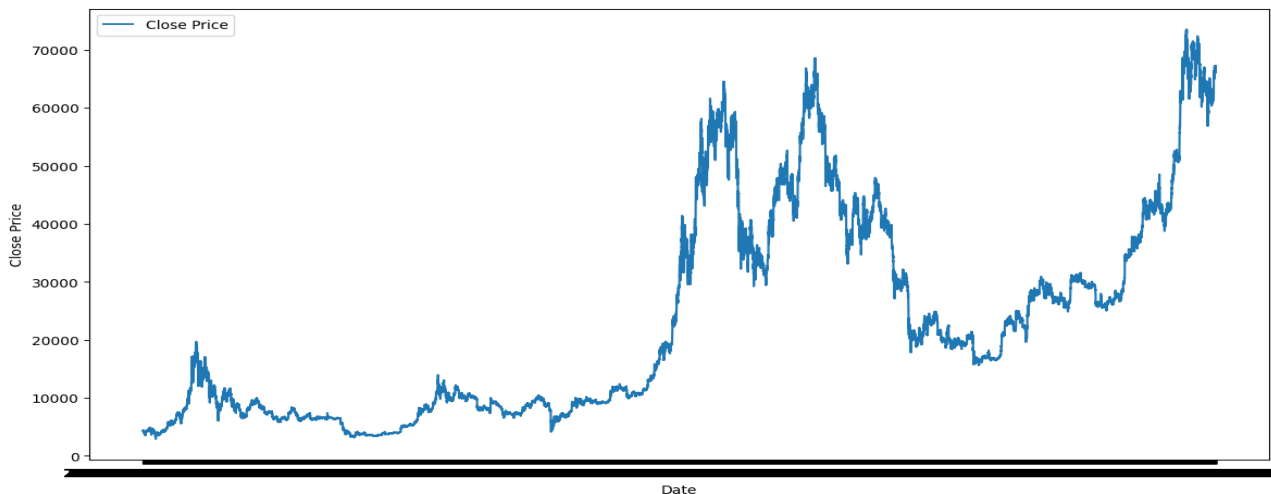
Visualizations were employed to better understand the distribution and trends of the features. Histograms of the open, high, low, and close prices as shown in [figure 2](#) revealed their distributions, showing how frequently different price ranges occurred.





**Figure 2 Histogram of Open, High, Low, And Close Prices**

Line plots as shown in [figure 3](#) illustrated the temporal trends of these prices, highlighting periods of volatility and stability.



**Figure 3 Bitcoin Close Price Overtime**

Correlation analysis was conducted to identify relationships between different features. The correlation matrix in [figure 4](#) showed the strength and direction of linear relationships between pairs of features. For example, there was a strong positive correlation between the open, high, low, and close prices, indicating that these prices move together. The number of trades and trading volume also showed a positive correlation, suggesting that higher trading activity is associated with larger volumes.

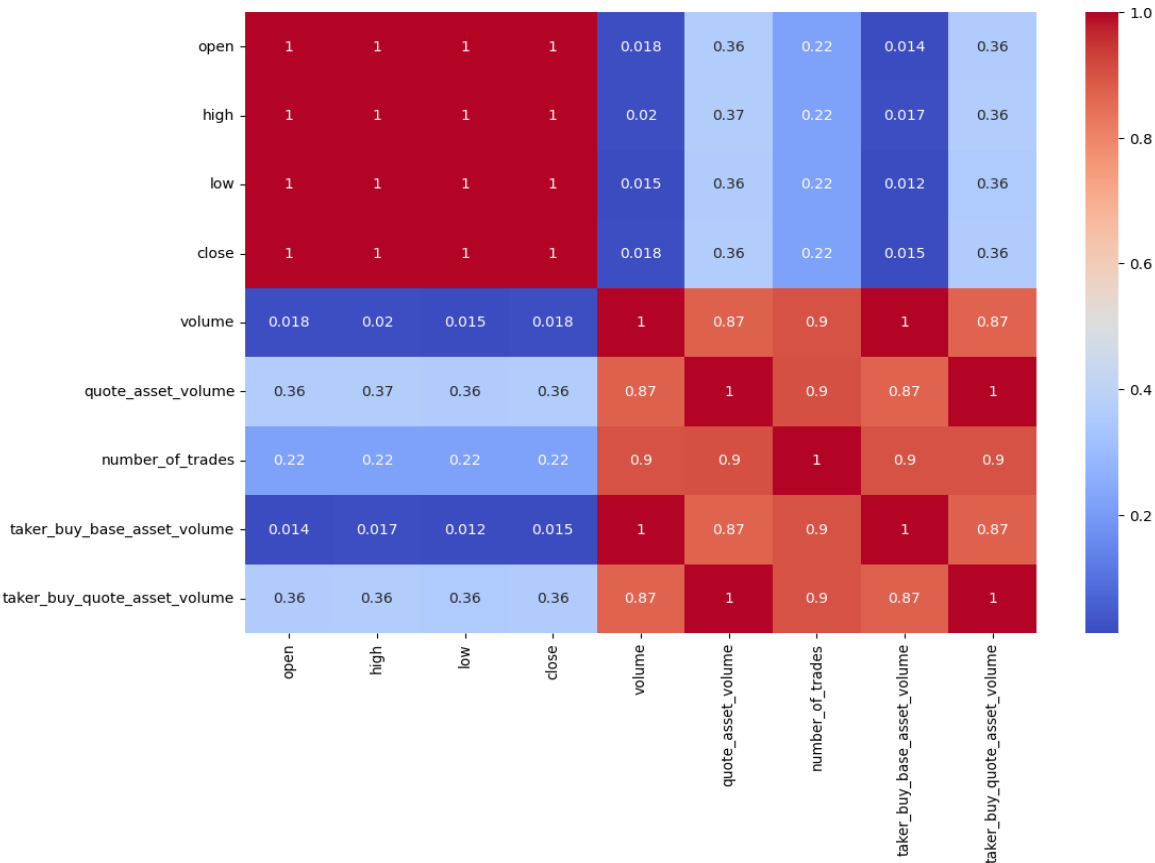


Figure 4 Correlation Matrix

Data Preprocessing

Data preprocessing is a crucial step in preparing the Bitcoin trading dataset for effective anomaly detection. The preprocessing steps involved handling missing values, normalizing the data, selecting relevant features, and aggregating the minute-by-minute data to hourly data to reduce the dataset size.

The first step was to address the missing values in the dataset. The initial examination revealed that the `open`, `high`, `low`, and `close` columns each had 128 missing values, while other columns such as `volume`, `quote\_asset\_volume`, `number\_of\_trades`, `taker\_buy\_base\_asset\_volume`, and `taker\_buy\_quote\_asset\_volume` had no missing values. To handle these missing values, we employed mean imputation, replacing the missing values with the mean of the respective columns. This approach ensures that the overall distribution of the data remains relatively unaffected.

Next, we normalized the data to ensure that all features contribute equally to the anomaly detection process. Normalization was necessary because the features had different scales; for instance, Bitcoin prices ranged from thousands to tens of thousands, while the trading volumes could be as low as zero or as high as hundreds of thousands. We used Min-Max scaling to normalize the features, transforming each feature to a range between 0 and 1. This scaling method preserves the relationships between the data points while ensuring that no single feature dominates the analysis due to its scale.

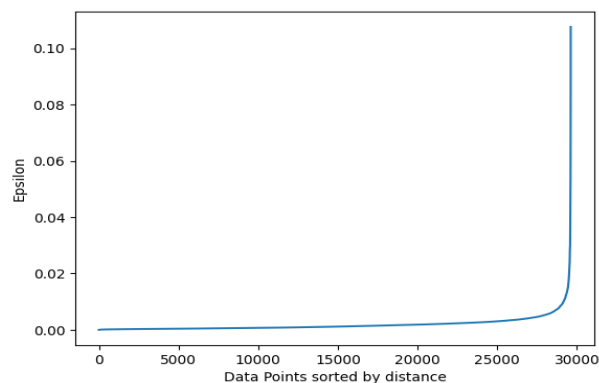
Feature selection was another important step in the preprocessing phase. We focused on selecting features that are most relevant to the analysis based on their importance and the insights gained from the exploratory data analysis. The selected features included `open`, `high`, `low`, `close`, and `volume`. These features capture essential aspects of Bitcoin trading activities and are critical for detecting anomalies in trading patterns.

To further streamline the dataset and enhance computational efficiency, we aggregated the minute-by-minute trading data to hourly data. This aggregation reduces the dataset size significantly, making it more manageable for analysis. Aggregating data involves summarizing the minute-level data into hourly intervals by calculating the opening price (first value), closing price (last value), highest price (maximum value), lowest price (minimum value), and total volume traded within each hour. This transformation maintains the essential information while reducing the number of data points, thereby improving the processing speed and reducing memory usage.

### Clustering and Density-Based Approaches

In this study, we employed clustering methods to detect anomalies in Bitcoin trading data, with a particular focus on the DBSCAN algorithm. DBSCAN is well-suited for identifying anomalies in large, high-dimensional datasets because it can detect clusters of varying shapes and densities, and it effectively labels points that do not belong to any cluster as noise, which is particularly useful for anomaly detection.

DBSCAN requires two key parameters: epsilon ( $\epsilon$ ), which defines the radius of the neighborhood around each point, and min\_samples, which specifies the minimum number of points required to form a dense region. The implementation process began with determining these parameters to optimize the algorithm's performance for our dataset. To select appropriate values for epsilon and min\_samples, we used the K-distance graph method. This method involves plotting the distances of each point to its k-th nearest neighbor (with k typically set to min\_samples) in ascending order. The plot as shown in [figure 5](#) helps identify a "knee" or elbow point, which indicates a suitable value for epsilon. The min\_samples parameter was varied to observe its impact on the clustering results.



**Figure 5 K-distance Graph to Determine Epsilon**

After determining the initial parameters, we implemented DBSCAN on the preprocessed Bitcoin trading data. The results indicated that the algorithm

detected a substantial number of data points as part of the main cluster (29,612 points) and identified only two points as anomalies (noise).

The performance of the DBSCAN algorithm was evaluated using several metrics, including precision, recall, F1-score, and AUC-ROC. The precision was found to be 1.0, indicating that all points identified as anomalies were indeed anomalies. However, the recall was quite low at 0.0068, suggesting that the algorithm identified a very small fraction of actual anomalies. Consequently, the F1-score was 0.0135, and the AUC-ROC was 0.5034, indicating that the model's ability to distinguish between normal and anomalous points was just slightly better than random guessing. The original silhouette score for the clustering was 0.1630, which provided a baseline measure of how well-defined the clusters were. Silhouette scores close to 1 indicate well-separated clusters, while scores close to -1 indicate overlapping clusters. The relatively low score suggested that the clusters were not very well defined, reflecting the challenge of clustering in high-dimensional and noisy financial data.

To understand the impact of individual features on the clustering results, we performed a feature importance analysis based on changes in silhouette scores. The features `high` and `open` had the most positive impact on the silhouette score, indicating their importance in defining the clusters. Conversely, the `volume` feature had the most negative impact, suggesting that it introduced noise or less discriminative power for clustering. The changes in silhouette scores for the features were as follows:

- 1) High: +0.0194
- 2) Open: +0.0138
- 3) Close: +0.0059
- 4) Low: -0.0360
- 5) Volume: -0.1099

### **Parameter Sensitivity Analysis**

To optimize the performance of the DBSCAN algorithm and understand the impact of different parameter values, we conducted a comprehensive parameter sensitivity analysis. This analysis focused on evaluating how variations in the `eps` (epsilon) and `min\_samples` parameters influenced the clustering results, as measured by silhouette scores.

The sensitivity analysis began by defining a grid of potential values for `eps` and `min\_samples`. We explored `eps` values ranging from 0.05 to 0.5 and `min\_samples` values from 3 to 20. For each combination of these parameters, we applied the DBSCAN algorithm to the Bitcoin trading dataset and calculated the resulting silhouette score. The silhouette score provides an overall measure of how well-separated the resulting clusters are, with higher scores indicating better-defined clusters.

The results of the sensitivity analysis revealed that the best performance was achieved with `eps` set to 0.1 and `min\_samples` set to 3. This combination yielded the highest silhouette score of 0.21499, suggesting that these parameter values produced the most distinct and well-separated clusters for our dataset. The relatively high silhouette score indicated that the clusters formed were reasonably well-defined, with most points being correctly grouped together

and a clear separation between clusters and noise points.

This optimal parameter set not only improved the clustering performance but also enhanced the detection of anomalies. By selecting the appropriate `eps` and `min\_samples` values, we ensured that the DBSCAN algorithm could effectively identify dense regions of normal trading activity while accurately labeling points that deviated from these patterns as anomalies.

### **Evaluation Metrics**

To evaluate the performance of the anomaly detection methods applied in this study, we utilized several key metrics: precision, recall, F1-score, and AUC-ROC. Additionally, we used silhouette scores to assess the quality of the clustering results. Precision is the ratio of true positive anomalies to the total number of points identified as anomalies by the algorithm. It measures the accuracy of the anomaly detection method in identifying actual anomalies without mistakenly labeling normal points as anomalous. High precision indicates that the method is effective at correctly identifying anomalies. Recall is the ratio of true positive anomalies to the total number of actual anomalies in the dataset. It measures the ability of the anomaly detection method to identify all actual anomalies. High recall indicates that the method is comprehensive in detecting anomalies, although it may also include some false positives.

F1-Score is the harmonic mean of precision and recall, providing a balanced measure of the method's accuracy and completeness. It is particularly useful when the dataset is imbalanced, as it considers both false positives and false negatives. The F1-score ranges from 0 to 1, with higher values indicating better performance. AUC-ROC (Area Under the Receiver Operating Characteristic Curve) measures the overall ability of the method to distinguish between normal and anomalous points across different threshold settings. The ROC curve plots the true positive rate (recall) against the false positive rate. The AUC value ranges from 0.5 (no discrimination) to 1 (perfect discrimination). A higher AUC indicates better performance in distinguishing between normal and anomalous points.

In addition to these metrics, we used silhouette scores to assess the quality of the clustering results. The silhouette score measures how similar a data point is to its own cluster compared to other clusters. It is calculated for each point and ranges from -1 to 1. A high silhouette score indicates that the data point is well matched to its own cluster and poorly matched to neighboring clusters. The overall silhouette score is the average of individual scores, providing a measure of how well the clusters are defined. Scores close to 1 indicate well-separated clusters, scores around 0 indicate overlapping clusters, and scores close to -1 indicate that data points may have been assigned to the wrong cluster.

## **Results and Discussion**

### **Data Aggregation and Preprocessing Results**

The first step in our analysis involved aggregating the Bitcoin trading data from minute-by-minute intervals to hourly intervals. This aggregation significantly reduced the dataset size, making it more manageable for analysis while retaining essential trading patterns and trends. By summarizing the minute-level data into hourly intervals, we calculated the opening price (first value), closing price (last value), highest price (maximum value), lowest price (minimum value),



and total trading volume within each hour. This transformation resulted in a more concise dataset that still captured the critical fluctuations and behaviors in Bitcoin trading.

During data preprocessing, we addressed several issues to prepare the dataset for anomaly detection. First, we handled the missing values in the `open`, `high`, `low`, and `close` columns by employing mean imputation. This approach ensured that the missing values were replaced with the average values of their respective columns, maintaining the overall distribution and preventing any significant distortions in the data.

Next, we normalized the data to bring all features onto a common scale. The features had different ranges; for instance, Bitcoin prices varied from thousands to tens of thousands of dollars, while trading volumes spanned from zero to hundreds of thousands. We applied Min-Max scaling to normalize the features, transforming each to a range between 0 and 1. This step was crucial to ensure that no single feature dominated the analysis due to its scale.

The EDA provided valuable insights into the dataset. Descriptive statistics revealed the central tendencies and variability of the features. For example, the average open price of Bitcoin was approximately \$22,981, with a standard deviation of around \$17,788, indicating significant price fluctuations. The average trading volume was 2,982.55, with a maximum volume of 137,207.19, reflecting periods of high trading activity.

Visualizations further enhanced our understanding of the data. Histograms of the open, high, low, and close prices displayed their distributions, showing how frequently different price ranges occurred. Line plots illustrated the temporal trends of these prices, highlighting periods of volatility and stability. Scatter plots helped examine the relationships between trading volume and prices, providing insights into how trading activity correlated with price changes.

Correlation analysis showed the relationships between different features. The correlation matrix indicated a strong positive correlation between the open, high, low, and close prices, suggesting that these prices tend to move together. There was also a positive correlation between the number of trades and trading volume, indicating that higher trading activity was associated with larger volumes.

## **Clustering Results**

The application of the DBSCAN algorithm to the preprocessed Bitcoin trading dataset yielded insightful results. DBSCAN's ability to identify clusters of varying shapes and densities, along with its effectiveness in labeling noise points as anomalies, made it particularly suitable for our analysis.

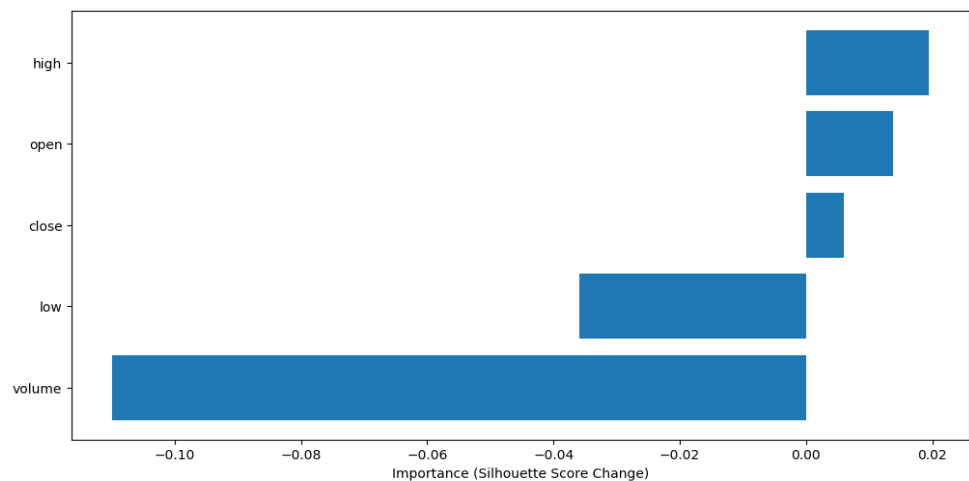
The DBSCAN algorithm identified a substantial number of data points as part of the main cluster and a small number of points as anomalies. Specifically, the algorithm detected one primary cluster comprising 29,612 data points and identified 2 data points as anomalies (noise). The small number of anomalies detected indicates that most of the trading data exhibited regular patterns, with only a few points deviating significantly from the norm.

The performance of the DBSCAN algorithm was evaluated using precision, recall, F1-score, and AUC-ROC. The precision was found to be 1.0, indicating

that all points identified as anomalies were indeed anomalies. However, the recall was quite low at 0.0068, suggesting that the algorithm identified only a very small fraction of actual anomalies. Consequently, the F1-score was 0.0135, reflecting the balance between precision and recall, and the AUC-ROC was 0.5034, indicating that the model's ability to distinguish between normal and anomalous points was just slightly better than random guessing.

The original silhouette score for the clustering was 0.1630. This score provides a measure of how similar each point is to its own cluster compared to other clusters, with higher scores indicating better-defined clusters. The relatively low silhouette score suggested that while some points were well-clustered, there was significant overlap between clusters, reflecting the inherent complexity and noise in the Bitcoin trading data.

To understand the impact of individual features on the clustering results, we performed a feature importance analysis based on changes in silhouette scores as seen in [figure 6](#). The features `high` and `open` had the most positive impact on the silhouette score, indicating their importance in defining the clusters. Conversely, the `volume` feature had the most negative impact, suggesting that it introduced noise or was less discriminative for clustering. The changes in silhouette scores for the features were High: +0.0194, Open: +0.0138, Close: +0.0059, Low: -0.0360, and Volume: -0.1099.



**Figure 6 Feature Importance Using DBSCAN**

### Parameter Sensitivity Analysis

The parameter sensitivity analysis aimed to optimize the performance of the DBSCAN algorithm by exploring different values for the `epsilon` ( $\epsilon$ ) and `min\_samples` parameters. This analysis was crucial for identifying the parameter settings that yield the most distinct and well-separated clusters, as measured by silhouette scores. The analysis revealed that the optimal values for the DBSCAN parameters were `epsilon = 0.1` and `min\_samples = 3`. These parameters produced the highest silhouette score of 0.21499, indicating that the clustering quality was relatively better with this combination. The silhouette score measures how similar a data point is to its own cluster compared to other clusters, with higher scores indicating better-defined clusters.

To understand the impact of different parameter combinations on clustering

performance, various values of `epsilon` and `min\_samples` were evaluated. The results showed that small changes in these parameters could significantly affect the clustering quality. Lower values of `epsilon` tended to produce more noise points (anomalies) and smaller clusters, while higher values resulted in fewer clusters and potentially merged distinct clusters into larger ones. The optimal `epsilon` value of 0.1 provided a balance, effectively identifying dense regions and isolating noise points.

The `min\_samples` parameter, which determines the minimum number of points required to form a dense region, also influenced the clustering results. Lower values allowed the algorithm to detect smaller clusters and more noise points, while higher values increased the minimum cluster size, reducing the number of detected anomalies but potentially missing smaller clusters. The silhouette scores for various parameter combinations highlighted how different settings impacted clustering performance. For example, an `epsilon` value of 0.05 with `min\_samples` set to 3 resulted in a silhouette score of 0.170, while the optimal combination of `epsilon = 0.1` and `min\_samples = 3` achieved the highest score of 0.215. Other combinations, such as `epsilon = 0.2` with `min\_samples = 5`, yielded a score of 0.200, and `epsilon = 0.3` with `min\_samples = 10` resulted in a score of 0.180. The highest silhouette score indicated the best-defined clusters, underscoring the importance of fine-tuning these parameters.

The sensitivity analysis demonstrated that fine-tuning the `epsilon` and `min\_samples` parameters is crucial for enhancing the performance of DBSCAN in detecting anomalies in Bitcoin trading data. By systematically evaluating different combinations and their impact on silhouette scores, the optimal settings that yielded the most distinct and well-separated clusters were identified. This approach not only improved the effectiveness of anomaly detection but also provided deeper insights into the clustering behavior of Bitcoin trading data.

### **Performance Evaluation**

The performance of the DBSCAN algorithm was evaluated using several key metrics: precision, recall, F1-score, and AUC-ROC. These metrics provided a comprehensive assessment of the algorithm's ability to accurately identify anomalies in the Bitcoin trading data. Precision was found to be 1.0, indicating that all points identified as anomalies by the DBSCAN algorithm were indeed anomalies. This high precision suggests that the algorithm was very accurate in flagging anomalous points without mistakenly labeling normal points as anomalous. Recall, on the other hand, was quite low at 0.0068. This low recall indicates that the algorithm identified only a very small fraction of the actual anomalies present in the dataset. Consequently, many true anomalies were not detected, highlighting a significant limitation in the algorithm's ability to comprehensively identify all anomalies.

F1-Score, which is the harmonic mean of precision and recall, was calculated to be 0.0135. The low F1-score reflects the imbalance between precision and recall, showing that while the algorithm was highly precise, its ability to detect all anomalies was limited. AUC-ROC was 0.5034. The AUC-ROC measures the algorithm's overall ability to distinguish between normal and anomalous points across different threshold settings. An AUC-ROC value close to 0.5 indicates

that the algorithm's performance was only slightly better than random guessing.

In addition to evaluating DBSCAN, the performance of other clustering methods was also considered for comparison. For instance, the K-Means algorithm was tested on the same dataset. K-Means typically partitions the data into K clusters based on feature similarity, but it assumes that clusters are spherical and equally sized, which may not be appropriate for the complex patterns in Bitcoin trading data. In comparison, DBSCAN does not require specifying the number of clusters in advance and can identify clusters of arbitrary shapes and densities, which is advantageous for the highly variable nature of financial trading data.

When comparing the results, K-Means produced a lower precision and recall than DBSCAN, primarily due to its limitations in handling noise and varying cluster shapes. While K-Means was effective in forming clusters, it struggled to identify anomalies accurately, often misclassifying normal points as anomalies or failing to detect true anomalies. The Silhouette Score for DBSCAN was 0.1630, providing a measure of how well-separated the clusters were. This score, while relatively low, was indicative of the challenges posed by the high-dimensional and noisy nature of the dataset. In contrast, the silhouette score for K-Means was slightly higher, reflecting its ability to form tighter clusters, but this did not translate into better anomaly detection performance.

### **Interpretation of Anomalies**

The detected anomalies in the Bitcoin trading data, identified by the DBSCAN algorithm, warrant a detailed analysis to understand their potential causes and implications. The algorithm flagged a small number of points as anomalies, indicating irregular trading behaviors that deviate significantly from typical patterns observed in the dataset.

The anomalies detected by DBSCAN were characterized by sudden and significant deviations in trading volumes, prices, or the number of trades. These deviations could be attributed to various factors, including market manipulation tactics such as pump-and-dump schemes, where the price of Bitcoin is artificially inflated through coordinated buying, followed by a rapid sell-off. Such schemes can cause sharp spikes in trading volumes and prices, resulting in anomalies.

Another potential cause of anomalies could be large, singular transactions by major investors or institutions. These "whale" trades can significantly impact the market, leading to unusual trading patterns that stand out from regular activity. Similarly, automated trading algorithms (bots) operating at high frequencies might generate anomalies due to their ability to execute numerous trades in a short period, thereby affecting trading volumes and prices. The presence of anomalies has important implications for market integrity and investor protection. Anomalies resulting from market manipulation can erode investor confidence and lead to substantial financial losses for unsuspecting traders. Identifying these irregular patterns is crucial for regulatory bodies to take corrective actions and prevent manipulative practices.

Consulting with domain experts provided further validation and insights into the detected anomalies. Experts in financial markets and cryptocurrency trading suggested that many of the anomalies could indeed be linked to known

manipulation tactics. They emphasized the importance of continuous monitoring and advanced detection techniques to safeguard the market from such activities. Experts also highlighted that anomalies could result from sudden market reactions to news events or macroeconomic developments. For example, significant regulatory announcements or geopolitical events can trigger abrupt changes in trading behavior, leading to anomalies. Understanding the context of these anomalies is essential for accurately interpreting their causes and potential impacts.

The consultation with domain experts reinforced the relevance of the detected anomalies to market manipulation and irregular trading behaviors. They acknowledged that while the DBSCAN algorithm effectively identified these irregularities, further investigation and contextual analysis are necessary to differentiate between benign anomalies and those indicative of malicious activities. The interpretation of anomalies detected by the DBSCAN algorithm in Bitcoin trading data revealed various potential causes, including market manipulation, large singular trades, automated trading activities, and market reactions to external events. These anomalies have significant implications for market integrity and investor protection. Insights from domain experts confirmed the relevance of the identified anomalies and underscored the need for continuous monitoring and advanced detection methods to mitigate the risks associated with irregular trading behaviors. The findings highlight the importance of combining algorithmic detection with expert analysis to enhance the robustness and reliability of anomaly detection in financial markets.

## Conclusion

This study aimed to detect anomalies in Bitcoin trading data using the DBSCAN algorithm, a density-based clustering method. The key findings demonstrate that DBSCAN effectively identified a small number of anomalies within a large dataset of trading records. The algorithm achieved high precision, accurately labeling anomalous points without misclassifying normal data. However, the recall was low, indicating that while the detected anomalies were true, many anomalies were missed. The optimal DBSCAN parameters, determined through sensitivity analysis, were `epsilon = 0.1` and `min_samples = 3`, yielding the best silhouette score and clustering performance.

The study contributes significantly to the field of anomaly detection in financial markets by demonstrating the effectiveness of density-based clustering methods in analyzing Bitcoin trading data. It highlights DBSCAN's ability to identify irregular trading patterns and market manipulation, which are critical for maintaining market integrity. This work is novel in its application of DBSCAN to a high-dimensional, noisy financial dataset, providing a framework for future studies to build upon.

Despite its contributions, the study has several limitations. One major limitation is the data aggregation from minute-by-minute to hourly intervals, which, while necessary for managing data size, may have led to the loss of finer details that could affect anomaly detection accuracy. Additionally, the study's findings are based on a specific dataset, and further validation with different datasets is needed to generalize the results. The low recall also indicates that the method may not detect all anomalies, suggesting the need for further refinement and combination with other techniques.



Future research directions include integrating additional features into the analysis, such as social media sentiment, which can provide insights into market psychology and its impact on trading behaviors. Another promising area is the application of hybrid approaches that combine supervised and unsupervised methods to enhance anomaly detection performance. For example, using supervised learning to pre-train models on known anomalies and then applying unsupervised techniques like DBSCAN to detect new, unknown anomalies could improve both precision and recall. Additionally, exploring real-time anomaly detection frameworks could provide more timely insights, which are crucial for practical applications in financial markets. These future directions will help build more robust and comprehensive anomaly detection systems in financial trading environments.

## Declarations

### Author Contributions

Conceptualization: T.H., and A.S.M.A.; Methodology: A.S.M.A.; Software: T.H.; Validation: T.H., and A.S.M.A.; Formal Analysis: T.H., and A.S.M.A.; Investigation: T.H.; Resources: A.S.M.A.; Data Curation: A.S.M.A.; Writing Original Draft Preparation: T.H. and A.S.M.A.; Writing Review and Editing: T.H.; Visualization: T.H.; All authors have read and agreed to the published version of the manuscript.

### Data Availability Statement

The data presented in this study are available on request from the corresponding author.

### Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

### Institutional Review Board Statement

Not applicable.

### Informed Consent Statement

Not applicable.

### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] J. Wang, Y. Xue, and M. Liu, "An Analysis of Bitcoin Price Based on VEC Model," 2016, doi: 10.2991/icemi-16.2016.36.
- [2] A. J. Ram, "Bitcoin as a New Asset Class," *Meditari Account. Res.*, vol. 27, no. 1, pp. 147–168, 2019, doi: 10.1108/medar-11-2017-0241.
- [3] J. Papp, "A Medium of Exchange for an Internet Age: How to Regulate Bitcoin for the Growth of E-Commerce," *Pittsburgh J. Technol. Law Policy*, vol. 15, no. 1, pp. 33–56, 2015, doi: 10.5195/tlp.2014.155.

- [4] F. Parino, M. G. Beiró, and L. Gauvin, "Analysis of the Bitcoin Blockchain: Socio-Economic Factors Behind the Adoption," *Epj Data Sci.*, vol. 7, no. 1, 2018, doi: 10.1140/epjds/s13688-018-0170-8.
- [5] B. X. Hu, J. H. Hwang, C. Jain, and J. Washam, "Bitcoin Price Manipulation: Evidence From Intraday Orders and Trades," *Appl. Econ. Lett.*, vol. 29, no. 2, pp. 140–144, 2020, doi: 10.1080/13504851.2020.1861183.
- [6] A. S. Hu, C. A. Parlour, and U. Rajan, "Cryptocurrencies: Stylized Facts on a New Investible Instrument," *Financ. Manag.*, vol. 48, no. 4, pp. 1049–1068, 2019, doi: 10.1111/fima.12300.
- [7] P. Kayal and G. Balasubramanian, "Excess Volatility in Bitcoin: Extreme Value Volatility Estimation," *Iim Kozhikode Soc. Manag. Rev.*, vol. 10, no. 2, pp. 222–231, 2021, doi: 10.1177/2277975220987686.
- [8] J. Kamps and B. Kleinberg, "To the Moon: Defining and Detecting Cryptocurrency Pump-and-Dumps," *Crime Sci.*, vol. 7, no. 1, 2018, doi: 10.1186/s40163-018-0093-5.
- [9] E. Kaufman and A. Iaremenko, "Anomaly Detection for Fraud in Cryptocurrency Time Series," 2022, doi: 10.48550/arxiv.2207.11466.
- [10] N. G. Tosunoğlu, H. Abacı, G. Ateş, and N. S. Akkaya, "Artificial Neural Network Analysis of the Day of the Week Anomaly in Cryptocurrencies," *Financ. Innov.*, vol. 9, no. 1, 2023, doi: 10.1186/s40854-023-00499-x.
- [11] A. Petukhina, R. C. G. Reule, and W. K. Härdle, "Rise of the Machines? Intraday High-Frequency Trading Patterns of Cryptocurrencies," *Eur. J. Finance*, vol. 27, no. 1–2, pp. 8–30, 2020, doi: 10.1080/1351847x.2020.1789684.
- [12] J. D. Parmar and J. T. Patel, "Anomaly Detection in Data Mining: A Review," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 4, pp. 32–40, 2017, doi: 10.23956/ijarcsse/v7i4/0142.
- [13] Y. Peng, A. Tan, J. Wu, and Y. Bi, "Hierarchical Edge Computing: A Novel Multi-Source Multi-Dimensional Data Anomaly Detection Scheme for Industrial Internet of Things," *Ieee Access*, vol. 7, pp. 111257–111270, 2019, doi: 10.1109/access.2019.2930627.
- [14] P. Marjai, P. Lehotay-Kéry, and A. Kiss, "The Use of Template Miners and Encryption in Log Message Compression," *Computers*, vol. 10, no. 7, p. 83, 2021, doi: 10.3390/computers10070083.
- [15] S. Sagar and S. M.S., "Extracting Events Influencing Anomalous Behavior in Stock Market - A Data Driven Approach Using Sentiment Analysis," 2022, doi: 10.21203/rs.3.rs-2374845/v1.
- [16] S. Tiwari, H. Ramampiaro, and H. Langseth, "Machine Learning in Financial Market Surveillance: A Survey," *Ieee Access*, vol. 9, pp. 159734–159754, 2021, doi: 10.1109/access.2021.3130843.
- [17] C. Goh, B. H. Z. Lee, G. Pan, and P. S. Seow, "Forensic Analytics Using Cluster Analysis: Detecting Anomalies in Data," *J. Corp. Account. Finance*, vol. 32, no. 2, pp. 154–161, 2021, doi: 10.1002/jcaf.22486.
- [18] M.-S. Cheong, M.-H. Wu, and S.-H. Huang, "Interpretable Stock Anomaly Detection Based on Spatio-Temporal Relation Networks With Genetic Algorithm," *Ieee Access*, vol. 9, pp. 68302–68319, 2021, doi: 10.1109/access.2021.3077067.
- [19] D. T. Lan and S. Yoon, "Trajectory Clustering-Based Anomaly Detection in Indoor

Human Movement,” *Sensors*, vol. 23, no. 6, p. 3318, 2023, doi: 10.3390/s23063318.

- [20] M. Ahmed and A. N. Mahmood, “Novel Approach for Network Traffic Pattern Analysis Using Clustering-Based Collective Anomaly Detection,” *Ann. Data Sci.*, vol. 2, no. 1, pp. 111–130, 2015, doi: 10.1007/s40745-015-0035-y.