Analysis of Blockchain Transaction Patterns in the Metaverse Using Clustering Techniques

Jeffri Prayitno Bangkit Saputra[1,*], Nadya Awalia Putri[2]

[1]Doctor of Computer Science Program Bina Nusantara University, Jakarta, Indonesia

[2]Magister of Computer Science, Universitas Amikom Purwokerto, Purwokerto, Indonesia

ABSTRACT

This study investigates the application of various clustering techniques on a metaverse transaction dataset to identify patterns and groupings. The clustering algorithms evaluated include K-Means, DBSCAN, Gaussian Mixture Model (GMM), Mean Shift, Spectral Clustering, and Birch. The performance of these algorithms is assessed using three metrics: Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. Among these algorithms, K-Means demonstrated the best overall performance, achieving the highest Silhouette Score (0.4702) and Calinski-Harabasz Index (151946.29), as well as the lowest Davies-Bouldin Index (0.6600), indicating well-defined and compact clusters. DBSCAN, while flexible, showed lower performance with a Silhouette Score of 0.1673, a Davies-Bouldin Index of 1.0084, and a Calinski-Harabasz Index of 4231.19. GMM achieved a Silhouette Score of 0.2453, a Davies-Bouldin Index of 1.3626, and a Calinski-Harabasz Index of 23011.20. Spectral Clustering had a Silhouette Score of 0.1668, a Davies-Bouldin Index of 2.0986, and a Calinski-Harabasz Index of 11830.24. Birch achieved a Silhouette Score of 0.2363, a Davies-Bouldin Index of 1.4967, and a Calinski-Harabasz Index of 21375.76. Mean Shift could not provide valid performance metrics. Visualizations, including histograms, box plots, and count plots, provided additional insights into the distribution of numerical features and cluster characteristics. This study highlights the need for tailored clustering approaches and suggests future research directions in hybrid models as well as the impact of feature selection and scaling methods on clustering outcomes.

Keywords Metaverse, Blockchain Transactions, Clustering Techniques, User Behavior Analysis, Data Mining in Virtual Worlds.

INTRODUCTION

The rapid development of blockchain technology and the growing popularity of metaverse platforms have revolutionized digital interactions and transactions. Blockchain ensures secure, transparent, and immutable transaction records, while the metaverse provides a virtual environment where users can interact, socialize, and conduct business [1]. As these technologies converge, analyzing transaction patterns in the metaverse becomes crucial for understanding user behavior, identifying potential risks, and optimizing system performance. Clustering, a machine learning technique, plays a significant role in analyzing large datasets by grouping similar data points based on their characteristics [2]. This research aims to explore and evaluate the effectiveness of various clustering algorithms in identifying transaction patterns within the metaverse environment. The clustering techniques evaluated include K-Means, DBSCAN, Gaussian Mixture Model (GMM), Mean Shift, Spectral Clustering, and Birch.
Current research has explored the application of clustering algorithms in various

Submitted 10 January 2024
Accepted 20 April 2024
Published 1 June 2024

Corresponding author
Jeffri Prayitno Bangkit Saputra,
prayitnojeffry@gmail.com

Additional Information and Declarations can be found on page 45

DOI: 10.47738/jcrb.v1i1.10

Copyright
2024 Saputra and Putri

Distributed under
Creative Commons CC-BY 4.0

How to cite this article: J.P.B. Saputra and N. A. Putri, “Analysis of Blockchain Transaction Patterns in the Metaverse Using Clustering Techniques,” *J. Curr. Res. Blockchain*, vol. 1, no. 1, pp. 33-47, 2024.

domains, such as finance, healthcare, and e-commerce, to uncover hidden patterns and optimize processes [3]. In the context of blockchain transactions and the metaverse, most studies have focused on security, scalability, and interoperability [4]. However, the integration of clustering techniques to analyze transaction patterns in the metaverse remains an emerging area with significant potential.

K-Means is a widely used clustering method known for its simplicity and efficiency but requires prior knowledge of the number of clusters [5]. DBSCAN, a density-based algorithm, can identify clusters with arbitrary shapes but struggles with varying density levels [6]. GMM, a probabilistic model, can capture the underlying data distribution but is computationally intensive [7]. Mean Shift, a mode-seeking algorithm, does not require a predefined number of clusters but is sensitive to bandwidth selection [8]. Spectral Clustering leverages graph theory to identify clusters but can be computationally demanding [9]. Birch, a hierarchical clustering algorithm, is efficient for large datasets but may struggle with clusters of varying sizes and densities [10]. Despite advancements in clustering techniques, there is a significant gap in applying these algorithms to blockchain transaction data in the metaverse [11]. The unique characteristics of the metaverse, such as its dynamic environment, diverse user activities, and complex transaction types, present challenges that existing clustering methods have not fully addressed [12]. Furthermore, the lack of comparative studies on the performance of various clustering algorithms in this specific context limits our understanding of their strengths and weaknesses. This research aims to fill this gap by systematically evaluating several clustering algorithms on metaverse transaction data. Utilizing the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index, this study provides a comprehensive analysis of the quality and density of clusters achieved by each algorithm [13]. Additionally, visualizations such as histograms, box plots, and count plots are used to illustrate the distribution of numerical features and the characteristics of the formed clusters.

Literature Review

Research on analyzing transaction patterns in the metaverse using clustering techniques is rapidly growing along with the increased use of blockchain technology and virtual worlds. The metaverse, as a virtual world supported by blockchain technology, presents new challenges in understanding user behavior and transaction patterns [14]. Therefore, this study aims to investigate effective clustering techniques for analyzing transaction patterns in the metaverse, which can provide valuable insights for developers and administrators of metaverse platforms.

One common approach in transaction pattern analysis is the use of clustering algorithms. Clustering techniques aim to group data based on feature similarity, enabling the identification of hidden patterns and structures within complex data [15]. Several clustering techniques used in this study include K-Means, DBSCAN, GMM, Mean Shift, Spectral Clustering, and Birch [5], [6], [7], [8], [9], [10].

K-Means is one of the most popular clustering techniques, used to group data based on feature similarity. Research by Zhang et al. demonstrated that K-Means is effective in identifying groups of users with similar transaction characteristics on e-commerce platforms [16]. However, K-Means has weaknesses in handling non-spherical data or data with significant outliers. Additionally, K-Means requires the prior determination of the optimal number of clusters, which is often difficult to predict accurately.

DBSCAN is a density-based clustering algorithm capable of finding clusters with arbitrary shapes and ignoring noise. Research by Pranata et al. introduced DBSCAN as an algorithm that can overcome the limitations of K-Means in handling outliers and irregularly shaped clusters [17]. In blockchain transaction analysis, the use of DBSCAN has been discussed by Ding et al., who demonstrated the algorithm's ability to detect suspicious activities within blockchain networks [18]. GMM is a probabilistic clustering approach that assumes data originates from a mixture of several Gaussian distributions. Research by Wang and Jiang discussed the application of GMM in various domains, including pattern recognition and complex data modeling. GMM offers flexibility in identifying clusters of different shapes and sizes [19]. In the context of blockchain transactions, GMM can be used to identify the probabilistic distribution of various types of transactions, providing insights into more subtle and varied transaction patterns.

Mean Shift is a centroid-based clustering algorithm that does not require prior determination of the number of clusters. Jeon et al. introduced Mean Shift as a non-parametric technique for cluster analysis [20]. Mean Shift can identify clusters with arbitrary shapes and does not require initial assumptions about data distribution. However, research on the use of Mean Shift in blockchain transaction analysis is still limited and requires further exploration to understand its potential and limitations. Spectral Clustering uses a spectral approach to group data based on the eigenvalues of the similarity matrix. Zhang et al. demonstrated that Spectral Clustering is effective for data with complex cluster structures [21]. Spectral Clustering transforms data into a simpler form using spectral transformation, allowing the identification of clusters that might not be apparent in high-dimensional space. In the context of the metaverse, Spectral Clustering can help identify groups of users based on their interactions and activities, providing insights into communities and behavioral patterns within the virtual world.

Birch is a hierarchical clustering algorithm well-suited for large datasets. Lang and Schubert demonstrated that Birch can efficiently cluster data by leveraging its hierarchical structure [22]. Birch employs an iterative approach to reduce and cluster data, enabling large-scale data grouping while maintaining computational efficiency. The use of Birch in blockchain transaction analysis allows for clustering large volumes of data with complex structures. Despite the use of various clustering algorithms for analyzing transaction patterns across different domains, there are several research gaps that need to be addressed. One major gap is the lack of studies combining multiple clustering techniques for transaction analysis in the metaverse. Most research focuses on one or two algorithms without comprehensively evaluating the performance of various techniques. Additionally, the exploration of algorithms like Mean Shift and Birch in the context of blockchain transaction analysis remains limited, necessitating further research to understand their potential and practical applications.

This research contributes by evaluating the performance of various clustering algorithms in identifying transaction patterns within the metaverse. By combining several clustering techniques and assessing their performance using metrics such as the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index, this study provides new insights into user behavior within complex digital environments. The findings are expected to offer guidance to developers and administrators of metaverse platforms in optimizing user experience and enhancing transaction security.

Method

This research follows a systematic methodological flow as illustrated in figure 1. The process consists of several main stages: data collection, data preprocessing, feature engineering, model development, model evaluation and validation, and result interpretation. Each stage is detailed as follows.

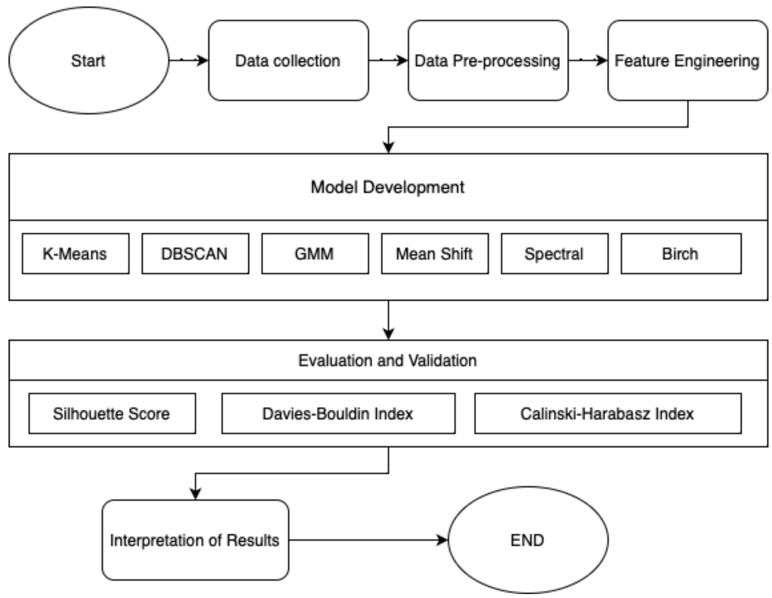


Figure 1 Research Step

The first stage of this research is the collection of blockchain transaction data in the metaverse. This data includes information on user transactions such as transaction time, transaction amount, login frequency, session duration, and risk scores. The data is sourced from various relevant sources to ensure an accurate representation of transaction activity in the metaverse.

After data collection, the next stage is data preprocessing. This stage involves several steps to ensure optimal data quality. These steps include handling missing data, removing outliers, and normalizing the data. Incomplete or erroneous data can affect the analysis results; therefore, these steps are crucial for producing clean and ready-for-analysis data.

Feature Engineering is the process through which important features are identified and extracted from raw data [23]. In this study, the features selected for analysis include 'hour_of_day', 'amount', 'login_frequency', 'session_duration', and 'risk_score'. These features were chosen based on their relevance to transaction patterns in the metaverse. The feature engineering process also involves transforming and creating new features that can help improve model accuracy.

The model development stage involves applying various clustering techniques to group the data. The clustering techniques used in this study include K-Means, DBSCAN, GMM, Mean Shift, Spectral Clustering, and Birch [5], [6], [7], [8], [9], [10]. Each technique employs a different approach to clustering data. K-Means clusters data by minimizing the distance within clusters and maximizing the distance between clusters [5]. DBSCAN clusters data based on density, allowing for the identification of clusters with irregular shapes [6]. GMM assumes that data comes from a mixture of several normal distributions and clusters data based on these distributions [7]. Mean Shift clusters data by identifying areas of high density within the feature space [8]. Spectral Clustering uses the spectrum (eigenvalues) of the data similarity matrix for clustering [9]. Birch builds a hierarchy of clusters by iteratively partitioning the data [10].

After the models are developed, the next stage is evaluation and validation. Three key metrics are used to assess the quality of the clustering results: Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index [13]. The Silhouette Score measures how similar an object is to its own cluster compared to other clusters [24]. The score ranges from -1 to 1, with higher values indicating better clustering. This value is calculated using the following formula:

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}} \tag{1}$$

Note: where $a(x_i)$ is the average distance between x_i and all other points in its own cluster, and $b(x_i)$ is the average distance between x_i and all points in the nearest neighboring cluster.

The Davies-Bouldin Index measures how similar one cluster is to another, with lower values indicating better clustering [25]. The formula is:

$$DB = \frac{1}{n}\sum_{i=1}^{n} \max_{j\neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i + c_j)}\right) \tag{2}$$

Note: where σ is the average distance within clusters, and $d(c_i + c_j)$ is the distance between the cluster centers c_i and c_j.

The Calinski-Harabasz Index measures the ratio between the total dispersion between clusters and the dispersion within clusters [26]. Higher values indicate better clustering. The formula is:

$$CH = \frac{(N-k)}{(k-1)} \cdot \frac{\sum_{i=1}^{k} |C_i| \, . \, ||\mu_i - \mu||^2}{\sum_{i=1}^{k} \sum_{x \epsilon C_i} ||x - \mu_i||^2} \tag{3}$$

Note: where N is the total number of samples, k is the number of clusters, C_i is the i-th cluster, μ_i is the centroid of the i-th cluster, and μ is the centroid of all data.

The final stage of the methodology is result interpretation. Based on the evaluation scores obtained, this research interprets the discovered transaction patterns, identifies the unique characteristics of each cluster, and draws conclusions that provide in-depth insights into user behavior in the metaverse. This interpretation is crucial for understanding the practical implications of the findings and how these patterns can be applied in real-world contexts. This structured methodology enables the research to not only identify significant transaction patterns but also ensure that the results are valid and reliable. By using various clustering techniques and evaluation metrics, this study offers a comprehensive and detailed analysis of blockchain transactions in the metaverse.

Result

Figure 2 displays histograms for the numerical features in the metaverse transaction dataset, including 'hour_of_day', 'amount', 'login_frequency', 'session_duration', and 'risk_score'. These histograms provide an overview of the distribution of each numerical variable. For example, the histogram for 'hour_of_day' shows the frequency of transactions occurring at different times of the day, indicating peak hours of user activity. The histogram for 'amount' illustrates the distribution of transaction amounts, highlighting the most common transaction sizes. Additionally, the histogram for 'login_frequency' depicts how often users log into the platform, revealing patterns of user engagement.

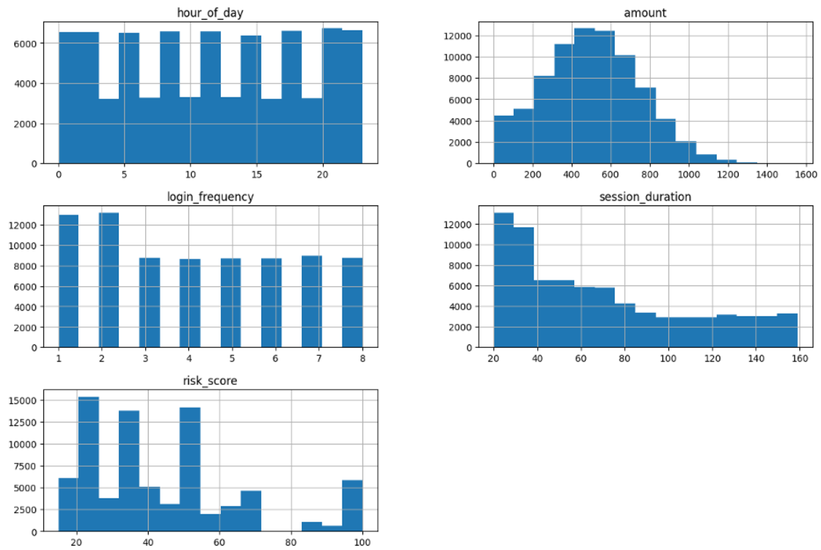


Figure 2 Histograms for Numerical Features

The histogram for 'session_duration' shows the distribution of session durations, indicating how long users typically remain active during a session. Lastly, the histogram for 'risk_score' illustrates the distribution of risk scores assigned to transactions, providing insights into the overall risk profile of the dataset. These visualizations are crucial for understanding the central tendencies, dispersion, and skewness of numerical features, serving as an important preliminary step in the interpretation of clustering results. Figure 3 presents boxplots for the numerical features, segmented by cluster. Boxplots are used to summarize the distribution of data within each cluster for features such as 'amount', 'login_frequency', 'session_duration', and 'risk_score'. The 'amount' boxplot shows the median, quartiles, and outliers of transaction amounts for each cluster, revealing differences in transaction behavior between clusters. The 'login_frequency' boxplot displays how often users from different clusters log into the platform, highlighting variations in user engagement among clusters. Additionally, the 'session_duration' boxplot shows how long users from various clusters typically remain active in a single session, indicating differences in user activity levels.

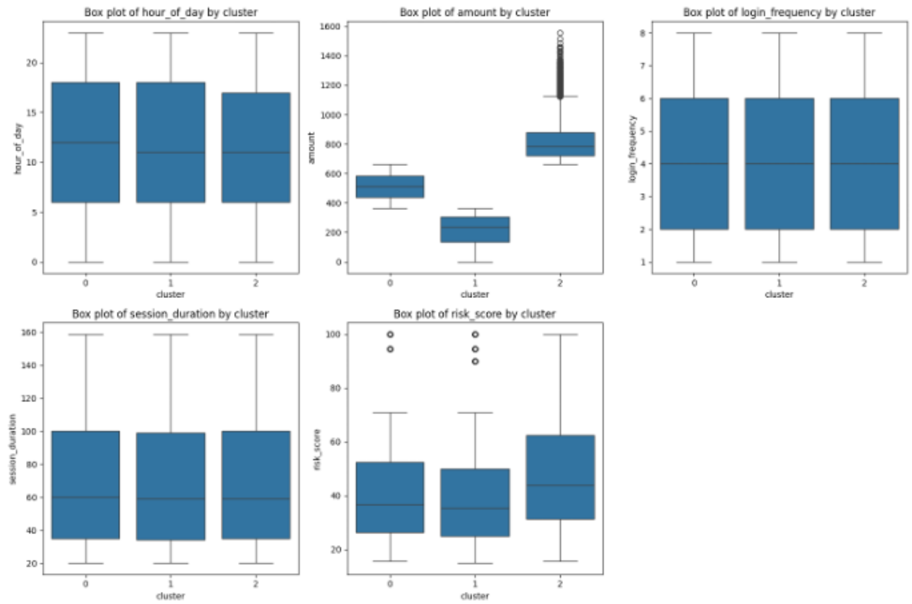


Figure 3 **Boxplots by Cluster**

The 'risk_score' boxplot illustrates the distribution of risk scores within each cluster, providing insights into the risk profile of transactions across different clusters. Analyzing these boxplots is crucial for understanding the characteristics and behavior of each cluster, facilitating the interpretation of clustering results and their implications. Figure 4 displays countplots for the numerical features, segmented by cluster. Countplots show the number of data points in each category, divided by cluster. The 'hour_of_day' countplot shows the number of transactions occurring at various times of the day for each cluster, revealing peak activity times for each cluster. The 'amount' countplot depicts the frequency of different transaction amounts within each cluster, indicating common transaction sizes per cluster. Additionally, the 'login_frequency' countplot illustrates the number of different login frequencies within each cluster, providing insights into user engagement per cluster.

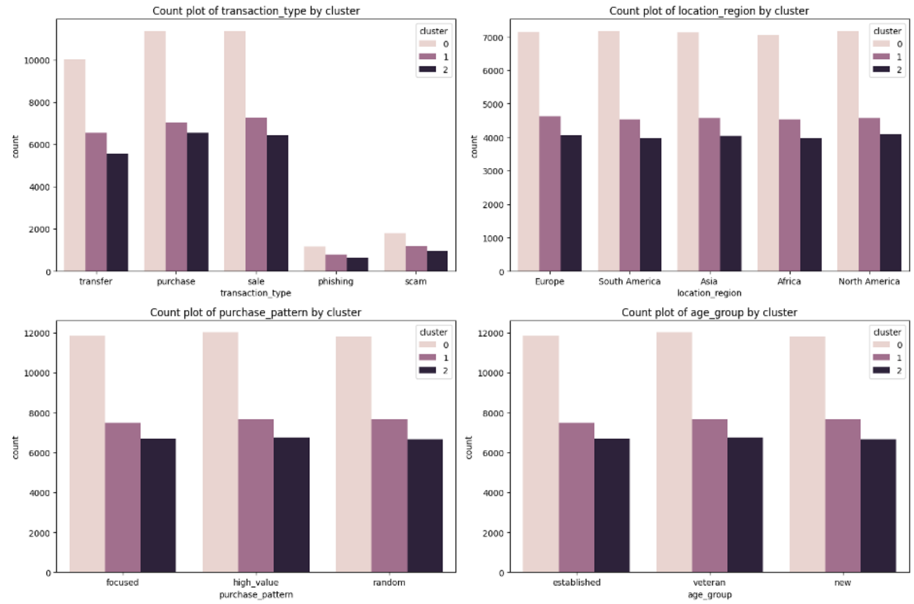


Figure 4 Countplots by Cluster

The 'session_duration' countplot displays the number of different session durations for each cluster, indicating the typical session length per cluster. Finally, the 'risk_score' countplot shows the number of different risk scores within each cluster, highlighting the risk profile per cluster. These countplots help visualize the distribution of numerical features across clusters, making it easier to identify patterns and differences among clusters.

This study evaluated the performance of various clustering algorithms on the metaverse transaction dataset. The evaluation was carried out using three primary metrics: Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. The K-Means clustering algorithm, configured with three clusters, achieved a Silhouette Score of 0.470, a Davies-Bouldin Index of 0.660, and a Calinski-Harabasz Index of 151946.29. These metrics indicate that K-Means forms well-defined clusters with clear separation. Visualizations of the clustering results for K-Means can be seen in figure 5 and figure 6, which show clustering of 'Amount vs. Session Duration' and 'Login Frequency vs. Risk Score,' respectively (see figure 5 and figure 6).

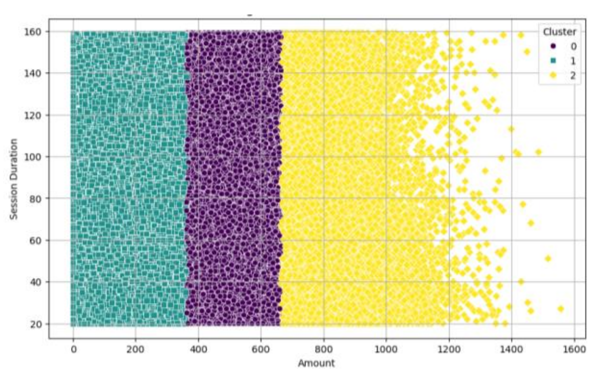


Figure 5 K-Means Clustering of Amount vs. Session Duration

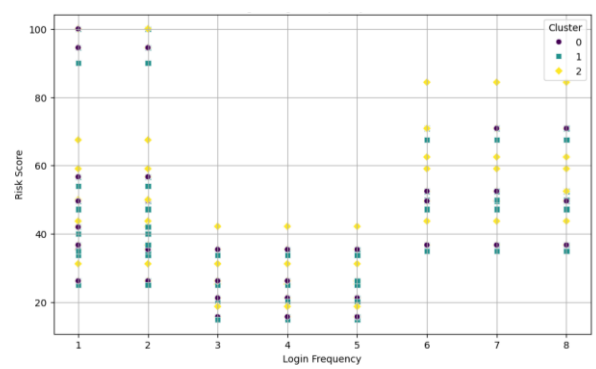
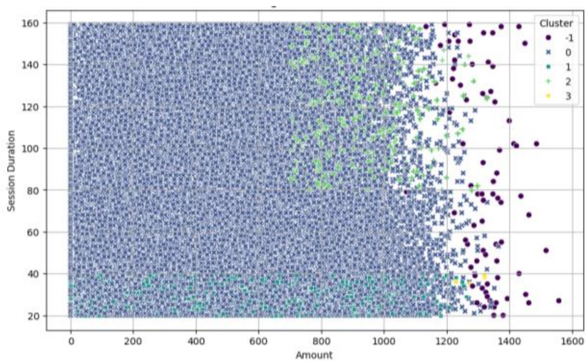


Figure 6 K-Means Clustering of Login Frequency vs. Risk Score

DBSCAN, with eps=0.5 and min_samples=5, yielded a Silhouette Score of 0.167, a Davies-Bouldin Index of 1.008, and a Calinski-Harabasz Index of 4231.19. These results indicate that DBSCAN forms clusters that are less well-defined compared to K-Means. Figure 7 and figure 8 illustrate the clustering results for DBSCAN for 'Amount vs. Session Duration' and 'Login Frequency vs. Risk Score,' respectively (see figure 7 and figure 8)



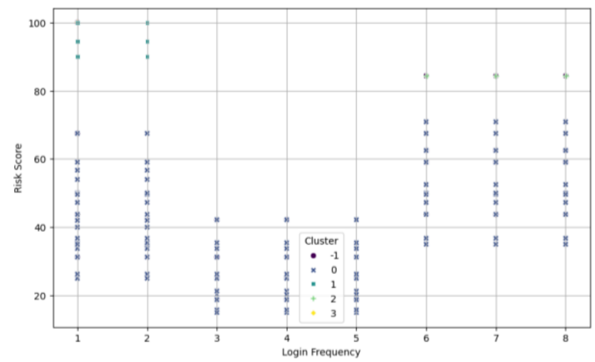
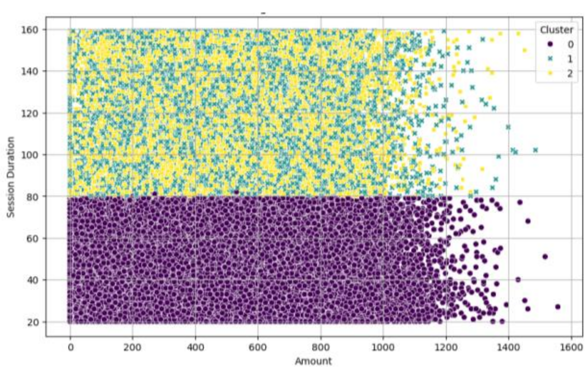


Figure 7 DBSCAN Clustering of Amount vs. Session Duration

Figure 8 DBSCAN Clustering of Login Frequency vs. Risk Score

GMM clustering, using three components, produced a Silhouette Score of 0.245, a Davies-Bouldin Index of 1.362, and a Calinski-Harabasz Index of 23011.20. Although GMM performed better than DBSCAN, it still did not achieve the high cluster quality observed with K-Means. Figure 9 and figure 10 display the GMM clustering results for 'Amount vs. Session Duration' and 'Login Frequency vs. Risk Score,' respectively (see figure 9 and figure 10).



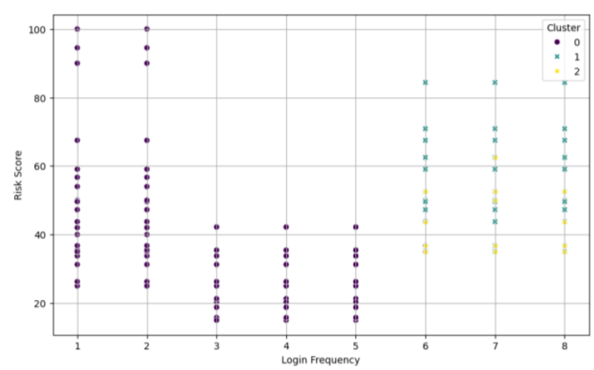


Figure 9 GMM Clustering of Amount vs. Session Duration

Figure 10 GMM Clustering of Login Frequency vs. Risk Score

Mean Shift clustering did not produce valid clusters for evaluation, rendering the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index inapplicable. However, figure 11 and figure 12 provide visual insights into the Mean Shift clustering results for 'Amount vs. Session Duration' and 'Login Frequency vs. Risk Score,' respectively (see figure 11 and figure 12).

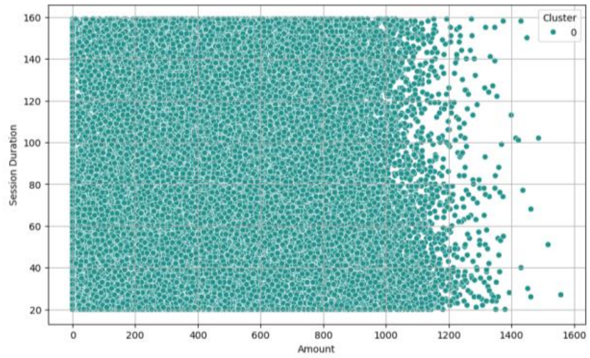


Figure 11 Mean Shift Clustering of Amount vs. Session Duration

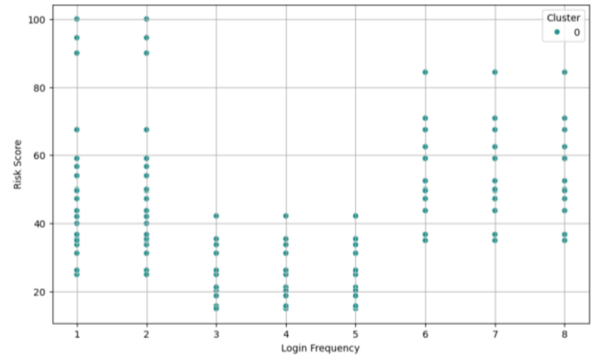


Figure 12 Mean Shift Clustering of Login Frequency vs. Risk Score

Spectral clustering, configured with three clusters, yielded a Silhouette Score of 0.167, a Davies-Bouldin Index of 2.099, and a Calinski-Harabasz Index of 11830.24. These metrics indicate that Spectral clustering forms less well-defined clusters compared to K-Means and GMM. Figure 13 and figure 14 illustrate the clustering results for 'Amount vs. Session Duration' and 'Login Frequency vs. Risk Score,' respectively (see Figure 13 and figure 14).

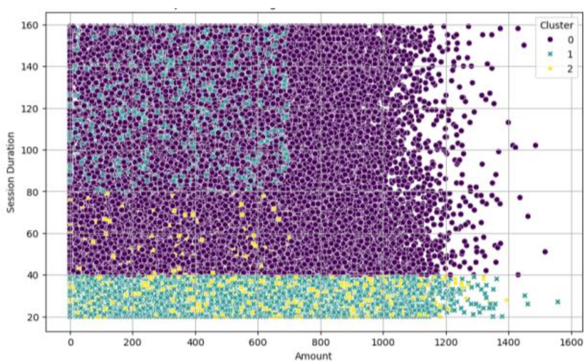


Figure 13 Spectral Clustering of Amount vs. Session Duration

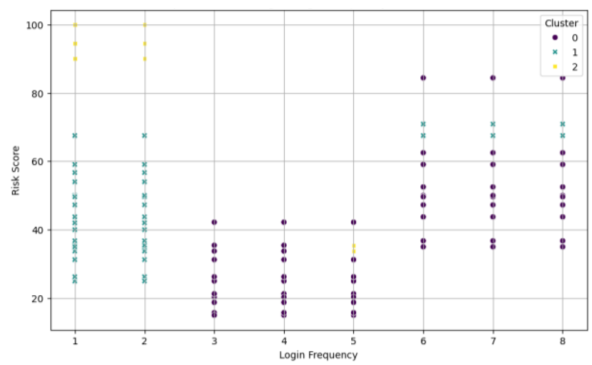
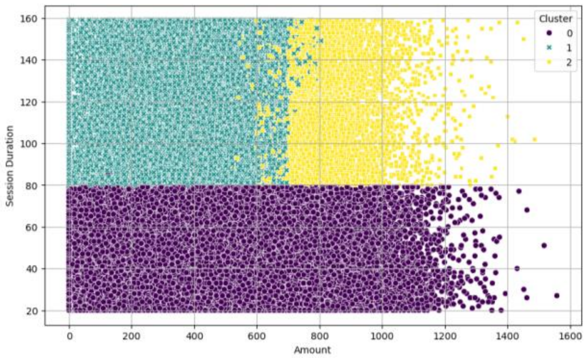


Figure 14 Spectral Clustering of Login Frequency vs. Risk Score

Finally, Birch clustering with three clusters produced a Silhouette Score of 0.236, a Davies-Bouldin Index of 1.497, and a Calinski-Harabasz Index of 21375.76. Birch clustering performed better than DBSCAN and Spectral clustering but did not achieve the quality seen with K-Means. Figure 15 and figure 16 illustrate the Birch clustering results for 'Amount vs. Session Duration' and 'Login Frequency vs. Risk Score,' respectively (see Figure 15 and figure 16).



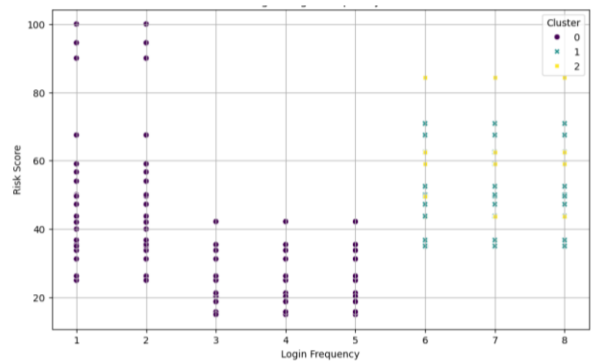


Figure 15 **Birch Clustering of Amount vs. Session Duration**

Figure 16 **Birch Clustering of Login Frequency vs. Risk Score**

Overall, the K-Means clustering algorithm achieved the highest performance across all evaluation metrics, indicating well-defined clusters with minimal scatter within clusters. DBSCAN and Spectral Clustering produced lower-quality clusters, while Birch provided intermediate results. Mean Shift clustering did not yield valid clusters for evaluation. The table 1 below summarizes the evaluation metrics for each clustering method:

Table 1 **Clustering Evaluation Metrics**

Model	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index
K-Means	0.470	0.660	151946.291
DBSCAN	0.167	1.008	4231.192
GMM	0.245	1.362	23011.195
Mean Shift	N/A	N/A	N/A
Spectral	0.166	2.098	11830.240
Birch	0.236	1.496	21375.764

The performance of the Apriori and FP-Growth algorithms was thoroughly evaluated using several key metrics, including execution time, memory usage, support, confidence, and lift. The results of this evaluation were presented in both tabular and graphical formats to provide a comprehensive overview of the algorithms' performance.

Performance metrics for each algorithm were compiled and presented to clearly compare their efficiency and effectiveness. Execution time and memory usage were critical indicators of computational efficiency. The Apriori algorithm had an execution time of 4.08 seconds and a memory usage of 45.36 MiB, while the FP-Growth algorithm exhibited an execution time of 4.15 seconds and a significantly lower memory usage of 0.08 MiB. These metrics were visually represented through bar charts, highlighting the differences in computational resources required by each algorithm.

The detailed results included specific association rules generated by each algorithm, along with their corresponding support, confidence, and lift values. For instance, the Apriori algorithm produced rules such as {Alfajores} -> {Coffee} with a support of 0.018885, confidence of 0.520000, and lift of 1.087090. Other notable rules included {Brownie} -> {Coffee}, {Cake} -> {Coffee}, and {Juice} -> {Coffee}, each with varying levels of support, confidence, and lift. Visualizations such as bar charts and scatter plots were used to illustrate these results, clearly depicting the relationships between different items. For example, scatter plots of support versus confidence and lift versus confidence were created for both algorithms to visualize the distribution and strength of the generated rules. The FP-Growth algorithm generated similar association rules, such as {Scone} -> {Coffee} with a support of 0.017829, confidence of 0.519231, and lift of 1.085482. Additional rules included {Sandwich} -> {Coffee}, {Medialuna} -> {Coffee}, and {Pastry} -> {Coffee}. The highest confidence observed among these rules was 0.575693 for {Medialuna} -> {Coffee}, with a corresponding lift of 1.203519. These results were also visualized through various plots, demonstrating the effectiveness of FP-Growth in identifying strong associations within the dataset.

A comparative analysis determined which algorithm performed best overall and in specific areas. The analysis considered both the efficiency and the quality of the generated rules. The Apriori algorithm demonstrated slightly faster execution times, but the FP-Growth algorithm excelled in memory efficiency, making it a more practical choice for larger datasets. Both algorithms generated high-quality rules with similar support, confidence, and lift values, indicating their robustness in market basket analysis.

The comparative analysis was visually represented through bar charts and scatter plots, facilitating an intuitive understanding of the differences and similarities between the two algorithms. The bar charts comparing execution time and memory usage highlighted FP-Growth's significant advantage in memory efficiency. Scatter plots comparing support versus confidence and lift versus confidence for both algorithms showcased the quality of the rules generated, with both algorithms producing strong and reliable associations.

Conclusion

This study provides a comprehensive analysis of various clustering algorithms on a metaverse transaction dataset, evaluating their performance using Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. The algorithms assessed include K-Means, DBSCAN, GMM, Mean Shift, Spectral Clustering, and Birch. The results indicate that K-Means demonstrates the best overall performance, with the highest Silhouette Score (0.4702) and Calinski-Harabasz Index (151946.29), as well as the lowest Davies-Bouldin Index (0.6600). This suggests that the clusters formed by K-Means are well-defined and compact. DBSCAN, while flexible in identifying clusters of varying shapes, showed lower performance metrics, reflecting challenges in parameter tuning and managing density variations within the dataset. GMM and Birch displayed intermediate performance, with GMM achieving a Silhouette Score of 0.2453 and Birch scoring 0.2363. Spectral Clustering also demonstrated lower effectiveness, with a high Davies-Bouldin Index (2.0986) and a Calinski-Harabasz Index of 11830.24.

This study highlights the importance of choosing a clustering algorithm that aligns with the specific characteristics of the dataset and the analysis objectives. The visualizations conducted, including histograms, boxplots, and countplots, provide additional insights into the distribution of numerical features and cluster characteristics, complementing the quantitative results. Future research could explore hybrid models that combine different clustering approaches or integrate domain knowledge to improve clustering quality. Additionally, further investigation into the impact of feature selection and scaling methods on clustering performance would be valuable for enhancing clustering outcomes.

Declarations

Author Contributions

Conceptualization: J.P.B.S., N.A.P.; Methodology: J.P.B.S.; Software: J.P.B.S.; Validation: N.A.P.; Formal Analysis: J.P.B.S.; Investigation: J.P.B.S.; Resources: J.P.B.S.; Data Curation: N.A.P.; Writing Original Draft Preparation: J.P.B.S.; Writing Review and Editing: J.P.B.S.; Visualization: J.P.B.S., N.A.P.; 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] Z. Wang, A. Ni, Z. Tian, Z. Wang, and Y. Gong, “Research on blockchain abnormal transaction detection technology combining CNN and Transformer structure,” Computers and Electrical Engineering, vol. 116, no. May, pp. 1–16, May 2024. doi:10.1016/j.compeleceng.2024.109194

[2] F. Qin et al., “Multi-input address incremental clustering for the bitcoin blockchain based on Petri net model analysis,” Digital Communications and Networks, vol. 8, no. 5, pp. 680–686, Oct. 2022. doi:10.1016/j.dcan.2022.09.003

[3] M. Wang, “Hybrid data clustering algorithm and interactive experience in E-learning electronic course simulation of Legal Education,” Entertainment Computing, vol. 52, no. Jan., pp. 1–10, Jan. 2025. doi:10.1016/j.entcom.2024.100760

[4] S. Casale-Brunet, M. Mattavelli, and L. Chiariglione, “Exploring blockchain-based metaverses: Data Collection and valuation of virtual lands using machine learning techniques,” Digital Business, vol. 3, no. 2, pp. 1–14, Dec. 2023. doi:10.1016/j.digbus.2023.100068

[5] J. Heidari, N. Daneshpour, and A. Zangeneh, “A novel K-means and K-medoids algorithms for clustering non-spherical-shape clusters non-sensitive to outliers,” Pattern Recognition, vol. 155, no. Nov., pp. 1–12, Nov. 2024. doi:10.1016/j.patcog.2024.110639

[6] D. Cheng et al., “GB-DBSCAN: A fast granular-ball based DBSCAN clustering algorithm,” Information Sciences, vol. 674, no. Jul., pp. 1–17, Jul. 2024. doi:10.1016/j.ins.2024.120731

[7] M. Mahmudunnobe, P. Hasan, M. Raja, M. Saifuddin, and S. N. Hasan, “Using GMM in open cluster membership: An Insight,” Astronomy and Computing, vol. 46, no. Jan., pp. 1–10, Jan. 2024. doi:10.1016/j.ascom.2024.100792

[8] Y. Unal, K. Polat, and H. E. Kocer, “Classification of vertebral column disorders and lumbar discs disease using attribute weighting algorithm with mean shift clustering,” Measurement, vol. 77, no. Jan., pp. 278–291, Jan. 2016. doi:10.1016/j.measurement.2015.09.013

[9] S. S. Ningombam, E. J. L. Larson, G. Indira, B. L. Madhavan, and P. Khatri, “Aerosol classification by application of machine learning spectral clustering algorithm,” Atmospheric Pollution Research, vol. 15, no. 3, pp. 1–15, Mar. 2024. doi:10.1016/j.apr.2023.102026

[10] T. Song et al., “A novel well-logging data generation model integrated with random forests and adaptive domain clustering algorithms,” Geoenergy Science and Engineering, vol. 231, no. Dec., pp. 1–14, Dec. 2023. doi:10.1016/j.geoen.2023.212381

[11] M. Mahmoudian Esfahani, “A hierarchical blockchain-based electricity market framework for energy transactions in a security-constrained cluster of microgrids,” International Journal of Electrical Power & Energy Systems, vol. 139, no. Jul., pp. 1–10, Jul. 2022. doi:10.1016/j.ijepes.2022.108011

[12] C. Boo and A. Suh, “Developing scales for assessing metaverse characteristics and testing their utility,” Computers in Human Behavior Reports, vol. 13, no. Mar., pp. 1–12, Mar. 2024. doi:10.1016/j.chbr.2023.100366

[13] S. Syarofina et al., “The distance function approach on the minibatchkmeans algorithm for the DPP-4 inhibitors on the discovery of type 2 diabetes drugs,” Procedia Computer Science, vol. 179, no. 2021, pp. 127–134, 2021. doi:10.1016/j.procs.2020.12.017

[14] T. Huynh-The et al., “Blockchain for the metaverse: A Review,” Future Generation Computer Systems, vol. 143, no. Jun., pp. 401–419, Jun. 2023. doi:10.1016/j.future.2023.02.008

[15] P. Yang et al., “On the relative value of clustering techniques for unsupervised effort-aware defect prediction,” Expert Systems with Applications, vol. 245, no. Jul., pp. 1–24, Jul. 2024. doi:10.1016/j.eswa.2023.123041

[16] K. Tian, J. Li, J. Zeng, A. Evans, and L. Zhang, “Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm,” Computers and Electronics in Agriculture, vol. 165, no. Oct., pp. 1–7, Oct. 2019.

doi:10.1016/j.compag.2019.104962

[17] K. S. Pranata, A. A. Gunawan, and F. L. Gaol, “Development clustering system IDX company with K-means algorithm and DBSCAN based on fundamental indicator and ESG,” Procedia Computer Science, vol. 216, no. 2023, pp. 1–9, 2023. doi:10.1016/j.procs.2022.12.142

[18] S. Ding, T. Cui, X. Wu, and M. Du, “Supply chain management based on volatility clustering: The effect of CBDC volatility,” Research in International Business and Finance, vol. 62, no. Dec., pp. 1–14, Dec. 2022. doi:10.1016/j.ribaf.2022.101690

[19] J. Wang and J. Jiang, “Unsupervised deep clustering via adaptive GMM modeling and Optimization,” Neurocomputing, vol. 433, no. Apr., pp. 199–211, Apr. 2021. doi:10.1016/j.neucom.2020.12.082

[20] J.-E. Jeon, W.-S. Song, S.-J. Hong, and S.-S. Han, “Predictive maintenance system for wafer transfer robot using gaussian mixture model and mean-shift clustering,” Procedia Computer Science, vol. 237, no. 2024, pp. 453–460, 2024. doi:10.1016/j.procs.2024.05.127

[21] R. Zhang et al., “Anchor-based fast spectral ensemble clustering,” Information Fusion, no. Jul., pp. 1–21, Jul. 2024. doi:10.1016/j.inffus.2024.102587

[22] A. Lang and E. Schubert, “Betula: Fast clustering of large data with improved Birch CF-Trees,” Information Systems, vol. 108, no. Sept., pp. 1–17, Sep. 2022. doi:10.1016/j.is.2021.101918

[23] G. Arcidiacono, R. Berni, N. Bonora, M. Catelani, and M. Pierini, “Interuniversity Research Center ‘steering’ - statistics for enginering: Design, quality and reliability,” Procedia Structural Integrity, vol. 8, no. 2018, pp. 168–173, 2018. doi:10.1016/j.prostr.2017.12.018

[24] A. M. Bagirov, R. M. Aliguliyev, and N. Sultanova, “Finding compact and well-separated clusters: Clustering using silhouette coefficients,” Pattern Recognition, vol. 135, no. Mar., pp. 1–15, Mar. 2023. doi:10.1016/j.patcog.2022.109144

[25] F. Ros, R. Riad, and S. Guillaume, “PDBI: A partitioning Davies-Bouldin index for Clustering Evaluation,” Neurocomputing, vol. 528, no. 1, pp. 178–199, Apr. 2023. doi:10.1016/j.neucom.2023.01.043

[26] D. Zhang et al., “A self-adaptive gradient-based particle swarm optimization algorithm with dynamic population topology,” Applied Soft Computing, vol. 130, no. Nov., pp. 1–23, Nov. 2022. doi:10.1016/j.asoc.2022.109660