Graduate Journal of Interdisciplinary Research, Reports & Reviews
(Grad.J.InteR^3)
ark:/85846/JPR3.2024AR.0203180
https://doi.org/10.5281/zenodo.14834228

ISSN(E): 2584-2919
Vol.02, No.03 (2024)
pp. 180-194

Credit Card Fraud Detection: A Comparative Study of Distance Metrics in Machine Learning

Vaishnav Menon[1], R Jai Akash[1], and Niveditta Batra[2,*]

[1]*Department of Data Sciences and Analytics, Ramaiah University of Applied Sciences, Bengaluru, 560054, Karnataka, India.*

[2]*Department of CSE-IT, Jaypee Institute of Information Technology, Noida, 201304, Uttar Pradesh, India.*

[*]Email: niveditta.batra@gmail.com (corresponding author)

(Received:17-11-2024; Accepted: 20-12-2024; Published Online:21-03-2025)

ABSTRACT:
Background: Credit card fraud detection is a critical problem due to the increasing volume of online transactions and the high costs associated with fraudulent activities. Previous studies in this field have investigated various machine-learning techniques to identify fraudulent transactions, with notable progress made through supervised learning methods. However, these models often face challenges due to the significant class imbalance in fraud detection datasets, where instances of fraud are much less frequent than legitimate transactions.
Purpose: As a result, there is growing interest in unsupervised techniques, such as clustering algorithms, which do not depend on labeled data and may offer improved generalization to new and unseen fraud patterns. These unsupervised approaches can autonomously identify anomalies by grouping transactions based on shared characteristics, making them a valuable alternative for detecting evolving fraudulent activities.
Methods: This work explores different distance metrics in clustering algorithms such as K-Means to identify fraudulent activity in a credit card dataset. The substantial class imbalance is highlighted by the European credit card transactions dataset, which consists of only 0.17% of fraudulent transactions. The research utilizes multiple sampling techniques to address class imbalance.
Results: The study found that the Euclidean distance metric produced the best results out of all potential techniques when applied to the K-Means algorithm. It emphasizes how crucial it is to deal with class disparities and use unsupervised methods for fraud detection in practical settings.
Conclusions: In future research, there is scope for improvements in fraud detection systems, particularly in terms of finding enhanced algorithms and expanding data availability.

Keywords: Credit Card Fraud detection, Distance Metrics in Machine Learning, unsupervised learning, k-means clustering, distance metrics, class imbalance

1. Introduction

Due to the growing dependence on electronic payment methods, credit card theft has become a major concern for businesses, consumers, and financial institutions globally. Every day, millions of transactions take place, and fraudulent activity not only causes large financial losses but also erodes customer confidence in online networks. The structure of the data makes addressing this difficulty much more difficult, as fraudulent transactions only make up a small portion of the total [1, 2]. The issue of class imbalance in transaction datasets, characterized by the rarity of fraudulent transactions relative to legitimate ones, necessitates the creation of precise detection models. These models must be capable of accurately identifying fraudulent activity while minimizing the occurrence of false positives. In previous works, there have been multiple techniques used for credit card fraud detection which mainly included supervised learning algorithms such as Decision Trees, Random Forest classifiers, and Logistic Regression [3, 4, 5]. These methods were used to classify the data into Fraud/Not Fraud data points. These works were mainly a comparative study to find the best supervised learning algorithm for credit card fraud detection [5, 6]. The proposed technique in this study involves the use of K-Means clustering algorithm, an unsupervised learning technique, which is a viable way to identify irregularities without the need for pre-labeled datasets. Additionally, our study delivers how crucial the distance metrics are for clustering accuracy, showcasing that the model performance can be greatly improved by comprehending their effects. The key contributions of this study can be summarised into the following objectives:

i) To assess how well the K-Means clustering method uses the Manhattan, Minkowski, and Euclidean distance metrics to detect fraudulent credit card transactions in extremely unbalanced datasets.

ii) To examine how different data sampling methods, such as SMOTE, oversampling, and undersampling, affect clustering performance and their capacity to rectify class imbalance in datasets used for fraud detection.

The dataset being analyzed consists of credit card transactions (both fraudulent and legitimate) made by cardholders in Europe in September 2013. There is a notable class imbalance in class, with 492 fraudulent transactions out of 284,807 total transactions, or just 0.172 percent of all transactions. To guarantee dataset quality and suitability for analysis, the methodology starts with data pre-processing, which includes addressing missing values and investigating dataset structure. After that, exploratory data analysis (EDA) techniques are used to learn more about feature characteristics, trends, and outliers. Visualization is used to examine the imbalance in the original dataset and the variation in transaction amounts. Several data sampling strategies are investigated in light of the dataset's extreme imbalance, including bootstrap resampling for undersampling and the SMOTE algorithm for oversampling. By either decreasing the overrepresented class or reproducing instances of the minority class, these strategies seek to balance the distribution of classes and address class imbalance. Further, a thorough study was conducted on different types of distance metrics i.e. Manhattan Distance, Euclidean Distance, and the Minkowski Distance. The distance metrics were applied to the four differently sampled datasets mentioned above. The results of which were compared to find the best distance metric to find the optimal value of k, for the K-means clustering analysis. The comparison study helped identify the Euclidean distance as the optimal distance metric to be used for K-Means clustering. The K-Means clustering algorithm is used in the study to find comparable groups of unlabeled data points. The dataset is clustered to expose underlying data structures, and then the fraud percentages and cluster distributions are visualized for each cluster. To compare results, the clustering technique is also applied to datasets that are under and oversampled. The entire dataset was not successfully used to classify transactions into fraud and legitimate categories, despite early successes in unsupervised clustering. This emphasizes the intricacy and difficulties involved in fraud detection tasks, underscoring the necessity of additional research and methodological improvement. In summary, this study advances our knowledge of credit card fraud detection by offering new perspectives on distance metrics, data sampling strategies, and the K-Means clustering algorithm. The study further highlights the necessity of additional research and methodological improvement in the field of credit card fraud detection.

2. Literature Review

There has been much research on the use of supervised learning techniques for credit card fraud detection, with different strategies tackling issues like feature selection, scalability, data imbalance, and real-time deployment [7]. Based on past data, the authors have successfully classified transactions as either legitimate or fraudulent using a multi-layer perceptron (MLP) architecture. In a study in 2017, the authors compared popular classifiers such as Naive Bayes, Random Forest, Decision

Trees, Support Vector Machines (SVMs), and Logistic Regression [3]. Although the classification of text reviews was the main focus, fraud detection can benefit greatly from the insights. The following year, a study with the same techniques was used explicitly for credit card fraud detection. The study highlighted standard classifiers such as Decision Trees, Random Forest, Logistic Regression, and SVMs, emphasizing their scalability and relevance to real-world applications. The study discussed in [8] was among the first to use neural networks in the field of credit card fraud detection through machine learning. This was one of the first to show that neural networks could be used to detect fraud because of their capacity to represent non-linear relationships in transactional data. Using actual transactional data, the study highlighted usefulness and offered insightful information on fraud detection at a time when such applications were still in their infancy [8]. Illebari et.al incorporated supervised learning with a Genetic Algorithm (GA) for feature selection, offering a more sophisticated investigation of fraud detection [9]. Using extensive metrics like accuracy, precision, recall, and F1-score, the study assessed several classifiers, including Random Forest, Logistic Regression, and SVMs. By lowering the dimensionality of the dataset, the application of GA for feature selection greatly enhanced model performance. Because of its ability to effectively handle imbalanced data and its resistance to overfitting, Random Forest turned out to be the best-performing algorithm. In 2023, a study centered on using automation and machine learning technologies to detect fraud in real-time [10]. Important topics like unbalanced datasets and dynamic transaction environments were covered in the paper. With a focus on real-world implementation, this study combined multiple AI methods for credit card fraud detection. It prioritised practical applicability by emphasising metrics like precision and recall. The evaluation framework placed a strong emphasis on striking a balance between false positives and negatives.

From handling imbalanced datasets and dynamic environments in recent studies to overcoming computational constraints in early works, the development of supervised learning techniques in credit card fraud detection reflects an increasing focus on adapting to real-world challenges [4, 5, 6, 7]. To increase the resilience and applicability of fraud detection systems, future research should keep filling in the gaps in scalability and computational efficiency while investigating innovative strategies like transformers. Although supervised learning approaches for fraud detection have been thoroughly studied in the past, these approaches frequently rely on labeled datasets, which aren't always accessible or accurately reflect real-world situations. The class imbalance feature of fraud detection datasets can also be a problem for them.

By using clustering algorithms like K-Means and investigating how distance metrics can increase clustering effectiveness, this study, on the other hand, focuses on an unsupervised learning approach. This work aims to address class imbalance while detecting patterns in fraudulent transactions without depending on predetermined labels by utilizing sophisticated sampling techniques like SMOTE. By providing a scalable and adaptable method to identify fraud in extremely dynamic and unbalanced datasets, this innovative approach seeks to supplement current supervised methodologies.

3. Methodology

Our approach to analyzing the credit card fraud dataset uses several key steps aimed at uncovering patterns, anomalies, and potential indicators of fraudulent behavior. The steps involved in the implementation have been summarized as follows:

Algorithm: Fraudulent Transaction Detection

Input: Dataset: European dataset containing transaction information.

Output: Clusters of transactions: Transactions categorized into fraudulent and non-fraudulent clusters.

Steps:

i Implementation of K-Means algorithm on the entire dataset.

ii Sampling the dataset into 10000 data points which includes all 492 fraudulent transactions, and applied the K-Means algorithm (with k=5

Table 1: Summarization of various papers on credit card fraud detection

Paper	**Key Contributions**	**Drawbacks**
Credit card fraud detection with a neural-network[7]	Neural Network implementation, Handling high dimensionality,Real-world dataset usage.	Computational limitations, Limited evaluation metrics, Imbalanced data handling.
Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification[3]	Comprehensive classifier comparison, Performance metrics.	No pre-processing considerations, Lack of advanced techniques.
A machine learning based credit card fraud detection using the GA algorithm for feature selection [9].	Domain-Specific Application, Genetic Algorithm for Feature Selection, Highlighting Random Forest.	Computational Overhead, Overfitting risk, Scalability concerns.
CreditCard Fraud Detection using Machine Learning(AI) [10]	Holistic Machine Learning, Focus on Imbalanced data, Integration of AI techniques.	Scalability Concerns, Dataset dependency, Lack of specificity on Model optimization.

which indicates the fraudulent transaction percentage in each cluster).

iii Reducing the data to 10% of the data points and applying K-means again, keeping the same fraudulent transactions in data.

iv Improvement of dataset through oversampling, undersampling, and SMOTE techniques to create a more balanced dataset.

v Re-applying K-Means on the three different datasets to compare the results.

Now, section 3.1 discusses the data and the environmental requirements needed for the implementation. The mathematical operations and an explanation for the steps involved in the above-mentioned implementation technique have been provided in detail in subsection 3.2.

3.1. Data and prerequisites

The dataset consists of a collection of credit card transactions made by European cardholders in September 2013. It covers the transactions made in two days, containing both legitimate and fraudulent transactions. There are 492 fraudulent transactions out of a total of 284,807 transactions, indicating a significant class imbalance, with fraudulent transactions representing only 0.172 percent of the total.

The dataset consists solely of numerical input variables resulting from a Principal Component Analysis(PCA) transformation. Features V1 through V28 are principal components derived from PCA, with 'Time' and 'Amount' being the only features not subjected to PCA transformation. 'Time' indicates the elapsed seconds between each transaction and the first one in the dataset, while 'Amount' represents the transaction amount, potentially useful for example-dependent cost-sensitive learning. The 'Class' feature serves as the response variable, taking a value of 1 for fraudulent transactions and 0 otherwise [2].

For setting up the environment, a quad-core, 64-bit processor, with 4 GB of RAM, and 80 GB of disk space was needed for optimal performance. Concerning software functions, Python was used for both the programming language and the integrated development environment (IDE).

3.2. Implementation technique

At first, we used the KMeans algorithm on the whole dataset where we got an output of different clusters indicating fraudulent and non-fraudulent transactions which were differentiated by colors. Later, we sampled the dataset into 10000 data points which includes all 492 fraudulent transactions, and applied the KMeans algorithm with a 'k' value of 5 which indicates the fraudulent transaction percentage in each cluster. Then we applied the same algorithm to 1000 data points in which we got the same output. During the process of analysis, a certain level of bias was found. This inference made us take a different approach toward data sampling and later implemented strategies like oversampling, under-sampling, and SMOTE for a fair sampled dataset. subsubsectionData pre-processing We begin by acquiring and preprocessing the dataset, ensuring its quality, consistency, and suitability for analysis. Import all the necessary packages and load the dataset. We check for missing values in the dataset and use the info() function to determine the structure of the dataset, identifying missing values and deciding how to handle them in your analysis.

The dataset exhibits strong data integrity, with each column containing 284,807 non-null entries, ensuring there are no missing values. This completeness is critical for the effective utilization of all features without the need for imputation or the removal of rows or columns. Examining the data types reveals that most columns are of type float64, which is expected since the features (V1 to V28) result from a PCA transformation, representing continuous variables. Principal component analysis (PCA) reduces the number of dimensions in large datasets to principal components that retain most of the original information. It does this by transforming potentially correlated variables into a smaller set of variables, called principal components [11, 12, 13]. The Class column, categorizing entries as fraud (1) or non-fraud (0), is of type int64. The dataset's memory usage stands at 67.4 MB, indicating it is relatively large but manageable within typical modern computing resources. Understanding memory usage is vital for optimizing performance, especially when considering scaling up or deploying models.

Performing an initial data check using $main_df.info()$ is a fundamental first step in any data analysis workflow. This check provides a quick overview of the dataset's structure and quality, helping to identify any immediate issues such as missing values or incorrect data types. Confirming the absence of missing values and understanding the data types allows for confident progression to subsequent preprocessing steps, such as scaling numerical features and addressing the class imbalance. Moreover, recognizing that all features are numerical simplifies feature engineering, enabling the direct application of scaling techniques and the consideration of all features for modeling without requiring additional transformations.

3.2.1. Data Exploration

The Data Frame initially displays the counts of each class, where the Class column is relabelled as "Non-Fraudulent" and "Fraudulent". The dataset comprises 284,315 non-fraudulent transactions and 492 fraudulent transactions, showcasing a significant class imbalance with approximately 0.17% representing fraudulent transactions [2].

Further analysis involves calculating descriptive statistics for transaction amounts in both categories. For fraudulent transactions, the count is 492, with a mean of 122.21 and a standard deviation of 256.68. The range spans from 0.00 to 2125.87, with quartiles positioned at 1.00 (25th percentile), 9.25 (median), and 105.89 (75th percentile). Conversely, non-fraudulent transactions amount to 284,315, with a mean of 88.29 and a standard deviation of 250.11. The transaction range mirrors that of fraudulent transactions, from 0.00 to $25,691.16$, with quartiles distributed at 5.65, 22.00, and 77.05.

These statistics unveil the variability in transaction amounts for both classes. Despite fraudulent transactions having a higher mean (122.21) compared to non-fraudulent ones (88.29), both exhibit significant dispersion. Quartile analysis further elucidates the distribution of transaction amounts within each class.

3.2.2. Data Sampling

Since credit card fraud data is imbalanced due to the number of non-fraudulent transactions exceeding the number of fraudulent transactions; we used techniques such as oversampling and undersampling or methods that can handle the algorithm's unequal classes.

Oversampling is used when the amount of data collected is insufficient. The most popular technique

is SMOTE (Synthetic Minority Oversampling Technique), which creates a synthetic sample from randomly occurring samples in a minority class[14, 15, 16]. **SMOTE** (Synthetic Minority Class Oversampling Technique) creates a new instance of the minority class by creating a synthetic instance concatenated from the minority class instance[17, 18, 19]. It not only reiterates existing minority themes; instead, it creates new situations based on the characteristics of minorities. SMOTE works by selecting a minority class and its nearest neighbors and then creating new conditions along the line segment connecting neighbors in a particular location. By doing this, SMOTE effectively connects minority classes by creating synthetic data that is similar but not identical to existing minority classes. Solving the class uncertainty problem can improve the performance of machine learning models trained on heterogeneous data[20, 21].
Undersampling is a technique used in machine learning to solve class inequality problems where one class (usually a minority class) is less represented than other classes. In case of undersampling, you reduce the number of samples in the overrepresented class to equalize the class distribution. When we repeat the samples with changes, we are essentially creating new synthetic data points by combining the data in the new paper by selectively selecting and copying the samples of the existing minorities and comparing the data points of the existing minorities. This process aims to equalize the class distribution by reducing the number of classes to fit the majority of classes[16, 18, 21].

3.2.3. Distance Metrics

We use distance metrics to assess the degree of similarity between data points and illustrate how this concept is essential to perform clustering tasks in machine learning. Since the dataset is passed through a k-means clustering model, we determine the number of clusters individually with the help of distance metrics. We use three different distance metrics: the Euclidean distance, the Manhattan distance, and the Minkowski distance each with its own characteristics and use cases. These metrics are used to get the within-cluster sum of squares (WCSS) for each number of clusters which helps in finding the correct number of clusters i.e., elbow point, where the rate of decrease in WCSS significantly decreases [22, 23, 24, 25].

Minkowski Distance
The Minkowski Distance is a very generalized distance metric. By 'generalized' we are referring to the fact that the Minkowski Distance formula to calculate the distance between two points can be manipulated for calculating it in different ways [26, 27].

$$Dist_{XY} = \left(\sum_{k=1}^{d} |X_{ik} - X_{jk}|^{\frac{1}{p}}\right)^{p}$$

By setting p =1, we get the Manhattan Distance p=2, we get Euclidean Distance [25]. **Manhattan Distance**
The Manhattan distance calculates the absolute differences between two objects' coordinates [28, 29]. We can manipulate the Minkowski Distance formula as mentioned above to calculate the Manhattan distance by setting p = 1,[30, 31]

$$Dist_{XY} = |X_{ik} - X_{jk}|$$

Euclidean Distance
The Euclidean distance is used by the k-means clustering algorithm to gauge how similar two objects are to one another [22, 32]. It is also one of the most commonly used distance metrics [33]. We can use the Minkowski Distance formula and set p =2, to be able to calculate the Euclidean Distance [33, 34].

$$Dist_{XY} = \sqrt{\sum_{k=1}^{m} (X_{ik} - X_{jk})^2}$$

Elbow Method
The Elbow Method is a heuristic used in cluster analysis to count the number of clusters in a given data set. Choosing the elbow of the curve to represent the number of clusters to use involves plotting the explained variation as a function of the number of clusters. The number of principal components to describe a data set, for example, or the number of parameters in other data-driven models can be determined using the same process [35, 36, 37].
First, we calculate the number of clusters for the original pre-processed data using the Euclidean distance.

We use the same data and get the number of clusters using the Manhattan distance and Minkowski distance. Similarly, we use the distance metrics for the undersampled, oversampled, and SMOTE datasets and observe that regardless of the undersampling, oversampling, or SMOTE, each distance metric correctly identifies the number of clusters that represent the optimal trade-off between model complexity and clustering quality. These evaluations help in understanding the variance present in the data to improve the performance of the model across different sampling techniques [38, 39].

3.2.4. Algorithm

K-Means: The K-Means algorithm works iteratively to divide the dataset into K pre-defined unique, non-overlapping subgroups, or clusters so that each data point is a member of only one group. The goal is to maintain as much distance between the clusters and as much similarity between the intra-cluster data points as feasible [40, 41, 42]. The process involves grouping data points into clusters so that the total squared distance between the cluster's centroid—the arithmetic mean of all the data points in that cluster—is as small as possible. The homogeneity (similarity) of data points within a cluster increases with decreasing variation within the cluster[37, 43, 44, 45]. The objective function is given by,

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} ||(x^i - \mu_k||^2$$

The available centroids are used to provide data points for clustering to obtain the best centroids. Select centroids or cluster key points based on the data points already assigned to the cluster [37, 46, 47, 48]. We then use histograms to show the distribution of the group of data. We try to show a visualization that gives us an idea about the percentage of fraud in each group after using the algorithm. We then use the same integration process as before, which is created from oversampled, undersampled, and SMOTE datasets [15, 16, 17].
Initially, the necessary libraries are imported, including the implementation of KMeans from scikit-learn. The number of clusters is set based on the comparative analysis of the distance metrics where the Euclidean Distance gave the k values as 9, 3, 3, and 3 for the original, oversampled, undersampled, and SMOTE datasets respectively, represented by the n_clust variable. Next, the KMeans object is initialized with this parameter and n_init is set to 10 to ensure robust centroid initialization. The model is then fitted to the dataset represented by df_features and KMeans is used to assign labels to all data points based on similarity [49, 50]. This set of articles is stored in a different tag. Finally, a histogram is created to visualize the distribution of data points into groups, providing an understanding of how data are normally grouped based on their characteristics. This unsupervised clustering approach will help understand the underlying structure in a dataset, facilitating analysis or subsequent machine learning studies.

4. Results and Analysis

The results obtained from the project focus on the percentage of distinct types of fraud within each cluster for the different datasets and also the number of clusters best suited for the model with the datasets. We used 3 sampling techniques after pre-processing the data and analyzed the graph output about the number of clusters ('k') which is best suited for the model.
In the first series of experiments, we applied the distance metrics to the original pre-processed data. We used the elbow method with 3 different distance metrics on the dataset and analyzed it to get the best value of ('k') for the model. The graph in figure 1, shows that the within-cluster sum of squares (WCSS) decreases significantly up to 6 or 7 clusters, and the rate of decrease becomes less beyond this point. This suggests that the optimal number of clusters (k) is either 6, 7, or 8, as adding more clusters beyond this range does not significantly reduce the WCSS. Therefore, for the normal dataset, choosing k as 6, 7, or 8 provides a good balance between model complexity and clustering quality. The elbow point in figure 2 appears around k=8 or 9 clusters, indicating a stabilization in the WCSS. This implies that the optimal number of clusters (k) for the K-means model is 8 or 9. These values provide a reasonable compromise between detailed clustering

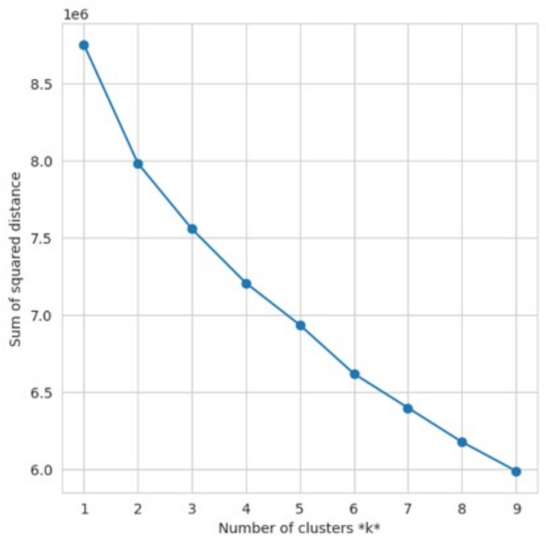

Figure 1: Elbow Method for K-Means Clustering using Euclidean Distance for Original Data

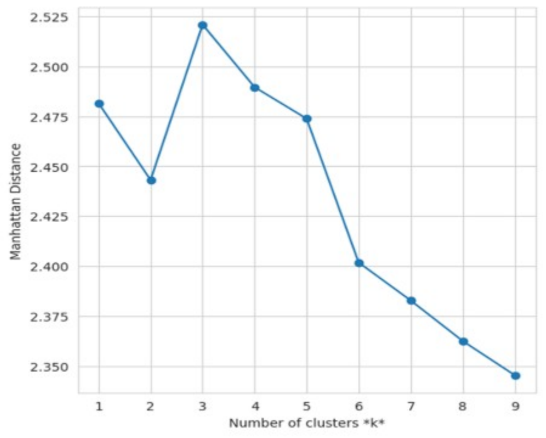

Figure 2: Elbow Method for K-Means Clustering using Manhattan Distance for Original Data

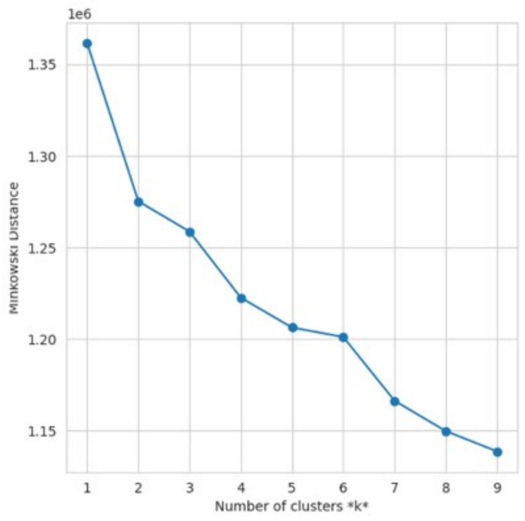

Figure 3: Elbow Method for K-Means Clustering using Minkowski Distance for Original Data

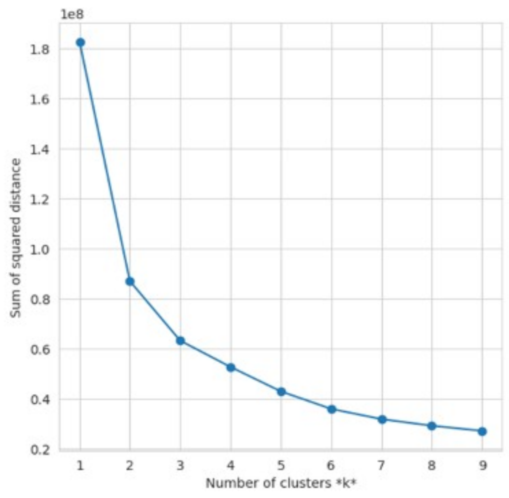

Figure 4: Elbow Method for K-Means Clustering using Euclidean Distance for Over sampled Data

and computational efficiency. In figure 3, the graph indicates that the WCSS stabilizes at around 7, 8, or 9 clusters. Thus, the optimal number of clusters (k) is 7, 8, or 9. These values ensure a good fit for the dataset without excessive computational load. Now, in the second series of experiments, we used the oversampled dataset with the same 3 distance metrics and analyzed it to get the best value of ('k') for the model. The graph in figure 4, shows stabilization at around 7, 8, or 9 clusters. Therefore, the optimal number of clusters (k) for the K-means model is 7, 8, or 9. These cluster numbers help capture the variability in the oversampled data while making the model less complex. However, in figure 5, the elbow point occurs around 6, 7, or 8 clusters. Consequently, the optimal number of clusters (k) is 6, 7, or 8. This range helps the model to work on oversampled datasets efficiently.

The graph in figure 10, shows stabilization around 8 or 9 clusters, suggesting these as the optimal number of clusters (k). This range efficiently captures the variations in the SMOTE dataset. The WCSS in figure 6, stabilizes at around 7 or 8 clusters. Therefore, the optimal number of clusters (k) for the K-means model in this case is 7 or 8. This provides a balanced approach to clustering the oversampled data.

Third experimentation series takes the undersampled dataset with the same 3 distance metrics and analyzes it to get the best value of ('k') for the model. The elbow point in figure 7, is evident around 8 or 9 clusters, indicating these as the optimal number of

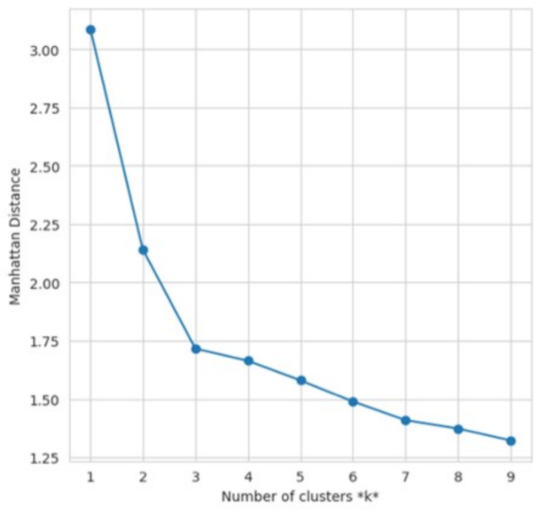

Figure 5: Elbow Method for K-Means Clustering using Manhattan Distance for Oversampled Data

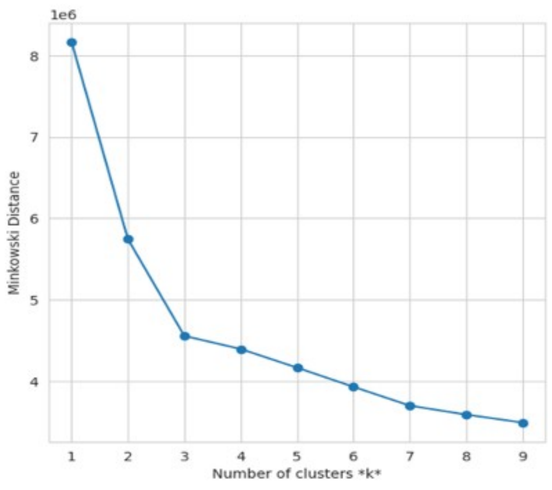

Figure 6: Elbow Method for K-Means Clustering using Minkowski Distance for Oversampled Data

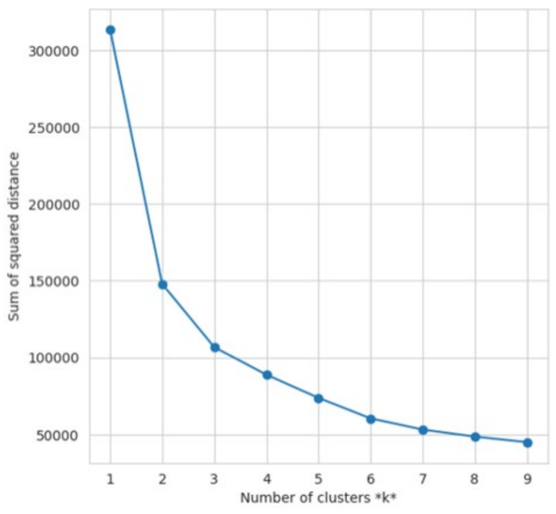

Figure 7: Elbow Method for K-Means Clustering using Euclidean Distance for Undersampled Data

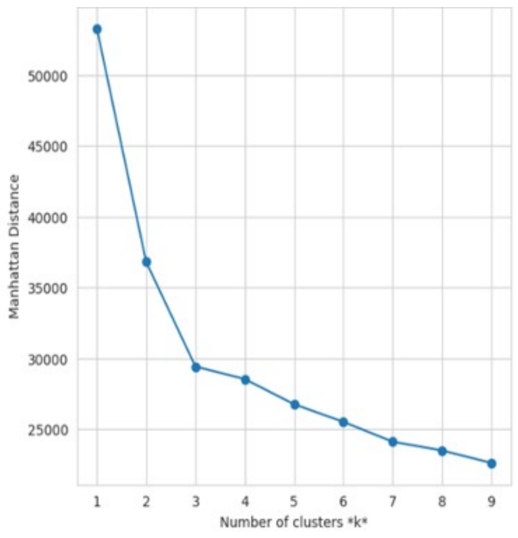

Figure 8: Elbow Method for K-Means Clustering using Manhattan Distance for Undersampled Data

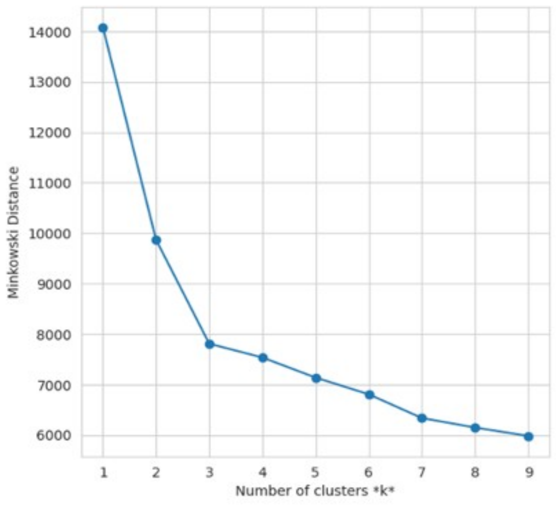

Figure 9: Elbow Method for K-Means Clustering using Minkowski Distance for Undersampled Data

clusters (k). This choice effectively helps in making correct inferences from the undersampled dataset.

The graph drawn in figure 8, indicates stabilization around 7 or 8 clusters. Thus, the optimal number of clusters (k) is 7 or 8, providing a detailed yet manageable clustering of the undersampled data.
The WCSS in figure 9, stabilizes at around 7 or 8 clusters. Consequently, the optimal number of clusters (k) is 7 or 8, balancing detail and making it less complex for the undersampled dataset.
At last, we used the SMOTE dataset with the same 3 distance metrics and analyzed to get the best value of (‘k’) for the model, The elbow point in figure 11,

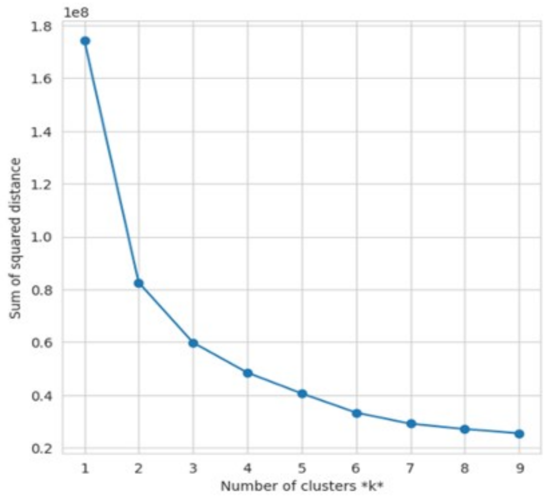

Figure 10: Elbow Method for K-Means Clustering using Euclidean Distance for SMOTE Data

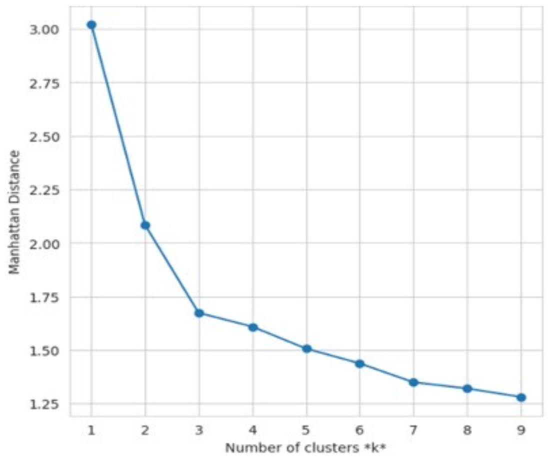

Figure 11: Elbow Method for K-Means Clustering using Manhattan Distance for SMOTE Data

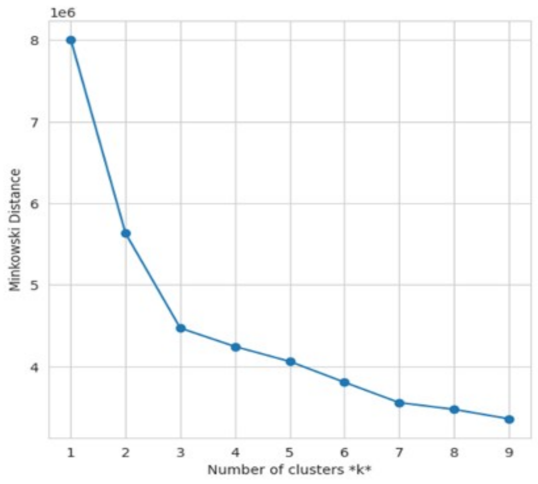

Figure 12: Elbow Method for K-Means Clustering using Minkowski Distance for SMOTE Data

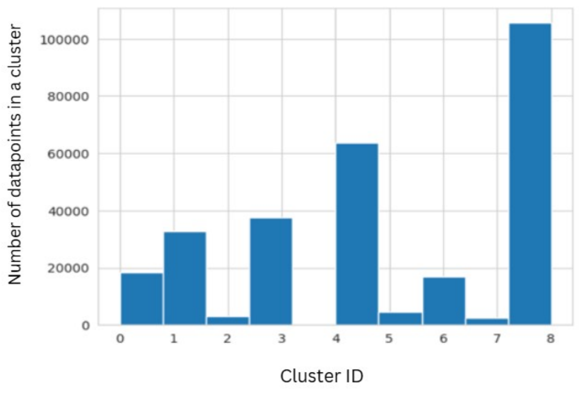

Figure 13: Clusters from the Original dataset

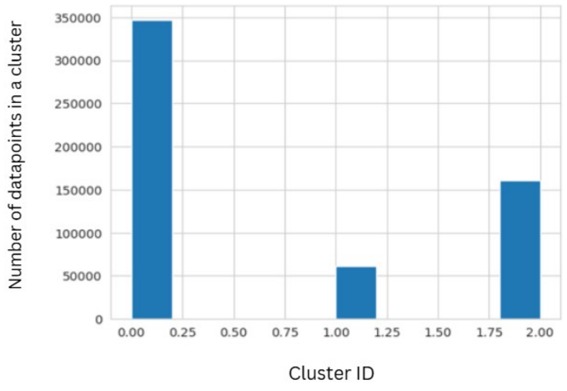

Figure 14: Clusters from the Oversampled dataset

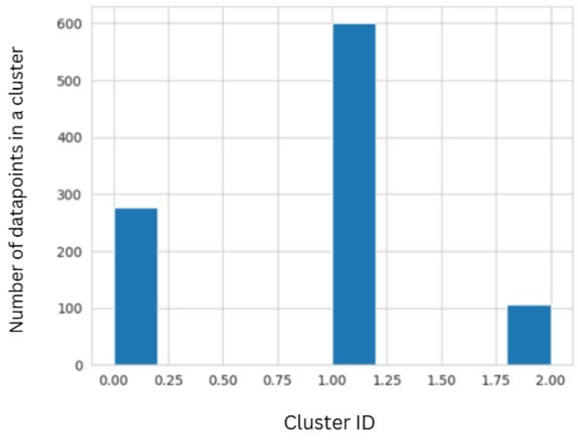

Figure 15: Clusters from the Undersampled dataset

appears at around 7 or 8 clusters. Thus, the optimal number of clusters (k) is 7 or 8, effectively clustering the SMOTE dataset. The WCSS in figure 12, stabilizes around 7 or 8 clusters. Therefore, the optimal number of clusters (k) is 7 or 8, ensuring a good balance between model complexity and performance for the SMOTE dataset.

K-Means Clustering result analysis on differently sampled datasets:

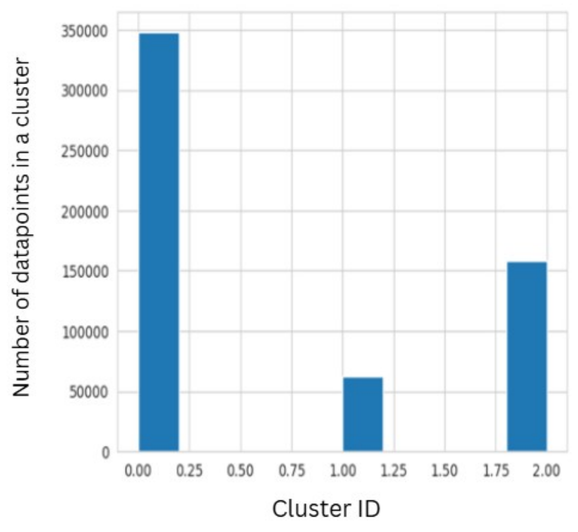

Figure 16: Clusters from the SMOTE dataset

After a thorough analysis using the K-Means clustering algorithm on four differently sampled datasets, namely, the original dataset, the oversampled dataset, the undersampled dataset, and the SMOTE dataset we were able to visualize the following histograms and were able to infer and come up with a very crucial assumption on the data that we used. The above histogram in figure 13, represents a total of 9 clusters for the original dataset, where each cluster contains a certain percentage of fraud data points, depicted in table 1.

The oversampled dataset consists of only three clusters where the K-means algorithm could determine only three clusters, as shown in table 2. A key point to notice here is that two separate clusters are completely consisting of fraud data points. The undersampled dataset was also clustered into 3 clusters, mentioned in table 3. (Note: Similar observation was made in the oversampled dataset.) The SMOTE dataset is clustered into three clusters as well, where we could observe the clusters as mentioned in table 4.

Table 2: Fraudulent cases distributed over various clusters in unsampled dataset

Cluster	Fraud
0	0.2%
1	0.0%
2	0.4%
3	0.4%
4	0.0%
5	3.5%
6	0.0%
7	1.0%
8	0.1%

Table 3: Fraudulent cases distributed over various clusters in Oversampled dataset

Cluster	Fraud
0	18%
1	100%
2	100%

Table 4: Fraudulent cases distributed over various clusters in Undersampled dataset

Cluster	Fraud
0	100%
1	18%
2	100%

Table 5: Fraudulent cases distributed over various clusters in SMOTE dataset

Cluster	Fraud
0	18.2%
1	100%
2	100%

5. Discussion

The results based on the comparative study of the three distance metrics i.e. Euclidean Distance, Manhattan Distance, and Minkowski Distance, as well as a sufficient amount of study in this field, led to the conclusion that the best distance metric to be used for K-Means clustering algorithm is the Euclidean Distance Metric, which finds the optimal k value using elbow method, this being true for all the differently sampled datasets. Thus, the values given by the Euclidean Distance Metric were taken to get the optimal cluster values for each dataset.

The comparisons in our study emphasize how important it is to choose the right clustering algorithms and distance measures when detecting credit card fraud. The study demonstrates the resilience of the Euclidean distance metric in producing reliable clustering findings by using Euclidean, Manhattan, and Minkowski distances across various sampled datasets. This result confirms that it works well with high-dimensional and PCA-transformed datasets.

The study offers important insights into how sampling strategies can affect clustering results because of its focus on correcting class imbalance using strategies including oversampling, undersampling, and SMOTE. The division of fraudulent transactions into discrete clusters, which raises the possibility of differences in fraudulent behavior patterns, is especially notable. This finding highlights the necessity for more thorough research into the traits of many fraudulent transactions and poses significant queries on the complexity of fraud kinds.

The study notes limitations such as the lack of transparency in the dataset and the lack of detailed fraud labels, which limits the capacity to classify fraud into particular categories, even if it has produced encouraging results. The availability of diverse, high-quality datasets that capture the intricacy of actual fraud situations is a larger problem in fraud detection research, highlighted by this constraint.

6. Conclusion

Building on the findings from our work, this study demonstrates that, among all the sampling procedures examined, the Euclidean distance metric is the most dependable option for K-Means clustering. Clustering quality was consistently optimized by the measure, as evidenced by the clusters created from the original, oversampled, undersampled, and SMOTE datasets. An important clue to the dataset's possible diversity in fraud types—an area ready for more research—is provided by identifying discrete clusters in the sampled datasets containing 100% fake data points. Although the study effectively determines the best clustering strategy for detecting credit card fraud, it is limited by the dataset's shortcomings, especially concerning diversity and granularity.

Thus, our study provides valuable direction for future research on detecting fraudulent credit card transactions. Expanding the dataset to include more specific information on fraudulent transactions would be one crucial direction for further research. Future research may significantly contribute to identifying more suitable clustering techniques and gaining a deeper insight into various fraud types using such data. Furthermore, the sampling strategies discussed here may be used on other datasets to check for comparable trends, especially concerning the clustering of fraudulent transactions. These beneficial developments have the potential to greatly enhance fraud detection while still being useful and accessible to both researchers and practitioners.

Acknowledgments
We acknowledge the cooperation and contributions of Laukik Chugh, Rayannagari Sree Manju Harsha, and Tanikaa V, without which this study would not have been possible.
Authorship contribution: The authors are accountable for the accuracy, originality, and integrity of the work.
Funding: No funding by any agency.
Conflict of interest: Author has no conflict of interest with anyone.
Declaration: We hereby declare that the work presented in this research paper titled ‘ Credit Card Fraud Detection: A Comparative Study of Distance Metrics in Machine Learning.’ is an original piece of work and has been carried out under the supervision/guidance of Ms. Niveditta Batra. All sources of information and data have been duly acknowledged, and the authors have adhered to the ethical standards and guidelines of research practice throughout the course of this study.
Similarity Index: The authors hereby confirm that there is no similarity index in abstract and conclusion while overall is less than 10% where individual source contribution is 2% or less than it.

References

[1] Rajeshwari, U., and B. Sathish Babu. "Real-time credit card fraud detection using streaming analytics." In 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), pp. 439-444. IEEE, 2016. https://doi.org/10.1109/ICATCCT.2016.7912039.

[2] Dataset: Andrea, “Credit Card fraud detection - Anonymized credit card transactions labeled as fraudulent or genuine” - Kaggle(2018). https://www.kaggle.com/.

[3] Pranckevičius, Tomas, and Virginijus Marcinkevičius. "Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification." Baltic

Journal of Modern Computing 5, no. 2 (2017): 221. https://doi.org/10.22364/bjmc.2017.5.2.05.

[4] Sahin, Yusuf, Serol Bulkan, and Ekrem Duman. "A cost-sensitive decision tree approach for fraud detection." Expert Systems with Applications 40, no. 15 (2013): 5916-5923.

[5] Awoyemi, John O., Adebayo O. Adetunmbi, and Samuel A. Oluwadare. "Credit card fraud detection using machine learning techniques: A comparative analysis." In 2017 international conference on computing networking and informatics (ICCNI), pp. 1-9. IEEE, 2017.

[6] Lakshmi, S. V. S. S., and Selvani Deepthi Kavilla. "Machine learning for credit card fraud detection system." International Journal of Applied Engineering Research 13, no. 24 (2018): 16819-16824.

[7] Ghosh, Sushmito, and Douglas L. Reilly. "Credit card fraud detection with a neural-network." In System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on, vol. 3, pp. 621-630. IEEE, 1994.

[8] Raj, S. Benson Edwin, and A. Annie Portia. "Analysis on credit card fraud detection methods." In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET), pp. 152-156. IEEE, 2011.

[9] Ileberi, E., Sun, Y. & Wang, Z. A machine learning based credit card fraud detection using the GA algorithm for feature selection. J Big Data 9, 24 (2022).

[10] Alonge, Dayo. "Credit Card Fraud Detection Using Machine Learning (AI)." (2023). https://www.researchgate.net/

[11] Lavado, N., & Calapez, T. (2011). Principal Components Analysis with Spline Optimal Transformations for Continuous Data. *IAENG International Journal of Applied Mathematics*,41(4).

[12] Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA).*Computers & Geosciences*19(3),303-342. https://doi.org/10.1016/0098-3004(93)90090-R.

[13] Ding, C., & He, X. (2004, July). K-means clustering via principal component analysis. *In Proceedings of the twenty-first international conference on Machine learning*,29.

[14] Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Machine Learning*, 113(7), 4903-4923.

[15] Dutta, P., Paul, S., & Majumder,M. (2021). An efficient SMOTE based machine learning classification for prediction & detection of PCOS. https://www.researchsquare.com/article/rs-1043852/v1.

[16] Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020, April). Machine learning with oversampling and undersampling techniques: overview study and experimental results. *2020 11th international conference on information and communication systems* (ICICS), 243-248, IEEE.

[17] Wongvorachan, T., He, S., & Bulut,O. (2023). A comparison of undersampling, oversampling, and SMOTE methods for classification dealing in with imbalanced educational data mining. *Information*, 14(1), 54, https://doi.org/10.3390/info14010054.

[18] Mansourifar, H., & Shi, W. (2020). Deep synthetic minority over-sampling technique. https://arxiv.org/abs/2003.09788.

[19] Mahin, M., Islam, M. J., Khatun, A., & Debnath, B. C. (2018, December). A comparative study of distance metric learning to find sub-categories of minority class from imbalance data. *international conference on innovation in engineering and technology*(ICIET), 1-6, IEEE.

[20] Chawla, N. V., Bowyer, K. W., Hall,L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique.*Journal of artificial intelligence research*,16, 321-357.

[21] Yen, S. J., & Lee, Y. S. (2009). Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3), 5718-5727.https://doi.org/10.1016/j.eswa.2008.06.108.

[22] Singh, A., Yadav, A., & Rana, A.(2013). K-means with Three different Distance Metrics. *International Journal of Computer Applications*, 67(10).

[23] Azevedo, B. F., Rocha, A. M. A.,& Pereira, A. I. (2023, September). A Collaborative Multi-objective Approach for Clustering Task Based on Distance Measures and Clustering Validity Indices. *International Conference on the Dynamics of Information Systems*, 54-68, Cham: Springer Nature Switzerland, https://doi.org/10.1007/978-3-031-50320-7_4.

[24] Jin, K., Li, Y., & Tang, Y. (2023,October). The Application of PCA and K-means Model in Detecting Anomalous Transformers. *IEEE 3rd International Conference on Data Science and Computer Application*(ICDSCA),690-695,IEEE, https://doi.org/10.1109/ICDSCA59871.2023.10393265.

[25] Sharma, P. (2020). Understanding distance metrics used in machine learning.https://www.analyticsvidhya.com/

[26] Lu, B., Charlton, M., Brunsdon, C.,& Harris, P. (2016). The Minkowski approach for choosing the distance metric in geographically weighted regression. *International Journal of Geographical Information Science*, 30(2),351-368, https://doi.org/10.1080/13658816.2015.1087001.

[27] Moghtadaiee, V., & Dempster, A. G. (2015). Determining the best vector distance measure for use in location fingerprinting. *Pervasive and Mobile Computing*, 23,59-79,https://doi.org/10.1016/j.pmcj.2014.11.002.

[28] Elgamel, M. S., & Dandoush, A.(2015). A modified Manhattan distance with application for localization algorithms in ad-hoc WSNs. *Ad Hoc Networks*, 33,

168-189.

[29] Faisal, M., & Zamzami, E. M.(2020,June).inter-centroid Comparative analysis of K-Means performance using euclidean distance, canberra distance and manhattan distance. *Journal of Physics:Conference Series* 1566(1), 12112, IOP Publishing.

[30] Crasta, G., & Malusa, A. (2007). The distance function from the boundary in a Minkowski space. *Transactions of the American Mathematical Society*, 359(12), 5725-5759,https://doi.org/10.1090/S0002-9947-07-04260-2.

[31] Kumar, R. (2017, October). Analysis of shape alignment using Euclidean and Manhattan distance metrics. *International Conference on Recent Innovations in Signal processing and Embedded Systems* (RISE),326-331,IEEE,https://doi.org/10.1109/RISE.2017.8378175.

[32] Elmore, K. L., & Richman, M. B.(2001). Euclidean distance as a similarity metric for principal component analysis. *Monthly weather review*, 129(3), 540-549.

[33] Dokmanic, I., Parhizkar, R.,Ranieri, J., & Vetterli, M. (2015). Euclidean distance matrices: essential theory, algorithms,and applications. *IEEE Signal Processing Magazine*,32(6),12-30. https://doi.org/10.1109/MSP.2015.2398954

[34] Gower, J. C. (1985). Properties of Euclidean and non-Euclidean distance matrices.*Linear algebra and its applications*, 67,81-97. https://doi.org/10.1016/0024-3795(85)90187-9

[35] Syakur, M. A., Khotimah, B. K.,Rochman, E. M. S., & Satoto, B. D. (2018, April). Integration k-means clustering method and elbow method for identification of the best customer profile cluster. *IOP conference series: materials science and engineering*, 336,12017, IOP Publishing.

[36] Ikotun, A. M., Ezugwu, A. E.,Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622, 178-210.

[37] Dabbura, I. (2018). K-means clustering: Algorithm, applications, evaluation methods,and drawbacks.https://towardsdatascience.com

[38] Rojas, J. A. R., Kery, M. B.,Rosenthal, S., & Dey, A. (2017, October). Sampling techniques to improve big data exploration. *IEEE 7th symposium on Large Data Analysis and Visualization (LDAV)*,26-35, IEEE.

[39] Koepsell, T. D., Martin, D. C.,Diehr, P. H., Psaty, B. M., Wagner, E. H., Perrin,E. B., & Cheadle, A. (1991). Data analysis and sample size issues in evaluations of community-based health promotion and disease prevention programs: a mixed-model analysis of variance approach. *Journal of clinical epidemiology*, 44(7),701-713. https://doi.org/10.1016/0895-4356(91)90030-D.

[40] Li, K., Cao, X., Ge, X., Wang, F.,Lu, X., Shi, M., ... & Chang, S. (2020). Meta-heuristic optimization-based two-stage residential load pattern clustering approach considering intra-cluster compactness and inter-cluster separation. *IEEE Transactions on Industry Applications*, 56(4), 3375-3384.

[41] Gharaei, N., Bakar, K. A., Hashim,S. Z. M., & Pourasl, A. H. (2019). Inter-and intra-cluster movement of mobile sink algorithms for cluster-based networks to enhance the network lifetime. *Ad Hoc Networks*, 85, 60-70. https://doi.org/10.1016/j.adhoc.2018.10.020

[42] dos Santos, T. R., & Zárate, L. E.(2015). Categorical data clustering: What similarity measure to recommend?. *Expert Systems with Applications*,42(3), 1247-1260.

[43] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7),881-892. https://doi.org/10.1109/TPAMI.2002.1017616.

[44] Na, S., Xumin, L., & Yong, G.(2010, April). Research on k-means clustering algorithm: An improved k-means clustering algorithm. *Third International Symposium on intelligent information technology and security informatics*, 63-67, IEEE.

[45] Bouhmala, N. (2016, July). How good is the euclidean distance metric for the clustering problem. *5th IIAI international congress on advanced applied informatics (IIAI-AAI)*,312-315,IEEE. https://doi.org/10.1109/IIAI-AAI.2016.26

[46] Zahra, S., Ghazanfar, M. A.,Khalid, A., Azam, M. A., Naeem, U., & Prugel-Bennett, A. (2015). Novel centroid selection approaches for KMeans-clustering based recommender systems. *Information sciences*,320,156-189. https://doi.org/10.1016/j.ins.2015.03.062.

[47] Leisch, F. (2006). A toolbox for k-centroids cluster analysis. *Computational statistics & data analysis*,51(2), 526-544.

[48] Fashoto, S. G., Owolabi, O.,Adeleye, O., & Wandera, J. (2016). Hybrid methods for credit card fraud detection using K-means clustering with hidden Markov model and multilayer perceptron algorithm. https://ir.kiu.ac.ug/.

[49] Jiang, B., Pei, J., Tao, Y., & Lin, X.(2011). Clustering uncertain data based on probability distribution similarity. *IEEE Transactions on Knowledge and Data Engineering*, 25(4), 751-763.

[50] Liu, C. L., Chang, T. H., & Li, H.H. (2013). Clustering documents with labeled and unlabeled documents using fuzzy semi-Kmeans. *Fuzzy Sets and Systems*, 221, 48-64. https://doi.org/10.1016/j.fss.2013.01.004.

Copyright and License

How to Cite?

Vaishnav Menon, R Jai Akash, & Niveditta Batra. (2024). Credit Card Fraud Detection: A Comparative Study of Distance Metrics in Machine Learning. *Graduate Journal of Interdisciplinary Research, Reports and Reviews*, 2(03), 180-194. Retrieved from https://jpr.vyomhansjournals.com/index.php/gjir/article/view/27

Authors ORCiD

Vaishnav Menon : https://orcid.org/0009-0000-5747-9583

R Jai Akash : https://orcid.org/0009-0009-6125-5043

Niveditta Batra: https://orcid.org/0009-0009-3552-852X