Anomaly Detection: A Comprehensive Overview of Methods and Applications of K-Means Based Detection Methods

Introducing the K-Means Based Anomaly Detection Algorithm and its Analysis on the Modified Version of UCI Housing Dataset

Wangjun Shen
College of Science, Engineering,
and Technology
The University of South Australia
Adelaide South Australia Australia
shewy009@mymail.unisa.edu.au

ABSTRACT

This paper offers a thorough overview of anomaly detection, including its importance and various approaches used to detect anomalies in data. The paper begins by defining anomaly detection and explaining why it has become increasingly crucial in the era of big data. The paper then introduces the three primary categories of anomaly detection methods: statistical, machine learning, and rule-based. It also provides an overview of clustering-based anomaly detection methods and compares their advantages and disadvantages. Next, the K-Means Based Anomaly Detection Algorithm is designed, which groups data objects into clusters using the k-means clustering algorithm and calculates an anomaly score for each object based on its distance to the closest centroid. Finally, the paper concludes with an analysis of the modified version of the UCI Housing dataset using the K-Means Based Anomaly Detection Algorithm. This algorithm is efficient, scalable, and effective in detecting anomalies in multidimensional spaces, making it a valuable unsupervised learning technique. Ultimately, the paper offers insights into appropriate methods for detecting anomalies based on the characteristics of the data.

CCS CONCEPTS

• Information systems → Data mining • Computing methodologies → Unsupervised learning • Computing methodologies → Anomaly detection

KEYWORDS

SAS EM, Anomaly Detection, Clustering-Based Anomaly Detection

ACM Reference format:

Shen, W. (2018). Anomaly Detection: A Comprehensive Overview of Methods and Applications of K-Means Based Detection Methods. In Proceedings of ACM Woodstock Conference (WOODSTOCK'18). ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/1234567890

1 Introduction

This section gives an introduction to anomaly detection, covering its significance and the various approaches used to detect anomalies in data. The initial part defines anomaly detection and elucidates why it has become increasingly vital in the age of big data. The next part sorts the three primary categories of anomaly detection methods: statistical, machine learning, and rule-based. The final part offers an overview of clustering-based anomaly detection methods and contrasts their advantages and disadvantages.

1.1 What is Anomaly Detection

Anomaly detection is a valuable technique that helps identify data points that significantly deviate from the expected behavior or norm. These deviations, also referred to as anomalies or outliers, may be caused by data errors, system faults, or fraudulent activity.

1.2 Why Use Anomaly Detection

Anomaly detection has become increasingly important in the era of big data. Its function is to identify unusual patterns within large amounts of data, which can be caused by anything from data errors to fraudulent activity or system faults. For instance, in the financial industry, anomaly detection is used to identify fraudulent transactions, which can help prevent financial losses. In healthcare, it is used to identify disease outbreaks, which can help prevent the spread of disease. In manufacturing, it is used to identify equipment failures, which can help prevent downtime and increase productivity. In general, anomaly detection has a wide range of applications, including improving the accuracy of predictions, eliminating the influence of outliers on analysis, and enhancing overall efficiency.

1.3 Types of Anomaly Detection Methods

There are three main categories of methods used to detect anomalies in data.

Statistical methods assume that the data follows a certain distribution and identify anomalies as any data point that deviates significantly from this distribution. This method is suitable for low-dimensional data with a clear statistical distribution. Machine learning methods use supervised or unsupervised learning to identify anomalies. Supervised learning uses labeled data to train a model that can identify anomalies in new data, while unsupervised learning uses unlabeled data to identify anomalies based on deviations from the expected pattern. This method is suitable for high-dimensional data or data with complex relationships. Rule-based methods use a set of rules to identify anomalies based on domain knowledge or statistical thresholds. It is suitable for situations where there is a clear set of rules that can be applied to identify anomalies.

In conclusion, anomaly detection is an essential technique used to identify unusual patterns in large amounts of data, and the appropriate method for anomaly detection depends on the data type, dimensionality, and expected pattern.

2 Review of Anomaly Detection Using Clustering

Clustering is a popular technique to detect anomalies in datasets. It involves dividing data points into clusters and using specific metrics to identify exceptional clusters as anomalous. This technique is useful for imbalanced datasets or limited instances of anomalies. There are different clustering methods for anomaly detection, such as distance-based, density estimation, subspace clustering, and hierarchical clustering. Each method has its strengths and limitations, depending on factors such as dimensionality, data sparsity, and computational resources.

2.1 Clustering-based Anomaly Detection Methods

Cluster-based anomaly detection methods involve dividing data points in a dataset into several clusters and then utilizing specific metrics to identify the clusters that stand out from the others. These exceptional clusters are then considered anomalous. This technique is generally utilized when the dataset is imbalanced or the number of anomalies is limited. [1]

2.2 Distance-Based Clustering Methods

Distance-based clustering methods detect anomalies by measuring the distance between data points and clustering them based on density thresholds. One of the most widely used distance-based clustering methods is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [2]. DBSCAN categorizes points as core, boundary, or noise points, with the latter being considered as anomalies. Another distance-based clustering method is OPTICS (Ordering Points to Identify the Clustering Structure) [3], which extends DBSCAN by providing a cluster-ordering representation and identifying anomalies as points with low reachability distances.

2.3 Density Estimation Methods

Density estimation methods are used to calculate the local density of each point and compare it to the densities of its neighbors to identify any anomalies. One of the most widely used density estimation methods is LOF (Local Outlier Factor) [4], which identifies anomalies as points with significantly lower densities than their neighbors. Another density estimation method is GLOSH (Global-Local Outlier Score Histogram) [5], which enhances LOF by incorporating global and local outlier scores to provide a more accurate representation of anomalies in high-dimensional data.

2.4 Subspace Clustering Methods

Subspace clustering methods detect anomalies by analyzing the local density within subspaces rather than the entire feature space. COF (Clustering with Outlier Factor) [6] is a widely used subspace clustering method that computes a subspace outlier factor to detect anomalies. NCMO (Nonlinear Clustering with Multiple Outliers) [7] applies subspace clustering to divide the feature space into subspaces and detect multiple outliers within each subspace.

2.5 Hierarchical Clustering Methods

Hierarchical clustering methods identify anomalies by examining the quality of cluster hierarchies and their stability. CLOPE (Clustering with Outliers by Maximizing the Prediction Strength) [8] employs hierarchical clustering to detect anomalies by measuring deviations from the expected number of instances within clusters. Another hierarchical clustering algorithm is CHAMELEON [9], which identifies anomalies by assessing the quality of cluster hierarchies and their stability.

2.6 Comparing Clustering-Based Anomaly Detection Methods

Distance-based clustering methods, such as DBSCAN and OPTICS, are robust in detecting clusters of varying shapes and sizes [10]. However, they may not perform well when dealing with high-dimensional data. Density estimation methods, such as LOF and GLOSH, are effective in identifying anomalies within clusters, but they may fail to detect anomalies in sparse regions [4]. Subspace clustering methods, such as COF and NCMO, are designed to handle high-dimensional data by analyzing local density within subspaces [11]. However, they may not be effective in detecting global anomalies. Hierarchical clustering methods, such as CLOPE and CHAMELEON, analyze the quality of cluster hierarchies and their stability [12]. However, they may be computationally expensive.

Overall, the selection of an appropriate method depends on the characteristics of the dataset, including its dimensionality, sparsity, and the nature of the anomalies. Researchers have proposed various modifications and extensions to these methods to improve their performance. The choice of a specific method may also depend on the availability of software implementations and computational resources [13].

3 K-Means Based Anomaly Detection Algorithm

The K-Means Based Anomaly Detection Algorithm groups data objects into clusters using the k-means clustering algorithm, then calculates an anomaly score for each object based on its distance to the closest centroid. It is simple, easy to implement, scalable for high-dimensional data sets, and effective in detecting anomalies in multidimensional spaces. However, it has limitations such as difficulty with non-convex cluster structures, sensitivity to noise and outliers, and the influence of initial centroid selection on clustering results. Alternative approaches may be needed for complex data sets or challenging outlier detection tasks. Overall, the K-Means Based Anomaly Detection Algorithm is a useful method with benefits and drawbacks.

3.1 Anomaly Score Definition

The anomaly score designed by me is calculated as the distance between a data object and its closest centroid, divided by the standard deviation of the distances between all data objects and their closest centroids.

This score is defined as follows:

$$\text{Anomaly Score} = \frac{d}{\sigma} \qquad (1)$$

d: *the distance to the closest centroid*

σ: *the standard deviation of the distances to the closest centroids*

This anomaly score can capture the anomalousness of data objects because it measures how far away a data object is from the center of its nearest cluster, relative to how dispersed the data is across all clusters. This means that data objects that are far away from any cluster center and/or are in sparsely populated regions of the data space are more likely to be identified as anomalies.

3.2 K-Means Based Anomaly Detection

Pseudo Code

```
Algorithm: KMBADN: a k-means based anomaly detection
Input: Data set X, number of clusters k, threshold value t
Output: Set of anomalies A
Methods:
  (1) Initialize centroids randomly
  (2) Assign each data object to its nearest centroid
  (3) Repeat until convergence:
  (4)     Update the centroids to be the mean of the assigned data objects
  (5)     Reassign each data object to its nearest centroid
  (6) Calculate the anomaly score for each data object
  (7) Identify all data objects with anomaly scores greater than t as anomalies
  (8) Output set of anomalies A
```

The K-Means Based Anomaly Detection Algorithm begins by initializing k centroids randomly. It then assigns each data object to its nearest centroid using Euclidean distance. The algorithm then iteratively updates the centroids to be the mean of the assigned data objects and reassigns each data object to its nearest centroid. This process continues until convergence is reached, i.e., when the centroids no longer move significantly.

After the convergence of the k-means clustering, the algorithm calculates the anomaly score for each data object using the formula described above. If the anomaly score of a data object is greater than the threshold value t, it is identified as an anomaly and added to the set of anomalies A.

The algorithm then outputs the set of anomalies A. This set contains all the data objects that are identified as anomalies based on their anomaly scores.

3.3 An Example for the Designed Algorithm

Consider the following two-dimensional data set consisting of 8 data objects: (2,1), (2,4), (3,2), (5,6), (6,5), (7,4), (8,5), (10, 10). Let's try to identify any anomalies in this data set using the K-Means Based Anomaly Detection Algorithm with k=2 and t=3. The algorithm first initializes two centroids randomly, say (3,2) and (7,4). It then assigns each data object to its nearest centroid and updates the centroids to be the mean of the assigned data objects. This process continues until convergence is reached. After convergence, the algorithm calculates the anomaly score for each data object. The distances of each data object to its closest centroid are as follows: (2,1): 1.41, (2,4): 2.24, (3,2): 0, (5,6): 2.24, (6,5): 1.41, (7,4): 0, (8,5): 1.41, (10, 10): 6.40. The standard deviation of these distances is approximately 1.886. Therefore, the anomaly scores of each data object are as follows with standard deviation: (2,1): 0.748, (2,4): 1.19, (3,2): 0, (5,6): 1.19, (6,5): 0.748, (7,4): 0, (8,5): 0.748, (10,10): 3.39. The recalculated anomaly scores show that (10,10) has an anomaly score of 3.39, which is greater than the threshold value of 3 and therefore is identified as an anomaly.

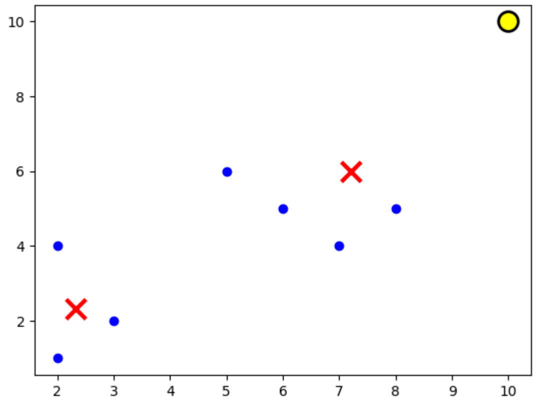


Figure 1: Visualization for the Example Data set

The cluster center refers to the center point found in the clustering algorithm, which represents the average value of the current cluster. This point can be used to represent the features of the current cluster. In the example below, the red cross represents the positions of two cluster centers. Different symbols represent different data points. Blue dots represent ordinary data points in the dataset, red crosses represent the found cluster centers, and yellow circles represent detected outliers. In this example, the outlier is data point (10,10). The algorithm calculates the outlier score for each data point based on the distance between the cluster center and each data point. If the outlier score is greater than the threshold t, it is identified as an outlier. In this example, the threshold t is set to 3, and the outlier score for data point (10,10) is 3.39, so it is identified as an outlier.

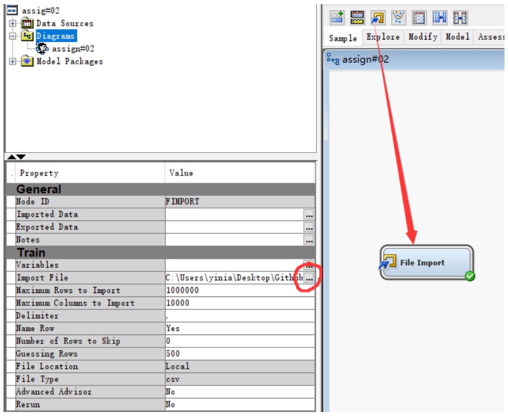
3.4 Advantages and Disadvantages

The advantages of the K-Means Based Anomaly Detection Algorithm are that it is relatively simple to implement, computationally efficient, and can be used with large data sets. Additionally, it does not require labeled data for training, making it a useful unsupervised learning technique. However, one major disadvantage of this algorithm is that it assumes that the data is normally distributed and that the clusters are spherical in shape, which may not be the case in real-world scenarios. Additionally, the algorithm requires the number of clusters k to be specified in advance, which can be difficult to determine in some cases. Finally, the algorithm can be sensitive to the initial placement of the centroids and may converge to suboptimal solutions depending on the initialization.

4 Use the k-means algorithm to detect anomalies in the dataset.

The modified version of the UCI Housing dataset includes housing information in the Boston suburbs and is used for applying k-means-based anomaly detection algorithms. SAS EM's cluster nodes and R language are used to implement a self-designed k-means algorithm to identify and analyze anomalies.

4.1 Import and Explore Data



Property	Value
General	
Node ID	FIMPORT
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Import File	C:\Users\yinia\Desktop\Github ...
Maximum Rows to Import	1000000
Maximum Columns to Import	10000
Delimiter	,
Name Row	Yes
Number of Rows to Skip	0
Guessing Rows	500
File Location	Local
File Type	csv
Advanced Advisor	No
Rerun	No

Figure 2: Import Dataset in Using SAS EM

First, create a SAM EM project and right-click on Diagrams to create a new diagram. Then, drag a File Import Node to this diagram and click on the "..." button next to "Import File" on the left side. Then, select the target dataset to import. Finally, right-click on the node to run it.

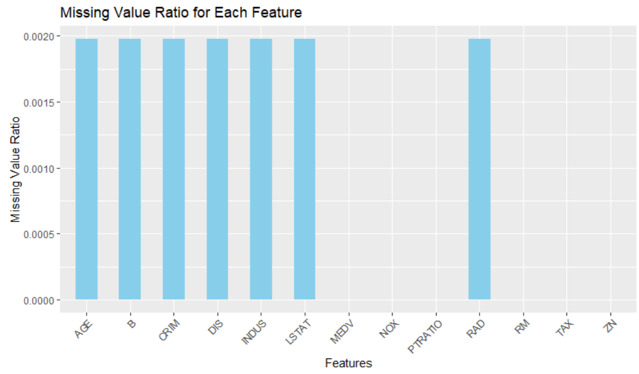
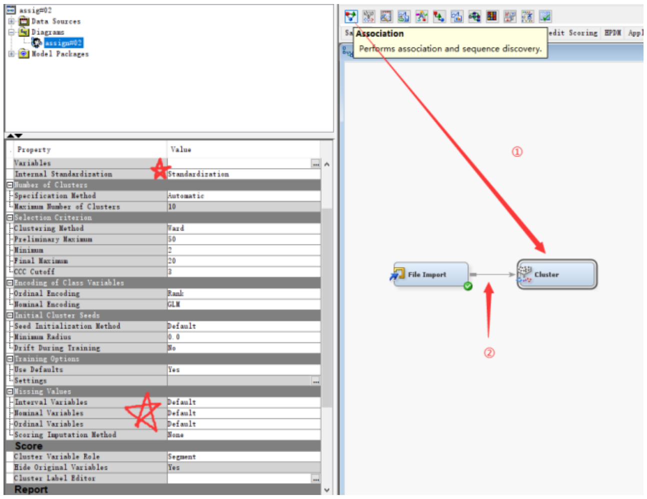


Figure 3: Visualization for Missing Values Ratio for Dataset

Based on the above results, it can be observed that most features have a missing value proportion of 0%, indicating that there are no missing values. However, a few features, such as CRIM, INDUS, AGE, DIS, RAD, B, and LSTAT, have a small proportion of missing values, approximately 0.20%. These missing values may impact data analysis and modeling. Therefore, to ensure accurate and reliable results in subsequent analysis, appropriate measures should be taken, such as removing samples that contain missing values or using interpolation methods to fill in missing values. In this case, the missing values processing method integrated in the Cluster Node was used, so no additional Impute Node was introduced for processing.

4.2 Cluster Data



Property	Value
Variables	...
Internal Standardization	Standardization
Number of Clusters	
Specification Method	Automatic
Maximum Number of Clusters	10
Selection Criterion	
Clustering Method	Ward
Preliminary Maximum	50
Minimum	2
Final Maximum	20
CCC Cutoff	3
Encoding of Class Variables	
Ordinal Encoding	Rank
Nominal Encoding	GLM
Initial Cluster Seeds	
Seed Initialization Method	Default
Minimum Radius	0.0
Drift During Training	No
Training Options	
Use Defaults	Yes
Settings	...
Missing Values	
Interval Variables	Default
Nominal Variables	Default
Ordinal Variables	Default
Scoring Imputation Method	None
Score	
Cluster Variable Role	Segment
Hide Original Variables	Yes
Cluster Label Editor	...
Report	

Figure 4: Cluster Data in SAS EM

To begin, drag the Cluster Node to the Diagram and connect it with the File Import Node. Keep in mind that the scale of different features within the raw data set may vary. Since the k-means algorithm is affected by distance, it's important to standardize the features to avoid any impact on the results. That's why "Standardization" is chosen as the parameter for "Internal Standardization". As for the Missing Values options, it's recommended to stick with the default parameters, "Default" and "None". When using SAS EM, "Default" means that any Missing Values will be ignored during the clustering process. This is a good choice when the dataset contains sufficient data, and the ratio of Missing Values is small enough.

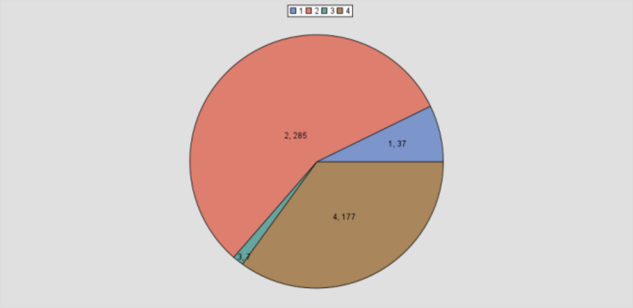
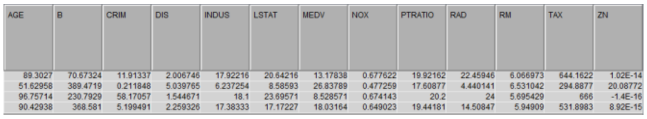


Figure 5: Segment Size for Each Cluster

According to the cluster analysis results, certain groups of samples in the data tend to be assigned to specific clusters. Specifically, Segment id = 2 appears 285 times in the clustering and Segment id = 4 appears 177 times. This indicates that these groups of samples share common characteristics or attributes that make them more closely aligned with the center of their respective clusters. Segment id = 1 appears only 37 times in the clustering, indicating that these samples differ in characteristics from other clusters, making it more likely for them to be assigned to Cluster 1. Finally, Segment id = 3 appears only 7 times in the clustering, which suggests that these samples possess significantly different characteristics from other clusters, making them more likely to be assigned to Cluster 3.



AGE	B	CRIM	DIS	INDUS	LSTAT	MEDV	NOX	PTRATIO	RAD	RM	TAX	ZN
89.3027	70.67324	11.91337	2.006746	17.92216	20.64216	13.17838	0.677622	19.92162	22.45946	6.066973	644.1622	1.02E-14
51.62958	389.4719	0.211848	5.039765	6.237254	8.58593	26.83789	0.477259	17.60877	4.440141	6.531042	294.8877	20.08772
96.75714	230.7929	58.17057	1.544671	18.1	23.69571	8.528571	0.674143	20.2	24	5.695429	666	-1.4E-16
90.42938	368.581	5.199491	2.259326	17.38333	17.17227	18.03164	0.649023	19.44181	14.50847	5.94909	531.8983	8.92E-15

Figure 6: Cluster heart for each cluster

The results of the cluster analysis have identified four segments, each with unique characteristics. Segment id 2.0 had the highest frequency, meaning that it contains the largest number of samples in the dataset. Segment id 3.0 had the highest root mean square standard deviation, indicating that these samples had a large variation within the cluster. On the other hand, Segment id 1.0 and Segment id 4.0 had a high maximum distance, which suggests the presence of outliers or anomalies within their clusters. Furthermore, Segment id 1.0 and Segment id 4.0 had a large distance to the nearest cluster, indicating that these samples had a significant difference from other clusters.

4.3 Anomaly Detection Using Standard Deviation

In the previous processing, steps (1) to (5) in the pseudocode have been completed using the Cluster Node provided by SAS EM.

Next, we need to calculate the Anomaly Score (refer to formula (1)), and then use the Anomaly Score for anomaly detection.

Firstly, calculate the distance between the data and the centroids of each cluster, choose the closest centroid from the four distances, and classify the data into that cluster.

The function is shown as below:

$$min_{dist} = min\|x_i - c_j\|, for\ j = 1\ to\ k \qquad (2)$$

Finally, use the distance between the data and the centroids of its cluster, as well as the standard deviation of the distances, to calculate the Anomaly Score.

$$Anomaly\ Score = \frac{\|x_i - c_{k(i)}\|}{\sigma_k} \qquad (3)$$

The value of Standard Deviation is 1.078073, to proceed, simply calculate the Anomaly Score for each data point. Keep in mind that any data point with an Anomaly Score greater than 3 should be considered an Anomaly data point.

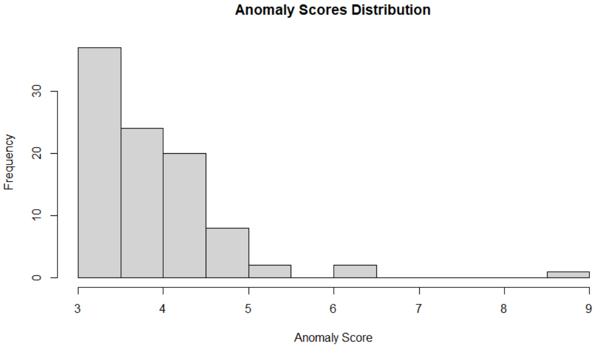


Figure 7: Anomaly Scores Distribution

From the visualization results, most of the Anomaly data have Anomaly scores distributed between 3 to 4, followed by 4 to 5. One interesting aspect is the existence of data with an Anomaly Score value between 8.5 to 9.

	data	mean
CRIM	23.6482	3.6207193
ZN	0	11.3142292
INDUS	18.1	11.1644554
NOX	0.671	0.5547168
RM	6.38	6.2819802
AGE	96.2	68.6144554
DIS	1.3861	3.7945671
RAD	24	9.5603960
TAX	666	408.4683794
PTRATIO	20.2	18.4549407
B	396.9	356.5927129
LSTAT	23.69	12.6711683
MEDV	13.1	22.5053360

Table 1: Explore that Special Anomaly Data

It is worth noting that the "CRIM" feature stands out as an anomaly with a value of 23.6482, which is significantly higher than its average of 3.6207193. Upon further analysis of the anomaly score, it is found that this particular data point has an anomaly score ranging from 8.5 to 9, which is considerably higher than other data points. This suggests that the "CRIM" feature behaves significantly differently compared to the rest of the dataset. This feature represents the per capita crime rate by town, and its substantially higher value may indicate an outlier or an area with an unusually high crime rate compared to the average. This finding may require further investigation into the underlying factors contributing to such a high crime rate and its potential implications on the housing market or other related socio-economic aspects. It's worth noting that the remaining features demonstrate relatively smaller deviations from their respective means. However, a more comprehensive analysis should be conducted to understand their individual characteristics and potential impact on the dataset.

5 Conclusion

In conclusion, anomaly detection is a crucial technique for identifying unusual patterns within large amounts of data, and the appropriate method for anomaly detection depends on the data type, dimensionality, and expected pattern. This paper provides a comprehensive overview of anomaly detection, including its significance, the various approaches used to detect anomalies in data, and the advantages and disadvantages of clustering-based anomaly detection methods. The K-Means Based Anomaly Detection Algorithm is also introduced, which is a useful unsupervised learning technique for detecting anomalies in multidimensional spaces. The algorithm is computationally efficient, scalable, and effective in detecting anomalies, making it a valuable tool for various applications. The analysis of the modified version of the UCI Housing dataset demonstrates the algorithm's effectiveness in detecting anomalies and identifying potential outliers. Overall, this paper provides insight into the appropriate methods for detecting anomalies based on the characteristics of the data, which is critical for accurate and reliable results in various applications.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection,” ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: https://doi.org/10.1145/1541880.1541882.

[2] International Conference On Knowledge Discovery And Data Mining, Portland, Or (United States), 2-4 Aug 1996 et al., A density-based algorithm for discovering clusters in large spatial databases with noise. Aaai Press, Menlo Park, Ca (United States, 1996.

[3] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “OPTICS,” ACM SIGMOD Record, vol. 28, no. 2, pp. 49–60, Jun. 1999, doi: https://doi.org/10.1145/304181.304187.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF,” ACM SIGMOD Record, vol. 29, no. 2, pp. 93–104, Jun. 2000, doi: https://doi.org/10.1145/335191.335388.

[5] R. J. G. B. Campello, P. Kröger, J. Sander, and A. Zimek, “Density-based clustering,” WIREs Data Mining and Knowledge Discovery, vol. 10, no. 2, Oct. 2019, doi: https://doi.org/10.1002/widm.1343.

[6] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, “Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data,” Advances in Knowledge Discovery and Data Mining, pp. 831–838, 2009, doi: https://doi.org/10.1007/978-3-642-01307-2_86.

[7] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, “On the Surprising Behavior of Distance Metrics in High Dimensional Space,” Database Theory — ICDT 2001, vol. 1973, pp. 420–434, 2001, doi: https://doi.org/10.1007/3-540-44503-x_27.

[8] N. Bolshakova and F. Azuaje, “Cluster validation techniques for genome expression data,” Signal Processing, vol. 83, no. 4, pp. 825–833, Apr. 2003, doi: https://doi.org/10.1016/s0165-1684(02)00475-9.

[9] G. Karypis, Eui-Hong Han, and V. Kumar, “Chameleon: hierarchical clustering using dynamic modeling,” Computer, vol. 32, no. 8, pp. 68–75, Aug. 1999, doi: https://doi.org/10.1109/2.781637.

[10] E. Knox and R. Ng, “Algorithms for Mining Distance-Based Outliers in Large Datasets.” Accessed: Jun. 16, 2023. [Online]. Available: https://www.vldb.org/conf/1998/p392.pdf

[11] H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, “Interpreting and Unifying Outlier Scores,” Proceedings of the 2011 SIAM International Conference on Data Mining, Apr. 2011, doi: https://doi.org/10.1137/1.9781611972818.2.

[12] L. Parsons, E. Haque, and H. Liu, “Subspace clustering for high dimensional data,” ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 90–105, Jun. 2004, doi: https://doi.org/10.1145/1007730.1007731.

[13] S. Guha, R. Rastogi, and K. Shim, “CURE,” ACM SIGMOD Record, vol. 27, no. 2, pp. 73–84, Jun. 1998, doi: https://doi.org/10.1145/276305.276312.