UDC 519.6

Unsupervised Detection of Anomalous Running Patterns Using Cluster Analysis

Ivan Ursul [1], **Andriy Pereymybida** [2]

[1] *Organization1,*
adress, e-mail: ivan.ursul@lnu.edu.ua
[2] *Organization2,*
adress, e-mail: andrii.pereimybida@lnu.edu.ua

Anomaly detection is an important problem in various domains such as user analysis, network intrusion detection, fraud detection and system monitoring. In this paper, the author provides a comprehensive review of anomaly detection algorithms for cluster applications. The author discusses different types of cluster-based algorithms and compares their performance based on metrics such as scalability, precision, recall, and f1 score. The paper provides a detailed analysis of various clustering techniques, including distance-based clustering, hierarchical clustering, and density-based clustering. It applies them to a customer data set to detect anomalies. The author also discusses the use of ensemble techniques and outlier ranking methods to improve the accuracy of anomaly detection in this data set. They compare the performance of these techniques using various evaluation metrics and provide a summary of the results. This work highlights the challenges of cluster-based anomaly detection such as selecting the appropriate number of clusters, dealing with high-dimensional data, and handling imbalanced datasets. The authors provide insights into how these challenges can be addressed and discuss future research directions in this field.

Key words: Clusters, Anomaly Detection, Unsupervised Learning

1. Introduction

In today's world, data in the modern world grows exponentially; scientists predict that the 'tsunami of data is coming in recent years' [1], as more users will join the global internet, as well as more signals will be gathered about them on a daily basis. With more data in the coming years, we expect to see an increased demand for automated data analysis [2]. Existing methods of data analysis are semi-automated; they require a certain amount of human interaction for doing the pre-processing: filtering the expected data, and finding the data that doesn't fit into any of the known patterns. The problem with finding the outlier data is that 'we do not know what we do not know': systems can be built to serve one set of functions, while they will later be used by bad actors for completely different purposes. The problem of limited knowledge of the data perfectly describes the problem of unsupervised learning: how to find an outlier if you do not know how it looks. It is worth mentioning that not all anomalies are bad: Anomaly detection can be used to identify customers who exhibit unique behaviors or preferences that may not be captured by traditional segmentation methods.

Cluster-based anomaly detection could be an answer to the mentioned problems. The idea is to cluster similar data points together and identify outliers that donвЂ™t fit perfectly well into the specific cluster. Depending on the type of clustering algorithm, a different method for detecting an outlier is used. This science paper is focused on partition-based, density-based and hierarchical clustering algorithms. The main challenge in the field of cluster-based anomaly detection is to find an approach that could work well for large-scale data, will be insensitive to the choice of input parameters and can provide insightful information about anomalies.

2. Review of clustering algorithms for anomaly detection

2.1. Partition-based clustering

Partition-based clustering Partition-based clustering is a commonly used clustering algorithm for anomaly detection. Partition-based algorithms such as K-Means [3] involve dividing the dataset into K clusters based on the similarity of data points. While these algorithms are effective in identifying patterns and grouping similar data points, they may not always be able to detect anomalies. To identify anomalies in K-Means [3] clustering, one approach is to define a threshold value based on the within-cluster sum of squares (WCSS) for each cluster. The WCSS measures the sum of the squared distances between each data point and the centroid of its assigned cluster. To detect anomalies, we can calculate the WCSS for each cluster and define a threshold value that is above the average WCSS for all clusters. Any data point with a WCSS value that is above this threshold can be considered an anomaly. Formally, let $C_1, C_2, ..., C_k$ be the K clusters generated by the K-Means [3] algorithm, and let $S_1, S_2, ..., S_k$ be the corresponding sum of squares for each cluster. The average WCSS, denoted by S_{avg}, can be calculated as:

$$S_{avg} = \frac{(S_1, S_2, ..., S_k)}{K} \tag{1}$$

Next, we define a threshold value, T, which is a multiple of the standard deviation of the sum of squares for each cluster. The threshold value can be calculated as:

$$T = S_{avg} + k * SD(S_1, S_2, ..., S_k) \tag{2}$$

where SD is the standard deviation function, and k is a constant that determines the sensitivity of the anomaly detection.Finally, any data point whose WCSS value is above the threshold value T can be identified as an anomaly. This approach can identify anomalies in partition-based algorithms such as K-Means [3], allowing for more robust and comprehensive data analysis. The Visualization of K-Means clustering is provided in figure 1.

2.2. Density-based clustering

Density-based clustering is another commonly used clustering algorithm for anomaly detection. The problem of density-based clusterings, like Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for anomaly detection, can be formally defined as follows: Given a dataset $X = \{x_1, x_2, ..., x_n\}$ consisting of n data points, where each data point x_i is represented by a d-dimensional feature vector $xi = [x_{i1}, x_{i2}, ..., x_{id}]$, the objective of density-based clustering for anomaly detection is to identify clusters of data points that have a high density of neighbouring data points and detect anomalies that have a low density of neighbouring data points.

To detect anomalies using DBSCAN [4], two parameters are used: epsilon (ϵ) and minimum points (MinPts). Epsilon is the maximum distance between two data points for them to be considered neighbours, and MinPts is the minimum number of neighbouring data points for a data point to be considered a core point.

A core point is a data point with at least MinPts neighbouring data points within a distance of ϵ. A border point is a data point that does not have enough neighbouring

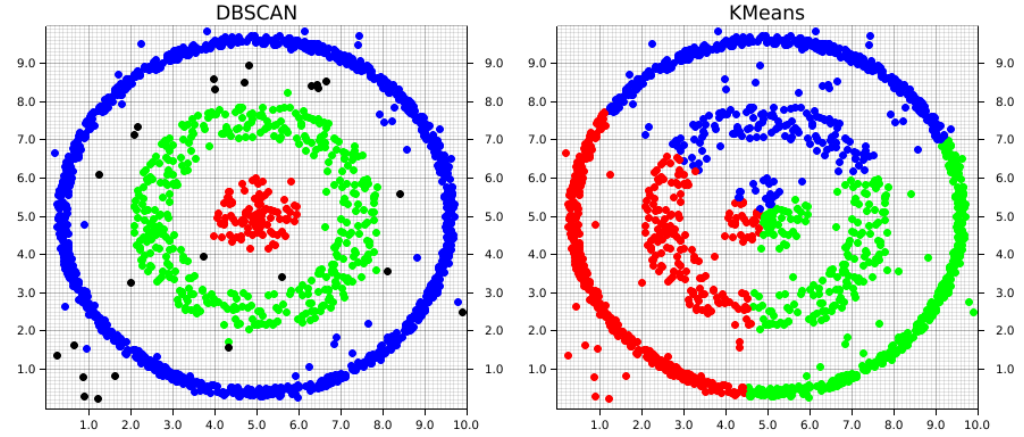

Fig. 1. The Visualization of Comparison of K-Means Clustering and DBSCAN image sources: [9]

data points to be a core point but is within a distance of ϵ from a core point. A noise point is a data point not a core or a border point.

DBSCAN [4] starts by selecting a random core point and finding all neighbouring core points within a distance of ϵ. These core points are then merged into a cluster. The process is repeated until all core points have been assigned to clusters. Border points are assigned to the cluster of their nearest core point, and noise points are not assigned to any cluster.

After clustering the data points, anomalies can be identified as noise points or data points that belong to clusters with a small number of data points.

The problem of density-based clustering like DBSCAN [4] for anomaly detection can be formulated as a binary classification problem, where the objective is to classify each data point as either an anomaly or a normal data point. DBSCAN [4] can be used in various applications, such as fraud detection, intrusion detection, and fault diagnosis. Let $C = \{C_1, C_2, ..., C_k\}$ be the set of all clusters identified by DBSCAN [4], where k is the total number of clusters. Let $S = \{S_1, S_2, ..., S_m\}$ be the set of all core points in the dataset, where m is the total number of core points. Let $B = \{B_1, B_2, ..., B_l\}$ be the set of all border points in the dataset, where l is the total number of border points. Then, the objective function of DBSCAN [4] can be written as:

$$C = \{C_1, C_2, ..., C_k\} = \{S_1 \cup B_1, S_2 \cup B_2, ..., S_k \cup B_l\} \tag{3}$$

Each C_i is a cluster, defined as the union of a core point and its corresponding border points. The number of clusters k is unknown in advance and may vary depending on the data and the chosen values of ϵ and minPts. DBSCAN [4] is a powerful tool for anomaly detection that can be used to identify clusters of data points with a high density of neighbouring data points and detect anomalies with a low density of neighbouring data points.

2.3. Hierarchical-based clustering

Hierarchical clustering is a commonly used unsupervised machine learning algorithm for anomaly detection. The problem of hierarchical clustering for anomaly detection can

be formally defined above in 2.2. To detect anomalies using hierarchical clustering, a distance metric is used to measure the similarity or dissimilarity between data points. One common distance metric used for anomaly detection is the Mahalanobis distance, as defined in the problem of agglomerative hierarchical clustering.

Hierarchical clustering can be performed using two approaches: agglomerative clustering and divisive clustering. Agglomerative clustering starts with each data point as a separate cluster and iteratively merges the closest clusters until a stopping criterion is met. Divisive clustering starts with all data points in a single cluster and recursively splits it into smaller clusters until a stopping criterion is met.

The similarity or dissimilarity between two clusters is measured using a linkage criterion, which defines the distance between two clusters. Different linkage criteria can be used for anomaly detection, such as single linkage, complete linkage, average linkage, and Ward's linkage. The linkage criterion used can affect the structure of the resulting hierarchy and the quality of the clustering.

After clustering the data points, anomalies can be identified as data points that do not belong to any of the clusters or belong to clusters with a small number of data points. The threshold for determining the size of a cluster can be set based on domain knowledge or using statistical methods such as the Elbow method or the Silhouette method.

The objective function of hierarchical clustering for anomaly detection can be written as:

$$minimize \sum C_i \in C \sum x_i, x_j \in C_i d_i, j \tag{4}$$

where k is the number of clusters, C_i is the i-th cluster, and d_i, j is the distance between data points x_i and x_j. The goal is to minimize the within-cluster dissimilarity and maximize the between-cluster dissimilarity.

3. Datasets used

In this study, we used a dataset of running activity data recorded by a wearable device. The dataset includes the following columns: datetime, athlete, distance, duration, gender, age group, country, and major. The datetime column indicates the date and time of the running activity, while the athlete column identifies the individual who performed the activity. The distance and duration columns indicate the distance and duration of the running activity, respectively. The gender and age group columns provide demographic information about the athlete, while the country and major columns provide additional background information. The dataset is balanced, with a total of 10,703,690 running activities recorded. It also contains a subset of false activity labels, where individuals recorded an activity without actually making the running session.

The pace(duration/distance) distribution chart helps us understand that the median pace is around 5:20, which is slightly higher than the average pace around the recreational runners, according to multiple studies. The analysis of the distance shows that the majority of running activities are within 5-10 kilometers range. Additionally, the dataset includes data about top performers, who may exhibit different running patterns than the general population. It is worth noting that the duration and distance of the running activities are quite imbalanced, with a wide range of values recorded. Specifically, the dataset includes running sessions for marathons, half-marathons, 10 and 5 kilometers were tracked. The balanced nature of the dataset, the variation of distances, along with the presence of false activity labels and top performer data, poses significant challenges for anomaly detection.

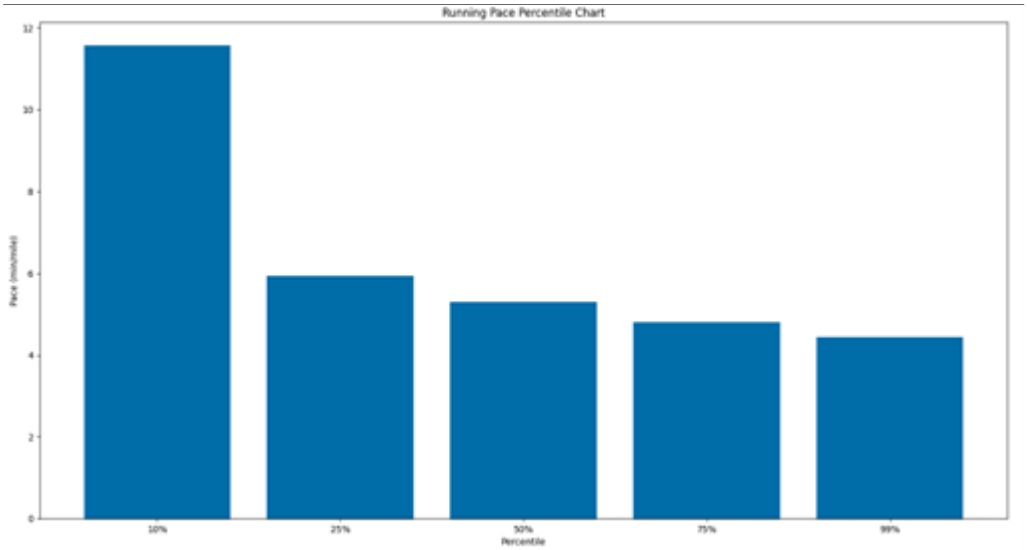

Fig. 2. Running Pace Percentile Chart

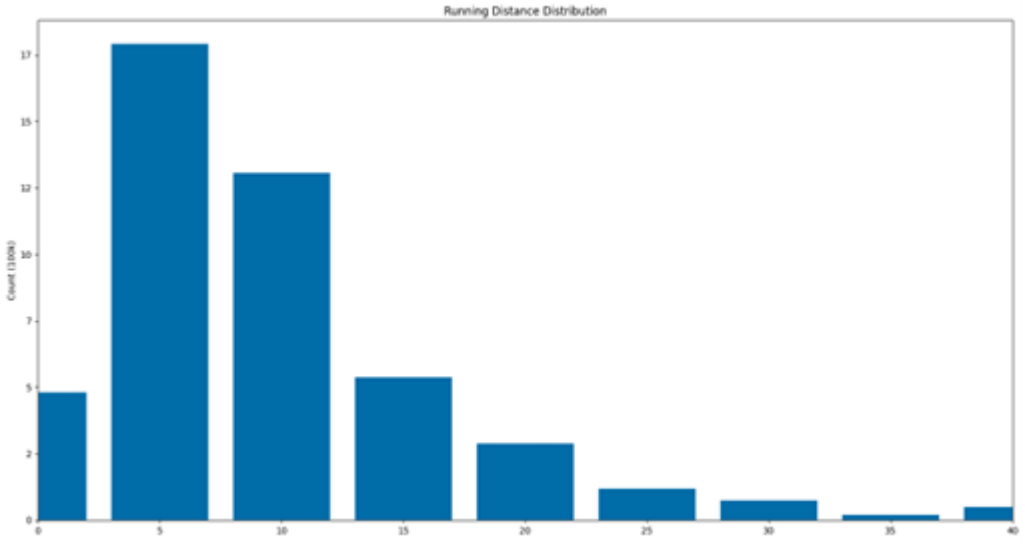

Fig. 3. Running Distance Distribution

3.1. Definition of anomaly

In the context of this study, an anomaly refers to a running activity that deviates significantly from the expected or normal pattern of behavior. The expected pattern of behavior is determined based on the characteristics of the dataset, including the distribution of running distances, durations, and paces.

We conducted a data analysis to determine the expected pattern of behavior for our

dataset. Based on this analysis, we define an anomalous running pattern as one of the following:

1. Top performers with the smallest pace (99p): In our dataset, some athletes may be top performers who exhibit a smaller pace than the average athlete, thus their running distances may be different from the distances of average performers. These athletes may exhibit different running patterns, which can be identified as anomalies.

2. Low performers with an unusually large pace: Similarly, some athletes in the dataset may be low performers who exhibit an unusually large pace. These athletes may also exhibit different running patterns, which can be detected as anomalies.

3. Different activity: In addition to false activity labels, the dataset may also include other activities, such as hiking or weight lifting, which can be distinguished from running activities. These activities may be detected as anomalies in the dataset.

4. Research

The main idea of the research was to compare different types of clustering algorithms. Precision, recall and F1 score metrics were used to compare the efficiency of clustering algorithms. Partition-based, Density-based and hierarchical clustering algorithms were picked for comparison.

Table 1. Performance Comparison of Clustering Algorithms

Algorithm	Precision	Recall	F1 Score
K-Means [3]	0.98	0.053	0.1
DBSCAN [4]	0.92	0.99	0.95
HDBSCAN [5]	0.88	0.05	0.1
Optics [6]	0.95	0.66	0.78
Local Outlier Factor [7]	0.84	0.09	0.16
Mean Shift [8]	0.96	0.05	0.09

4.1. Error minimisation techniques

In this exploration, we concentrated on conducting anomaly detection in cluster-based operations by employing robust ways to minimize errors and enhance the overall performance of our clustering algorithms. Two essential ways were applied to achieve these are feature selection and parameter tuning.

Feature selection aimed to identify a subset of applicable features, $F' \subseteq F$, where F denotes the original set of features, to capture the beginning structure of the data and effectively separate between normal and anomalous cases. We employed various ways, similar to correlation analysis and collective information, to quantify the significance of each feature in the environment of anomaly discovery.

Also, we also performed expansive parameter tuning to optimize the hyperparameters for each clustering algorithm under disquisition. Let θ denote the set of hyperparameters for a given algorithm, and let $L(\theta)$ represent the loss function that quantifies the divagation between the algorithm's prognostications and the true markers. Our ideal was to find the optimal hyperparameters θ^* that minimize the loss function:

$$\theta^* = argmin\theta L(\theta) \tag{5}$$

Through an iterative process involving ways similar as grid hunt and Bayesian optimization, we linked the stylish set of hyperparameters that yielded optimal clustering results for each algorithm. By integrating these ways, feature selection and parameter tuning, we achieved more accurate and dependable anomaly discovery performance across clustering algorithms. The methodical approach of incorporating these strategies allowed us to reduce errors and enhance the effectiveness of our chosen algorithms in the environment of cluster-based anomaly detection algorithms.

5. Conclusion

In this study, we proposed a cluster-based unsupervised anomaly detection method for identifying anomalous running patterns in a running activity dataset. Our method was able to detect anomalous running patterns, including those exhibited by top performers, low performers, and other types of physical activities.

We evaluated the performance of several clustering and anomaly detection algorithms, including DBSCAN [4], HDBSCAN [5], OPTICS [6], Mean Shift [8], Local Outlier Factor [7], and K-Means [3]. Our results showed that DBSCAN [4] exhibited the best performance in terms of the quality of anomaly detection. DBSCAN [4] was also computationally efficient for large datasets like the one used in our research. We found that HDBSCAN [5], OPTICS [6], Mean Shift [8], Local Outlier Factor [7], and K-Means [3] also showed promising results in detecting anomalous running patterns. However, we noted that the Local Outlier Factor [7] was computationally slow on our large dataset. Our study provides valuable insights into the detection of anomalous running patterns using cluster analysis. Our method can be used to identify not only the expected patterns of behavior but also the unexpected ones like incorrectly tracked activities or other types of activities. These insights can be used to improve individual and group running performance and health outcomes.

However, it is important to note that our method cannot detect cheating activity due to the low dimensionality of the dataset and the limited feature selection. One option to detect cheating could be the use of telemetry data from wearable devices in combination with time-series anomaly detection techniques.

Overall, our study demonstrates the potential of unsupervised anomaly detection methods for identifying anomalous running patterns in large and complex datasets. We recommend further investigation into the use of these methods for analyzing physical activity data in other contexts, and the development of more advanced techniques for detecting cheating in running activity data.

1. Andrae, A.S., 2019. Prediction studies of electricity use of global computing in 2030. International Journal of Science and Engineering Investigations, 8(86), pp.27-33.
2. Wang, D., Liao, Q.V., Zhang, Y., Khurana, U., Samulowitz, H., Park, S., Muller, M. and Amini, L., 2021. How much automation does a data scientist want?. arXiv preprint arXiv:2101.03970.
3. Hartigan, J.A. and Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics), 28(1), pp.100-108.
4. Hinneburg, A., 1996. A density based algorithm for discovering clusters in large spatial databases with noise. In KDD Conference, 1996.
5. Campello, R.J., Moulavi, D. and Sander, J., 2013. Density-based clustering based on hierarchical density estimates. In Advances in Knowledge Discovery and Data Mining: 17th

Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17 (pp. 160-172). Springer Berlin Heidelberg.

6. Ankerst, M., Breunig, M.M., Kriegel, H.P. and Sander, J., 1999. OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record, 28(2), pp.49-60.
7. Bhatt, V., Dhakar, M. and Chaurasia, B.K., 2016. Filtered clustering based on local outlier factor in data mining. International Journal of Database Theory and Application, 9(5), pp.275-282.
8. Cheng, Y., 1995. Mean shift, mode seeking, and clustering. IEEE transactions on pattern analysis and machine intelligence, 17(8), pp.790-799.
9. Page, J.T., Liechty, Z.S., Huynh, M.D. and Udall, J.A., 2014. BamBam: genome sequence analysis tools for biologists. BMC Research Notes, 7(1), pp.1-5.