Pakistan Journal of Engineering and Technology
ISSN (Print): 2664-2042; ISSN (Online): 2664-2050
Received: 2024-01-21
Accepted: 2024-02-29
https://doi.org/10.51846/vol6iss4pp42-49

PAKJET
Pakistan Journal of Engineering & Technology

Article

Anomaly Detection Algorithms for Low-Dimensional and High-Dimensional Data: A Critical Study

Mujeeb Ur Rehman [1], Muhammad Waseem [2], Abdul Sattar [3], and Muti Ullah [4]

[1] Department of Software Engineering, University of Management and Technology, Sialkot, Pakistan
[2] Department of Artificial Intelligence, University of Management and Technology, Sialkot, Pakistan
[3] Department of Cyber Security, Khwaja Fareed University of Engineering and IT, Abudhabi Road, Rahim Yar Khan, Pakistan
[4] Department of Computer Science, Khwaja Fareed University of Engineering and IT, Abudhabi Road, Rahim Yar Khan, Pakistan
* Correspondence: Mujeeb Ur Rehman (mujeeb.rehman@skt.umt.edu.pk)

Abstract: Suspicious events or objects that differ from the norm in data can be discovered using anomaly identification. Identifying anomalies is critical for many applicable domains of life, e.g., preventing credit card theft and spotting intrusions into networks. It is possible to detect anomalies on a global scale as well as at the local level. A global outlier is a data point beyond the norm for the entire dataset, while a local outlier may be inside the norm for the entire dataset but outside the surrounding data points. Data outlier identification methods that are performed locally are inadequate. Therefore, better algorithms are required to investigate the high velocity of data and identify local outliers. Machine learning and data mining techniques need to be investigated to determine the pros and cons of anomaly identification residing inside data. The density-based LOF method can be applied as the best choice for identifying local outliers. While many variants of LOF exist for low-dimensional data, none are suitable for high-dimensional data. This research study discusses LOF, COF, and CBLOF methods for spotting local outliers in low and high-dimensional data. Regarding the size of the dimension, the performance of density-based algorithms is examined based on accuracy and time complexity. In this scenario, CBLOF achieves outstanding results due to its distinctive method of employing cluster-based local outlier detection.

Keywords: Anomaly Detection, Density-Based Algorithm, High Dimensional Data

1. Introduction

Identifying rare or unusual events is essential as they are recognized as outliers, and various sectors, including healthcare, credit card fraud detection, and computer network intrusion detection, heavily rely on outlier detection techniques. Many of these methods are designed for static data and face challenges when applied to dynamic data streams, which are integral in numerous applications. Data mining tools specialized in outlier detection are commonly employed to identify anomalies within datasets, resulting in improved data quality. Data mining is dedicated to discovering and interpreting patterns in data.

The use of anomaly detection techniques or data cleansing is critical in data processing since a single anomaly data point can lead to erroneous results. In network traffic patterns, anomalies can hypothetically direct data sent from a conceded or hacked computer. Failure to identify and remove anomalies can delude machine learning algorithms. Data mining is dedicated on the discovery of anomalies within datasets, with samples including the detection of unusual MRI results that may propose health problems or the identification of anomaly sensor readings indicating impending problems.

Local and global anomaly detection are two common methods to recognize outliers. Global anomaly detection evaluates the distance of each data point from all others in the dataset, while local anomaly detection considers a more constrained context. The k-Nearest Neighbors (kNN) technique is repeatedly used to weigh the likelihood of an anomaly within a dataset by examining its vicinity to neighboring data points.

Anomaly detection relies heavily on specialized algorithms and techniques, and this domain has been comprehensively reviewed in various articles and books. Scientists have discussed approaches to identifying anomalies up to 2019, while others discovered data anomaly identification beyond that year.

Traditionally, research readings have successfully established methods for detecting criminal behavior, computer intrusions, credit card fraud, and other problems in data streams. Furthermore, researchers addressed topics like data stream drift, anomaly detection, and anomaly identification, with a primary focus on a method dependent on size of data stream. The body of literature on anomaly identification spans various domains and offers valuable applications [1].

Parametric-based methods assume distribution models like Gaussian to describe data and estimate parameters. Gaussian methods, skilled using Maximum Likelihood Estimates (MLE), are employed to find anomalies in data. Discordancy tests can also detect anomalies. Scientists have established unsupervised anomaly identification algorithms using Gaussian Mixture Models (GMM) with Locality Preserving

Projections (LPP) for more precise anomaly detection. Regression prototypes are another method for anomaly detection, where data points are evaluated against the regression model to identify anomalies.

While dealing with data that doesn't follow a classic distribution and absences earlier information, a non-parametric method, mentioned as distribution-free, is essential. This method involves applying criteria to discover inliers and outliers, with standard tools such as histograms and kernel density estimation (KDE). Histogram exploration is used to create histogram sheets and kernel methods for each attribute in the dataset, merging independent feature probabilities like Naive Bayes. KDE, another non-parametric method, matches local data point densities to neighbors. Reference [2] improved performance and scalability using kernel-based methods, utilizing KDE to estimate sensor data distribution, particularly for detecting cancerous nodes.

The Local Correlation Integral (LOCI) method employs the parameter k to influence its performance, utilizing a maximizing strategy to address the k-value selection problem. It considers all potential k values for each data point, selecting the best score using a growing radius r. LOCI estimates local density using a half-Gaussian distribution, focusing on the number of data points in nearby neighbors rather than distances. However, LOCI is computationally intensive for large datasets due to radius expansion. The Approximate Local Correlation Integral (aLOCI) was introduced for faster processing, utilizing quadtrees and accurate estimates for data points within quad-tree cells. The choice of quadtree depth impacts algorithm performance, but it is necessary to tailor the approximation tree to each data point [3].

The Cluster-Based Local Outlier Factor (CBLOF) employs clustering to detect outliers within a dataset. During model training, it estimates the density of each cluster. The process initiates with a k-means clustering algorithm, which then separates large and small clusters using a heuristic technique. To calculate outlier scores, CBLOF multiplies the distance between a data point and its cluster center by the number of data points in the cluster. Small clusters are evaluated based on their distance from the nearest significant clusters. CBLOF's advantage lies in its cluster-based approach rather than relying on the nearest neighbor method. However, a drawback of this method is its dependence on the k-value parameter in the k-means clustering algorithm [4].

The influenced approach is employed when the data contains clusters of varying densities situated closely, sometimes within cluster boundaries. However, the LOF method struggles to score these data values accurately. The Influenced Outlierness (INFLO) algorithm utilizes the K-Nearest Neighbors (K-NN) technique and incorporates reverse nearest neighbors, which refers to a set of data points located near other data points in its neighbor definition. To calculate the INFLO score, both sets of neighbors are combined. Subsequently, the LOF technique is applied to compute the local reachability score and density [5].

The influenced approach shares similarities with the Local Outlier Probability (LoOP) method, which utilizes nearest neighbors to assess local density. However, LoOP employs a distinct formula to calculate density. While the Local Outlier Factor (LOF) utilizes an outlier score to detect outlier data points, LoOP employs the probability of an outlier to make identifications. The distances between the dataset and its nearest neighbors conform to a distribution curve. Because these distances are positive, LoOP assumes a "half-Gaussian" distribution technique and employs it to establish the density distribution based on probability set distances. Data points are compared to their nearest neighbors to identify potential outliers, generating a local outlier identification score. This score is subsequently transformed into a probability through normalization and Gaussian error function. An advantage of local outlier probability (LoOP) is its dependence on probability scores, making it more honest to implement. Nevertheless, LoOP has shortcomings, including lengthier execution times and the likely of inaccuracies due to the probabilistic nature of data points [6].

The CBLOF technique exclusively relies on the count of cluster numbers and does not consider its cluster density. This restriction is rectified by the Cluster-Based Local Outlier Factor (CBLOF) algorithm, which incorporates an evaluation of cluster densities, assuming that cluster members reveal a spherical distribution. To exhibit this, CBLOF initially employs the k-means procedure to cluster the dataset and consequently categorizes the clusters into large and small clusters using the CBLOF technique.

The unpredictable growth of data generated by websites, social media stages like Facebook and Twitter, and other causes has made data mining increasingly challenging. Data discovery involves extracting knowledge from vast datasets stored on mainframe hard drives. To extract knowledge from such extensive datasets, decision-making models, often employing machine learning, are employed in data mining.

In [7], the authors looked into how deep learning methods can be used to find breaches in computer security. There was a proportional study and a review of the datasets used as part of the investigation, with a focus on deep learning-based intrusion detection methods.

The authors in [8] investigated a spatiotemporal AD-C-L method and gave a full study and analysis. They also talked about some of the problems that come with AD-C-L. The writers also categorized and compiled existing AD-C-L strategies, using standard performance metrics to evaluate each one's unique pros and cons. They put together and presented the results of their comparative analysis, along with any open research questions and the next steps they plan to take in their study.

Reference [9] presented the assessment of ten Machine Learning (ML) algorithms for manufacturing anomaly detection based on various Deep Neural Networks (DNN). These evaluations use versatile algorithms. To determine anomalies in complete system efficiency and task delivery dates, multiple data discovery techniques are applied, including traditional ML and Deep Neural Network (DNN) exemplifications.

The researchers steered a comparative analysis between two machine learning tools, WEKA and Rapid Miner, in the context of network intrusion detection [10]. They utilized Random Tree and Random Forest methods for data mining-based intrusion detection, with the KDD's 99 attack dataset serving as the basis for their comparative analysis. The results revealed that WEKA excels in terms of tools, while Random Forest stands out for its effective methods.

In their paper, the authors [11] introduced a method in which malicious users can be classified as either "awakened" or "asleep." This marked the first instance of categorizing a new type of malicious user (MU) known as the "lazy malicious user" (LMU). Whether awake or asleep, LMU randomly reported aberrant sensing data similar to an "always yes malicious user" (AYMU) or an "always no malicious user" (ANMU) in terms of reporting PU activity appropriately.

In their study, the authors [12] focused solely on conventional anomaly detection settings without considering streaming data. The significance of analyzing the performance of widely used anomaly detection methods on streaming data lies in the inherent time-series properties such as trends, seasonality, and change points.

In their paper, the authors [13] aimed to uncover the fundamental principles and assumptions that underlie various approaches in the field of anomaly detection. The author elaborated on how both classic shallow and emerging deep techniques can mutually influence and expand upon each other, with a focus on the former. Recent explainability techniques were utilized to provide an empirical review of main existing methodologies and presented practical cases along with recommendations. The authors also addressed unresolved issues and proposed new directions for anomaly detection research.

The authors [14] conducted an examination keeping in view machine learning in an IoT environment to detect anomalies. They assessed some of the most effective methods for real-time anomaly detection, offering valuable guidance to practitioners facing uncertainty about the strategies to employ in specific situations.

In [15], the authors presented a significant growth in ML research over the past decade, with the introduction of several new algorithms that have been successfully implemented in various industries. The success of machine learning algorithms heavily relies on a wide range of inputs, including parameter adjustments and data cleaning, which often require substantial manual effort.

In [16], the authors looked at anomaly detection and feature extraction techniques with the goal of methodically finding odd events. Their study concentrated on procedures that could recognize anomalous patterns in multivariate data automatically.

Reference [17] presented a new anomaly detection algorithm, which displays several advantages over traditional algorithms. Because it can directly detect unusual characteristics and effectively identify different types of abnormalities, it effectively connects the gap between judging abnormalities and screening features.

The presence of outliers and defects in the sensors deployed within IoT systems can significantly impact the functionality and outcomes of IoT systems. The primary objective of this study [18] is to detect outliers in IoT devices resulting from sensor tampering, with a specific focus on utilizing machine learning methods. A detailed analysis of all approaches is briefed in Table 1.

Table 1. Comparative analysis of the existing proposed research studies.

Ref.	Learning Approach	Data Source	Performance Metric	Research Aim
[17] (2020)	Anomaly detection algorithms	Publicly available dataset	Accuracy, temperature, power consumption	The authors aimed for implementing anomaly detection techniques in edge devices.
[18] 2022)	Local constant algorithmic approaches	Three Publicly available datasets	Accuracy Per Rate (AUC PR)	The authors used three real-world energy usage statistics where context abnormalities were precisely detected under the time fluctuation of three buildings' energy consumption profiles.
[10] 2020	Random Forest using WEKA and Rapid Miner	KDD's 99 attack dataset	Accuracy Score	The authors compared two machine learning tools WEKA and Rapid Miner, for network intrusion detection, using Random Forest method.
[14] 2018	Ten unsupervised anomaly detection methods	CC-based feature grouping (CCFSRFG) dataset	Accuracy and true positive rate (TPR)	The authors aimed to use unsupervised infrequent pattern detection (UIPD) using unsupervised anomaly detection methods.
[9] 2022)	Ten Machine Learning (ML) models	Real production schedules	System efficiency and accuracy	The authors presented the ML models are trained on real production schedules to discover inefficiencies and job delivery date violations.
[4] (2022)	Density-based unsupervised techniques	Publicly available dataset at Kaggle	Accuracy Score	To compare the density-based techniques with previous approaches based on high-performance accuracy and different parameters using low and high dimensional data.

2. Materials and Methods

2.1. Methodology

The low dimensional dataset and high dimensional datasets are used to conduct the research experiments. The used datasets are publicly available for research purposes. We have extracted these datasets and used them in our research study. Both datasets are preprocessed to make them useful for models. The data analysis is applied to examine the outlier patterns. The clean and preprocessed dataset is divided into two parts. One part of each dataset is utilized to train the applied method, and the other part of the dataset is used to evaluate the models. The density based LOF, COF, and CBLOF models are applied to the comparison. Then, the outperformed model is used to detect the outliers from the data. All the applied models are evaluated using different parameters. The accuracy, precision, recall, and f1 score are our compared evaluation parameters, as shown in Fig. 1.

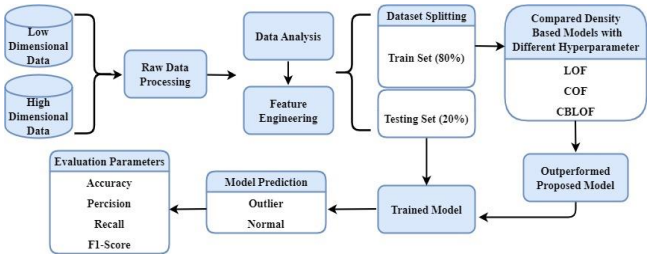

Figure 1. Proposed methodology.

The low-dimensional and high-dimensional datasets at the first step are preprocessed. Then, the data is analyzed, and feature engineering is applied to the dataset to select the best-fit features. Then dataset is portioned into the train (80%) and test (20%) portions. The LOF, COF, and CBLOF models are the models applied to the dataset. Finally, the proposed CBLOF model is trained and tested. The different evaluation parameters are determined by the model.

2.2. Datasets

The two publicly available datasets based on low dimensional and high dimensional features are used for conducting the research experiments. One dataset contains low feature dimensions, and the other dataset contains high feature dimensions. The results of each density-based model are evaluated on both low and high-dimensional datasets in our research study.

The low-dimensional dataset is based on anomaly detection benchmark data named the ADBench dataset. The low-dimensional benchmark dataset is publicly available in the official PyOD API module. The developer of PyOD API module specifically designed the dataset for anomaly detection purposes. We have imported and used this low-dimensional dataset as it is relevant to our research study. The benchmark dataset contains a very small number of features. The total five features act as the training data in the dataset. The last column is the target column, which labels the data as an outlier or normal.

For our research trials, we employ a high-dimensional dataset based on Credit Card Fraud Detection. Credit card transactions were anonymized and classified as fraudulent outliers or authentic transactions. Credit card transactions done by European cardholders in the year September 2013 are added in the dataset. This dataset comprises 492 frauds records out of 284,807 transactions that happened during a two-day period. It only accepts numerical input variables resulting from a PCA transformation. Unfortunately, due to confidentiality concerns, the authors of the dataset are unable to release the original characteristics and additional background data information on the data.

2.3. Raw Data Processing

The raw data was gathered. As a consequence, data cleansing has been performed using a number of approaches, including the removal of duplicates and null values. This approach is used in data mining to convert raw data into an understandable format. Real-world data is sometimes incomplete, inconsistent, or absent. These are some of the pre-processing techniques.

2.4. Outliers Data Analysis

The outlier data analysis for both low-dimensional and high-dimensional datasets is performed. The low dimensional dataset features are visualized in 2-D scatter graph separated by the normal and outlier as target class labels. The high dimensional dataset features are visualized in a 2-D scatter graph separated by the valid transaction and fraudulent outlier as target class labels, as shown in Fig. 2 and Fig. 3.

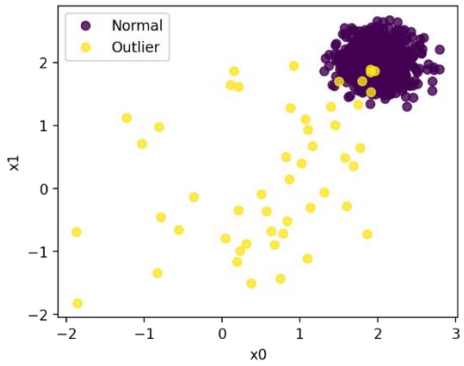

Figure 2. Low dimensional data.

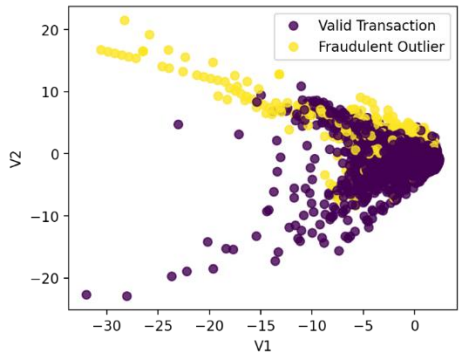

Figure 3. High dimensional data.

2.5. Low and High-Dimensional Dataset Splitting

The high-dimensional and low-dimensional dataset features are split into two portions for applied models' comparisons. One portion of the dataset is reserved for training, and the other portion of the dataset is used to evaluate the models. These two datasets exploited the split ratio of 20:80. 20% of the dataset is used for testing each applied model, and 80% of the dataset is used for training.

2.6. Applied Density Based Models

Regarding density-based outlier detection, regions with lower data point density are measured as likely anomalies since they deviate from the expected pattern in the dataset. This approach is particularly effective in handling datasets with asymmetrical shapes and fluctuating densities, making it appropriate for applications where anomalies may not conform to a specific distance-based benchmark.

The local outlier factor (LOF) is a very popular method that detects anomalies efficiently in a dataset. Its variants are also well-recognized. LOF employs the conception of nearest neighbors to compute the anomaly or outlier score and is an anomaly detection technique that computes the local density deviation values of a data point in relation to its neighbors. Outliers are defined as samples with a much lower density value than their neighbors whereas the LOF of a data point is defined by the ratios between the point's local density and the local densities of its neighbors and considers the relative density of data points. In simple terms, LOF compares a point's local density to the local density of its k-nearest neighbors and returns a score as a result, as shown in Table 2.

Table 2. Different hyperparameters of model.

#	Parameter	Value
1	Algorithm	auto
2	contamination	0.05
3	leaf_size	30
4	Metric	Minkowski
5	metric_params	None
6	n_jobs	1
7	n_neighbors	20
8	Novelty	True
9	P	2

The connectivity-based outlier factor (COF) is an approach for detecting outliers. It is an enhanced version of the LOF (local outlier factor) method. The Connectivity-based Outlier method assigns a degree of outlier to every data point. This degree of outlier is termed the connectivity-based outlier factor, COF of the piece of data. The high COF value of the data point reflects the increased chance of being an outlier. COF finds the connectivity-based value of the outlier factor for observations, becoming the comparison of chaining distances among observations subjected to outlier scoring value and surrounding observations. The COF function is useful for outlier discovery in clustering as well as other multidimensional domains, as shown in Table 3.

Table 3. Different hyperparameters of model.

#	Parameter	Value
1	contamination	0.05
2	Method	fast
3	n_neighbors	None

The Cluster-based Local Outlier Factor (CBLOF) accepts the data set and the cluster model developed by a clustering method as inputs. Using the parameters alpha and beta, the clusters are divided into small and big clusters. The anomaly score is then computed using the size of the cluster to which the location belongs as well as the distance to the next major cluster. Use weighting for the outlier factor depending on cluster sizes, as recommended in the original paper. This is deactivated by default since it may result in unexpected behavior (outliers close to tiny clusters are not discovered). Outliers' ratings are purely determined by their proximity to the next major cluster center, as shown in Table 4.

Table 4. Different hyperparameters of model.

#	Parameter	Value
1	Alpha	0.9
2	Beta	5
3	check_estimator	False
4	clustering_estimator	
5	contamination	0.05
6	n_clusters	8
7	n_jobs	None
8	random_state	None
9	use_weights	False

2.7. Outlier Detection Evaluations

As outlier detection identifies contradictory data points, the distribution model governs outliers and inliers. There are three types of outlier identification strategies: supervised, semi-supervised, and unsupervised. Statistical outlier detection research gives two approaches to dealing with outlier points in a dataset. Firstly, it investigates the outliers, and then, the data model should manage outliers accurately. The performance evaluation parameters of each applied model is analyzed to determine the efficiency of selected models.

The accuracy of the ML model indicates how many times it was accurate overall.

$$Accuracy = \frac{number\ of\ correct\ predictions}{total\ predictions} \quad (1)$$

Precision measures how well a model predicts a specific category.

$$Percision = \frac{True\ Positive\ value}{True\ Positive\ value\ + False\ Positive\ value} \quad (2)$$

Recall indicates how many times the model detected a specific category.

$$Recall = \frac{True\ Positive\ value}{True\ Positive\ value\ + False\ Negative\ value} \quad (3)$$

The F1 score is calculated using the harmonic mean of accuracy and recall.

$$F1\ score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

3. Results and Discussion

Results of the implementation and evaluation of the LOF, COF, and CBLOF models are elaborated in this section. A low-dimensional and high-dimensional dataset will be used to test our model's performance in the first phase. The findings of the applied models are shown in this part, along with the implementation of the models. A GPU-based system with Jupyter Colab as a compiler and 3.2 GHz processors was employed in the experimental configuration, which is the minimum simulation requirement for the experimental setup. As a first step, we looked at how well our models detected outlier anomaly datasets based on accuracy, precision, recall, and F1.

3.1. LOF, COF, CBLOF-based Detection Results

This section contains the experimental specifics of simulation settings utilized in the categorization model that we created. In this research, we have used Python programming language software for simulation and building density-based models. The anomaly detection results of applied LOF, COF, and CBLOF techniques are analyzed in this section. The results are based on low-dimensional data and high-dimensional datasets. The different parameter scores are evaluated for analysis.

The results for low-dimensional datasets based on different accuracy parameters are analyzed. The model achieved 92%, 90%, and 87% accuracy scores on a low dimensional dataset for LOF, COF, and CBLOF, respectively, as shown in Fig. 4, Fig. 5 and Fig. 6.

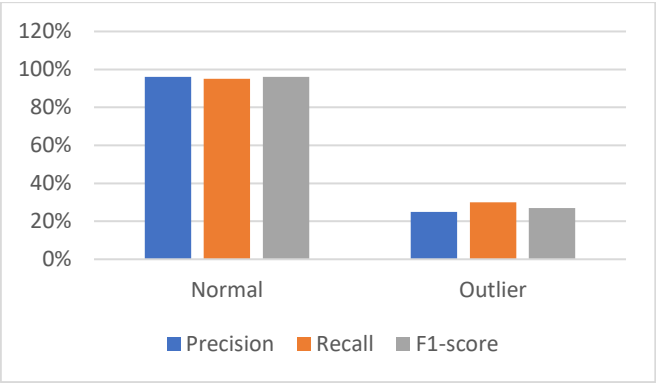

Figure 4. Model test results for LOF (Low Dimensions).

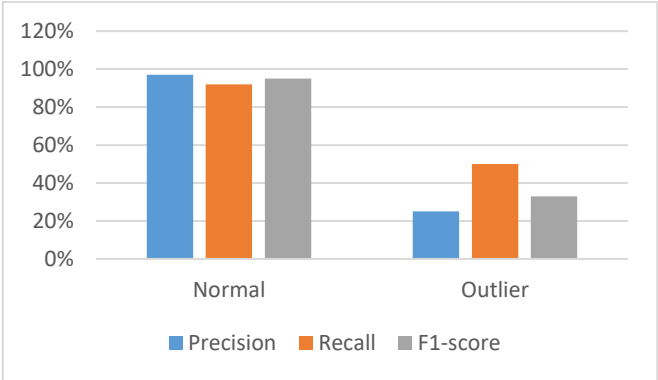

Figure 5. Model test results for COF (Low Dimensions).

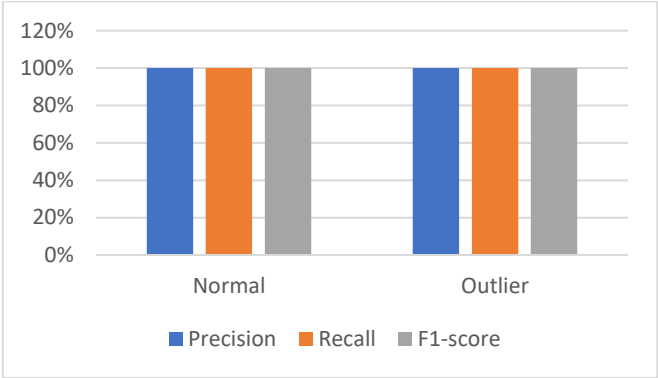

Figure 6. Model test results for CBLOF (Low Dimensions).

The results for a high-dimensional dataset based on different accuracy parameters are analyzed. The model achieved 89%, 87%, and 94% accuracy scores on high dimensional datasets for LOF, COF, and CBLOF, respectively, as shown in Fig. 7, Fig. 8 and Fig. 9.

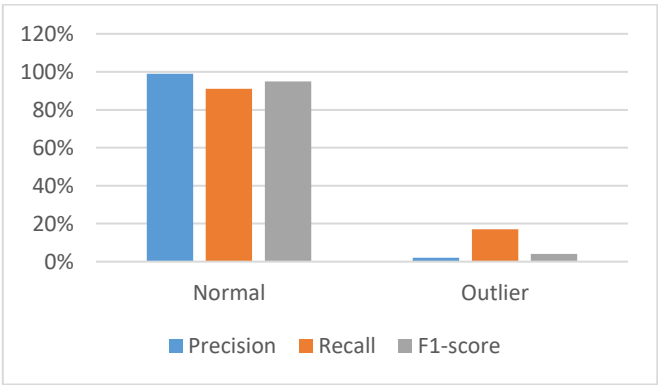

Figure 7. Model test results for LOF (High Dimensions).

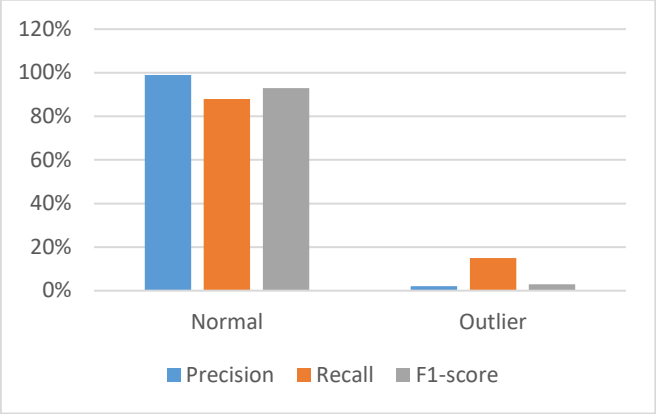

Figure 8. Model test results for COF (High Dimensions).

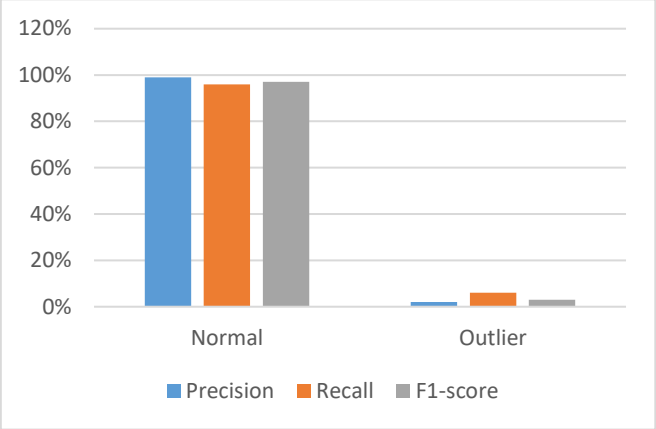

Figure 9. Model test results for CBLOF (High Dimensions).

3.2. Applied Density-based Methods for Comparative Analysis

We have applied three density-based models for anomaly detection in our study and compared the results. The results show that the CBLOF model is our best model due to its high-performance accuracy for low- and high-dimensional datasets in comparison, as shown in Table 5.

Table 5. Outlier and inliers analysis.

Applied Method	Low Dimensional Data		High Dimensional Data	
	Inliers	Outlier	Inliers	Outlier
LOF	188	12	3675	374
COF	180	20	3568	481
CBLOF	190	10	3867	182

This analysis clearly illustrates that the CBLOF model consistently delivered impressive results for both high and low-dimensional datasets, as evidenced in Fig. 10. The CBLOF model distinguishes other models as it uses a cluster-based local outlier factor approach to calculate outlier scores. This approach determines the anomaly score in the CBLOF model by measuring the distance of each data point from its cluster center.

Furthermore, a comprehensive examination of the runtime computations for the applied density-based models is detailed in the table below. The results in Table 6 unequivocally demonstrate that the CBLOF model exhibited significantly shorter runtime computations for high-dimensional datasets, whereas its performance is poor for low-dimensional datasets.

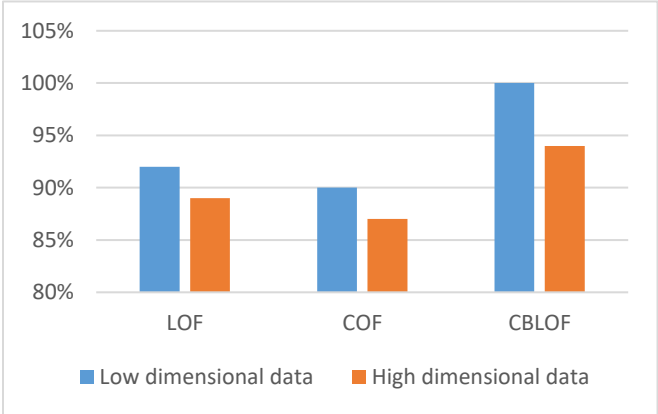

Figure 10. Applied models results for different dimensions.

Table 6. Runtime computation analysis.

Applied Method	Runtime Computation Time (seconds)	
	Low Dimensional Data	High Dimensional Data
LOF	0.0537	12.6037
COF	0.8505	241.954
CBLOF	3.7707	2.1339

The analysis of runtime computations reveals that our proposed CBLOF model outperforms the competition in terms of processing time when dealing with high-dimensional data. In contrast, models relying on nearest neighbors exhibit slower performance when it comes to calculating outliers. The outperforming CBLOF technique extricates itself by extracting outlier scores through a cluster-based local outlier factor methodology. This approach computes the outlier score in the CBLOF model based on the distance of each data point from its relevant cluster center. On the other hand, other models like COF and LOF exclusively calculate the local density deviation value of a data point with respect to its nearest neighbors. This major difference in methodology is the main reason behind the CBLOF model's higher performance, as it leverages a cluster-based local outlier factor that results in high accuracy. Another contributing factor is that the CBLOF method doesn't require the computation of complicated density estimations, further enhancing its effectiveness and efficiency.

4. Conclusion

Finding anomalies, or events or data points that don't happen very often in a dataset, is very important for finding things like credit card fraud and network breaches. In the field of data mining, there are two main ways to find anomalies, namely local methods and global techniques. Local outliers are different from their nearby friends right now, and global outliers are way outside the normal range of data for the whole set. Most of the time, the focus is on finding local outliers. The Local Outlier Factor (LOF) is a popular way to do this by using density-based values. Still, density-based methods have

problems when they are used on big datasets with a lot of dimensions, which means they need more advanced algorithms and methods. This research study looks into density-based methods for finding outliers in all kinds of datasets, whether they are low-dimensional or high-dimensional. By testing different methods on datasets with different sizes, it was found that the CBLOF algorithm worked better than the others, especially when it came to accuracy and processing time. Still, these density-based models didn't work as well on datasets with a lot of variables, which shows that we need more advanced models that can deal with these kinds of situations.

References

[1] D. Chakraborty, V. Narayanan, and A. Ghosh, "Integration of deep feature extraction and ensemble learning for outlier detection," Pattern Recognition, vol. 89, pp. 161–171, May 2019, doi: https://doi.org/10.1016/j.patcog.2019.01.002.

[2] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier Detection for Temporal Data: A Survey," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 9, pp. 2250–2267, Sep. 2014, doi: https://doi.org/10.1109/tkde.2013.184..

[3] N. Ishaq, T. J. Howard, and N. M. Daniels, "Clustered Hierarchical Anomaly and Outlier Detection Algorithms," 2021 IEEE International Conference on Big Data (Big Data), Dec. 2021, doi: https://doi.org/10.1109/bigdata52589.2021.9671566.

[4] K. Miyazaki and K. Tanaka, "Outlier Removal for Improving the Accuracy of Electric Vehicle Behavior Prediction," Jun. 2020, doi: https://doi.org/10.1109/melecon48756.2020.9140457.

[5] G. Geetha and K. M. Prasad, "An Hybrid Ensemble Machine Learning Approach to Predict Type 2 Diabetes Mellitus," Webology, vol. 18, no. Special Issue 02, pp. 311–331, Apr. 2021, doi: https://doi.org/10.14704/web/v18si02/web18074.

[6] S. Prykhodko, N. Prykhodko, and K. Knyrik, "Estimating the Efforts of Mobile Application Development in the Planning Phase Using Nonlinear Regression Analysis," Applied Computer Systems, vol. 25, no. 2, pp. 172–179, Dec. 2020, doi: https://doi.org/10.2478/acss-2020-0019.

[7] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, vol. 50, p. 102419, Feb. 2020, doi: https://doi.org/10.1016/j.jisa.2019.102419..

[8] A. Gholami and A. K. Srivastava, "Comparative Analysis of ML Techniques for Data-Driven Anomaly Detection, Classification and Localization in Distribution System," Apr. 2021, doi: https://doi.org/10.1109/naps50074.2021.9449712.

[9] A. Kharitonov, A. Nahhas, M. Pohl, and K. Turowski, "Comparative analysis of machine learning models for anomaly detection in manufacturing," Procedia Computer Science, vol. 200, pp. 1288–1297, Jan. 2022, doi: https://doi.org/10.1016/j.procs.2022.01.330.

[10] N. Fatima, L. Liu, S. Hong, and H. Ahmed, "Prediction of Breast Cancer, Comparative Review of Machine Learning Techniques, and Their Analysis," IEEE Access, vol. 8, pp. 150360–150376, 2020, doi: https://doi.org/10.1109/access.2020.3016715

[11] A. Ahmed, Muhammad Sajjad Khan, N. Gul, I. Uddin, S. Kim, and J. Kim, "A Comparative Analysis of Different Outlier Detection Techniques in Cognitive Radio Networks with Malicious Users," Wireless Communications and Mobile Computing, vol. 2020, pp. 1–18, Dec. 2020, doi: https://doi.org/10.1155/2020/8832191.

[12] M. Munir, M. A. Chattha, A. Dengel, and S. Ahmed, "A Comparative Analysis of Traditional and Deep Learning-Based Anomaly Detection Methods for Streaming Data," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Dec. 2019, doi: https://doi.org/10.1109/icmla.2019.00105.

[13] L. Ruff et al., "A Unifying Review of Deep and Shallow Anomaly Detection," arxiv.org, Sep. 2020, doi: https://doi.org/10.1109/JPROC.2021.3052449

[14] S. Brady, D. Magoni, J. Murphy, H. Assem, and A. O. Portillo-Dominguez, "Analysis of Machine Learning Techniques for Anomaly Detection in the Internet of Things," 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Nov. 2018, doi: https://doi.org/10.1109/la-cci.2018.8625228.

[15] M. Bahri, F. Salutari, A. Putina, and M. Sozio, "AutoML: state of the art with a focus on anomaly detection, challenges, and research directions," International Journal of Data Science and Analytics, Feb. 2022, doi: https://doi.org/10.1007/s41060-022-00309-0

[16] M. Flach et al., "Multivariate anomaly detection for Earth observations: a comparison of algorithms and feature extraction techniques," Earth System Dynamics, vol. 8, no. 3, pp. 677–696, Aug. 2017, doi: https://doi.org/10.5194/esd-8-677-2017.

[17] Z. Wu, X. Yang, X. Wei, P. Yuan, Y. Zhang, and J. Bai, "A self-supervised anomaly detection algorithm with interpretability," *Expert Systems with Applications*, vol. 237, p. 121539, Mar. 2024, doi: https://doi.org/10.1016/j.eswa.2023.121539.

[18] H. Bilakanti, S. Pasam, V. Palakollu, and S. Utukuru, "Anomaly detection in IoT environment using machine learning," *Security and Privacy*, Jan. 2024, doi: https://doi.org/10.1002/spy2.366.