AN INTEGRATED ANOMALY DETECTION FRAMEWORK FOR STREAM DATA USING EXTENDED LOF WITH WINDOW METHOD

AKURI SREE RAMA CHANDRA MURTHY[1], Dr. CH. VENKATA NARAYANA[2]

[1]Research Scholar, Dept. of CSE at JNTUK, Kakinada, A.P, India.

[2]Supervisor, Professor, Dept. of CSE at Lakireddy Bali Reddy College of Engineering, Mylavaram, A.P, India.

E-mail: [1]sreeram.ramu2k3@gmail.com, [2]cvnreddy.chejarla@gmail.com

ABSTRACT

Multivariate time series and data streams are closely linked, but the latter typically show a larger time dependence and do not require real-time processing. Numerous fields, including network intrusion detection, financial fraud detection, and defect detection in industrial and infrastructure systems, depend on the ability to identify anomalies in streaming data. The majority of anomaly detection (AD) algorithms now in use work well with static data, when all accessible information is available at the time of detection. However, they are unable to handle dynamic data streams. Our study's EM-W-LOF (Extended Local Outlier Factor) algorithm, which depends on Expectation Maximization and the window model, outperforms traditional techniques and offers an effective way to detect anomalies in data streams. Expectation maximization (EM) is a method applied to process data rectification. To lower the false alarm rate, data windows are incorporated as update units. Several tests are conducted here to distinguish between candidate and actual anomalies. The enhanced EM-W-LOF's false positive rate demonstrates its benefit. Additionally, data points of detected actual anomalies are immediately deleted by the suggested technique. Through practical studies with both synthetic and actual data sets, we examined the enhanced algorithm's performance as well as the sensitivity of specific parameters. The experimental findings show that, in comparison to the standard algorithms and their enhancements, the suggested improved algorithm performed better in terms of both detection rate and false alarm rate

Keywords: *Window Model, Data Streams, Anomaly Detection, Incremental Local Outlier Factor Algorithm*

1. INTRODUCTION

The anomaly or novelty identification issue, which seeks to identify anomalous or unusual incidences, is among the most crucial machine learning tasks [1][2]. The absence of a precise explanation of anomalous cases in the stream makes this challenge a difficult one. A stream may exhibit anomalies in the form of individual data points, data points that deviate from temporal patterns, or even large clusters that abruptly explode and then vanish. Various categories of anomalies are described in Figure 1. Finding anomalous patterns or observations in a dataset that substantially departs from the expected behavior is the goal of AD[5][9]. However, much less research has been done on anomaly identification in multidimensional streaming data. Rapid anomaly identification improves safety and reduces risks. Additionally, AD has been widely applied in domains including fraudulent transactions [12], medical AD [8][13-14], and network intrusion [11][19-20].

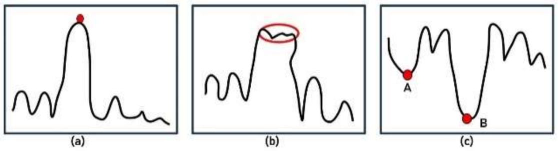

Figure.1.(a) Point Anomaly (b) Collective Anomaly (c) Contextual anomaly

There are 2 kinds of AD techniques: supervised & unsupervised learning. Supervised approaches use pre-labeled training data [4] to learn general features of normal points, and they have two inherent drawbacks [10]: it is hard to find pre-labeled data, and it is hard to identify novel outlier categories. Even typical behaviors might change in a continuously shifting data stream, and anomalies frequently lack a consistent pattern. Consequently, it is possible that the model derived from the pre-labeled training examples will not work. Unsupervised learning techniques, on the other hand, can get over these restrictions since they don't need pre-labeled data and use density, similarity, and distance measurements between samples to identify outliers that deviate significantly from typical data. The following are a few of the frequently used methods for unsupervised outlier detection: cluster-based local outlier factor (LOF), angle-based outlier detection, principal component analysis, feature bagging, k-nearest neighbors, isolation forest, LOF, minimum covariance determinant, histogram-based outlier score, one-class support vector machines, and various ensemble voting methodologies.

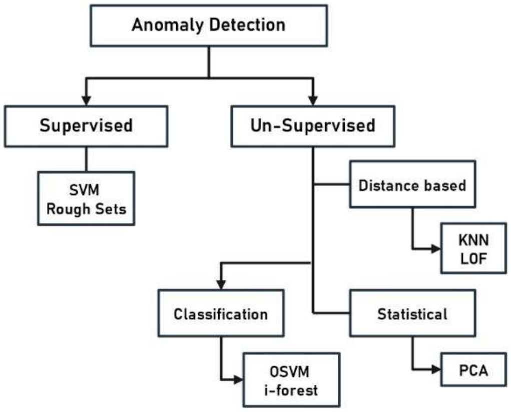

Figure.2. Anomaly Detection Techniques for Stream Data

Among the general strategies for AD is the LOF [3], which has seen a lot of recent uses, such as the detection of system intrusions and credit card fraud. The local density of all the sample points about the points in its immediate vicinity is used by LOF to calculate an anomaly score. The two hyperparameters used by LOF are contamination and neighborhood size. The percentage of the most isolated points that should be forecasted as anomalies depends on the contamination. Even though many data are acquired in data streams, the majority of AD techniques employ static data sets that have records of all data stored prior to finding. In video surveillance, for example, fresh video and picture data are continuously generated over time [7]. With the advent of fresh data in a very short period, data streams are constantly and swiftly changing, and continuous flow could result in an infinite data scale [6]. As a result, static and ineffective approaches are inappropriate for examining anomalies in data streams. Rather, in a dynamic and complex context, AD for data streams requires online and real-time analysis. These specifications and traits make AD methods more challenging and present a research challenge. This paper suggests an enhanced LOF called EW-W-LOF with Expectation Maximization (EM) and window models to improve the performance of the algorithm. By raising the TP (true positive) rate and lowering the FP rate, the suggested approach preserves the great computational efficiency of the incremental algorithm while enhancing the perfecting of anomaly identification in data streams. The enhanced method's innovation consists of:

- By updating multiple data points instead of only one, the window technique lowers the FP rate by assisting in the identification of anomalous and new normal patterns.
- With threshold, identified anomalies from normal data also avoid misinterpreting points near the window edges.
- Automatically identified anomalies to prevent them from clustering with the proposed integrated AD framework EM-W-LOF. According to experimental results, the enhanced algorithm performs better at detecting anomalies in data streams when contrasted to the original approaches.

This is how the remaining article is structured. In Section 2, we reviewed relevant literature, and in Section 3, we presented the LOF method and the EM-W-LOF algorithm. Our enhanced algorithms were thoroughly examined and analyzed in Section 4. In Section 5, we contrasted the outcomes of our algorithm with those of the actual program through experimental investigations utilizing both simulated and real-world data. Section 6 concludes by summarizing the result and future work.

2. RELATED WORK

Finding data instances that vary noticeably from the common data objects is the goal of AD. Various AD methods, including semi-supervised, unsupervised, and supervised approaches, have been proposed depending on the availability of labels. Unsupervised AD procedures frequently imply that anomalies are located in low-density areas and typically assume no access to labeled data[8]. The most popular and straightforward unsupervised global AD technique for point anomalies is k-NN AD. The anomalous score is determined by this distance-based approach using the k-nearest-neighbors distance [21]. In addition to being computationally costly, this method is very reliant on the value of k as well as may not work if normal data points lack sufficient neighbors. [25] introduced LOF, the maximum used unsupervised technique for local density-based AD. Each instance's k-nearest-neighbors set in LOF are established by calculating the distances to every other instance. This algorithm's fundamental premise is that the neighbors of the data instances have been dispersed spherically. However, the spherical density estimate is not suitable in some application cases when normal data points are arranged in a linearly connected fashion. [28] suggested an enhanced version of LOF called COF (Connectivity based Outlier Factor), which enhances the linear structure taken into consideration. One drawback of this algorithm is that it occasionally estimates outlier scores incorrectly when groups including distinct densities are very close to one another. [29] suggests a technique for quick anomaly identification that takes advantage of Gaussian Mixture Models (GMM). Here, a GMM is used to measure the optical flowof moving objects within the video frame's windows. By considering a sample's Mahalanobis distance from each element of the model mixture, an anomaly is identified. In a sliding window, [30] suggested using continuous nearest neighbor queries over sliding windows. The most often used method for identifying outliers in streaming data is this one. With a sliding window, the well-known LOF technique put forward by [31] is implemented as SWLOF (Sliding Window LOF. Although LOF remains among the most dependable choices for general-purpose outlier detection, streaming data cannot be used because of its greater computational cost. The accuracy of the here-implemented SWLOF is equal to that of the incremental LOF method suggested by [32], while it uses alternative indexing strategies to shorten run times. A sliding window was added, and fewer normal points were identified as abnormal points [28]. However, the drawback is that some outdated information is removed, making it difficult to tell new behaviors from old ones. For recently entered points, this could lower the AD accuracy rate. Furthermore, the issue of the points at the window's end—which is analogous to the issue of identifying a newly entering point—is disregarded by the sliding window. The n data points with higher scores are identified as outliers using the Top n kind of technique. Conversely, the statistical method identifies a point as an outlier if its score exceeds the distribution's mean plus α times its standard deviation, where α is a user-specified number. Adaptive threshold setting has been suggested previously, although that instance used time series [27]. The technique divides a time series into unequal-length segments according to the data, based on the Adaptive Piecewise Constance Approximation (APCA) representation of the time series. However, unlike in our situation, this approach works best with univariate data and isn't appropriate for multivariate data. Nevertheless, the LOF algorithm and its enhancement have a comparatively greater false-positive rate since they identify every point as soon as it enters the data set. Additionally, regular patterns in data streams are subject to alteration. The new normal points could be mistakenly identified as outliers if every altered point is identified as soon as it enters the data set. A timeframe was added, and fewer normal points were identified as abnormal points, to address this issue. The drawback, though, is that some older points are eliminated, making it difficult to tell new behaviors from old ones using an auto adaptive threshold. This could lower the AD accuracy rate for recently entered locations []. Furthermore, the issue of the points at the window's end—that is analogous to the issue of identifying a single freshly entering point—is ignored by the window. Additionally, inadequateinsertion of new pattern points results in a high false positive rate. In the study, we suggested an Extended LOF depending on window and EM models to address these issues.

3. BACKGROUND AND METHODS

There are two portions in this section. In the first part, we present the motivation behind to design new framework. In the second part, we present related methods for AD in stream data.

3.1. Motivation

Businesses from a variety of industries, such as manufacturing, healthcare, travel, accommodation, fashion, food, and logistics, are devoting significant resources to gathering large amounts of data and examining any hidden abnormalities to better serve their clientele. The majority of the time, the data that is gathered is in the form of streams, making it complex to accurately identify point anomalies in them. With fresh data arriving in a short period, data streams are continuously and quickly changing, and continual flow may result in an infinite amount of data. Moreover, we are employing an unsupervised approach since it is nearly hard to categorize vast volumes of data in the majority of real-world situations. The traditional distance-based AD techniques are unable to find point anomalies in stream data. Numerous studies increased the algorithm's accuracy or efficiency [22–24] [26], but they continued to overlook issues like the high rate of false positives & the loss of historical data. Rather, in a dynamic and complex context, AD for data streams requires online and real-time analysis. To enhance the performance of the AD in the data stream, this study proposes an

EM-W-LOF with EM and window model. Initially, he streamed data and trained with unsupervised learning methods like LOF, Iforest, SVM, etc. Later the new data point is rapidly generated with an expectation-maximization (EM).

3.2. Methods

Prior to this description of our developed algorithm framework, we demonstrated the GMM for Gaussian densities of data samples $\boldsymbol{Z}$,LOF algorithm for examining local outliers in the static data sets, & Window Method for pruning data samples.

3.2.1. Gaussian Mixture Model

All generated data samples are formed from a mixture of finite Gaussian densities, according to a type of probabilistic model referred to as a GMM. Thus, GMM uses multiple Gaussian densities to simulate a data set's distribution.Generally speaking, a Gaussian density $\boldsymbol{G}$ in a d-dimensional space is described below and is demonstrated in eq(1):

$$G(Z\,|\,\mu,\Sigma)=\frac{1}{(2\pi)^{d/2}(|\sum|)^{1/2}}*exp(-\frac{1}{2}(Z-\mu)^{T}\frac{1}{\sum}(Z-\mu))$$

From eq(1), $\boldsymbol{Z}$ represents a data vector, $\boldsymbol{\mu}$ and $\sum$ is the mean and covariance matrix of G. In GMM, generally eq(1), is usually called a component.With that a k-component GMM defined in eq(2):

$$C_{K}^{G}(Z)=\sum_{k=1}^{K}W_{k}*G(Z\,|\,\mu_{k},\Sigma_{k})$$

From eq(2), $\{W_k,\mu_k,\Sigma_k\}$ are the parameters of GMM and are unknown in general. Represented all these parametersas $\theta=\{W_k,\mu_k,\Sigma_k\}$ and are estimated by the EM (Expectation-Maximization) algorithm.

3.2.2. Elliptic Envelope

An unsupervised machine learning method called the EE algorithm fits a reliable covariance estimate to the data. It determines the inlier position and covariance robustly, independent of outliers, assuming that the inlier data are usually distributed. Using the minimal covariance determinant, the EE method attempts to fit an ellipsoid around the information [49,50]. The Mahalanobis distance is used to estimate the ellipsoid's radii along each axis. A measure of outlier is then derived from the computed Mahalanobis distances. Outliers will be described as examples that deviate from the ellipsoid. To provide the anticipated percentage of outliers to be found, the algorithm needs a contamination parameter. In this current research, we executed the EE technique using the scikit-learn covariance module's Elliptic Envelope function.

3.2.3. Window Method (WM)

Data streams (DS) are collections of data objects that come in a timely manner and are ordered unbounded sequences. A basic window w is made up of a few data points that arrive constantly over a predetermined period of time. $\mathbf{W}=\{z_1,z_2,z_3,....,z_j\}$, $0<i<j$ .The window's length is length, $|W|$, the total number of data points.A landmarkwindow $W_L=\{w_1,w_2,w_3,...,w_k\}$ includes various continuous neighboring common windows, where w_1 is the original basic window. The landmark window's length is dynamic as it grows when new neighbors are added.

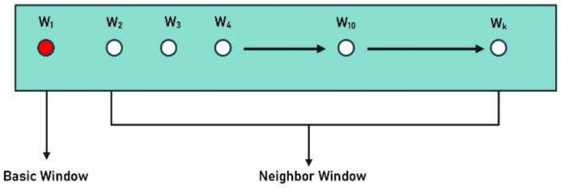

Figure.3. A Window Model for Pruning of Stream Data

3.2.2. Local Outlier Factor (LOF)

An outlier detection approach which is density-based called LOF may recognize outliers in datasets with unequal distributions by determining the local deviation of a specified data point. Depending on the density between all data points as well as their neighboring points, the outlier is determined. The point is more likely to be recognized as an outlier if its density is lower. The following definitions are applied to compute the LOF values for all data point, predicting that the data set is D, that p is a sample of the data, and that the algorithm displays the full concept of LOF.

There are two primary benefits to the LOF algorithm:

i. To examine the LOF of all points, only the local density, not all the data, might be considered.
ii. The distribution of the data sets is not necessary for this strategy.
iii.

Proposed Method: EM-W-LOF Method for Anomaly Detection in Stream Data

The LOF algorithm has certain drawbacks, such as misidentifying typical patterns and overlooking some outliers, despite being an effective technique for identifying anomalies in data streams. To solve

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

these issues, we offered an EM-W-LOF depending on Expectation Maximization (EM) and window model. Since there is no algorithmic parameter to be specified prior to anomaly rectification, the well-known EM methodology [12] is an intriguing statistical inference method. Hence, the proposed method is preferred EM to identify the anomaly in a positive effect. However, the EM algorithm's low solution efficiency was caused by the huge number of parameters that needed to be calculated. Thus, the suggested approach makes use of window data, verifies outliers using a number of tests, and then eliminates actual abnormalities. The EM algorithm is a Maximum likelihood (ML) estimate technique that uses iteration to determine a model's maximum probability. The expectation step and the maximizing step are the two main steps in this algorithm. Until a specific stop condition is met or a predetermined number of iterations are finished, the EM algorithm alternates between carrying out an expectation step and a maximization phase in order to attain maximum likelihood.

Algorithm_1: Local Outlier Factor (LOF)

```
Step-1: $Z^t = \{z_1, \dots, z_{i+1}, \dots\dots, z_p\}$ with 'p' attributes at 't'
$N_k(Z_t) \leftarrow kNN(Z_t)$
Step-2:
    $k\text{-dist}(Z_t) \leftarrow$ distance from Z to its $k^{th}$ nearest neighbor, $k \in N$
    $k\text{-dist}(Z_t, q) \leftarrow$ distance from Z to $q$, $q \in N_k(Z_t)$, and
        $k\text{-dist}(Z_t, q) < k\text{-dist}(Z_t)$
Step-3:
    $r\text{-dist}(Z_t, m) \leftarrow \max(k\text{-dist}(m), k\text{-dist}(Z_t, m))$
        where $k\text{-dist}(Z_t, m) = \text{Euclidian distance}(Z_t, m)$
Step-4:
    $LRD(Z_t) \leftarrow \frac{1}{k} \sum_{m \in N_k(Z_t)} \frac{1}{r\text{-dist}(Z_t, m)}$
Step-5:
    $LOF(Z_t) = \frac{1}{k} \sum_{m \in N_k(Z_t)} \frac{LRD(m)}{LRD(Z_t)}$
```

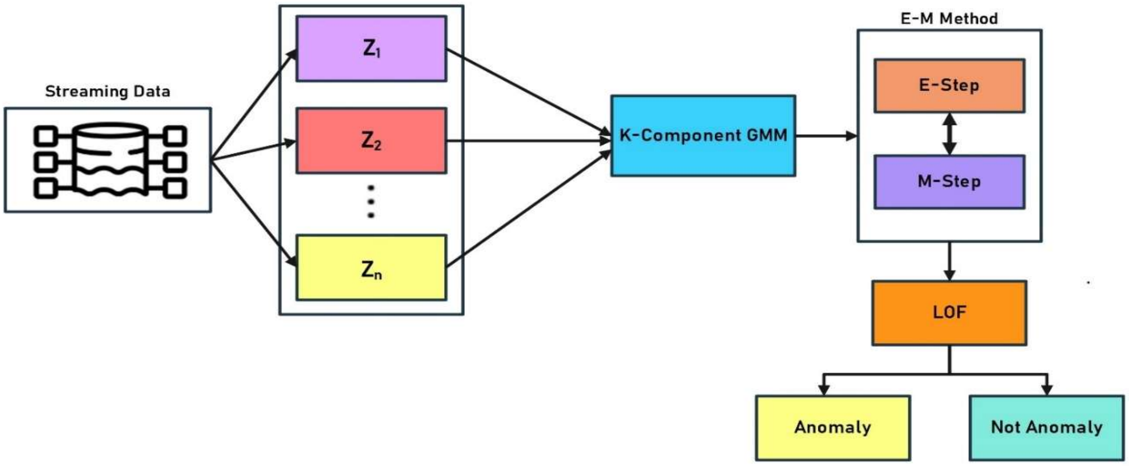

Figure 4. EM-W-LOF For Steam Data Anomaly Detection

For a given K-component GMM with respect to a data set $\mathbf{Z}=\{z_1, z_2, \ldots\ldots, z_n\}$ and applied the EM algorithm to determine the $\theta=\{\mathbf{w}_k, \mu_k, \Sigma_k\}$.

Expectation step: The EM algorithm determines the likelihood that each sample is generated by components. The likelihood $L(C_K^G(Z))$ that the kthcomponent $C_K^G()$ generates sample $\mathbf{z}_i$ can be obtained by (4) :

$$L(C_K^G(Z)) = \frac{W_k * G(Z \mid \mu_k, \Sigma_k)}{\sum_{k=1}^{K} W_k * G(Z \mid \mu_k, \Sigma_k)} \quad (4)$$

Maximum-likehood step: The EM algorithm uses the likelihoods determined in the expectation step as shown in (5)–(7) to update the mean, covariance, and mixing weights of each component:

$$W_k = \frac{\sum_{i=1}^{n} L(C_K^G(z_i))}{n} \quad (5)$$

$$\mu_k = \frac{\sum_{i=1}^{n} L(C_K^G(z_i)) * z_i}{n * W_k} \quad (6)$$

$$\Sigma_k = \frac{\sum_{i=1}^{n} L(C_K^G(z_i)) * (z_i - \mu_k) * (z_i - \mu_k)^T}{n * W_k}$$

(7)

Applied the above two steps to the LOF algorithm to estimate the gaussian densities of data samples with GMM. Later applied the Landmark Window strategy to the Extended LOF with GMM and EM Steps.The benefit of using the embedding window concept with the EM-LOF is that it will exclude actual aberrant points and confirm outliers using numerous tests and an adaptive threshold. Let the data stream $\mathbf{D_S} = \mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_{i+1}, \ldots\ldots, \mathbf{z}_p\}$ is defined with $\mathbf{p}$ attributes and its Gaussian densities generated with S. The data points in the landmark window are represented as $\mathbf{W_L} = \{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ The data points in are grouped in represented as $\mathbf{W_1} = \{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$, $\mathbf{W_2} = \{\mathbf{w}_{m+1}, \ldots, \mathbf{w}_n\}$, etc. For every data point $\mathbf{W}$ performed KNN search to determine the best neighbors. The LOF value for each is calculated and the highest LOF is selected as the better neighbor. Based on threshold values classify normal and anomaly in a better way. The complete idea of the suggested AD algorithm for stream data is shown in Figure 4.

5. Experimental Results and Discussion

In this portion, the suggested procedure EM-W-LOF is performed on real datasets shown in Section 5.1. We examined the effects of varying window sizes and test numbers on our refined algorithm and contrasted the outcomes with those of ILOF, EM-LOF, and WLOF. Python 3.6 is used to implement every program. The experimental outcomes of the actual data are displayed in Section 5.2, accordingly.

5.1 Experimental Datasets

Adopted six datasets have been taken from the KDD CUP and UCI Machine-Learning Repository, which are the glass, wdbc, Forest Cover, http, SA, and SF [15-18] given in Table.1.

Table 1: The Real Data Stream Data For AD

Dataset	Repository	Samples	Dimension	Anomalies
HTTP	KDD Cup	96554	27	1.2%
SA	KDD Cup	976158	41	1.0%
SF	KDD Cup	699691	4	0.3%
Forest Cover	UCI	286048	10	0.96%
glass	UCI	214	9	4.2%
wdbc	UCI	378	30	5.6%

5.2. Evaluation Metrics

TP, FP, TN, FN, TPR, and FPR are the evaluation indicators that were used in this investigation. The number of TP instances, or positive cases that were accurately predicted, is known as TP. It describes the percentages of positive cases that the algorithm classifies as such. The false positive case numbers, or positive cases that the algorithm predicts to be negative cases, are known as FP. In the same way, TN is a true negative case & FN is a false negative case. Anomalies require extra attention because the major objective of AD is to find the anomalies. Anomalies are therefore typically regarded as positive samples. The ratio of correctly identified (TP) outliers to total actual anomalies is known as the AD rate. The number of actual AD points is indicated by this rate. Consequently, improved performance is indicated by a higher AD rate. The ratio of FP cases to total points that the algorithm deems anomalous is known as the false alarm rate. A lower false alarm rate is preferable since it shows the proportion of all detected anomalies that are incorrectly identified. Equations (8) and (9), respectively, define the two metrics.

$$\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP+FT}} \quad (8)$$

$$\mathrm{FPR}=\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}} \quad (9)$$

The role of a classification model on the positive class is summarized by a figure known as a receiver operating characteristic curve or ROC curve. The x-axis displays the False Positive Rate, and the y-axis describes the TP Rate, which is represented by the symbol i. Furthermore, the capacity of the model to differentiate between typical & unusual occurrences across various threshold settings is assessed by applying the area under the receiver operating characteristic curve (AUC-ROC).Better discrimination performance is shown by higher AUC-ROC values. Researchers may evaluate the efficiency of the supervised AD models by thoroughly evaluating these metrics, which will help them choose the best algorithms, such as LOF, SVM, GMM, etc.

Algorithm-2:The Extended LOF(EM-W-LOF) for Stream Data AD

```
Input:
  $\text{Stream Data}=Z_t=\{z_1,\ldots,z_{i+1},\ldots\ldots,z_p\}$ with 'p' attributes at 't'
  $W=\text{Window Size (i.e.,} \sqrt{p}),\ \delta = \text{threshold}, k = \text{no.of nearest neighbors}$
Output:
  $\mathrm{LOF}(Z_t)$
Step-1: Intialization
  $A \leftarrow Z^t; S \leftarrow \{\psi_1,\psi_2,\ldots.,\psi_p\}; M \leftarrow \text{median}(s)$
Step-2:
  while new $N_k$ found do
      for every $Z_i^t$ do
          Find $\psi_i \leftarrow G(N_{ki})$ # Guassian Density of $N_{ki}$ i.e.,$G(N_{ki})$
      if $\psi_i < M$ then
          Split $Z^t$ in $W_1,W_2,\ldots\ldots,W_k$
            $N_{ki}$¬find_N($Z_i^t$,$W_i$)
            $W_i \leftarrow W_i+1$
else
          $N_{ki} \leftarrow$ find_N($Z_i^t$)
            $N_{kz} \leftarrow N_{ki}$
$\mathrm{LRD}(Z^t,N_k) \leftarrow \frac{1}{k}\sum_{m\in N_k(Z^t)} \frac{1}{\text{r-dist}(Z^t,m)}$
$\mathrm{LOF}(Z^t,N_k) \leftarrow \frac{1}{k}\sum_{m\in N_k(Z^t)} \frac{\mathrm{LRD}(m)}{\mathrm{LRD}(Z^t)}$
```

```
for every $Z_i^t \in Z^t$ do
        if $LOF_i > \delta$ then
                $A \leftarrow Z_i^t$
return LOF
```

The Results and Discussion

Instead of identifying the points including high LOF ultimately, the data has been computed, AD for data streams needs to compare the LOF of freshly entered data points with the threshold in real time to output anomalies. We established thresholds for the algorithms EM-LOF, W-LOF, and EM-W-LOF to accomplish real-time detection; these thresholds were separated into fixed thresholds for data studies. Figure 5 displays the performance outcomes of various algorithms on each data set. Figure 5 shows AUC-ROC curves of all algorithms on distinct data sets when fixed thresholds had been set for I-LOF, EM-LOF & EM-W-LOF. As for I-LOF, EM-LOF, and EM-W-LOF, except for the glass dataset, the outcomes of the other five data sets are good, including a high AD rate at a mean of 0.97. However, EM-W-LOF performs the best on every data set when compared to other algorithms that use thresholds and the window idea. The enhanced method has significantly lower false positive rates than other algorithms, although having somewhat lower AD rates on specific data sets. Table 2 gives a TPR summary of the mean outcomes of each method across all data sets. Both false positive rates are decreased by the window idea as compared to the threshold. Furthermore, the suggested technique EM-W-LOF outperforms both the original LOF algorithm and its enhanced algorithms in AD for data streams.Table.2 shows the TPR of LOF, Improved LOF (ILOF), One-Class SVM(OCSVM), SGD-OCSVM (S-OCSVM), GMM, EM-LOF, EM-W-LOF on six datasets namely HTTP, SA, SF, Forest Cover, glass, and wdbc. LOF and One-Class SVM perform well on non-stream datasets as the streaming data is a concern, its performance decreases. Compared to the OCSVM and standard LOF, the proposed EM-LOF and EM-W-LOF methods produced better performance over the streaming dataset.

Table 2: Results of proposed methods in terms of TPR

Dataset	LOF	ILOF	OCSVM	S-OCSVM	GMM	EM-LOF	EM-W-LOF
HTTP	0.34	0.94	0.92	0.00	0.04	1.00	1.00
SA	0.41	0.91	0.48	0.79	0.86	0.90	0.81
SF	0.43	0.89	0.90	0.20	0.79	0.97	0.97
Forest Cover	0.66	0.88	0.52	0.99	0.96	0.84	0.90
glass	0.84	0.69	0.44	0.26	0.77	0.68	0.77
wdbc	0.84	0.82	0.46	0.04	0.71	0.93	0.94

The influence of GMM and Window based methods shows significant improvement in the detection rate of proposed methods in terms of AUC-ROC. In the case of four datasets HTTP, SA, SF, and wdbc proposed methods produced an average of 95% accuracy. The highest accuracy value is derived by the HTTP 100% and the lowest value 77% is retained in the dataset respectively with the EM-W-LOF method. Whereas in EM-LOF the highest and lowest values 100% and 68% are lower compared to the EM-W-LOF method.Table 3 displays the variations in EM-W-LOF AD and false positive rates using the HTTP Dataset. Outliers were detected using a range of test counts for window widths of 20, 60, 130, 180, 250, and 320, respectively. Even while the multiple test method's AD rate somewhat declines, the false alarm rate sharply declines. The AD rate typically drops as the threshold value rises when comparing the various thresholds of the enhanced method, especially when the window size is between 250 and 320. A higher threshold value results in a much lower false positive rate as well as a higher detection rate. Table 4 shows the changes in AD and false positive rates of EM-W-LOF with SA Dataset, where outliers had been isolated by varying window sizes respectively. The AD rate is slightly reduced from 0.774 to 0.436 by varying the window size from 20 to 320 with Three threshold values respectively. Moreover, it is observed that window sizes 250 and 320 with higher threshold values produce a better detection rate i.e., 0.804 and 0.819.Table 5 shows the changes in AD and false positive rates of EM-W-LOF with SF Dataset, where outliers were recognized using varying window sizes respectively.

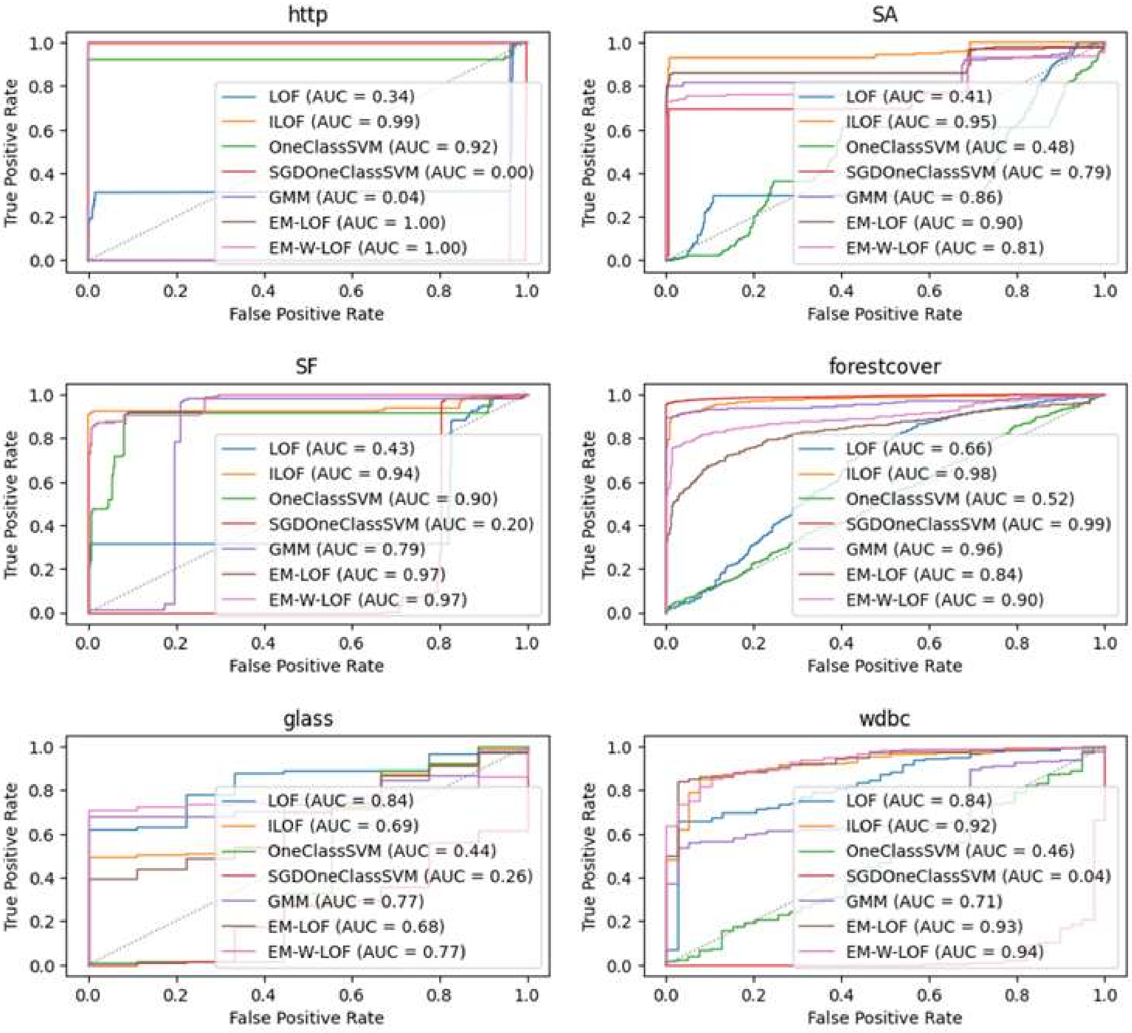

Figure. 5 AUC-ROC Performance of AD Methods over Standard Datasets

Table 3 Experiment results of HTTP Dataset by EM-W-LOF

Parameter	TP	FP	TN	FP	DR	FA
W=20, T=1	966	8690	86899	0.0	0.913	0.908
W=60, T=2	966	13207	82371	0.0	0.921	0.933
W=130, T=3	966	7958	87630	0.0	0.942	0.892
W=180, T=1	966	1598	93990	0.0	0.965	0.643
W=250, T=2	966	1702	93876	0.0	0.974	0.624
W=320, T=3	966	992	94596	0.0	0.993	0.516

Table 4 Experiment results of SA Dataset by EM-W-LOF

Parameter	TP	FP	TN	FN	DR	FA
W=20, T=1	9762	605218	361178	0.0	0.753	0.774
W=60, T=2	9762	771165	195232	0.0	0.760	0.801
W=130, T=3	9762	566172	400225	0.0	0.777	0.767
W=180, T=1	9762	283086	683311	0.0	0.796	0.536
W=250, T=2	9762	117139	849257	0.0	0.804	0.549
W=320, T=3	9762	58569	907827	0.0	0.819	0.436

Table 5 Experiment results of SF Dataset by EM-W-LOF

Parameter	TP	FP	TN	FN	DR	FA
W=20, T=1	6997	433808	258886	0.0	0.895	0.882
W=60, T=2	9762	552756	139938	0.0	0.903	0.915
W=130, T=3	9762	405821	286873	0.0	0.923	0.876
W=180, T=1	9762	202910	489784	0.0	0.946	0.738
W=250, T=2	9762	83962	608731	0.0	0.955	0.672
W=320, T=3	9762	41981	650713	0.0	0.973	0.581

Table 6 Experiment results of ForestCover Dataset by EM-W-LOF

Parameter	TP	FP	TN	FN	DR	FA
W=20, T=1	2861	177350	105838	0.0	0.840	0.834
W=60, T=2	2861	225978	57210	0.0	0.847	0.866
W=130, T=3	2861	165908	117280	0.0	0.867	0.776
W=180, T=1	2861	82954	200234	0.0	0.888	0.590
W=250, T=2	2861	34326	248862	0.0	0.896	0.535
W=320, T=3	2861	17163	266025	0.0	0.914	0.485

Table 7 Experiment results of glass Dataset by EM-W-LOF

Parameter	TP	FP	TN	FN	DR	FA
W=20, T=1	21	122	71	0.0	0.703	0.755
W=60, T=2	21	141	51	0.0	0.709	0.730
W=130, T=3	21	113	79	0.0	0.725	0.699
W=180, T=1	21	60	133	0.0	0.743	0.589
W=250, T=2	21	26	167	0.0	0.738	0.501
W=320, T=3	21	13	180	0.0	0.765	0.498

Table 8 Experiment results of wdbc Dataset by EM-W-LOF

Parameter	TP	FP	TN	FN	DR	FA
W=20, T=1	38	215	125	0.0	0.867	0.876
W=60, T=2	38	249	91	0.0	0.875	0.864
W=130, T=3	38	200	140	0.0	0.895	0.818
W=180, T=1	38	106	234	0.0	0.917	0.708
W=250, T=2	38	45	295	0.0	0.925	0.622
W=320, T=3	38	23	318	0.0	0.943	0.414

Moreover, it is observed that window sizes 25o and 320 with higher threshold values produce a better detection rate i.e., 0.955 and 0.973.Table 6 shows the changes in AD and false positive rates of EM-W-LOF with ForestCover Dataset, where outliers had been recognized by different window sizes respectively. The AD rate is slightly reduced from 0. 834 to 0.485 by varying the window size from 20 to 320 with Three threshold values respectively. Moreover, it is observed that window sizes 25o and 320 with higher threshold values produce a better detection rate i.e., 0.840 and 0.914.Table 7 shows the changes in AD and false positive rates of EM-W-LOF with glass Dataset, where outliers have been identified using varying window sizes respectively. The AD rate is slightly reduced from 0. 755 to 0.498 by varying the window size from 20 to 320 with Three threshold values respectively. Moreover, it is observed that window sizes 25o and 320 with higher threshold values produce a better detection rate i.e., 0.738 and 0.765.Table 8 shows the changes in AD and false positive rates of EM-W-LOF with of wdbc Dataset, where outliers had been recognized using varying window sizes respectively. The AD rate is slightly reduced from 0.876 to 0.414 by varying the window size from 20 to 320 with Three threshold values respectively. Moreover, it is observed that window sizes 25o and 320 with higher threshold values produce a better detection rate i.e., 0.925 and 0.943.

6. CONCLUSION

The EM-W-LOF algorithm gives a moral technique for the AD of data streams. The research uses EM and window models to progress the LOF algorithm & get the best performance. The EM algorithm helps to determine the Gaussian densities for the data samples and helps to distinguish anomaly from normal data. In addition, the windows method is applied to the EM algorithms to classify a better abnormal from the normal points of new patterns and it also decreases the false positive rate. This is added in addition to the threshold along with the multiple tests for anomalies, which split up points into windows and produce large LOF values to distinguish anomalies. The algorithm's overall detection performance is impacted by the two LOF enhancements. The algorithm is better able to identify new outliers when the true anomalies are promptly deleted, and the low false positive rate serves as the basis for the proper deletion of anomalies. Empirical experiments of the proposed algorithm were researched in six real data sets. The results demonstrate that EM-W-LOF enhances the effect of AD for data streams when compared to other methods using thresholds and windows. The proposed EM-W-LOF has notably better performance than the original and its variants of LOF. In all the data sets, the average AD rate of EM-W-LOF is 90% higher than that of LOF, and its variants EM-LOF and ILOF. Furthermore, the average AD rate of EM-W-LOF rises by 4% in comparison to the enhanced algorithms ILOF and EM-LOF. Therefore, in stream data sets, the enhanced EM-W-LOF method with the EM and window model typically performs good on the detection rate. The window size can also be chosen using some optimization techniques, which are being investigated further. Furthermore, when the volume of data increases, retaining all points in the data set aside from abnormalities that have been detected results in a significant computational load and memory needs. To improve AD in stream data, several suitable techniques, including clustering and iterating, must be implemented.

REFERENCES:

[1] Ahmad, S.; Purdy, S. Real-time anomaly detection for streaming analytics. arXiv 2016, arXiv:1607.02480

[2]Zhang, M.; Guo, J.; Li, X.; Jin, R. Data-Driven Anomaly Detection Approach for Time-Series Streaming Data. Sensors 2020, 20, 5646.

[3] Alghushairy, O.; Alsini, R.; Soule, T.; Ma, X. A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. Big Data Cogn. Comput. 2021, 5, 1.

[4] Blázquez-García, A.; Conde, A.; Mori, U.; Lozano, J.A. A Review on Outlier/Anomaly Detection in Time Series Data. ACM Comput. Surv. 2021, 54, 1–33.

[5] Shao, P.; Ye, F.; Liu, Z.; Wang, X.; Lu, M.; Mao, Y. Improving iForest for Hydrological Time Series Anomaly Detection. In Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, New York, NY, USA, 2–4 October 2020; pp. 170–183

[6] Zubaroğlu, A.; Atalay, V. Data stream clustering: A review. *Artif. Intell. Rev.* **2021**, *54*, 1201–1236.

[7] Din, S.U.; Shao, J.; Kumar, J.; Mawuli, C.B.; Mahmud, S.M.H. Data stream classification with novel class detection: A review, comparison and challenges. *Knowl. Inf. Syst.* **2021**, *63*, 2231–2276.

[8] Nassif, A.B.; Talib, M.A.; Nasir, Q.; Dakalbab, F.M. Machine learning for anomaly detection: A systematic review. *IEEE Access* **2021**, *9*, 78658–78700.

[9] Pang, G.; Shen, C.; Cao, L.; Hengel, A.V.D. Deep learning for anomaly detection: A review. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–38.

[10] Blázquez-García, A.; Conde, A.; Mori, U.; Lozano, J.A. A review on outlier/anomaly detection in time series data. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–33.

[11] Cook, A.A.; Mısırlı, G.; Fan, Z. Anomaly detection for IoT time-series data: A survey. *IEEE Internet Things J.* **2019**, *7*, 6481–6494.

[12] Souiden, I.; Omri, M.N.; Brahmi, Z. A survey of outlier detection in high dimensional data streams. *Comput. Sci. Rev.* **2022**, *44*, 100463.

[13] Li, L.; Yan, J.; Wang, H.; Jin, Y. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1177–1191.

[14] Agrahari, S.; Singh, A.K. Concept drift detection in data stream mining: A literature review. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 9523–9540.

[15] IBM, Anomaly detection, 2023, https://www.ibm.com/docs/en/cognosanalytics/10.2.2?topic=analysis-anomaly-detection. (Accessed 20 July 2023).

[16] Galaxyh, KDD cup 1999 data, 2023, https://www.kaggle.com/datasets/galaxyh/ kdd-cup-1999-data. (Accessed 20 July 2023)

[17] Yilmaz, S.F.; Kozat, S.S. PySAD: A Streaming Anomaly Detection Framework in Python. arXiv 2020, arXiv:2009.02572.

[18] Zhao, Y.; Nasrullah, Z.; Li, Z. PyOD: A Python Toolbox for Scalable Outlier Detection. J. Mach. Learn. Res. 2019, 20, 1–7.

[19] Calikus, E.; Nowaczyk, S.; Sant'Anna, A.; Dikmen, O. No free lunch but a cheaper supper: A general framework for streaming

anomaly detection. Expert Syst. Appl. 2020, 155, 113453.

[20] Alghushairy, O.; Alsini, R.; Soule, T.; Ma, X. A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. Big Data Cogn. Comput. 2021, 5, 1.

[21] Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. ACM Comput. Surv. (CSUR) 2009, 41, 1–58.

[22] Kamran, S.; Haas, O. A multilevel traffic incidents detection approach: Identifying traffic patterns and vehicle behaviours using real-time gps data. In Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; pp. 912–917.

[23] Zhang, M.; Li, T.; Yu, Y.; Li, Y.; Hui, P.; Zheng, Y. Urban Anomaly Analytics: Description, Detection and Prediction. *IEEE Trans. Big Data* **2020**, *8*, 809–826.

[24] Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 413–422.

[25] Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; pp. 93–104.

[26] Alghushairy, O.; Alsini, R.; Soule, T.; Ma, X. A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. *Big Data Cogn. Comput.* **2021**, *5*, 1.

[27] G.N. Basavaraj, K. Lavanya, Y Sowmya Reddy, B. Srinivasa Rao, Reliability-driven time series data analysis in multiple-level deep Learning methods utilizing soft computing methods,Measurement: Sensors, Volume 24,2022,100501,ISSN 2665-9174, https://doi.org/10.1016/j.measen.2022.100501.

[28] Pokrajac, David &Reljin, Natasa & Pejcic, Nebojsa & Lazarevic, Aleksandar. (2008). Incremental Connectivity-Based Outlier Factor Algorithm. 211-224. 10.14236/ewic/VOCS2008.18.

[29] Bahrololum, Marjan & Khaleghi, Mahmoud. (2008). Anomaly Intrusion Detection System Using Gaussian Mixture Model. 1162 - 1167. 10.1109/ICCIT.2008.17.

[30] K. Mouratidis and D. Papadias, "Continuous Nearest Neighbor Queries over Sliding Windows," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 6, pp. 789-803, June 2007, doi: 10.1109/TKDE.2007.190617.

[31] Tang, Xianghong. (2015). The Stream Detection Based on Local Outlier Factor. Journal of Information and Computational Science. 12. 6361-6369. 10.12733/jics20107038.

[32] D. Pokrajac, A. Lazarevic and L. J. Latecki, "Incremental Local Outlier Detection for Data Streams," *2007 IEEE Symposium on Computational Intelligence and Data Mining*, Honolulu, HI, USA, 2007, pp. 504-515, doi: 10.1109/CIDM.2007.368917.