© پیش نویس مجله علمی محاسبات نرم، برای داوری در مجله

یک مدل جدید مبتنی بر پنجره لغزان برای تشخیص ناهنجاری در سری‌های زمانی چند متغیره

الهام عنایتی[1*]، دانشجوی دکتری، رضا مرتضوی[2]، استادیار، عبدالعلی بصیری[3]، دانشیار
جواد قاسمیان[4] ، استادیار، محمود معلم[5]، استادیار

[1] دانشکده ریاضی و علوم کامپیوتر– دانشگاه دامغان ـ دامغان ـ ایران ـ e.enayati@std.du.ac.ir
[2] دانشکده مهندسی ـ دانشگاه دامغان ـ دامغان ـ ایران ـ mortazavi@du.ac.ir
[3] دانشکده ریاضی و علوم کامپیوتر –دانشگاه دامغان ـ دامغان ـ ایران ـbasiri@du.ac.ir
[4] دانشکده ریاضی و علوم کامپیوتر – دانشگاه دامغان ـ دامغان ـ ایران ـghasemian@du.ac.ir
[5] دانشکده ریاضی و علوم کامپیوتر – دانشگاه دامغان ـ دامغان ـ ایران ـmoallem@du.ac.ir

چکیده: شناسایی ناهنجاری در سری‌های زمانی چند متغیره یکی از موضوع‌های فعال پژوهشی است که در حوزه‌های مختلفی کاربرد دارد. در حوزه تشخیص ناهنجاری، استفاده از متدهای مبتنی بر پنجره بسیار مرسوم است. این متدها تنها قابلیت تشخیص پنجره ناهنجار را دارند و قابلیت شناسایی نقطه ناهنجاری را ندارند حتی اگر تمام نقاط آن پنجره ناهنجار نباشند. این یک محدودیت اساسی در حوزه متدهای مبتنی بر پنجره است. برای حل این مشکل، ما یک مدل غیرنظارتی مبتنی بر پنجره لغزان برای شناسایی ناهنجاری‌های تجمعی در سری‌های زمانی چند متغیره پیشنهاد دادیم. مدل ما، یک مکانیزم پنجره لغزان را بر روی سری زمانی ورودی چندین بار اجرا می‌کند و سپس از یک تابع تجمیع برای تجمیع درجه‌های ناهنجاری تخصیص داده شده به پنجره‌ها استفاده می‌نماید. این مکانیزم، تشخیص زیردنباله‌های با ناهنجاری بیشتر را تسهیل می‌کند به نحوی که امکان تشخیص این زیردنباله‌ها حتی اگر دقیقا در یک پنجره قرار نداشته باشند نیز فراهم می‌شود. برای ارزیابی عملکرد مدل پیشنهادی از چندین مجموعه داده مصنوعی و واقعی نظیر مجموعه‌های SKAB و MSL استفاده شده است. نتایج به دست آمده برتری مدل پیشنهادی را تایید می‌کنند. برای مدل پیشنهادی شاخص درجهF برای مجموعه داده SKAB با مقدار ۰,۹۰۲ و برای مجموعه داده MSL با مقدار ۰,۶۲۰ به دست آمد که این نتایج در مقایسه با مقدار این شاخص برای سایر روش‌ها به میزان دو برابر بهتر است.

واژه‌های کلیدی: تشخیص ناهنجاری، سری زمانی چندمتغیره، پنجره لغزان، تابع تجمیع، الگوریتم خوشه‌بندی

* **نویسنده مسئول،** r_mortazavi@du.ac.ir

SCJ Accepted

© پیش نویس مجله محاسبات نرم، برای داوری در مجله

A Novel Sliding Window-Based Model for Outlier Detection in Multivariate Time Series

Elham Enayati [1], PhD student, Reza Mortazavi [2*], Assistant professor, Abdolali Basiri [3], Associate professor
Javad Ghasemian [4], Assistant professor, Mahmoud Moallem [5], Assistant professor

[1] School of Mathematics and Computer Sciences, Damghan University, Damghan, Iran, e.enayati@std.du.ac.ir
[2] School of Engineering, Damghan University, Damghan, Iran, r_mortazavi@du.ac.ir
[3] School of Mathematics and Computer Sciences, Damghan University, Damghan, Iran, basiri@du.ac.ir
[4] School of Mathematics and Computer Sciences, Damghan University, Damghan, Iran, ghasemian@du.ac.ir
[5] School of Mathematics and Computer Sciences, Damghan University, Damghan, Iran, moallem@du.ac.ir

Abstract: Anomaly detection in multivariate time series has been an active research area due to its widespread application in various fields. Window-based methods are popular in the anomaly detection domain. These methods identify anomalous windows rather than specific anomalous points, even if not all points within the window are anomalies. It is a critical limitation of window-based methods. We propose an unsupervised sliding window-based model for detecting anomalies in multivariate time series to address this limitation. Our model employs a sliding mechanism to iterate through the input time series multiple times and utilizes a consensus function to aggregate different window anomaly scores. This mechanism facilitates the discovery of more anomalous subsequences, even if they are not precisely confined within a specific window. To evaluate the performance of the proposed method, several experiments on synthetic and real-world datasets, including SKAB and MSL, with multiple indices. The results confirm the superiority of the proposed method. The method achieves an $Fscore$ of 0.902 for SKAB and 0.620 for MSL, which are twice as good as the results achieved by other methods.

Keywords: *Anomaly Detection, Multivariate Time Series, Sliding Window, Consensus Function, Clustering Algorithm*

*** Corresponding author, r_mortazavi@du.ac.ir**

.

SCJ Accepted

1. Introduction[1]

Time series has garnered significant attention across various fields due to its generation by many applications [1], [2]. Anomaly detection plays a crucial role in this domain. Time series anomaly detection (TAD) aims to identify unexpected changes within a given time series [3].

A time series is a sequence of data points ordered based on time intervals. The time series which records one observation at each time point is referred to as univariate time series. A multivariate time series records multiple observations at a time. Time series anomalies are data points that deviate from the regular patterns of the series based on specific measures or models. These anomalies can occur at individual time points (point outliers) or subsequences of time points (collective or contextual outliers) [4]. Anomalies may appear in one or multiple channels of a multivariate time series. Multivariate time series anomalies can reveal significant events depending on the domain of application, such as cyber-attacks on water distribution systems [5], traffic control [6], disease outbreak detection [7], earth science [8], etc.

Many outlier detection approaches have been proposed, considering various data characteristics. TAD algorithms commonly employ sliding window-based mechanisms, where the input time series is divided into segments known as sliding windows. Window segmentation plays an important role in two aspects. Firstly, dividing the time series into smaller subsequences is advantageous for handling time-consuming processes more efficiently. Secondly, in online or stream applications where the complete time series may not be available at the time of execution, window-based methods offer the capability to operate on existing buffered windows. This allows for real-time analysis and detection of anomalies as new data becomes available [9].

Window-based methods face several challenges:

(1) The method's effectiveness is strongly influenced by the selected window size [10]. Employing wider sliding windows generally results in higher TAD accuracy [11], but it also leads to an increas in False Positives (FP). On the other hand, smaller sliding windows can reduce FP.

(2) Anomalies are considered anomalous windows, meaning that all data points within an anomalous window are treated as anomalies. This leads to an increase in false positives and a decrease in true negatives (TN).

False positives occur when the method incorrectly identifies normal data points as anomalies. If an anomalous window includes some normal data points alongside the real anomalies, then the method would identify those normal points as anomalies, leading to an increase in false positives. Also, true negatives represent the correct identification of normal data points as non-anomalous. If all data points within an anomalous window are labeled as anomalies, it's likely that true negatives would decrease. This is because any normal data points occurring within an anomalous window would be incorrectly treated as anomalies, causing a decrease in true negatives.

The specific effects on the performance of a TAD method can indeed arise from the disadvantages associated with an increase in FP and a decrease in TN. More formally, the performance of a TAD method is evaluated by several indices such as accuracy, precision, and $Fscore$. From a mathematical perspective, the accuracy value is directly influenced by TN, as indicated by the formulas provided in Section 4.2. Consequently, a decrease in TN results in a corresponding decrease in the accuracy index. Similarly, the precision index inversely correlates with FP, meaning that an increase in FP leads to a reduction in the precision value. Furthermore, a decrease in the precision value contributes to a decline in the $Fscore$.

The motivation of this paper is to introduce a novel anomaly detection method called ANNOTATE, which incorporates the $slid\&cons$ mechanism to enhance the overall performance of window-based TAD methods. The $slid\&cons$ mechanism improves the performance by decreasing FP and increasing TN. The mechanism uses a sliding process and an anomaly score consensus function. Before describing the proposed method in detail, the following motivational examples and lemma illustrate the problems and a potential solution.

Example 1: Figure 1.a depicts one dimension of a multivariate time series called *Synth1*, consisting of 1362 data points. *Synth1* contains a collective anomaly of length 277 located at position 819. We assume the existence of an ideal window-based TAD method, denoted as $iTAD$. $iTAD$ correctly identifies an anomalous window.

The optimal window size for $iTAD$ would be 277, which aligns with the length of the collective anomaly. Therefore, *Synth1* is divided into five windows, each with a length of 277 (Figure 1.b). The collective anomaly spans two of these windows, namely w_4 and

SCJ Accepted

w_5. Specifically, w_4 contains 12 anomalous points, while w_5 contains 265 anomalous points. $iTAD$ method correctly identifies w_5 as the most anomalous window. The resulting $iTAD$ outcomes are as follows: TP=265, $FP = 12$, $TN = 1073$, and $FN = 12$ (Figure 1.b).

Ideally, an $iTAD$ method should yield a $Fscore$ of 1, with $FP = 0$ and $FN = 0$. However, in this example, a $Fscore$ of 0.957 was obtained. This discrepancy is attributed to the fact that twelve anomalous points were not included in the identified anomalous window. Consequently, $FP = 12$ and $FN = 12$ were observed. This example highlights that even when the ideal model is aware of the correct anomalous window and correct window size, it may not consistently yield optimal results. Specifically, a TAD method that utilizes a sliding window mechanism inherently introduces conditions that can result in the occurrence of false positives. These false positives, in turn, lead to a decrease in the number of true negatives. In Section 4.3, we demonstrate that the proposed $slid\&cons$ mechanism shows a significant improvement in decreasing false positives, with a remarkable 79% reduction, and a 2% increase in true negatives for this particular example. ■

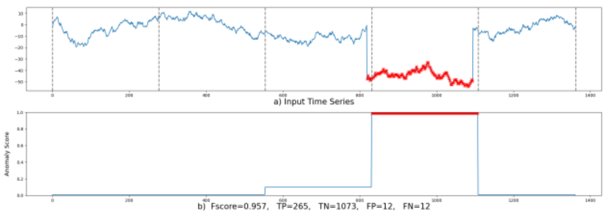


Figure 1: Anomaly detection with $iTAD$ on *synth1*

In the following, a basic theoretical calculation for evaluating the performance of both $iTAD$ and $qTAD$ algorithms is presented based on the example above. The $qTAD$ is a potential solution that involves a simple sliding and consensus process.

Lemma 1*: $iTAD$ and $qTAD$ performance evaluation*

Let's assume that $x(t)$ represents a time series that contains a collective anomaly and $iTAD$ is an ideal window-based time series algorithm. $iTAD$ identifies an anomalous window correctly (W_{ano}) with a window size of w in $x(t)$. Also, t is the time series length, and l is the length of a collective anomaly within the time series. It is specified that at least half of the anomaly ($l/2$) is in W_{ano}(Figure 2). Other assumptions are in the following.

$$\begin{cases} 0 < l \le w \\ x(t) = (x_1, \dots, x_t) \\ w_{ano} = x(j, j+w-1) = (x_j, \dots, x_{j+w-1}), \\ \qquad 1 \le j < j+w \le t \\ x(i, i+l-1) = (x_i, \dots, x_{i+l-1}), \\ \qquad 1 \le i < i+l \le t, j \le i + \frac{l}{2} \le w \end{cases} \tag{1}$$

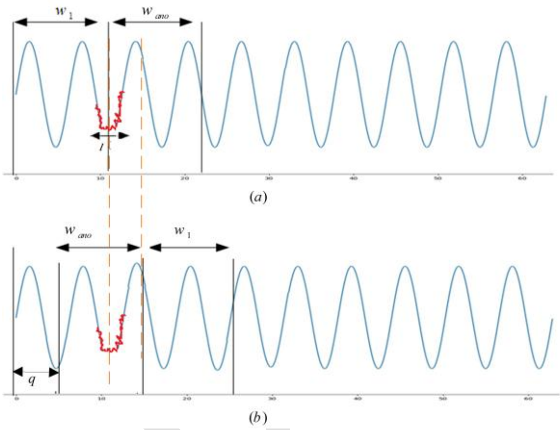


Figure 2: The evaluation of (a) $iTAD$ (b) $qTAD$

The evaluation results of $iTAD$ algorithm are as follows:

$$iTAD = \begin{cases} TP = \frac{l}{2}, FP = w - \frac{l}{2}, FN = \frac{l}{2} \\ Precision = \frac{l/2}{l/2 + w - l/2} = \frac{l}{2w} \\ Recall = \frac{l/2}{l/2 + l/2} = \frac{1}{2} \\ Fscore = 2 \times \frac{\frac{l}{2w} \times \frac{1}{2}}{\frac{l}{2w} + \frac{1}{2}} = \frac{\frac{l}{2w}}{\frac{l+w}{2w}} = \frac{l}{l+w} \end{cases} \tag{2}$$

$qTAD$ is a window-based TAD algorithm that operates as follows:

1- $iTAD$ find W_{ano1} on W^0
2- Time series windows are slid by a value of q where $0 < q \le \frac{w}{2}$ and creates W^1
3- $iTAD$ find W_{ano2} on W^1
4- $W_{ano} = W_{ano1} \cap W_{ano2}$

where W^i is windows set that has been slid i times.

The evaluation results of $qTAD$ algorithm are in Equation (3).

$$qTAD = \begin{cases} TP = \frac{l}{2}, FP = q - \frac{l}{2}, FN = \frac{l}{2} \\ Precision = \frac{\frac{l}{2}}{\frac{l}{2} + q - \frac{l}{2}} = \frac{l}{2q} \\ Recall = \frac{\frac{l}{2}}{\frac{l}{2} + \frac{l}{2}} = \frac{1}{2} \\ Fscore = 2 \times \frac{\frac{l}{2q} \times \frac{1}{2}}{\frac{l}{2q} + \frac{1}{2}} = \frac{\frac{l}{2q}}{\frac{l+q}{2q}} = \frac{l}{l+q} \end{cases} \quad (3)$$

The ratio of $Fscore_{qTAD}$ to $Fscore_{iTAD}$ is given by:

$$\begin{aligned} \frac{Fscore_{qTAD}}{Fscore_{iTAD}} &= \frac{\frac{l}{l+q}}{\frac{l}{l+w}} = \frac{l+w}{l+q} \\ &= \frac{l+w}{l+\frac{w}{2}} \frac{l+w+l-l}{\frac{2l+w}{2}} \\ &= 2 - \frac{2l}{2l+w} > 1 \left(q = \frac{w}{2}\right) \end{aligned} \quad (4)$$

$$\frac{Fscore_{qTAD}}{Fscore_{iTAD}} = 2 \times \frac{2w}{3w} = \frac{4}{3} > 1 \quad \left(q = \frac{w}{2}, w = l\right) \quad (5)$$

The calculations of Equations (4) and (5) demonstrate that the ratio is greater than one, indicating the superiority of $qTAD$ over $iTAD$.

■

Example 2 provides a numerical illustration of Lemma 1.

Example 2: Let be $t = 30, w = l = 10,\ q = 5$, $x(30) = (x_1, \dots, x_{30})$, and collective anomaly $= x(5{,}14) = (x_5, \dots, x_{14})$. According to lemma 1, $iTAD$ operates on $x(t)$ as follows:

$w_1 = x(1{,}10) = (x_1, \dots, x_{10})$,
$w_{ano1} = x(11{,}20) = (x_{11}, \dots, x_{20})$

$$iTAD = \begin{cases} TP = 5, FP = 5, FN = 5 \\ Precision = \frac{5}{5+5} = \frac{5}{10} = \frac{1}{2} \\ Recall = \frac{5}{5+5} = \frac{1}{2} \\ Fscore = 2 \times \frac{\frac{1}{2} \times \frac{1}{2}}{\frac{1}{2} + \frac{1}{2}} = \frac{\frac{1}{2}}{1} = \frac{1}{2} \end{cases} \quad (6)$$

And $qTAD$ with $q = 5$ operates on $x(t)$ as follows:

$$w_1 = x(15{,}24) = (x_{15}, \dots, x_{24}) \quad (7)$$

$w_{ano2} = x(5{,}14) = (x_1, \dots, x_{14})$,
$w_{ano1} \cap w_{ano2} = (x_{11}, \dots, x_{14})$

$$qTAD = \begin{cases} TP = 4, FP = 0, FN = 6 \\ Precision = \frac{4}{4} = 1 \\ Recall = \frac{4}{4+6} = \frac{4}{10} = \frac{2}{5} \\ Fscore = 2 \times \frac{1 \times \frac{2}{5}}{1 + \frac{2}{5}} = \frac{\frac{4}{5}}{\frac{7}{5}} = \frac{4}{7} \end{cases}$$

$$\frac{Fscore_{qTAD}}{Fscore_{iTAD}} = \frac{\frac{4}{7}}{\frac{1}{2}} = \frac{8}{7} = 1.14$$

In the example, by employing $iTAD$, only data points within identified anomalous windows are labeled as anomalies. This approach may label neighboring anomalous points, which do not precisely fall within the identified anomalous windows, as normal points, leading to an increase in false negatives (FN). In addition, the sliding and consensus mechanism in $qTAD$ addresses this limitation by identifying anomalous neighbors close to the identified anomalous windows. This is achieved through multiple iterations of $qTAD$ execution and the aggregation of anomaly scores assigned to data points. ■

Based on the preceding discussion, the sliding mechanism in window-based methods can improve the overall performance by decreasing FP and increasing TN. In this paper, the proposed method utilizes a Base TAD algorithm in an iterative manner on sliding windows ($slid\&cons$). The main idea of the $slid\&cons$ mechanism is when a specific range of the time series obtains high anomaly scores in multiple consecutive sliding processes, the aggregation function assigns a higher degree of anomaly to that window. In contrast, for time points that consistently have low anomaly scores, even if they occasionally achieve high anomaly scores in a small number of slides, the aggregation function assigns a final low anomaly score to them.

The proposed $slid\&cons$ mechanism effectively leverages the aggregated results obtained from several iterations to improve the overall performance of the proposed method.

The main contributions of this paper can be summarized as follows:

- Development of an unsupervised sliding window-based anomaly detection method for multivariate time series called ANNOTATE.
- Proposal of a novel $slid\&cons$ mechanism that utilizes sliding windows and a consensus aggregation technique to improve anomaly detection performance by decreasing FP and increasing TN.

SCJ Accepted

The paper is organized as follows. In Section 2, window-based TAD methods are briefly reviewed. Section 3 describes the proposed method in detail. Several experiments on some synthetic and real-world datasets are presented in Section 4. Conclusions are given in Section 5.

2. Related Works

Outlier detection in multivariate time series involves diverse approaches, ranging from basic statistical analyses and machine learning methods to advanced deep learning techniques. The task of sliding window segmentation is an essential part of these techniques. Window-based methods are classified into two categories: fixed-length and variable-length. Fixed-length sliding windows are called Static Sliding Window (SSW), while variable-length sliding windows are known as Dynamic Sliding Window (DSW). The SSW approach utilizes sliding windows of a predetermined, fixed size, whereas the DSW approach adapts the window size based on the characteristics of the time series [12].

The segmentation process in TAD can be done in two ways: Top-Down or Bottom-Up. Top-Down algorithms recursively divide a time series into smaller segments until a specific stop condition is met. Bottom-Up algorithms start with the initial points of the input time series and gradually add points until certain conditions specified by the method are satisfied [13].

Bottom-Up methods can be applied to both online and offline inputs, making them suitable for handling streaming data and static datasets. Conversely, Top-Down methods are better suited for time series inputs that are not streaming or online, typically working effectively with static or offline datasets where the entire time series is available for analysis [13].

In TAD algorithms, a fixed-length sliding window is the most common [3]. The main objective of these algorithms is to approximate the optimal sliding window size. The length of sliding windows is often determined either by expert users or by employing brute-force methods within the algorithms [14]. Izakian et al. employed a fixed-length window for detecting anomalies. They utilized the fuzzy c-means clustering algorithm to identify anomalies within the sliding windows [15].

The window length influences the performance of the SSW method. The window length is typically determined based on the overall trend of the input time series. While the window length should be a function of data fluctuations, taking into account the varying characteristics of the time series.

To enhance the performance of TAD models using fixed-length windows, some studies have incorporated additional steps into their models. Yin et al. conducted research in the IoT domain utilizing a deep convolutional network. Their model employed a two-stage sliding window approach during preprocessing. In the first stage, the time series was divided into fixed-size windows, and in the second stage, these windows were further divided into smaller subsequences to extract features [16].

As mentioned earlier, window-based TAD methods classify all points within an anomalous window as anomalies, potentially leading to increased FP. Researchers have introduced an overlapping mechanism that creates sliding windows with overlaps between neighboring windows to address this. In reference [17], the authors utilized overlapped windows to enhance the efficiency of their proposed method for multidimensional TAD.

Another solution is to employ a dynamic or adaptive sliding window mechanism, which utilizes a specialized algorithm to determine specific window lengths for each window. Smrithy et al. employed a dynamic approach with the Weighted Moving Average (WMA) method to detect outliers in the healthcare domain. This algorithm estimates the size of the subsequent sliding window by evaluating the variance between the preceding sliding window and the current sliding window [18].

In a study on road anomaly detection, the authors introduced a dynamic sliding window mechanism. The algorithm determines the length of windows using the DSW method, which leverages vehicle speed. Additionally, this method calculates a dynamic overlap value for each window [12].

In recent times, researchers have employed deep learning methods in the sliding window process. Baig et al. worked on multivariate time series of data center resources [19]. They proposed an adaptive sliding window approach that utilizes a 4-layer MLP to determine the length of each window dynamically.

Utilizing mathematical estimation methods is another approach for time series segmentation. Carmona-Poyato presented the optimal window segmentation technique, OSTS, which employs the A^* algorithm to achieve efficient segmentation of time series. This algorithm is employed to calculate optimal polygonal approximations of the time series [20].

SCI Accepted

Yao et al. employed a dynamic sliding window approach for anomaly detection in wireless networks [21]. Their proposed method combines a basic window size and historical information to determine the optimal window size for streaming data. The dynamic model operates by analyzing the continuous local fluctuations within the data.

An adaptive sliding window method is proposed in [11] to improve outlier detection efficiency. Farahani et al. discovered the normal behavior of the input time series using overlapped windows with a DSW mechanism to cluster the data [22].

3. The Proposed Method

This section presents ANNOTATE, an unsupervised sliding window-based TAD method for identifying anomalies in multivariate time series. The method utilizes a Base TAD model to detect anomalous windows. To enhance the outcomes of the Base TAD algorithm, the original time series is shifted multiple times ($slid$), and Base TAD is rerun on each shifted time series. The $slid$ mechanism modifies the position of points within windows by shifting the input time series. This change leads to changes in the assignment of anomaly score values. The assigned anomaly scores are then aggregated using a consensus function (cons) over multiple iterations. If a subsequence within consistently acquires a high anomaly degree after aggregating the scores, it is classified as an abnormal subsequence.

An overview of the methods is shown in Figure 3. The proposed model includes a main loop encompassing segmentation, representation, anomaly scoring, and $slid\&cons$ steps. The combination of these three steps, namely segmentation, clustering representation, and anomaly score calculation, is referred to as Base TAD.

The input time series is prepared in the pre-processing step by detrending tasks. The pre-processed time series is divided into fixed-length windows during the segmentation step. These windows are then transformed into a new form using a clustering algorithm ($OSCM$) [23] in the representation step. Anomaly scores are assigned to the transformed windows in the next step by computing d-neighbor distances. The main loop concludes with the sliding mechanism, which slides the input time series by a factor of q . The loop repeats n times, as illustrated in Figure 4. The $Cons$ function (Equation (2)) combines all the anomaly scores generated in n iterations for time series points and assigns new anomaly scores (AS).

The normality or abnormality status of points is determined based on the computed AS. Algorithm 1 presents the pseudocode of the proposed method. Additional details can be found in the subsequent subsections.

3.1 Preprocess

Assume that $x(t) = (x_1, \ldots, x_t)$ represents a multivariate time series of length t in the preprocessing step where $x_i = (x_{1,i}, x_{2,i}, \ldots, x_{p,i})$ denotes the ith point of the series with p dimensions. Time series $\Delta x(t-1) = (\Delta x_1, \ldots, \Delta x_{t-1})$ is constructed where $\Delta x_i = x_{i+1} - x_i,\ i \in [1, t-1]$ (Line 3 of Algorithm 1).

3.2 Window Segmentation

In this step, m multivariate subsequences of length w are generated. More formally, the time series $\Delta x(t-1)$ is converted into a set of windows, $W^s = \{w_1^s, \ldots, w_m^s\}$ where w_i^s is the ith window in the sth iteration and $m = \lfloor (t-1)/w \rfloor$ (Line 9).

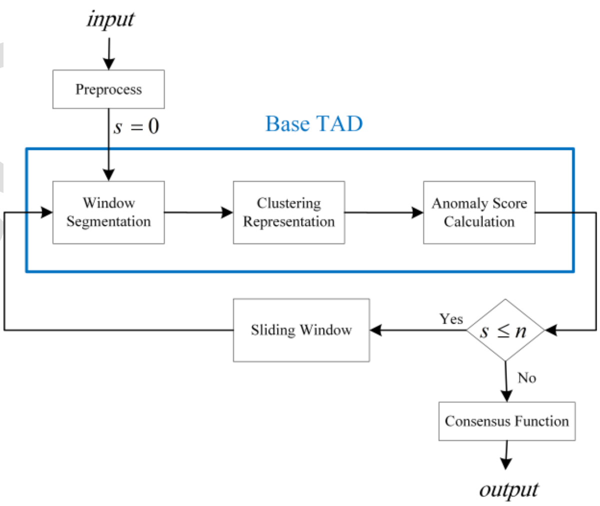


Figure 3: Overall scheme of the proposed TAD

Algorithm 1: Pseudo-code of ANNOTATE

Inputs: $x(t) = (x_1, \ldots, x_t)$: Multivariate time series
w: Window size, k : The number of clusters
n : Maximum number of sliding steps
sp : Sliding percentage
Outputs: Labels time series points (0:Normal, 1:Abnormal)
1: $m \leftarrow \lfloor (t-1)/w \rfloor$
2: $q \leftarrow sp \times w$
3: **for** i=1 to t-1 **do**
4: $\Delta x_i = x_{i+1} - x_i$ ▷ $\Delta x(t-1) = \{\Delta x_1, \ldots, \Delta x_{t-1}\}$
5: **end for**
6: **for** s=0 to n **do**
7: $start \leftarrow s \times q$
8: $\Delta x^s \leftarrow \Delta x(start, t)$
9: $W^s \leftarrow$ divide Δx^s to m windows ▷ $W^s = \{w_1^s, \ldots, w_m^s\}$
10: $C^s \leftarrow OSCM(W^S, k)$ ▷ $C^s = \{c_1^s, \ldots, c_m^s\}$
11: $AS^s \leftarrow AS(C^s)$ by Equation (8) ▷ $AS^s = \{as_1^s, \ldots, as_m^s\}$
12: **end for**

```
13: for i=1 to m do
14:      compute AS_i^T by Equation (9)  ▷ AS^T = { AS_i^T, ..., AS_i^m}
15: end for
16: compute τ by Equation (11)
17: compute Label(x_j) by Equation (10)
18: return Label
```

3.3 Clustering Representation

Each $w_i^s, i \in [0, m]$ is divided into k clusters utilizing a clustering method. A modified version of optimal clustering for sequential data ($OSCM$) is developed for multivariate time series based on the OSC method [23]. An individual cluster is represented by its center point. A set of cluster centers $C^s = \{c_1^s, \dots, c_m^s\}$ is formed in Line 10, where c_i^s is a set of k cluster centers in ith sliding window and sth iteration.

3.4 Neighboring Distance as Anomaly Score

Given that a set of windows $W^s = \{w_1^s, \dots, w_m^s\}$ and cluster center set $C^s = \{c_1^s, \dots, c_m^s\}$ were constructed by previous steps, the anomaly score set $AS^s = \{as_1, \dots, as_m\}$ is generated by Equation (8) (Line 11). The anomaly score for the ith window in sth iteration, as_i^s, is equal to the average of the distances from the d-previous neighbors. For Algorithm 1, $d = 2$ is defined.

$$as_i^s = \begin{cases} 0, & i = 1 \\ \sum_{j=1}^{k} \frac{\sum_{z=1}^{i-1} \left| c_{z,j}^s - c_{i,j}^s \right|}{d}, & 1 < i \le d \\ \sum_{j=1}^{k} \frac{\sum_{z=1}^{d} \left| c_{z-i,j}^s - c_{i,j}^s \right|}{d}, & d < i \le m \end{cases} \tag{8}$$

where $c_{i,j}^s$ is the center of jth cluster of the ith sliding window, and k is the number of clusters.

3.5 Sliding Window and Consensus Function ($slid\&cons$)

In this step, sliding windows are shifted n times with a step size of q, as illustrated in Figure 4, generating sets of W^s where $s \in [1, n]$. The clustering (described in Subsection 3.3) and scoring (described in Subsection 3.4) phases are then performed on each W^s. This results in anomaly score sets, AS^s (Line 11). This process constructs a specific set of anomaly scores is $AS_i = \{ as_i^1, \dots, as_i^n\}$ for ith window.

A consensus function defined by Equation (9) is used to aggregate anomaly scores in AS_i^T (Line 14).

$$AS_i^T = \sum_{s=1}^{n} \exp(as_i^s), i \in [1, m] \tag{9}$$

where AS_i^T is the aggregated anomaly score for the ith window.

Anomalous points are detected by applying Equation (10), which is implemented in Line 17 of the algorithm.

$$Label(x_j) = \begin{cases} 1, & AS_i^T > \tau \\ 0, & other \end{cases}, i \in [1, m], x_j \in w_i \tag{10}$$

where τ is the anomaly threshold, which is defined as follows:

$$\begin{aligned} \mu &= mean(AS_i^T), \quad i = 1,2,3 \\ \sigma &= std(AS_i^T), \quad i = 1,2,3 \\ \tau &= \mu + \sigma \end{aligned} \tag{11}$$

where $mean$ and std are the average and standard deviation functions, respectively. Equation (11) used the anomaly threshold computed in [24].

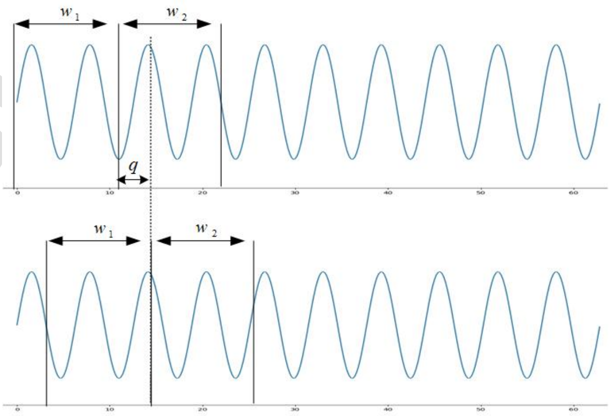


Figure 4: Sliding process

4. Experimental Studies

This section presents and discusses the experimental results that demonstrate the effectiveness of the proposed method. The evaluation of the proposed algorithm involved comparing it against 12 existing methods, which were implemented using the scikit-learn[2], AGOTS[3], and CUBOID[4] packages. The proposed method was implemented in Python 3.8. The

[2] https://scikit-learn.org
[3] https://github.com/KDD-OpenSource/agots
[4] https://github.com/ir1979/CUBOID

SCJ Accepted

experiments were conducted on a computer with an Intel(R) Core(TM) i7-7700HQ processor running at 2.80 GHz, equipped with 16.0 GB RAM, and operating on Windows 10.

For all experiments, the proposed method was executed with $k = 3$. The comparative methods were evaluated using different window sizes (w). The experiments with the best results are reported in this section.

The following subsections introduce datasets used in the experiments (subsection 4.1) and details the performance indices employed for evaluation (subsection 4.2). Furthermore, subsections 4.3, and 4.4 describe and discuss two experiments conducted to assess the performance of the proposed method.

4.1 Datasets

Experiments in this study utilized a combination of synthetic and real-world multivariate time series datasets. A brief overview of these datasets is provided below.

- The synthetic dataset, AGOTS, consists of 150 multivariate time series that contain collective anomalies. These anomalies can be categorized into four types: extreme, shift, trend, and variance. The dataset was generated using the AGOTS package[5]. The time series have random lengths ranging from 1000 to 3500, and the length of collective anomalies was randomly generated between 100 and 400 with a seed value of 3. The collective anomalies were inserted at random positions within the time series.

- MSL dataset comprises 27 spacecraft telemetry multivariate signals obtained from NASA's Curiosity Rover on Mars[6] [25].

- SKAB anomaly benchmark dataset includes 34 time series with collective anomalies. These time series were collected from a water system equipped with sensors in a testbed[7] [26]. The dataset includes three sub-datasets: other, valve1, and valve2.

For more detailed information about these datasets, please refer to Table 1. On the table, Clctv stands for "collective," and Num stands for "numerical."

[5] https://github.com/KDD-OpenSource/agots
[6] https://s3-us-west-2.amazonaws.com/telemanom
[7] https://github.com/waico/SKAB

Table 1: Summary of the datasets

Dataset	Count	Average Length	Ave # Anomalous Points	Ave Anomalies Percentage	#Features	Type of anomalies	Data Type
AGOTS	150	2444.68	248.38	11.49	4	Clctv	Num
SKAB	34	1101.74	389.44	35.48	8	Clctv	Num
MSL	27	2730.7	286.3	1.33	55	Clctv	Num

4.2 Performance Evaluation Criteria

The evaluation of anomaly detection methods commonly employs performance criteria such as $Accuracy,\ Precision,\ Recall,\ and\ Fscore$ (Equation (12)). These metrics are derived from the confusion matrix, which includes True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

$$\begin{gathered} Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \\ Precision = \frac{TP}{TP + FP} \\ Recall = \frac{TP}{TP + FN} \\ Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall} \end{gathered} \tag{12}$$

Additionally, the method's performance is quantified using the Area Under the Curve (AUC) of the receiver operating characteristics (ROC). The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR). The TPR is the percentage of actual positives that are predicted as positive, and the FPR is the percentage of actual negatives that are predicted as positive. The AUC provides a summary of the ROC curves, depicting the trade-off between true positives and false positives. The AUC is a measure of the overall performance of the method. A perfect method would have an AUC of 1, while a random method would have an AUC of 0.5.

In evaluating anomaly detection methods, $Fscore$ and AUC are considered the most crucial indices for assessing performance. These metrics effectively measure the method's performance in the TAD domain.

4.3 Visualization of $Slid\&cons$

The efficiency of the proposed model is visualized in this subsection. Time series *Synth1* and *Synth2* were selected from the synthetic dataset described in

SCJ Accepted

subsection 4.1. An explanation of the window-based method's problems on *Synth1* was given in the Introduction section. The results of executing Base TAD and ANNOTATE on *Synth1* and *Synth2* time series are presented in Figures 5, 6, and Table 2. Note that the Base TAD method applied to *Synth1* in Section 1 is $iTAD$, which is different from the Base TAD in the proposed method. Therefore, the results of their iterations are different. The parameters used in Base TAD in the experiment are $n = 2$ and $k = 3$. For *Synth1*, the parameter sp is set to 2%, and for *Synth2*, the parameter sp is set to 1%.

Figure 5 illustrates the executions of Base TAD and its two iterations on *Synth1*. The bottom graph of the figure (Figure 5.e) shows the result of aggregated scores from different runs of Base TAD using the $cons$ mechanism. It is evident that Base TAD identifies different anomaly ranges in each iteration. However, the red range in the last graph indicates that the anomalies detected by the $slid\&cons$ mechanism are more consistent with the true range of the anomaly. This signifies an improvement in the performance of Base TAD with the assistance of the $slid\&cons$ mechanism.

Furthermore, the numerical results in Table 2 support this superiority. The table reveals a 79% improvement in false positives (decreasing from 24 to 5) and a 72% improvement in false negatives (decreasing from 36 to 10) using the proposed mechanism compared to the basic model. Additionally, applying the $slid\&cons$ mechanism leads to a 10% increase in true positives and a 2% increase in true negatives in the *Synth1* time series.

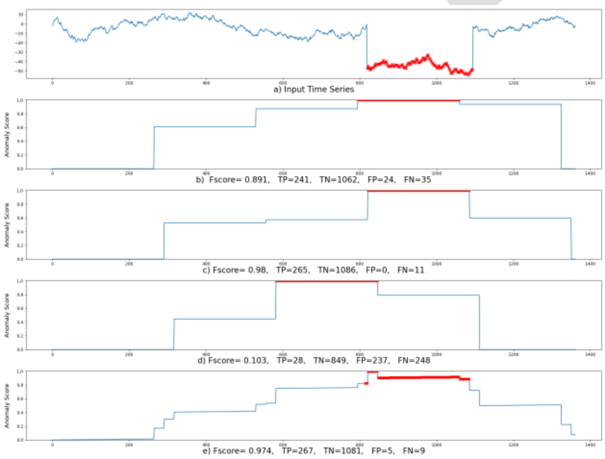


Figure 5: Anomaly detection with $slid\&cons$ on *Synth1*

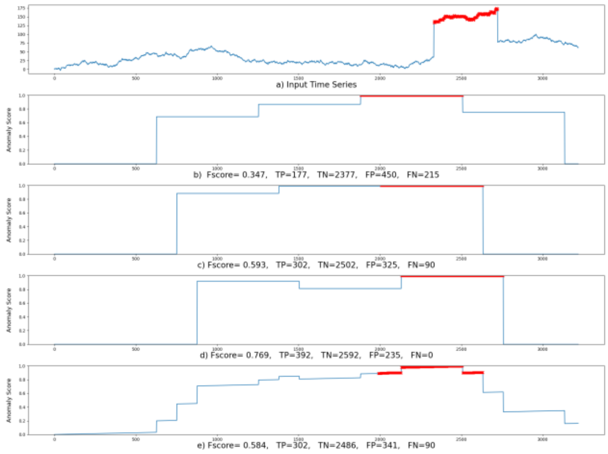


Figure 6: Anomaly detection with $slid\&cons$ on *Synth2*

Synth2 is a multivariate time series comprising four variables and exhibiting a collective anomaly, as depicted in Figure 6.a. Specifically, there is a collective anomaly of length 392 located at position 2331 within the time series. The total length of the time series is 3218.

Figure 6 shows the evaluation results of Base TAD and ANNOTATE methods employing a window size of 627. By comparing the red range in Figure 6.b with Figure 6.e, it becomes evident that the proposed mechanism detects an anomaly range that aligns more closely with the anomaly points in the time series. The aggregation mechanism has adjusted the red range to match the actual anomaly range, resulting in a more consistent identification of anomalous points that correspond to the true anomalies in the input time series.

The results presented in Table 2 demonstrate the effectiveness of the proposed mechanism. The proposed mechanism yields significant improvement by reducing the false negative value by 58% from 215 to 90. Moreover, there is a notable increase in true positives, rising from 177 in Base TAD to 305 in ANNOTATE, leading to a 41% enhancement in performance. The table results also indicate improvements in other performance parameters achieved by employing the proposed mechanism.

Table 2: Evaluation results of *Synth1* and *Synth2*

Series Name	Method Name	TP	TN	FP	FN
Synth1	Base TAD	241	1062	24	36
	ANNOTATE	267	1081	5	10
Synth2	Base TAD	177	2377	450	215
	ANNOTATE	302	2486	341	90

4.4 Effectiveness of the ANNOTATE

In this experiment, the overall performance of the proposed method was evaluated on a synthetic and two real-world datasets. The performance indices used in the experiment were $accuracy$, $precision$, $recall$, $Fscore$, and AUC. Comparative experiments were conducted using unsupervised methods such as CBLOF (cluster-based local outlier factor) [27], COF (Connectivity-Based Outlier Factor) [28], HBOS (Histogram-Based Outlier Score) [29], KNN (K-Nearest Neighbors Detector) [30], LOF (Local Outlier Factor) [31], MCD (Minimum Covariance Determinant) [32], PAA (Piecewise Aggregate Approximation) [33], PCA (Principal Component Analysis) [34], SOD (Subspace Outlier Detection) [35], SOS (Stochastic Outlier Selection) [36], and OCSVM (One-class SVM detector) [37].

4.4.1 Synthetic Dataset Experiments

To test the performance of the proposed algorithm, the AGOTS dataset is used in the experiment with $sp = 10\%$ and $n = 4$ parameters for ANNOTATE.

The results of performance indices are summarized in Figure 7 and Table 3. The superior performance of the proposed method is visually depicted in Figure 7, where the ANNOTATE 's line plots appear at the top compared to other methods.

Table 3 shows outputs of other approaches are poor. MCD was the best competitive method with $Fscore = 0.286$ and $AUC = 0.784$. The proposed method with $Fscore = 0.509$ and $AUC = 0.910$ shows significant improvement over the others.

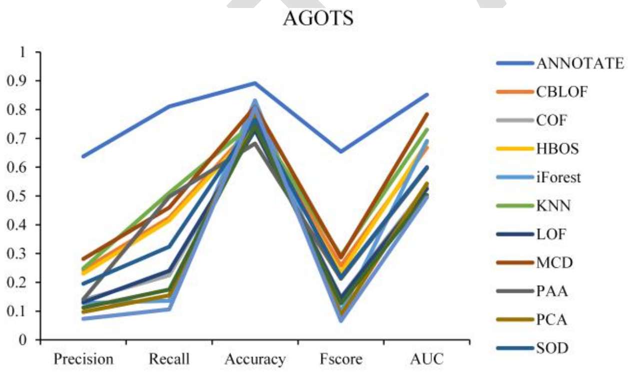


Figure 7: The performance indices' line plots for the AGOTS dataset

4.4.2 Real-world Dataset Experiments

Two real-world multivariate time series, SKAB, and MSL, were analyzed in the experiment to evaluate the proposed. The proposed algorithm was applied to SKAB with sp=1% and n=5, while for MSL, the parameters used were sp=5% and n=3. The outcomes of the SKAB and MSL experiments are presented in Figures 8, 9, and Tables 4, and 5, respectively. Since SKAB and MSL datasets are widely used, we present their performance using box plots (Figures 8 and 9), which offer additional statistical information about the experimental results.

Based on the results presented in Table 4, the proposed models demonstrate higher average values for all performance indices on the SKAB dataset. The average values for indices in the proposed models are as follows: $precision =$ 0.906, $recall$= 0.915, $accuracy = 0.936$, $Fscore$= 0.902, and AUC= 0.957.

Furthermore, Figure 8 provides additional evidence supporting these findings. The box plots of the ANNOTATE are at the top of the other plots and are also positioned closer to 1. This indicates that the ANNOTATE model is effective in detecting anomalies.

Moreover, the box plots of the proposed model include narrow interquartile ranges and small upper whiskers, suggesting that the values of the indices obtained for 75% of the SKAB time series are highly concentrated. Additionally, the median values in the box plots are closer to Q3, indicating that a significant portion of the dataset (50%) achieves high performance within the IQR range. These findings further indicate the proposed model's stable behavior and high efficiency.

In contrast, the box plots of the other competing methods display larger interquartile ranges and whiskers that extend over a wider range on both sides. Most of these methods have Q3 values below 0.6, except for $\boldsymbol{accuracy}$, suggesting higher fluctuations and lower overall performance.

The results of the experiments conducted on the MSL dataset are visually and numerically presented in Figure 9 and Table 5.

Table 5 indicates that the proposed model achieves an average $Fscore$ value of 0.620, which is the highest among other methods for the MSL dataset. The value denotes a significant improvement compared to the second-best result obtained by the PAA method, which scored only 0.279. The superiority of the proposed method becomes evident in the $Fscore$ box plots, where the ANNOTATE plot demonstrates a lower quartile value near 0.5. In contrast, the upper quartile is below 0.5 for other methods.

Accepted

Table 3: Performance evaluation of the AGOTS dataset

Method	Precision	Recall	Accuracy	Fscore	AUC
ANNOTATE	**0.637**±0.30	**0.811**±0.20	**0.892**±0.11	**0.654**±0.22	**0.852**±0.17
CBLOF	0.243±0.26	0.425±0.34	0.751±0.18	0.258±0.23	0.668±0.19
COF	0.139±0.11	0.224±0.15	0.744±0.08	0.148±0.08	0.534±0.07
HBOS	0.230±0.23	0.415±0.36	0.739±0.18	0.233±0.20	0.685±0.20
iForest	0.128±0.17	0.136±0.25	0.832±0.09	0.097±0.14	0.691±0.18
KNN	0.248±0.17	0.511±0.33	0.757±0.10	0.292±0.18	0.730±0.19
LOF	0.131±0.10	0.240±0.18	0.729±0.10	0.145±0.09	0.525±0.09
MCD	0.281±0.30	0.460±0.43	0.810±0.13	0.286±0.29	0.784±0.21
PAA	0.141±0.14	0.497±0.46	0.682±0.11	0.212±0.21	0.596±0.20
PCA	0.097±0.18	0.155±.30	0.784±0.13	0.086±0.15	0.543±0.27
SOD	0.195±0.13	0.324±0.17	0.765±0.07	0.217±0.11	0.600±0.11
SOS	0.112±0.06	0.175±0.05	0.744±0.05	0.127±0.05	0.503±0.01
OCSVM	0.073±0.16	0.106±0.24	0.807±0.10	0.066±0.14	0.495±0.28

Table 4: Performance evaluation of the SKAB dataset

Methods	Precision	Recall	Accuracy	Fscore	AUC
ANNOTATE	**0.906±0.18**	**0.915**±0.20	**0.936**±0.10	**0.902**±0.18	**0.957**±0.10
CBLOF	0.353±0.12	0.222±0.10	0.592±0.05	0.266±0.11	0.512±0.06
COF	0.370±0.10	0.174±0.07	0.604±0.04	0.231±0.08	0.508±0.04
HBOS	0.377±0.12	0.268±0.15	0.598±0.05	0.300±0.14	0.546±0.10
iForest	0.414±0.12	0.282±0.15	0.606±0.05	0.317±0.13	0.569±0.10
KNN	0.362±0.12	0.212±0.10	0.598±0.05	0.260±0.11	0.529±0.08
LOF	0.360±0.11	0.146±0.06	0.608±0.05	0.201±0.08	0.501±0.05
MCD	0.388±0.20	0.387±0.30	0.619±0.12	0.371±0.25	0.589±0.19
PAA	0.465±0.12	0.455±0.13	0.623±0.05	0.458±0.12	0.706±0.12
PCA	0.345±0.13	0.213±0.13	0.591±0.05	0.249±0.13	0.522±0.09
SOD	0.376±0.07	0.169±0.07	0.607±0.04	0.225±0.08	0.519±0.06
SOS	0.364±0.06	0.212±0.03	0.589±0.03	0.266±0.03	0.504±0.02
OCSVM	0.360±0.14	0.214±0.16	0.602±0.05	0.250±0.15	0.526±0.10

Table 5: Performance evaluation of the MSL dataset

Methods	Precision	Recall	Accuracy	Fscore	AUC
ANNOTATE	**0.574**±0.30	**0.881**±0.18	**0.891**±0.13	**0.620**±0.25	**0.889**±0.16
CBLOF	0.163±0.14	0.264±0.16	0.775±0.12	0.168±0.11	0.638±0.13
COF	0.201±0.15	0.336±0.32	0.751±0.20	0.174±0.11	0.583±0.11
HBOS	0.156±0.13	0.350±0.25	0.738±0.16	0.183±0.11	0.595±0.13
iForest	0.163±0.14	0.299±0.16	0.748±0.14	0.179±0.11	0.608±0.12
KNN	0.225±0.12	0.278±0.29	0.794±0.19	0.171±0.11	0.612±0.12
LOF	0.183±0.15	0.163±0.18	0.826±0.14	0.126±0.12	0.567±0.07
MCD	0.191±0.20	0.304±0.34	0.761±0.22	0.154±0.16	0.630±0.16
PAA	0.235±0.31	0.509±0.42	0.715±0.13	0.279±0.29	0.659±0.19
PCA	0.150±0.14	0.214±0.16	0.765±0.18	0.137±0.10	0.595±0.13
SOD	0.258±0.20	0.078±0.11	0.866±0.13	0.093±0.09	0.527±0.05
SOS	0.000±0.00	0.000±0.00	0.880±0.13	0.000±0.00	0.500±0.00
OCSVM	0.170±0.16	0.375±0.26	0.740±0.17	0.195±0.16	0.608±0.16

Furthermore, the ANNOTATE with the largest average value of recall = 0.881 and box plot with an

SCI Accepted

almost narrow IQR in a high range (0.75-1), demonstrates that the proposed model successfully enhances Recall by minimizing false negatives.

In general, the distributions of box plots for accuracy are considerably high and close to one for all methods. However, the proposed method with the median closer to the upper quartile and small whiskers on both sides indicates a negative skewness. These evidences imply that the proposed model accurately detects anomalies in this dataset and show the methods outperform.

Moreover, the superiority of ANNOTATE is clearly evident in the precision index. The proposed method attained the highest value of 0.574, significantly higher than the values of the other methods (Table 5). This superiority is further supported by the corresponding box plot in Figure 9.

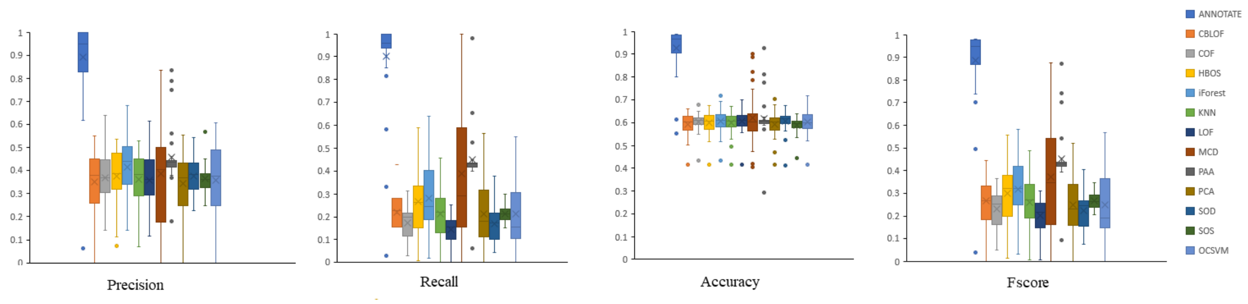
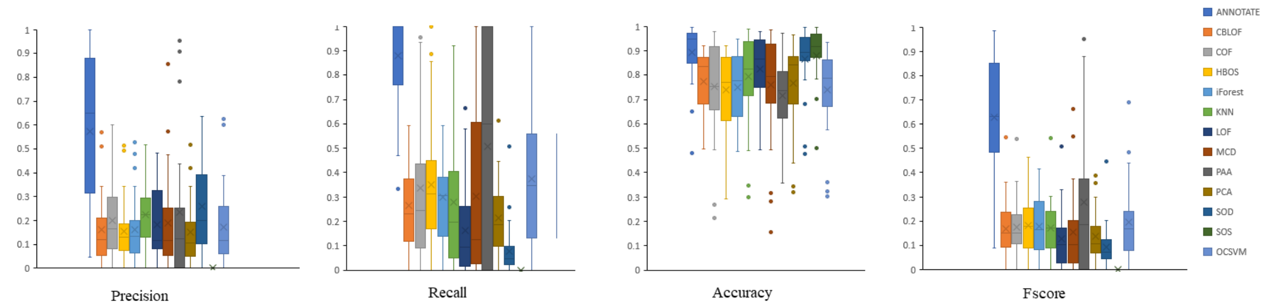


Figure 8: The performance indices' box plots for the SKAB dataset



(b)

Figure 9: The performance indices' box plots for the MSL dataset

Overall, the consistent results from both the tables and the box plots confirm the superior performance and stability of the proposed model compared to the competing methods. A careful examination of Tables 3, 4, and 5 reveals that the standard deviations associated with the proposed method are consistently below 0.2, indicating a high level of consistency and reliability. Moreover, when observing the ANNOTATE box plots depicted in Figures 8 and 9, it becomes evident that the their interquartile ranges usually are narrow and their upper whiskers are relatively small. These box plot characteristics further validate the stable manner in which the proposed model operates, reinforcing its reliability and robustness.

5. Conclusion

This paper introduced a novel unsupervised multivariate TAD method called ANNOTATE. ANNOTATE utilizes a window-based approach, employing a sliding and consensus mechanism (*slid&cons*) to combine anomaly scores from multiple executions of the method enhancing the final results. Several synthetic and real-world datasets are used to demonstrate the effectiveness of the proposed approach. Experiments show that the method outperforms all indices on datasets.

For future research, we recommend that researchers focus on adaptive window techniques. The paper used a fixed-length window size, but adaptive window techniques have the potential to enhance the results more efficiently by incorporating sliding mechanisms and aggregating functions. Furthermore, exploring various sliding mechanisms and consensus functions within this domain could further contribute to advancements in the field.

References

[1] Mohseni, N., Nemªtzadeh, H., Akbarib, E., and Motameni, H.,"*Outlier Detection in Test Samples using Standard Deviation and Unsupervised Training Set Selection*", International Journal of Engineering., 36(1), 119–129(2023), doi:https://doi.org/10.5829/IJE.2023.36.01A.14.

[2] M. Sheikhan and A. Abbasi, "*A novel combined solution for intrusion detection in computer networks using computational ntelligence algorithms*," Soft Computing Journal, vol. 6, no. 1, pp. 48–65, 2021[In Persian].

[3] Wu et al., W., "*Developing an Unsupervised Real-time Anomaly Detection Scheme for Time Series with Multi-seasonality*, " IEEE Transactions on Knowledge and Data Engineering, 4347(c), 1–1(2020), doi:https://doi.org/10.1109/tkde.2020.3035685.

[4] Blázquez-García, A., Conde, A., Mori, U., and J., Lozano, A. ,"*A Review on Outlier/Anomaly Detection in Time Series Data*," ACM Computer Survey, 54(3), 1-9 (2021), doi: https://doi.org/10.1145/3444690.

[5] Ramotsoela, D. T., Hancke, G. P., and Abu-Mahfouz, A. M., "*Attack detection in water distribution systems using machine learning*", Human-centric Computer Information Sciences., 9(1), 1-22(2019), doi:https://doi.org/10.1186/s13673-019-0175-8.

[6] A. Khaleghitabar and R. Farazkish, "*Routing Improvement for Vehicular Ad Hoc Networks (VANETs) Using Nature Inspired Algorithms* Soft", Computing Journal, vol. 6, no. 2, pp. 72–85, 2021[In Persian].

[7] H. Veisi, H. Ghaedsharff, and M. Ebrahimi, "*Improving the Performance of Machine Learning Algorithms for Heart Disease Diagnosis by Optimizing Data and Features*", Soft Computing Journal, vol. 8, no. 1, pp. 70–85, 2021, doi: 10.22052/8.1.70[In Persian].

[8] Karadayi, Y., Aydin, M. N., and Ög˘renci, A. S., "*A hybrid deep learning framework for unsupervised anomaly detection in multivariate spatio-temporal data*", Applied Sciences (Switzerland), 10(15), 2020, doi:https://doi.org/ 10.3390/app10155191.

[9] Castanhel, G. R., Heinrich, T., Ceschin, F., and Maziero, C. A., "*Sliding Window: The Impact of Trace Size in Anomaly Detection System for Containers Through Machine Learning*", Anais da XVIII Escola Regional de Redes de Computadores,141–146(2021), doi: https://doi.org/ 10.5753/errc.2020.15203.

[10] Chandola, V., Banerjee, A., and Kumar, V., "*Anomaly detection for discrete sequences: A survey*", IEEE Transactions on Knowledge and Data Engineering, 24(5), 823–839(2012), doi:https://doi.org/10.1109/TKDE.2010.235.

[11] Yao, H., Yang, Y., Fu, X., and Mi, C., "*An Adaptive Sliding-Window Strategy for Outlier Detection in Wireless Sensor Networks for Smart Port Construction*", Journal of Coastal Research., 82(10082), 245–253(2018), doi: https://doi.org/ 10.2112/SI82-036.1.

[12] Chibani, N., Sebbak, F., Cherifi, W., and Belmessous, K., "*Road anomaly detection using a dynamic sliding window technique*", Neural Computing and Applications, 34(21), 19015–19033(2022), doi:https://doi.org/10.1007/s00521-022-07436-6.

SCJ Accepted

[13] Keogh, E., Chu, S., Hart, D., and Pazzani, M.,"*Segmenting time series: a survey and novel approach*", Data Mining in Time Series Databases, 1–21, 2004.

[14] Hu, Y., Guan, P., Zhan, P., Ding, Y.,and Li, X., "*A novel segmentation and representation approach for streaming time series*", IEEE Access, 7(c), 184423–184437(2019), doi:https://doi.org/10.1109/ACCESS.2018.2828320.

[15] Izakian, H., and Pedrycz, W., "*Anomaly detection in time series data using a fuzzy c-means clustering*", Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 1513–1518(2003), doi: https://doi.org/ 10.1109/IFSA-NAFIPS.2013.6608627.

[16] Yin, C., Zhang, S., Wang, J., and Xiong, N. N., "*Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series*", IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(1), 112–122(2022), doi:https://doi.org/10.1109/TSMC.2020.2968516.

[17] Kieu, T., Yang, B., and Jensen, C. S., "*Outlier Detection for Multidimensional Time Series Using Deep Neural Networks*", 19th IEEE International Conference on Mobile Data Management (MDM), 125–134, 2018, doi:https://doi.org/10.1109/MDM.2018.00029.

[18] Smrithy, G. S., Balakrishnan, R., and Sivakumar, N., *Anomaly Detection Using Dynamic Sliding Window in Wireless Body Area Networks,* 99–108. 2019, doi:https://doi.org/10.1007/978-981-10-7641-1_8.

[19] Baig, S.-R., Iqbal, W., Berral, J. L.,and Carrera, D., "*Adaptive sliding windows for improved estimation of data center resource utilization*", Future Generation Computer Systems, 104, 212–224(2020), doi:https://doi.org/ 10.1016/j.future.2019.10.026.

[20] Carmona-Poyato, Á., Fernández-García, N. L., Madrid-Cuevas, F. J., and Durán-Rosal, A. M., "*A new approach for optimal time-series segmentation*", Pattern Recognition Letters, 135, 153–159(2020), doi:https://doi.org/ 10.1016/j.patrec.2020.04.006.

[21] Yao, H., Yang, Y., Fu, X., and Mi, C., "*An Adaptive Sliding-Window Strategy for Outlier Detection in Wireless Sensor Networks for Smart Port Construction*", Journal of Coastal Research, 82(10082), 245–253(2018), doi:https://doi.org/10.2112/SI82-036.1.

[22] Vasheghani Farahani, I., Chien, A., King, R. E., Kay, M. G., and Klenz, B., "*Time series anomaly detection from a markov chain perspective*", 18th IEEE International Conference on Machine Learning and Applications, ICMLA, 1000–1007(2019), doi:https://doi.org/10.1109/ICMLA.2019.00170.

[23] Lin C.-R., and Chen, M.-S., "*On the Optimal Clustering of Sequential Data*", SIAM International Conference on Data Mining, 141–157(2002), doi:https://doi.org/10.1137/1.9781611972726.9.

[24] Aburakhia, S., Tayeh, T., Myers, R., and Shami, A., "*A Transfer Learning Framework for Anomaly Detection Using Model of Normality*", 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 55–61(2020), doi:https://doi.org/10.1109/IEMCON51383.2020.9284916.

[25] Tariq S., et al., "*Detecting anomalies in space using multivariate convolutional LSTM with mixtures of probabilistic PCA*", ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2123–2133(2019), doi:https://doi.org/10.1145/3292500.3330776.

[26] Katser I. D., and Kozitsin, V. O., "Skoltech anomaly benchmark (SKAB)", Kaggle, 2020.

[27] He, Z., Xu, X., and Deng, S., "*Discovering cluster-based local outliers*", Pattern recognition letters, 24(9–0), 1641–1650(2003), doi:https://doi.org/10.1016/S0167-8655(03)00003-5.

[28] Pokrajac, D., Reljin, N., Pejcic, N., and Lazarevic, A., "*Incremental Connectivity-Based Outlier Factor Algorithm*", Visions of Computer Science-BCS International Academic Conference, 211–223(2008), doi: https://doi.org/10.14236/ewic/vocs2008.18.

[29] Goldstein M., and Dengel, A., "*Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm*", KI-2012 Poster Demo Track, 9(1), 59–63(2012).

[30]Ramaswamy, S., Rastogi, R., and Shim, K., "*Efficient Algorithms for Mining Outliers from Large Data Sets*", ACM SIGMOD International Conference on Management of Data, 427–438(2000), doi:https://doi.org/10.1145/342009.335437.

[31] Breunig, M. M., Kriegel, H. P., Ng, R. T., and Sander, J., "*LOF: Identifying Density-Based Local Outliers*", ACM SIGMOD International Conference on Management of Data, 93–104(2000), doi:https://doi.org/10.1145/342009.335388.

[32] Hardin J., and Rocke, D. M., "*Outlier detection in the multiple cluster setting using the minimum*

covariance determinant estimator", Computational Statistics and Data Analysis, 44(4),625–638(2004), doi:https://doi.org/10.1016/S0167-9473(02)00280-3.

[33] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S., "*Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases*", Knowledge and Information Systems, 3(3), 263–286(2001), doi:https://doi.org/10.1007/PL00011669.

[34] Shyu, M. L., Chen, S. C., Sarinnapakorn, K., and Chang, L., "*A Novel Anomaly Detection Scheme Based on Principal Component Classifier*", 3rd IEEE International Conference on Data Mining, 353-365(2003).

[35] Kriege, H. P., Kr□oger, P., Schubert, E., and Zimek, A., "*Outlier detection in axis-parallel subspaces of high dimensional data*", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5476 LNAI, 831–838(2008), doi:https://doi.org/10.1007/978-3-642-01307-2_86.

[36] Janssens, J. H. M., Huszar, F., Postma, E. O., and Herik, H. J. van den, *Stochastic Outlier Selection*, Tilburg center for Creative Computing, techreport, 1, 2012.

[37] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C., "*Estimating the support of a high-dimensional distribution*", Neural Computation, 13(7), 1443–1471(2001), doi:https://doi.org/ 10.1162/089976601750264965.

SCJ Accepted