High-efficiency Anomaly Detection of Traffic Data Stream Using Sequential Bi-Iteration SVD

Ming Xu, Jiani Li

***Abstract*—Fast traffic anomaly detection is vital to traffic management, which is required to make a timely response decisions for anomalous events. However, real-time anomaly detection becomes challenging with the ongoing traffic volume growth. Current studies rely on subspace-based methods, such as PCA, by projecting the original data onto the residual subspace. However, this process is very time-consuming and computationally intensive when dealing with huge datasets. Therefore, this paper proposes an anomaly detection method based on a sequential bi-iteration SVD algorithm (S-BiSVD) to improve detection efficiency. S-BiSVD is a streaming algorithm that quickly learns the subspace of traffic for real-time anomaly detection by handling the newly updated data column instead of the entire data matrix. The experimental results show that the detection efficiency of the proposed approach is much higher than the baseline regarding online traffic data updating, and our method's detection accuracy is at the same level as the baseline.**

***Index Terms*—Anomaly detection, Adaptive algorithm, Sliding window, Traffic data**

I. Introduction

DETECTING traffic anomalies, which is the premise of analyzing the root causes of unusual phenomena and taking a response is an important task for traffic management [1]. For instance, anomalies in urban road networks, such as widespread traffic jams, necessitate swift detection to enable authorities to promptly address these issues, thereby reducing road congestion duration and enhancing traffic efficiency to some extent. Furthermore, anomalies in data obtained from sensor networks play a vital role in pinpointing malfunctioning sensors [2]. Anomalous traffic patterns in a computer network could indicate a compromised system transmitting sensitive data to an unauthorized destination. In this case, network monitoring techniques, including MAC spoofing, IP spoofing, TCP/UDP fanout, detection of duplicate IP and MAC addresses, virus detection, bandwidth anomaly detection, and connection rate detection, help detect threats from various network infrastructure elements. Additionally, anomaly detection helps track the profiles of every system, application, or network [3]. However, traffic anomaly detection is facing increasing challenges regarding processing efficiency as the volume of traffic datasets is experiencing explosive growth [4]. The rapid growth of various emerging technologies, such as sensors, connected devices, smart home appliances, smart cities, 5G communication media, smartphones, mobile cloud, healthcare applications, multimedia, virtual reality, and autonomous automobiles, contribute to the huge accumulation of real-time data flowing in a network [5]. Networks generate an estimated 2.5 exabytes of data daily through rapid, extensive, and diverse traffic [6]. Similarly, the expanding road network and the growing number of vehicles produce a significant volume of real-time data in road traffic. In the realm of the Contemporary Internet of Things (IoT), diverse connected and mobile devices engage in machine-to-machine communication, generating extensive sensor data every second [3]. This data needs real-time monitoring, collection, and analysis to detect anomalous behaviors. Moreover, the collected sensor network may encompass various data types, including binary, discrete, continuous, audio, and video, contributing to the realm of big data. As reported in [7], 2.3 zettabytes of Internet Protocol (IP) traffic traversed the Internet in 2020, signifying an increase of 879 exabytes from 2015. This surge in data contributes to a delay in real-time anomaly detection [3].

The subspace-based signal analysis involves splitting the observations into a set of desired and disturbing components, which can be viewed in the signal and noise subspaces. The basic intuition of the subspace-based approach in anomaly detection is that a sample can be considered anomalous if it has a high component after projecting it into the noise subspace. This approach has been widely studied in previous research. However, in the context of massive online data being updated quickly, real-time detection does not work perfectly because the subspace-based method must process the entire data matrix, which is bulky and time-consuming to realize dimensionality reduction in each step.

This paper proposes a new real-time traffic anomaly detection method using big data. A streaming algorithm uses a sliding window to learn the traffic subspace rapidly. It differs from the subspace tracking methods mentioned above as it only processes the newly updated column data instead of the entire data matrix. The proposed approach is much more efficient than the baseline, achieving the same detection accuracy. The primary value of the proposed approach is its high-speed operation, which facilitates real-time traffic anomaly detection under big data. This finding is expected to help timely detect traffic anomalies, reduce the costs caused by these anomalies, and improve the efficiency of traffic systems.

The remainder of this paper is organized as follows. Section 2 reviews the literature relating to anomaly detection in traffic networks. Section 3 discusses the theory related to the proposed approach and provides a concise example for detecting anomalies. Section 4 introduces the experimental procedure and discusses the experimental results. Finally, section 5 summarizes and concludes this work and provides future research directions.

Manuscript received April 25, 2023; revised December 9, 2023.

This work was funded by the National Natural Science Foundation of China under the Grant No.62173171.

Ming Xu is an Associate Professor of School of Software, Liaoning Technical University, Huludao 125105, China (phone: +8615810261581; e-mail: xum.2016@tsinghua.org.cn).

Jiani Li is a postgraduate student of School of Software, Liaoning Technical University, Huludao 125105, China (e-mail: 786125906@qq.com).

II. Research Background

A. Anomaly Detection in a Road Network

Recent studies applied a spatiotemporal model based on tensors for anomaly detection in a road network. This method decomposes tensors through sliding windows and measures the deviations of diverse spatiotemporal patterns to identify different types of anomalies [8] [9]. In [10], GPS data from vehicles have been used to detect traffic congestion. Additionally, Pan et al. comprehensively considered the drivers' behavior and social media to identify anomalous events according to traveling behavior. Then, they mined representative words from social media to depict the captured anomalous events [11]. In [12] and [13], the authors adapted the likelihood ratio test to detect anomalies in GPS data rapidly. In order to solve the widespread data sparsity in the real world, a data fusion method based on probability and utilizing datasets from different domains was proposed [14]. Zheng et al. used taxi trajectories to detect flaws in road network planning [15], and Liu et al. constructed causality trees to reveal the interaction among spatial and temporal anomalies and the potential defects in road network design [16]. Xu et al. used a ranking algorithm based on taxi trajectories to find the key nodes in a road network, which can be considered a special type of anomaly since their failure would result in sharp drops in traffic efficiency [17]. Additionally, PCA is often used in anomaly detection. Indeed, considering taxi trajectories, Chawla et al. used PCA to detect anomalies and then applied an optimization technique to infer anomalous paths by solving the L1 inverse problem [18].

B. Anomaly Detection in Other Networks

Traffic anomaly detection is also widely studied in IP, power, and sensor networks. In the network-wide anomaly detection algorithm [19], local monitors measure the total traffic volume (in bytes) on each network link and periodically centralize the data by pushing all recent measurements to a coordinator. Then, the coordinator performs PCA on an assembled matrix to detect anomalies. An anomaly detection method based on PCA and random matrix perturbation analysis was developed by Huang et al., and its accuracy was validated in the Abilene network. This Internet2 high-performance backbone network interconnects many universities and other research institutes [20]. Adrian Taylor et al. proposed an anomaly detector scheme based on a long short-term memory neural network to detect the controller area network bus attacks [21]. Livani et al. used distributed PCA and fixed-width clustering to establish a global normal profile and detect anomalies [22], while Xie et al. explored the distance-based anomaly detection method through PCA as a feature reduction technique [23]. Subspace-based methods have been widely studied in anomaly detection. However, most traditional detection methods suffer from low processing speed when dealing with big datasets. In contrast to the abovementioned work, the proposed approach significantly improves the detection speed and guarantees detection accuracy.

Given the features of streaming data, Ding et al. [24] introduced an online ensemble learning anomaly detection algorithm based on the IForest algorithm. However, the approach updates the anomaly detector by discarding the oldest isolation trees, potentially eliminating well-performing isolation trees and deteriorating the overall anomaly detection performance. Wang et al. [25] devised a streaming DBSCAN clustering algorithm on the spark platform for swift anomaly detection in large-scale electricity consumption data streams. Nevertheless, the constructed model demonstrates considerable complexity. Tin et al. [26] introduced an approach that utilizes ensemble learning techniques to address concept drift and incorporates an adaptive window mechanism to adapt to varying data distributions effectively. The adaptive window dynamically adjusts its size based on the characteristics of the data stream and specific application requirements, leading to improved performance and accuracy.

III. Methodology

A. Bi-Orthogonal Iteration SVD Algorithm

The bi-orthogonal iteration algorithm has been widely investigated by Strobach, who proposed various subspace tracking algorithms designed for exponential forgetting windows [27], [28]. This iterative algorithm computes the dominant singular values and vectors of a data matrix $X \in \mathrm{R}^{L\times T}$. The SVD of X is the factorization $X = U\Sigma V^T$, where U and V are orthonormal matrices, and Σ is a non-negative diagonal matrix: $\Sigma = \mathrm{diag}\,(\sigma_1, \sigma_2, \ldots, \sigma_r)$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$ and $r = \min(L, T)$. Thus, the γ dominant singular values are $\{\sigma_1, \sigma_2, \ldots, \sigma_\gamma\}$, with $\gamma \leq r$, the dominant left γ singular vectors are the γ first columns of the matrix U and the dominant right γ singular vectors are the γ first columns of the matrix V. In each iteration, this algorithm generates two auxiliary matrices to assist the next iteration, and unit matrices are considered initial auxiliary matrices in the first iteration. The bi-orthogonal iteration SVD can be adapted into a pattern for streaming data called sequential bi-iteration SVD, in which the U, Σ and V^T of its previous-step outputs and the currently updated data matrix are used as inputs. This strategy reproduces the decomposition results of this step.

B. Sequential Bi-Iteration SVD Algorithm

Many adaptive techniques are only applied to signals that change slowly [29]. In contrast, a few subspace trackers are based on sliding windows, which typically require more computation but provide a faster tracking response to sudden signal changes [30], [31].

S-BiSVD evolved from the classic sequential bi-iteration SVD algorithm and uses a sliding window. Data in the sliding window are denoted as:

$$X(T) = [x(t)x(t-1)\cdots x(t-L+1)]^T \qquad (1)$$

where $x(t)$ is the N-dimensional data vector at time t, and L is the window length. The details of S-BiSVD are reported in Table I. In each step, S-BiSVD only uses the newly updated data $x(t)$ as input. During the "Initialize" stage, I_r is the unit matrix, where $r = min(L, N)$. In the first iteration of each step, $B(t)$ comprises the first L rows of $B'(t)$, $U(t)$ and $B^*(t)$ in this step are the results of $B(t)$ after QR factorization. In the second iteration of each step $u_1(t)$, is the column vector obtained by transposing the first row of

$U(t)$, $V(t)$ and $S(t)$ in this step are the results of $A(t)$ after QR factorization. After two iterations in a step, the $U(t)$, $S(t)$ and $V(t)$ at time t are output.

TABLE I
DETAILS OF S-BISVD.

Initialize: $V(0) = \begin{bmatrix} I_r \\ 0 \end{bmatrix}; U(0) = \begin{bmatrix} I_r \\ 0 \end{bmatrix}; S(0) = [I_r];$
For each time interval do:
Input: x(t)
First Iteration:
$h(t) = V(t-1)^T x(t)$
$B'(t) = \begin{bmatrix} x(t)^T \\ \hline (t-1)S(t-1)^T \end{bmatrix}$
Take the first L rows of $B'(t)$ to obtain $B(t)$
$B(t) = U(t)B^*(t)$
Second Iteration:
$x_\perp(t) = x(t) - V(t-1)h(t)$
$A(t) = V(t-1)B^*(t)^T + x_\perp(t)u_1(t)^T$
$A(t) = V(t)S(t)$

The S-BiSVD algorithm has the following advantages:

(1) Some subspace-based estimation methods require calculating a standard orthogonal subspace basis at each step, as has been proved in music signals [32].

(2) Fast-tracking response to abrupt signal changes by a sliding window.

(3) Tracking the entire singular value decomposition (SVD), which may be useful for rank estimation and tracking purposes [28], [33].

(4) It relies on an approximate data matrix with fewer constraints than the classic projection method, leading to better tracking results [30]. This tracking algorithm has been tested to cope with transients and proven robust and fast.

C. Subspace-based Method in Anomaly Detection

Subspace-based methods, such as PCA, have been widely used for dimensionality reduction and lossy compression in data mining [34], [35]. PCA exploits the observation that in most explicitly high-dimensional data sets, there is a high implicit correlation between many dimensions, which can be inferred by carrying out an eigendecomposition of the data covariance matrix. This method selects a new dependent basis for the data, called the principal components. They are the eigenvectors of the covariance matrix of the data, which is always symmetric and positively definite. It has been noted that for most real data sets, most variance exists in a small fraction of the higher principal components. Thus, by projecting the data into the first few principal components, most of the variance in the data can be preserved.

SVD is typically used to implement PCA. When the matrix is decomposed into three matrices U, Σ and V^T, the dimensionality can be reduced by multiplying the original matrix with some components of these matrices. When V^T is used for column vectors, SVD achieves the same results as PCA. Intuitively, U and V^T can be regarded as rotation operations for the original matrix, and Σ as scaling operations.

The advantage of this dimensionality reduction method is that the spatio-temporal correlation can be simultaneously captured by properly specifying the covariance matrix structure. However, its main disadvantage is that separating subspaces spanned by higher principal components from those spanned by lower principal components is usually arbitrary, and the results are sensitive to the decision [36].

D. Procedures for Traffic Anomaly Detection Based on S-BiSVD

This work proposes a method based on the S-BiSVD algorithm to quickly detect anomalies in traffic networks and reduce the time spent on dimensionality reduction when processing real-time updated data.

Previous studies on subspace-based methods have concluded that S-BiSVD has high accuracy in anomaly detection and achieves continuous subspace tracking by monitoring and updating new data. However, taking the covariance matrix suffers from low speed.

Unlike the traditional subspace-based methods, the proposed method significantly improves the processing speed when executing dimensionality reduction. This is because it exploits the dimensionality reduction results of historical data and does not need a complex matrix multiplication to compute the covariance matrix, achieving fast and accurate dimensionality reduction every time the data is updated.

Some important notations and descriptions that will be used in this paper are reported in Table II. In this paper, the rows of the data matrix represent the nodes, and the columns are the time bins. The proposed methodology for anomaly detection is as follows:

(1) Select the values of parameter r, k, γ and θ. The size of a sliding window r is the number of time points contained in a sliding window. The step size of a sliding window k is related to how many time points are updated in each step. The dimension of the principal components γ is decided by the number of changing traffic flow patterns in the whole network. The threshold θ represents the upper limit of the L2 norm of each node, used to tell anomalies and normal points. The choice of θ is highly subjective, often based on the performance of normal points and also influenced by the value of r. Samples that exceed this threshold are defined as anomalies. When the threshold is set too high, some anomalous samples may go undetected, increasing false negatives detections. However, minor local anomalies may be incorrectly identified as significant when the threshold is too low, resulting in false positives. In such cases, the accuracy of S-BiSVD decreases.

(2) Perform S-BiSVD on the original data spanning time range r after centralizing each column data, and then obtain the three auxiliary matrices U, Σ and V^T. If it is the first step of the whole process, the bi-orthogonal iteration SVD algorithm should be used.

(3) Use the decomposed matrices, in which the first γ dimensions are deleted, to project original data on the residual subspace and then reconstruct them to the original space. S-BiSVD can reduce the dimensionality of a row, column or the whole matrix. Here we operated only the columns of the matrix, i.e., the time dimension of the dataset.

(4) Calculate the difference between the original and reconstructed data, and calculate the L2 norm of each row. If the L2 norm of a point exceeds θ, this point is considered as a candidate anomalous point.

(5) Repeat steps (2)-(4) to make the window matrix slide automatically with continuously updated data.

TABLE II
IMPORTANT NOTATION THAT WILL BE USED IN THIS PAPER

Notation	Description
l_n	A node number where $n \in N$
t_n	A time point where $n \in N$
$L(t)$	All the data collected until time t where $t \in t_n$
k	Step size of a sliding window
r	Size of a sliding window
Γ	Dimension of principal components
θ	A threshold to distinguish anomalies and normal points
$Lt(t)$	Data matrix to be processed in a sliding window at time t where $t \in t_n$
$U(t), \Sigma(t)$ and $V^T(t)$	Three output auxiliary matrices at time t where $t \in t_n$
$V^{'T}(t)_{r\times(r-\gamma)}$ and $V^{'}(t)_{r-\gamma}\times r$	The matrices used for reconstruction and the latter is the transposition of the former
$Lt'(t)$	Data projected on the residual subspace at time t where $t \in t_n$
σ(t)	L2 norm of difference between $Lt(t)$ and $Lt'(t)$ calculated by row
X_i	A data sample
μ	Symbolizes the mean of all samples
N	Denotes the total number of samples, and the constant
λ	A significant impact on the selection of candidate anomalies

The proposed method involves two parameters that have strong influences on anomaly detection. Indeed, γ determines the division of the normal and anomalous space, and θ affects the recognition of candidate anomalous points.

Example of the proposed approach: Next, we demonstrate the efficiency of the proposed S-BiSVD-based method for anomaly detection. Suppose matrix $L(t)$ contains all data until time t. Each row in the matrix represents the traffic counts of a certain node over time, and each column represents the traffic counts of different nodes over a specific period. So, the number of columns in $L(t)$ is increasing over time. Meanwhile, choosing a longer time interval would reduce the frequency of anomaly detection, but it might result in slower notifications when anomalies occur. On the other hand, opting for a very short interval would lead to unnecessary wastage of computational resources due to the limited traffic variations within a short duration.

For example, the $L(8)$ presented in Table III contains 5 nodes and 8 time points, highlighting an anomalous behavior from t_6, since compared with their past counts, the traffic counts of l_2 drop suddenly and that of l_3 rapidly increase. The data describes a scenario: Node l_2 has a traffic accident at t_6, resulting in a long-term road closure. Node l_3 is around and close to node l_2, and it carries many cars which should have been on l_2 if there were no road closures on l_2.

TABLE III
THE CONTENT OF L(8)

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
l_1	0	5	15	5	0	5	15	5
l_2	6	11	18	12	5	0	0	0
l_3	3	7	14	6	2	17	25	18
l_4	0	5	9	4	2	6	11	7
l_5	9	12	15	10	7	10	14	11

Step 1: For this example, we have the parameters r =5, k =1, γ=1 and θ=4.

Step 2: S-BiSVD is used for matrix factorization. Note that when processing streaming data, each step in the whole process must depend on the three matrices produced in the previous step. However, we do not have these matrices as input in the first step, so we use bi-orthogonal iteration SVD to generate an initial set of matrices that serve as the S-BiSVD launcher. Then S-BiSVD can be used in later steps.

Suppose we are at t_6 the bi-orthogonal iteration SVD algorithm has already generated $U(5)$, $\Sigma(5)$ and $V^T(5)$ at t_5. We can easily use these three matrices and the sliding window data at $t_6(Lt(6))$ as inputs to perform S-BiSVD. Note that centralization of data is necessary. The output matrices are shown as $U(6)$, $\Sigma(6)$ and $V^T(6)$.

$$U(6) = \begin{bmatrix} 0.37 & 0.29 & -0.62 & 0.47 & -0.44 \\ 0.43 & 0.75 & 0.44 & -0.23 & -0.06 \\ -0.53 & 0.56 & -0.45 & -0.18 & 0.43 \\ 0.15 & 0.06 & 0.23 & 0.74 & 0.62 \\ 0.63 & -0.21 & -0.40 & -0.39 & 0.49 \end{bmatrix} \quad (2)$$

$$\Sigma(6) = \begin{bmatrix} -11.71 & 1.04 & 0.22 & -0.11 & 0.09 \\ 0 & 4.94 & -0.24 & 0.23 & -0.21 \\ 0 & 0 & 3.62 & -0.65 & 0.55 \\ 0 & 0 & 0 & -1.27 & 0.89 \\ 0 & 0 & 0 & 0 & 0.53 \end{bmatrix} \quad (3)$$

$$V^T(6) = \begin{bmatrix} 0.53 & 0.40 & 0.49 & 0.56 & 0.07 \\ -0.41 & -0.03 & 0.53 & 0.04 & -0.74 \\ 0.08 & -0.63 & 0.62 & -0.22 & 0.41 \\ -0.18 & -0.55 & -0.25 & 0.78 & 0 \\ 0.71 & -0.38 & -0.20 & -0.17 & -0.53 \end{bmatrix} \quad (4)$$

Step 3: Now we reduce the dimensionality of the original data Lt(6) by projecting it on the subspace spanned by the lowest $r-\gamma$ principal components.

The relationship among the reconstructed data matrix $t'(6)$, the original data matrix Lt(6), and the matrices $U(6)$, $\Sigma(6)$ and $V^T(6)$ is presented below. Since $r-\gamma=4$, the shape of $V'^T(6)$ is 5×4 and $V'^T(6)$ comprises the last 4 columns of $V^T(6)$.

$$Lt'(6)_{5\times5} = Lt(6)_{5\times5} \times V'^T(6)_{5\times4} \times V'(6)_{4\times5} \quad (5)$$

Step 4: It calculates the difference between the original and reconstructed data and the L2 norm of each row. $\delta(6)$ shows that the second and third nodes have very high L2 norms, higher than θ we set before. Thus, the technique correctly identifies node 2 and node 3 as anomalies at t_6.

$$\delta(6) = \left\| Lt(6) - Lt'(6) \right\| = \begin{bmatrix} 3.57 & 4.61 & 6.19 & 0.84 & 2.84 \end{bmatrix} \quad (6)$$

Step 5: Steps (2)-(4) have already completed the anomaly detection at t_6, and this step is mainly to realize the sliding of the window. At t_7, after the new data is updated, the new sliding window data $Lt(7)$ is produced. With the update of traffic flow data, steps (2)-(5) are repeated continuously. Real-time anomaly detection is achieved due to the fast matrix decomposition of S-BiSVD.

Fig. 1 depicts all detection results of data $L(8)$, where the dotted line shows the threshold θ. Since the L2 norm of l_3 is higher than θ, we can confirm that an anomaly appears from t_6.

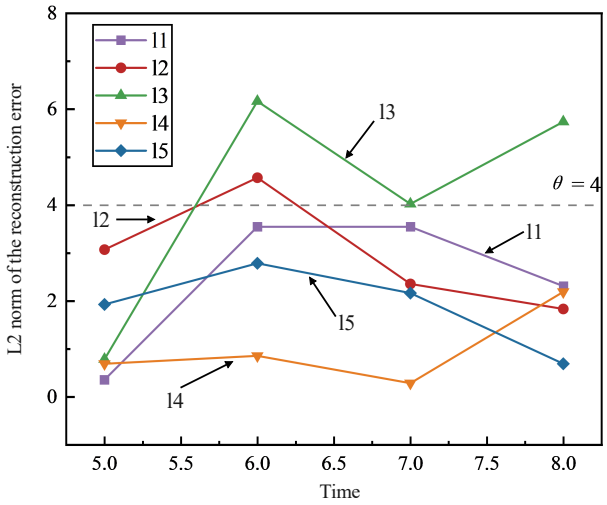

Fig. 1. The L2 norm of the reconstruction error for each node.

IV. SIMULATION EXPERIMENTS

This section conducts simulation experiments to evaluate our proposed method. All methods are implemented in Python3.8 and executed on a server with an Intel Xeon(R) Platinum 8370 CPU and RTX 3090 24G GPU.

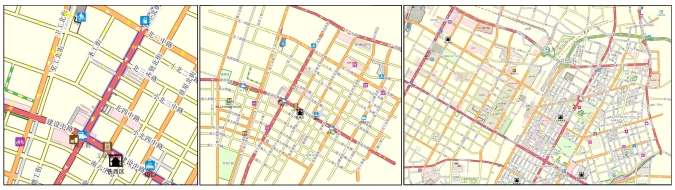

Fig. 2. The simulated road network of different scales in Shenyang. (a) Network A with 176 segments. (b) Network B with 1004 segments. (c) Network C with 9840 segments.

A. Datasets

Based on the map of Shenyang, we constructed three different scale road networks using SUMO [37], named Network A, Network B, and Network C. The road network A, B, and C comprises 176, 1004, and 9840 road segments and 58, 377, and 3980 intersections, respectively, as illustrated in Fig. 2. The dataset is generated as follows. First, through video surveillance, we collected traffic data on the arterial roads of this area. Next, we inferred the OD demands within the region using TransCAD based on the traffic data. Then, these OD demands were employed to configure the simulated road network and further simulate the traffic flows. Finally, we injected anomalous traffic situations. Specifically, we randomly cut off a small number of arterial road segments in some intervals by reducing their traffic capacity to 10%, which resulted in a sharp traffic drop in these segments and a rise in their alternatives, as depicted in Fig. 3. The proportion of anomalous segments in each network is 20%.

The time interval is set to 15 minutes, and we investigate the performance variation of our method for different sliding window sizes r.

B. Evaluation Metrics

The effectiveness of S-BiSVD is evaluated based on the following performance metrics.

(1) Precision, which measures the probability of authentic anomalies in the samples detected as anomalies.

(2) Recall, which is the probability that the sample anomalies are accurately detected.

(3) F1-score, which is the combination or balance point of precision and recall. It is often used as a comprehensive performance metric. The performance metrics presented above are mathematically defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

C. Baseline

Considering their effectiveness and efficiency, S-BiSVD is challenged against the following baseline methods for anomaly detection. The rationale for choosing these algorithms as baselines is that they belong to unsupervised learning categories like S-BiSVD.

(1) PCA [20]: PCA is used similarly to S-BiSVD. We first use PCA to reduce the dimensionality and then reconstruct the data. Anomaly detection is performed by calculating the difference between the original and reconstructed data to identify samples that are difficult to reconstruct.

(2) I-Forest [38]: It is a fast outlier detection method based on ensemble, which has linear time complexity and high accuracy.

(3) LOF [39]: This is a density-based outlier detection methods.

(4) KMeans [40]: This is a method of discovering sample outliers through clustering.

(5) SOS [41]: This is an anomaly detection method that measures the degree of correlation between observation points and other points.

(6) COF [42]: It is a variant of LOF.

(7) OCSVM [43]: It is anomaly detection algorithm based on SVM.

D. Impact of Parameter Threshold

Appropriately setting the threshold parameter is important for anomaly detection. This works adopt the following threshold:

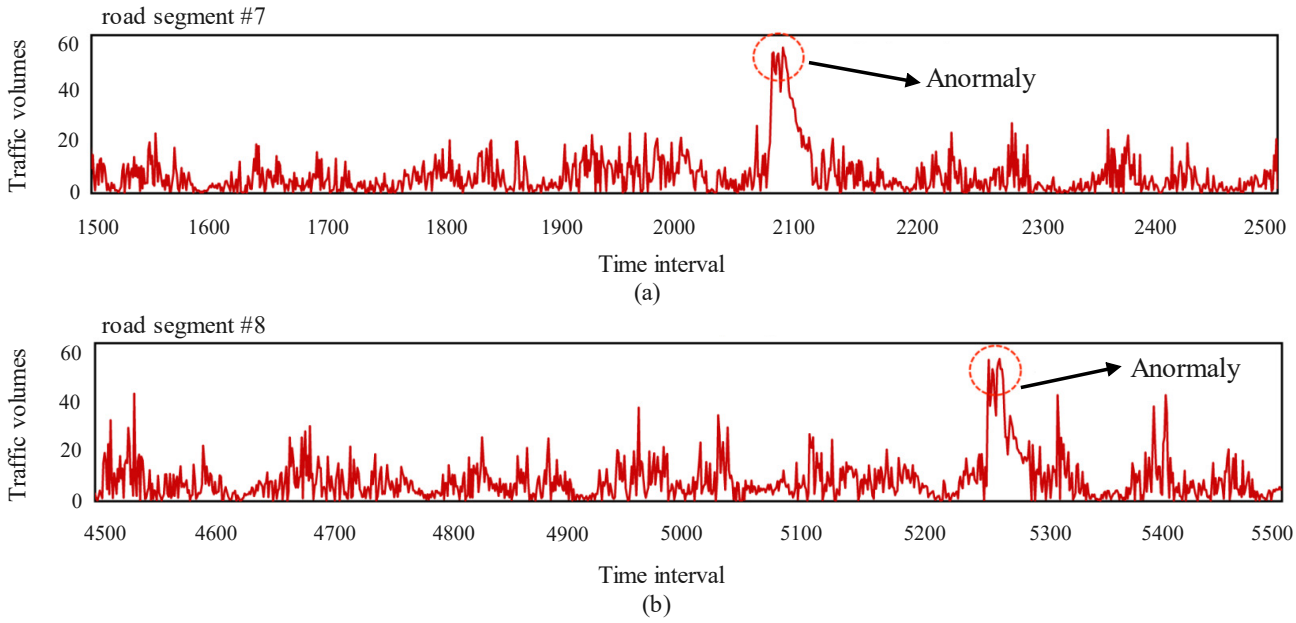

Fig. 3. Traffic flow over time for two segments. (a) segment #7 has an anomaly at interval 2100. (b) segment #8 has an anomaly at interval 5250.

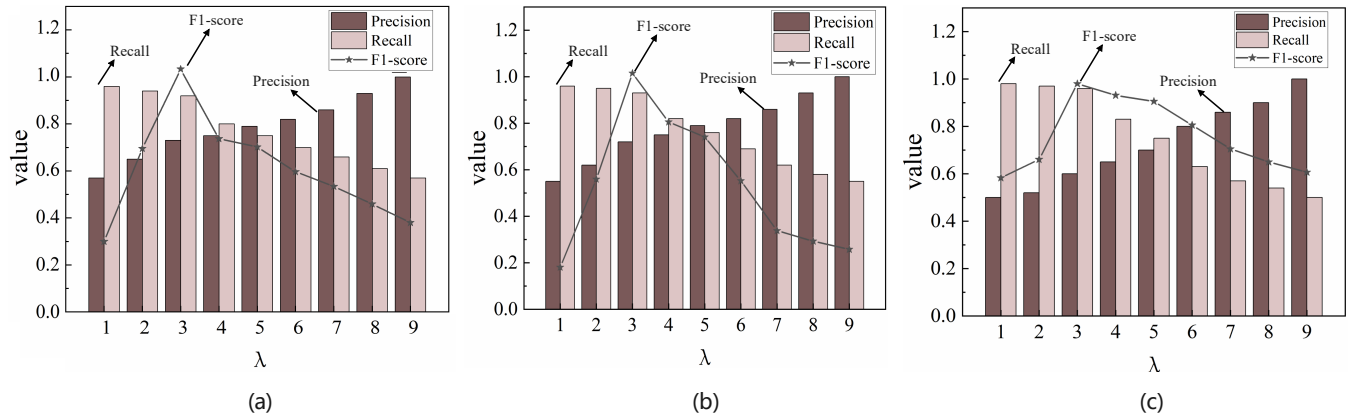

Fig. 4. The impact of coefficient lamda on performance metrics. (a) Network A. (b) Network B. (c) Network C.

$$\theta = \mu + \lambda \sqrt{\frac{\sum_{i=1}^{N} (X_i - \mu)^2}{N}} \tag{10}$$

where X_i represents a data sample, μ denotes the mean of all samples, N denotes the total number of samples, and the coefficient λ has a significant impact on the selection of candidate anomalies. Any sample with a large component projected on the anomaly subspace can be considered as an anomaly. The experimental results are presented in Fig. 4, where we gradually increased the coefficient λ from 1 to 9. The results indicate that as the coefficient increases, the precision of S-BiSVD shows an uptrend and the recall rate downtrend. Considering these indicators, a threshold of 3 is most suitable for S-BiSVD.

E. Efficiency

We compare the proposed method against PCA, I-Forest, LOF, KMeans, SOS, COF, and the OCSVM algorithms. We compare the runing time of S-BiSVD with the baseline, and the corresponding results are reported in the Table IV. According to Table IV, S-BiSVD demonstrates superior detection efficiency, achieving an average improvement of 3, 30, 40 and 43 times compared to the PCA, I-Forest, LOF, KMeans, SOS, COF, and the OCSVM, respectively. When dealing with large-scale networks, the improvement in efficiency is more significant. The results demonstrate that S-BiSVD is very efficient in processing large-scale streaming data.

TABLE IV
COMPARISON OF EXECUTION TIMES OF THE DIFFERENT METHODS FOR THREE DATASETS

	Network A	Network B	Network C
S-BiSVD	0.0323	0.1931	78.5621
PCA	0.0969	0.6356	234.6549
I-Forest	0.3913	6.6446	246.2561
LOF	0.2558	5.4518	529.5984
KMeans	0.5864	8.3514	365.2357
SOS	0.5745	8.2398	364.5864
COF	0.2213	5.2521	528.3568
OCSVM	0.5927	8.5102	421.1547

F. Performance Analysis of Anomaly Detection

This section compares S-BiSVD with the baselines to evaluate its effectiveness in anomaly detection. The cor-

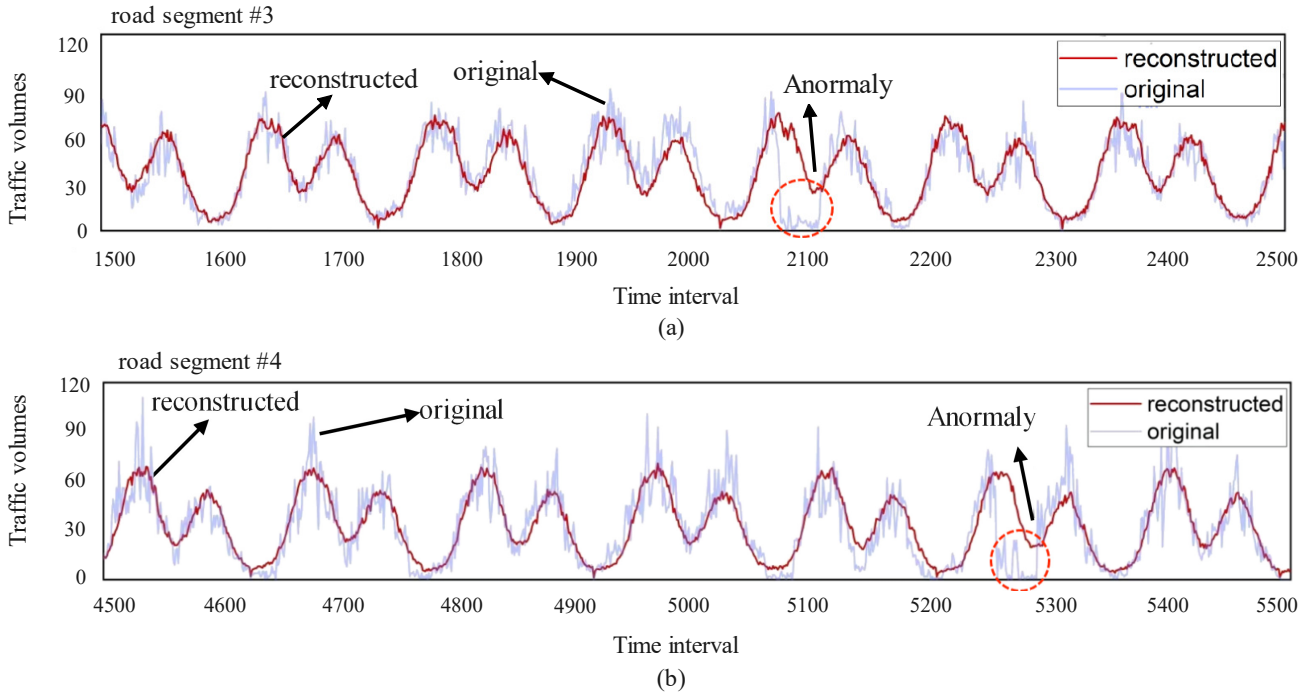

Fig. 5. The original traffic data and the traffic data reconstructed by S-BiSVD for #3 and #4 road segment.

responding results are reported in Table V, revealing that the precision of S-BiSVD is 85.47%, 82.68%, and 79.23%, respectively, and its performance is comparable to PCA in the three datasets. Taking Network C as an example, S-BiSVD outperforms I-Forest, LOF, KMeans, SOS, COF, and the OCSVM. Compared with recall, S-BiSVD is the best, affording 89.21%, and its precision of 79.23% is the best among all methods. Regarding the F1-score, S-BiSVD outperforms all competitor approaches significantly. Fig. 5 compares the original traffic data with the traffic data reconstructed by S-BiSVD for two representative segments. For an anomaly, there is a large deviation between the original value and its reconstructed value.

TABLE V
RESULTS OF COMPARATIVE EXPERIMENTS

		S-BiSVD	PCA	I-Forest	LOF
Network A	Precision(%)	85.47	86.34	71.96	75.15
	Recall(%)	93.25	94.12	81.64	84.34
	F1-score(%)	89.19	90.06	76.49	79.48
Network B	Precision(%)	82.68	82.70	64.16	63.82
	Recall(%)	90.88	90.92	75.18	91.33
	F1-score(%)	86.59	86.62	69.23	75.14
Network C	Precision(%)	79.23	79.46	61.11	60.96
	Recall(%)	89.21	90.00	74.18	80.48
	F1-score(%)	83.92	84.40	67.01	69.37
		KMeans	SOS	COF	OCSVM
Network A	Precision(%)	72.56	71.54	76.00	72.43
	Recall(%)	83.95	83.12	84.79	82.12
	F1-score(%)	77.84	76.90	80.15	76.97
Network B	Precision(%)	67.10	68.17	63.90	66.70
	Recall(%)	80.00	79.34	92.00	79.89
	F1-score(%)	72.98	73.33	75.42	72.70
Network C	Precision(%)	60.56	61.24	61.21	61.75
	Recall(%)	78.98	78.32	78.36	77.90
	F1-score(%)	68.55	68.73	68.73	68.89

Next, to further evaluate the performance of S-BiSVD, we adjusted the proportion of injected anomalies. Specifically, we set the proportion of anomalous segments in Network C to 10%, 20%, and 30%, respectively, to simulate the uncertainty of the road network.

Fig. 6 depicts the results of anomaly detection using different methods in datasets with different proportions of anomalous segments. It can be seen that S-BiSVD and PCA perform similarly and significantly better than the other methods. As the proportion of anomalies increases, the performance of all methods decreases. However, the decreases of S-BiSVD and PCA are relatively small.

G. Sensitivity Analysis of the Window Size

We tested the effect of different window sizes on S-BiSVD. Fig. 7 shows the performance of S-BiSVD and its variants on all datasets. As the window decreases, long time correlation information is lost, resulting in lower recall and F1-scores. On the contrary, as the window size increases, the model takes into account longer temporal correlation information and ignores local patterns, resulting in a slight decrease in precision and F1-score. In our experiments, when the window size is set to 12, F1-score reaches its highest value.

V. CONCLUSION

This paper proposes an anomaly detection method based on the sequence bi-iteration SVD algorithm (S-BiSVD), a streaming algorithm that can quickly learn the subspace of traffic and be used for real-time anomaly detection. Our method captures anomalies effectively by processing newly added data columns instead of operating on the entire data matrix. We have conducted extensive experimental comparisons of our proposed method with the baseline method in terms of detection speed and accuracy, and the results show that our method is highly efficient in anomaly detection, and

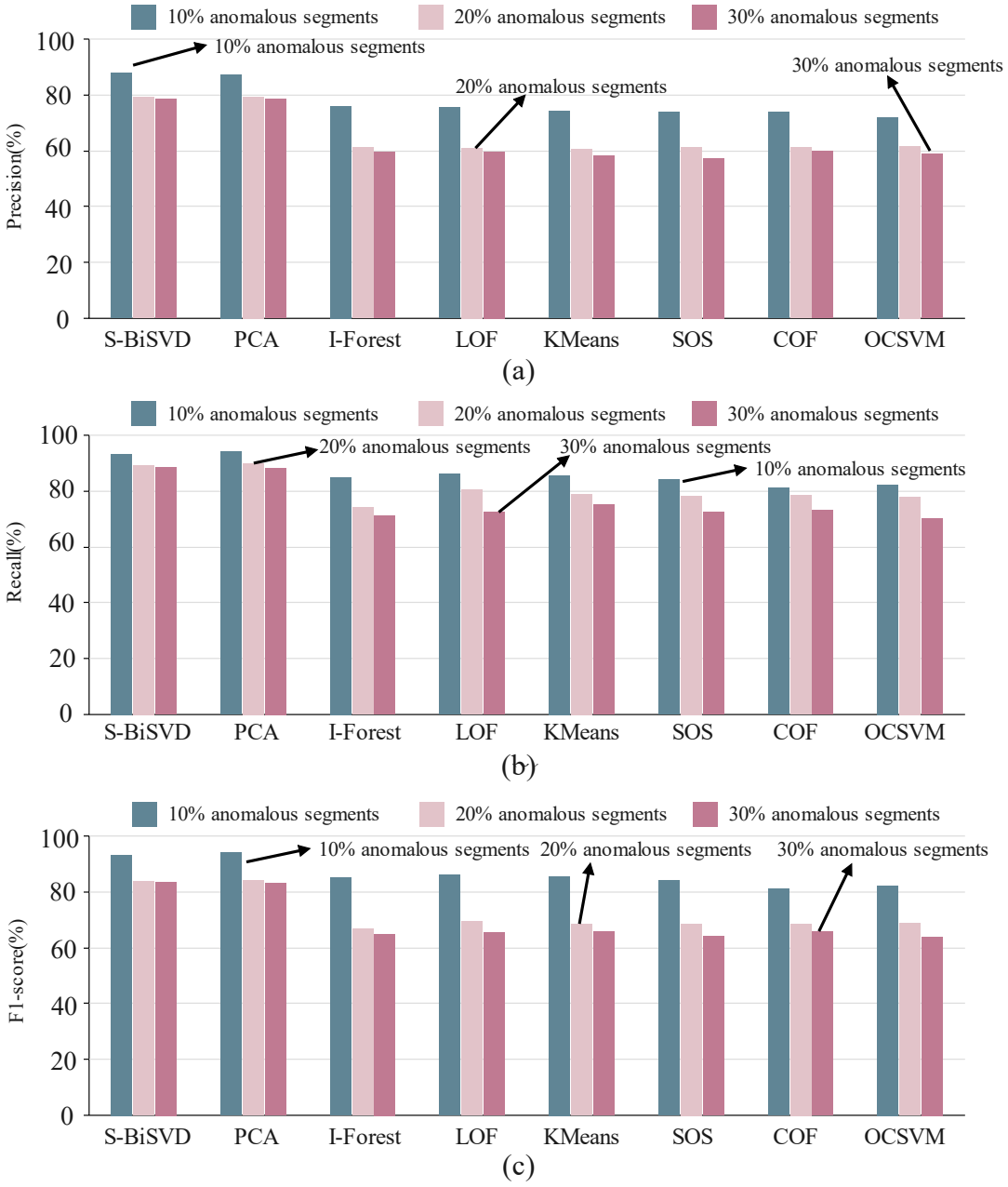

Fig. 6. Comparison of different metrics for each method on datasets with different proportions of anomalies. (a) Precision. (b) Recall. (c) F1-score.

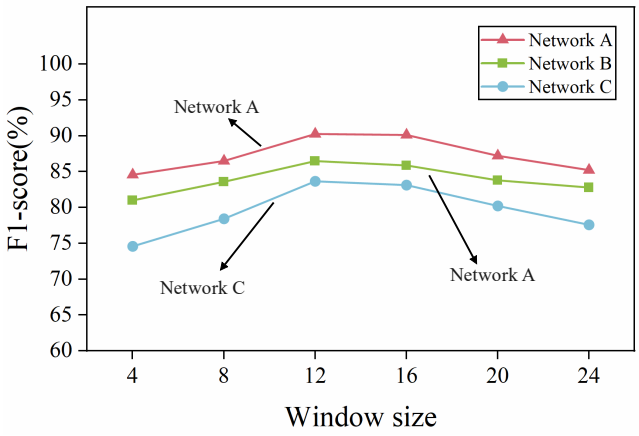

Fig. 7. F1-score with window size.

also provides a significant improvement over the baseline method in processing streaming data.

In future work, the proposed method will be applied to other networks with massive real-time data streams, such as IP networks and sensor networks.

REFERENCES

[1] Chawla, Sanjay, Yu Zheng, and Jiafeng Hu, "Inferring the Root Cause in Road Traffic Anomalies," 2012 IEEE 12th International Conference on Data Mining, Brussels, Belgium, IEEE, pp141-150, 2012

[2] Song Wang, Juan Fernando Balarezo Serrano, Kandeepan Sithamparanathan, Akram Hourani, Karina Mabell Gomez Chavez, and Benjamin Rubinstein, "Machine Learning in Network Anomaly Detection: A Survey," IEEE Access, vol. 9, pp152379-152396, 2021

[3] Habeeb, Riyaz Ahamed Ariyaluran, et al, "Real-time big data processing for anomaly detection: A Survey," International Journal of Information Management, vol. 45, pp289-307, 2019

[4] Ji Zhang, Hongzhou Li, Qigang Gao, Hai Wang, and Yonglong Luo, "Detecting anomalies from big network traffic data using an adaptive detection approach," Information Sciences, vol. 318, pp91-110, 2015

[5] Mohammed S. Hadi, Ahmed Q. Lawey, Taisir E.H. El-Gorashi, and Jaafar M.H. Elmirghani, "Big data analytics for wireless and wired network design: a survey," Computer networks, vol. 132, pp180-199, 2018

[6] Junaid Qadir, Nauman Ahad, Erum Mushtaq, and Muhammad Bilal, "SDNs, Clouds, and Big Data: New Opportunities," 2014 12th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, IEEE, pp28-33, 2014

[7] Habeeb, Riyaz Ahamed Ariyaluran, et al, "Real-time big data processing for anomaly detection: A survey," International Journal of Information Management, vol. 45, pp289-307, 2019

[8] Ming Xu, Jianping Wu, Haohan Wang, and Mengxin Cao, "Anomaly detection in road networks using sliding-window tensor factorization," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 12, pp4704-4713, 2019

[9] Kuen-Fang Jea and Chao-Wei Li, "A sliding-window based adaptive approximating method to discover recent frequent itemsets from data streams," Lecture Notes in Engineering and Computer Science, vol. 2180, no. 1, pp532-539, 2010

[10] Wen Dong and Alex Pentland, "A network analysis of road traffic with vehicle tracking data,"AAAI Spring Symposium: Human Behavior Modeling, pp7-12, 2009

[11] Bei Pan, Yu Zheng, David Wilkie, and Cyrus Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp344-353, 2013

[12] Linsey Xiaolin Pang, Sanjay Chawl, Wei Liu, and Yu Zheng, "On detection of emerging anomalous traffic patterns using gps data," Data & Knowledge Engineering, vol. 87, pp357–373, 2013

[13] Mingxi Wu, Xiuyao Song, Chris Jermaine, Sanjay Ranka, and John Gums, "A LRT framework for fast spatial anomaly detection," Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp887-896, 2009

[14] Yu Zheng, Huichu Zhang, and Yong Yu, "Detecting collective anomalies from multiple spatio-temporal datasets across different domains," Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp1-10, 2015

[15] Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie, "Urban computing with taxicabs," Ubiquitous Computing, pp89-98, 2011

[16] Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xing Xie, "Discovering spatio-temporal causal interactions in traffic data streams," Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp1010-1018, 2011

[17] Ming Xu, Jianping Wu, Mengqi Liu, Yunpeng Xiao, Haohan Wang, and Dongmei Hu, "Discovery of critical nodes in road networks through mining from vehicle trajectories," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 2, pp583-593, 2018

[18] Yuhan Jia, Jianping Wu, and Ming Xu, "Traffic Flow Prediction with Rainfall Impact Using a Deep Learning Method," Journal of Advanced Transportation, pp1-10, 2017

[19] Anukool Lakhina, Mark Crovella, and Christophe Diot, "Diagnosing network-wide traffic anomalies," ACM SIGCOMM Computer Communication Review, vol. 34, no. 4, pp219-230, 2004

[20] Ling Huang, Xuanlong Nguyen, Minos Garofalakis, Michael I. Jordan, Anthony Joseph, and Nina Taft, "In-Network PCA and Anomaly Detection," Advances in Neural Information Processing Systems, vol. 19, 2006

[21] Taylor, Adrian, Sylvain Leblanc, and Nathalie Japkowicz, "Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks," 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, IEEE, pp130-139, 2016

[22] Mohammad Ahamdi Livani and Mahdi Abadi, "Distributed PCA-based anomaly detection in wireless sensor networks," In 2010 International Conference for Internet Technology and Secured Transactions, IEEE, pp1-8, 2010

[23] Miao xie, Song Han, and Biming Tian, "Highly Efficient Distance-Based Anomaly Detection through Univariate with PCA in Wireless Sensor Networks," 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, pp564-571, 2011

[24] Zhiguo Ding, Dajun Du, and Minrui Fei, "An Isolation Principle Based Distributed Anomaly Detection Method in Wireless Sensor Networks," International Journal of Automation and Computing, vol. 12, no. 4, pp402-412, 2015

[25] Guilan Wang, Guoliang Zhou, Hongshan Zhao, and Zengqiang Mi, "Fast clustering and anomaly detection technique for large-scale power data stream," Automation of Electric Power Systems, vol. 40, no. 24, pp27-33, 2016

[26] Tin Mar Myint and Khin Thidar Lynn, "Handling the Concept Drifts Based on Ensemble Learning with Adaptive Windows," IAENG International Journal of Computer Science, vol. 48, no. 3, pp471-486, 2021

[27] Peter Strobach, "Low-rank adaptive filters," IEEE Transactions on Signal Processing, vol. 44, no. 12, pp2932-2947, 1996

[28] Peter Strobach, "Bi-iteration SVD subspace tracking algorithms," IEEE Transactions on Signal Processing, vol. 45, no. 5, pp1222-1240, 1997

[29] Roland Badeau, Gael Richard, and Bertrand David, "Sliding window adaptive SVD algorithms," IEEE Transactions on Signal Processing, vol. 52, no. 1, pp1-10, 2004

[30] Bin Yang, "Projection approximation subspace tracking," IEEE Transactions on Signal Processing, vol. 43, no. 1, pp95-107, 1995

[31] Roland Badeau, Karim Abed-Meraim, Gael Richard, and Bertrand David, "Sliding window orthonormal PAST algorithm," 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, China, ppV-261, 2003

[32] Ralph Otto Schmidt, "A signal subspace approach to multiple emitter location and spectral estimation," Stanford University, 1982

[33] E.C. Real, D.W. Tufts, and J.W. Cooley, "Two algorithms for fast approximate subspace tracking," IEEE Transactions on Signal Processing, vol. 47, no. 7, pp1936-1945, 1999

[34] Flip Korn, H. V. Jagadish, and Christos Faloutsos, "Efficiently supporting ad hoc queries in large datasets of time sequences," ACM SIGMOD Record, vol. 26, no. 2, pp289-300, 1997

[35] Pangning Tan, Michael Steinbach, and Vipin Kumar, "Introduction to Data Mining: Pearson New International Edition," English Edition, 2013

[36] Daniela Brauckhoff, Kavé Salamatian, and Martin May, "Applying PCA for Traffic Anomaly Detection: Problems and Solutions," IEEE INFOCOM 2009, IEEE, pp2866-2870, 2009

[37] Pablo Alvarez Lope, Michael Behrisc, Laura Bieker-Walz, Jakob Erdmann, et al, "Microscopic Traffic Simulation using SUMO," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp2575-2582, 2018

[38] Fei Tony Liu, Kaiming Ting, and Zhihua Zhou, "Isolation Forest," 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, IEEE, pp413-422, 2008

[39] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander, "LOF: identifying density-based local outliers," Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp93-104, 2000

[40] John. A. Hartingan and Manchek. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," Journal of the Royal Statistical Society. series c (applied statistics), vol. 28, no. 1, pp100-108, 1979

[41] Jeroen H.M. Janssens, Eric O. Postma, and Jaap H.J. van den Herik, "Maritime anomaly detection using stochastic outlier selection," MAD 2011 Workshop Proceedings, pp121, 2011

[42] Agnieszka Nowak-Brzezińska and Czesław Horyń, "Outliers in rules-the comparision of LOF, COF and KMEANS algorithms," Procedia Computer Science, vol. 176, pp1420-1429, 2020

[43] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Computation, vol. 13, no. 7, pp1443-1471, 2001

Ming Xu received his Ph.D. degree in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 2015. From 2016 to 2019, he was a Post-Doctoral Fellow with Tsinghua University, Beijing, China. He is currently a professor with the software college, Liaoning Technical University. He has published over 20 papers in journals and conferences, including TITS\BIBM\TCSS. His research work has reported by MIT Technology Reviews. He is the recipient of "the 2020 World Artificial Intelligence Conference Youth Outstanding Paper" (WAICYOP) award. His research interests include: graph learning, spatio-temporal data mining, reinforcement learning and data-driven traffic simulation.

Jiani Li is a postgraduate student of School of Software, Liaoning Technical University, Huludao, China. Her reasearch interests include: graph deep learning, spatio-temporal data mining, anomaly detection.