Neurocomputing 501 (2022) 596–608

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An extreme learning machine for unsupervised online anomaly detection in multivariate time series

Xinggan Peng [a], Hanhui Li [b,*], Feng Yuan [c], Sirajudeen Gulam Razul [c], Zhebin Chen [a], Zhiping Lin [a]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
[b] School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen 518107, China
[c] Temasek Laboratories, Nanyang Technological University, 637553, Singapore

ARTICLE INFO

Article history:
Received 3 September 2021
Revised 18 May 2022
Accepted 11 June 2022
Available online 20 June 2022

Keywords:
Online anomaly detection
Extreme learning machine
Unsupervised learning
Multivariate time series
Kernel selection

ABSTRACT

Unsupervised anomaly detection in time series remains challenging, due to the rare and complex patterns of anomalous data. Previous change point detection methods based on extreme learning machine and mutual information (ELM-MI) are potential solutions for this problem. However, the kernels in these methods are randomly initialized on test data, which imposes a constraint that these methods can only be used for offline inference. Moreover, these methods are limited in utilizing temporal contexts, and require the ensemble of multiple models to improve the robustness. To tackle these problems, we introduce a multivariate ELM-MI framework, and combine it with a dynamic kernel selection method, which performs a hierarchical clustering procedure on unlabeled training data and utilizes the clusters to determine the kernels in ELM-MI. In this way, our method can tackle the unsupervised online detection of various anomalous (e.g., point anomalies and group anomalies) and reduce the computational cost. Extensive experiments on three public datasets and our collection of real-life 4G Long-Term Evolution data demonstrate that the proposed method outperforms state-of-the-art methods in terms of effectiveness and efficiency. For demo, see this link: https://personal.ntu.edu.sg/ezplin/NC-demo.htm.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Anomaly detection is an important research topic in events detection [1,2], time-series signal processing [3,4] and computer vision [5,6] etc. In the last decade, extensive methods have been proposed to tackle anomaly detection, from the early distance and statistics based methods [7] to the currently popular deep learning methods [8]. Due to the extremely imbalanced class distribution and the cumbersome labeling cost, unsupervised anomaly detection becomes a major but difficult objective for existing methods. Particularly, the lack of prior knowledge, low recall rate, complex anomalies, and high dimensional data are the typical challenging factors in unsupervised anomaly detection [8]. In addition, the computational resources (e.g., hardware requirements, training time, and inference latency) might be restricted, which further limits the progress in unsupervised anomaly detection.

In this paper, we aim at tackling unsupervised anomaly detection by introducing a novel scheme, which is inspired by the extreme learning machine with mutual information estimation (ELM-MI). The original ELM-MI method [9] was proposed for detecting change points in time-series data. It has several prominent advantages, such as an efficient architecture (only one hidden layer), an explainable closed-form expression for learnable weights, and low training cost. These advantages make ELM-MI potentially feasible for real-life applications.

However, directly applying ELM-MI into anomaly detection is impractical, due to the following three reasons. (i) Change points are merely a potential type of anomalies [7], as demonstrated in Fig. 1. In fact, whether a detected change point is anomalous shall be determined by the specific context. For instance, daily internet traffic peaks are quite normal. Furthermore, long-term flat data that do not contain change points, e.g., a plateau of internet data usage in non-peak hours, can be the sign of group anomaly as well. (ii) One of the key components in ELM-MI is the hidden layer, which is composed of the radial basis function (RBF) kernels. However, the parameters governing the RBF kernels are randomly initialized on the test data. Consequently, the original ELM-MI can only be applied to offline inference, whereas the detection/response time is critical to many real-life applications. (iii) Moreover, as the RBF kernels are randomly initialized, it is difficult for ELM-MI to exploit prior knowledge and temporal contexts.

* Corresponding author.
E-mail address: lihh77@mail.sysu.edu.cn (H. Li).

https://doi.org/10.1016/j.neucom.2022.06.042
0925-2312/© 2022 Elsevier B.V. All rights reserved.

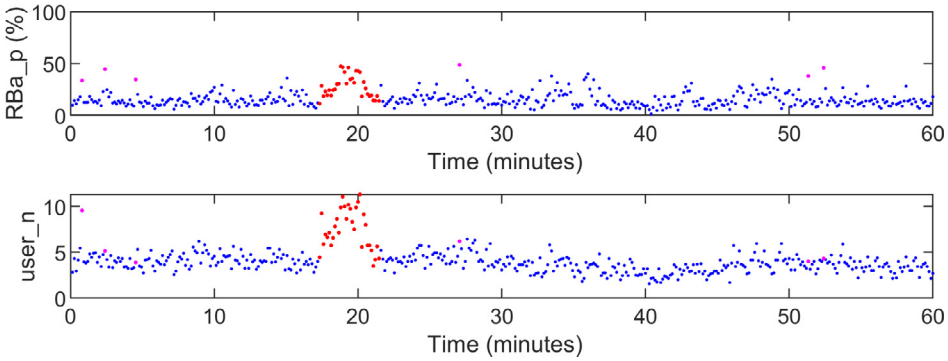


Fig. 1. Demonstration of the difference between change points and data anomalies. Two channels of 4G LTE data are reported in this figure, and we manually annotate a few change points and anomalous data. The normal data are marked in blue, change points are in magenta, and anomalous data are in red, respectively. Due to the dramatical changes in LTE data, considering change points as anomalous data is ineffective.

To overcome the above mentioned limitations, we propose to augment ELM-MI with a dynamic kernel selection (DKS) method for anomaly detection. The key idea of our DKS method is to determine the parameters of the RBF kernels adaptively for each test sequence. To this end, we first collect a training dataset, on which we conduct a hierarchical clustering procedure. Note that our training data are unlabeled and unlike previous one-class classification methods [10,11], we allow our training dataset to contain anomalous data. With the clustering procedure, training data with similar patterns tend to be grouped into the same cluster. Given a test sequence, we calculate its similarity to each cluster, which is used as the weight of sampling training data from the cluster. In other words, we prefer to sample training data that are similar to the test sequence, because they are more relevant and can better reflect the temporal context of the test sequence. Subsequently, the sampled training data are used to initialize the parameters of the RBF kernels, so that our improved ELM-MI can accomplish online anomaly detection.

To validate the effectiveness of the proposed method, we conduct extensive experiments on three public datasets including one SinSignal dataset [12], one gesture dataset [13] and the Skoltech anomaly benchmark (SKAB) [14], and apply the proposed method to a real-life application of 4G Long-Term Evolution (LTE) data analysis. Our results demonstrate that, the proposed method not only outperforms state-of-the-art methods in terms of multiple metrics, but also has the faster inference speed than the original ELM-MI.

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

- We introduce a novel framework for unsupervised online anomaly detection in multivariate time series, which is based on estimating the mutual information among test sequences via the extreme learning machine.
- We replace the random kernel initialization strategy of ELM-MI with the proposed dynamic kernel selection method, which overcomes the limitations of ELM-MI in online anomaly detection. Furthermore, the proposed method can better exploit temporal context to detect anomalies of various types.
- Extensive experiments on three public datasets and our collected 4G LTE dataset validate that the proposed framework is an effective and efficient solution for anomaly detection. In addition, a series of ablation studies demonstrate that the proposed method maintains robust with various experimental settings.

The remaining parts of this paper are organized as follows. Section 2 reviews existing methods on anomaly detection and ELM-MI. Section 3 describes the details of the proposed method. Section 4 describes the experiments for validating the proposed method, and we conclude this paper in Section 5.

2. Related work

In this section, we review previous methods that are closely related to the proposed method in this paper. We begin by introducing existing techniques for anomaly detection in Section 2.1, including traditional machine learning methods and deep learning methods. Recent advances in extreme learning machine (ELM) are summarized in Section 2.2 as well.

2.1. Anomaly detection

Anomaly detection is defined as tasks aiming to find patterns that do not comply with expected behavior [7]. Anomaly detection involving time-series data is a widely discussed research topic in artificial intelligence due to its various engineering applications [15–18]. Typically, these tasks can be handled via traditional machine learning methods and deep learning methods.

As discussed in prior works [7], classification based, density or distance estimation based, statistical, and information-theoretic methods are popular among traditional machine learning methods. To be specific, widely used classification based machine learning methods include support vector machine (SVM) [19–21] and one-class SVM [10,11], because of their robustness in constructing hyperplanes among samples. But they may be inefficient for high-dimensional data caused by computational scalability limitation [22]. Isolation forest [23], local outlier factor (LOF) [24], KNN [25], and tree decomposition [26] are typical examples of density or distance based anomaly detection algorithms. If appropriate distance measurements for the given data are provided, the process of applying such density or distance based anomaly detection algorithms is straightforward. However, anomalous samples can be dense or involve close neighbours, and consequently the detection performance of these methods will be affected. Statistics based anomaly detection algorithms assume that the occurrence probability of normal and anomalous samples conforms with specific statistical models, e.g., Gaussian processes [27]. But in real-world scenarios, especially for high dimensional cases, data are not generated from a particular distribution. As a result, the performance of traditional statistics based anomaly detection algorithms might be limited.

Deep learning methods [28–32] are current popular methods for anomaly detection. For example, deep support vector data description (Deep SVDD) [33,34] and Deep SVDD based on variational autoencoder (Deep SVDD-VAE) [35] aim to overcome the difficulties of traditional methods in selecting relevant features for complex data. Long short-term memory (LSTM) [36,37] based algo-

rithms have also been applied into anomaly detection. For example, a LSTM based method for detecting urban anomalies using LTE traffic data with labeled anomalous samples was proposed in [38]. Another Semi-Supervised algorithm based on LSTM-Autoencoder (LSTM-AE) was implemented to achieve mobile traffic anomaly detection with contextual anomalies based on a specific temporal context [39]. However, one of the common limitations of deep learning methods is that they need a large number of data samples and labels for training, which is impractical for many real-world applications. Moreover, compared with traditional machine learning methods, the training process of the deep learning based methods costs more time and usually requires powerful computational resources.

2.2. Extreme learning machine

ELM [40,9] is a feed forward neural network with a single hidden layer, and the weights between the input and the hidden layer are randomly initialized. ELM is originally proposed for classification and regression tasks because of its fast learning speed and good generalization performance [41,40], and it is closely related to the recent analytic learning methods [42–44]. Recently, the voting method incorporated into ELM to lower the variance among different realizations to achieve better classification results has been discussed in [45]. One class extreme learning machine (OC-ELM) has been introduced in [46], which outperforms many conventional one-class classifiers. A recent improvement of OC-ELM, which combines a multilayer neural network with an ELM classifier to better deal with complex and multi-class classification tasks, is presented in [47]. Besides classification, ELM extended with semi-supervised learning tasks have been presented in [48].

ELM has been exploited in various real-world applications. For example, a semi-supervised extreme learning machine (SS-ELM) is applied to driver distraction detection [49]. Landmark recognition tasks have also been completed by the ensemble of ELM [50,51]. ELM autoencoders have also been proposed to perform anomaly detection on aviation data [52] and FPGA-Based edge device detectors [53]. Additionally, the online sequential extreme learning machine (OSELM) has been proposed to chaotic time series and time-varying system prediction by cooperating with robust M-estimator-based cost function [54], and generalized regularization and adaptive forgetting factor [55], respectively. Moreover, ELM has been applied to medical/biomedical applications such as epileptic EEG patterns recognition [56], diagnosis of hepatitis [57] and protein–protein interaction prediction [58]. Unlike other ELM based methods that predict anomalies based on single data sequence (e.g., OC-ELM and ELM-autoencoder), ELM-MI adopts the mutual information to measure the similarity between the two sequences selected from time-series data at different time stamps. One sequence works as the target/test sequence, and the other works as the reference; hence ELM-MI can better handle conditional anomalies if the appropriate reference is provided. In addition, it is common practice in ELM-MI to select the sequence overlapped with (or adjacent to) the test one as the reference, which also helps to exploit the temporal context of the test sequence. However, to the best of our knowledge, ELM based methods have not been exploited to deal with anomaly detection of multivariate fast changing time-series data, such as 4G LTE data. Therefore, this paper explores the applicability of applying ELM into this topic.

3. Online ELM-MI with dynamic kernel selection

The details of the proposed method for unsupervised online anomaly detection are presented in this section. We first formulate the problem of anomaly detection within the ELM-MI framework in Section 3.1. We then introduce the core of our method, which is the DKS method for augmenting ELM-MI in Section 3.2.

3.1. ELM-MI for anomaly detection

Mutual information (MI) is a widely-used measure of the correlation between two random variables. To utilize the MI into anomaly detection, given a test sequence, we can sample another sequence as the reference and calculate the MI value between them. Since anomalous data are rare in real-life applications, most sampled references are normal. Consequently, the higher the MI value is, the more likely the test sequence is normal.

Specifically, given a time-series $\mathbf{x} \in \mathbb{R}^{C\times L}$ as the test sequence, where C is the number of channels and L is the length of sequence, we select a reference sequence $\mathbf{y} \in \mathbb{R}^{C\times L}$, and calculate the squared-loss MI [59] for $\mathbf{x}$ and $\mathbf{y}$ as follows:

$$I(\mathbf{x},\mathbf{y}) = \frac{1}{2}\int\int\left(\frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} - 1\right)^2 p(\mathbf{x})p(\mathbf{y})d\mathbf{x}d\mathbf{y}, \tag{1}$$

where $p(\mathbf{x})$ and $p(\mathbf{y})$ are the marginal probability of $\mathbf{x}$ and $\mathbf{y}$, and $p(\mathbf{x},\mathbf{y})$ is their joint probability. For the convenience of calculation and expression, we concatenate all the C channels of each sequence to represent our sequences as vectors. As the exact inference of Eq. (1) is difficult, Suzuki et al. [59] propose to introduce an auxiliary density ratio function $f(\mathbf{x},\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}$, and approximate $f(\mathbf{x},\mathbf{y})$ with the following linear combination of multiple kernels:

$$f(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{N}\beta_i k_i(\mathbf{x},\mathbf{y}), \tag{2}$$

where k_i is a kernel function, β_i is the weight of k_i, and N is the number of kernels. To combine ELM with the MI measure, we can realize k_i as the RBF kernel as follows:

$$k_i(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{||\mathbf{x}-\mu_i||^2}{2\sigma_i^2}\right)\exp\left(-\frac{||\mathbf{y}-\mu_i||^2}{2\sigma_i^2}\right), \tag{3}$$

where σ_i and μ_i denote the bandwidth and center of the kernel. The original ELM-MI method [9] proposed to randomly sample σ_i from the uniform distribution in $[0,1]$, and set μ_i as the samples randomly selected from test data. As we have emphasized, such a random strategy imposes multiple limitations on the ELM-MI method. Hence we propose to replace the random strategy with our DKS method introduced in the next subsection.

Once the kernels are determined, the optimal value of $\boldsymbol{\beta} = [\beta_1,\ldots,\beta_N]^T$ can be calculated via the following equations [59]:

$$\begin{aligned}\boldsymbol{\beta} &= (\mathbf{H}+\lambda\mathbf{I})^{-1}\mathbf{k},\\ \mathbf{H} &= \mathbf{k}\mathbf{k}^T,\end{aligned} \tag{4}$$

where $\mathbf{k} = [k_1(\mathbf{x},\mathbf{y}),\ldots,k_N(\mathbf{x},\mathbf{y})]^T$ and hence $\mathbf{H}$ is a $N\times N$ Gram matrix, $\mathbf{I}$ is the identity matrix with the same size as $\mathbf{H}$, and λ is an empirical parameter for regularization. With Eq. (4), we are now ready to estimate $I(\mathbf{x},\mathbf{y})$ and interested readers can refer to [59,60] for the detailed derivation. Considering that the abnormal score is more straightforward for utilization, we adopt the negative of the estimated MI as the anomaly score, which is calculated as follows:

$$s(\mathbf{x},\mathbf{y}) = \frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}\boldsymbol{\beta} - \mathbf{k}^T\boldsymbol{\beta} + \frac{1}{2}. \tag{5}$$

Hence, if $s(\mathbf{x},\mathbf{y})$ is larger than a user-defined threshold τ, we consider that the test sequence $\mathbf{x}$ is anomalous. Since we aim at online anomaly detection for time-series data, which usually runs at a sequential processing manner, we select the sequence measured

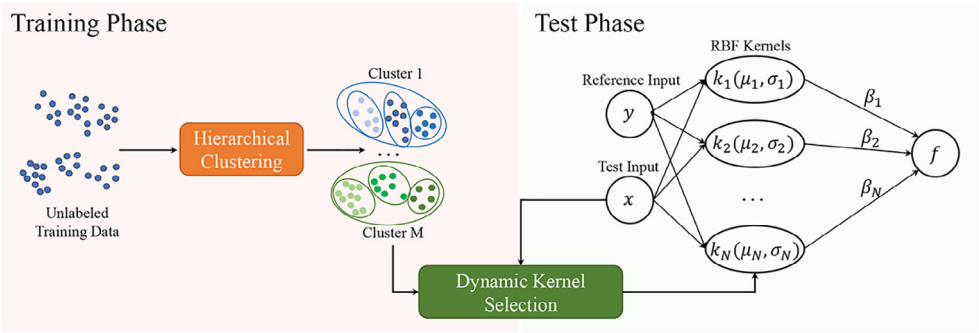


Fig. 2. The proposed unsupervised anomaly detection framework. The core of this framework is the dynamic kernel selection method, which groups unlabeled training data into clusters and utilizes them to determine the parameters of RBF kernels.

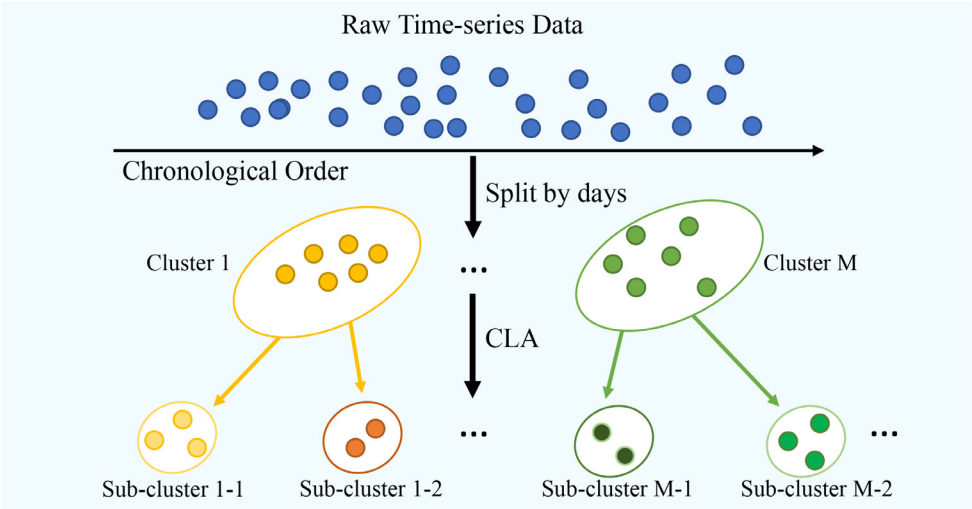


Fig. 3. Illustration of the 2-layer hierarchical clustering procedure with data collected in multiple days. Due to the natural hierarchical structure of time-series data, we can first split the data by days then apply the CLA clustering method to data in each day. This allows us to maintain the efficiency of the proposed method.

at time $t - \triangle t$ as the reference for the test sequence measured at time t, i.e., the anomaly score of $\mathbf{x}_t$ is given by $s(\mathbf{x}_t, \mathbf{x}_{t-\triangle t})$.

3.2. Dynamic kernel selection

Due to the random initialization of RBF kernels, the original ELM-MI method is limited to offline change point detection. To tackle this issue, we introduce the DKS method into the ELM-MI framework, as illustrated in Fig. 2. Our ELM-MI with DKS can be divided into the training phase and the test phase. In the training phase, we gather an unlabeled training dataset and perform the hierarchical clustering procedure (HCP) on it, while in the test phase, we adaptively determine the parameters of RBF kernels for each test sequence based on the clustered training data.

The purpose of the HCP is to group data with similar patterns into the same cluster, so that we can sample data on the cluster level and ensure that data of various patterns can be retrieved effectively. Here, we emphasize the hierarchical structure of the clustering process, because most time-series data can be generated and recorded hierarchically (e.g., year/month/day/hour). To demonstrate the idea of the HCP, here we take the HCP of 2 layers (e.g., day/hour) as an example. As shown in Fig. 3, assume that we have collected the training data for M days. In this case, the first layer of our HCP is divided into M clusters (each for one day). Next, data in each of the M clusters are further divided into sub-clusters by adopting existing clustering techniques, to construct the second layer of the HCP. Note that any clustering technique can be used, and in this paper we apply the communication with local agents (CLA) method [61] because it can determine the number of clusters automatically. More generally, for the HCP of multiple layers, we only apply the CLA method to data in the bottom layer, and the rest layers are generated by user-defined splits. In this way, the overall computational cost of the HCP is low and we can maintain the advantage of our framework in short training time.

As defined in Eq. (3), there are two parameters governing a RBF kernel, i.e., the center μ and the bandwidth σ. σ is less important, because we can normalize the data and the weight of kernel β is learnable. Therefore, we consider that σ is independent from the test data and we initialize it via sampling $\sigma \sim U(0, 1)$. On the other hand, μ shall be chosen appropriately, since it serves as the intermediate variable for estimating the MI between the inputs of the kernel. We consider μ to be the sampled training data that are similar to the test sequence, because they are more likely to encode the contexts of the test sequence. Note that the training data can also be sampled based on the reference sequence, because our MI is calculated based on both the test sequence and the reference sequence. In the following sections, we take the test sequence as an example for convenience of understanding.

Specifically, given the test sequence, our objective is to sample N training data as the centers of our kernels. In our case of 2-layer HCP, we first calculate the Euclidean distance between the test sequence and the center of each cluster in the first layer, that is,

$$d(\mathbf{x}, \mathbf{c}_m) = \sqrt{(\mathbf{x} - \mathbf{c}_m)^2}, \tag{6}$$

where $\mathbf{c}_m$ denotes the m-th center, $m = 1, \ldots, M$. To alleviate the curse of dimensionality problem, we perform principal component analysis (PCA) on the training dataset to obtain the principal components, which we project the test data on when we calculate the distance, with L2 or higher order distance. We then define the normalized weight of the m-th cluster with respect to the test sequence as follows:

$$w(\mathbf{x}, \mathbf{c}_m) = \frac{d(\mathbf{x}, \mathbf{c}_m)^{-1}}{\sum_{j=1}^{M} d(\mathbf{x}, \mathbf{c}_j)^{-1}}, \tag{7}$$

which is in reverse proportion to their distance. $w(\mathbf{x}, \mathbf{c}_m)$ is used to control the number of samples selected from the m-th cluster as follows,

$$N_m = \lfloor w(\mathbf{x}, \mathbf{c}_m) N \rfloor, \tag{8}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. In this way, a cluster that is more similar to the test sequence will have a higher weight, and its data will be sampled more likely. After that, we can determine the numbers of samples from the sub-clusters in the second layer in the same way. Note that the above process can be implemented as a recursive function, and be applied to our hierarchical data conveniently. Our DKS with HCP is summarized in Algorithm 1.

Table 1
Comparison of anomaly detection performances of the proposed method against other methods on the synthetic dataset.

Method	F_1 (%)	Precision (%)	Recall (%)
LSTM-KQE [63]	96.37	94.65	98.15
LSTM-AE with KQE [12]	96.94	95.83	98.07
LSTM-AE with OCSVM [12]	99.02	98.45	99.59
Ours	**99.30**	**98.81**	**99.80**

Table 2
Comparison of anomaly detection performances of the proposed method against other methods on the real-world gesture dataset.

Method	F_1 (%)	Precision (%)	Recall (%)
LOF [24]	53.28	**62.05**	46.68
LSCP [64]	60.01	47.01	82.95
LODA [65]	61.19	46.72	88.63
Isolation Forest [23]	64.42	47.92	98.24
Ours	**68.30**	51.94	**99.73**

4. Experiment

In this section, we validate the effectiveness of the proposed anomaly detection framework via a series of experiments. Specifically, we first evaluate the proposed framework on three public

Algorithm 1: Dynamic kernel selection with HCP

Input: Test sequence $\mathbf{x}$, unlabeled training data set S, number of kernels in ELM-MI N, number of hierarchical clustering layers H.

Define `Function` $DKS(\mathbf{x}, S, N, h, H)$ as follows:
`// Assume at the` h`-th layer,` S `has` M `clusters, i.e.,` $S = \bigcup_{m=1}^{M} S_m$
for $m = 1, ..., M$ **do**
  calculate N_m using Eq. (6) to (8);
end
if $h == H$ **then**
  `// Reach the bottom layer of hierarchical data`
  return $\bigcup_{m=1}^{M} \{N_m$ random samples from $S_m\}$;
else
  return $\bigcup_{m=1}^{M} DKS(\mathbf{x}, S_m, N_m, h+1, H)$;
end

Output: $DKS(\mathbf{x}, S, N, 1, H)$.

datasets including one synthetic SinSignal dataset [12], one gesture dataset [13] and the Skoltech anomaly benchmark (SKAB) [14] in Section 4.2. We then compare the proposed method with several state-of-the art methods in a real-life application of 4G LTE data analysis in Section 4.3. We apply a smoothing filter after our proposed method for all the datasets. We also conduct an ablation study in Section 4.4 with multiple variants of the proposed method, to provide a comprehensive study of the contribution of each proposed component and to evaluate the robustness of the proposed framework.

4.1. Setup

Following previous anomaly detection methods [14,12], we consider three widely-used metrics for evaluation, including the F_1 score, Precision, and Recall:

$$\begin{aligned} F_1 &= \frac{TP}{TP+\frac{1}{2}(FP+FN)}, \\ Precision &= \frac{TP}{FP+TP}, \\ Recall &= \frac{TP}{FN+TP}, \end{aligned} \tag{9}$$

where TP denotes the number of correct anomalous predictions, FP/FN denotes the number of incorrect anomalous/normal predictions. Higher F_1 score, precision and recall indicate better performances. Note that the Skoltech anomaly benchmark (SKAB) [14] posts F_1 score, false alarm rate (FAR), and missing alarm rate (MAR) of each method, so we keep using these three metrics in this benchmark, while using F_1 score, precision and recall fort the rest of experiments.

The proposed method is implemented with MATLAB, and runs on a PC with an Intel Core^{TM} i9-9880H 2.30 GHz processor and 64 GB RAM. Some comparison methods are implemented using frameworks provided by [62]. The anomaly score threshold τ is determined via cross validation, the number of kernels $N = 100$ for the synthetic SinSignal dataset and the SAKB dataset, $N = 200$ for the gesture dataset and $N = 150$ for the 4G LTE data analysis, the weight of regularization $\lambda = 0.1$ for the synthetic SinSignal dataset and the gesture dataset, $\lambda = 0.01$ for the SKAB dataset and $\lambda = 0.001$ for the 4G LTE data analysis, and the time difference of the reference sequence $\triangle t = 1$ frame as in [9]. N and λ are determined empirically. Generally, larger N improves the capacity of ELM and is suitable for complex data. Additionally, λ controls the weight of the regularization term in ELM-MI. If data are noisy or change dramatically, a small λ is preferable.

4.2. Evaluation on public datasets

The first public dataset is a synthetic SinSignal dataset provided by [12]. It contains 6988 normal samples in the training set and 2989 samples in the test set. In the test set, the 999*th* sample to the 1499*th* are target continues anomalies. This dataset has been evaluated by LSTM-KQE [63], LSTM-AE with KQE [12] and LSTM-AE with OCSVM [12], and the results using these three methods were summarized and presented in [12]. We adopt the performance of these methods reported in [12] for comparison. Table 1 presents the results of the proposed method and other compared methods on the synthetic SinSignal dataset. The proposed method outperforms other methods in all three metrics with the F_1 score of 99.30%. This shows the proposed method achieves considerable performance on detecting continues anomalies.

The second public dataset is a real-world gesture dataset provided by [13]. It records time-series measures of the X and Y coordinates of one actor's righthand extracted from a video in the real world. The actor keeps performing actions with the right hand, and one time period that contains anomalous actions is recorded. We implement four state-of-the-art anomaly detection methods for comparison using codes provided by [62] with their default parameters, including LOF [24], Locally Selective Combination of Parallel Outlier Ensembles (LSCP) [64], Lightweight on-line detector of anomalies (LODA) [65] and Isolation Forest [23]. Performances of the proposed method and other compared methods on the real-world gesture dataset are shown in Table 2. The results show that the proposed method achieves the best performance in F_1 score and recall (68.30% and 99.73%) and the second-best performance in precision, which means the proposed method achieves competitive performances among the compared state-of-the-art anomaly detection methods [1].

Finally, the SKAB [14] contains 34 subsets of multivariate time-series data, which is sufficiently large for evaluation. We follow the training/test splits provided by the benchmark and conduct pre-processing. The SKAB benchmark provides 10 anomaly detection methods as the baselines for comparison, including the multivariate state estimation technique (MSET) [66], Hotelling's T-squared statistic + Q statistic (SPE index) based on PCA (T-squared + Q (PCA)) [67], LSTM [68], multi-scale convolutional recurrent encoder-decoder (MSCRED) [69], Hotelling's T-squared statistic (T-squared) [70], feed-forward neural network with autoencoder (Autoencoder) [71], Isolation Forest [23], and three variants of these baselines. We adopt the performance of these methods reported by the benchmark for comparison.

Table 3 summarizes the results of the proposed method and the baselines. From these results, we can see that the proposed method outperforms the baselines in two metrics. The F_1 score of the proposed method is 87.51%. The proposed method also achieves the lowest MAR (11.04%). The baselines are hard to achieve the balance between FAR and MAR. For example, Isolation Forest obtains the best FAR (6.86%), on the other hand, its MAR is 72.09%, which indicates that it misses the anomalous data severely. Due to the DKS method, we can select the appropriate training data to determine the RBF kernels for each test sequence. Consequently, the proposed method exploits the temporal context effectively and achieves the considerable performance.

4.3. Evaluation on LTE data traffic

To demonstrate the potential of the proposed method in real-life applications, we propose to collect a 4G LTE traffic dataset and apply the proposed method on it. We consider Singtel, which is one of the main network operators providing LTE services in Singapore. We target at its 2.6 GHz frequency band and capture the LTE downlink traffic data within a specific area in Nanyang Technological University. The hardware experimental setup consists of Universal Software Radio Peripheral (USRP), NI-2954 Software Defined Radio (SDR), one Omnidirectional antenna, and a high-speed streaming enabled desktop computer, as shown in Fig. 4. The 4G LTE data has been recorded using the USRP, allocated for the operators' downlink signals. Then MATLAB LTE Toolbox [2] is used to decode the recorded data every 500 ms. Three types of raw data are decoded as follows:

1. RBa_p(%): The total percentage of resource blocks (RBs) allocated in the downlink direction at each timestamp.
2. user_n: Number of users detected at each timestamp.
3. user_PRB: The percentage of RBs allocated for each user detected at each timestamp.

In addition, to better analyze the potential anomalous traffic of

[1] For demo, see this link: https://personal.ntu.edu.sg/ezplin/NC-demo.htm.
[2] https://www.mathworks.com/help/lte/ug/ue-detection-using-downlink-signals.html

Table 3
Comparison of anomaly detection performances of the proposed method against other methods on the SKAB benchmark.

Method	F_1 (%)	FAR (%)	MAR (%)
Isolation Forest [23]	40	**6.86**	72.09
Autoencoder [71]	45	7.56	66.57
T-squared [70]	56	12.14	52.56
LSTM-VAE [14]	56	9.13	55.03
LSTM [68]	64	15.4	39.93
MSCRED [69]	64	13.56	41.16
T-squared + Q (PCA) [67]	67	13.95	36.32
LSTM-AE [14]	68	14.24	35.56
MSET [66]	73	20.82	20.08
Conv-AE [14]	79	13.69	17.77
Ours	**87.51**	7.85	**11.04**

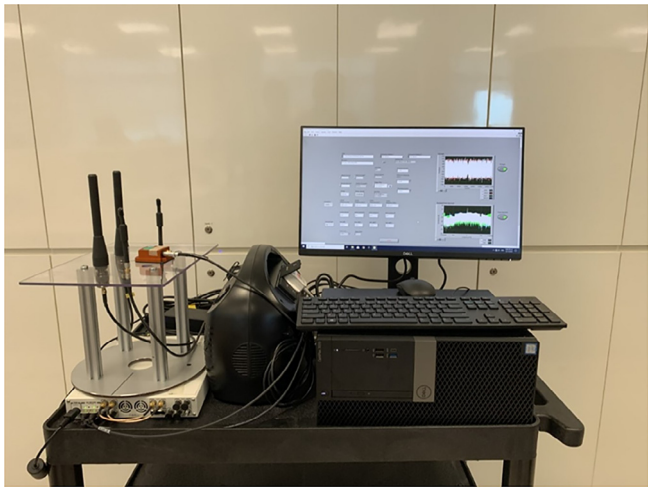


Fig. 4. The hardware setup for recording and decoding 4G LTE data.

each user, we further extract the information from user_PRB into (a) user_PRB_max and (b) user_PRB_mean, which represents the maximum and mean percentage of RBs allocated for each user detected at each timestamp, respectively.

In this way, each of our data sequence consists of four channels, namely, (a) RBa_p(%), (b) user_n, (c) user_PRB_max, and (d) user_PRB_mean. We collect a total number of 396,000 sequences in nine days. 20% of the collected data are selected as the test data while the remaining 80% are used for training. For the purpose of evaluation, we label the test data manually.

We implement seven state-of-the-art anomaly detection methods for comparison. Among them, COPOD [72], Isolation Forest [23], LOF [24], ROD [73], and LODA [65] are implemented based on their publicly available codes with their default parameters [62]. We also consider LSTM, as it is the representative deep learning technique for anomaly detection, and implement two variants of LSTM, including the unsupervised one combined with autoencoder (denoted as LSTM-AE) [39] and the one trained with heuristic classification labels (denoted as LSTM-cls) [74]. The same 2-layer HCP is adopted on this dataset.

Table 4 reports the quantitative comparison between the proposed method and other methods. It is clear that the proposed method reaches the highest F_1 score, precision and recall among all methods. The precision of LSTM-cls (86.58%) is closed to that of the proposed method (90.80%), but its recall (80.96%) is far lower than that of the proposed method (92.89%). This is mainly caused by the insufficiency of labels and anomalous data. LSTM-cls utilizes the signal properties obtained from *Fourier transform* to roughly label normal and anomalous samples. But such a criterion is empirical and hence its labels are noisy and limited in describing various anomalous patterns. On the other hand, the proposed method is fully unsupervised. Additionally, unlike the proposed method that uses the information of two overlapped data windows to generate one anomaly score, LSTM-cls generates outputs only based on one data window. This narrows down the temporal context and results in low recall rate. For example, as shown in Fig. 5, the anomaly score of LSTM-cls suddenly drops at around timestamp equal to 30 min, which indicates that the LSTM-cls misses the data anomalies.

Table 4
Comparison of the anomaly detection performance of the proposed method against other methods on our collected LTE dataset.

Method	F_1 (%)	Precision (%)	Recall (%)
COPOD [72]	67.23	61.27	74.48
Isolation Forest [23]	68.65	89.56	55.65
ROD [73]	78.49	86.62	71.76
LSTM-AE [39]	79.26	74.13	85.15
LOF [24]	80.85	82.25	79.50
LODA [65]	81.09	84.39	78.03
LSTM-cls [74]	83.68	86.58	80.96
Ours	**91.83**	**90.80**	**92.89**

We also provide the qualitative comparison of the proposed method against other methods in Fig. 5. Here we consider the baselines with the top 3 highest F_1 scores, including LOF, LODA and LSTM-cls. From Fig. 5, we can observe that our predicted anomaly scores are more stable, compared with these baselines. We owe this to the proposed HCP, because it organizes training sequences with similar patterns into the same cluster. Consequently, even if a cluster contains outliers, other sequences in the same cluster still can be selected to reduce the side-effects of the outliers.

4.4. Ablation study

In this section, we conduct the ablation study on our collected LTE dataset to analyze the proposed method in depth. First, to demonstrate the effectiveness of the DKS method, we realize four variants of the proposed method, including the original offline ELM-MI [9], the online ELM-MI baseline that randomly selects training data to initialize its RBF kernels, the online ELM-MI w/o HCP that adopts only 1 layer of clusters, i.e., the CLA method is only applied on the data directly, and the online ELM-MI w/ days that splits training data by days but without clustering data in each day.

The results of these variants are summarized in Table 5. The offline ELM-MI has the worst performance, which is reasonable as we have emphasized its limitations in exploiting temporal contexts. The online ELM-MI baseline is slightly better than the offline variant, but is still worse than the proposed method in F_1 scores and recall. This validates that randomly selecting training data to initialize the RBF kernels is ineffective, and hence the proposed DKS method is necessary. The online ELM-MI w/o HCP and the online ELM-MI w/ days together indicate that the hierarchical structure does exist in time-series data, and it helps to improve the overall performance.

Next, we propose to demonstrate that, the proposed framework overcomes the limitation of the original ELM-MI that it can only detect point anomalies. Fig. 6 presents an example sequence with both the point and group anomaly. We can see that the point anomaly is detected by the offline and online baselines, as well as the proposed method. However, for the group anomaly, the predicted scores of the two baselines drop apparently, while those of the proposed method are as high as those of point anomalies. This example validates the effectiveness of the DKS method: for the two baselines, their kernels are randomly initialized, and hence if the reference sequence is anomalous, the predicted anomaly score will

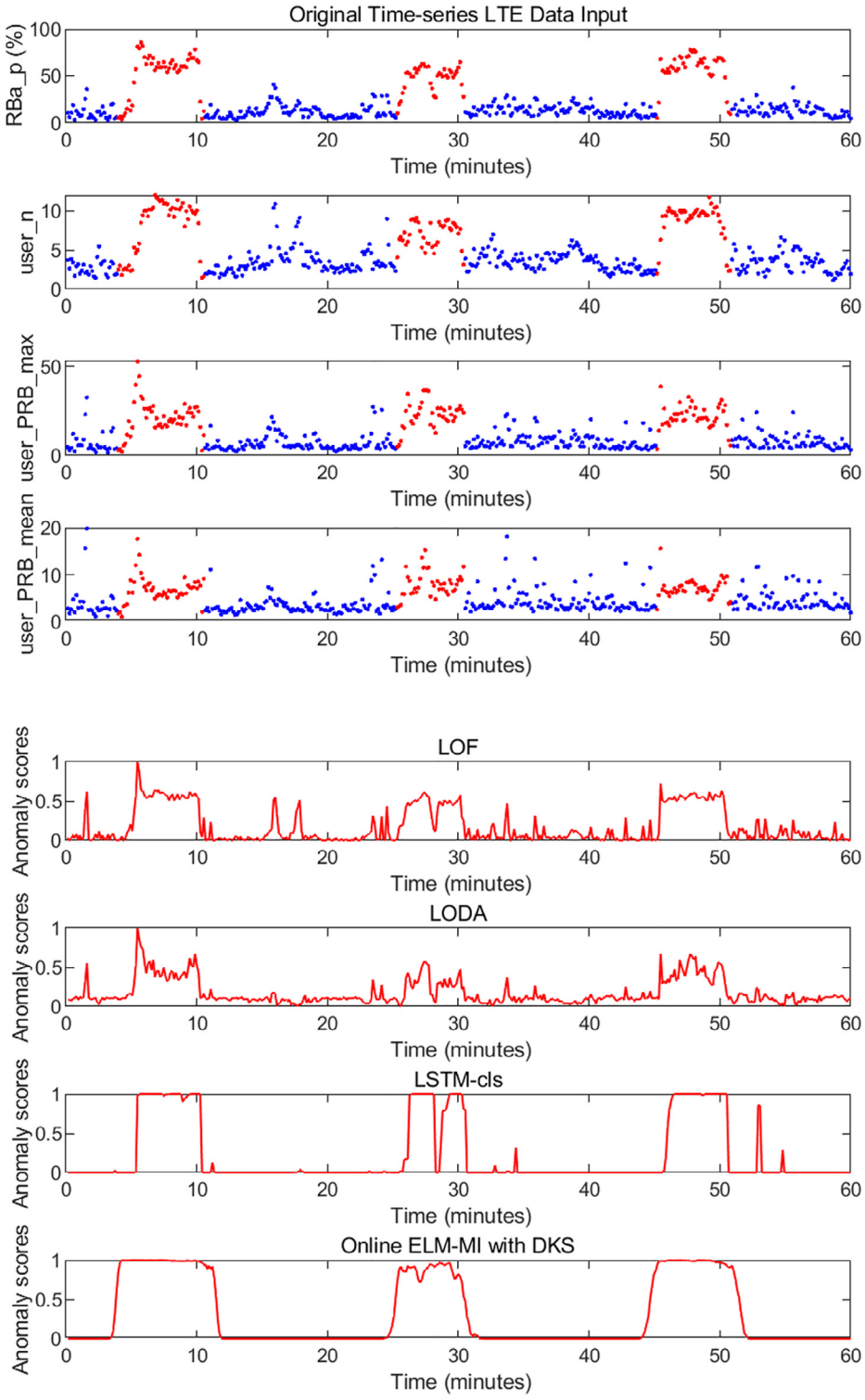


Fig. 5. Qualitative comparison of the proposed method and other methods on our collected LTE dataset. Normal data are labeled in blue while anomalous data are in red. Note that the figure has been normalized to 0 to 1 for better visualization purposes.

become smaller. As for the proposed method, the anomalous training data tend to be grouped into the same cluster, and hence their effects in the kernels are mitigated (because we select samples from multiple clusters). Consequently, even if both the reference and test sequences are anomalous, their anomaly score still can be high. This can also be validated by the example of contextual anomaly in Fig. 7. In this example, a latent peak in allocated resource blocks appears but the number of users remains low,

Table 5
Ablation study of the proposed method on our collected LTE dataset.

Method	F_1 (%)	Precision (%)	Recall (%)
Offline ELM-MI [9]	82.15	86.21	78.45
Online ELM-MI baseline	83.41	**92.82**	75.73
Online ELM-MI w/o HCP	84.65	88.94	80.75
Online ELM-MI w/ days	90.40	84.27	**97.49**
Ours	**91.83**	90.80	92.89

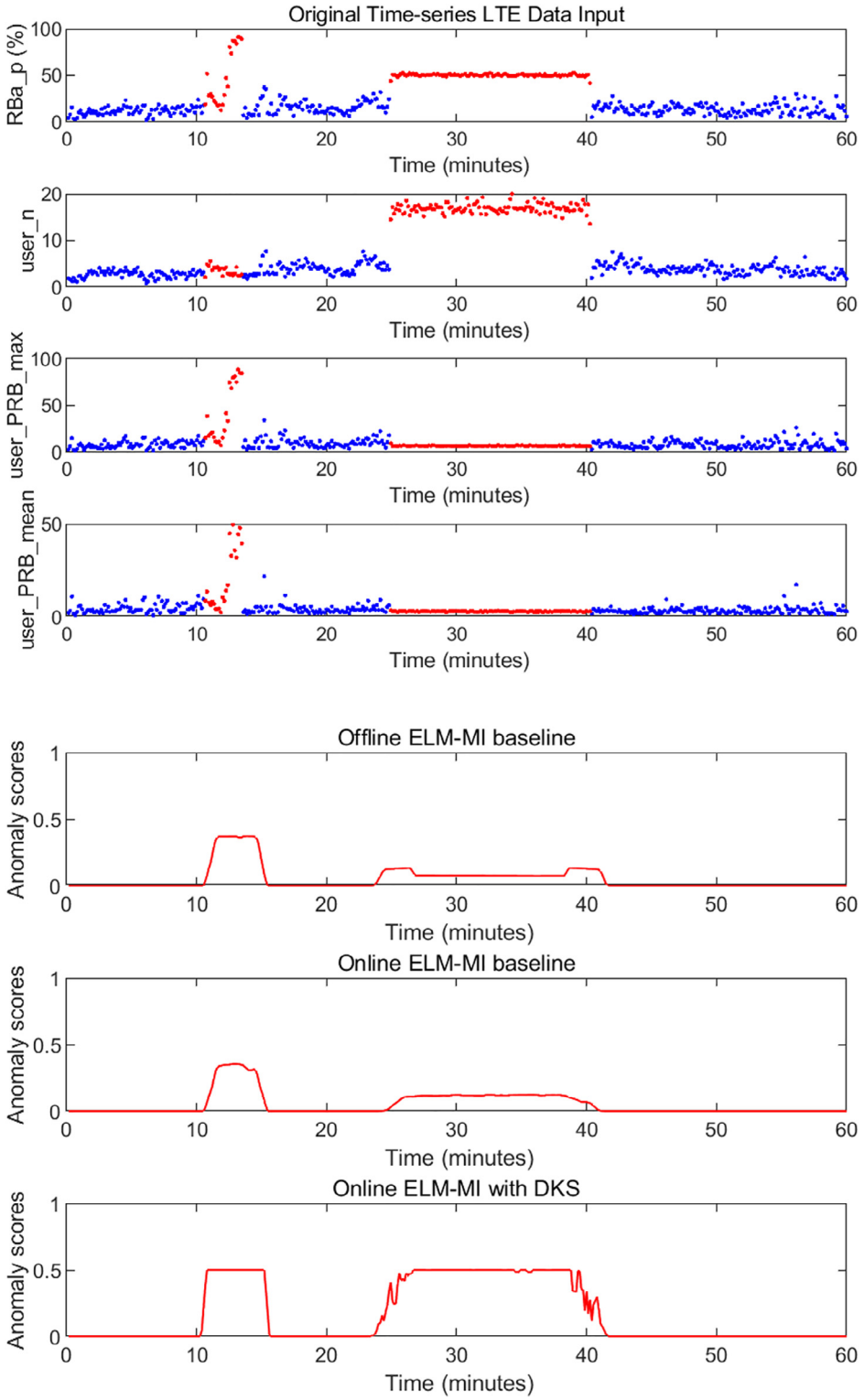


Fig. 6. Selective LTE sequences with point anomaly and group anomaly (top), and the predictions generated by the variants of our method (bottom). Normal data are labeled in blue while anomalous data are in red.

which is a sign of contextual anomaly that a few users occupy most resources. It is obvious that our predicted scores can describe this situation well.

Prior works [9,59,60] focus on RBF kernels to build the framework, but we are also interested in the performance of the proposed framework with different kernels. Therefore, we use two different types of kernels, including Polynomial kernels of degrees 1 and 2, and Sigmoid kernel[3], to evaluate the robustness of the proposed method. The results of using different types of kernels are summarized in Table 6. These results show that F_1 scores with differ-

[3] http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/

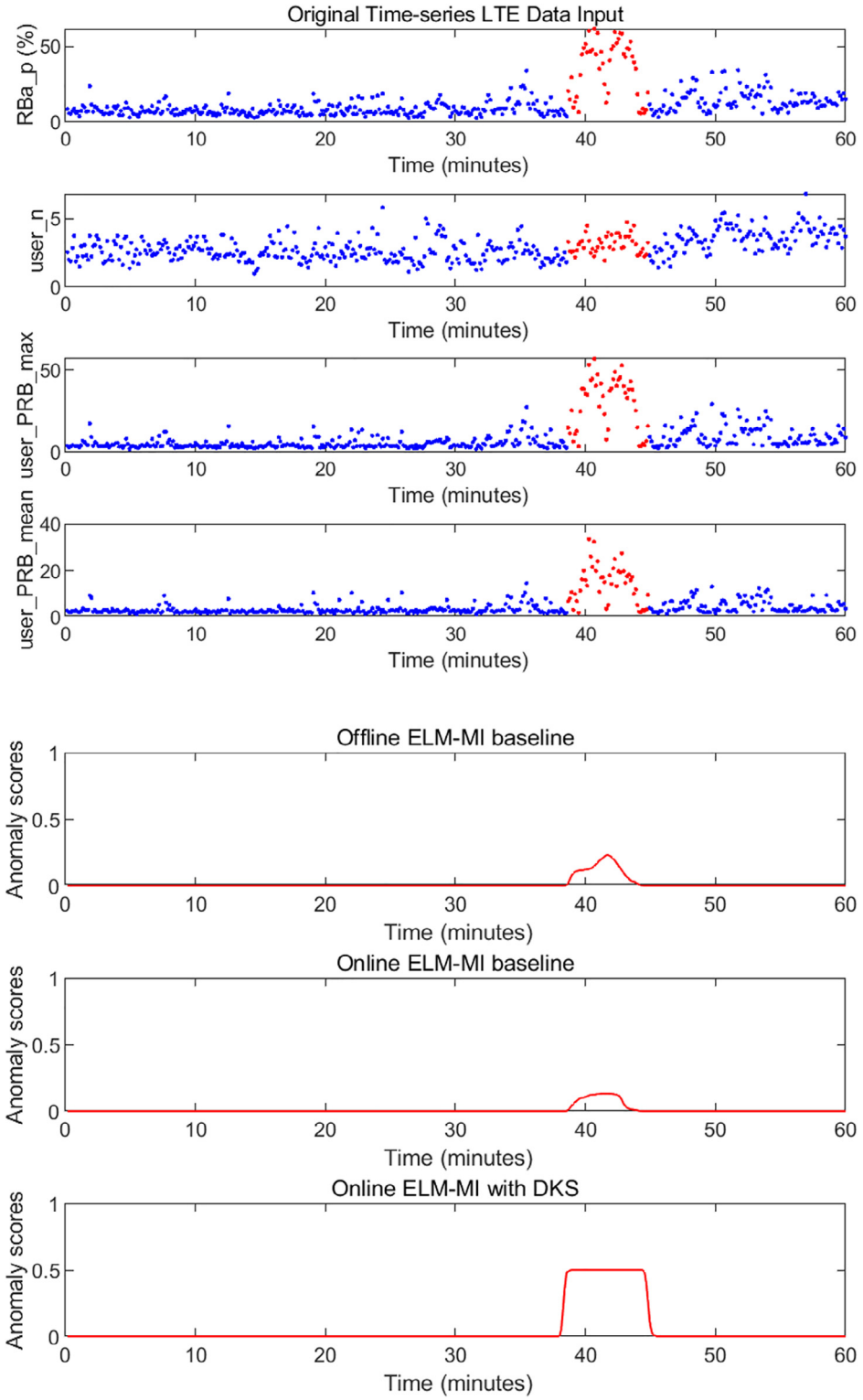


Fig. 7. Selective LTE sequences with contextual anomaly (top), and the predictions generated by the variants of our method (bottom). Normal data are labeled in blue while anomalous data are in red.

ent types of kernels are similar, which indicates that different kernels can be adopted in our method.

Next, we report the computational cost of the proposed method and three other methods in Table 7. We can see that the proposed method has the significant advantage in training speed (about 18 s), compared with the deep learning based methods like LSTM-AE (about 1,600 s). Furthermore, note that the offline ELM-MI needs the ensemble of multiple models, its inference speed is slower than that our method. Compared with the online ELM-MI w/o HCP, the training and inference speeds of our method are also

faster, mainly because the real clustering process is conducted on the bottom layer and it generates fewer clusters.

Besides, we evaluate the robustness of the proposed method using different choices of window size and threshold values. Table 8 shows the results of using different threshold values (ranging from 0.15 to 0.45 with increment of 0.05). Our F_1 scores remain stable with these settings of threshold and is the best when the threshold is set to 0.35. The results of the proposed method using different window sizes (based on best F_1 scores) are summarized in Table 9. These results show that the setting of window size has larger effects on the performance, compared with that of the threshold. As the window size mainly determines the context that can be received by the model, these results again validate our opinion that exploiting context helps to tackle the anomaly detection problem.

In addition, we aim to show that the proposed method can achieve effective detection performances without requiring the accumulation of a large amount of data. The results using different ratios of training data are listed in Table 10. We can see that even with only 20% training data, the proposed method can tackle anomaly detection effectively. The F_1 scores are stable with training data of ratios from 20% to 100%, which means the proposed method can achieve satisfactory performance without requiring the accumulation of a large amount of training data.

As the proposed method aims at unsupervised anomaly detection, we do not assume that all our training data are normal and do not make use of the label information of the training data. To study the effects of anomalous training data, we conduct an experiment to evaluate the proposed method with different ratios of anomalous training data. The original ratio of anomalies in our LTE training data is about 2.6%. We consider the case that the training data is anomaly-free, and the case that the ratio of anomalous training data is about 5%. Results of the proposed method and the online ELM-MI baseline using different ratios of anomalies in the training data are summarized in Table 11. Results show that F_1 scores of the proposed method with no anomalies or two different ratios of anomalies are very close, which means that anomalies in the training data have limited effect. On the contrary, there are clearer fluctuations of performances for online ELM-MI baseline. These fluctuations reflect that randomly selecting training data to initialize the RBF kernels is ineffective in dealing with anomalies in the training data.

Based on the above experimental results, we can conclude that the proposed method is robust under various circumstances, and hence it is an effective solution for online anomaly detection.

Table 6
Comparison of the anomaly detection performance of the proposed method with different types of kernels on our collected LTE dataset.

Type of kernel	F_1 (%)	Precision (%)	Recall (%)
Polynomial kernel (degree 1)	90.52	92.19	88.91
Polynomial kernel (degree 2)	89.54	**92.43**	86.82
Sigmoid kernel	89.32	92.20	86.61
RBF kernel*	**91.83**	90.80	**92.89**

* Current type of kernel of the proposed method.

Table 7
The computational cost of different methods (in seconds) on our collected LTE dataset.

Method	Training Time	Inference Time (per 100 samples)
LSTM-cls [74]	1601.06	0.45
Offline ELM-MI	N.A.	1.68
Online ELM-MI w/o HCP	26.91	0.37
Ours	18.32	0.32

Table 8
Comparison of the anomaly detection performance of the proposed method with different threshold values on our collected LTE dataset.

Value of threshold	F_1 (%)	Precision (%)	Recall (%)
0.15	91.23	85.40	**97.91**
0.2	91.52	86.57	97.07
0.25	91.82	87.79	96.23
0.3	91.76	89.31	94.35
0.35*	**91.83**	90.80	92.89
0.4	89.93	91.18	88.70
0.45	88.16	**92.63**	84.10

* Current value of threshold of the proposed method.

Table 9
Comparison of the anomaly detection performance of the proposed method with different window size on our collected LTE dataset.

Window size	F_1 (%)	Precision (%)	Recall (%)
25	87.28	83.14	91.84
30*	**91.83**	**90.80**	**92.89**
35	85.74	85.30	86.19
40	87.13	84.48	89.96

* Current window size of the proposed method.

Table 10
Comparison of the performance of the proposed method with different ratios of training data on our collected LTE dataset.

Ratio of data (%)	F_1 (%)	Precision (%)	Recall (%)
20	89.88	84.00	**96.65**
40	91.16	86.77	96.03
60	89.80	86.02	93.93
80	90.85	87.43	94.56
100*	**91.83**	**90.80**	92.89

* Current ratio of data used for the proposed method during the training phase.

Table 11
Comparison of the proposed method and the online ELM-MI baseline with different ratios of anomalies in the training data.

Ratio of anomalies (%)	Method	F_1 (%)	Precision (%)	Recall (%)
0	Online ELM-MI baseline	85.68	**92.91**	79.50
2.6*	Online ELM-MI baseline	83.41	92.82	75.73
5	Online ELM-MI baseline	83.16	91.90	75.94
0	**Ours**	**92.08**	88.44	**96.03**
2.6*	**Ours**	91.83	90.80	92.89
5	**Ours**	91.88	90.30	93.51

* Current ratio of anomalies in our collected LTE dataset.

5. Conclusion

In this paper, we have proposed an extreme learning machine based mutual information estimation framework to tackle the anomaly detection problem in multivariate time series. The proposed framework adopts the dynamic kernel selection method, which conducts the hierarchical clustering on unsupervised training data to generate clusters and determine the RBF kernels adaptively. With the proposed method, we better exploit the temporal contexts to complete unsupervised online anomaly detection, while maintaining the efficiency of extreme learning machine. Experimental results on three public datasets and our collected 4G LTE dataset demonstrate that the proposed method outperforms existing anomaly detection methods.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. Chen, P. Liu, Z. Liu, H. Tang, L. Hong, J. Dong, J. Conradt, A. Knoll, Neuroaed: Towards efficient abnormal event detection in visual surveillance with neuromorphic vision sensor, IEEE Trans. Inf. Forensics Secur. 16 (2021) 923–936, https://doi.org/10.1109/TIFS.2020.3023791.

[2] S. Lee, H.G. Kim, Y.M. Ro, Bman: Bidirectional multi-scale aggregation networks for abnormal event detection, IEEE Trans. Image Process. 29 (2020) 2395–2408, https://doi.org/10.1109/TIP.2019.2948286.

[3] Y. Liu, Y. Lin, Q. Xiao, G. Hu, J. Wang, Self-adversarial variational autoencoder with spectral residual for time series anomaly detection, Neurocomputing 458 (2021) 349–363, https://doi.org/10.1016/j.neucom.2021.06.030.

[4] C. Yin, S. Zhang, J. Wang, N.N. Xiong, Anomaly detection based on convolutional recurrent autoencoder for iot time series, IEEE Trans. Syst. Man Cybern.: Syst. (2020) 1–11, https://doi.org/10.1109/TSMC.2020.2968516.

[5] Y. Zhang, M. Li, Z. Ji, W. Fan, S. Yuan, Q. Liu, Q. Chen, Twin self-supervision based semi-supervised learning (ts-ssl): retinal anomaly classification in sd-oct images, Neurocomputing 462 (2021) 491–505, https://doi.org/10.1016/j.neucom.2021.08.051.

[6] Z. Li, Y. Zhang, Hyperspectral anomaly detection via image super-resolution processing and spatial correlation, IEEE Trans. Geosci. Remote Sens. 59 (3) (2021) 2307–2320, https://doi.org/10.1109/TGRS.2020.3005924.

[7] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. 41 (3) (2009) 1–58.

[8] G. Pang, C. Shen, L. Cao, A.V.D. Hengel, Deep learning for anomaly detection: a review, ACM Comput. Surv. 54 (2) (2021) 1–38.

[9] B.-S. Oh, L. Sun, C.S. Ahn, Y.K. Yeo, Y. Yang, N. Liu, Z. Lin, Extreme learning machine based mutual information estimation with application to time-series change-points detection, Neurocomputing 261 (2017) 204–216.

[10] Y. Tian, M. Mirzabagheri, S.M.H. Bamakan, H. Wang, Q. Qu, Ramp loss one-class support vector machine; a robust and effective approach to anomaly detection problems, Neurocomputing 310 (2018) 223–235, https://doi.org/10.1016/j.neucom.2018.05.027.

[11] S.M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning, Pattern Recogn. 58 (2016) 121–134.

[12] H. Nguyen, K.P. Tran, S. Thomassey, M. Hamad, Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management, Int. J. Inf. Manage. 57 (2021) 102282.

[13] E. Keogh, J. Lin, A. Fu, Hot sax: efficiently finding the most unusual time series subsequence, in: Fifth IEEE International Conference on Data Mining (ICDM'05), 2005, pp. 8 pp.–. doi:10.1109/ICDM.2005.79..

[14] I.D. Katser, V.O. Kozitsin, Skoltech anomaly benchmark (skab), https://www.kaggle.com/dsv/1693952 (2020). doi:10.34740/KAGGLE/DSV/1693952..

[15] H. Wang, L. Li, P. Pan, Y. Wang, Y. Jin, Online detection of abnormal passenger out-flow in urban metro system, Neurocomputing 359 (2019) 327–340, https://doi.org/10.1016/j.neucom.2019.04.075.

[16] T.-C. Fu, A review on time series data mining, Eng. Appl. Artif. Intell. 24 (1) (2011) 164–181.

[17] R.A.A. Habeeb, F. Nasaruddin, A. Gani, I.A.T. Hashem, E. Ahmed, M. Imran, Real-time big data processing for anomaly detection: a survey, Int. J. Inf. Manage. 45 (2019) 289–307.

[18] Z. Wang, Y. Guo, Rumor events detection enhanced by encoding sentimental information into time series division and word representations, Neurocomputing 397 (2020) 224–243, https://doi.org/10.1016/j.neucom.2020.01.095.

[19] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.

[20] W.D. Fisher, T.K. Camp, V.V. Krzhizhanovskaya, Anomaly detection in earth dam and levee passive seismic data using support vector machines and automatic feature selection, J. Comput. Sci. 20 (2017) 143–153.

[21] P. Cheema, N.L.D. Khoa, M. Makki Alamdari, W. Liu, Y. Wang, F. Chen, P. Runcie, On structural health monitoring using tensor analysis and support vector machine with artificial negative data, in: Proceedings of the 25th ACM international on conference on information and knowledge management, 2016, pp. 1813–1822..

[22] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S.A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: International conference on machine learning, PMLR, 2018, pp. 4393–4402.

[23] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data (TKDD) 6 (1) (2012) 1–39.

[24] S. Mishra, M. Chawla, A comparative study of local outlier factor algorithms for outliers detection in data streams, in: Emerging Technologies in Data Mining and Information Security, Springer, 2019, pp. 347–356.

[25] G. Wu, Z. Zhao, G. Fu, H. Wang, Y. Wang, Z. Wang, J. Hou, L. Huang, A fast knn-based approach for time sensitive anomaly detection over data streams, in: International Conference on Computational Science, Springer, 2019, pp. 59–74.

[26] M. Shao, P. Sun, J. Li, Q. Yan, Z. Feng, Tree decomposition based anomalous connected subgraph scanning for detecting and forecasting events in attributed social media networks, Neurocomputing 407 (2020) 83–93, https://doi.org/10.1016/j.neucom.2020.04.064.

[27] B. Liu, Y. Qi, K.-J. Chen, Sequential online prediction in the presence of outliers and change points: An instant temporal structure learning approach, Neurocomputing 413 (2020) 240–258, https://doi.org/10.1016/j.neucom.2020.07.011.

[28] W. Luo, W. Liu, S. Gao, Normal graph: Spatial temporal graph convolutional networks based prediction network for skeleton based video anomaly detection, Neurocomputing 444 (2021) 332–337, https://doi.org/10.1016/j.neucom.2019.12.148.

[29] M. Yu, S. Sun, Policy-based reinforcement learning for time series anomaly detection, Eng. Appl. Artif. Intell. 95 (2020) 103919.

[30] R. Wu, S. Li, C. Chen, A. Hao, Improving video anomaly detection performance by mining useful data from unseen video frames, Neurocomputing 462 (2021) 523–533, https://doi.org/10.1016/j.neucom.2021.05.112.

[31] J. von Schleinitz, M. Graf, W. Trutschnig, A. Schröder, Vasp: An autoencoder-based approach for multivariate anomaly detection and robust time series prediction with application in motorsport, Eng. Appl. Artif. Intell. 104 (2021) 104354.

[32] M. Canizo, I. Triguero, A. Conde, E. Onieva, Multi-head cnn–rnn for multi-time series anomaly detection: An industrial case study, Neurocomputing 363 (2019) 246–260, https://doi.org/10.1016/j.neucom.2019.07.034.

[33] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S.A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 4393–4402.

[34] L. Ruff, R.A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, M. Kloft, Deep semi-supervised anomaly detection, arXiv preprint arXiv:1906.02694..

[35] Y. Zhou, X. Liang, W. Zhang, L. Zhang, X. Song, Vae-based deep svdd for anomaly detection, Neurocomputing 453 (2021) 131–140, https://doi.org/10.1016/j.neucom.2021.04.089.

[36] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[37] L. Xia, Z. Li, An abnormal event detection method based on the riemannian manifold and lstm network, Neurocomputing 463 (2021) 144–154, https://doi.org/10.1016/j.neucom.2021.08.017.

[38] H.D. Trinh, L. Giupponi, P. Dini, Urban anomaly detection by processing mobile traffic traces with lstm neural networks, in: 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), 2019, pp. 1–8..

[39] H.D. Trinh, E. Zeydan, L. Giupponi, P. Dini, Detecting mobile traffic anomalies through physical control channel fingerprinting: A deep semi-supervised approach, IEEE Access 7 (2019) 152187–152201, https://doi.org/10.1109/ACCESS.2019.2947742.

[40] J. Cao, K. Zhang, H. Yong, X. Lai, B. Chen, Z. Lin, Extreme learning machine with affine transformation inputs in an activation function, IEEE Trans. Neural Netw. Learn. Syst. 30 (7) (2018) 2093–2107.

[41] J. Cao, K. Zhang, M. Luo, C. Yin, X. Lai, Extreme learning machine and adaptive sparse representation for image classification, Neural Netw. 81 (2016) 91–102.

[42] H. Zhuang, Z. Lin, K.-A. Toh, Training a multilayer network with low-memory kernel-and-range projection, J. Franklin Inst. 357 (1) (2020) 522–550.

[43] H. Zhuang, Z. Lin, K.-A. Toh, Correlation projection for analytic learning of a classification network, Neural Process. Lett. (2021) 1–22.

[44] H. Zhuang, Z. Lin, K.-A. Toh, Blockwise recursive moore-penrose inverse for network learning, IEEE Transactions on Systems, Man, and Cybernetics: Systems..

[45] J. Cao, Z. Lin, G.-B. Huang, N. Liu, Voting based extreme learning machine, Inf. Sci. 185 (1) (2012) 66–77.

[46] Q. Leng, H. Qi, J. Miao, W. Zhu, G. Su, One-class classification with extreme learning machine, Mathematical problems in engineering 2015..

[47] H. Dai, J. Cao, T. Wang, M. Deng, Z. Yang, Multilayer one-class extreme learning machine, Neural Netw. 115 (2019) 11–22.

[48] G. Huang, S. Song, J.N. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405–2417.

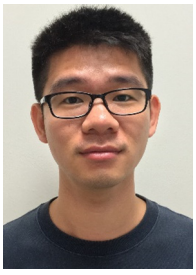
[49] T. Liu, Y. Yang, G.-B. Huang, Y.K. Yeo, Z. Lin, Driver distraction detection using semi-supervised machine learning, IEEE Trans. Intell. Transp. Syst. 17 (4) (2015) 1108–1120.

[50] J. Cao, T. Chen, J. Fan, Fast online learning algorithm for landmark recognition based on bow framework, in: 2014 9th IEEE Conference on Industrial Electronics and Applications, IEEE, 2014, pp. 1163–1168.

[51] J. Cao, T. Chen, J. Fan, Landmark recognition with compact bow histogram and ensemble elm, Multimed. Tools Appl. 75 (5) (2016) 2839–2857.
[52] V.M. Janakiraman, D. Nielsen, Anomaly detection in aviation data using extreme learning machines, in: 2016 international joint conference on neural networks (IJCNN), IEEE, 2016, pp. 1993–2000.
[53] M. Tsukada, M. Kondo, H. Matsutani, Os-elm-fpga: An fpga-based online sequential unsupervised anomaly detector, in: European Conference on Parallel Processing, Springer, 2018, pp. 518–529.
[54] W. Guo, T. Xu, K. Tang, M-estimator-based online sequential extreme learning machine for predicting chaotic time series with outliers, Neural Comput. Appl. 28 (12) (2017) 4093–4110.
[55] W. Guo, T. Xu, K. Tang, J. Yu, S. Chen, Online sequential extreme learning machine with generalized regularization and adaptive forgetting factor for time-varying system prediction, Math. Probl. Eng. (2018).
[56] Y. Song, J. Zhang, Automatic recognition of epileptic eeg patterns via extreme learning machine and multiresolution feature extraction, Expert Syst. Appl. 40 (14) (2013) 5477–5489.
[57] Y. Kaya, M. Uyar, A hybrid decision support system based on rough set and extreme learning machine for diagnosis of hepatitis disease, Appl. Soft Comput. 13 (8) (2013) 3429–3438.
[58] Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, B. Wang, Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis, BMC bioinformatics, vol. 14, Springer, 2013, pp. 1–11.
[59] T. Suzuki, M. Sugiyama, T. Kanamori, J. Sese, Mutual information estimation reveals global associations between stimuli and biological processes, Bioinformatics 10 (1) (2009) 1–12.
[60] M. Sugiyama, Machine learning with squared-loss mutual information, Entropy 15 (1) (2013) 80–112.
[61] Z. Wang, Z. Yu, C.P. Chen, J. You, T. Gu, H.-S. Wong, J. Zhang, Clustering by local gravitation, IEEE Trans. Cybern. 48 (5) (2017) 1383–1396.
[62] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, J. Mach. Learn. Res. 20 (96) (2019) 1–7, URL: http://jmlr.org/papers/v20/19-011.html.
[63] K.P. Tran, H. Du Nguyen, S. Thomassey, Anomaly detection using long short term memory networks and its applications in supply chain management, IFAC-PapersOnLine 52 (13) (2019) 2408–2412.
[64] Y. Zhao, Z. Nasrullah, M.K. Hryniewicki, Z. Li, Lscp: Locally selective combination in parallel outlier ensembles, in: Proceedings of the 2019 SIAM International Conference on Data Mining, SIAM, 2019, pp. 585–593.
[65] T. Pevnỳ, Loda: Lightweight on-line detector of anomalies, Mach. Learn. 102 (2) (2016) 275–304.
[66] N. Zavaljevski, K.C. Gross, Sensor fault detection in nuclear power plants using multivariate state estimation technique and support vector machines., Tech. rep., Argonne National Lab., Argonne, IL (US) (2000)..
[67] S. Joe Qin, Statistical process monitoring: basics and beyond, J. Chemom.: J. Chemom. Soc. 17 (8–9) (2003) 480–502.
[68] P. Filonov, A. Lavrentyev, A. Vorontsov, Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model, arXiv preprint arXiv:1612.06676..
[69] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N.V. Chawla, A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 1409–1416..
[70] H. Hotelling, Multivariate quality control-illustrated by the air testing of sample bombsights, 1947..
[71] J. Chen, S. Sathe, C. Aggarwal, D. Turaga, Outlier detection with autoencoder ensembles, in, in: Proceedings of the SIAM international conference on data mining, 2017, pp. 90–98.
[72] Z. Li, Y. Zhao, N. Botta, C. Ionescu, X. Hu, Copod: copula-based outlier detection, arXiv preprint arXiv:2009.09463..
[73] Y. Almardeny, N. Boujnah, F. Cleary, A novel outlier detection method for multivariate data, IEEE Trans. Knowl. Data Eng. (2020), https://doi.org/10.1109/TKDE.2020.3036524, 1–1.
[74] Y. Cherdo, P. d. Kerret, R. Pawlak, Training lstm for unsupervised anomaly detection without a priori knowledge, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2020, pp. 4297–4301..

Xinggan Peng received the B.Eng. (Hons.) degree from both University of Electronic Science and Technology of China and University of Glasgow in 2017. He received the M.S. degree in Electrical and Computer Engineering from The Ohio State University in 2019. He is currently a Ph.D. candidate at school of Electrical and Electronic Engineering, Nanyang Technological University. His main interests include signal processing, intelegent transportation system and deep learning.

Hanhui Li received the Ph.D. degree in computer software and theory from Sun Yatsen University, Guangzhou, China, in 2018, where he also received the B.S. degree in computer science and technology in 2012. He was a research fellow in Nanyang Technological University, Singapore. He is currently an Associate Researcher with Sun Yat-sen University, Shenzhen Campus. His research interests include image processing, computer vision and deep learning.



Feng Yuan received the B. Eng. (Hons.) and Ph.D. degrees from School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore, in 2013 and 2018 respectively. He was a Project Officer with the School of Electrical and Electronic Engineering, NTU, from 2017 to 2018. He is working as a Research Scientist with Temasek Lab at NTU since Aug. 2018. His current research interests are localization and tracking algorithm, sensor networks and propagation models for wireless communication systems.

Sirajudeen received his B.Eng and M.Eng in Electrical and Electronics Engineering from the Nanyang Technological University, Singapore in 1997 and 2000 respectively. He received his Ph.D from the University of Cambridge, UK in 2003. His Ph.D work was on Bayesian estimation using Markov Chain Monte Carlo techniques. He was a faculty member in the School of EEE, Nanyang Technological University from 2004 to 2009. He joined Temasek Labs@NTU in 2009. He is currently the programme director of the centre for communications and signal processing at TL@NTU. His research interests include Bayesian signal processing, statistical signal processing and array processing.

Zhebin Chen received the Bachelor and Master degree in mathematics and statistics from Chongqing University in 2018 and University of Edinburgh in 2019, respectively. He is now a Ph.D. student in Nanyang Technological University, Singapore. His research interests include machine learning, data analytics, and their applications to power engineering.

Zhiping Lin received the B.Eng. degree in control engineering from the South China Institute of Technology, Guangzhou, China, in 1982, and the Ph.D. degree in information engineering from the University of Cambridge, Cambridge, U.K., in 1987. He worked with the University of Calgary, Calgary, AB, Canada, Shantou University, Shantou, China, and DSO National Laboratories, Singapore, before joining the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 1999. His research interests are in statistical and biomedical signal/image processing, and machine learning. Dr. Lin was the Editor-in-Chief of Multidimensional Systems and Signal Processing from 2011 to 2015, and has been in its editorial board since 1993. He was an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-PART II: EXPRESS BRIEFS and the Subject Editor for the Journal of the Franklin Institute. He is the coauthor of the 2007 Young Author Best Paper Award from the IEEE SIGNAL PROCESSING SOCIETY. He was a Distinguished Lecturer of the IEEE Circuits and Systems Society (CAS) from 2007 to 2008, and served as the Chair of IEEE CAS Singapore Chapter from 2007 to 2008, and in 2019.