Real-Time Nonparametric Anomaly Detection in High-Dimensional Settings

Mehmet Necip Kurt, *Student Member, IEEE*,
Yasin Yılmaz, *Member, IEEE*, and Xiaodong Wang, *Fellow, IEEE*

Abstract—Timely detection of abrupt anomalies is crucial for real-time monitoring and security of modern systems producing high-dimensional data. With this goal, we propose effective and scalable algorithms. Proposed algorithms are nonparametric as both the nominal and anomalous multivariate data distributions are assumed unknown. We extract useful univariate summary statistics and perform anomaly detection in a single-dimensional space. We model anomalies as persistent outliers and propose to detect them via a cumulative sum-like algorithm. In case the observed data have a low intrinsic dimensionality, we find a submanifold in which the nominal data are embedded and evaluate whether the sequentially acquired data persistently deviate from the nominal submanifold. Further, in the general case, we determine an acceptance region for nominal data via Geometric Entropy Minimization and evaluate whether the sequentially observed data persistently fall outside the acceptance region. We provide an asymptotic lower bound and an asymptotic approximation for the average false alarm period of the proposed algorithm. Moreover, we provide a sufficient condition to asymptotically guarantee that the decision statistic of the proposed algorithm does not diverge in the absence of anomalies. Experiments illustrate the effectiveness of the proposed schemes in quick and accurate anomaly detection in high-dimensional settings.

Index Terms—High-dimensional data, summary statistic, geometric entropy minimization (GEM), principal component analysis (PCA), real-time anomaly detection, nonparametric, cumulative sum (CUSUM)

1 Introduction

1.1 Background

Anomaly refers to deviation from the expected (regular) behavior. Anomaly detection has been widely studied and to name a few, many distance-based, density-based, subspace-based, support vector machine (SVM)-based, neural networks-based, and information theoretic anomaly detection techniques have been proposed in the literature in a variety of application domains such as intrusion detection in computer and communication networks, credit card fraud detection, industrial damage detection, etc. [1], [2], [3]. Early and accurate detection of anomalies has a critical importance for safe and reliable operation of many modern systems such as the power networks (smart grid) and the Internet of Things (IoT) networks that produce high-dimensional data streams. Such sudden anomalies often correspond to changes in the underlying statistical properties of the observed processes. To detect the changes, the framework of quickest detection [4], [5] is quite suitable, where the statistical inference about the monitored process is typically done through observations acquired sequentially over time and the goal is to detect the changes as soon as possible after they occur while limiting the risk of false alarm.

The well-known quickest detection algorithms are model-based: they require either the exact knowledge or parameter estimates of the probability density functions (pdfs) of the observed data stream for both the pre- and post-change cases [4], [5], [6]. For instance, the generalized likelihood ratio (GLR) approach estimates the unknown pdf parameters, plugs them back into the likelihood ratio term, and performs the change/anomaly detection accordingly [7], [8], [9]. On the other hand, in high-dimensional settings, e.g., large-scale complex networks consisting of large number of nodes that exhibit complex interactions, it is usually difficult to model or intractable to estimate the high-dimensional multivariate pdfs. Moreover, it is, in general, quite difficult to model all possible types of anomalies. Hence, in a general anomaly detection problem, the post-change (anomalous) pdf is totally unknown. To overcome such difficulties, we propose to extract useful univariate summary statistics from the observed high-dimensional data and perform the anomaly detection task in a single-dimensional space, through which we also aim to make more efficient use of limited computational resources and to speed up the algorithms, that is especially required in time-sensitive online settings.

Although a summary statistic may not completely characterize a random process, it can be useful to evaluate the non-similarity between random processes with different statistical properties. In our problem, there are two main challenges to determine good summary statistics: (i) summary statistics should be well informative to (statistically) distinguish anomalous data from nominal (non-anomalous)

- *M. N. Kurt and X. Wang are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA. E-mail: m.n.kurt@columbia.edu, wangx@ee.columbia.edu.*
- *Y. Yılmaz is with the Department of Electrical Engineering, University of South Florida, Tampa, FL 33620 USA. E-mail: yasiny@usf.edu.*

Manuscript received 11 Aug. 2018; revised 28 Apr. 2019; accepted 21 Jan. 2020. Date of publication 30 Jan. 2020; date of current version 3 June 2021. (Corresponding author: Xiaodong Wang.)
Recommended for acceptance by R. Wildes.
Digital Object Identifier no. 10.1109/TPAMI.2020.2970410

data, (ii) since we are in an online setting, computation of the summary statistics should be simple to allow for real-time processing. In this paper, we consider two alternative summary statistics: (i) if the observed nominal data has a low intrinsic dimensionality, first finding a representative low-dimensional submanifold in which the nominal data are embedded and then computing a statistic that shows how much the incoming data stream deviates from the nominal submanifold; (ii) in the general case, determining an acceptance region for the nominal data via the Geometric Entropy Minimization (GEM) method [10], [11] and then computing a nearest neighbor (NN) statistic that shows how much the incoming data stream is away from the acceptance region. We propose to first compute a set of nominal summary statistics that constitute the baseline in an offline phase and then monitor possible deviations of online summary statistics from the baseline statistics.

Anomaly detection schemes based on parametric models are vulnerable to model mismatch that limits their applicability. For instance, it is common to fit a Gaussian or Gaussian mixture model to the observed data or the data after dimensionality reduction [1], [2], [12], [13] and to assume Gaussian noise or residual terms, see e.g., [14]. Such parametric approaches are powerful only if the observed data perfectly matches with the presumed model. On the other hand, nonparametric (model-free) data-driven techniques are robust to the data model mismatch, that results in wider applicability of such techniques in a variety of problems. Moreover, in high-dimensional settings, the lack of parametric models is common and complicated parameter-laden algorithms generally result in low performance, overfitting, and bias towards particular anomaly types [15]. Hence, in this paper, we do not make parametric model assumptions for the observed high-dimensional data stream nor for the extracted summary statistics. However, note that if the observed data stream or the summary statistics can be well modeled, then a parametric detection method can be preferred since the parametric methods usually have higher statistical power and their performance can be more easily analyzed. Hence, our method should be mainly advantageous in high-dimensional settings where the data models are unknown.

Conventional anomaly detection schemes ignore the temporal relation between anomalous data points and make sample-by-sample decisions [1], [2]. Such schemes are essentially outlier detectors that are vulnerable to false alarms since it is possible to observe non-persistent random outliers in a normal system operation (no anomaly) due to e.g., heavy-tailed random noise processes. On the other hand, if a system produces persistent outliers, then this may indicate an actual anomaly. Hence, we define an anomaly as persistent outliers and from the observed data stream, we propose to accumulate statistical evidence for anomaly over time, similarly to the accumulation of log-likelihood ratios (LLRs) in the well-known cumulative sum (CUSUM) algorithm for change detection [Section 2.2] [5]. With the goal of making a reliable decision, we declare an anomaly only if we have a strong evidence for that. The sequential decision making based on the accumulated evidence also enables the detection of small but persistent changes, that would be missed by outlier detectors.

1.2 Related Work

Batch algorithms are widely encountered in the anomaly detection literature [1], [2], that require the entire data before processing. Clearly, such techniques are not suitable in the online settings. For instance, in [16], [17], via the principal component analysis (PCA), the data are decomposed into normal and anomalous components and the data points with large anomalous components are classified as anomalous. The well-known nonparametric statistical tests such as the Kolmogrov-Smirnov test, the Wilcoxon signed-rank test, and the Pearson's chi-squared test are also mainly designed for batch processing. Although several sliding window-based versions of them have been proposed for online anomaly detection, see e.g., [9], [18], the window-based approach has an inherent detection latency caused by the window size. More importantly, such tests are primarily designed for univariate data, with no direct extensions for multivariate data.

Various online anomaly detection techniques for multivariate data streams have also been proposed in the literature. The SVM-based one-class classification algorithms in [19], [20] determine a decision region for nominal data after mapping the data onto a kernel space, where there is no clear control mechanism on the false alarm rate. Moreover, the choice and complexity of computing the kernel functions are among the disadvantages of such algorithms. A similar algorithm is presented in [21] where the training data might contain a small number of anomalous data points or outliers. An extension of the one-class SVM algorithm [19] is proposed in [22] where the objective is to detect anomalies in the presence of multiple classes. Furthermore, in [10], [11], [23], NN graph-based anomaly detection schemes are proposed with sample-by-sample decisions. As discussed earlier, the sequential decision making is more effective and reliable compared to the sample-by-sample decisions. In [24], [25], [26], two-sample tests are proposed to evaluate whether two datasets have the same distribution, where the test statistics are the distance between the means of the two samples mapped into a kernel space in [24] and the relative entropy, i.e., the Kullback-Leibler (KL) divergence, between the two samples in [25], [26]. Such approaches mainly suffer from low time resolution since they need large sample sizes for reliable decisions.

In [27], an online sliding window-based two-sample test is proposed based on NN graphs and an accurate approximation is presented for its average false alarm period. The method requires to form a new NN graph after each observation and a search over all possible window partitions, that might be prohibitive for real-time processing. In [28], first in an offline phase the high-dimensional nominal observation space is partitioned into several subregions and then the online phase decides, via hypothesis testing, if an incoming batch of observations fall inside the predetermined subregions consistently with the nominal case. Similar to the other window-based approaches, the algorithm is mainly designed for batch processing and hence it cannot operate as fully sequential. In [29], a new interpretation of the CUSUM algorithm based on the discrepancy theory and the GEM method are presented to detect anomalies in real-time, where the presented algorithm asymptotically achieves the CUSUM algorithm under certain conditions, however, no mechanism is provided to control its false alarm rate.

1.3 Contributions

In this paper, we propose real-time nonparametric anomaly detection schemes for high-dimensional data streams. We list our main contributions as follows:

- We propose to extract easy-to-compute univariate summary statistics from the observed high-dimensional data streams, where the summary statistics are useful to distinguish anomalous data from nominal data. We do not impose any restrictive model assumptions for both the observed high-dimensional data stream and the extracted summary statistics. Hence, the proposed schemes are completely nonparametric.
- We propose a low-complexity CUSUM-like real-time anomaly detection algorithm that makes use of the summary statistics.
- We provide an asymptotic lower bound and an asymptotic approximation for the average false alarm period of the proposed algorithm, where the bound and the approximation can be easily controlled by choosing the significance level for outliers and the decision threshold of the proposed algorithm.
- We provide a sufficient condition to (asymptotically) prevent false alarms due to divergence of the decision statistic of the proposed algorithm in the absence of anomalies.

1.4 Organization

The remainder of the paper is organized as follows. We present the problem description and our solution approach in Section 2, derivations of the univariate summary statistics in Section 3, and the proposed real-time anomaly detection schemes in Section 4. We then evaluate the proposed schemes in different application settings via simulations in Section 5. Finally, Section 6 concludes the paper. Throughout the paper, boldface letters denote vectors and matrices, all vectors are column vectors, and $\cdot^{\mathrm{T}}$ denotes the transpose operator.

2 Problem Description and Solution Approach

2.1 Problem Description

We observe a high-dimensional stationary data stream, particularly, at each time t we acquire a new data point $\mathbf{x}_t \in \mathbb{R}^p$ where $p \gg 1$ is the dimensionality of the original data space, also called the ambient dimension, and the data points are independent and identically distributed (i.i.d.) over time. Suppose that an abrupt anomaly, e.g., an unfriendly intervention (attack/intrusion) or an unexpected failure, happens in the observed process at an unknown time τ, called the change-point, and continues thereafter. That is, the process is under regular operating conditions up to time τ and then its underlying statistical properties suddenly change at time τ due to an anomaly. Denoting the pdfs of $\mathbf{x}_t$ under regular (pre-change) and anomalous (post-change) conditions as $f_0^{\mathbf{x}}$ and $f_1^{\mathbf{x}} \neq f_0^{\mathbf{x}}$, respectively, we have

$$\mathbf{x}_t \sim \begin{cases} f_0^{\mathbf{x}}, & \text{if } t < \tau \\ f_1^{\mathbf{x}}, & \text{if } t \geq \tau. \end{cases}$$

Our goal is to detect changes (anomalies) with minimal possible delays and also with minimal rates of false alarm for a secure and reliable operation of the observed system. In other words, we aim to detect the changes as quickly as possible after they occur. The framework of quickest detection well matches with this purpose. A well-known problem formulation in the quickest detection framework is the minimax problem proposed by Lorden [30]. In the minimax problem, the goal is to minimize the worst-case detection delay subject to false alarm constraints. More specifically, let Γ denote the stopping time at which a change is declared and $\mathbb{E}_\tau$ denote the expectation measure if the change happens at time τ. The Lorden's worst-case average detection delay is given by

$$J(\Gamma) \triangleq \sup_{\tau} \operatorname*{ess\,sup}_{\mathcal{F}_\tau} \mathbb{E}_\tau\big[(\Gamma-\tau)^+ \,|\mathcal{F}_\tau\big], \tag{1}$$

where $(\cdot)^+ = \max\{0,\cdot\}$, $\mathcal{F}_\tau$ is the history of observations up to the change-point τ, and $\operatorname{ess\,sup}$ denotes the essential supremum, a concept in measure theory, which is practically equivalent to the supremum of a set. $J(\Gamma)$ is called the worst-case delay since it is computed based on the least favorable change-point and the least favorable history of observations up to the change-point. The minimax problem can then be written as follows:

$$\inf_{\Gamma}\; J(\Gamma) \;\; \text{subject to} \;\; \mathbb{E}_\infty[\Gamma] \geq \beta, \tag{2}$$

where $\mathbb{E}_\infty[\Gamma]$ is the average false alarm period, i.e., the average stopping time when no change occurs at all ($\tau = \infty$), and β is the desired lower bound on the average false alarm period.

If both $f_0^{\mathbf{x}}$ and $f_1^{\mathbf{x}}$ are known, then the well-known CUSUM algorithm is the optimal solution to the minimax problem given in (2) [31]. Let

$$\ell_t \triangleq \log\left(\frac{f_1^{\mathbf{x}}(\mathbf{x}_t)}{f_0^{\mathbf{x}}(\mathbf{x}_t)}\right),$$

denote the LLR at time t. In the CUSUM algorithm, the LLR is considered as the statistical evidence for change at a time and the LLRs are accumulated over time. If the accumulated evidence exceeds a predefined threshold, then a change is declared. Denoting the CUSUM decision statistic at time t by g_t and the decision threshold by h, the CUSUM algorithm is given by

$$\begin{aligned} \Gamma &= \inf\{t : g_t \geq h\}, \\ g_t &= \max\{0, g_{t-1} + \ell_t\}, \end{aligned} \tag{3}$$

where $g_0 = 0$.

Since it is practically difficult to model all types of anomalies, $f_1^{\mathbf{x}}$ needs to be assumed unknown for a general anomaly detection problem. In that case, if only $f_0^{\mathbf{x}}$ is known and also has a parametric form, slight deviations from the parameters of $f_0^{\mathbf{x}}$ can be detected using a generalized CUSUM algorithm [5, Section 5.3], [9], [32]. However, in a general high-dimensional problem, it might be difficult to model or estimate the high-dimensional multivariate nominal pdf $f_0^{\mathbf{x}}$. Hence, in this study, we assume that both $f_0^{\mathbf{x}}$ and $f_1^{\mathbf{x}}$ are unknown. We propose to use an alternative technique in that we extract useful univariate summary statistics from the observed high-dimensional data stream and perform the anomaly detection task in a single-dimensional space based on the extracted summary statistics, as detailed below.

2.2 Proposed Solution Approach

First, we assume that there is an available set of nominal data points $\mathcal{X} \triangleq \{\mathbf{x}_i : i = 1, 2, \dots, N\}$, that are free of anomaly. Practically, this is, in general, possible since the monitored system/process produces a data point at each sampling instant and a set of nominal data points can be obtained under regular system operation (no anomaly). Using $\mathcal{X}$, we aim to extract univariate baseline statistics that summarize the regular system operation such that the summary statistics corresponding to anomalous data deviate from the baseline statistics. To this end, summary statistics should be well informative to distinguish anomalous conditions from the regular operating conditions.

Let the summary statistic corresponding to $\mathbf{x}_t$ be denoted by d_t. Since the statistical properties of $\mathbf{x}_t$ changes at time τ, we assume that the statistical properties of d_t also changes at τ. Denoting the nominal and anomalous pdfs of d_t as f_0^d and $f_1^d \neq f_0^d$, respectively, we then have

$$d_t \sim \begin{cases} f_0^d, & \text{if } t < \tau \\ f_1^d, & \text{if } t \geq \tau, \end{cases}$$

where we assume that f_0^d and f_1^d are both unknown. Nonetheless, extracting a set of nominal summary statistics from $\mathcal{X}$ and using this set as i.i.d. realizations of the nominal pdf f_0^d, we can form an empirical distribution function (edf) of the nominal summary statistics that estimates the nominal cumulative distribution function (cdf) F_0^d of d_t. Then, based on the nominal edf of the summary statistics, for an incoming data point $\mathbf{x}_t$ at time t and its corresponding summary statistic d_t, we can estimate the corresponding tail probability (p-value), denoted with p_t. In statistical outlier detection, a data point $\mathbf{x}_t$ is considered as an outlier with respect to the level of α if its p-value is less than α, i.e., $p_t < \alpha$. Let

$$s_t \triangleq \log\left(\frac{\alpha}{p_t}\right). \tag{4}$$

Then, for an outlier $\mathbf{x}_t$, we have $s_t > 0$ and similarly, for a non-outlier $\mathbf{x}_t$, we have $s_t \leq 0$.

Under normal system operation, we may observe random non-persistent outliers due to e.g., high-level random system noise. However, if a system produces persistent outliers, then this may indicate an actual anomaly. Hence, we can model anomalies as persistent outliers. Considering s_t in (4) as a positive/negative statistical evidence for anomaly at time t, we can accumulate s_t's over time and obtain an accumulated evidence for anomaly. We can then declare an anomaly only if we have a strong (reliable) evidence supporting an anomaly. This gives rise to the following CUSUM-like anomaly detection algorithm where we replace the LLR ℓ_t in the CUSUM algorithm (see (3)) with s_t:

$$\begin{aligned} \Gamma &= \inf\{t : g_t \geq h\}, \\ g_t &= \max\{0, g_{t-1} + s_t\}, \end{aligned} \tag{5}$$

where $g_0 = 0$.

In the following section, we present derivations of the proposed summary statistics. Then, in Section 4, we explain the estimation of the tail probability p_t (and hence s_t) based on the nominal summary statistics, that results in the final proposed detection algorithm.

3 Summary Statistics

In this section, we first explain our methodology to derive summary statistics for a general high-dimensional data stream. We then explain the derivation of summary statistics in a specific case where the observed data exhibit a low intrinsic dimensionality.

3.1 GEM-Based Summary Statistics

Given a nominal dataset $\mathcal{X}$ and a chosen significance level of α, the GEM method [10] determines an acceptance region $\mathcal{A}$ for the nominal data based on the asymptotic theory of random euclidean graphs such that if a data point falls outside $\mathcal{A}$, it is considered as an outlier with respect to the level α, otherwise considered as a non-outlier. The GEM method is based on the NN statistics that capture the local interactions between data points governed by the underlying statistical properties of the observed data stream.

A computationally efficient GEM method presented in [11] is based on bipartite kNN graphs (BP-GEM). In this method, first $\mathcal{X}$ is uniformly randomly partitioned into two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ with sizes N_1 and $N_2 = N - N_1$, respectively. Then, for each data point $\mathbf{x}_j \in \mathcal{S}_2$, the kNNs of $\mathbf{x}_j$ among the set $\mathcal{S}_1$ are determined. Denoting the euclidean distance of $\mathbf{x}_j$ to its ith NN in $\mathcal{S}_1$ by $e_j(i)$, the sum of distances of $\mathbf{x}_j$ to its kNNs can be written as follows:

$$d_j \triangleq \sum_{i=1}^{k} e_j(i). \tag{6}$$

After computing $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$, d_j's are sorted in ascending order and the $(1-\alpha)$ fraction of $\mathbf{x}_j$'s in $\mathcal{S}_2$ corresponding to the smallest $(1-\alpha)$ fraction of d_j's form the acceptance region $\mathcal{A}$. Then, for a new data point $\mathbf{x}_t$, if its sum of distances to its kNNs among $\mathcal{S}_1$, denoted with d_t, is greater than the smallest $(1-\alpha)$ fraction of d_j's, i.e.,

$$\frac{\sum_{\mathbf{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_t > d_j\}}{N_2} > 1 - \alpha,$$

then $\mathbf{x}_t$ is considered as an outlier with respect to the level of α, where $\mathbb{1}\{\cdot\}$ denotes an indicator function.

Let $\delta(\cdot)$ be the Lebesgue measure in $\mathbb{R}^p$. As $k/N_1 \to 0$ and $k, N_2 \to \infty$, the acceptance region $\mathcal{A}$ determined by the BP-GEM method almost surely converges to the minimum volume set of level α [11], given by

$$\Lambda_\alpha \triangleq \min\left\{\delta(\mathcal{A}) : \int_{\mathbf{z}\in\mathcal{A}} f_0^{\mathbf{x}}(\mathbf{z}) d\mathbf{z} \geq 1 - \alpha\right\},$$

where $\delta(\mathcal{A})$ denotes the volume of $\mathcal{A}$. Moreover, if $f_0^{\mathbf{x}}$ is a Lebesgue density, the minimum volume set and the minimum Rényi entropy set are equivalent [11]. Hence, the BP-GEM method asymptotically achieves the minimum entropy set, i.e., the most compact acceptance region for the nominal data.

If $\mathbf{x}_t$ is an outlier, then it falls outside the acceptance region $\mathcal{A}$, i.e., the corresponding NN statistic d_t takes a higher value compared to non-outliers. Moreover, if the observed data stream persistently fall outside the acceptance region, or equivalently if we persistently observe high NN statistics over time, then this may indicate an anomaly. Hence, we can use the GEM-based NN statistic as a summary statistic to distinguish anomalous data from nominal data. Moreover, we can use $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$ as a set of GEM-based nominal summary statistics.

A salient feature of extracting summary statistics based on the BP-GEM method is that with the incoming data points in an online setting, there is no need to recompute the NN graph for the entire dataset. This is because for each data point, either newly acquired or belonging to the set $\mathcal{S}_2$, the NNs are always searched among the time-invariant set $\mathcal{S}_1$. Hence, obtaining new data does not alter the NNs of the points in $\mathcal{S}_2$. In the online phase, the main computational complexity is then searching the NNs of incoming data points among the set $\mathcal{S}_1$. To further reduce the complexity, fast NN search algorithms can be employed to approximately determine the NNs, see e.g., [33].

Finally, since we capture local interactions between data points via their kNNs, k should not be chosen too large. On the other hand, since the set $\mathcal{S}_1$ might contain some outliers, an incoming data point might fall geometrically close to a few of such outliers. Then, k should not be chosen too small in order to reduce the risk of evaluating an outlier or anomalous data point as a non-outlier. Therefore, a moderate k value should best fit to our purpose of extracting useful GEM-based summary statistics for anomaly detection.

3.2 Summary Statistics for High-Dimensional Data Exhibiting Low Intrinsic Dimensionality

In many practical applications, observed high-dimensional data exhibits a sparse structure so that the intrinsic dimensionality of the data is lower than the ambient dimension, and hence the data can be well represented in a lower-dimensional subspace. In such cases, we can model the data as follows:

$$\mathbf{x}_t = \mathbf{y}_t + \mathbf{r}_t, \tag{7}$$

where $\mathbf{y}_t$ is the representation of $\mathbf{x}_t$ in a submanifold and $\mathbf{r}_t$ is the residual term, i.e., the departure of $\mathbf{x}_t$ from the submanifold, mostly consisting of noise.

Suppose that we learn a submanifold that the nominal data are embedded in. Since the learned manifold is mainly representative for the nominal data, anomalous data points are expected to deviate from the nominal submanifold and hence the magnitude of the residual term, i.e., $\|\mathbf{r}_t\|_2$, is expected to take higher values for anomalous data compared to nominal data. Hence, the magnitude of the residual term can be used as a summary statistic to distinguish anomalous data. Given a nominal dataset $\mathcal{X}$, let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two subsets of $\mathcal{X}$, i.e., $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{X}$, with sizes N_1 and N_2, respectively, where $N_1, N_2 \leq N$. First, using $\mathcal{S}_1$, we can determine a representative submanifold that the nominal data are embedded in. Then, using $\mathcal{S}_2$, we can compute the magnitude of the residual terms, i.e., $\{\|\mathbf{r}_j\|_2 : \mathbf{x}_j \in \mathcal{S}_2\}$, that can be used as a set of nominal summary statistics.

There are various methods to determine the underlying submanifold, among which the PCA is well known for finding a linear submanifold, called the principal subspace [34, Section 12.1]. Next, we explain the PCA and the PCA-based summary statistics.

The PCA is a nonparametric linear submanifold learning technique as it is computed directly from a given dataset without requiring any data model. Given a set of nominal data points $\mathcal{S}_1$, the PCA provides a linear subspace with dimensionality $r \leq p$ such that (i) the variance of the projected data onto the r-dimensional subspace is maximized and (ii) the sum of squares of the projection errors (residual magnitudes) is minimized [34, Section 12.1].

In the PCA method, denoting $\bar{\mathbf{x}}$ as the sample mean, i.e.,

$$\bar{\mathbf{x}} \triangleq \frac{1}{N_1} \sum_{\mathbf{x}_i \in \mathcal{S}_1} \mathbf{x}_i, \tag{8}$$

and $\mathbf{Q}$ as the sample data covariance matrix, i.e.,

$$\mathbf{Q} \triangleq \frac{1}{N_1} \sum_{\mathbf{x}_i \in \mathcal{S}_1} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^{\mathrm{T}}, \tag{9}$$

first, the eigenvalues $\{\lambda_j : j = 1, 2, \ldots, p\}$ and the eigenvectors $\{\mathbf{v}_j : j = 1, 2, \ldots, p\}$ of $\mathbf{Q}$ are computed, where

$$\mathbf{Q}\,\mathbf{v}_j = \lambda_j \mathbf{v}_j, \quad j = 1, 2, \ldots, p.$$

Then, the dimensionality of the submanifold, r, can be determined based on the desired fraction of data variance retained in the submanifold, given by

$$\gamma \triangleq \frac{\sum_{j=1}^{r} \lambda_j}{\sum_{j=1}^{p} \lambda_j} \leq 1, \tag{10}$$

where the r-dimensional principal subspace is spanned by the orthonormal eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ corresponding to the r largest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_r$ of $\mathbf{Q}$. Let $\mathbf{V} \triangleq [\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_r]$. The representation of $\mathbf{x}_t$ in the linear submanifold can then be determined as follows:

$$\begin{aligned} \mathbf{y}_t &= \bar{\mathbf{x}} + \sum_{j=1}^{r} \mathbf{v}_j \mathbf{v}_j^{\mathrm{T}} (\mathbf{x}_t - \bar{\mathbf{x}}) \\ &= \bar{\mathbf{x}} + \mathbf{V}\mathbf{V}^{\mathrm{T}} (\mathbf{x}_t - \bar{\mathbf{x}}). \end{aligned} \tag{11}$$

Then, the residual term can be computed as

$$\begin{aligned} \mathbf{r}_t &= \mathbf{x}_t - \mathbf{y}_t \\ &= (\mathbf{I}_p - \mathbf{V}\mathbf{V}^{\mathrm{T}})(\mathbf{x}_t - \bar{\mathbf{x}}), \end{aligned} \tag{12}$$

where $\mathbf{I}_p \in \mathbb{R}^{p \times p}$ is an identity matrix.

To obtain the PCA-based nominal summary statistics, first, using $\mathcal{S}_1$, we can compute $\mathbf{Q}$ based on (9), and then its eigenvalues and eigenvectors. Then, for a chosen γ (see (10)), we can determine r and the corresponding $\mathbf{V}$. Finally, using $\mathcal{S}_2$ and (12), we can compute $\{\|\mathbf{r}_j\|_2 : \mathbf{x}_j \in \mathcal{S}_2\}$, that forms a set of nominal PCA-based summary statistics.

Note that although here we have only focused on the PCA and the linear submanifolds, using the same data model in (7) and following a similar methodology, summary statistics can be extracted for any (possibly nonlinear) manifold learning algorithm as long as it is appropriate for the observed high-dimensional data stream and it allows for efficient computation of the residual terms $\mathbf{r}_t$ (see (7)) both for a given nominal dataset and also for the sequentially acquired out-of-sample data, without re-running the manifold learning algorithm.

4 Real-Time Nonparametric Anomaly Detection

4.1 Proposed Algorithm

We first discuss the statistical outlier detection based on a set of nominal summary statistics. Notice that for outliers, both of the proposed summary statistics, d_t and $\|\mathbf{r}_t\|_2$, take higher values compared to non-outliers (see Section 3). Hence, outliers in fact correspond to the right tail events based on the nominal pdf of the summary statistics. Let us specifically consider d_t. In case the knowledge of the nominal pdf of d_t, i.e., f_0^d, is available, we would compute the corresponding right tail probability as follows:

$$p_t = \int_{d_t}^{\infty} f_0^d(z)dz = 1 - F_0^d(d_t), \tag{13}$$

where F_0^d is the cdf of d_t. If $p_t < \alpha$, we can then consider d_t (correspondingly $\mathbf{x}_t$) as an outlier with respect to the significance level α.

In our problem, although we do not have the knowledge of f_0^d (and F_0^d), using a set of i.i.d. realizations of the nominal summary statistics, we can obtain an edf that estimates F_0^d. Let $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$ be the set of nominal summary statistics. Then, the corresponding edf is given by

$$\hat{F}_{0,N_2}^d(z) \triangleq \frac{1}{N_2} \sum_{\mathbf{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j \leq z\}. \tag{14}$$

Moreover, by the Glivenko-Cantelli theorem, $\hat{F}_{0,N_2}^d$ pointwise almost surely converges to the actual cdf F_0^d as $N_2 \to \infty$ [35]. Then, we can estimate p_t based on $\hat{F}_{0,N_2}^d$ as follows:

$$\begin{aligned} \hat{p}_t &= 1 - \hat{F}_{0,N_2}^d(d_t) \\ &= \frac{1}{N_2} \sum_{\mathbf{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j > d_t\}. \end{aligned} \tag{15}$$

That is, $\hat{p}_t$ is simply the fraction of the nominal summary statistics $\{d_j : \mathbf{x}_j \in S_2\}$ greater than d_t. If $\hat{p}_t < \alpha$, then we can consider $\mathbf{x}_t$ as an outlier with respect to the level of α.

Let

$$\hat{s}_t \triangleq \log\left(\frac{\alpha}{\hat{p}_t}\right). \tag{16}$$

Notice that for an outlier $\mathbf{x}_t$ with respect to a level of α, we have $\hat{s}_t > 0$ and similarly, for a non-outlier $\mathbf{x}_t$, we have $\hat{s}_t \leq 0$. Then, by replacing $\hat{s}_t$ with s_t in (5), we propose the following model-free CUSUM-like anomaly detection algorithm:

$$\Gamma = \inf\{t : g_t \geq h\}, g_t = \max\{0, g_{t-1} + \hat{s}_t\}, \tag{17}$$

where $g_0 = 0$[1]. Since we consider anomalies as equivalent to persistent outliers, the decision statistic g_t has a positive drift in case of an anomaly and a non-positive drift in the absence of anomalies.

Algorithm 1. GEM-Based Real-Time Nonparametric Anomaly Detection

Offline Phase

Uniformly randomly partition the nominal dataset $\mathcal{X}$ into two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ with sizes N_1 and N_2, respectively.
for $j : \mathbf{x}_j \in \mathcal{S}_2$ **do**
  Search for the kNNs of $\mathbf{x}_j$ among the set $\mathcal{S}_1$.
  Compute d_j using (6).
end for
Sort $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$ in ascending order.

Online Detection Phase

Initialization: $t \leftarrow 0$, $g_0 \leftarrow 0$.
while $g_t < h$ **do**
  $t \leftarrow t + 1$.
  Obtain the new data point $\mathbf{x}_t$.
  Search for the kNNs of $\mathbf{x}_t$ among the set $\mathcal{S}_1$ and compute d_t using (6).
  $\hat{p}_t = \frac{1}{N_2}\sum_{\mathbf{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j > d_t\}$.
  $\hat{s}_t = \log(\alpha/\hat{p}_t)$.
  $g_t \leftarrow \max\{0, g_{t-1} + \hat{s}_t\}$.
end while
Declare an anomaly and stop the procedure.

Algorithm 2. PCA-Based Real-Time Nonparametric Anomaly Detection

Offline Phase

Choose subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ of $\mathcal{X}$ with sizes N_1 and N_2, respectively.
Compute $\bar{\mathbf{x}}$ and $\mathbf{Q}$ over $\mathcal{S}_1$ using (8) and (9), respectively.
Compute the eigenvalues $\{\lambda_j : j = 1, 2, \ldots, p\}$ and the eigenvectors $\{\mathbf{v}_j : j = 1, 2, \ldots, p\}$ of $\mathbf{Q}$.
Based on a desired level of γ (see (10)), determine r and form the matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_r]$.
for $j : \mathbf{x}_j \in \mathcal{S}_2$ **do**
  $\mathbf{r}_j = (\mathbf{I}_p - \mathbf{V}\mathbf{V}^{\mathrm{T}})(\mathbf{x}_j - \bar{\mathbf{x}})$.
  Compute $\|\mathbf{r}_j\|_2$.
end for
Sort $\{\|\mathbf{r}_j\|_2 : \mathbf{x}_j \in \mathcal{S}_2\}$ in ascending order.

Online Detection Phase

Initialization: $t \leftarrow 0$, $g_0 \leftarrow 0$.
while $g_t < h$ **do**
  $t \leftarrow t + 1$.
  Obtain the new data point $\mathbf{x}_t$.
  $\mathbf{r}_t = (\mathbf{I}_p - \mathbf{V}\mathbf{V}^{\mathrm{T}})(\mathbf{x}_t - \bar{\mathbf{x}})$ and compute $\|\mathbf{r}_t\|_2$.
  $\hat{p}_t = \frac{1}{N_2}\sum_{\mathbf{x}_j \in \mathcal{S}_2} \mathbb{1}\{\|\mathbf{r}_j\|_2 > \|\mathbf{r}_t\|_2\}$.
  $\hat{s}_t = \log(\alpha/\hat{p}_t)$.
  $g_t \leftarrow \max\{0, g_{t-1} + \hat{s}_t\}$.
end while
Declare an anomaly and stop the procedure.

We summarize the proposed GEM-based and PCA-based detection schemes in Algorithms 1 and 2, respectively. Moreover, we present a diagram of the proposed schemes in Fig. 1. The proposed schemes consist of an offline phase for extracting the baseline statistics for a given set of nominal data points and an online phase for anomaly detection. The offline phases are explained in Section 3. In

1. In case where $\sum_{\mathbf{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j > d_t\} = 0$, we have $\hat{p}_t = 0$ (see (15)), and hence $g_t = \infty$. In this case, a small nonzero value, e.g., $1/N_2$, can be assigned to $\hat{p}_t$ in order to prevent the decision statistic to raise to infinity due to a single outlier. This modification can be useful to reduce the false alarm rate especially in the small-sample settings (small N_2).

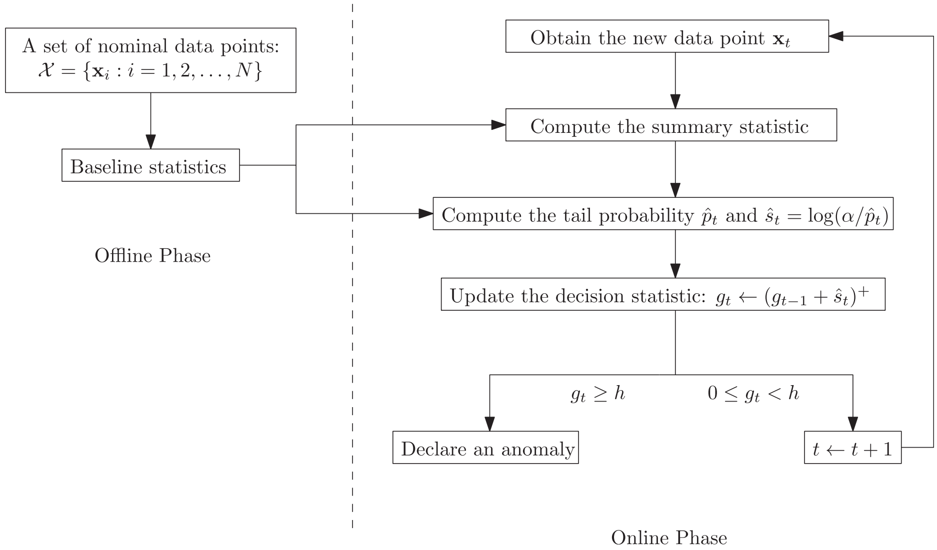

Fig. 1. Diagram of the proposed detection schemes.

the online phase, at each time t, a new data point $\mathbf{x}_t$ is observed and using the baseline statistics, the summary statistic corresponding to $\mathbf{x}_t$ is computed and then the tail probability $\hat{p}_t$ and the statistical evidence $\hat{s}_t$ are estimated. The decision statistic g_t is then updated and if it exceeds the predetermined decision threshold h, an anomaly is declared, otherwise the algorithm proceeds to the next time interval and acquires a further observation. Note that the proposed detection mechanism is generic in the sense that after extracting useful summary statistics for nominal data and computing an edf for the nominal summary statistics, the proposed CUSUM-like algorithm in (17) can be employed for real-time anomaly detection.

4.2 Analysis

4.2.1 False Alarm Rate

Under regular conditions (no anomaly), if the decision statistic g_t exceeds the test threshold h, then a false alarm occurs. In anomaly detection, false alarm is an undesired event and for reliability of an anomaly detection scheme, performance guarantees regarding the false alarm rate are often desirable. With this purpose, first the following theorem provides an asymptotic upper bound on the level of α such that in the absence of anomalies, the decision statistic g_t (almost surely) does not diverge in the mean squared sense.

Theorem 1. *In the absence of anomalies, i.e., $\tau = \infty$, if $\alpha < 1/e$, where e denotes the Euler's number, we have as $N_2 \to \infty$,*

$$\mathbb{P}\left(\sup_{t\geq 0} \; \mathbb{E}\left[g_t^2 \,|\, g_0 = 0\right] \; < \; \infty\right) = 1,$$

i.e., the decision statistic does not grow unbounded in the mean squared sense, with the probability 1.

Proof. See Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.2970410. □

Theorem 1 provides a guidance to choose the level of α to reliably employ the proposed algorithm. Specifically, α can be chosen smaller than $1/e$ to asymptotically ensure that the decision statistic of the proposed algorithm stays finite over time under regular conditions, that eliminates false alarms due to the divergence of the decision statistic.

In the proposed CUSUM-like algorithm given in (17), the decision statistic at time t, g_t, is determined by $\{\hat{s}_n : n \leq t\}$ where $\hat{s}_n$'s are i.i.d. over time in the absence of anomalies. Hence, we have actually a random walk driven by $\{\hat{s}_n\}$ with lower threshold 0 and upper (decision) threshold h and our aim is to determine the average false alarm period (also called the average run length), i.e., the first time, on average, the upper threshold h is crossed in the no-anomaly case. In the literature, this problem has been considered in several studies and some approximations and bounds are provided for this quantity as the exact computation is analytically intractable [5, Section 5.2.2], [36], [37], [38], [39]. To be able to provide a performance guarantee regarding the false alarm rate, we first derive an asymptotic lower bound on the average false alarm period of the proposed algorithm, as stated in the following theorem.

Theorem 2. *For chosen $0 < \alpha < 1/e$ and $h > 0$, the average false alarm period of the proposed algorithm, $\mathbb{E}_\infty[\Gamma]$, asymptotically (as $N_2 \to \infty$) achieves the following lower bound:*

$$\mathbb{E}_\infty[\Gamma] \geq e^{(1-\theta)h}, \tag{18}$$

where $0 < \theta < 1$ is uniquely given by

$$\theta = \frac{W(\alpha \log(\alpha))}{\log(\alpha)}, \tag{19}$$

and $W(c)$ denotes the Lambert-W function[2] providing solutions z to the equation $z\,e^z = c$.

Proof. See Appendix B, available in the online supplemental material. □

Based on Theorem 2, α and h can be chosen to asymptotically satisfy the minimum acceptable level of average false

2. There is a built-in MATLAB function **lambertw**.

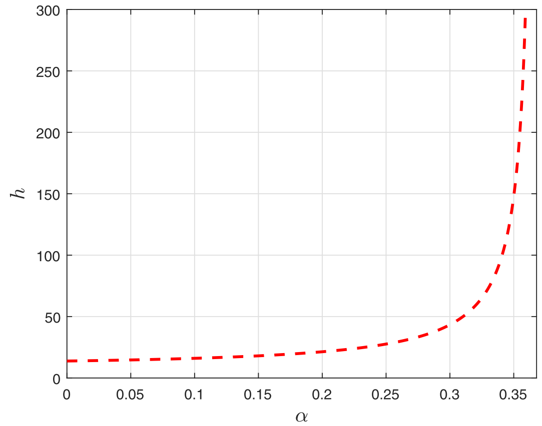

Fig. 2. The lower bound (dashed curve) on the decision threshold h of the proposed algorithm for $\alpha < 1/e$ such that $\mathbb{E}_\infty[\Gamma] \geq 10^6$ as $N_2 \to \infty$.

alarm period. Specifically, if the desired lower bound is $L > 0$, then

$$\mathbb{E}_\infty[\Gamma] \geq e^{(1-\theta)h} \geq L,$$

which is equivalent to

$$h \geq \frac{\log(L)}{1 - W(\alpha \log(\alpha))/\log(\alpha)},$$

providing a lower bound on the test threshold h for a chosen level of α. As an example, for $L = 10^6$, Fig. 2 illustrates the lower bound on h for $0 < \alpha < 1/e$.

Next, we investigate the tightness of the presented lower bound on the average false alarm period in the asymptotic regime (as $N_2 \to \infty$). In particular, for different α levels, by varying the test threshold h, we plot the average false alarm period versus the presented lower bound in Fig. 3. The figure shows that the average false alarm period is approximately linear with the lower bound, where the ratio between them depends on the level of α. Based on this observation, for a chosen $0 < \alpha < 1/e$ and $h > 0$, we can make the following (asymptotic) approximation to the average false alarm period:

$$\mathbb{E}_\infty[\Gamma] \approx g(\alpha)\, e^{(1-\theta)h}, \tag{20}$$

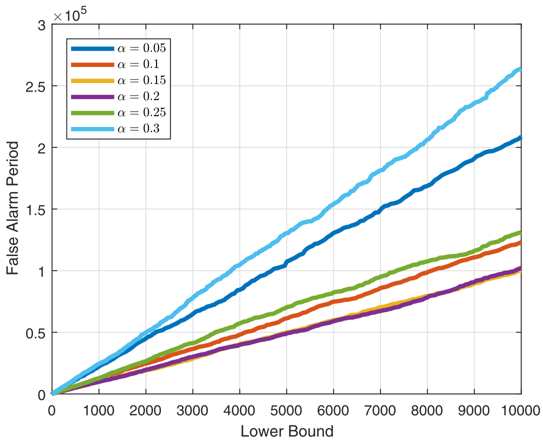

Fig. 3. Average false alarm period versus the presented lower bound in the asymptotic regime where $N_2 \to \infty$.

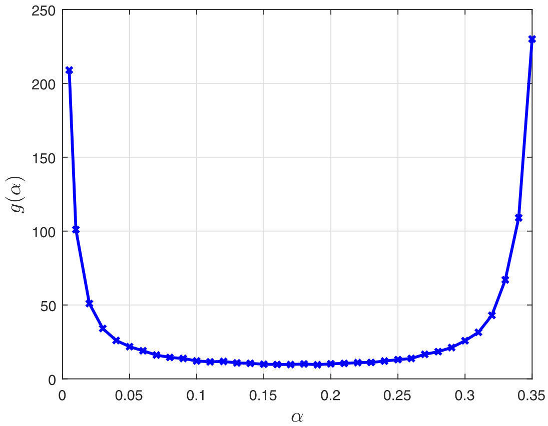

Fig. 4. $g(\alpha)$ versus α.

where θ is as given in (19) and $g(\alpha)$ numerically computed over a Monte Carlo simulation is given in Fig. 4 and Table 1 for some α levels. Then, for a chosen $0 < \alpha < 1/e$ and for a desired average false alarm period A, the test threshold h can be chosen as

$$h = \frac{\log(A/g(\alpha))}{1 - W(\alpha \log(\alpha))/\log(\alpha)},$$

that asymptotically makes

$$\mathbb{E}_\infty[\Gamma] \approx A.$$

Finally, note that lower α and/or higher h lead to larger false alarm periods and also larger detection delays. This is because lower α results in lower $\hat{s}_t$ and hence lower g_t, that increases the stopping time Γ (see (16) and (17)). Similarly, higher h results in a larger stopping time (see (17)). Hence, α and h are essentially tradeoff parameters that can be used to strike a desired balance between the false alarm rate and the average detection delay of the proposed algorithm. However, since the post-change (anomalous) case is totally unknown and no anomalous data is available, it seems difficult to provide theoretical results regarding the average detection delay of the proposed algorithm. Nonetheless, we know that as the discrepancy, e.g., the KL divergence, between the nominal and anomalous pdfs increases, it is likely to observe more significant outliers after anomaly happens, that increases $\hat{s}_t$ (see (15) and (16)) for $t \geq \tau$, which in turn decreases the detection delays (see (17)).

Remark 1. For the proposed CUSUM-like test, we also derive the Wald's approximation to the average false alarm period in the asymptotic regime where $N_2 \to \infty$. In particular, based on [5, Section 5.2.2.2] and with derivations similar to Theorems 1 and 2, for a chosen $0 < \alpha < 1/e$ and $h > 0$ and as $N_2 \to \infty$, we can state the Wald's approximation as follows:

TABLE 1
$g(\alpha)$ Computed Over a Monte Carlo Simulation for Some α Levels

α	0.01	0.05	0.1	0.15	0.2	0.25	0.3	0.35
$g(\alpha)$	101	21.8	12.1	9.9	10.1	13	25.8	230

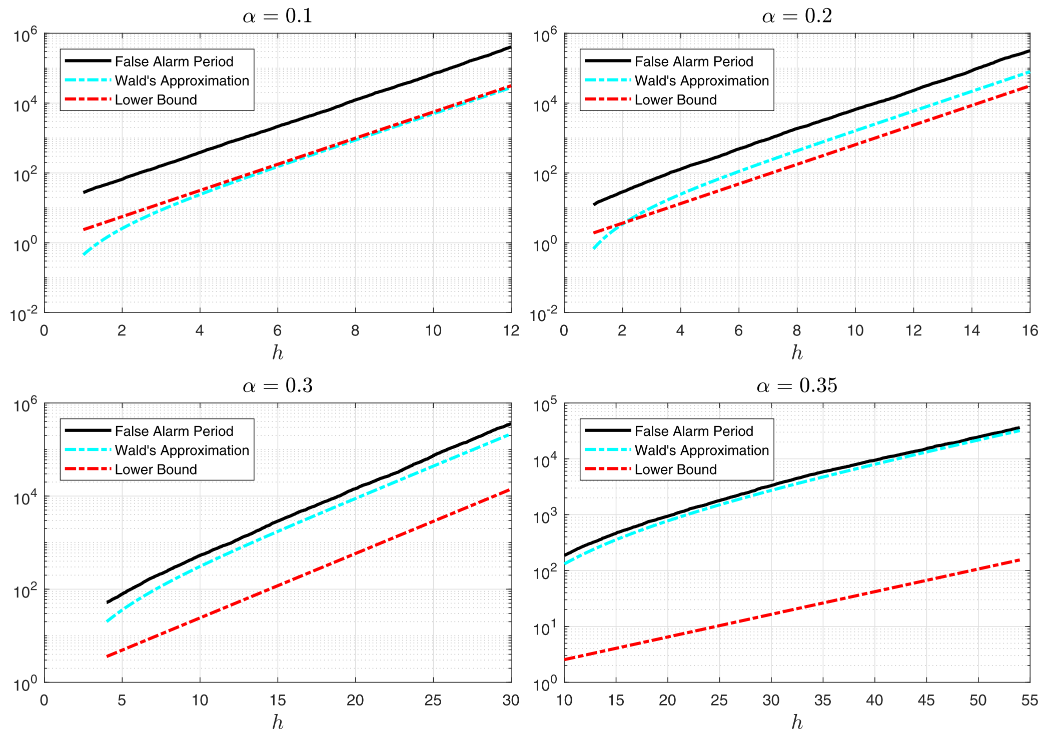

Fig. 5. Average false alarm period, the Wald's approximation, and the presented lower bound for various α and h levels.

$$\mathbb{E}_\infty[\Gamma] \approx \frac{1}{1+\log(\alpha)}\left(h + \frac{e^{(1-\theta)h}-1}{\theta-1}\right). \tag{21}$$

However, since the Wald's approximation ignores the excess over boundary (overshoot), it significantly underestimates the false alarm period as α decreases towards 0, as for smaller α, the decision statistic g_t of the proposed CUSUM-like test more frequently hits to the lower threshold 0 during the random walk. On the other hand, the approximation gets better as α increases towards $1/e$. Fig. 5shows the average false alarm period, the Wald's approximation given in (21), and the lower bound presented in Theorem 2 for various α and h levels obtained via simulations in the asymptotic regime as $N_2 \to \infty$. As expected, the approximation gets worse as α decreases, e.g., it becomes even lower than the presented lower bound for $\alpha = 0.1$.

4.2.2 Space and Time Complexity

Algorithm 1: In the offline phase, Algorithm 1 computes and stores $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$ and for the computation of any d_j, it stores the smallest k distances, i.e., $e_j(i)$'s, see (6). Hence, the space complexity of the offline phase is $O(k+N_2)$. For each $\mathbf{x}_j \in \mathcal{S}_2$, the algorithm computes its Euclidean distance to each data point in $\mathcal{S}_1$ and computes the corresponding d_j by summing up the smallest k distances. The algorithm finally sorts the set $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$. Hence, the time complexity of the offline phase is $O(N_2(N_1 p + k + \log(N_2)))$. In the online phase, the space complexity is $O(k)$. Moreover, at each time t, the time complexity of the online phase is $O(N_1 p + k + \log(N_2))$. The term $\log(N_2)$ is because the computation of $\hat{p}_t$ requires to determine how many of $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$ are larger than d_t, which is equivalent to determine the position of d_t among the sorted set of $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$.

Algorithm 2. The offline phase of Algorithm 2 requires the storage of $\{\|\mathbf{r}_j\|_2 : \mathbf{x}_j \in \mathcal{S}_2\}$ and $(\mathbf{I}_p - \mathbf{V}\mathbf{V}^{\mathrm{T}})$. Hence, the space complexity is $O(p^2 + N_2)$. Due to the computation of the sample covariance matrix, the eigenvalue decomposition, the computation of $\|\mathbf{r}_j\|_2$ for each $\mathbf{x}_j$ in $\mathcal{S}_2$, and sorting them out, the time complexity of the offline phase is $O(p^3 + p^2(N_1+N_2) + N_2\log(N_2))$. The online phase requires the storage of $\mathbf{r}_t$ to compute $\|\mathbf{r}_t\|_2$ at any time t. Hence, the space complexity is $O(p)$. Moreover, for an incoming data point $\mathbf{x}_t$, $\|\mathbf{r}_t\|_2$ is computed and its position in the sorted set of $\{\|\mathbf{r}_j\|_2 : \mathbf{x}_j \in \mathcal{S}_2\}$ is determined, which has a time complexity of $O(p^2 + \log(N_2))$ at each time t.

Table 2 summarizes the space and time complexity of the proposed algorithms.

5 Performance Evaluation

In this section, we evaluate the performance of the proposed detection schemes using both synthetic and real data[3]. In particular, we evaluate the GEM-based scheme in detection of cyber-attacks targeting the smart grid. Moreover, we evaluate both the GEM-based and the PCA-based schemes in detection of changes in human physical activity and botnet attacks in an IoT network. Throughout the section, we choose $\alpha = 0.2$ and make the aforementioned modification for the proposed algorithms: in case where $\sum_{\mathbf{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j > d_t\} = 0$, we assign $\hat{p}_t = 1/N_2$ to prevent $g_t = \infty$ due to a single outlier. For all the proposed and benchmark tests, we obtain the tradeoff curves between the average detection delay, $\mathbb{E}_\tau[(\Gamma-\tau)^+]$, and the

3. The MATLAB codes for our experiments are available at https://github.com/mnecipkurt/pami20.

TABLE 2
Space and Time Complexity of the Proposed Algorithms

		Space	Time
Algorithm 1	Offline	$O(k + N_2)$	$O(N_2(N_1p + k + \log(N_2)))$
	Online	$O(k)$	$O(N_1p + k + \log(N_2))$
Algorithm 2	Offline	$O(p^2 + N_2)$	$O(p^3 + p^2(N_1 + N_2) + N_2\log(N_2))$
	Online	$O(p)$	$O(p^2 + \log(N_2))$

average false alarm period, $\mathbb{E}_\infty[\Gamma]$, by varying their test threshold h. In computing the detection delays, we assume that anomalies happen at $\tau = 1$, that corresponds to the worst-case detection delay for the proposed algorithms since the decision statistic g_t is equal to zero just before the anomalies happen (recall that $g_0 = 0$).

We also present the receiver operating characteristic (ROC) curves for all tests by varying their test thresholds. As the computation of the true positive rate (TPR), also called recall, requires to count the number of detected and missed trials, we need to define an upper bound on the detection latency, i.e., the maximum acceptable detection delay, such that if the anomaly/change is detected within this bound, we assume it is detected successfully, otherwise missed. In the simulations, as an example, we choose this bound as 10 time units. We then compute the TPR out of 10000 trials via Monte Carlo simulations as follows:

$$\mathrm{TPR} = \frac{\#\ \text{trials}\ (\tau \le \Gamma \le \tau + 10)}{\#\ \text{trials}\ (\tau \le \Gamma \le \tau + 10) + \#\ \text{trials}\ (\Gamma > \tau + 10)},$$

where "# trials" means "the number of trials with". Furthermore, we consider the false alarm rate (FAR) as equivalent to the reciprocal of the average false alarm period and use

$$\mathrm{FAR} = \frac{1}{\mathbb{E}_\infty[\Gamma]}.$$

Then, as the ROC curve, we plot TPR versus FAR. In the following, we first briefly explain the benchmark tests and then present the application setups along with the corresponding performance curves.

5.1 Benchmark Algorithms

5.1.1 Nonparametric CUSUM Test

In cases where the univariate test statistic is expected to take higher values in the post-change case compared to the pre-change case, a nonparametric CUSUM test can be used for change detection, where the difference between the test statistic and its mean value in the pre-change case is accumulated over time and a change is declared if the accumulated statistic exceeds a predetermined threshold. For instance, the chi-squared statistic [36] and the magnitude of the innovation sequence in the Kalman filter [40] are expected to increase in case of an anomaly and several variants of the nonparametric CUSUM test have been proposed in the context of anomaly/attack detection in the smart grid [36], [40].

In our case, both summary statistics, i.e., d_t and $\|\mathbf{r}_t\|_2$, are expected to increase in case of an anomaly compared to their nominal mean values. Hence, after obtaining a set of nominal summary statistics, we can compute the empirical mean of them and then apply the nonparametric CUSUM test for real-time anomaly detection. Let us specifically consider d_t and let

$$\bar{d} \triangleq \frac{1}{N_2} \sum_{\mathbf{x}_j \in \mathcal{S}_2} d_j,$$

be the nominal empirical mean of d_t. The nonparametric CUSUM test is given by

$$\begin{aligned} \Gamma &= \inf\{t : g_t \ge h\}, \\ g_t &= \max\{0, g_{t-1} + d_t - \bar{d}\}, \end{aligned} \tag{22}$$

where $g_0 = 0$. In each simulation setup presented below, in an offline phase, we first compute the empirical mean of the nominal summary statistics (either for d_t or $\|\mathbf{r}_t\|_2$) and then employ the nonparametric CUSUM test.

5.1.2 Online Discrepancy Test (ODIT)

The ODIT algorithm presented in [29] is a sequential nonparametric anomaly detection method based on the GEM. It consists of offline and online phases where its offline phase is identical to the offline phase of Algorithm 1. In the online phase, instead of using the entire set $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$, only a threshold distance $d_{[K]}$ is used, denoting the largest Kth element among $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$. In particular, for a chosen significance level α, $K = \lceil \alpha N_2 \rceil$ is chosen, denoting the smallest integer greater than αN_2. The online phase is identical to the nonparametric CUSUM test given in (22), after replacing $\bar{d}$ with $d_{[K]}$.

5.1.3 Information Theoretic Multivariate Change Detection (ITMCD) Algorithm

The ITMCD algorithm presented in [25] is a sequential nonparametric change detection algorithm for multivariate data streams. In particular, it is a two sample test based on the KL divergence between the multivariate distributions corresponding to two consecutive (over time) sliding windows of observations. The KL divergence is estimated in a nonparametric way based on the distances of observations to their NNs both within a window and between the windows.

Let $\mathcal{X}_{t,w_1}$ and $\mathcal{X}_{t,w_2}$ denote the most recent consecutive sliding windows of observations at time t with sizes w_1 and w_2, respectively. That is, at time t, we have $\mathcal{X}_{t,w_1} = \{\mathbf{x}_{t-w_1+1}, \ldots, \mathbf{x}_t\}$ and $\mathcal{X}_{t,w_2} = \{\mathbf{x}_{t-w_1-w_2+1}, \ldots, \mathbf{x}_{t-w_1}\}$. Moreover, let

$e_{m,n}(i)$ denote the euclidean distance between $\mathbf{x}_i \in \mathcal{X}_{t,w_m}$ and its kth NN among the set $\mathcal{X}_{t,w_n}$, where $m, n \in \{1, 2\}$. The KL divergence between the multivariate distributions corresponding to $\mathcal{X}_{t,w_m}$ and $\mathcal{X}_{t,w_n}$ is estimated as follows [25], [41]:

$$\mathrm{KL}_{t,m,n} \triangleq \log\left(\frac{w_n}{w_m - 1}\right) + \frac{p}{w_m} \sum_{\mathbf{x}_i \in \mathcal{X}_{t,w_m}} \log\left(\frac{e_{m,n}(i)}{e_{m,m}(i)}\right),$$

where $\mathbf{x}_t \in \mathbb{R}^p$. The ITMCD algorithm is then given by

$$\Gamma = \inf\{t : \mathrm{KL}_{t,1,2} + \mathrm{KL}_{t,2,1} \geq h\}.$$

The algorithm is based on the fact that the discrepancy between the multivariate distributions increases in case of a change/anomaly. Particularly, after an anomaly, since the window $\mathcal{X}_{t,w_1}$ includes recently acquired anomalous observations before $\mathcal{X}_{t,w_2}$, the distribution of the observations in $\mathcal{X}_{t,w_1}$ changes while the observations in $\mathcal{X}_{t,w_2}$ still have the nominal distribution for some time period. Then, the KL divergence between the two windows of observations increases compared to the case where the both windows have the same nominal distribution. Note that the ITMCD algorithm requires, after obtaining each new observation, repeating the search for the kth NN for each data point within both its own window and the other window. This is computationally intensive for an online algorithm. Further, the window-based approach reduces the time resolution and induces an inherent detection latency. Throughout the section, we choose $k = 4$ and the window sizes as $w_1 = 20$ and $w_2 = 100$ for the ITMCD algorithm.

5.1.4 NN-Based Online Change Detection Algorithm

In [27], a nonparametric online two-sample test is presented based on NN graphs. Particularly, for a sliding window of observations, the algorithm partitions the window into two sets and decides whether the two sets of observations have the same distribution by evaluating how many observations have their NNs from the other set. Given the most recent W observations $\mathcal{S}_W \triangleq \{\mathbf{x}_{t-W+1}, \dots, \mathbf{x}_t\}$ with indices $t_W \triangleq \{t-W+1, \dots, t\}$, the stopping time is given by

$$\Gamma = \inf\{t : \max_{t-n_1 \leq m \leq t-n_0} Z_W(m,t) \geq h\}, \tag{23}$$

where

$$Z_W(m,t) \triangleq \frac{-R_W(m,t) + \mathbb{E}[R_W(m,t)]}{\sqrt{\mathrm{Var}[R_W(m,t)]}},$$

$$R_W(m,t) \triangleq \sum_{i \in t_W} \sum_{j \in t_W} (A_{t_W,ij} + A_{t_W,ji}) B_{ij}(m, t_W),$$

$$A_{t_W,ij} \triangleq \mathbb{1}\{\mathbf{x}_j \text{ is one of the first } k \text{ NNs of } \mathbf{x}_i \text{ among } \mathcal{S}_W\},$$
$$B_{ij}(m,t_W) \triangleq \mathbb{1}\big\{\big(P_{t_W}(i) \leq m, P_{t_W}(j) > m\big) \text{ or } \big(P_{t_W}(i) > m, P_{t_W}(j) \leq m\big)\big\},$$

and $P_{t_W}(\cdot)$ denotes the random permutation among the indices t_W. The mean $\mathbb{E}[R_W(m,t)]$ and variance $\mathrm{Var}[R_W(m,t)]$ under random permutation are given in [27, Section 2]. In our simulations, we choose $W = 50$, $k = 10$, $n_0 = 10$, and $n_1 = 40$.

The test in (23) is based on the idea that if the data distribution changes at some time, then each set of observations are likely to find their NNs within their own set rather than the other set, that leads to a larger decision statistic. To employ the test, at each time, the sliding observation window is updated with the incoming data point and a new NN graph is formed for the new window of observations. Partitioning the observation window into two parts is also a part of the decision process. Particularly, for all possible $n_1 - n_0 + 1$ partitions of the observation window, $Z_W(m,t)$ is computed and maximum among them is considered as the decision statistic. The computational complexity at a time due to building a new NN graph and searching the decision statistic among all possible window partitions might be prohibitive in time-sensitive online settings. Moreover, since the decision mechanism is mainly a two-sample test, the method cannot operate as fully sequential and for reliable decisions, the window size should be chosen sufficiently large. That reduces the time resolution and usually leads to larger detection delays. Furthermore, as also argued in [27], the method is mainly effective in the detection of sharp changes/anomalies, as otherwise difference between the two samples would not be significant. This makes the method ineffective against gradual changes/anomalies such as stealthily designed small-magnitude false data injection (FDI) attacks in the smart grid and low-rate distributed denial of service (DDoS) attacks in IoT networks.

5.1.5 QuantTree

The QuantTree algorithm presented in [28] partitions the high-dimensional observation space using a nominal dataset into a finite number of subregions, say K, such that the nominal data fall into the K subregions with pre-specified probabilities $\pi_1, \dots, \pi_K$, where $\sum_{i=1}^K \pi_i = 1$. Then, in the online phase, for a batch of W observations, it counts how many observations fall into the predetermined subregions, say $y_1, \dots, y_K$, where $\sum_{i=1}^K y_i = W$. In the nominal case, expected number of observations in the subregions are $W\pi_1, \dots, W\pi_K$. The Pearson's chi-squared test is then used to determine whether the observed $y_1, \dots, y_K$ are likely in the nominal case. The algorithm is mainly designed for batch processing, but it can be extended for real-time processing via a sliding window of observations, that leads to the sliding-window chi-squared test, as described in [9]. In this case, $y_1, \dots, y_K$ are determined based on the most recent W observations $\mathbf{x}_{t-W+1}, \dots, \mathbf{x}_t$. The corresponding stopping time is then given by

$$\Gamma = \inf\left\{t : \chi_t \triangleq \sum_{i=1}^K \frac{(y_i - W\pi_i)^2}{W\pi_i} \geq h\right\},$$

where the decision statistic χ_t is asymptotically (as $W \to \infty$) a chi-squared random variable with $K - 1$ degrees of freedom. In our simulations, we choose $W = 256$ and $K = 16$ with $\pi_i = 1/16, \forall i \in \{1, \dots, 16\}$.

5.2 Real-Time Cyber-Attack Detection in Smart Grid

We consider the IEEE-57 bus power system that consists of 57 buses and 80 smart meters. Let $\phi_t \in \mathbb{R}^{57}$ denote the voltage angles (phases) of the buses and $\mathbf{x}_t \in \mathbb{R}^{80}$ denote the

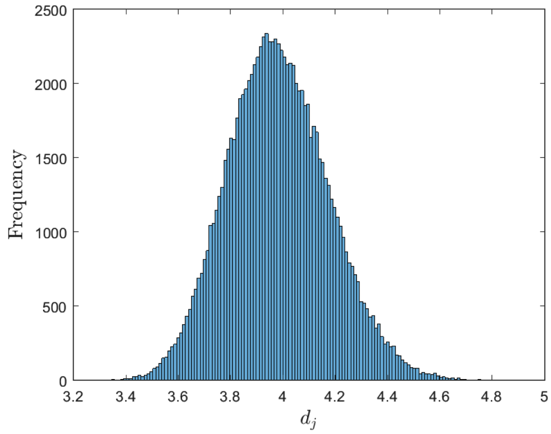

Fig. 6. GEM-based nominal summary statistics for the IEEE-57 bus power system.

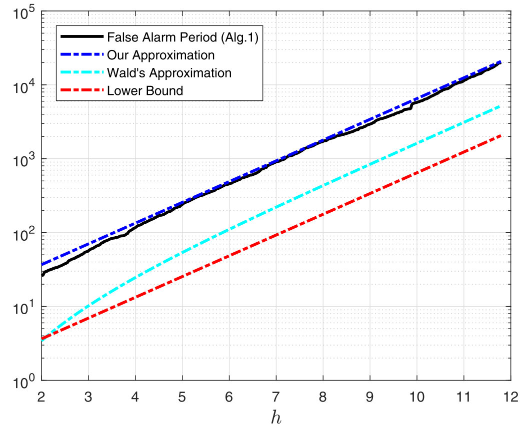

Fig. 7. Average false alarm period, approximations, and the lower bound for various test thresholds for the smart grid data.

measurement vector collected through the smart meters at time t. Suppose that the smart grid operates according to the following linearized DC model [42]:

$$\mathbf{x}_t = \mathbf{H}\,\phi_t + \omega_t, \tag{24}$$

where $\mathbf{H} \in \mathbb{R}^{80\times57}$ is the measurement matrix determined based on the power network topology and $\omega_t \in \mathbb{R}^{80}$ is the measurement noise vector. Moreover, let

$$\omega_t \sim \mathcal{N}(\mathbf{0}_{80}, \sigma^2\mathbf{I}_{80}), \tag{25}$$

where $\mathbf{0}_{80} \in \mathbb{R}^{80}$ consists of all zeros and σ^2 denotes the noise variance for each measurement. We simulate the DC optimal power flow for case-57 using MATPOWER [43] and obtain the nominal voltage angles ϕ_t. Since we consider a steady-state, i.e., static, power system model, we expect that the voltage angles stay nearly the same in the absence of anomalies.

Notice that (24) defines the regular system operation. However, in case of an anomaly, e.g., a cyber-attack, the measurement model in (24) no longer holds. For instance, in case of an FDI attack launched at time τ, the measurement vector takes the following form:

$$\mathbf{x}_t = \mathbf{H}\,\phi_t + \mathbf{a}_t + \omega_t,\ t \geq \tau, \tag{26}$$

where $\mathbf{a}_t \triangleq [a_{t,1}, a_{t,2}, \ldots, a_{t,80}]^{\mathrm{T}}$ is the injected malicious data at time t. We aim to timely detect the FDI attacks targeting the smart grid.

Based on (24) and (25), we have

$$\mathbf{x}_t \sim \mathcal{N}(\mathbf{H}\,\phi_t, \sigma^2\mathbf{I}_{80}), \tag{27}$$

i.e., the nominal data covariance matrix is diagonal and every dimension has equal variance. If we collect a set of nominal data points and perform the PCA, we can observe that every dimension is equally important so that the observed high-dimensional nominal data do not exhibit a low intrinsic dimensionality. Nevertheless, we can still use our proposed GEM-based detector (see Algorithm 1).

In this setup, we generate synthetic data based on the system and attack models presented above. Specifically, during the normal system operation given by (27), we assume $\sigma^2 = 10^{-2}$ and acquire $N = 10^5$ nominal data points, and then uniformly partition them into two parts $\mathcal{S}_1$ and $\mathcal{S}_2$ with sizes $N_1 = 2\times10^3$ and $N_2 = 9.8\times10^4$, respectively. We choose $k = 4$ and for each data point $\mathbf{x}_j \in \mathcal{S}_2$, we compute d_j, the sum of distances of $\mathbf{x}_j$ to its first k NNs among $\mathcal{S}_1$ (see (6)). Then, we obtain the histogram of $\{d_j : \mathbf{x}_j \in \mathcal{S}_2\}$, as given in Fig. 6.

Fig. 7 shows the average false alarm period computed over a Monte Carlo simulation, the asymptotic lower bound presented in Theorem 2, our asymptotic approximation to the average false alarm period, and the Wald's asymptotic approximation (see Remark 1), as the test threshold h varies. We observe that our asymptotic approximation is reasonably close to the actual average false alarm period. Furthermore, Figs. 8 and 9 illustrate the performance of the all tests in detection of an FDI attack against smart grid, where $a_{t,i} \sim \mathcal{U}[-0.14, 0.14], \forall i \in \{1, 2, \ldots, 80\}, \forall t \geq \tau$ and $\mathcal{U}[\rho_1, \rho_2]$ denotes a uniform random variable in the range $[\rho_1, \rho_2]$. The figures show that the proposed algorithm outperforms or at least performs nearly with the benchmark tests. In this example, we also note that the detection delays critically depend on the attack magnitude: smaller detection delays are obtained for larger attack magnitudes. Finally, to illustrate how the proposed algorithm works, we present a sample path of decision statistic g_t over time in Fig. 10, where the FDI attack is launched at $\tau = 200$. We observe that after the attack is launched, the decision statistic steadily increases and exceeds the test threshold h, illustrated with

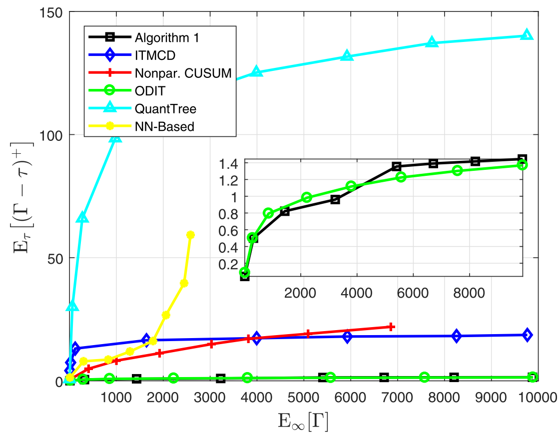

Fig. 8. Average detection delay versus average false alarm period in detection of an FDI attack against the smart grid.

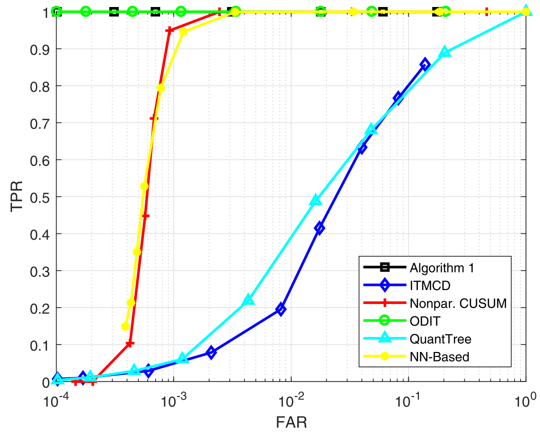

Fig. 9. ROC curve in detection of an FDI attack against the smart grid.

the red dashed line, while staying near zero under regular operating conditions.

5.3 Real-Time Detection of Changes in Human Physical Activity

The Human Activities and Postural Transitions (HAPT) dataset [44] obtained from the UCI Machine Learning Repository [45] contain data for six physical activities: sitting, standing, laying, walking, walking upstairs, and walking downstairs. The first three, i.e., sitting, standing, and laying, are static and the remaining three are dynamic activities. We divide the given dataset into two parts based on the given activity labels such that the first part of the dataset contains data for static activities and the second part contains data for dynamic activities. Our goal is to timely and reliably detect changes from a static to a dynamic activity where each data point is 561-dimensional. We hence consider the static activities as the pre-change (nominal) state and the dynamic activities as the post-change (anomalous) state. Although there are finite number of data points in the given dataset, we assume that at each time we sequentially observe a new data point. Particularly, up to the change-point τ, at each time, we observe a data point chosen uniformly among the set of data points corresponding to static activities and after the change-point, at each time, we observe a data point chosen uniformly from the set of dynamic activities.

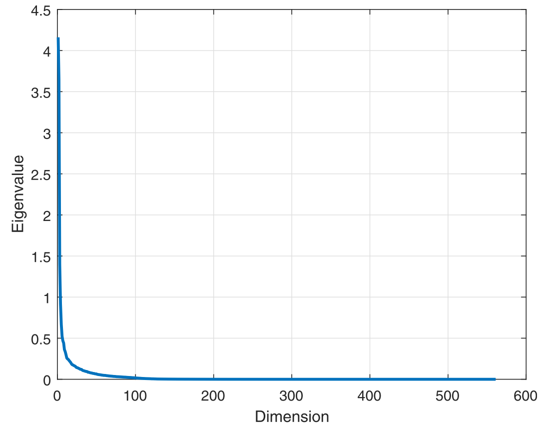

Fig. 11. Eigenvalues of the sample data covariance matrix for a representative set of static activities in the HAPT dataset.

We first uniformly select 2500 data points from the set of data points corresponding to static activities and using the PCA method (see Algorithm 2), we obtain the eigenvalues of the corresponding sample data covariance matrix, as shown in descending order in Fig. 11. We observe through Fig. 11 that the nominal data exhibit a low intrinsic dimensionality. We then choose the minimum desired γ as 0.99. Accordingly, we choose $r = 115$ and retain approximately $\gamma = 0.9903$ fraction of the data variance in the 115-dimensional principal subspace. Then, for the entire set of static activities ($\mathcal{S}_2 = \mathcal{X}$), we compute the PCA-based nominal summary statistics that form the histogram shown in Fig. 12.

In cases where the observed data stream exhibits a low intrinsic dimensionality, another approach is applying the proposed GEM-based detection scheme (Algorithm 1) after dimensionality reduction. That is, after obtaining the matrix $\mathbf{V}$ as described in Algorithm 2, each data point in the nominal training set, $\mathbf{x}_i \in \mathcal{X}$, and also each sequentially available data point, $\mathbf{x}_t$, can be projected onto a r-dimensional space as $\mathbf{V}^{\mathrm{T}}\mathbf{x}_i$ and $\mathbf{V}^{\mathrm{T}}\mathbf{x}_t$, respectively. Algorithm 1 can then be employed over the low-dimensional space, which is computationally more efficient compared to employing the algorithm over the original data space. We employ Algorithm 1 over the projected data, where we obtain the projection matrix $\mathbf{V}$ as described above and uniformly choose $\mathcal{S}_1$ and $\mathcal{S}_2$ (in Algorithm 1) with sizes $N_1 = 1000$ and $N_2 = 4738$,

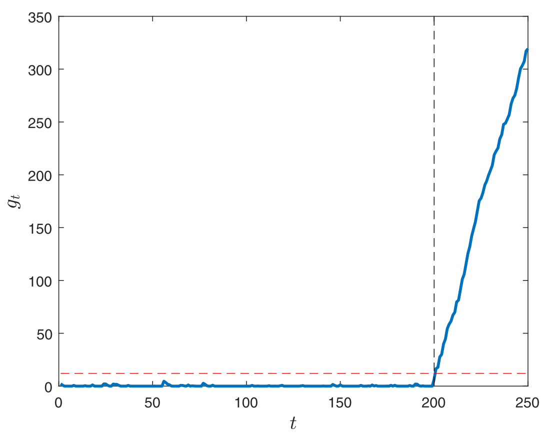

Fig. 10. Sample path of the decision statistic where the FDI attack is launched at $\tau = 200$.

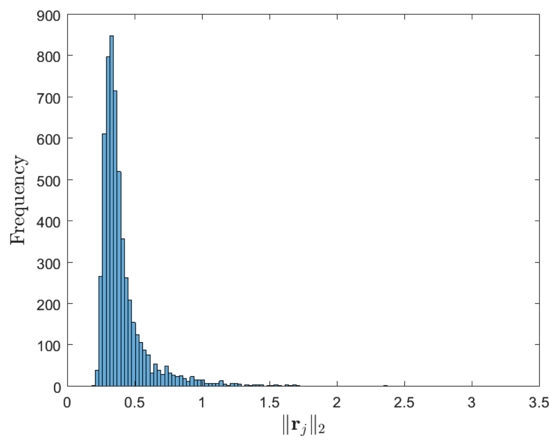

Fig. 12. PCA-based nominal summary statistics for static activities in the HAPT dataset.

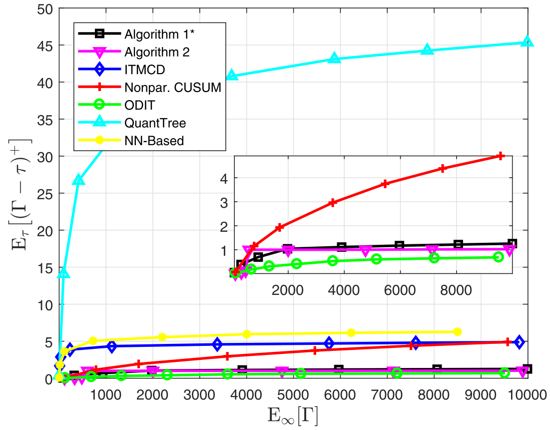

Fig. 13. Average detection delay versus average false alarm period for detecting changes in human physical activities.

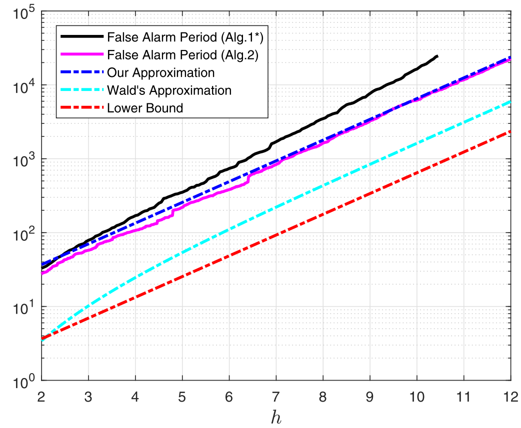

Fig. 15. Average false alarm period, approximations, and the lower bound for various test thresholds for the human activity data.

respectively. Figs. 13 and 14 show that the proposed algorithms perform superior or at least comparable to the benchmark algorithms. Furthermore, Fig. 15 illustrates the average false alarm period, the lower bound, and the approximations for the proposed algorithms. In all the relevant figures, we use an asterisk for Algorithm 1 to emphasize that it is employed based on the projected low-dimensional data.

5.4 Real-Time Detection of IoT Botnet Attacks

Data for network-based detection of IoT botnet attacks (N-BaIoT) [46] obtained from the UCI Machine Learning Repository [45] contain network traffic statistics for an IoT network under both normal and attack conditions, where the IoT network consists of nine devices, namely a thermostat, a baby monitor, a webcam, two doorbells, and four security cameras and the IoT devices are connected via Wi-Fi to several access points. In case of botnet attacks, attackers search for vulnerable devices in the network and inject malwares to the vulnerable devices. Then, they take control of the compromised devices and use them as a part of a bot network (botnet) to perform large-scale attacks such as DDoS attacks over the entire network [46], [47], [48]. In the N-BaIoT dataset, statistical features such as time intervals between packet arrivals, packet sizes and counts are extracted from the real network traffic for each IoT device such that each data point is 115-dimensional. For each device, the data are obtained under both normal operating conditions and several different attacks performed by BASHLITE and Mirai botnets.

Timely and accurate detection of IoT botnet attacks has a critical importance to prevent further malware propagation over the network, e.g., by disconnecting the compromised devices immediately after the detection. As an illustrative attack case, we consider that a spam attack is performed over the network by the BASHLITE botnet [46] and we monitor the thermostat for anomaly detection. First, based on the PCA method summarized in Algorithm 2, 6500 data points chosen uniformly among the nominal dataset are used to compute the sample data covariance matrix, where the corresponding eigenvalues are presented in Fig. 16. We observe that the nominal data can be represented in a lower-dimensional linear subspace and choosing $r = 5$, we retain nearly all the data variance in the 5-dimensional principal subspace, i.e., $\gamma \approx 1$. Then, using the entire nominal dataset, we compute the magnitudes of the residual terms, constituting the nominal summary statistics, a histogram of which is presented in Fig. 17 where the frequencies are shown in the log-scale to have a better illustration.

We assume that before the attack launch time τ, at each time, we observe a nominal data point chosen uniformly among the set of nominal data points and after the attack, at

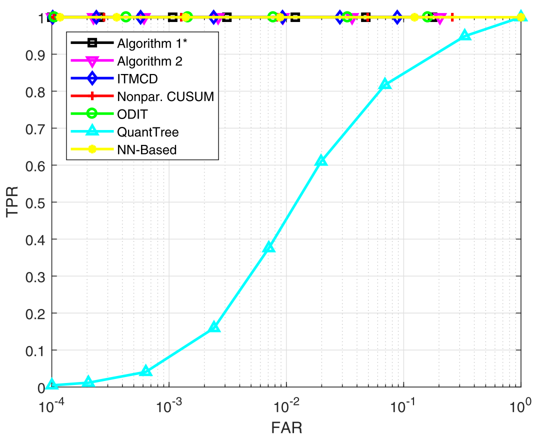

Fig. 14. ROC curve in detection of human activity change.

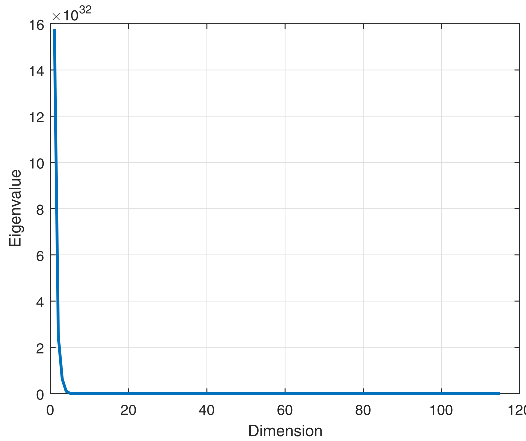

Fig. 16. Eigenvalues of the sample data covariance matrix for a representative set of nominal data points (thermostat) in the N-BaIoT dataset.

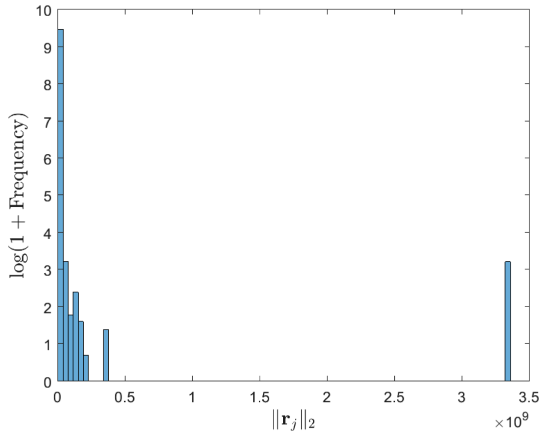

Fig. 17. PCA-based nominal summary statistics for the nominal data (thermostat) in the N-BaIoT dataset.

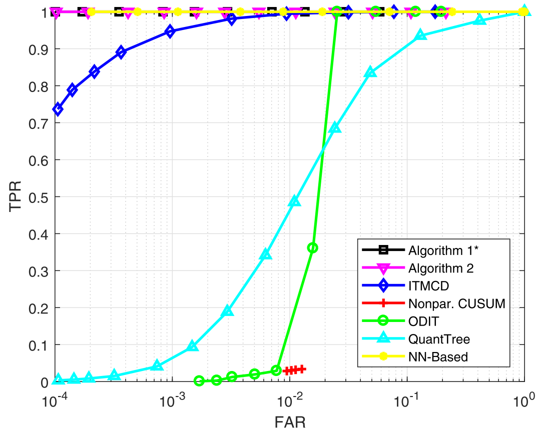

Fig. 19. ROC curve in detection of a spam attack launched by a BASHLITE botnet.

each time, we observe a data point chosen uniformly from the "junk" dataset given for the thermostat [46]. The corresponding performance curves are presented in Figs. 18 and 19. Similarly to the previous application case, we employ Algorithm 1 using the projected r-dimensional data where we uniformly choose S_1 and S_2 in Algorithm 1 with sizes $N_1 = 1215$ and $N_2 = 11896$, respectively.

In this example, we observe that the nonparametric CUSUM test and the ODIT perform considerably worse compared to the other detectors. This is due to some significant outliers in the nominal dataset. Particularly, we observe through Fig. 17 that the baseline summary statistics mostly lie on an interval of smaller values, i.e., the majority of the nominal data points fit well to the principal subspace. On the other hand, we also observe that for some nominal data points, the summary statistics take significantly high values, that dramatically increase the empirical mean of the nominal summary statistics. This, in turn, leads to higher detection delays for the nonparametric CUSUM test. Moreover, the significant outliers among the nominal data points (with very large $\|\mathbf{r}_t\|_2$) also increase the false alarm rate of the nonparametric CUSUM test. Similarly, the significant nominal outliers with large NN distances lead to frequent false alarms in the ODIT and the advantage of Algorithm 1 over the ODIT appears: Algorithm 1 depends on the likelihood of the NN distances (specifically, the p-value) rather than the distances itself, that makes it more reliable in case of significant (distant) nominal outliers. Finally, Fig. 20 shows the average false alarm period, the theoretical lower bound, and the approximations for the proposed algorithms as their test thresholds vary.

6 Conclusion

In this paper, we have proposed nonparametric data-driven real-time anomaly detection schemes. The proposed schemes are reliable, effective, scalable, and hence ideally suited for high-dimensional settings. Moreover, they are widely applicable in a variety of applications as we do not make unrealistic data model assumptions. We have considered both the special case where the observed data stream has a low intrinsic dimensionality and the general case. In both cases, we have proposed to extract and process univariate summary statistics from the observed high-dimensional data streams, where the summary statistics are useful to distinguish anomalous data from nominal data. We have proposed a low-complexity CUSUM-like anomaly detection algorithm based on the extracted summary statistics. We have provided a sufficient condition to asymptotically ensure that the decision statistic of the proposed algorithm does not grow unbounded in the absence of anomalies. We have also provided a controllable

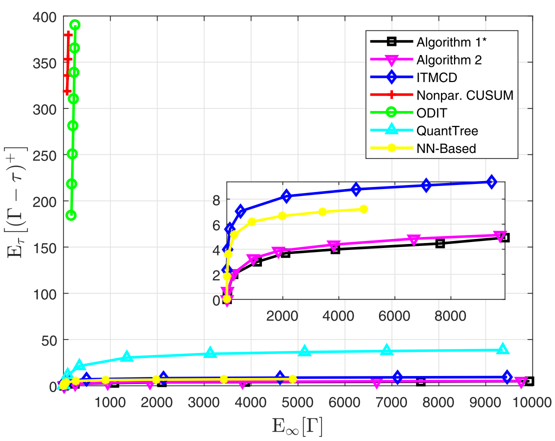

Fig. 18. Average detection delay versus average false alarm period in detection of a spam attack launched by a BASHLITE botnet.

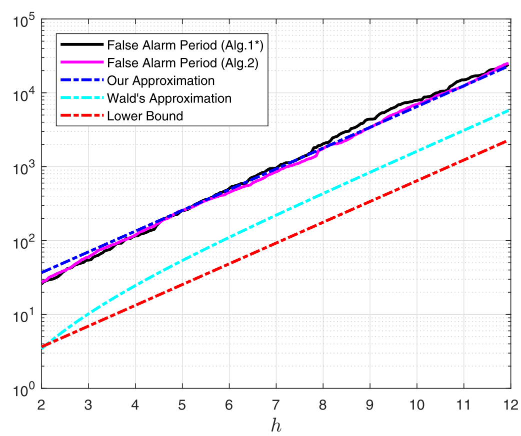

Fig. 20. Average false alarm period, approximations, and the lower bound for various test thresholds for the IoT data.

asymptotic lower bound and an asymptotic approximation for the average false alarm period of the proposed algorithm. Experiments with synthetic and real-world data demonstrate the effectiveness of the proposed schemes in timely and accurate detection of anomalies in a variety of high-dimensional settings.

This work has studied stationary high-dimensional data streams. However, in practice, the observed data stream might be nonstationary. In such cases, a common approach is assuming a slowly time-varying submanifold underlying the observed data stream [13], [14], [49]. We can extend our results to this case where we can employ a subspace tracking algorithm [50] to dynamically estimate the underlying submanifold and using a (sequentially acquired) nominal dataset, for each nominal data point, we can compute the distance between the data point and its representation in the estimated submanifold, that form a set of nominal summary statistics. Then, in the online anomaly detection phase, the proposed CUSUM-like algorithm can be employed based on the extracted summary statistics to evaluate whether the sequentially available data stream rapidly deviates from the nominal dynamic submanifold. Notice that for this approach to be effective, it is still required that the nominal summary statistic has a stationary distribution over time.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under Grants ECCS-1405327 and CNS-1737598, and in part by the U.S. Office of Naval Research under Grant N000141712827.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, 2009, Art. no. 15.
[2] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, 2014.
[3] J. Kittler *et al.*, "Domain anomaly detection in machine perception: A system architecture and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 845–859, May 2014.
[4] H. V. Poor and O. Hadjiliadis, *Quickest Detection*. Cambridge, U.K. : Cambridge Univ. Press, 2008.
[5] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[6] M. N. Kurt and X. Wang, "Multi-sensor sequential change detection with unknown change propagation pattern," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 3, pp. 1498–1518, Jun. 2019.
[7] T. L. Lai, "Information bounds and quick detection of parameter changes in stochastic systems," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2917–2929, Nov. 1998.
[8] Y. Cao, L. Xie, Y. Xie, and H. Xu, "Sequential change-point detection via online convex optimization," *Entropy*, vol. 20, no. 2, 2018, Art. no. 108.
[9] M. N. Kurt, Y. Yilmaz, and X. Wang, "Real-time detection of hybrid and stealthy cyber-attacks in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 498–513, Feb 2019.
[10] A. O. Hero, III, "Geometric entropy minimization (GEM) for anomaly detection and localization," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 585–592.
[11] K. Srichanran and A. O. Hero, "Efficient anomaly detection using bipartite K-NN graphs," in *Proc. Advances Neural Inf. Process. Syst.*, 2011, pp. 478–486.
[12] S. V. Georgakopoulos, S. K. Tasoulis, and V. P. Plagianakos, "Efficient change detection for high dimensional data streams," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2015, pp. 2219–2222.
[13] X. J. Hunt and R. Willett, "Online data thinning via multi-subspace tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1173–1187, May 2019.
[14] Y. Xie, J. Huang, and R. Willett, "Change-point detection for high-dimensional time series with missing data," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 1, pp. 12–27, Feb. 2013.
[15] R. Laxhammar and G. Falkman, "Online learning and sequential anomaly detection in trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1158–1173, Jun. 2014.
[16] R. Dunia and S. J. Qin, "Multi-dimensional fault diagnosis using a subspace approach," in *Proc. Amer. Control Conf.*, 1997.
[17] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, 2004, pp. 219–230.
[18] A. L. Toledo and X. Wang, "Robust detection of selfish misbehavior in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1124–1134, Aug. 2007.
[19] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[20] G. Ratsch, S. Mika, B. Scholkopf, and K. . Muller, "Constructing boosting algorithms from SVMs: An application to one-class classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1184–1199, Sep. 2002.
[21] M. Wu and J. Ye, "A small sphere and large margin approach for novelty detection using training data with outliers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2088–2092, Nov. 2009.
[22] V. Jumutc and J. A. K. Suykens, "Multi-class supervised novelty detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2510–2523, Dec. 2014.
[23] M. Zhao and V. Saligrama, "Anomaly detection with score functions based on nearest neighbor graphs," in *Proc. Advances Neural Inf. Process. Syst.*, 2009, pp. 2250–2258.
[24] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Advances Neural Inf. Process. Syst.*, 2007, pp. 513–520.
[25] L. Faivishevsky, "Information theoretic multivariate change detection for multisensory information processing in internet of things," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Mar. 2016, pp. 6250–6254.
[26] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi, "An information-theoretic approach to detecting changes in multi-dimensional data streams," in *Proc. Symp. Interface Statist. Comput. Sci. Appl.*, 2006.
[27] H. Chen, "Sequential change-point detection based on nearest neighbors," *Ann. Statist.*, vol. 47, no. 3, pp. 1381–1407, 2019.
[28] G. Boracchi, D. Carrera, C. Cervellera, and D. Macciò, "QuantTree: Histograms for change detection in multivariate data streams," in *Proc. 35th Int. Conf. Mach. Learn.*, D. Jennifer and K. Andreas, Eds., 2018, vol. 80, pp. 639–648.
[29] Y. Yilmaz, "Online nonparametric anomaly detection based on geometric entropy minimization," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2017, pp. 3010–3014.
[30] G. Lorden, "Procedures for reacting to a change in distribution," *Ann. Math. Statist.*, vol. 42, no. 6, pp. 1897–1908, 1971.
[31] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," *Ann. Statist.*, vol. 14, no. 4, pp. 1379–1387, 1986.
[32] M. N. Kurt, Y. Yilmaz, and X. Wang, "Distributed quickest detection of cyber-attacks in smart grid," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 8, pp. 2015–2030, Aug. 2018.
[33] A. Ghoting, S. Parthasarathy, and M. E. Otey, "Fast mining of distance-based outliers in high-dimensional datasets," *Data Mining Knowl. Discovery*, vol. 16, no. 3, pp. 349–364, 2008.
[34] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer, 2006.
[35] A. W. Van der Vaart, *Asymptotic Statistics*, vol. 3, Cambridge, U.K.: Cambridge Univ. Press, 1998.
[36] C. Murguia and J. Ruths, "Cusum and chi-squared attack detection of compromised sensors," in *Proc. IEEE Conf. Control Appl.*, Sep. 2016, pp. 474–480.
[37] R. A. Khan, "Wald's approximations to the average run length in cusum procedures," *J. Statist. Planning Inference*, vol. 2, no. 1, pp. 63–77, 1978.
[38] C. W. Champ and S. E. Rigdon, "A a comparison of the markov chain and the integral equation approaches for evaluating the run length distribution of quality control charts," *Commun. Statist.-Simul. Comput.*, vol. 20, no. 1, pp. 191–204, 1991.

[39] M. R. Reynolds, "Approximations to the average run length in cumulative sum control charts," *Technometrics*, vol. 17, no. 1, pp. 65–71, 1975.

[40] Q. Yang, L. Chang, and W. Yu, "On false data injection attacks against kalman filtering in power system dynamic state estimation," *Secur. Commun. Netw.*, vol. 9, no. 9, pp. 833–849, 2016.

[41] Q. Wang, S. R. Kulkarni, and S. Verdu, "A nearest-neighbor approach to estimating divergence between continuous random vectors," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2006, pp. 242–246.

[42] A. Abur and A. Gomez-Exposito, *Power System State Estimation: Theory and Implementation*. New York, NY, USA: Marcel Dekker, 2004.

[43] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[44] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[45] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[46] Y. Meidan *et al.*, "N-baiot: Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul. 2018.

[47] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[48] E. Bertino and N. Islam, "Botnets and Internet of Things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[49] R. Zimmermann, B. Peherstorfer, and K. Willcox, "Geometric subspace updates with applications to online adaptive nonlinear model reduction," *SIAM J. Matrix Anal. Appl.*, vol. 39, no. 1, pp. 234–261, 2018.

[50] D. J. Pierre, *Subspace Tracking for Signal Processing*. Hoboken, NJ, USA: Wiley-Blackwell, 2010, ch. 4, pp. 211–270.

Mehmet Necip Kurt (Student Member, IEEE) received the BS and MS degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2014 and 2016, respectively. He is currently working toward the PhD degree in electrical engineering with Columbia University, New York, NY. His current research interests include sequential analysis, statistical signal processing, and machine learning with applications to cybersecurity and cyber-physical systems.

Yasin Yılmaz (Member, IEEE) received the PhD degree in electrical engineering from Columbia University, New York, NY, in 2014. He is currently an assistant professor of electrical engineering with the University of South Florida, Tampa, FL. His research interests include statistical signal processing, machine learning, and their applications to cybersecurity, cyber-physical systems, IoT networks, social networks, and communication systems. He received the Collaborative Research Award from Columbia University in 2015 and the Research Initiation Award from the Southeastern Center for Electrical Engineering Education in 2017.

Xiaodong Wang (Fellow, IEEE) received the PhD degree in electrical engineering from Princeton University. He is currently a professor of electrical engineering with Columbia University, New York, NY. His research interests include the general areas of computing, signal processing and communications, and has published extensively in these areas. Among his publications is a book entitled "*Wireless Communication Systems: Advanced Techniques for Signal Reception*", published by Prentice Hall in 2003. His current research interests include wireless communications, statistical signal processing, and genomic signal processing. Dr. Wang received the 1999 NSF CAREER Award, the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award, and the 2011 IEEE Communication Society Award for Outstanding Paper on New Communication Topics. He has served as an Associate Editor for the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Signal Processing*, and the *IEEE Transactions on Information Theory*. He is listed as an ISI highly-cited author.