Data-Driven Quickest Change Detection

Mehmet Necip Kurt

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

Columbia University

2020

ABSTRACT

Data-Driven Quickest Change Detection

Mehmet Necip Kurt

The quickest change detection (QCD) problem is to detect abrupt changes in a sensing environment as quickly as possible in real time while limiting the risk of false alarm. Statistical inference about the monitored stochastic process is performed through observations acquired sequentially over time. After each observation, QCD algorithm either stops and declares a change or continues to have a further observation in the next time interval. There is an inherent tradeoff between speed and accuracy in the decision making process. The design goal is to optimally balance the average detection delay and the false alarm rate to have a timely and accurate response to abrupt changes.

The objective of this thesis is to investigate effective and scalable QCD approaches for real-world data streams. The classical QCD framework is model-based, that is, statistical data model is assumed to be known for both the pre- and post-change cases. However, real-world data often exhibit significant challenges for data modeling such as high dimensionality, complex multivariate nature, lack of parametric models, unknown post-change (e.g., attack or anomaly) patterns, and complex temporal correlation. Further, in some cases, data is privacy-sensitive and distributed over a system, and it is not fully available to QCD algorithm. This thesis addresses these challenges and proposes novel data-driven QCD approaches that are robust to data model mismatch and hence widely applicable to a variety of practical settings.

In Chapter 2, online cyber-attack detection in the smart power grid is formulated as a partially observable Markov decision process (POMDP) problem based on the

QCD framework. A universal robust online cyber-attack detection algorithm is proposed using the model-free reinforcement learning (RL) for POMDPs. In Chapter 3, online anomaly detection for big data streams is studied where the nominal (i.e., pre-change) and anomalous (i.e., post-change) high-dimensional statistical data models are unknown. A data-driven solution approach is proposed, where firstly a set of useful univariate summary statistics is computed from a nominal dataset in an offline phase and next, online summary statistics are evaluated for a persistent deviation from the nominal statistics.

In Chapter 4, a generic data-driven QCD procedure is proposed, called DeepQCD, that learns the change detection rule directly from the observed raw data via deep recurrent neural networks. With sufficient amount of training data including both pre- and post-change samples, DeepQCD can effectively learn the change detection rule for all complex, high-dimensional, and temporally correlated data streams. Finally, in Chapter 5, online privacy-preserving anomaly detection is studied in a setting where the data is distributed over a network and locally sensitive to each node, and its statistical model is unknown. A data-driven differentially private distributed detection scheme is proposed, which infers network-wide anomalies based on the perturbed and encrypted statistics received from nodes. Furthermore, analytical privacy-security tradeoff in the network-wide anomaly detection problem is investigated.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my Ph.D. advisor Prof. Xiaodong Wang for his guidance throughout my doctoral studies. It was a great pleasure and an invaluable experience to work with him. His wisdom and determination in research have been always inspiring for me.

I would like to thank Prof. Yasin Yılmaz for many fruitful discussions and collaborations during my doctoral studies.

I am thankful for the thesis committee, Prof. John Paisley, Prof. Jingchen Liu, Prof. Yao Xie, and Prof. James Anderson, for their time and efforts in reading and evaluating my Ph.D. thesis.

I also thank my friends Oyetunji Ogundijo, Şan Gültekin, Morteza Ashraphijuo, Le Zheng, and many others at Columbia for their kind support and friendship.

I would like to express my deepest gratitude to my dear family. This thesis would not have been possible without their lovely support and encouragement. I owe special thanks to my parents Hatice Kurt and Mahmut Kurt, and my sisters Mine Kurt, Ayşe Erdem, and Merve Baran. My love for them is beyond words.

To my beloved parents

Sevgili Anneme ve Babama

Chapter 1

Introduction

Suppose that a stochastic process produces a data stream $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots$, where $\boldsymbol{x}_t \in \mathbb{R}^p$ is the observation made at time t and $p \geq 1$ is the data dimensionality. An abrupt change happens in the statistical properties of the observed process at an unknown time τ, called the change-point, such that

$$\boldsymbol{x}_t \sim \begin{cases} f_0, & \text{if } t < \tau, \\ f_1, & \text{if } t \geq \tau, \end{cases} \tag{1.1}$$

where f_0 and f_1 denote the pre- and post-change probability density functions (pdfs) of $\boldsymbol{x}_t$, respectively. In the quickest change detection (QCD) framework, the objective is to develop a sequential procedure for detecting changes as quickly as possible while limiting the risk of false alarm.

In the QCD algorithms, at each time step, a decision is made based on the available information on whether to stop and declare a change or continue to acquire a further observation in the next time interval. The stopping time Γ, at which a change is declared, is a random variable depending on the observations seen so far. That is, $\{\Gamma = t\} \in \mathcal{F}_t, \forall t \geq 1$, where $\mathcal{F}_t = \sigma(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t)$ denotes the sigma-algebra generated by the observations up to time t. The event $\{\Gamma < \tau\}$ corresponds to a false alarm event and $(\Gamma - \tau)^+$ is called the detection delay, where $(\cdot)^+ \triangleq \max\{0, \cdot\}$. While the goal is to minimize both the detection delays and the frequency of false alarm events, there is an inherent tradeoff between these two objectives. Hence, ideally the stopping time is chosen to optimally balance the detection speed and the detection accuracy.

The QCD problem finds diverse applications in many fields such as critical infrastructure monitoring, industrial process control, environmental monitoring, cybersecurity, finance, video surveillance, navigation system monitoring, biomedical signal processing, neuroscience, cognitive radio, healthcare, onset of a disease outbreak, social networks, and military [5, 65, 67, 68, 108, 133]. This thesis investigates effective and scalable QCD approaches applicable to real-world settings.

The classical QCD framework is model-based, that is, the pre- and post-change pdfs f_0 and f_1 are both assumed to be known, possibly with some unknown parameters [5, 108]. However, the real-world data often exhibits significant challenges for data modeling such as high dimensionality, complex multivariate nature, lack of parametric models, unknown post-change (e.g., attack or anomaly) patterns, and complex temporal correlation. For instance, in large-scale heterogeneous networks, such as Internet of Things (IoT) networks, due to complex interactions between disparate IoT devices, it is difficult to model and resource-demanding (computation, energy, and time) to accurately estimate the joint probability distribution of the observed high-dimensional data. Moreover, the real-world data often do not well fit into the existing parametric models [68, 74]. Furthermore, in online attack or anomaly detection, the post-change pdf is usually unknown due to unknown capabilities and strategies of attackers and a myriad of vulnerabilities of modern complex systems [67]. Additionally, complex temporal correlation between observations in real-world data streams (e.g., surveillance videos) render accurate data modeling quite challenging. Further, in some cases, the data is privacy-sensitive and distributed over a network, and it is not fully available to QCD algorithm. This thesis addresses several of these challenges over different problems and proposes novel data-driven QCD approaches. See Fig. 1.1

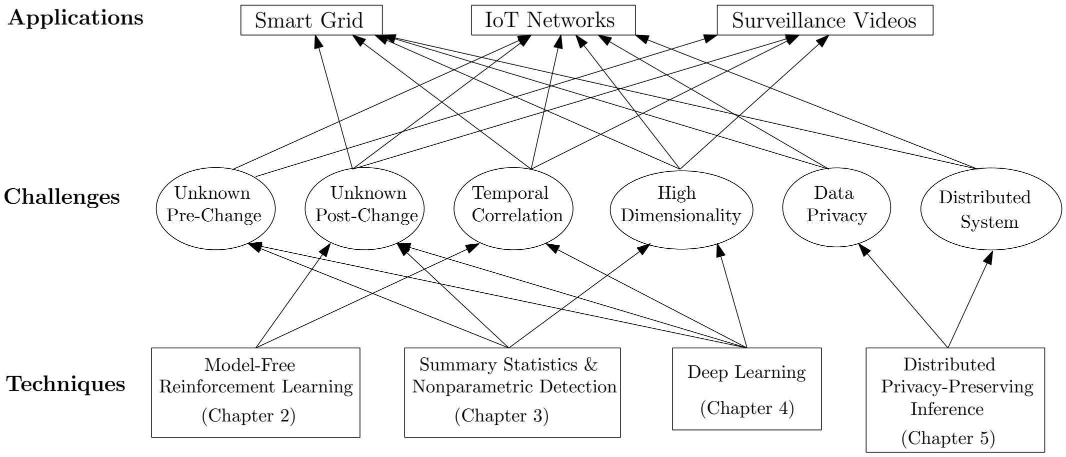


Figure 1.1: An overview of the applications, challenges, and solution techniques.

for an overview of the applications, challenges, and solution techniques.

Chapter 2 addresses early detection of cyber-attacks for a safe and reliable operation of the smart power grid. In particular, the online attack detection is formulated as a partially observable Markov decision process (POMDP) problem. Then, a universal robust online detection algorithm is proposed as a solution using the framework of model-free reinforcement learning (RL) for POMDPs. The proposed detector does not require attack models and hence it can detect new and unknown attack types. Moreover, it is sensitive to detect slight deviations of sensor measurements from the normal system operation, which significantly limits the action space of an attacker. Numerical studies illustrate the effectiveness of the proposed RL-based detector in timely and accurate detection of cyber-attacks targeting the smart grid.

Chapter 3 investigates real-time detection of abrupt anomalies in modern systems producing big (i.e., high-dimensional) data streams. With this goal, effective and scalable algorithms are proposed, where the proposed algorithms are model-free as both the nominal and anomalous multivariate data distributions are assumed to be unknown. Useful univariate summary statistics are extracted from the observed high-dimensional data and the online anomaly detection is performed in a single-dimensional space. Anomalies are defined as being equivalent to persistent outliers and detected via a nonparametric cumulative sum (CUSUM)-like algorithm. In case the observed data has a low intrinsic dimensionality, a submanifold, in which the nominal data is embedded, is learned in an offline phase and next, the sequentially acquired data is evaluated for a persistent deviation from the nominal submanifold. Further, in the general case, an acceptance region is learned for nominal data via the Geometric Entropy Minimization [48] in an offline phase, and next, it is evaluated if the sequentially observed data persistently fall outside the nominal acceptance region. An asymptotic lower bound and an asymptotic approximation for the average false alarm period of the proposed algorithm are provided. Moreover, a sufficient condition is provided to asymptotically guarantee that the decision statistic of the proposed algorithm does not diverge in the absence of anomalies. Experiments illustrate the effectiveness of proposed algorithms in quick and accurate anomaly detection in high-dimensional settings.

Chapter 4 aims to unify the quickest change detection (QCD) framework via a novel data-driven approach. To this end, a generic neural network architecture is proposed for the QCD task, composed of initial data processing, recurrent, and regression layers. The neural network learns the change detection rule directly from data without needing the knowledge of statistical data models and feature engineering. Specifically, the initial data processing layers filter out noise and extract useful features from data. The recurrent layers keep an internal state summarizing the data stream seen so far and update the state as new information comes in. Finally, the regression layers map the internal state into the decision statistic, defined as the posterior probability of having change. Comparisons with the existing model-based QCD algorithms demonstrate the power of the proposed data-driven approach, called DeepQCD, under several scenarios including transient changes and temporally correlated data streams. Further, experiments with high-dimensional real-world data illustrate the superior performance of DeepQCD compared to the existing alternatives in online anomaly detection over surveillance videos and online attack detection over IoT networks.

Chapter 5 studies online privacy-preserving anomaly detection in a setting where the data is distributed over a network and locally sensitive to each node, and its statistical model is unknown. A novel data-driven differentially private solution scheme is designed and analyzed in which each node observes a possibly high-dimensional data stream for which it computes a local outlierness score at each time, perturbs and encrypts it, and then sends it to a network-wide decision maker. The decision maker decrypts the received messages, obtains an aggregate statistic over the network, and then performs online anomaly detection via a generalized CUSUM algorithm. An asymptotic lower bound and an asymptotic approximation are provided for the average false alarm period of the proposed algorithm. Additionally, an asymptotic upper bound and asymptotic approximation are provided for the average detection delay of the proposed algorithm under certain given anomaly signal. The analytical tradeoff between the anomaly detection performance and the differential privacy level is shown, which is controlled via the amount of local perturbation noise. Experiments with real-world data illustrate that the proposed solution scheme offers a good tradeoff between quick anomaly detection and data privacy.

Chapter 2

Online Cyber-Attack Detection in Smart Grid: A Reinforcement Learning Approach

2.1 Introduction

2.1.1 Background and Related Work

With the recent advancements in monitoring, sensing, signal processing, control, and communication, advanced technologies are being integrated into the next-generation power systems (i.e., smart grid). Due to such features, the smart grid depends on a critical cyber infrastructure which makes it vulnerable to hostile cyber threats [79,135,141]. Main objective of attackers is to damage or mislead the state estimation mechanism in the smart grid to cause wide-area power blackouts or to manipulate electricity market prices [138]. There are many types of cyber-attacks, among them false data injection (FDI), jamming, and denial of service (DoS) attacks are well known. FDI attacks add malicious fake data to sensor measurements [9, 66, 77, 82], jamming attacks corrupt sensor measurements via additive noise [67], and DoS attacks block the access of system to sensor measurements [4, 66, 147]. The smart grid is a complex network and any failure or anomaly in a part of the system may lead to huge damages on the overall system in a short period of time. Hence, early detection of cyber-attacks is critical for a timely and effective response. In this context, the QCD framework [5, 107, 108, 133] can be quite useful.

If the probability density functions (pdfs) of sensor measurements for the pre-change (i.e., normal system operation) and the post-change (i.e., after an attack or anomaly) cases can be modeled sufficiently accurately, the well-known cumulative sum (CUSUM) test is the optimal online detector [94] based on the Lorden's minimax criterion [84]. Moreover, if the pdfs can be modeled with some unknown parameters, the generalized CUSUM test, which makes use of the estimates of unknown parameters, has asymptotic optimality properties [5]. However, CUSUM-based detection schemes require perfect data models for both the pre- and post-change cases. In practice, capabilities of an attacker and correspondingly attack types and strategies can be totally unknown. For instance, an attacker can arbitrarily combine and launch multiple attacks simultaneously or it can launch a new unknown type of attack [67]. Then, it may not be always possible to know the attacking strategies ahead of time and to accurately model the post-change case. Hence, universal detectors, not requiring any attack model, are needed in general. Moreover, the (generalized) CUSUM algorithm has optimality properties in minimizing a least favorable (worst-case) detection delay subject to false alarm constraints [5, 94]. Since the worst case detection delay is a pessimistic metric, it is, in general, possible to obtain algorithms performing better than the (generalized) CUSUM algorithm.

Considering the pre-change and the post-change cases as hidden states due to the unknown change-point, a QCD problem can be formulated as a partially observable Markov decision process (POMDP) problem. For the problem of online attack/anomaly detection in the smart grid, in the pre-change state, the system is operated under normal conditions and using the system model, the pre-change measurement pdf can be specified highly accurately. On the other hand, the post-change measurement pdf can take different unknown forms depending on the attacker's strategy. Furthermore, the transition probability between the hidden states is unknown in general. Hence, the exact model of the POMDP is unknown.

Reinforcement learning (RL) algorithms are known to be effective in controlling uncertain environments. Hence, the described POMDP problem can be effectively solved using RL. In particular, as a solution, either the underlying POMDP model can be learned and then a model-based RL algorithm for POMDPs [30, 114] can be used or a model-free RL algorithm [53, 73, 83, 101, 102] can be used without learning the underlying model. Since the model-based approach requires a two-step solution that is computationally more demanding and only an approximate model can be learned in general, we prefer to use the model-free RL approach.

Outlier detection schemes such as the Euclidean detector [87] and the cosine-similarity metric based detector [111] are universal as they do not require any attack model. They mainly compute a dissimilarity metric between actual sensor measurements and predicted measurements (by the Kalman filter) and declare an attack/anomaly if the amount of dissimilarity exceeds a certain predefined threshold. However, such detectors do not consider temporal relation between attacked/anomalous measurements and make sample-by-sample decisions. Hence, they are unable to distinguish instantaneous high-level random system noise from long-term (persistent) anomalies caused, for example, by an unfriendly intervention to the system. Hence, compared to the outlier detection schemes, more reliable universal attack detection schemes are needed.

In this chapter, we consider the smart grid security problem from the defender's perspective and seek for an effective online detection scheme using RL techniques (single-agent RL). Note that the problem can be considered from an attacker's perspective as well, where the objective would be to determine the attacking strategies leading to the maximum possible damage on the system. Such a problem can be particularly useful in vulnerability analysis, that is, to identify the worst possible damage an attacker may introduce to the system and accordingly to take necessary precautions [19, 41, 140]. In the literature, several studies investigate vulnerability analyses using RL, see for example [19] for FDI attacks and [140] for sequential network topology attacks. We further note that the problem can also be considered from both defender's and attacker's perspectives simultaneously, that corresponds to a game-theoretic setting.

Extension of single-agent RL to multiple agents is the multi-agent RL framework involves game theory since in this case, the optimal policies of agents depend both on the environment and the policies of the other agents. Moreover, stochastic games extend the Markov decision processes to multi-agent case where the game is sequential and consists of multiple states, and the transition from one state to another and also the payoffs (rewards or costs) depend on joint actions of all agents. Several RL-based solution approaches have been proposed for stochastic games, see for example [21,51,80,98,136]. Further, if the game (i.e., the underlying state of the environment, actions and payoffs of other agents) is partially observed, then it is called a partially observable stochastic game, for which finding a solution is more difficult in general. In this chapter, we consider the single-agent RL setting from the defender's perspective.

2.1.2 Contributions

We propose an online cyber-attack detection algorithm using the framework of model-free RL for POMDPs [64]. The proposed algorithm is universal, that is, it does not require attack models. This makes the proposed scheme widely applicable and also proactive in the sense that new unknown attack types can be detected. Since we follow a model-free RL approach, the defender learns a direct mapping from observations to actions (*stop* or *continue*) by trial-and-error. In the training phase, although it is possible to obtain or generate measurement data for the pre-change case using the system model under normal operating conditions, it is generally difficult to obtain real attack data. For this reason, we follow a robust detection approach by training the defender with low-magnitude attacks that corresponds to the worst-case scenarios from a defender's perspective since such attacks are quite difficult to detect. Then, the trained defender becomes sensitive to detect slight deviations of sensor measurements from the normal system operation. The robust detection approach significantly limits the action space of an attacker as well. In other words, to prevent the detection, an attacker can only exploit very low attack magnitudes, which are practically not much of interest due to their minimal damage on the system.

2.1.3 Organization

We introduce the system model and the state estimation mechanism in Sec. 2.2. We present the problem formulation in Sec. 2.3 and the proposed solution approach in Sec. 2.4. We then illustrate the performance of the proposed RL-based detection scheme via extensive simulations in Sec. 2.5. Finally, Sec. 2.6 concludes the chapter. Boldface letters denote vectors and matrices, all vectors are column vectors, and $\cdot^{\mathrm{T}}$ denotes the transpose operator. Furthermore, $\mathbb{P}$ and $\mathbb{E}$ denote the probability and expectation operators, respectively.

2.2 System Model and State Estimation

2.2.1 System Model

Suppose that there are K sensors in a power grid consisting of $N+1$ buses, where usually $K > N$ to have the necessary measurement redundancy against noise [1]. One of the buses is considered as a reference bus and the system state at time t is denoted with $\boldsymbol{\phi}_t = [\phi_{1,t}, \dots, \phi_{N,t}]^{\mathrm{T}}$ where $\phi_{n,t}$ denotes the phase angle at bus n at time t and the phase angle of the reference bus is set to zero. Let the measurement taken at sensor k at time t be denoted with $x_{k,t}$ and the measurement vector be denoted with

$\boldsymbol{x}_t = [x_{1,t}, \dots, x_{K,t}]^{\mathrm{T}}$. We model the smart grid as a discrete-time linear dynamic system as follows [1,23,25,120]:

$$\boldsymbol{\phi}_t = \boldsymbol{A}\boldsymbol{\phi}_{t-1} + \boldsymbol{v}_t, \tag{2.1}$$

$$\boldsymbol{x}_t = \boldsymbol{H}\boldsymbol{\phi}_t + \boldsymbol{w}_t, \tag{2.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is the system (state transition) matrix, $\boldsymbol{H} \in \mathbb{R}^{K \times N}$ is the measurement matrix determined based on the network topology, $\boldsymbol{v}_t = [v_{1,t}, \dots, v_{N,t}]^{\mathrm{T}}$ is the process noise vector, and $\boldsymbol{w}_t = [w_{1,t}, \dots, w_{K,t}]^{\mathrm{T}}$ is the measurement noise vector. We assume that $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are independent additive white Gaussian random processes where $\boldsymbol{v}_t \sim \boldsymbol{\mathcal{N}}(\boldsymbol{0}, \sigma_v^2 \boldsymbol{I}_N)$, $\boldsymbol{w}_t \sim \boldsymbol{\mathcal{N}}(\boldsymbol{0}, \sigma_w^2 \boldsymbol{I}_K)$, and $\boldsymbol{I}_K \in \mathbb{R}^{K \times K}$ is an identity matrix. We assume that the system is stable under normal operating conditions. Moreover, we assume that the system is observable, that is, the observability matrix

$$\boldsymbol{O} \triangleq \begin{bmatrix} \boldsymbol{H} \\ \boldsymbol{H}\boldsymbol{A} \\ \vdots \\ \boldsymbol{H}\boldsymbol{A}^{N-1} \end{bmatrix}$$

has rank N.

The system model given in Eq. (2.1) and Eq. (2.2) corresponds to the normal system operation. In case of a cyber-attack, however, the measurement model in Eq. (2.2) is no longer true. For instance, in case of a(n):

1. FDI attack launched at time τ, the measurement model can be written as

$$\boldsymbol{x}_t = \boldsymbol{H}\boldsymbol{\phi}_t + \boldsymbol{w}_t + \boldsymbol{b}_t \mathbb{1}\{t \geq \tau\},$$

where $\mathbb{1}$ is an indicator function and $\boldsymbol{b}_t \triangleq [b_{1,t}, \dots, b_{K,t}]^{\mathrm{T}}$ denotes the injected malicious data at time $t \geq \tau$ and $b_{k,t}$ denotes the injected false datum to the kth sensor at time t,

2. Jamming attack with additive noise, the measurement model can be written as

$$\boldsymbol{x}_t = \boldsymbol{H}\boldsymbol{\phi}_t + \boldsymbol{w}_t + \boldsymbol{u}_t \mathbb{1}\{t \geq \tau\},$$

where $\boldsymbol{u}_t \triangleq [u_{1,t}, \dots, u_{K,t}]^{\mathrm{T}}$ denotes the random noise realization at time $t \geq \tau$ and $u_{k,t}$ denotes the jamming noise corrupting the kth sensor at time t,

3. Hybrid FDI/jamming attack [67], the sensor measurements take the following form:
$$\boldsymbol{x}_t = \boldsymbol{H}\boldsymbol{\phi}_t + \boldsymbol{w}_t + (\boldsymbol{b}_t + \boldsymbol{u}_t)\mathbb{1}\{t \geq \tau\},$$

4. DoS attack, sensor measurements can be partially unavailable to the system controller. The measurement model can then be written as
$$\boldsymbol{x}_t = \boldsymbol{D}_t(\boldsymbol{H}\boldsymbol{\phi}_t + \boldsymbol{w}_t),$$
where $\boldsymbol{D}_t = \text{diag}(d_{1,t}, \ldots, d_{K,t})$ is a diagonal matrix consisting of 0s and 1s. Particularly, if $x_{k,t}$ is available, then $d_{k,t} = 1$, otherwise $d_{k,t} = 0$. Note that $\boldsymbol{D}_t = \boldsymbol{I}_K$ for $t < \tau$,

5. Network topology attack, the measurement matrix changes. Denoting the measurement matrix under topology attack at time $t \geq \tau$ by $\bar{\boldsymbol{H}}_t$, we have
$$\boldsymbol{x}_t = \begin{cases} \boldsymbol{H}\boldsymbol{\phi}_t + \boldsymbol{w}_t, & \text{if } t < \tau \\ \bar{\boldsymbol{H}}_t\,\boldsymbol{\phi}_t + \boldsymbol{w}_t, & \text{if } t \geq \tau, \end{cases}$$

6. Mixed topology and hybrid FDI/jamming attack, the measurement model can be written as follows:
$$\boldsymbol{x}_t = \begin{cases} \boldsymbol{H}\boldsymbol{\phi}_t + \boldsymbol{w}_t, & \text{if } t < \tau \\ \bar{\boldsymbol{H}}_t\,\boldsymbol{\phi}_t + \boldsymbol{w}_t + \boldsymbol{b}_t + \boldsymbol{u}_t, & \text{if } t \geq \tau. \end{cases}$$

2.2.2 State Estimation

Since the smart grid is regulated based on estimated system states, state estimation is a fundamental task in the smart grid, that is conventionally performed using the static least squares (LS) estimators [9,36,82]. However, in practice, the smart grid is a highly dynamic system due to time-varying load and power generation [127]. Furthermore, time-varying cyber-attacks can be designed and performed by the adversaries. Hence, dynamic system modeling as in Eq. (2.1) and Eq. (2.2) and correspondingly using a dynamic state estimator can be quite useful for real-time operation and security of the smart grid [66, 67].

For a discrete-time linear dynamic system, if the noise terms are Gaussian, the Kalman filter is the optimal linear estimator in minimizing the mean squared state estimation error [56]. Note that for the Kalman filter to work correctly, the system needs to be observable. The Kalman filter is an online estimator consisting of prediction and measurement update steps at each iteration. Denoting the state estimates at time t with $\hat{\boldsymbol{\phi}}_{t|t'}$ where $t' = t-1$ and $t' = t$ for the prediction and measurement update steps, respectively, the Kalman filter equations at time t can be written as follows:

Prediction:

$$\begin{aligned} \hat{\boldsymbol{\phi}}_{t|t-1} &= \boldsymbol{A}\hat{\boldsymbol{\phi}}_{t-1|t-1}, \\ \boldsymbol{F}_{t|t-1} &= \boldsymbol{A}\boldsymbol{F}_{t-1|t-1}\boldsymbol{A}^{\mathrm{T}} + \sigma_v^2\, \boldsymbol{I}_N, \end{aligned} \tag{2.3}$$

Measurement update:

$$\begin{aligned} \boldsymbol{G}_t &= \boldsymbol{F}_{t|t-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{F}_{t|t-1}\boldsymbol{H}^{\mathrm{T}} + \sigma_w^2\, \boldsymbol{I}_K)^{-1}, \\ \hat{\boldsymbol{\phi}}_{t|t} &= \hat{\boldsymbol{\phi}}_{t|t-1} + \boldsymbol{G}_t(\boldsymbol{x}_t - \boldsymbol{H}\hat{\boldsymbol{\phi}}_{t|t-1}), \\ \boldsymbol{F}_{t|t} &= \boldsymbol{F}_{t|t-1} - \boldsymbol{G}_t\boldsymbol{H}\boldsymbol{F}_{t|t-1}, \end{aligned} \tag{2.4}$$

where $\boldsymbol{F}_{t|t-1}$ and $\boldsymbol{F}_{t|t}$ denote the estimates of the state covariance matrix based on the measurements up to $t-1$ and t, respectively. Moreover, $\boldsymbol{G}_t$ is the Kalman gain matrix at time t.

We next demonstrate the effect of cyber-attacks on the state estimation mechanism via an illustrative example. In IEEE-14 bus power system with system parameters chosen as $\boldsymbol{A} = \boldsymbol{I}_N$, $\sigma_v^2 = 10^{-4}$, and $\sigma_w^2 = 2 \times 10^{-4}$, we consider a random FDI attack with various magnitude/intensity levels and show how the mean squared state estimation error of the Kalman filter changes when FDI attacks are launched on the system. We assume that the attacks are launched at $\tau = 100$, that is, the sys-

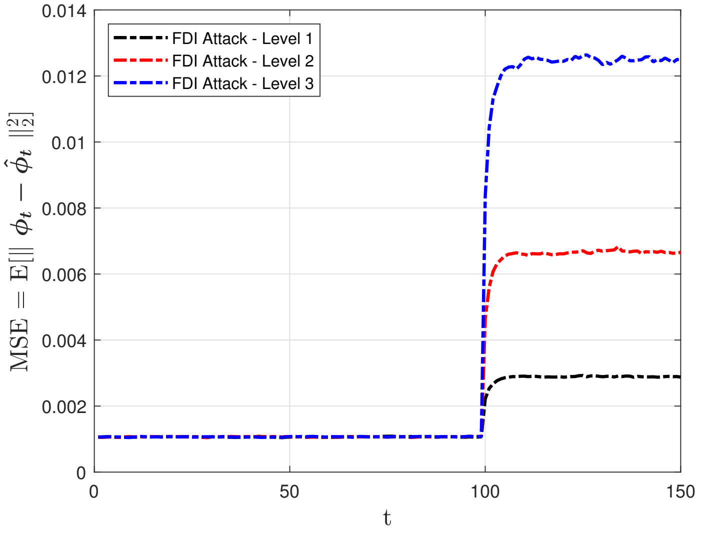


Figure 2.1: Mean squared state estimation error vs. time where random FDI attacks with various magnitude levels are launched at time $\tau = 100$.

tem is operated under normal (non-anomalous) conditions up to time 100 and under attacking conditions afterwards. Three attack magnitude levels are considered:

- Level 1: $b_{k,t} \sim \mathcal{U}[-0.04, 0.04]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$,
- Level 2: $b_{k,t} \sim \mathcal{U}[-0.07, 0.07]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$,
- Level 3: $b_{k,t} \sim \mathcal{U}[-0.1, 0.1]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$,

where $\mathcal{U}[\zeta_1, \zeta_2]$ denotes a uniform random variable in the range of $[\zeta_1, \zeta_2]$. The corresponding mean squared error (MSE) versus time curves are presented in Fig. 2.1. We observe that in case of cyber-attacks, the state estimates are deviated from the actual system states where the amount of deviation increases with the attack magnitude.

2.3 Problem Formulation

Before we introduce our problem formulation, we briefly explain a POMDP setting as follows. Given an agent and an environment, a discrete-time POMDP is defined by the seven-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \mathcal{G}, \gamma)$ where $\mathcal{S}$ denotes the set of (hidden) states of the environment, $\mathcal{A}$ denotes the set of actions of the agent, $\mathcal{T}$ denotes the set of conditional transition probabilities between the states, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the reward function that maps the state-action pairs to rewards, $\mathcal{O}$ denotes the set of observations of the agent, $\mathcal{G}$ denotes the set of conditional observation probabilities, and $\gamma \in [0, 1]$ denotes a discount factor that indicates how much present rewards are preferred over future rewards.

At each time t, the environment is in a particular hidden state $s_t \in \mathcal{S}$. Obtaining an observation $o_t \in \mathcal{O}$ depending on the current state of the environment with the probability $\mathcal{G}(o_t|s_t)$, the agent takes an action $a_t \in \mathcal{A}$ and receives a reward $r_t = \mathcal{R}(s_t, a_t)$ from the environment based on its action and the current state of the environment. At the same time, the environment makes a transition to the next state s_{t+1} with the probability $\mathcal{T}(s_{t+1}|s_t, a_t)$. The process is repeated until a terminal state is reached. In this process, the goal of the agent is to determine an optimal policy $\pi : \mathcal{O} \to \mathcal{A}$ that maps observations to actions and maximizes the agent's expected total discounted reward, that is, $\mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t r_t\big]$. Equivalently, if an agent receives costs instead of rewards from the environment, then the goal is to minimize the expected total discounted cost. Considering the latter, the POMDP problem can be written as follows:

$$\min_{\pi:\, \mathcal{O} \to \mathcal{A}} \; \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big]. \tag{2.5}$$

Next, we explain the online attack detection problem in a POMDP setting. We assume that at an unknown time τ, a cyber-attack is launched to the system and our aim is to detect the attack as quickly as possible after it occurs, where the attacker's capabilities/strategies are completely unknown. This defines a QCD problem where the aim is to minimize the average detection delay (ADD) as well as the false alarm rate (FAR). This problem can, in fact, be expressed as a POMDP problem (see Fig. 2.2). In particular, due to the unknown attack launch time τ, there are two hidden states: *pre-attack* and *post-attack*. At each time t, after obtaining the measurement

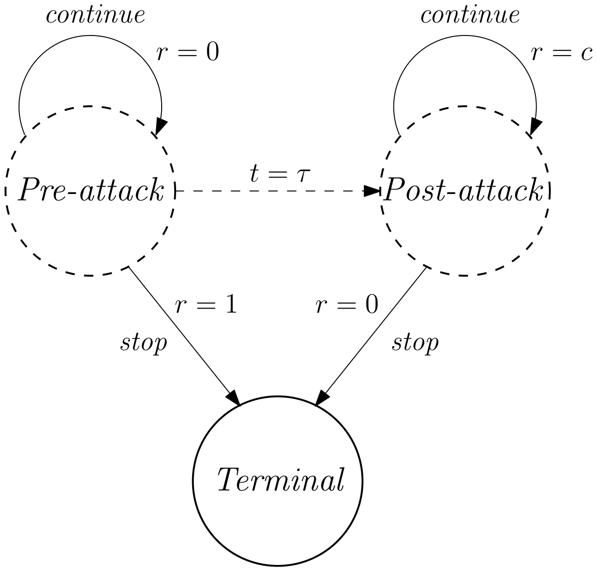


Figure 2.2: State-machine diagram for the considered POMDP setting. The hidden states and the (hidden) transition between them happening at time $t = \tau$ are illustrated with the dashed circles and the dashed line, respectively. The defender receives costs (r) depending on its actions and the underlying state of the environment. Whenever the defender chooses the action *stop*, the system moves into a *terminal* state and the defender receives no further cost.

vector $\boldsymbol{x}_t$, two actions are available for the agent (defender): *stop* and declare an attack or *continue* to have further measurements. We assume that whenever the action *stop* is chosen, the system moves into a *terminal* state, and always stays there afterwards.

Furthermore, although the conditional observation probability for the *pre-attack* state can be inferred based on the system model under normal operating conditions, since the attacking strategies are unknown, the conditional observation probability for the *post-attack* state is assumed to be totally unknown. Moreover, due to the unknown attack launch time τ, state transition probability between the *pre-attack* and the *post-attack* states is unknown.

Since our aim is to minimize the ADD as well as the FAR, both the false alarm and the detection delay events should be associated with some costs. Let the relative cost of a detection delay compared to a false alarm event be $c > 0$. Then, if the true underlying state is *pre-attack* and the action *stop* is chosen, a false alarm occurs and the defender receives a cost of 1. On the other hand, if the underlying state is *post-attack* and the action *continue* is chosen, then the defender receives a cost of c due to the detection delay. For all other (hidden) state-action pairs, the cost is assumed to be zero. Also, once the action *stop* is chosen, the defender does not receive any further costs while staying in the *terminal* state. The objective of the defender is to minimize its expected total cost by properly choosing its actions. Particularly, based on its observations, the defender needs to determine the stopping time at which an attack is declared.

Let Γ denote the stopping time chosen by the defender. Moreover, let $\mathbb{P}_k$ denote the probability measure if the attack is launched at time k, that is, $\tau = k$, and let $\mathbb{E}_k$ denote the corresponding expectation. Note that since the attacking strategies are unknown, $\mathbb{P}_k$ is assumed to be unknown. For the considered online attack detection problem, we can derive the expected total discounted cost as follows:

$$\begin{aligned}\mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big] &= \mathbb{E}_\tau\Big[\mathbb{1}\{\Gamma < \tau\} + \sum_{t=\tau}^{\Gamma} c\Big] \\ &= \mathbb{E}_\tau\big[\mathbb{1}\{\Gamma < \tau\} + c\,(\Gamma - \tau)^+\big] \\ &= \mathbb{P}_\tau(\{\Gamma < \tau\}) + c\,\mathbb{E}_\tau\big[(\Gamma - \tau)^+\big], \end{aligned} \tag{2.6}$$

where $\gamma = 1$ is chosen since the present and future costs are equally weighted in our problem, $\{\Gamma < \tau\}$ is a false alarm event that is penalized with a cost of 1, and $\mathbb{E}_\tau\big[(\Gamma - \tau)^+\big]$ is the ADD where each detection delay is penalized with a cost of c and $(\cdot)^+ = \max(\cdot, 0)$.

Based on Eq. (2.5) and Eq. (2.6), the online attack detection problem can be written as follows:

$$\min_{\Gamma}\ \mathbb{P}_\tau(\{\Gamma < \tau\}) + c\,\mathbb{E}_\tau\big[(\Gamma - \tau)^+\big]. \tag{2.7}$$

Since c corresponds to the relative cost between the false alarm and the detection delay events, by varying c and solving the corresponding problem in Eq. (2.7), a tradeoff curve between ADD and FAR can be obtained. Moreover, $c < 1$ can be chosen to prevent frequent false alarms.

Since the exact POMDP model is unknown due to unknown attack launch time τ and the unknown attacking strategies and since the RL algorithms are known to be effective over uncertain environments, we follow a model-free RL approach to obtain an approximate solution to Eq. (2.7). Then, a direct mapping from observations to the actions, that is, the stopping time Γ, needs to be learned. Note that the optimal action is *continue* if the underlying state is *pre-attack* and *stop* if the underlying state is *post-attack*. Then, to determine the optimal actions, the underlying state needs to be inferred using observations and the observation signal should be well informative to reduce the uncertainty about the underlying state. As described in Sec. 2.2, the defender observes the measurements $\boldsymbol{x}_t$ at each time t. The simplest approach can be forming the observation space directly with the measurement vector $\boldsymbol{x}_t$ but we would like to process the measurements and form the observation space with a signal related to the deviation of system from its normal operation.

Furthermore, it is, in general, possible to obtain identical observations in the *pre-attack* and the *post-attack* states. This is called perceptual aliasing and prevent us from making a good inference about the underlying state by only looking at the observation at a single time. We further note that in our problem, deciding on an attack solely based on a single observation corresponds to an outlier detection scheme for which more practical detectors are available not requiring a learning phase, see for example [87, 111]. However, we are particularly interested in detecting sudden and persistent attacks/anomalies that more likely happen due to an unfriendly intervention to the system rather than random disturbances due to high-level system noise realizations.

Since different states require different optimal actions, the ambiguity on the un-

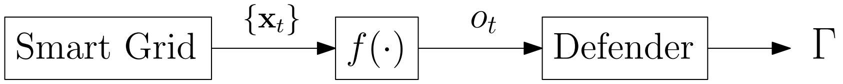


Figure 2.3: A graphical description of the online attack detection problem in the smart grid. The measurements $\{\boldsymbol{x}_t\}$ are collected through sensors and processed to obtain $o_t = f(\{\boldsymbol{x}_t\})$. The defender observes $f(\{\boldsymbol{x}_t\})$ at each time t and decides on the attack declaration time Γ.

derlying state should be further reduced with additional information derived from the history of observations. In fact, there may be cases where the entire history of observations is needed to determine the optimal solution in a POMDP problem [92]. However, due to computational limitations, only finite memory can be used in practice and an approximately optimal solution can be obtained. A simple approach is to use a finite-size sliding window of observations as a memory and map the most recent history window to an action, as described in [83]. This approach is particularly suitable for our problem as well since we assume persistent attacks/anomalies that happen at an unknown point of time and continue thereafter. That is, only the observations obtained after an attack are significant from the attack detection perspective.

Let the function that processes finite history of measurements and produces the observation signal be denoted with $f(\cdot)$ so that the observation signal at time t is $o_t = f(\{\boldsymbol{x}_t\})$, where $\{\boldsymbol{x}_t\}$ denotes a sequence of measurement vectors from recent history. Then, at each time, the defender observes $f(\{\boldsymbol{x}_t\})$ and decides on the stopping time Γ, as illustrated in Fig. 2.3. The aim of the defender is to obtain a solution to Eq. (2.7) by using an RL algorithm, as detailed in the subsequent section.

2.4 Solution Approach

First, we explain our methodology to obtain the observation signal $o_t = f(\{\boldsymbol{x}_t\})$. Note that the pdf of sensor measurements in the *pre-attack* state can be inferred using the baseline measurement model in Eq. (2.2) and the state estimates provided by the Kalman filter. In particular, the pdf of the measurements under normal operating conditions can be estimated as follows:

$$\boldsymbol{x}_t \sim \boldsymbol{\mathcal{N}}(\boldsymbol{H}\hat{\boldsymbol{\phi}}_{t|t}, \sigma_w^2 \boldsymbol{I}_K).$$

The likelihood of measurements based on the baseline density estimate, denoted with $L(\boldsymbol{x}_t)$, can then be computed as follows:

$$\begin{aligned} L(\boldsymbol{x}_t) &= (2\pi\sigma_w^2)^{-\frac{K}{2}} \exp\Big(\frac{-1}{2\sigma_w^2}(\boldsymbol{x}_t - \boldsymbol{H}\hat{\boldsymbol{\phi}}_{t|t})^{\mathrm{T}}(\boldsymbol{x}_t - \boldsymbol{H}\hat{\boldsymbol{\phi}}_{t|t})\Big) \\ &= (2\pi\sigma_w^2)^{-\frac{K}{2}} \exp\Big(\frac{-1}{2\sigma_w^2}\eta_t\Big), \end{aligned}$$

where

$$\eta_t \triangleq (\boldsymbol{x}_t - \boldsymbol{H}\hat{\boldsymbol{\phi}}_{t|t})^{\mathrm{T}}(\boldsymbol{x}_t - \boldsymbol{H}\hat{\boldsymbol{\phi}}_{t|t}) \tag{2.8}$$

is the estimate of the negative log-scaled likelihood.

In case the system is operated under normal conditions, the likelihood $L(\boldsymbol{x}_t)$ is expected to be high. Equivalently, small (close to zero) values of η_t may indicate the normal system operation. On the other hand, in case of an attack/anomaly, the system deviates from normal operating conditions and hence the likelihood $L(\boldsymbol{x}_t)$ is expected to decrease in such cases. Then, persistent high values of η_t over a time period may indicate an attack/anomaly. Hence, η_t may help to reduce the uncertainty about the underlying state to some extent.

However, since η_t can take any nonnegative value, the observation space is continuous and hence learning a mapping from each possible observation to an action is computationally infeasible. To reduce the computational complexity in such continuous spaces, we can quantize the observations. We then partition the observation space into I mutually exclusive and disjoint intervals using the quantization thresholds $\beta_0 = 0 < \beta_1 < \cdots < \beta_{I-1} < \beta_I = \infty$ so that if $\beta_{i-1} \leq \eta_t < \beta_i, i \in 1, \dots, I$, the observation at time t is represented with θ_i. Then, possible observations at any given time are $\theta_1, \dots, \theta_I$. Since θ_i's are representations of the quantization levels, each θ_i needs to be assigned to a different value.

Furthermore, as explained before, although η_t may be useful to infer the underlying state at time t, it is possible to obtain identical observations in the *pre-attack* and *post-attack* states. For this reason, we propose to use a finite history of observations. Let the size of the sliding observation window be M so that there are I^M possible observation windows and the sliding window at time t consists of the quantized versions of $\{\eta_j : t - M + 1 \leq j \leq t\}$. Henceforth, by an observation o, we refer to an observation window so that the observation space $\mathcal{O}$ consists of all possible observation windows. For instance, if $I = M = 2$, then $\mathcal{O} = \{[\theta_1, \theta_1], [\theta_1, \theta_2], [\theta_2, \theta_1], [\theta_2, \theta_2]\}$.

For each possible observation-action pair (o, a), we propose to learn a $Q(o, a)$ value (i.e., the expected future cost) using an RL algorithm where all $Q(o, a)$ values are stored in a Q-table of size $I^M \times 2$. After learning the Q-table, the policy of the defender will be choosing the action a with the minimum $Q(o, a)$ for each observation o. In general, increasing I and M improves the learning performance but at the same time results in a larger Q table, that would require to increase the number of training episodes and hence the computational complexity of the learning phase. Hence, I and M should be chosen considering the expected tradeoff between performance and computational complexity.

The considered RL-based detection scheme consists of learning and online detection phases. In the literature, state-action-reward-state-action (SARSA), a model-free RL control algorithm [126], was empirically shown to perform well over the model-free POMDP settings [102]. Hence, in the learning phase, the defender is trained with many episodes of experience using the SARSA algorithm and a Q-table is learned by the defender. For training, a simulation environment is created and during the training procedure, at each time, the defender takes an action based on its observation and receives a cost in return of its action from the simulation environment, as illustrated in Fig. 2.4. Based on this experience, the defender updates and learns a Q-table.

Algorithm 2.1 Learning Phase – SARSA Algorithm

```
Initialize $Q(o, a)$ arbitrarily, $\forall o \in \mathcal{O}$ and $\forall a \in \mathcal{A}$.
for $e = 1 : E$ do
    $t \leftarrow 0$
    $s \leftarrow$ pre-attack
    Choose an initial $o$ based on the pre-attack state and choose the initial $a =$ continue.
    while $s \neq$ terminal and $t < T$ do
        $t \leftarrow t + 1$
        if $a =$ stop then
            $s \leftarrow$ terminal
            $r \leftarrow \mathbb{1}\{t < \tau\}$
            $Q(o, a) \leftarrow Q(o, a) + \alpha\,(r - Q(o, a))$
        else if $a =$ continue then
            if $t >= \tau$ then
                $r \leftarrow c$
                $s \leftarrow$ post-attack
            else
                $r \leftarrow 0$
            end if
            Collect the measurements $\boldsymbol{x}_t$.
            Employ the Kalman filter using Eq. (2.3) and Eq. (2.4).
            Compute $\eta_t$ using Eq. (2.8) and quantize it to obtain $\theta_i$ if $\beta_{i-1} \leq \eta_t < \beta_i, i \in 1, \dots, I$.
            Update the sliding observation window $o$ with the most recent entry $\theta_i$ and obtain $o'$.
            Choose action $a'$ from $o'$ using the $\epsilon$-greedy policy based on the $Q$-table (that is being learned).
            $Q(o, a) \leftarrow Q(o, a) + \alpha\,(r + Q(o', a') - Q(o, a))$
            $o \leftarrow o'$, $a \leftarrow a'$
        end if
    end while
end for
Output: $Q$-table, that is, $Q(o, a)$, $\forall o \in \mathcal{O}$ and $\forall a \in \mathcal{A}$.
```

Then, in the online detection phase, based on the observations, the action with the lowest expected future cost (Q value) is chosen at each time using the previously learned Q-table. The online detection phase continues until the action *stop* is chosen by the defender. Whenever *stop* is chosen, an attack is declared and the process is terminated.

Note that after declaring an attack, whenever the system is recovered and returned back to the normal operating conditions, the online detection phase can be restarted. That is, once a defender is trained, no further training is needed. We summarize the learning and the online detection stages in Alg. 2.1 and Alg. 2.2, respectively. In Alg. 2.1, E denotes the number of learning episodes, T denotes the maximum length

Algorithm 2.2 Online Attack Detection

```
Input: $Q$-table learned in Alg. 2.1.
Choose an initial $o$ based on the *pre-attack* state and choose the initial $a =$ *continue*.
$t \leftarrow 0$
**while** $a \neq$ *stop* **do**
    $t \leftarrow t + 1$
    Collect the measurements $\boldsymbol{x}_t$.
    Determine the new $o$ as in the lines 20–22 of Alg. 2.1.
    $a \leftarrow \arg\min_a Q(o, a)$.
**end while**
Declare an attack and terminate the procedure.
```

of a learning episode, α is the learning rate, and ϵ is the exploration rate, where the ϵ-greedy policy chooses the action with the minimum Q value with probability $1-\epsilon$ and the other action (for exploration purposes during the learning process) with probability ϵ.

Since RL is an iterative procedure, same actions are repeated at each iteration (learning episode). The time complexity of an RL algorithm can then be considered as the time complexity of a single iteration [61]. As the SARSA algorithm performs one update on the Q-table at a time and the maximum time limit for a learning episode is T, the time complexity of Alg. 2.1 is $O(T)$. Moreover, the overall complexity of the learning procedure is $O(TE)$, as E is the number of learning episodes. Notice that the time complexity does not depend on the size of the action and observation spaces. On the other hand, as I and/or M increase, a larger Q-table needs to be learned, that requires to increase E for a better learning. Furthermore, the space complexity (memory cost) of Alg. 2.1 is $M + 2\,I^M$ due to the sliding observation window of size M and the Q-table of size $I^M \times 2$. Note that the space complexity is fixed over time. During the learning procedure, based on the smart grid model and some attack models (that are used to obtain low-magnitude attacks that correspond to small deviations from the normal system operation), we can obtain the measurement data online and the defender is trained with the observed data stream. Hence, the learning phase does not require storage of large amount of training data as only a sliding observation window of size M needs to be stored at each time.

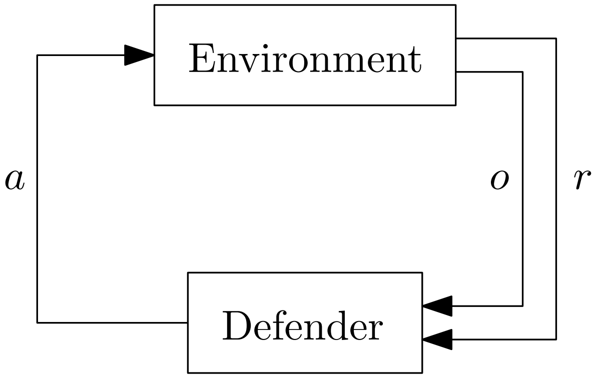


Figure 2.4: An illustration of the interaction between defender and the simulation environment during the learning procedure. The environment provides an observation o based on its internal state s, the agent chooses an action a based on its observation and receives a cost r from the environment in return of its action. Based on this experience, the defender updates $Q(o, a)$. This process is repeated many times during the learning procedure.

In Alg. 2.2, at each time, the observation o is determined and using the Q-table (learned in Alg. 2.1), the corresponding action a with the minimum cost is chosen. Hence, the complexity at a time is $O(1)$. This process is repeated until the action *stop* is chosen at the stopping time Γ. Furthermore, similarly to Alg. 2.1, the space complexity of Alg. 2.2 is $M + 2\, I^M$.

Remark 1: Since our solution approach is model-free, that is, it is not designed for specific types of attacks, the proposed detector does not distinguish between an attack and other types of persistent anomalies such as network topology faults. In fact, the proposed algorithm can detect any attack/anomaly as long as the effect of such attack/anomaly on the system is at a distinguishable level, that is, the estimated system states are at least slightly deviated from the actual system states. On the other hand, since we train the agent (defender) with low-magnitude attacks with some known attack types (to create the effect of small deviations from actual system operation in the *post-attack* state) and we test the proposed detector against various cyber-attacks in the numerical section (see Sec. 2.5), we lay the main emphasis on online attack detection in this chapter. In general, the proposed detector can be considered as an online anomaly detection algorithm.

Remark 2: The proposed solution scheme can be applied in a distributed smart grid system, where the learning and detection tasks are still performed at a single center but the sensor measurements are obtained in a distributed manner. We briefly explain this setup as follows:

- In the wide-area monitoring model of smart grids, there are several local control centers and a global control center. Each local center collects and processes measurements of a set of sensors in its neighborhood, and communicates with the global center and with the neighboring local centers, see for example [66].
- The system state is estimated in a distributed manner, for example, using the distributed Kalman filter designed for wide-area smart grids in [66].
- Let $\boldsymbol{h}_k^T \in \mathbb{R}^N$ be the kth row of the measurement matrix, $\boldsymbol{H}^T = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K]$. Then, estimate of the negative log-scaled likelihood, η_t, can be written as follows (see Eq. (2.8)):

$$\eta_t = \sum_{k=1}^{K} (x_{k,t} - \boldsymbol{h}_k^T \hat{\boldsymbol{\phi}}_{t|t})^2. \tag{2.9}$$

By employing the distributed Kalman filter, the local centers can estimate the system state at each time t. Then, they can compute the term $(x_{k,t} - \boldsymbol{h}_k^T \hat{\boldsymbol{\phi}}_{t|t})^2$ for the sensors in their neighborhood. Let the number of local centers be R and the set of sensors in the neighborhood of the rth local center be denoted with $\mathcal{S}_r$. Then, η_t in Eq. (2.9) can be rewritten as follows:

$$\begin{aligned} \eta_t &= \sum_{r=1}^{R} \underbrace{\sum_{k \in \mathcal{S}_r} (x_{k,t} - \boldsymbol{h}_k^T \hat{\boldsymbol{\phi}}_{t|t})^2}_{\eta_{t,r}} \\ &= \sum_{r=1}^{R} \eta_{t,r}. \end{aligned}$$

- In the distributed implementation, each local center can compute $\eta_{t,r}$ and report it to the global center, which then sums $\{\eta_{t,r}, r = 1, 2, \ldots, R\}$ and obtain η_t.

- The learning and detection tasks (Alg. 2.1 and Alg. 2.2) are performed at the global center in the same way as explained above.

2.5 Simulation Results

2.5.1 Simulation Setup and Parameters

Simulations are performed on an IEEE-14 bus power system that consists of $N+1 = 14$ buses and $K = 23$ sensors. The initial state variables (phase angles) are determined using the DC optimal power flow algorithm for case-14 in MATPOWER [150]. The system matrix $\boldsymbol{A}$ is chosen to be an identity matrix and the measurement matrix $\boldsymbol{H}$ is determined based on the IEEE-14 power system. The noise variances for the normal system operation are chosen as $\sigma_v^2 = 10^{-4}$ and $\sigma_w^2 = 2 \times 10^{-4}$.

For the proposed RL-based online attack detection scheme, the number of quantization levels is chosen as $I = 4$ and the quantization thresholds are chosen as $\beta_1 = 0.95 \times 10^{-2}$, $\beta_2 = 1.05 \times 10^{-2}$, and $\beta_3 = 1.15 \times 10^{-2}$ via an offline simulation by monitoring $\{\eta_t\}$ during the normal system operation. Further, $M = 4$ is chosen, that is, sliding observation window consists of 4 entries. Moreover, the learning parameters are chosen as $\alpha = 0.1$ and $\epsilon = 0.1$, and the episode length is chosen to be $T = 200$. In the learning phase, the defender is firstly trained over 4×10^5 episodes where the attack launch time is $\tau = 100$ and then trained further over 4×10^5 episodes where $\tau = 1$ to ensure that the defender sufficiently explores the observation space under normal operating conditions as well as the attacking conditions. More specifically, since a learning episode is terminated whenever the action *stop* is chosen and observations under an attack become available to the defender only for $t \geq \tau$, we choose $\tau = 1$ in the half of the learning episodes to make sure that the defender is sufficiently trained under the post-attack regime.

To illustrate the tradeoff between the ADD and PFA, the proposed algorithm is trained for both $c = 0.02$ and $c = 0.2$. Moreover, to obtain a detector that is robust and effective against small deviations of measurements from the normal system operation, the defender needs to be trained with very low-magnitude attacks that correspond to slight deviations from the baseline. For this purpose, some known attack types with low magnitudes are used. In particular, in one half of the learning episodes, random FDI attacks are used with attack magnitudes being realizations of the uniform random variable $\pm\mathcal{U}[0.02, 0.06]$, that is, $b_{k,t} \sim \mathcal{U}[0.02, 0.06]$ is the injected false datum to the kth sensor at time $t \geq \tau$, $\forall k \in \{1, \dots, K\}$. In the other half of the learning episodes, random hybrid FDI/jamming attacks are used where $b_{k,t} \sim \mathcal{U}[0.02, 0.06]$, $u_{k,t} \sim \mathcal{N}(0, \sigma_{k,t})$, and $\sigma_{k,t} \sim \mathcal{U}[2\times 10^{-4}, 4\times 10^{-4}]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$. The total training time costs are recorded approximately as 5018 seconds and 5106 seconds for $c = 0.2$ and $c = 0.02$, respectively.

2.5.2 Performance Evaluation

In this section, performance of the proposed RL-based attack detection scheme is evaluated and compared with some existing detectors in the literature. Firstly, we report the average false alarm period, $\mathbb{E}_\infty[\Gamma]$, of the proposed detection scheme, that is, the first time on the average the proposed detector gives an alarm although no attack/anomaly happens at all ($\tau = \infty$). The average false alarm periods are obtained approximately as $\mathbb{E}_\infty[\Gamma] = 9.4696 \times 10^5$ for $c = 0.2$ and $\mathbb{E}_\infty[\Gamma] = 7.9210 \times 10^6$ for $c = 0.02$. As expected, FAR of the proposed detector reduces as the relative cost of the false alarm event, $1/c$, increases.

Based on the optimization problem in Eq. (2.7), our performance metrics are the PFA (i.e., $\mathbb{P}_\tau(\{\Gamma < \tau\})$) and the ADD (i.e., $\mathbb{E}_\tau\big[(\Gamma - \tau)^+\big]$). Notice that both performance metrics depend on the unknown attack launch time τ. Hence, in general, the performance metrics need to be computed for each possible τ. For a representative

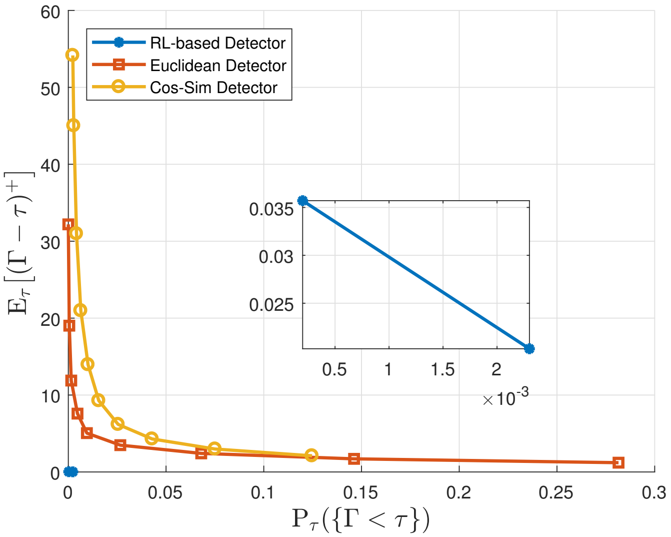


Figure 2.5: ADD vs. PFA curves for the proposed algorithm and the benchmark tests in case of a random FDI attack.

performance illustration, we choose τ as a geometric random variable with parameter ρ such that $P(\tau = k) = \rho\,(1-\rho)^{k-1}, k = 1, 2, 3, \ldots$ where $\rho \sim \mathcal{U}[10^{-4}, 10^{-3}]$ is a uniform random variable.

With Monte Carlo simulations over 10000 trials, we compute the PFA and the ADD of the proposed detector, the Euclidean detector [87], and the cosine-similarity metric based detector [111]. To obtain the performance curves, we vary the thresholds of the benchmark tests and vary c for the proposed algorithm. To evaluate the proposed algorithm, we use Alg. 2.2 that makes use of the Q-tables learned in Alg. 2.1 for $c = 0.02$ and $c = 0.2$. Furthermore, we report the precision, recall, and F-score for all simulation cases. As the computation of these measures requires computing the number of detected and missed trials, we define an upper bound on the detection delay (that corresponds to the maximum acceptable detection delay) such that if the attack is detected within this bound we assume the attack is detected, otherwise missed. As an example, we choose this bound as 10 time units. Then, we compute

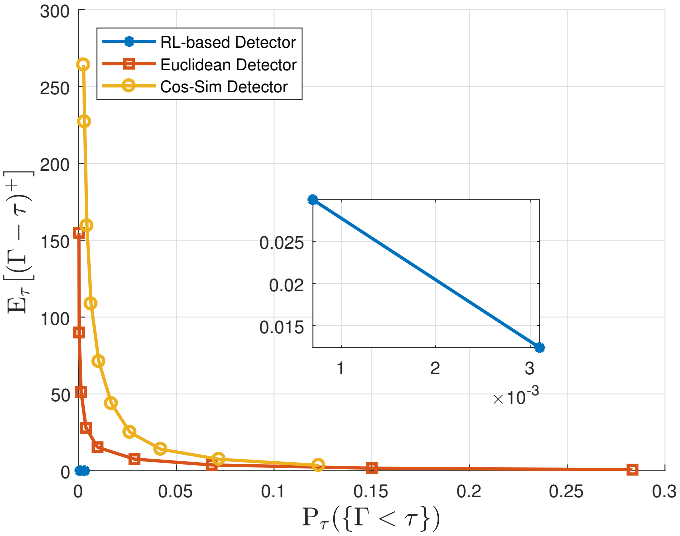


Figure 2.6: Performance curves for the proposed algorithm and the benchmark tests in case of a structured FDI attack.

the precision, recall, and F-score out of 10000 trials as follows:

$$\text{Precision} = \frac{\#\text{ trials } (\tau \le \Gamma \le \tau + 10)}{\#\text{ trials } (\tau \le \Gamma \le \tau + 10) + \#\text{ trials } (\Gamma < \tau)},$$

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

and

$$\text{F-score} = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where “# trials” means “the number of trials with”.

We evaluate the proposed and the benchmark detectors under the following attack scenarios:

1. Firstly, we evaluate the detectors against a random FDI attack where $b_{k,t} \sim \mathcal{U}[-0.07, 0.07]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \ge \tau$. The corresponding tradeoff curves are presented in Fig. 2.5.

2. We then evaluate the detectors against a structured FDI attack [82], where the injected data $\boldsymbol{b}_t$ lies on the column space of the measurement matrix $\boldsymbol{H}$. We

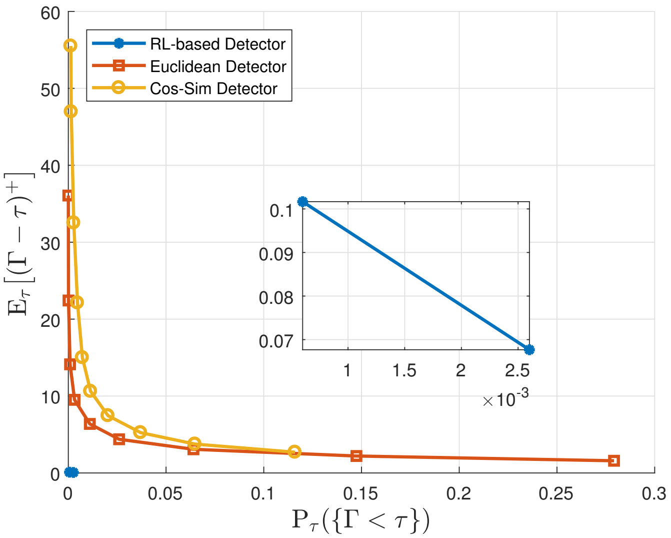


Figure 2.7: Performance curves for the proposed algorithm and the benchmark tests in case of a jamming attack with AWGN.

choose $\boldsymbol{b}_t = \boldsymbol{H}\boldsymbol{g}_t$ where $\boldsymbol{g}_t \triangleq [g_{1,t}, \dots, g_{N,t}]^{\mathrm{T}}$ and $g_{n,t} \sim \mathcal{U}[0.08, 0.12]$, $\forall n \in \{1, \dots, N\}$ and $\forall t \geq \tau$. The corresponding performance curves are illustrated in Fig. 2.6.

3. Then, we evaluate the detectors in case of a jamming attack with zero-mean AWGN where $u_{k,t} \sim \mathcal{N}(0, \sigma_{k,t})$ and $\sigma_{k,t} \sim \mathcal{U}[10^{-3}, 2 \times 10^{-3}]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$. The corresponding tradeoff curves are presented in Fig. 2.7.

4. Next, we evaluate the detectors in case of a jamming attack with jamming noise correlated over the sensors where $\boldsymbol{u}_t \sim \boldsymbol{\mathcal{N}}(\boldsymbol{0}, \boldsymbol{U}_t)$, $\boldsymbol{U}_t = \boldsymbol{\Sigma}_t \boldsymbol{\Sigma}_t^{\mathrm{T}}$, and $\boldsymbol{\Sigma}_t$ is a random Gaussian matrix with its entry at the ith row and the jth column is $\boldsymbol{\Sigma}_{t,i,j} \sim \mathcal{N}(0, 8 \times 10^{-5})$. The corresponding performance curves are given in Fig. 2.8.

5. Moreover, we evaluate the detectors under a hybrid FDI/jamming attack where $b_{k,t} \sim \mathcal{U}[-0.05, 0.05]$, $u_{k,t} \sim \mathcal{N}(0, \sigma_{k,t})$, and $\sigma_{k,t} \sim \mathcal{U}[5 \times 10^{-4}, 10^{-3}]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$. The corresponding tradeoff curves are presented in

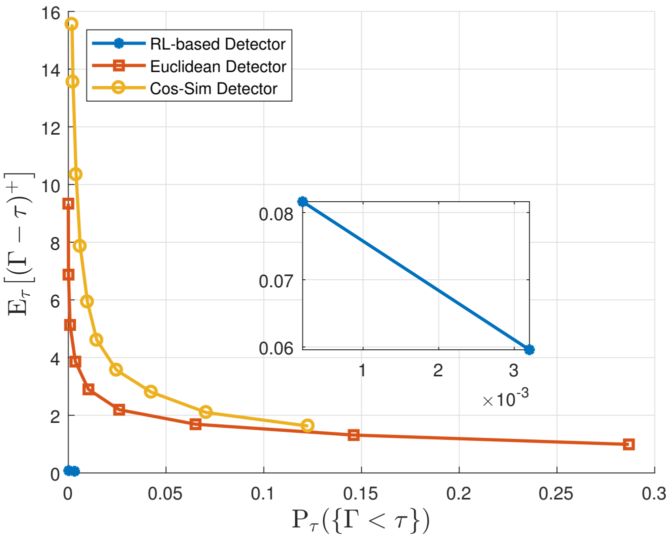


Figure 2.8: Performance curves for the proposed algorithm and the benchmark tests in case of a jamming attack with jamming noise correlated over the space.

Fig. 2.9.

6. Then, we evaluate the detectors in case of a random DoS attack where the measurement of each sensor become unavailable to the system controller at each time with probability 0.2. That is, for each sensor k, $d_{k,t}$ is 0 with probability 0.2 and 1 with probability 0.8 at each time $t \geq \tau$. The performance curves against the DoS attack are presented in Fig. 2.10.

7. Further, we consider a network topology attack where the lines between the buses 9-10 and 12-13 break down. The measurement matrix $\bar{\boldsymbol{H}}_t$ for $t \geq \tau$ is determined accordingly. The corresponding tradeoff curves are given in Fig. 2.11.

8. Finally, we consider a mixed topology and hybrid FDI/jamming attack, where the lines between buses 9-10 and 12-13 break down for $t \geq \tau$ and further, we have $b_{k,t} \sim \mathcal{U}[-0.05, 0.05]$, $u_{k,t} \sim \mathcal{N}(0, \sigma_{k,t})$, and $\sigma_{k,t} \sim \mathcal{U}[5 \times 10^{-4}, 10^{-3}]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$. The corresponding performance curves are presented in Fig. 2.12.

Table 2.1 and Table 2.2 summarize the precision, recall, and F-score for the proposed RL-based detector for $c = 0.2$ and $c = 0.02$, respectively against all the considered simulation cases above. Moreover, for the random FDI attack case, Fig. 2.13 illustrates the precision versus recall curves for the proposed and benchmark detectors. Since we obtain similar results for the other attack cases, we report the results for the random FDI attack case as a representative.

For almost all cases, we observe that the proposed RL-based detection scheme significantly outperforms the benchmark tests. This is because through the training process, the defender learns to differentiate the instantaneous high-level system noise from persistent attacks launched to the system. Then, the trained defender is able to significantly reduce its FAR. Moreover, since the defender is trained with low attack magnitudes, it becomes sensitive to detect small deviations of the system from its normal operation. On the other hand, the benchmark tests are essentially outlier detection schemes making sample-by-sample decisions and hence they are unable to distinguish high-level noise realizations from real attacks that makes such schemes more vulnerable to false alarms. Furthermore, in case of DoS attacks, since the sensor measurements become partially unavailable so that the system greatly deviates from its normal operation, all detectors are able to detect the DoS attacks with almost zero ADD (see Fig. 2.10).

Measure	FDI	Jamming	Corr. Jamm.	Hybrid	DoS	Structured FDI	Topology	Mixed
Precision	0.9977	0.9974	0.9968	0.9973	0.9977	0.9968	0.9972	0.9973
Recall	1	1	1	1	1	0.9756	0.9808	1
F-score	0.9988	0.9987	0.9984	0.9986	0.9988	0.9861	0.9890	0.9986

Table 2.1: Precision, recall, and F-score for the proposed detector ($c = 0.2$) in detection of various cyber-attacks.

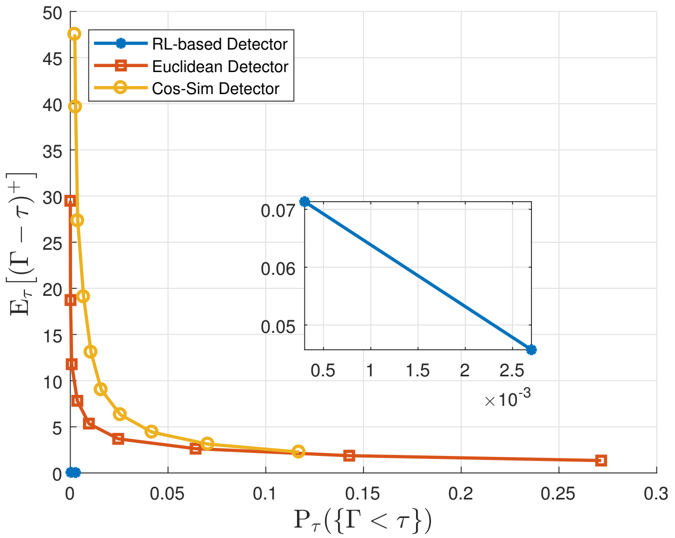


Figure 2.9: Performance curves for the proposed algorithm and the benchmark tests in case of a hybrid FDI/jamming attack.

Measure	FDI	Jamming	Corr. Jamm.	Hybrid	DoS	Structured FDI	Topology	Mixed
Precision	0.9998	0.9994	0.9998	0.9997	0.9995	0.9993	0.9999	0.9995
Recall	1	1	1	1	1	0.9449	0.9785	1
F-score	0.9999	0.9997	0.9999	0.9998	0.9997	0.9713	0.9891	0.9997

Table 2.2: Precision, recall, and F-score for the proposed detector ($c = 0.02$) in detection of various cyber-attacks.

2.6 Concluding Remarks

In this chapter, an online cyber-attack detection problem is formulated as a POMDP problem based on the QCD framework and a solution based on the model-free RL for POMDPs is proposed. The numerical studies illustrate the advantages of the proposed detection scheme in fast and reliable detection of cyber-attacks targeting the smart grid. The results also demonstrate the high potential of RL algorithms in solving complex cyber security problems.

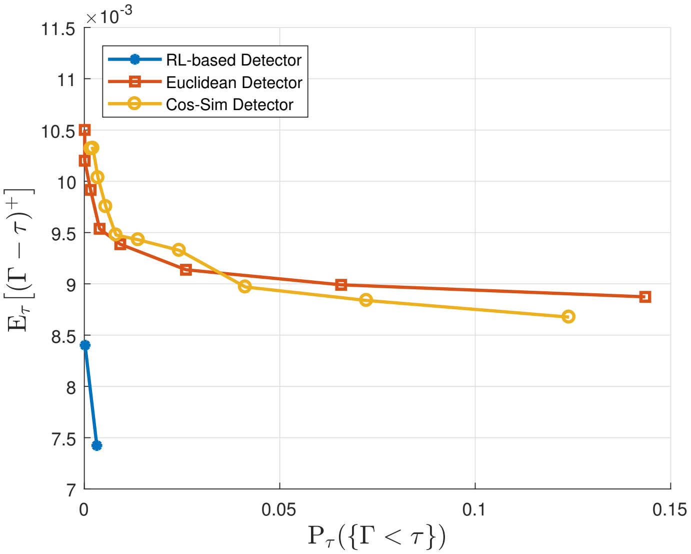


Figure 2.10: Performance curves for the proposed algorithm and the benchmark tests in case of a DoS attack.

Note that the proposed online detection method is widely applicable to any QCD problem where the pre-change model can be derived with some accuracy but the post-change model is unknown. This is, in fact, commonly encountered in many practical applications where the normal system operation can be modeled sufficiently accurately and the objective is the online detection of anomalies or attacks that are difficult to model in general. Moreover, depending on specific applications, if real post-change data can be obtained, the real data can be further enhanced with simulated data and the training can be performed accordingly, that would potentially improve the detection performance.

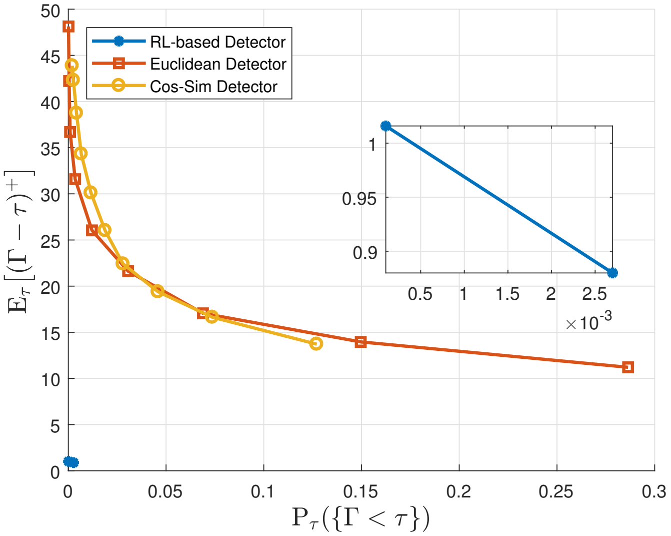


Figure 2.11: Performance curves for the proposed algorithm and the benchmark tests in case of a network topology attack.

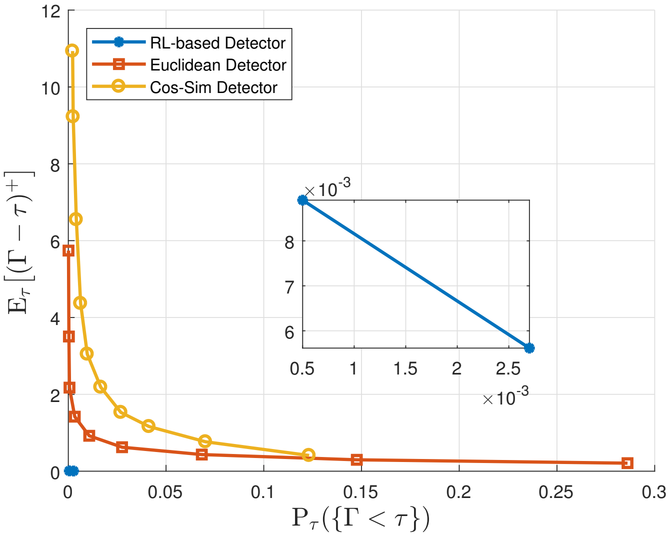


Figure 2.12: Performance curves for the proposed algorithm and the benchmark tests in case of a mixed network topology and hybrid FDI/jamming attack.

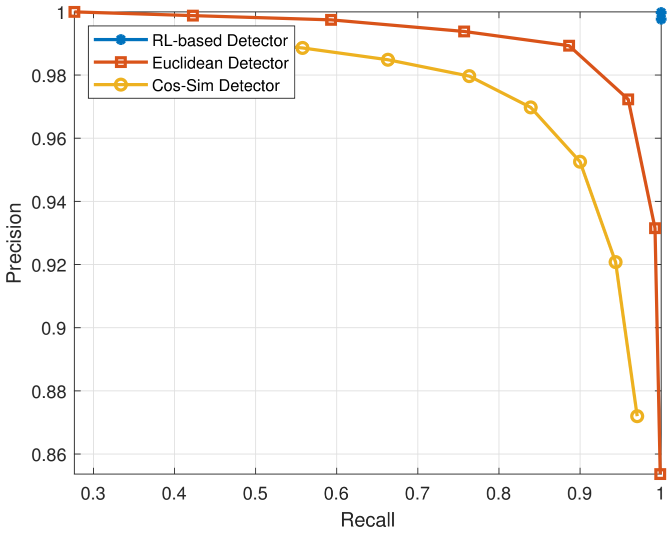


Figure 2.13: Precision vs. recall for the proposed and the benchmark detectors against a random FDI attack.

Chapter 3

Online Nonparametric Anomaly Detection for High-Dimensional Data Streams

3.1 Introduction

3.1.1 Background

Anomaly refers to deviation from the expected 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, and industrial damage detection [17, 60, 104]. Early and accurate detection of anomalies has a critical importance for safe and reliable operation of many modern systems such as the smart grid [64,69] and the Internet of Things (IoT) networks [91] that produce big (i.e., 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 QCD [5,108] 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 QCD algorithms are model-based, that is, 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 [5,65,108]. 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 [13, 67, 71]. On the other hand, in high-dimensional settings, for example, 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 (i.e., 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 (i.e., non-anomalous) 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 chapter, we consider two alternative summary statistics: (i) if the observed nominal data has a low intrinsic dimensionality, firstly learning 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, learning an acceptance region for the nominal data via the Geometric Entropy Minimization (GEM) [48, 122] 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 [17, 40, 52, 104] and to assume Gaussian noise or residual terms, see for example [139]. Such parametric approaches are powerful only if the observed data perfectly matches with the presumed model. On the other hand, nonparametric (i.e., model-free) data-driven techniques are robust to data model mismatch, that results in wider applicability of such techniques. Moreover, in high-dimensional settings, the lack of parametric models is common and complicated parameter-laden algorithms generally result in low performance, over-fitting, and bias towards particular anomaly types [74]. Hence, in this chapter, 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 analyzed more easily. Hence, our method should be mainly advantageous in high-dimensional settings where the statistical data models are unknown.

Conventional anomaly detection schemes ignore the temporal relation between anomalous data points and make sample-by-sample decisions [17, 104]. Such schemes are essentially outlier detectors that are vulnerable to false alarms since it is possible to observe non-persistent random outliers under normal system operation (i.e., no anomaly) due to for example, 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 the QCD [5, Sec. 2.2]. With the goal of making a reliable decision, we declare an anomaly only if there is a strong evidence for that. The sequential decision making based on the accumulated evidence also enables the detection of small but persistent changes, which would be missed by the outlier detectors.

3.1.2 Related Work

Batch algorithms are widely encountered in the anomaly detection literature [17, 104], that require the entire data before processing. Clearly, such techniques are not suitable in the online settings. For instance, in [31, 72], 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 for example [67, 129], 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 [110, 117] 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 [137] where the training data might contain a small number of anomalous data points or outliers. An extension of the one-class SVM algorithm [117] is proposed in [55] where the objective is to detect anomalies in the presence of multiple classes. Furthermore, in [48, 122, 149], NN graph-based anomaly detection schemes are proposed with sample-by-sample decisions. As discussed earlier, sequential decision making is more effective and reliable compared to sample-by-sample decisions. In [24, 37, 45], 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 [45] and the relative entropy (i.e., the Kullback-Leibler (KL) divergence) between the two samples in [24, 37]. Such approaches mainly suffer from low time resolution because they need large sample sizes for reliable decisions.

In [18], 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 (FAP). 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 [11], firstly 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 [143], 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.

3.1.3 Contributions

In this chapter, we propose online nonparametric anomaly detection schemes for high-dimensional data streams [68]. 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 online anomaly detection algorithm that makes use of the summary statistics.
- We provide an asymptotic lower bound and an asymptotic approximation for the FAP 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.

3.1.4 Organization

The remainder of the chapter is organized as follows. We present the problem description and our solution approach in Sec. 3.2, the proposed univariate summary statistics in Sec. 3.3, and the proposed online anomaly detection schemes with a false alarm rate analysis in Sec. 3.4. We then evaluate the proposed schemes over various application settings in Sec. 3.5. Finally, Sec. 3.6 concludes the chapter.

3.2 Problem Description and Solution Approach

3.2.1 Problem Description

We observe a high-dimensional stationary data stream, particularly, at each time t we acquire a new data point $\boldsymbol{x}_t \in \mathbb{R}^p$ 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, such as 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 due to an anomaly. Denoting the pdfs of $\boldsymbol{x}_t$ under regular (pre-change) and anomalous (post-change) conditions as $f_0^{\boldsymbol{x}}$ and $f_1^{\boldsymbol{x}} \neq f_0^{\boldsymbol{x}}$, respectively, we have

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

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 monitored system. In other words, we aim to detect the changes as quickly as possible after they occur. The framework of QCD well matches with this purpose. A well-known problem formulation in the QCD framework is the minimax problem proposed by Lorden [84]. 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 (ADD) is given by

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

where $(\cdot)^+ = \max\{0,\cdot\}$, $\mathcal{F}_t = \sigma(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t)$ is the sigma-algebra generated by the observations up to time t, and 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:

$$\inf_{\Gamma}\ J(\Gamma) \ \text{ subject to } \ \mathbb{E}_\infty[\Gamma] \geq \beta, \tag{3.1}$$

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

If both $f_0^{\boldsymbol{x}}$ and $f_1^{\boldsymbol{x}}$ are known, then the well-known CUSUM algorithm is the optimal solution to the minimax problem given in Eq. (3.1) [94]. Let

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

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

$$\begin{aligned} \Gamma &= \inf\{t : g_t \geq h\}, \\ g_t &= \max\{0, g_{t-1} + \ell_t\}, \end{aligned} \tag{3.2}$$

where $g_0 = 0$.

Since it is practically difficult to model all types of anomalies, $f_1^{\boldsymbol{x}}$ needs to be assumed unknown for a general anomaly detection problem. In that case, if only $f_0^{\boldsymbol{x}}$

is known and also has a parametric form, slight deviations from the parameters of $f_0^{\boldsymbol{x}}$ can be detected using a generalized CUSUM algorithm [5, Sec. 5.3], [66,67]. However, in a general high-dimensional problem, it might be difficult to model or estimate the high-dimensional multivariate nominal pdf $f_0^{\boldsymbol{x}}$. Hence, in this chapter, we assume that both $f_0^{\boldsymbol{x}}$ and $f_1^{\boldsymbol{x}}$ 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.

3.2.2 Proposed Solution Approach

Firstly, we assume that there is an available set of nominal data points $\mathcal{X} \triangleq \{\boldsymbol{x}_i : i = 1, 2, \dots, N\}$, 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. 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 $\boldsymbol{x}_t$ be denoted by d_t. Since the statistical properties of $\boldsymbol{x}_t$ 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 $\boldsymbol{x}_t$ 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 $\boldsymbol{x}_t$ is considered as an outlier with respect to the level of α if its p-value is less than α, that is, $p_t < \alpha$. Let

$$s_t \triangleq \log\left(\frac{\alpha}{p_t}\right). \tag{3.3}$$

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

Under normal system operation, we may observe random non-persistent outliers due to, for example, 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 Eq. (3.3) as a positive/negative statistical evidence for anomaly at time t, we can accumulate s_t's over time and obtain 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 Eq. (3.2)) with s_t:

$$\begin{aligned}\Gamma &= \inf\{t : g_t \geq h\},\\ g_t &= \max\{0, g_{t-1} + s_t\},\end{aligned} \tag{3.4}$$

where $g_0 = 0$.

In the following section, we present the proposed summary statistics. Then, in Sec. 3.4, we explain the estimation of the tail probability p_t (and hence s_t) based on the nominal summary statistics.

3.3 Summary Statistics

We explain our methodology to derive summary statistics for a general high-dimensional data stream. We then explain the derivation of summary statistics in a special case where the observed data exhibit a low intrinsic dimensionality.

3.3.1 GEM-based Summary Statistics

Given a nominal dataset $\mathcal{X}$ and a chosen significance level of α, the GEM method [48] 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 nearest neighbor (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 [122] is based on bipartite kNN graphs (BP-GEM). In this method, firstly $\mathcal{X}$ is uniformly 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 $\boldsymbol{x}_j \in \mathcal{S}_2$, the kNNs of $\boldsymbol{x}_j$ among the set $\mathcal{S}_1$ are determined. Denoting the Euclidean distance of $\boldsymbol{x}_j$ to its ith NN in $\mathcal{S}_1$ by $e_j(i)$, the sum of distances of $\boldsymbol{x}_j$ to its kNNs can be computed as follows:

$$d_j \triangleq \sum_{i=1}^{k} e_j(i). \tag{3.5}$$

After computing $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$, d_j's are sorted in ascending order and the $(1-\alpha)$ fraction of $\boldsymbol{x}_j$'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 $\boldsymbol{x}_t$, 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, that is,

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

then $\boldsymbol{x}_t$ 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 [122], given by

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

where $\delta(\mathcal{A})$ denotes the volume of $\mathcal{A}$. Hence, the BP-GEM method asymptotically achieves the most compact acceptance region for the nominal data.

If $\boldsymbol{x}_t$ is an outlier, then it falls outside the acceptance region $\mathcal{A}$, that is, 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 $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$ 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. 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 for example [43].

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 best fits to our purpose of extracting useful GEM-based summary statistics for anomaly detection.

3.3.2 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:

$$\boldsymbol{x}_t = \boldsymbol{y}_t + \boldsymbol{r}_t, \tag{3.6}$$

where $\boldsymbol{y}_t$ is the representation of $\boldsymbol{x}_t$ in a submanifold and $\boldsymbol{r}_t$ is the residual term, that is, the departure of $\boldsymbol{x}_t$ 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 is expected to deviate from the nominal submanifold and hence the magnitude of the residual term, that is, $\|\boldsymbol{r}_t\|_2$, 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 subsets of $\mathcal{X}$, that is, $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{X}$, with sizes N_1 and N_2, respectively, where $N_1, N_2 \leq N$. Using $\mathcal{S}_1$, we can learn a representative submanifold that the nominal data are embedded in. Then, using $\mathcal{S}_2$, we can compute the magnitude of the residual terms, that is, $\{\|\boldsymbol{r}_j\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$, as a set of nominal summary statistics.

There are various methods to determine the underlying submanifold, among which the PCA is well known for learning a linear submanifold, called the principal subspace [8, Sec. 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 [8, Sec. 12.1].

In the PCA method, denoting $\bar{\boldsymbol{x}}$ as the sample mean, that is,

$$\bar{\boldsymbol{x}} \triangleq \frac{1}{N_1} \sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \boldsymbol{x}_i \tag{3.7}$$

and $\boldsymbol{Q}$ as the sample data covariance matrix, that is,

$$\boldsymbol{Q} \triangleq \frac{1}{N_1} \sum_{\boldsymbol{x}_i \in \mathcal{S}_1} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{\mathrm{T}}, \tag{3.8}$$

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

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

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

$$\gamma \triangleq \frac{\sum_{j=1}^{r} \lambda_j}{\sum_{j=1}^{p} \lambda_j} \leq 1, \tag{3.9}$$

where the r-dimensional principal subspace is spanned by the orthonormal eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\ldots$, $\boldsymbol{v}_r$ corresponding to the r largest eigenvalues λ_1, λ_2, $\ldots$, λ_r of $\boldsymbol{Q}$. Let $\boldsymbol{V} \triangleq [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_r]$. The representation of $\boldsymbol{x}_t$ in the linear submanifold can then be determined as follows:

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

Then, the residual term can be computed as

$$\begin{aligned} \boldsymbol{r}_t &= \boldsymbol{x}_t - \boldsymbol{y}_t \\ &= (\boldsymbol{I}_p - \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}})(\boldsymbol{x}_t - \bar{\boldsymbol{x}}), \end{aligned} \tag{3.10}$$

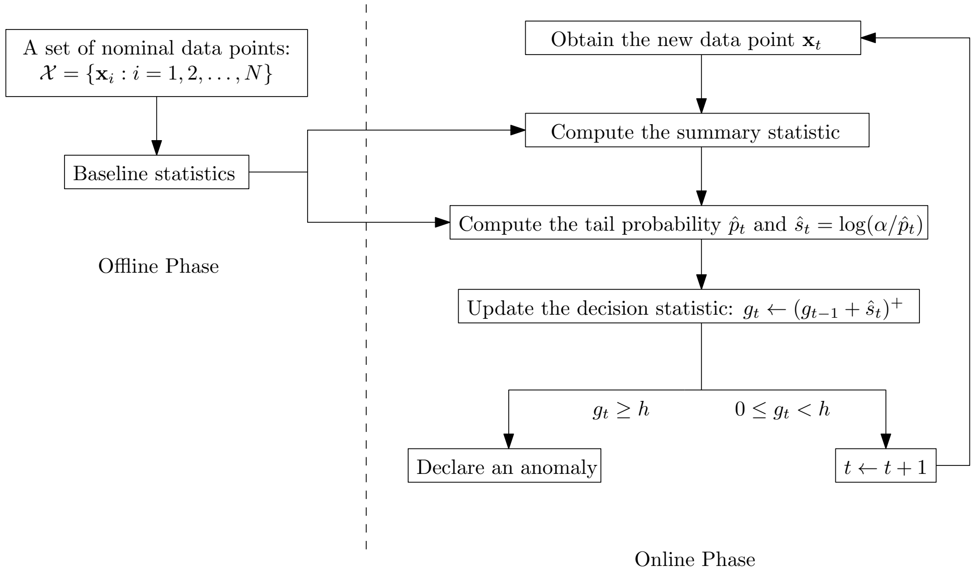

A set of nominal data points: $\mathcal{X} = \{\mathbf{x}_i : i = 1, 2, \ldots, N\}$
Baseline statistics
Offline Phase
Obtain the new data point $\mathbf{x}_t$
Compute the summary statistic
Compute the tail probability $\hat{p}_t$ and $\hat{s}_t = \log(\alpha/\hat{p}_t)$
Update the decision statistic: $g_t \leftarrow (g_{t-1} + \hat{s}_t)^+$
$g_t \geq h$
$0 \leq g_t < h$
Declare an anomaly
$t \leftarrow t + 1$
Online Phase

Figure 3.1: Diagram of the proposed detection schemes.

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

To obtain the PCA-based nominal summary statistics, firstly, using $\mathcal{S}_1$, we compute $\boldsymbol{Q}$ based on Eq. (3.8), and then its eigenvalues and eigenvectors. Then, for a chosen γ (see Eq. (3.9)), we determine r and the corresponding $\boldsymbol{V}$. Finally, using $\mathcal{S}_2$ and Eq. (3.10), we compute $\{\|\boldsymbol{r}_j\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$, 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 Eq. (3.6) 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 $\boldsymbol{r}_t$ (see Eq. (3.6)) both for a given nominal dataset and also for the sequentially acquired out-of-sample data, without re-running the manifold learning algorithm.

3.4 Online Nonparametric Anomaly Detection

3.4.1 Proposed Algorithm

We firstly 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 $\|\boldsymbol{r}_t\|_2$, take higher values compared to non-outliers (see Sec. 3.3). Hence, outliers 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 could compute the corresponding right tail probability as follows:

$$p_t = \int_{d_t}^{\infty} f_0^d(z)dz = 1 - F_0^d(d_t), \tag{3.11}$$

where F_0^d is the cdf of d_t. If $p_t < \alpha$, we can then consider d_t (correspondingly $\boldsymbol{x}_t$) 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 $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$ be the set of nominal summary statistics. Then, the corresponding edf is given by

$$\hat{F}_{0,N_2}^d(z) \triangleq \frac{1}{N_2} \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j \leq z\}. \tag{3.12}$$

Moreover, by the Glivenko-Cantelli theorem, $\hat{F}_{0,N_2}^d$ pointwise almost surely converges to the actual cdf F_0^d as $N_2 \to \infty$ [132]. Then, we can estimate p_t based on $\hat{F}_{0,N_2}^d$ as follows:

$$\begin{aligned} \hat{p}_t &= 1 - \hat{F}_{0,N_2}^d(d_t) \\ &= \frac{1}{N_2} \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j > d_t\}. \end{aligned} \tag{3.13}$$

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

Let

$$\hat{s}_t \triangleq \log\left(\frac{\alpha}{\hat{p}_t}\right). \tag{3.14}$$

Notice that for an outlier $\boldsymbol{x}_t$ with respect to a level of α, we have $\hat{s}_t > 0$ and similarly, for a non-outlier $\boldsymbol{x}_t$, we have $\hat{s}_t \leq 0$. Then, by replacing $\hat{s}_t$ with s_t in Eq. (3.4), we propose the following model-free CUSUM-like anomaly detection algorithm:

$$\begin{aligned} \Gamma &= \inf\{t : g_t \geq h\}, \\ g_t &= \max\{0, g_{t-1} + \hat{s}_t\}, \end{aligned} \tag{3.15}$$

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.

3.4.1.1 Summary of the Proposed Schemes

We summarize the proposed GEM-based and PCA-based detection schemes in Alg. 3.1 and Alg. 3.2, respectively. Moreover, we present a diagram of the proposed schemes in Fig. 3.1. The proposed schemes consist of an offline phase for extracting the baseline statistics for a given set of nominal data and an online phase for anomaly detection. The offline phases are explained in Sec. 3.3. In the online phase, at each time t, a new data point $\boldsymbol{x}_t$ is observed and using the baseline statistics, the summary statistic corresponding to $\boldsymbol{x}_t$ 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 further data. Note that

[1] In case where $\sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j > d_t\} = 0$, we have $\hat{p}_t = 0$ (see Eq. (3.13)), and hence $g_t = \infty$. In this case, a small nonzero value, for example, $1/N_2$, 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).

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 can be employed for online anomaly detection.

3.4.1.2 Space and Time Complexity

Algorithm 3.1: In the offline phase, Alg. 3.1 computes and stores $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$ and for the computation of any d_j, it stores the smallest k distances (i.e., $e_j(i)$'s), see Eq. (3.5). Hence, the space complexity of the offline phase is $O(k + N_2)$. For each $\boldsymbol{x}_j \in \mathcal{S}_2$, 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 $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$. 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 $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$ are larger than d_t, which is equivalent to determine the position of d_t among the sorted set of $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$.
Algorithm 3.2: The offline phase of Alg. 3.2 requires the storage of $\{\|\boldsymbol{r}_j\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$ and $(\boldsymbol{I}_p - \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}})$. 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 $\|\boldsymbol{r}_j\|_2$ for each $\boldsymbol{x}_j$ 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 $\boldsymbol{r}_t$ to compute $\|\boldsymbol{r}_t\|_2$ at any time t. Hence, the space complexity is $O(p)$. Moreover, for an incoming data point $\boldsymbol{x}_t$, $\|\boldsymbol{r}_t\|_2$ is computed and its position in the sorted set of $\{\|\boldsymbol{r}_j\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$ is determined, which has a time complexity of $O(p^2 + \log(N_2))$ at each time t.

Table 3.1 summarizes the space and time complexity of the proposed algorithms.

Algorithm 3.1 GEM-Based Online Nonparametric Anomaly Detection

```
Offline Phase
Uniformly 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 $j : \boldsymbol{x}_j \in \mathcal{S}_2$ do
    Search for the $k$NNs of $\boldsymbol{x}_j$ among the set $\mathcal{S}_1$.
    Compute $d_j$ using Eq. (3.5).
end for
Sort $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$ 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 $\boldsymbol{x}_t$.
    Search for the $k$NNs of $\boldsymbol{x}_t$ among the set $\mathcal{S}_1$ and compute $d_t$ using Eq. (3.5).
    $\hat{p}_t = \frac{1}{N_2} \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j > d_t\}$.
    $\hat{s}_t = \log(\alpha/\hat{p}_t)$.
    $g_t \leftarrow \max\{0, g_{t-1} + \hat{s}_t\}$.
end while
Declare an anomaly and stop the procedure.
```

		Space	Time
Algorithm 3.1	Offline	$O(k + N_2)$	$O(N_2(N_1 p + k + \log(N_2)))$
	Online	$O(k)$	$O(N_1 p + k + \log(N_2))$
Algorithm 3.2	Offline	$O(p^2 + N_2)$	$O(p^3 + p^2(N_1 + N_2) + N_2 \log(N_2))$
	Online	$O(p)$	$O(p^2 + \log(N_2))$

Table 3.1: Space and time complexity of the proposed algorithms.

3.4.2 Analysis

If the decision statistic g_t exceeds the test threshold h under regular conditions (no anomaly), then a false alarm is given. 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, firstly 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 3.1: In the absence of anomalies, that is, $\tau = \infty$, if $\alpha < 1/e$, where e

Algorithm 3.2 PCA-Based Online 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 $\bar{\boldsymbol{x}}$ and $\boldsymbol{Q}$ over $\mathcal{S}_1$ using Eq. (3.7) and Eq. (3.8), respectively.
Compute the eigenvalues $\{\lambda_j : j = 1, 2, \dots, p\}$ and the eigenvectors $\{\boldsymbol{v}_j : j = 1, 2, \dots, p\}$ of $\boldsymbol{Q}$.
Based on a desired level of $\gamma$ (see Eq. (3.9)), determine $r$ and form the matrix $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots \boldsymbol{v}_r]$.
for $j : \boldsymbol{x}_j \in \mathcal{S}_2$ do
    $\boldsymbol{r}_j = (\boldsymbol{I}_p - \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}})(\boldsymbol{x}_j - \bar{\boldsymbol{x}})$.
    Compute $\|\boldsymbol{r}_j\|_2$.
end for
Sort $\{\|\boldsymbol{r}_j\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$ 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 $\boldsymbol{x}_t$.
    $\boldsymbol{r}_t = (\boldsymbol{I}_p - \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}})(\boldsymbol{x}_t - \bar{\boldsymbol{x}})$ and compute $\|\boldsymbol{r}_t\|_2$.
    $\hat{p}_t = \frac{1}{N_2} \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \mathbb{1}\{\|\boldsymbol{r}_j\|_2 > \|\boldsymbol{r}_t\|_2\}$.
    $\hat{s}_t = \log(\alpha / \hat{p}_t)$.
    $g_t \leftarrow \max\{0, g_{t-1} + \hat{s}_t\}$.
end while
Declare an anomaly and stop the procedure.
```

denotes the Euler's number, it holds asymptotically that, as $N_2 \to \infty$,

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

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

Proof. See Appendix A. □

Theorem 3.1 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 Eq. (3.15), 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 FAP (also called the average run length), that is, the first time, on average, the upper threshold h is crossed in the absence of anomalies. 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, Sec. 5.2.2], [16, 58, 97, 113]. To be able to provide a performance guarantee regarding the false alarm rate, we firstly derive an asymptotic lower bound on the FAP of the proposed algorithm, as stated in the following theorem.

Theorem 3.2: For chosen $0 < \alpha < 1/e$ and $h > 0$, the FAP of the proposed algorithm, $\mathbb{E}_\infty[\Gamma]$, asymptotically (as $N_2 \to \infty$) achieves the following lower bound:

$$\mathbb{E}_\infty[\Gamma] \geq e^{(1-\theta)h}, \tag{3.16}$$

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

$$\theta = \frac{W(\alpha \log(\alpha))}{\log(\alpha)}, \tag{3.17}$$

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

Proof. See Appendix B. □

Based on Theorem 3.2, α and h can be chosen to asymptotically achieve the minimum acceptable level of FAP. 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

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

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

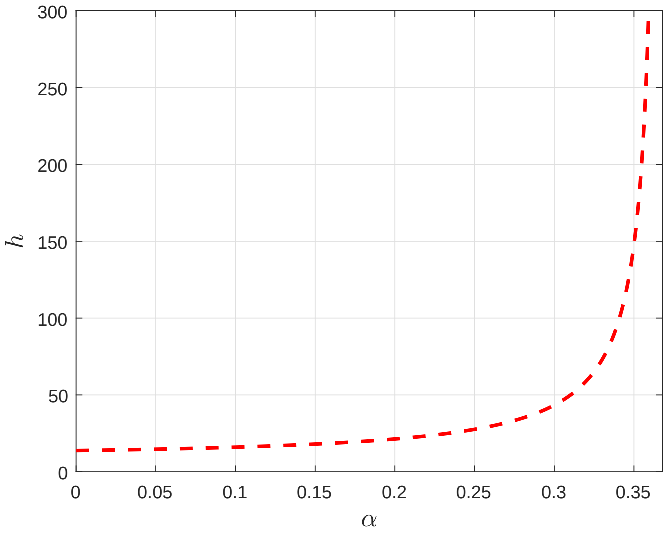


Figure 3.2: The lower bound (dashed curve) on the decision threshold h of the proposed algorithm for $\alpha < 1/e$ such that $\mathbb{E}_\infty[\Gamma] \geq 10^6$ as $N_2 \to \infty$.

providing a lower bound on the test threshold h for a chosen level of α. As an example, for $L = 10^6$, Fig. 3.2 illustrates the lower bound on h for $0 < \alpha < 1/e$.

Next, we investigate the tightness of the presented lower bound on the FAP in the asymptotic regime (as $N_2 \to \infty$). In particular, for different α levels, by varying the test threshold h, we plot the FAP versus the presented lower bound in Fig. 3.3. The figure shows that the FAP 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 propose the following (asymptotic) approximation to the FAP:

$$\mathbb{E}_\infty[\Gamma] \approx g(\alpha)\, e^{(1-\theta)h}, \tag{3.18}$$

where θ is as given in Eq. (3.17) and $g(\alpha)$ numerically computed over a Monte Carlo simulation is given in Fig. 3.4 and Table 3.2 for some α levels. Then, for a chosen $0 < \alpha < 1/e$ and for a desired FAP A, the test threshold h can be chosen as

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

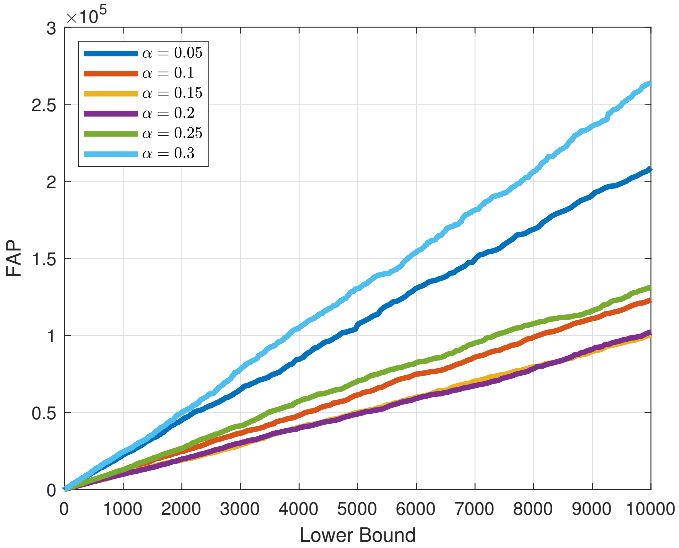


Figure 3.3: FAP vs. the presented lower bound in the asymptotic regime where $N_2 \to \infty$.

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

Table 3.2: $g(\alpha)$ computed over a Monte Carlo simulation for some α levels.

that asymptotically yields

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

Finally, note that lower α and/or higher h lead to a larger FAP and also a larger ADD. This is because lower α results in lower $\hat{s}_t$ and hence lower g_t, that increases the stopping time Γ (see Eq. (3.14) and Eq. (3.15)). Similarly, higher h results in a larger stopping time (see Eq. (3.15)). Hence, α and h are essentially tradeoff parameters that can be used to strike a desired balance between the false alarm rate and the ADD 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 ADD of the proposed algorithm. Nonetheless, we know that as the discrepancy, for example, the KL divergence, between the nominal

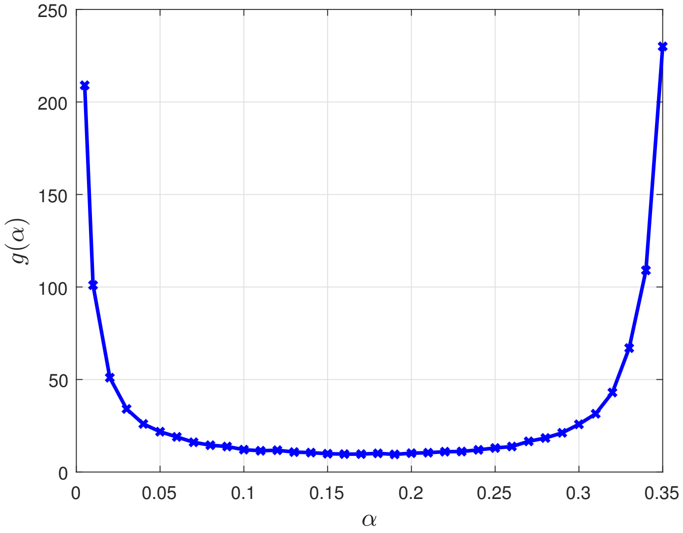


Figure 3.4: $g(\alpha)$ vs. α.

and anomalous pdfs increases, it is likely to observe more significant outliers after anomaly happens, that increases $\hat{s}_t$ (see Eq. (3.13) and Eq. (3.14)) for $t \geq \tau$, which in turn decreases the detection delays (see Eq. (3.15)).

Remark 1: For the proposed CUSUM-like test, we also derive the Wald's approximation to the FAP in the asymptotic regime where $N_2 \to \infty$. In particular, based on [5, Sec. 5.2.2.2] and with derivations similar to Theorem 3.2, for chosen $0 < \alpha < 1/e$ and $h > 0$ and as $N_2 \to \infty$, the Wald's approximation is given as follows:

$$\mathbb{E}_\infty[\Gamma] \approx \frac{1}{1+\log(\alpha)} \left(h + \frac{e^{(1-\theta)h} - 1}{\theta - 1} \right). \tag{3.19}$$

However, since the Wald's approximation ignores the excess over boundary (overshoot), it significantly underestimates the FAP as α decreases towards 0, as for smaller α, the decision statistic g_t of the proposed CUSUM-like test more frequently hits the lower threshold 0 during the random walk. On the other hand, the approximation gets better as α increases towards $1/e$. Fig. 3.5 shows the FAP, the Wald's approximation given in Eq. (3.19), and the lower bound presented in Theorem 3.2 for various α and h levels obtained via Monte Carlo simulations in the asymptotic regime as $N_2 \to \infty$.

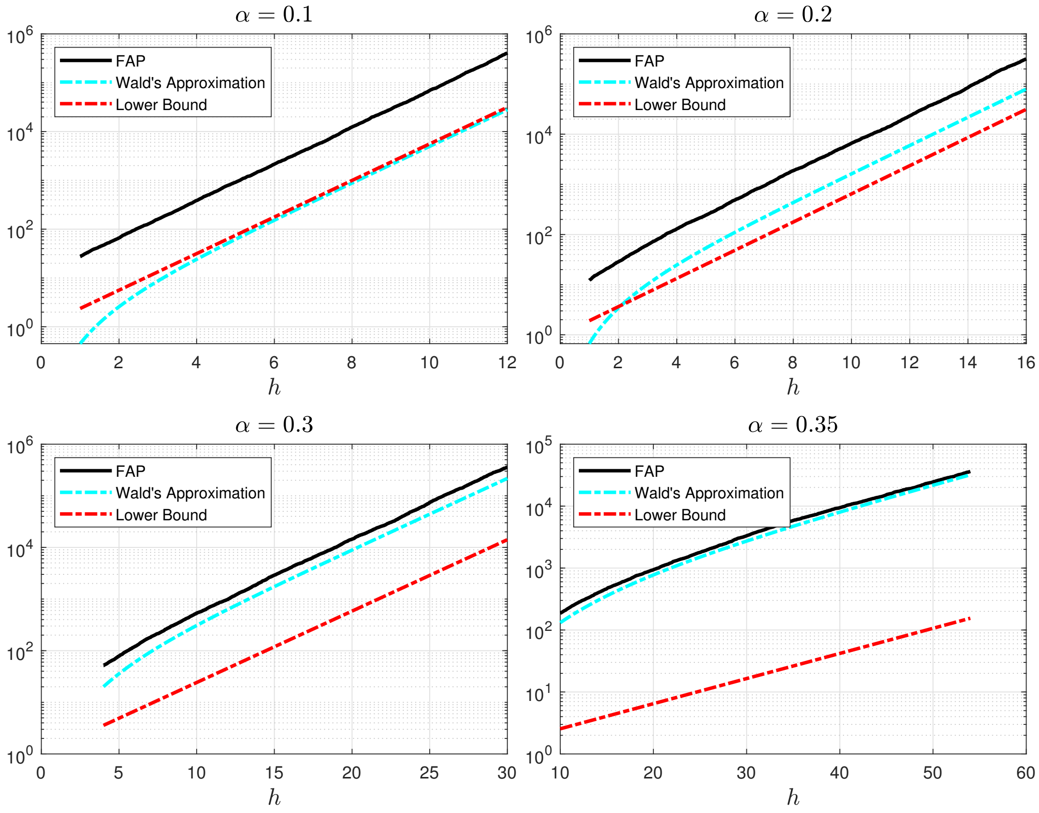


Figure 3.5: FAP, the Wald's approximation, and the presented lower bound for various α and h levels.

As expected, the Wald's approximation gets worse as α decreases, for example, it becomes even lower than the presented lower bound for $\alpha = 0.1$.

3.5 Performance Evaluation

In this section, we evaluate the performance of the proposed detection schemes. 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 over an IoT network. Throughout the section, we choose $\alpha = 0.2$ and make the aforementioned modification for the proposed algorithms: in case where $\sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \mathbb{1}\{d_j > d_t\} = 0$, we assign $\hat{p}_t = 1/N_2$ to prevent $g_t = \infty$ due to a single outlier. For all the proposed and the benchmark tests, we obtain the tradeoff curves between the ADD, $\mathbb{E}_\tau\big[(\Gamma - \tau)^+\big]$, and the FAP, $\mathbb{E}_\infty[\Gamma]$, 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 CUSUM-like 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) requires to count the number of detected and missed trials, we need to define an upper bound on the detection latency, that is, the maximum acceptable detection delay, such that if the anomaly/change is detected within this bound, we assume it is successfully detected, otherwise missed. In our experiments, 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:

$$\text{TPR} \triangleq \frac{\#\text{ trials } (\tau \leq \Gamma \leq \tau + 10)}{\#\text{ trials } (\tau \leq \Gamma \leq \tau + 10) + \#\text{ trials } (\Gamma > \tau + 10)},$$

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

$$\text{FAR} \triangleq \frac{1}{\mathbb{E}_\infty[\Gamma]}.$$

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

3.5.1 Benchmark Algorithms

3.5.1.1 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 [97] and the magnitude of the innovation sequence in the Kalman filter [142] are expected to increase in case of an anomaly (see Chapter 2) and several variants of the nonparametric CUSUM test have been proposed in the context of anomaly/attack detection in the smart grid [97, 142].

In our case, both summary statistics, that is, d_t and $\|\boldsymbol{r}_t\|_2$, 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 online anomaly detection. Let us specifically consider d_t and let

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

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

$$\begin{aligned} \Gamma &= \inf\{t : g_t \geq h\}, \\ g_t &= \max\{0, g_{t-1} + d_t - \bar{d}\}, \end{aligned} \tag{3.20}$$

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

3.5.1.2 Online Discrepancy Test (ODIT)

The ODIT presented in [143] is a sequential nonparametric anomaly detection algorithm based on the GEM. It consists of offline and online phases where its offline phase is identical to the offline phase of Alg. 3.1. In the online phase, instead of using the entire set $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$, only a threshold distance $d_{[K]}$ is used, denoting the largest Kth element among $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$. Specifically, 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 Eq. (3.20), after replacing $\bar{d}$ with $d_{[K]}$.

3.5.1.3 Information Theoretic Multivariate Change Detection (ITMCD) Algorithm

The ITMCD algorithm presented in [37] 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 $\mathcal{X}_{t,w_1} = \{\boldsymbol{x}_{t-w_1+1}, \dots, \boldsymbol{x}_t\}$ and $\mathcal{X}_{t,w_2} = \{\boldsymbol{x}_{t-w_1-w_2+1}, \dots, \boldsymbol{x}_{t-w_1}\}$. Moreover, let $e_{m,n}(i)$ denote the Euclidean distance between $\boldsymbol{x}_i \in \mathcal{X}_{t,w_m}$ 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 [37, 134]:

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

where $\boldsymbol{x}_t \in \mathbb{R}^p$. 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.

3.5.1.4 NN-Based Online Change Detection Algorithm

In [18], 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 $\mathcal{S}_W \triangleq \{\boldsymbol{x}_{t-W+1}, \dots, \boldsymbol{x}_t\}$ with indices $t_W \triangleq \{t - W + 1, \dots, t\}$, the stopping time is given by

$$\Gamma = \inf\{t : \max_{t-n_1 \leq m \leq t-n_0} Z_W(m,t) \geq h\}, \tag{3.21}$$

where

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

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

$$A_{t_W,ij} \triangleq \mathbb{1}\{\boldsymbol{x}_j \text{ is one of the } k\text{NNs of } \boldsymbol{x}_i \text{ among } \mathcal{S}_W\},$$

$$B_{ij}(m, t_W) \triangleq \mathbb{1}\big\{\big(P_{t_W}(i) \leq m, P_{t_W}(j) > m\big) \\ \text{or } \big(P_{t_W}(i) > m, P_{t_W}(j) \leq m\big)\big\},$$

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 [18, Sec. 2]. In our experiments, we choose $W = 50$, $k = 10$, $n_0 = 10$, and $n_1 = 40$.

The test in Eq. (3.21) 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 argued in [18], the method is mainly effective in the detection of sharp changes/anomalies, as otherwise the difference between two samples would not be significant. That 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 over IoT networks.

3.5.1.5 QuantTree

The QuantTree algorithm presented in [11] 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 prespecified 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 [67]. In this case, $y_1, \dots, y_K$ are determined based on the most recent W observations $\boldsymbol{x}_{t-W+1}, \dots, \boldsymbol{x}_t$. 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 experiments, we choose $W = 256$ and $K = 16$ with $\pi_i = 1/16, \forall i \in \{1, \dots, 16\}$.

3.5.2 Online Cyber-Attack Detection in Smart Grid

We consider the IEEE-57 bus power system that consists of 57 buses and 80 sensors. Let $\boldsymbol{\phi}_t \in \mathbb{R}^{57}$ denote the bus voltage angles (phases) and $\boldsymbol{x}_t \in \mathbb{R}^{80}$ denote the measurement vector collected through the sensors at time t. Suppose that the smart grid operates according to the following linearized DC model [1]:

$$\boldsymbol{x}_t = \boldsymbol{H}\, \boldsymbol{\phi}_t + \boldsymbol{\omega}_t, \tag{3.22}$$

where $\boldsymbol{H} \in \mathbb{R}^{80 \times 57}$ is the measurement matrix determined based on the power network topology and $\boldsymbol{\omega}_t \in \mathbb{R}^{80}$ is the measurement noise vector. Moreover, let

$$\boldsymbol{\omega}_t \sim \mathcal{N}(\mathbf{0}_{80}, \sigma^2 \boldsymbol{I}_{80}), \tag{3.23}$$

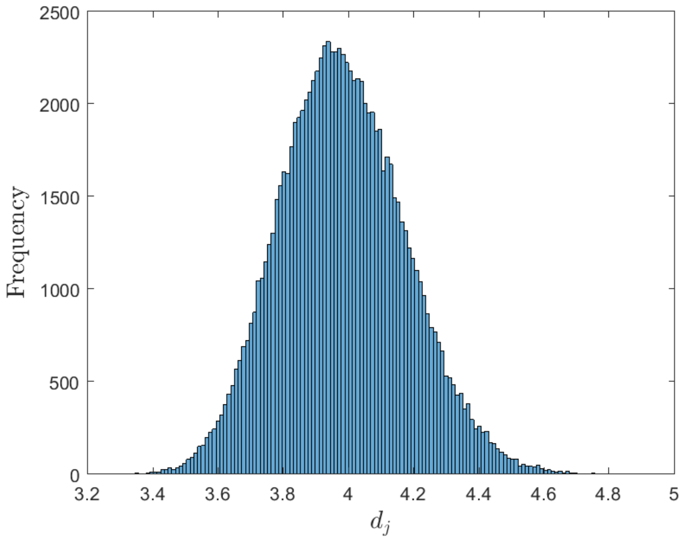


Figure 3.6: Histogram of the GEM-based nominal summary statistics for the IEEE-57 bus power system.

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 [150] and obtain the nominal voltage angles $\boldsymbol{\phi}_t$. 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 Eq. (3.22) defines the regular system operation. However, in case of an anomaly (e.g., a cyber-attack), the measurement model in Eq. (3.22) no longer holds. For instance, in case of an FDI attack launched at time τ, the measurement vector takes the following form:

$$\boldsymbol{x}_t = \boldsymbol{H}\,\boldsymbol{\phi}_t + \boldsymbol{a}_t + \boldsymbol{\omega}_t,\ t \geq \tau,$$

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

Based on Eq. (3.22) and Eq. (3.23), we have

$$\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{H}\,\boldsymbol{\phi}_t, \sigma^2 \boldsymbol{I}_{80}), \tag{3.24}$$

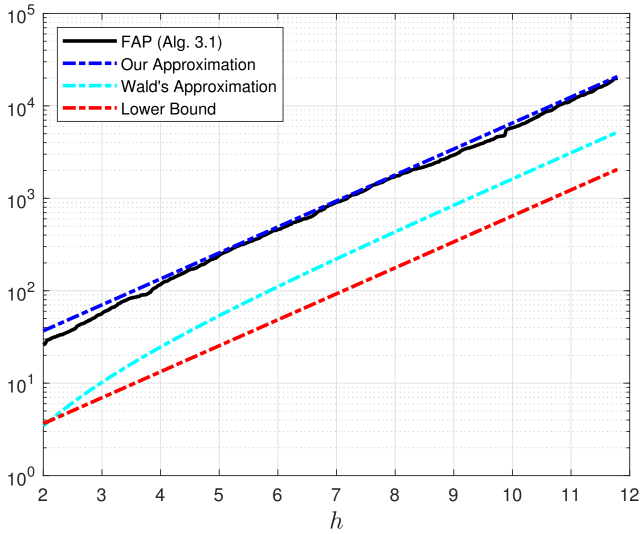


Figure 3.7: FAP of Alg. 3.1 for the smart grid data, the theoretical approximations, and the theoretical lower bound for various test thresholds.

that is, the nominal data covariance matrix is diagonal and every dimension has equal variance. If we collect a set of nominal data and perform the PCA, we can observe that every dimension is equally important so that the observed high-dimensional nominal data does not exhibit a low intrinsic dimensionality. Nevertheless, we can still use the proposed GEM-based detector (see Alg. 3.1).

In this setup, we generate synthetic data based on the system and attack models presented above. Specifically, during the normal system operation (see Eq. (3.24)), 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 $N_1 = 2 \times 10^3$ and $N_2 = 9.8 \times 10^4$, respectively. We choose $k = 4$ and for each data point $\boldsymbol{x}_j \in \mathcal{S}_2$, we compute d_j, the sum of distances of $\boldsymbol{x}_j$ to its kNNs among $\mathcal{S}_1$ (see Eq. (3.5)). Then, we obtain the histogram of $\{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$, as given in Fig. 3.6.

Fig. 3.7 shows the FAP of Alg. 3.1, the asymptotic lower bound given in Eq. (3.16), our asymptotic approximation to the FAP given in Eq. (3.18), and the Wald's asymptotic approximation given in Eq. (3.19), as the test threshold h varies. We observe

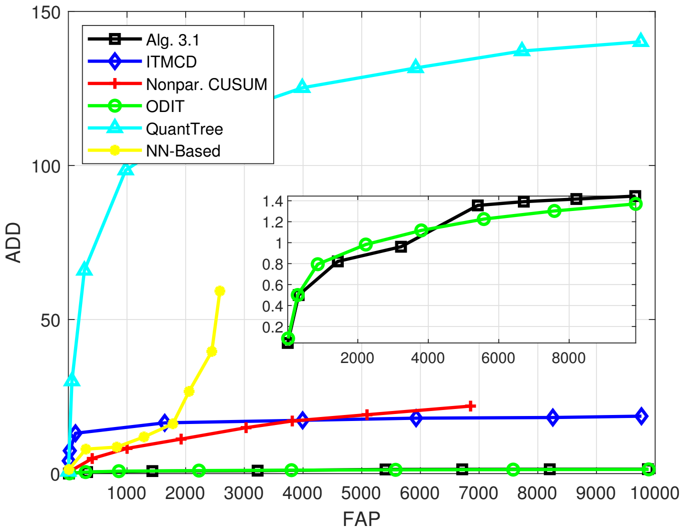


Figure 3.8: ADD vs. FAP in detection of an FDI attack against the smart grid.

that our approximation is quite close to the actual FAP. Furthermore, Fig. 3.8 and Fig. 3.9 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 illustrate that the proposed algorithm outperforms the benchmark tests or at least performs nearly with them. In this experiment, we also note that the detection delays critically depend on the attack magnitude, particularly, 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. 3.10, 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 the red dashed line, while staying near zero before the attack.

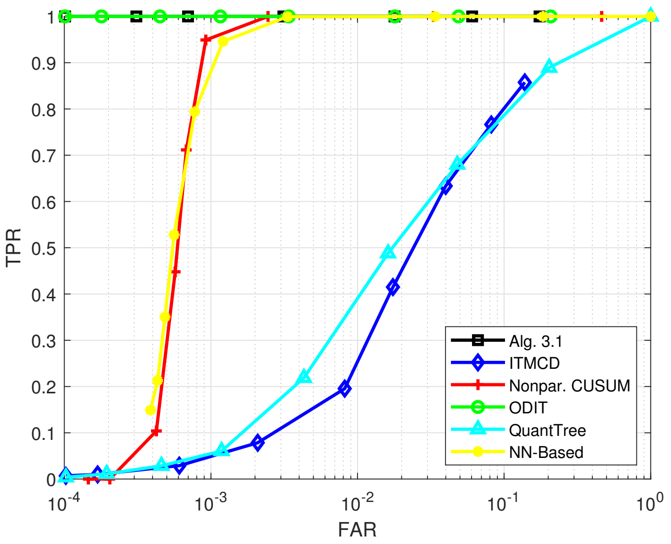


Figure 3.9: ROC curve in detection of an FDI attack against the smart grid.

3.5.3 Online Detection of Changes in Human Physical Activity

The Human Activities and Postural Transitions (HAPT) dataset [112] obtained from the UCI Machine Learning Repository [28] contain data for six physical activities: sitting, standing, laying, walking, walking upstairs, and walking downstairs. The first three, that is, 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 quickly 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 sampled uniformly among the set of data

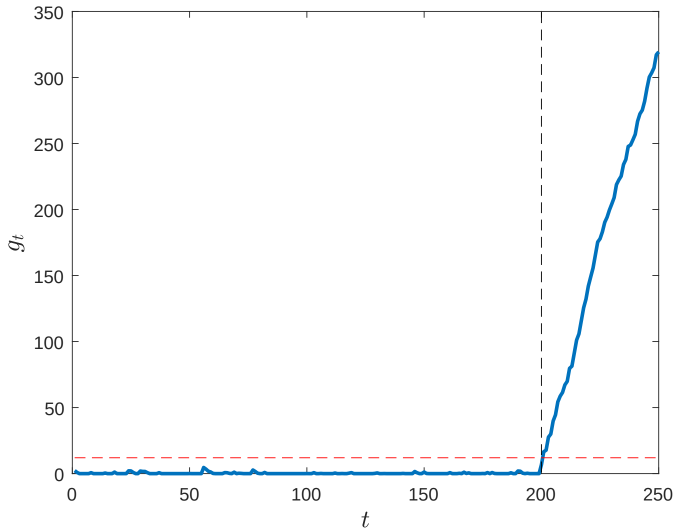


Figure 3.10: Sample path of the decision statistic where the FDI attack is launched at $\tau = 200$.

points corresponding to static activities and after the change-point, at each time, we observe a data point sampled uniformly from the set of dynamic activities.

We firstly uniformly select 2500 data points from the set of data points corresponding to static activities and using the PCA method (see Alg. 3.2), we obtain the eigenvalues of the corresponding sample data covariance matrix, as shown in descending order in Fig. 3.11. We observe through Fig. 3.11 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. 3.12.

In cases where the observed data stream exhibits a low intrinsic dimensionality, another approach is applying the proposed GEM-based detection scheme (Alg. 3.1) after dimensionality reduction. That is, after obtaining the matrix $\boldsymbol{V}$ as described in Alg. 3.2, each data point in the nominal training set, $\boldsymbol{x}_i \in \mathcal{X}$, and also each

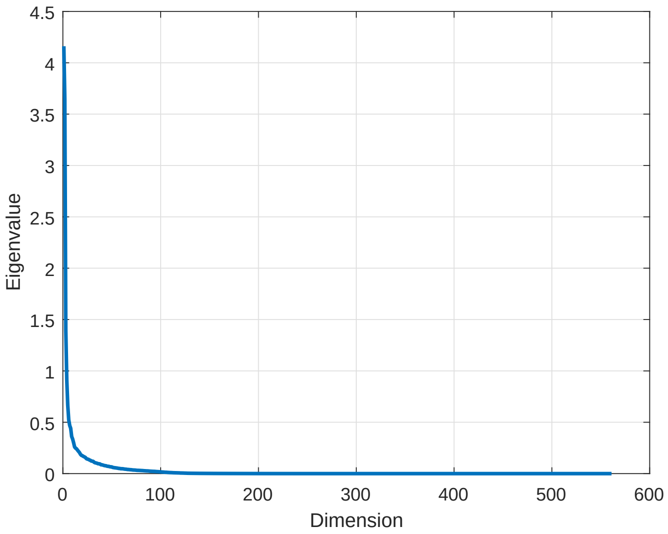


Figure 3.11: Eigenvalues of the sample data covariance matrix for a representative set of static activities in the HAPT dataset.

sequentially available data point, $\boldsymbol{x}_t$, can be projected onto a r-dimensional space as $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}_i$ and $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}_t$, respectively. Alg. 3.1 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 Alg. 3.1 over the projected data, where we obtain the projection matrix $\boldsymbol{V}$ as described above and uniformly choose $\mathcal{S}_1$ and $\mathcal{S}_2$ (in Alg. 3.1) with sizes $N_1 = 1000$ and $N_2 = 4738$, respectively. Fig. 3.13 and Fig. 3.14 show that the proposed algorithms perform superior or at least comparable to the benchmark algorithms. Furthermore, Fig. 3.15 illustrates the FAP, the asymptotic lower bound, and the asymptotic approximations for the proposed algorithms. In all the relevant figures, we use an asterisk for Alg. 3.1 to emphasize that it is employed based on the projected low-dimensional data.

3.5.4 Online Detection of IoT Botnet Attacks

Data for network-based detection of IoT botnet attacks (N-BaIoT) [91] obtained from the UCI Machine Learning Repository [28] contain network traffic statistics for an IoT

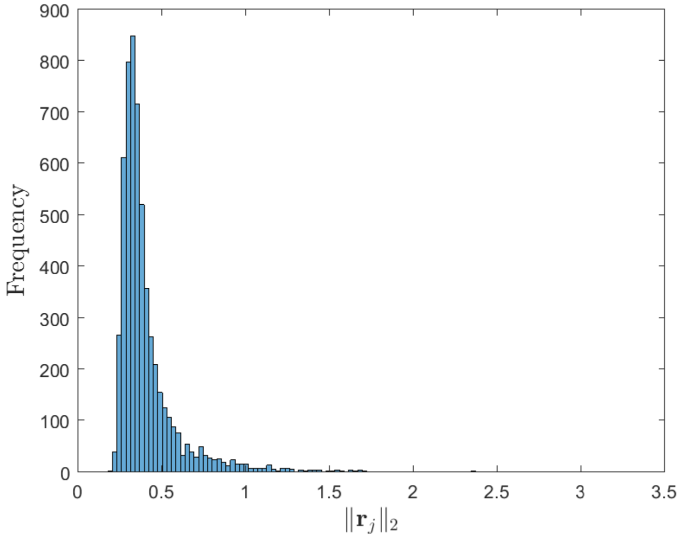


Figure 3.12: Histogram of the PCA-based nominal summary statistics corresponding to the static activities in the HAPT dataset.

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 [6, 62, 91]. 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 is 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, for example, by disconnecting the compromised devices immediately after the detection. As an illustrative attack

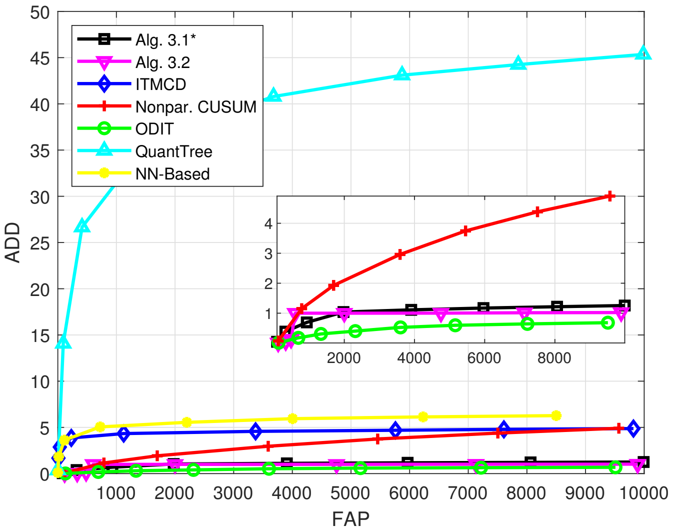


Figure 3.13: ADD vs. FAP for detecting changes in human physical activities.

case, we consider that a spam attack is performed over the network by the BASHLITE botnet [91] and we monitor the thermostat device for anomaly detection. Firstly, based on the PCA method summarized in Alg. 3.2, 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. 3.16. 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, that is, $\gamma \approx 1$. 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. 3.17 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 sampled uniformly from the nominal dataset and after the attack, at each time, we observe a data point sampled uniformly from the "junk" dataset given for the thermostat [91]. The corresponding performance curves are presented in Fig. 3.18 and Fig. 3.19. Similarly to the previous application case, we employ

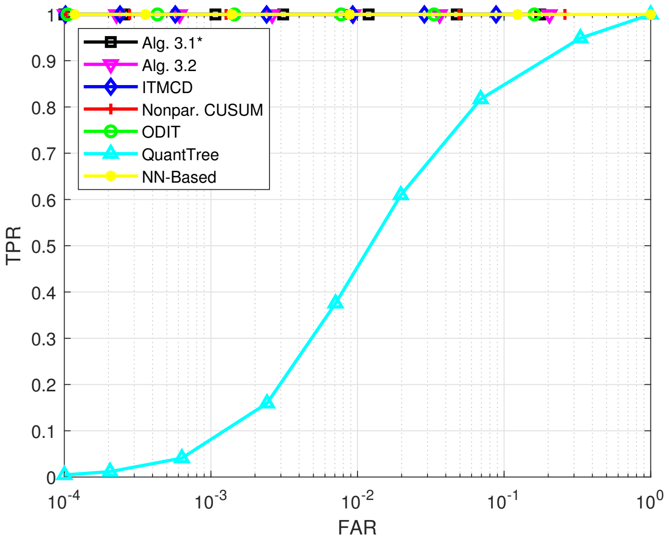


Figure 3.14: ROC curve in detection of human activity change.

Alg. 3.1 using the projected r-dimensional data where we uniformly choose S_1 and S_2 in Alg. 3.1 with sizes $N_1 = 1215$ and $N_2 = 11896$, respectively.

In this experiment, we observe that the nonparametric CUSUM test and the ODIT perform considerably worse than the other detectors. This is because of some significant outliers in the nominal dataset. Specifically, we observe through Fig. 3.17 that the baseline summary statistics mostly lie on an interval of smaller values, that is, the majority of the nominal data points well fit to the principal subspace. However, 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 large detection delays for the nonparametric CUSUM test. Moreover, the significant outliers among the nominal data points (with very large $\|\boldsymbol{r}_t\|_2$) also increase the false alarm rate of the nonparametric CUSUM test. Similarly, the ODIT gives frequent false alarms because of the significant nominal outliers with large NN distances. This is where an advantage of Alg. 3.1 over the ODIT appears: Alg. 3.1 depends on how likely it is to observe a NN distance, specifically, the p-value, rather than the distance itself, which makes it more reliable

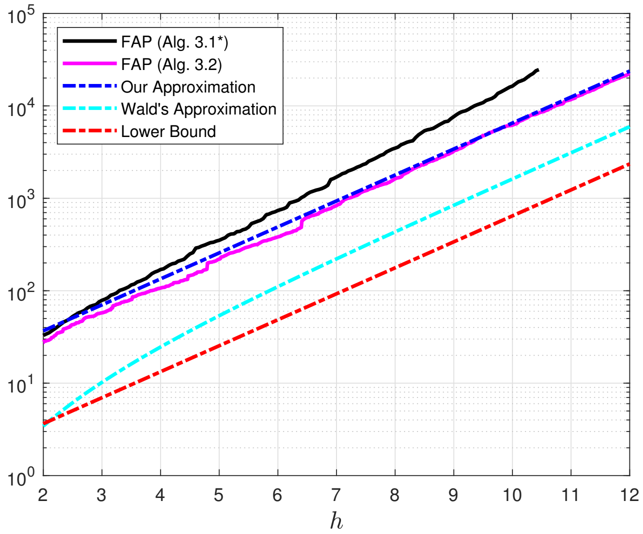


Figure 3.15: FAP of the proposed algorithms for the human activity data, the theoretical approximations, and the theoretical lower bound for various test thresholds.

in the case of significant nominal outliers. Finally, Fig. 3.20 illustrates the FAP of the proposed algorithms along with the asymptotic approximations and the asymptotic lower bound, as the test threshold varies.

3.6 Concluding Remarks

We have proposed nonparametric data-driven online anomaly detection schemes for big data streams. The proposed schemes are reliable, effective, scalable, and hence ideally suited for high-dimensional settings. Moreover, they are widely applicable in a variety of settings 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

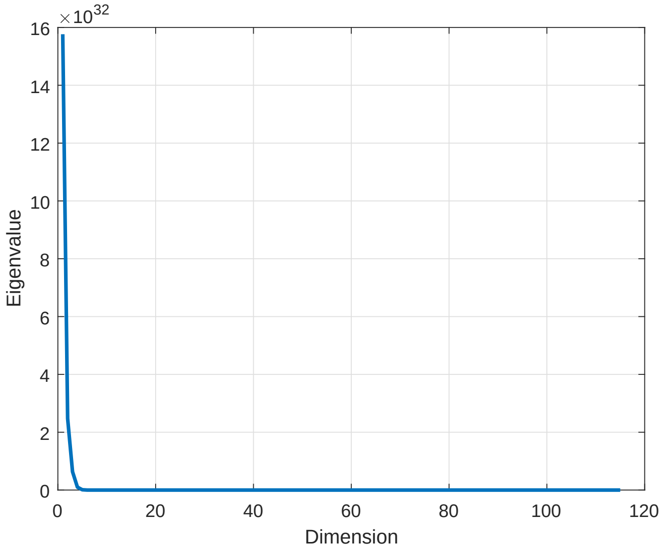


Figure 3.16: Eigenvalues of the sample data covariance matrix for a representative set of nominal data points (thermostat) in the N-BaIoT dataset.

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 asymptotic lower bound and an accurate asymptotic approximation for the FAP 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.

3.7 Appendix to Chapter 3

3.7.1 Proof of Theorem 3.1

Proof. Firstly, we derive the asymptotic distribution of $\hat{s}_t$ (see Eq. (3.14)) in the absence of anomalies, that is, for $t < \tau$. By the Glivenko-Cantelli theorem, the edf of the nominal summary statistics, that is, $\hat{F}^d_{0,N_2}$ given in Eq. (3.12), converges to the

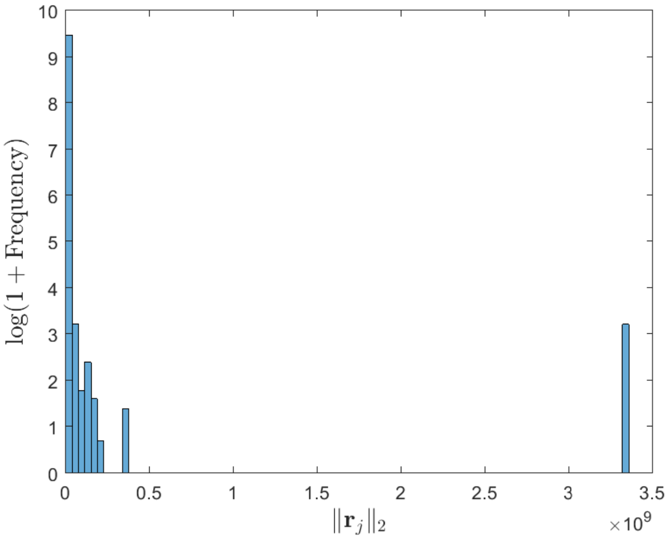


Figure 3.17: Histogram of the PCA-based summary statistics for the nominal data (thermostat) in the N-BaIoT dataset.

cdf F_0^d as $N_2 \to \infty$ [132]. Hence, $\hat{p}_t$ in Eq. (3.13) converges to p_t in Eq. (3.11) and equivalently $\hat{s}_t$ converges to s_t in Eq. (3.3). The proposed CUSUM-like detector in Eq. (3.15) thus converges to the algorithm in Eq. (3.4). It is well known that the cdf of any continuous random variable is uniformly distributed $\mathcal{U}[0, 1]$ [99]. Then, we have

$$p_t = 1 - F_0^d(d_t) \sim \mathcal{U}[0, 1].$$

The cdf of $s_t, t < \tau$, denoted with $F_0^{s_t}$, is then given by

$$\begin{aligned} F_0^{s_t}(y) &= \mathbb{P}(s_t \leq y) = \mathbb{P}\left(\log\left(\frac{\alpha}{p_t}\right) \leq y\right) \\ &= \mathbb{P}\left(p_t \geq \frac{\alpha}{e^y}\right) \\ &= \begin{cases} 1 - \frac{\alpha}{e^y}, & \text{if } y > \log(\alpha) \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

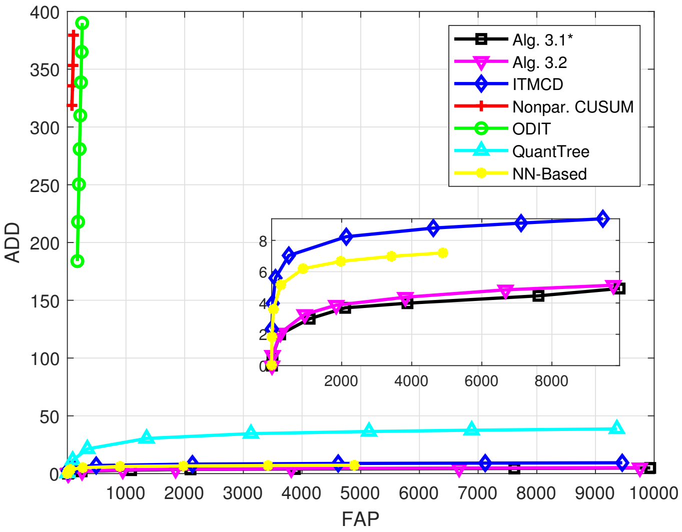


Figure 3.18: ADD vs. FAP in detection of a spam attack launched by a BASHLITE botnet.

Moreover, the pdf of $s_t, t < \tau$, denoted with $f_0^{s_t}$, is given as follows:

$$f_0^{s_t}(y) = \frac{\partial F_0^{s_t}(y)}{\partial y}$$
$$= \begin{cases} \alpha\, e^{-y}, & \text{if } y > \log(\alpha) \\ 0, & \text{otherwise.} \end{cases} \tag{3.25}$$

Then, based on Eq. (3.25), we have $\mathbb{E}[s_t] = 1 + \log(\alpha)$ and $\mathbb{E}[s_t^2] = 1 + (1 + \log(\alpha))^2$.

From Eq. (3.4), we have $g_t = \max\{0, g_{t-1} + s_t\}$, that implies $g_t^2 \leq (g_{t-1} + s_t)^2$. We can then write

$$\begin{aligned} \mathbb{E}[g_t^2 \,|\, g_{t-1}] &\leq \mathbb{E}[(g_{t-1} + s_t)^2 \,|\, g_{t-1}] \\ &= g_{t-1}^2 + 2g_{t-1}\mathbb{E}[s_t] + \mathbb{E}[s_t^2] \\ &= g_{t-1}^2 + 2g_{t-1}(1 + \log(\alpha)) + 1 + (1 + \log(\alpha))^2. \end{aligned} \tag{3.26}$$

Next, we solve the following inequality for $\mathbb{E}[g_t^2 \,|\, g_{t-1}] \leq g_{t-1}^2$:

$$g_{t-1}^2 + 2g_{t-1}(1 + \log(\alpha)) + 1 + (1 + \log(\alpha))^2 \leq g_{t-1}^2, \tag{3.27}$$

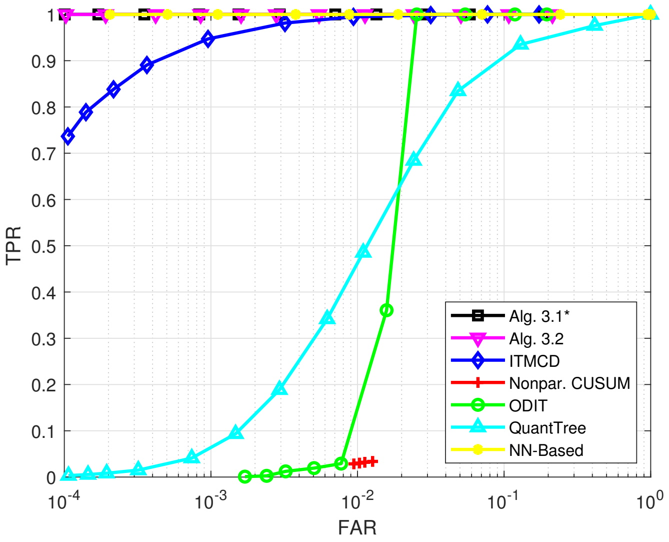


Figure 3.19: ROC curve in detection of a spam attack launched by a BASHLITE botnet.

which is equivalent to

$$-2g_{t-1}(1+\log(\alpha)) \geq 1+(1+\log(\alpha))^2. \tag{3.28}$$

Recalling that $g_{t-1} \geq 0$ and since the RHS of Eq. (3.28) is positive, the solution to Eq. (3.28) is given as follows:

$$\alpha < 1/e \text{ and } g_{t-1} \geq \frac{1+(1+\log(\alpha))^2}{-2(1+\log(\alpha))}. \tag{3.29}$$

Firstly, let $\alpha < 1/e$ and $g_{t-1} \geq f(\alpha)$, where

$$f(\alpha) \triangleq \frac{1+(1+\log(\alpha))^2}{-2(1+\log(\alpha))} > 0.$$

Then, based on Eq. (3.26), Eq. (3.27), and Eq. (3.29), we have

$$\mathbb{E}[g_t^2 \,|\, g_{t-1}] \leq g_{t-1}^2. \tag{3.30}$$

Moreover, since $g_{t-1} \geq f(\alpha) > 0$, we have

$$g_{t-1} = \max\{0, g_{t-2}+s_{t-1}\} = g_{t-2}+s_{t-1}. \tag{3.31}$$

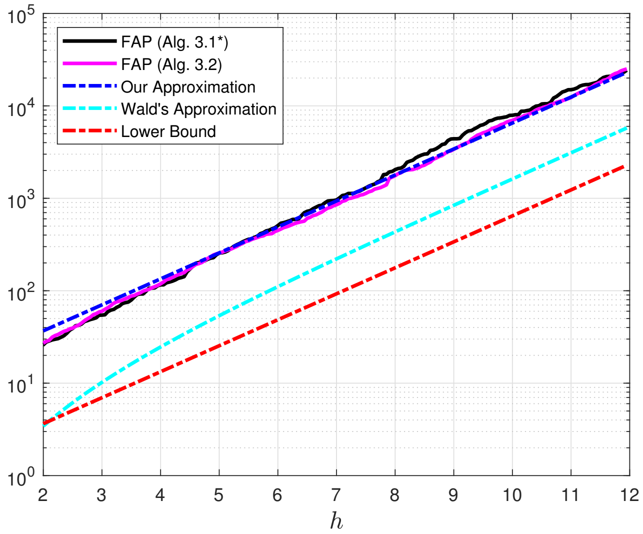


Figure 3.20: FAP of the proposed algorithms for the IoT data, the theoretical approximations, and the theoretical lower bound for various test thresholds.

Here, we can either have $g_{t-2} < f(\alpha)$ or $g_{t-2} \geq f(\alpha)$. In the case where $g_{t-2} < f(\alpha) < \infty$, since $\mathbb{P}(s_{t-1} < \infty) = 1$ (see Eq. (3.25)), we have $\mathbb{P}(g_{t-1} < \infty) = 1$ (see Eq. (3.31)). Then, from Eq. (3.30),

$$\mathbb{P}(\mathbb{E}[g_t^2 \,|\, g_{t-1}] < \infty) = 1. \tag{3.32}$$

Note that

$$\mathbb{E}\left[\mathbb{E}[g_t^2 \,|\, g_{t-1}] \,|\, g_0 = 0\right] = \mathbb{E}[g_t^2 \,|\, g_0 = 0], \tag{3.33}$$

where in the LHS of Eq. (3.33), the inner expectation is with respect to (wrt) $g_t \,|\, g_{t-1}$ and the outer expectation is wrt $g_{t-1} \,|\, g_0 = 0$. Moreover, in the RHS of Eq. (3.33), the expectation is wrt $g_t \,|\, g_0 = 0$. Then, based on Eq. (3.32) and Eq. (3.33), and since the expectation of a finite variable is also finite, we have

$$\mathbb{P}(\mathbb{E}[g_t^2 \,|\, g_0 = 0] < \infty) = 1.$$

Further, in the case where $g_{t-2} \geq f(\alpha)$, similar to Eq. (3.30), we have

$$\mathbb{E}[g_{t-1}^2 \,|\, g_{t-2}] \leq g_{t-2}^2. \tag{3.34}$$

Using nested expectations, Eq. (3.30), and Eq. (3.34), we can write

$$\begin{aligned}\mathbb{E}[g_t^2 \,|\, g_{t-2}] &= \mathbb{E}\left[\mathbb{E}[g_t^2 \,|\, g_{t-1}] \,|\, g_{t-2}\right] \\ &\leq \mathbb{E}[g_{t-1}^2 \,|\, g_{t-2}] \leq g_{t-2}^2.\end{aligned}$$

Here, since $g_{t-2} \geq f(\alpha) > 0$, we have

$$g_{t-2} = \max\{0, g_{t-3} + s_{t-2}\} = g_{t-3} + s_{t-2}.$$

Again, there are two possibilities: we either have $g_{t-3} < f(\alpha)$ or $g_{t-3} \geq f(\alpha)$ and as such the procedure repeats itself backward in time. The conclusion is that if there exists $\omega \leq t$ such that $g_{t-\omega} < f(\alpha)$, then we have $\mathbb{P}(\mathbb{E}[g_t^2 \,|\, g_0 = 0] < \infty) = 1$. Since $g_0 = 0 < f(\alpha)$, there indeed exists at least one ω, which is $\omega = t$, such that $g_{t-\omega} < f(\alpha)$. Then, in case where $\alpha < 1/e$ and $g_{t-1} \geq f(\alpha)$, we have $\mathbb{P}(\mathbb{E}[g_t^2 \,|\, g_0 = 0] < \infty) = 1$.

Next, let $\alpha < 1/e$ and $g_{t-1} < f(\alpha)$. Since $g_t = \max\{0, g_{t-1} + s_t\}$, we either have $g_t = 0$ or $g_t = g_{t-1} + s_t$. If $g_t = 0$, we clearly have $\mathbb{E}[g_t^2 \,|\, g_0 = 0] = 0 < \infty$. On the other hand, if $g_t = g_{t-1} + s_t$, we have

$$\begin{aligned}\mathbb{E}[g_t^2 \,|\, g_{t-1}] &= g_{t-1}^2 + 2g_{t-1}\mathbb{E}[s_t] + \mathbb{E}[s_t^2] \\ &= g_{t-1}^2 + 2g_{t-1}(1 + \log(\alpha)) + 1 + (1 + \log(\alpha))^2 \\ &< g_{t-1}^2 + 1 + (1 + \log(\alpha))^2 \qquad (3.35) \\ &< f(\alpha)^2 + 1 + (1 + \log(\alpha))^2 < \infty, \qquad (3.36)\end{aligned}$$

where Eq. (3.35) follows since $g_{t-1}(1 + \log(\alpha)) < 0$ and Eq. (3.36) follows since $g_{t-1} < f(\alpha)$. Then, using nested expectations and the fact that the expectation of a finite variable is finite, we obtain the following inequality:

$$\mathbb{E}\left[\mathbb{E}[g_t^2 \,|\, g_{t-1}] \,|\, g_0 = 0\right] = \mathbb{E}[g_t^2 \,|\, g_0 = 0] < \infty,$$

that also implies

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

In conclusion, if $\alpha < 1/e$, we have shown above that for both of the complementary conditions, namely $g_{t-1} \geq f(\alpha)$ and $g_{t-1} < f(\alpha)$, we have

$$\mathbb{P}(\mathbb{E}\left[g_t^2 \,|\, g_0 = 0\right] < \infty) = 1. \tag{3.37}$$

The implication is that $\alpha < 1/e$ is a sufficient condition to obtain Eq. (3.37), asymptotically as $N_2 \to \infty$.

□

3.7.2 Proof of Theorem 3.2

Proof. As discussed in Appendix 3.7.1, as $N_2 \to \infty$, $\hat{s}_t$ converges to s_t and hence the proposed CUSUM-like detector in Eq. (3.15) converges to the algorithm in Eq. (3.4). Note that if $\alpha < 1/e$, then $\mathbb{E}[s_t] = 1+\log(\alpha) < 0$. In [5, Sec. 5.2.2.4], for CUSUM-like algorithms such as Eq. (3.4), a lower bound on the FAP is then given as follows:

$$\mathbb{E}_\infty[\Gamma] \geq e^{-w_0 h},$$

where $w_0 < 0$ is the solution to

$$\mathbb{E}[e^{-w_0 s_t}] = 1. \tag{3.38}$$

Defining $\theta \triangleq w_0+1$, we can rewrite Eq. (3.38) based on the pdf of s_t (see Eq. (3.25)) as follows:

$$\begin{aligned}\mathbb{E}[e^{-w_0 s_t}] &= \int_{\log(\alpha)}^{\infty} e^{(1-\theta)y} \alpha e^{-y} dy \\ &= \alpha \int_{\log(\alpha)}^{\infty} e^{-\theta y} dy \\ &= \alpha \, \frac{e^{-\theta \log(\alpha)}}{\theta} = 1, \end{aligned} \tag{3.39}$$

provided that $\theta > 0$. The conditions $w_0 = \theta - 1 < 0$ and $\theta > 0$ together lead to $0 < \theta < 1$. Moreover, we can rewrite Eq. (3.39) as follows:

$$\theta \, e^{\theta \log(\alpha)} = \alpha,$$

and multiplying both sides by $\log(\alpha)$, we have

$$\theta \log(\alpha)\, e^{\theta \log(\alpha)} = \alpha \log(\alpha). \tag{3.40}$$

Now, define $z \triangleq \theta \log(\alpha)$ and $c \triangleq \alpha \log(\alpha)$ so that Eq. (3.40) can be rewritten as

$$z\, e^{z} = c,$$

where $z = W(c) = W(\alpha \log(\alpha))$. Then, we have

$$\theta = \frac{W(\alpha \log(\alpha))}{\log(\alpha)}.$$

Next, we show that there exists a unique solution to Eq. (3.39) by contradiction. Assume that there exists two solutions θ_1 and θ_2 to Eq. (3.39) where $\theta_1 \neq \theta_2$. Further, without loss of generality, assume $\theta_2 < \theta_1$. Then, we have $0 < \theta_2 < \theta_1 < 1$. From Eq. (3.39), we have

$$\alpha\, \frac{e^{-\theta_1 \log(\alpha)}}{\theta_1} = 1,$$

which implies that

$$\frac{\log(\theta_1)}{1-\theta_1} = \log(\alpha), \tag{3.41}$$

and similarly,

$$\frac{\log(\theta_2)}{1-\theta_2} = \log(\alpha). \tag{3.42}$$

Then, based on Eq. (3.41) and Eq. (3.42), we have

$$\frac{\log(\theta_1)}{1-\theta_1} = \frac{\log(\theta_2)}{1-\theta_2}. \tag{3.43}$$

Furthermore, for $0 < \theta < 1$, we have

$$\begin{aligned} \frac{\partial \frac{\log(\theta)}{1-\theta}}{\partial \theta} &= \frac{1/\theta - 1 + \log(\theta)}{(1-\theta)^2} \\ &= \frac{\mu - 1 - \log(\mu)}{(1-1/\mu)^2} > 0, \end{aligned} \tag{3.44}$$

where $\mu \triangleq 1/\theta > 1$. The inequality in Eq. (3.44) follows due to the fact that $\log(\mu) < \mu - 1$ for $\mu > 1$. Then, since the first order derivative of the function

$$\frac{\log(\theta)}{1-\theta}$$

is positive, it is monotonically increasing in the range of $0 < \theta < 1$. Since $0 < \theta_2 < \theta_1 < 1$, we then have

$$\frac{\log(\theta_2)}{1-\theta_2} < \frac{\log(\theta_1)}{1-\theta_1},$$

which contradicts with Eq. (3.43). Hence, there exists a unique solution to Eq. (3.39).

□

Chapter 4

DeepQCD: A Unified Data-Driven Approach to Quickest Change Detection

4.1 Introduction

This chapter aims to unify the QCD framework via a novel easy-to-implement data-driven solution approach, which is applicable to a variety of real-world settings.

4.1.1 Background

Existing QCD approaches can be classified in terms of assumptions on the change-point model, density knowledge, observation model, and internal memory structure, as detailed next.

4.1.1.1 Change-point Model

Based on the change-point model, two common problem formulations exist: the Bayesian and the minimax. The Bayesian formulation assumes that the change-point is random with a known prior distribution where the distribution of the change-point can either be dependent on the observations or independent from the observations [95]. The minimax formulation assumes that the change-point is deterministic but unknown.

In the Bayesian setting, the change-point is commonly assumed to be a geometric random variable with parameter ρ, denoted with $\tau \sim \text{geo}(\rho)$, where

$$\pi_t \triangleq \mathbb{P}(\tau = t) = \rho\,(1-\rho)^{t-1},\ t = 1, 2, 3, \ldots \tag{4.1}$$

The design goal is to minimize the average detection delay (ADD) subject to an upper bound on the probability of false alarm (PFA). Let $\mathbb{P}_t$ and $\mathbb{E}_t$ be the probability and the expectation operators, respectively, given that the change happens at time $\tau = t$. The ADD is then given by

$$\text{ADD}(\Gamma) \triangleq \mathbb{E}[(\Gamma - \tau)^+] = \sum_{t=1}^{\infty} \pi_t\, \mathbb{E}_t[(\Gamma - t)^+],$$

and the PFA is given by

$$\text{PFA}(\Gamma) \triangleq \mathbb{P}(\Gamma < \tau) = \sum_{t=1}^{\infty} \pi_t\, \mathbb{P}_t(\Gamma < t).$$

The Bayesian QCD problem is then written as follows:

$$\min_{\Gamma}\ \text{ADD}(\Gamma)\ \text{ subject to }\ \text{PFA}(\Gamma)\ \leq\ \alpha, \tag{4.2}$$

where $\alpha \in (0, 1)$ is the desired upper bound on the PFA.

In the minimax setting, the design goal is to minimize the worst-case ADD subject to an upper bound on the average false alarm period (FAP). There exist two well-known problem formulations in the minimax setting depending on the definition of the worst-case ADD. Firstly, the Lorden's definition on the worst-case ADD is given by [84]

$$J(\Gamma) \triangleq \sup_{\tau}\ \operatorname*{ess\,sup}_{\mathcal{F}_\tau}\ \mathbb{E}_\tau\big[(\Gamma - \tau)^+\,|\mathcal{F}_\tau\,\big]. \tag{4.3}$$

Notice that $J(\Gamma)$ is computed based on the least favorable change-point and the least favorable history of observations up to the change-point. The Lorden's minimax problem is then written by [84]

$$\inf_{\Gamma} \ J(\Gamma) \ \text{ subject to } \ \mathbb{E}_{\infty}[\Gamma] \geq \beta, \tag{4.4}$$

where $\mathbb{E}_{\infty}[\Gamma]$ denotes the FAP, that is, the average stopping time when no change occurs at all ($\tau = \infty$) and $\beta \geq 1$ is the desired lower bound on the FAP. The false alarm rate (FAR) is defined to be the reciprocal of the FAP:

$$\text{FAR} \triangleq \frac{1}{\mathbb{E}_{\infty}[\Gamma]}.$$

Hence, a lower bound on the FAP corresponds to an upper bound on the FAR. Secondly, Pollak has the following definition on the worst-case ADD [105]:

$$D(\Gamma) \triangleq \sup_{\tau} \mathbb{E}_{\tau}\big[\Gamma - \tau \,|\, \Gamma \geq \tau\big]. \tag{4.5}$$

Notice that $D(\Gamma)$ is less pessimistic than $J(\Gamma)$. The Pollak's problem is written as follows [105]:

$$\inf_{\Gamma} \ D(\Gamma) \ \text{ subject to } \ \mathbb{E}_{\infty}[\Gamma] \geq \beta. \tag{4.6}$$

4.1.1.2 Density Knowledge

In terms of the density knowledge, the following three cases are common: (i) known pre- and post-change pdfs, (ii) known pre-change pdf, unknown post-change pdf, and (iii) unknown pre- and post-change pdfs. The first case has been extensively studied, especially for the independent and identically distributed (i.i.d.) observation setting, and the optimality properties of the well-known QCD algorithms have been established in this case [5, 133]. In particular, many existing QCD algorithms require the computation of the likelihood ratio (LR), which is possible only if the pre- and post-change pdfs are known.

In the second case, several approaches have been developed depending on the knowledge about the post-change pdf. In particular, if the post-change pdf belongs to a certain parametric family of distributions, either the unknown parameters can be estimated and the estimated parameters are incorporated into the change detection process, called the GLR approach [13, 66, 71] or a robust detection approach can be employed where the worst-case parameters are determined and the change detection is performed accordingly [131]. However, if the post-change pdf is completely unknown, the online data stream can be evaluated as to whether it statistically fits to the pre-change pdf [64,67,129]. In the third case, if both the pre- and post-change pdfs belong to a parametric distribution family with some unknown parameters, then as before, either the unknown parameters can be estimated or the worst-case parameters can be determined. Alternatively, a nonparametric (i.e., model-free) approach can be taken if the data distributions are completely unknown [68, 70].

4.1.1.3 Observation Model

There exist many time series models: i.i.d. observations, independent but non-identically distributed observations, autoregressive (AR) model, autoregressive moving average (ARMA) model, state-space model, hidden Markov model (HMM), and so forth [5, 39, 93, 106]. Almost all prior work in the QCD framework studied the i.i.d. observation setting, for both the pre- and post-change cases. This is mainly because it gets more complicated (increasing space and time complexity) to compute the LR as the observations get more correlated over time. Hence, the assumption of temporally independent observations greatly simplifies the QCD algorithms, enabling low-complexity recursive implementations suitable for real-time processing. In the general non-i.i.d. observation setting, asymptotically optimal model-based QCD procedures are extended from the i.i.d. observation setting, where simple recursive implementations are usually no longer possible [71, 128, 133].

4.1.1.4 Internal Memory Structure

In terms of utilizing the observation history in the decision making process, there exist two different QCD approaches: window-limited and fully-sequential. On the one hand, the window-limited procedures restrict the internal memory to a finite number of recent observations and decide on the change accordingly while ignoring the further past observations, see [46] for example. The window-limited procedures have an inherent detection latency because of the window size, which is itself a design parameter. Specifically, a larger window size leads to a larger ADD while a smaller window size ignores more (possibly useful) information from the past. In the extreme case, the past observations are completely ignored and only the latest observation is utilized for the change detection, see the Shewhart test [96] for example. On the other hand, the fully-sequential procedures decide on the change based on the entire observation history via keeping a summary of the observations seen so far in their internal memory. The fully-sequential QCD procedures are usually preferred over the window-limited procedures as they can ideally capture all the relevant information from the observation history. However, to reduce the memory requirements, the existing fully-sequential QCD procedures usually make simplified assumptions such as temporally independent observations.

4.1.2 Motivations

Real-world data streams often have unknown statistical properties [68]. Then, for the change detection, either the underlying probabilistic data models can be learned from data and an existing model-based QCD procedure can be employed accordingly or the change detection rule can be directly learned from data. The former approach requires fitting an existing parametric model to the observed data stream and estimation of unknown model parameters, which might be costly or even intractable for complex, temporally correlated, and high-dimensional data streams. Moreover, it is vulnerable to model mismatch as the real-world data often do not perfectly fit into the existing parametric models and even the lack of parametric models is common in high-dimensional settings [74]. Hence, in practice, usually only an approximate model can be learned, which degrades the performance of the corresponding model-based QCD procedure. The latter direct approach is model-free, robust to data model mismatch, and widely applicable to a variety of practical settings. Hence, this chapter aims to propose an effective model-free solution approach.

In [3], responding to changes in a potentially adversarial environment rapidly yet accurately is addressed as a problem faced by animal brains at every level from neurons to behavior. It is argued that neurons are optimized for tracking abrupt changes in the input spike statistics and moreover, intracellular dynamics of the leaky-integrate-and-fire neurons remarkably resemble the information accumulation and decision process of the Bayesian-optimal Shiryaev algorithm [119] in the QCD framework. It is also argued that the neurons have a tradeoff between speed and accuracy in their decision making process, just like the QCD procedures. These well motivate us to propose a data-driven QCD procedure based on the deep neural networks and the Shiryaev algorithm. Finally, deep learning (DL) does not require explicit data modeling or feature engineering, which enables an entirely data-driven QCD procedure.

4.1.3 Contributions

We propose a novel generic fully-sequential data-driven QCD procedure, called DeepQCD, that learns the change detection rule directly from the observed raw data via deep neural networks. With sufficient amount of training data, DeepQCD can effectively learn the change detection rule for all types of data distributions including complex and high-dimensional distributions, all observation settings including temporally correlated data streams, and all change-point models. Hence, the proposed approach well unifies the QCD framework.

Motivated from the Bayesian-optimal Shiryaev algorithm [119], DeepQCD estimates the posterior probability that change has taken place given all observations seen so far and declares a change as soon as the posterior probability is reliably high. For this purpose, an internal state representation is built from the observation history via recurrent memory cells, which summarizes the useful information relevant to the QCD task. In addition to being intuitive and easy-to-implement, DeepQCD turns out to be very powerful as its performance is similar as or even superior to that of the existing model-based QCD procedures that are designed with the complete statistical knowledge of the observed data stream. Furthermore, in challenging real-world applications such as online anomaly detection over surveillance videos and online cyber-attack detection over Internet of Things (IoT) networks, DeepQCD significantly outperforms existing benchmark algorithms.

4.1.4 Organization

The remainder of the chapter is organized as follows. Sec. 4.2 provides a brief literature review on the related work. Sec. 4.3 presents the generic QCD procedure that motivates the proposed solution approach. Sec. 4.4 describes the proposed DeepQCD algorithm. Sec. 4.5 justifies the proposed data-driven approach through comparisons with the existing model-based QCD algorithms. Sec. 4.6 evaluates the performance of DeepQCD in real-world applications. Finally, Sec. 4.7 concludes the chapter.

4.2 Related Work

In the nonparametric detection, classical procedures are the statistical goodness-of-fit tests such as the Kolmogrov-Smirnov test, Anderson–Darling test, Wilcoxon signed-rank test, and Pearson's chi-squared test [124]. The goodness-of-fit tests are mainly designed for processing univariate data in batch, where their window-limited versions can be employed for online change detection [67, 129]. Furthermore, for multivariate data streams, support vector machine (SVM)-based one-class classification algorithms

[55, 117, 137], nearest neighbor (NN) graph-based algorithms [18, 68, 122, 143, 149], subspace-based algorithms [68,109], and information theoretic algorithms [24,37] have been proposed for online data-driven change and anomaly detection.

Since the fundamental building block of the proposed DeepQCD algorithm is the recurrent memory cells, we provide a brief review on how the recurrent neural networks (RNNs) have been used in the relevant literature. There are three common uses of the RNNs in online change and anomaly detection. The first approach is based on learning a pre-change prediction model via training the RNN only with the pre-change data [10, 63, 86]. For online anomaly detection, the trained RNN predicts the future observations and an anomaly is declared if large prediction errors occur. The second approach builds a classifier on the entire sequence via training the RNN with both pre-change (nominal) and post-change (anomalous) data [144]. In the online detection phase, after observing the entire data sequence, the trained RNN makes a binary decision as to whether the sequence is nominal or anomalous. In the third approach, an RNN-based autoencoder is trained only with the pre-change data in order to learn a nominal reconstruction model [85, 88, 100]. The underlying assumption is that only the nominal data can be reconstructed well. Hence, in the online detection phase, an anomaly is declared if large reconstruction errors occur. In addition to these three common use cases, in a recent study [34], an RNN is designed with a special structure by making use of the wavelet layers, called the pyramid RNN, with the purpose of detecting gradual and multi-scale changes more effectively.

4.3 Generic QCD Procedure

In the QCD framework, the stopping time Γ is determined based on the information derived from the history of observations. Let $\boldsymbol{s}_t$ be the internal state representation (memory) of the decision maker at time t, based on the data stream seen so far, that is, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$, where $\boldsymbol{x}_t \in \mathbb{R}^p$ is the observation made at time t and $p \geq 1$ is the data dimensionality. Ideally, the internal state effectively captures and summarizes all the relevant (to the QCD task) information in the observation history and as new observations are made, the internal state is updated accordingly. Let $\phi(\cdot,\cdot)$ be a function that processes the latest observation $\boldsymbol{x}_t$ and updates the internal state, as given by

$$\boldsymbol{s}_t = \phi(\boldsymbol{x}_t, \boldsymbol{s}_{t-1}). \tag{4.7}$$

Based on the internal state, the decision maker either chooses to stop and declare a change or to continue acquiring further observations. More particularly, the internal state $\boldsymbol{s}_t$ is mapped to a decision statistic d_t via a function $\omega(\cdot)$ as

$$d_t = \omega(\boldsymbol{s}_t), \tag{4.8}$$

and the decision is made based on a threshold-crossing mechanism. In other words, the decision maker declares a change as soon as the decision statistic exceeds a certain threshold $h > 0$. Hence, the stopping time is given by

$$\Gamma = \inf\{t : d_t \geq h\}. \tag{4.9}$$

This procedure (see Fig. 4.1) is common for all QCD algorithms. What make algorithms different from each other are the state update function $\phi(\cdot,\cdot)$ and the decision mapping $\omega(\cdot)$. In particular, the existing QCD algorithms characterize $\phi(\cdot,\cdot)$ and $\omega(\cdot)$ in special forms depending on the assumptions on pre- and post-change pdfs, observation model, change-point model, and internal memory structure. In this chapter, we consider the most general case where no assumptions are imposed on the observed data stream. Instead, we aim to learn both $\phi(\cdot,\cdot)$ and $\omega(\cdot)$ from data and thereby to achieve an entirely data-driven (model-free) QCD procedure. Before introducing the proposed DeepQCD algorithm, we briefly explain below a few existing QCD algorithms and how they fit into the generic QCD procedure (see Fig. 4.1).

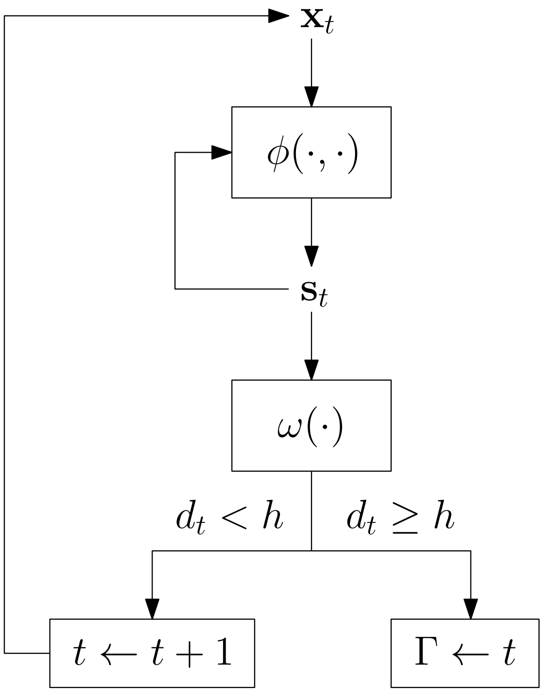


Figure 4.1: Generic QCD procedure.

4.3.1 CUSUM Algorithm

In each QCD algorithm, the internal state has a particular definition. Moreover, to achieve low space and time complexity, usually a recursive state update rule is desired, for which simplifying assumptions are imposed such as temporally independent observations. The cumulative sum (CUSUM) algorithm is the optimal solution to the Lorden's minimax problem given in Eq. (4.4) [94]. In the CUSUM algorithm, the log-likelihood ratio (LLR) is considered as the statistical evidence for change at a time and the accumulation of LLRs over time correspond to the state of the algorithm. In particular, let ℓ_t be the LR at time t, given by

$$\ell_t \triangleq \frac{f_1(\boldsymbol{x}_t)}{f_0(\boldsymbol{x}_t)}. \tag{4.10}$$

Assuming that the observations are i.i.d. over time for both the pre- and post-change cases, the state is updated recursively as follows:

$$s_t = \max\left\{0, s_{t-1} + \log(\ell_t)\right\}, \tag{4.11}$$

where $s_0 = 0$. The decision statistic is identical to the state, that is,

$$d_t = s_t, \tag{4.12}$$

and hence the decision mapping $\omega(\cdot)$ is an identity function. The change is declared at the first time the accumulated statistical evidence is reliably high:

$$\Gamma = \inf\{t : d_t \geq h\}. \tag{4.13}$$

Notice that the CUSUM update given in Eq. (4.11) is a special form of the generic state update in Eq. (4.7), and the CUSUM decision mapping given in Eq. (4.12) is a special form of the generic mapping in Eq. (4.8). Moreover, the CUSUM stopping time given in Eq. (4.13) is identical to Eq. (4.9).

4.3.2 Shiryaev Algorithm

The Shiryaev algorithm is the optimal solution to the Bayesian QCD problem given in Eq. (4.2) [119]. In the Shiryaev algorithm, the state corresponds to the probability that change has taken place given the data stream seen so far, that is,

$$s_t = \mathbb{P}\left(t \geq \tau \,|\, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\right). \tag{4.14}$$

The Bayesian formulation assumes that $\tau \sim \text{geo}(\rho)$, see Eq. (4.1). Moreover, assuming that the observations are independent over time, the following recursive state update rule is employed:

$$s_t = \frac{\tilde{s}_{t-1}\ell_t}{\tilde{s}_{t-1}\ell_t + (1 - \tilde{s}_{t-1})}, \tag{4.15}$$

where

$$\tilde{s}_t \triangleq s_t + \rho(1 - s_t),$$

and $s_0 = 0$. The decision statistic is identical to the state,

$$d_t = s_t, \tag{4.16}$$

and the change is declared as soon as the posterior probability of having change exceeds the test threshold:

$$\Gamma = \inf\{t : d_t \geq h\}, \tag{4.17}$$

where $h \in (0, 1)$. Notice that the Shiryaev state update given in Eq. (4.15) is a special form of Eq. (4.7). Similarly, the decision mapping given in Eq. (4.16) and the stopping time given in Eq. (4.17) both fit to the generic QCD procedure.

4.3.3 Shiryaev-Roberts Procedure

The Shiryaev-Roberts (SR) procedure is an asymptotically optimal solution to the Pollak's minimax problem given in Eq. (4.6) [133]. The SR procedure is obtained from the Shiryaev algorithm in a special regime where $\rho \to 0$, that is, where the change is a rare event and the change-point τ is uniform over time. The state of the SR procedure is updated recursively as follows:

$$s_t = (1 + s_{t-1})\ell_t, \tag{4.18}$$

where $s_0 = 0$. As before, the decision statistic is the same with the state, that is,

$$d_t = s_t, \tag{4.19}$$

and the stopping time is given by

$$\Gamma = \inf\{t : d_t \geq h\}. \tag{4.20}$$

Notice that the SR procedure summarized through Eq. (4.18)-(4.20) also fits to the generic QCD procedure.

4.3.4 Window-Limited Goodness-of-Fit Tests

In the CUSUM, Shiryaev, and the SR procedures, the state is well aligned with the QCD task, the space complexity (memory requirement) is low, and the state can be recursively updated with low time complexity. However, these algorithms require that both the pre- and post-change pdfs are known so that the LR can be computed, and moreover the observations are independent over time. In some cases, such assumptions might not hold and alternative solutions are sought. For example, when the pre-change pdf is known but the post-change pdf is completely unknown, the observed data stream can be analyzed as to whether it fits to or deviates from the pre-change pdf. For this purpose, a window-limited goodness-of-fit test [67, 129] can be used. In this case, let an online sliding-window state be formed with the latest $K \geq 1$ observations, given by

$$\boldsymbol{s}_t = [\boldsymbol{x}_{t-K+1}, \boldsymbol{x}_{t-K+2}, \dots, \boldsymbol{x}_t]. \tag{4.21}$$

At each time t, the goodness-of-fit test analyzes the sliding-window state and makes a decision accordingly.

As an example, the sliding-window chi-squared test described in [67] considers an i.i.d. univariate data stream $x_1, x_2, x_3, \dots$ with a known pre-change pdf f_0 and divides the range of f_0 into L disjoint and mutually exclusive intervals $I_1, I_2, \dots, I_L$ such that $p_1 = \mathbb{P}(x_t \in I_1)$, $p_2 = \mathbb{P}(x_t \in I_2)$, $\dots$, $p_L = \mathbb{P}(x_t \in I_L)$ are the probabilities that the data resides in each interval in the pre-change case, where $\sum_{i=1}^{L} p_i = 1$. Then, for any given state, the expected number of data points in the predetermined intervals are Kp_1, Kp_2, $\dots$, Kp_L. At time t, for the state $\boldsymbol{s}_t$, let the number of data points residing in each interval be counted as $N_{1,t}$, $N_{2,t}$, $\dots$, $N_{L,t}$, where $\sum_{i=1}^{L} N_{i,t} = K$. The decision statistic is then computed as

$$d_t = \sum_{i=1}^{L} \frac{(N_{i,t} - Kp_i)^2}{Kp_i}, \tag{4.22}$$

where d_t is asymptotically (as $K \to \infty$) a chi-squared random variable with $L-1$ degrees of freedom in the pre-change case, that is, for $t < \tau$. However, if the observed data stream deviates from the pre-change pdf, the decision statistic given in Eq. (4.22) is expected to increase. The stopping time of this procedure is thus given by

$$\Gamma = \inf \{t : d_t \geq h\}.$$

Notice that the window-limited online goodness-of-fit tests can also be expressed within the generic QCD procedure (see Fig. 4.1) via a simple sliding-window state as given in Eq. (4.21) and batch processing the observations in the current state to compute a decision statistic, see the chi-squared decision statistic in Eq. (4.22) for example. For the sliding-window state, at each time t, the state update is performed by simply removing the oldest observation $\boldsymbol{x}_{t-K}$ and including the newest observation $\boldsymbol{x}_t$.

4.3.5 Shewhart Test

Finally, in terms of the internal memory, an extreme case is deciding on the change based only on the latest observation while ignoring the past observations at all. In the corresponding setting, the classical QCD procedure is the Shewhart test. In case the pre- and post-change pdfs are known, the Shewhart test computes the LLR and compares it with a certain threshold. Here, the internal state corresponds to the latest observation, see Eq. (4.21) when $K = 1$, that is,

$$\boldsymbol{s}_t = \boldsymbol{x}_t, \tag{4.23}$$

the decision statistic is the LLR, given by

$$d_t = \log\left(\frac{f_1(\boldsymbol{s}_t)}{f_0(\boldsymbol{s}_t)}\right), \tag{4.24}$$

and the change is declared whenever the LLR exceeds a certain threshold:

$$\Gamma = \inf \{t : d_t \geq h\}. \tag{4.25}$$

The Shewhart test described through Eq. (4.23)-(4.25) also fits into the generic QCD procedure. Next, we explain the proposed model-free DeepQCD algorithm.

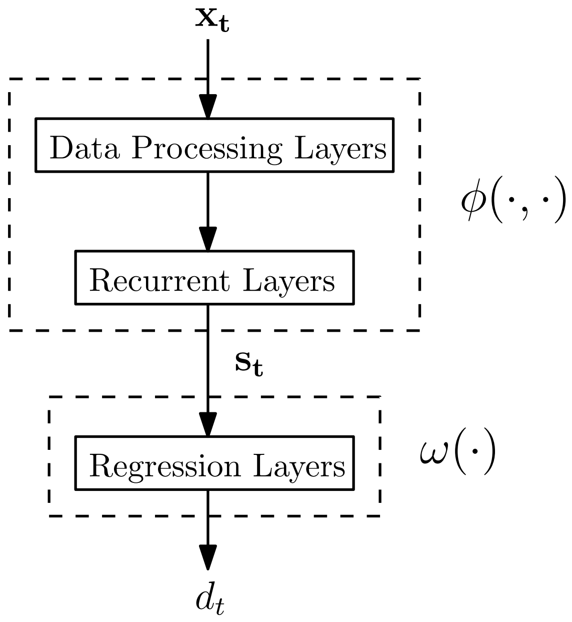


Figure 4.2: Proposed generic neural network architecture for the data-driven QCD (DeepQCD). At each time t, the new observation $\boldsymbol{x}_t$ is processed and the extracted features are used to update the internal state $\boldsymbol{s}_t$. The internal state is then mapped to the decision statistic d_t.

4.4 Data-Driven QCD via Deep Learning

Suppose that a dataset is available including both pre- and post-change samples. To obtain an effective data-driven QCD procedure, we propose to train a deep neural network consisting of three building blocks: initial data processing layers, recurrent layers, and regression layers (see Fig. 4.2). The purpose of this architecture is to learn both the state update and the decision mapping from data without imposing model assumptions on the observed data stream. As illustrated in Fig. 4.2, the data processing and recurrent layers together correspond to the state update function $\phi(\cdot,\cdot)$ and the regression layers correspond to the decision mapping $\omega(\cdot)$.

4.4.1 Data Processing Layers

The first few layers of the network perform an initial processing on the new observation $\boldsymbol{x}_t$ to extract useful features from it. The initial data processing can be especially useful for high-dimensional data to filter out the noise and keep only the relevant informative (i.e., task-oriented) part of the data. Moreover, to extract the most relevant and useful information, the data processing layers should be chosen specific to the observed data type. For instance, for a video data stream, convolutional layers can be used. Further, depending on the observed data type, the data processing layers can simply consist of dense layers or even the initial data processing might not be necessary at all, that is, the raw data can be directly input to the recurrent layers.

4.4.2 Recurrent Layers

The recurrent layers serve as the memory of the network, which summarizes the data stream seen so far, where at each time t, the internal state is updated based on the new observation $\boldsymbol{x}_t$. The recurrent layers are the most fundamental building block of our solution structure. This is because how effective the recurrent layers are in capturing the relevant information from the observation history critically affects the performance of the QCD procedure. In practice, advanced recurrent memory cells such as the long short-term memory (LSTM) [49] and gated recurrent unit (GRU) [20] can be used in the recurrent layers.

The RNNs encode the observation history into an internal state via sequential data processing, just like the QCD procedures build an internal state from the sequentially acquired observations. Although, in principle, the latest RNN state captures all the past observations, remembering arbitrarily long histories can be practically difficult [29]. Nevertheless, from the QCD perspective, the recent history is particularly more important since the observations after the change (i.e., $\{\boldsymbol{x}_t\}_{t\geq\tau}$) are the most informative about the change event. Furthermore, the advanced memory cells such as the LSTM and GRU can make a selective use of the observation history by suppressing the noise and keeping the relevant information. Therefore, we expect that RNNs can well serve as the memory of our data-driven QCD procedure.

Notice that unlike the sliding-window state where the memory is hard-coded with the latest K observations and K is a design parameter, the RNNs automatically discover a way to build the internal state from the observation history. In fact, a key advantage of DL is that the solution depends much less on human input and interpretation but rather the data speaks itself. In other words, the neural network learns to perform the desired task by directly interacting with the data. Note, however, that for an effective learning, the amount of data should be sufficiently large.

4.4.3 Regression Layers

In the proposed solution structure, finally, the recurrent layers are followed by a few dense layers that map the internal state $\boldsymbol{s}_t$ to the decision statistic d_t. Motivated by the Bayesian-optimal Shiryaev algorithm, our decision statistic at time t is the posterior probability that change has occurred given all the observations seen until time t, that is,

$$d_t = \mathbb{P}\left(t \geq \tau \,|\, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\right). \tag{4.26}$$

It appears that for this decision statistic, setting the ground truth labels in the training phase is particularly convenient. That is, in the pre-change case, the ground truth label $\hat{d}_t$ is given by, see Eq. (4.26),

$$\hat{d}_t = 0,\ t < \tau, \tag{4.27}$$

and in the post-change case, the ground truth label is given by, see Eq. (4.26),

$$\hat{d}_t = 1,\ t \geq \tau. \tag{4.28}$$

The dense layers essentially perform a regression task mapping the internal state into the posterior probability of having change.

4.4.4 DeepQCD

The proposed DeepQCD algorithm consists of an offline training and an online detection phases. In the training phase, the neural network is trained end-to-end with the objective of minimizing the difference between the estimated decision statistics d_t and the ground truth labels $\hat{d}_t$. As the loss function, either the mean squared error, given by

$$\mathbb{E}_{\mathcal{D}}\left[(\hat{d}_t - d_t)^2\right],$$

or the binary cross entropy loss, given by

$$-\mathbb{E}_{\mathcal{D}}\left[\hat{d}_t \log(d_t) + (1-\hat{d}_t)\log(1-d_t)\right],$$

can be used where $\mathcal{D}$ denotes the training dataset. During the training phase, the neural network weights are adjusted towards minimizing the loss function.

After the training phase is over, the trained neural network is used for the QCD task. Specifically, in the online phase, at each time t, the new observation $\boldsymbol{x}_t$ is given as an input to the network and the network outputs the decision statistic d_t (see Fig. 4.2). A change is declared at the first time the decision statistic exceeds a predetermined threshold $h \in (0, 1)$:

$$\Gamma = \inf\left\{t : d_t \geq h\right\}.$$

The test threshold h controls the speed-accuracy tradeoff in the decision making process. Specifically, a larger threshold reduces the frequency of false alarms but also increases the ADD.

4.5 Justification of the Data-Driven QCD Procedure

In this section, we consider a few cases studied in the QCD literature and in each case, we evaluate the existing model-based QCD algorithms designed with the full

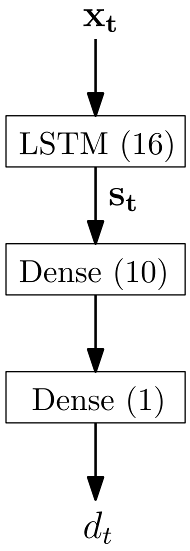


Figure 4.3: Neural network architecture for the experiments in Sec. 4.5.

statistical knowledge of the observed data stream. Further, we evaluate the proposed DeepQCD algorithm implemented with no prior knowledge about the observed data. The purpose here is to justify and exhibit the power of DeepQCD as compared to the (optimal) model-based approaches.

4.5.1 I.I.D. Observation Process

Consider a data stream described through Eq. (1.1) and let the observations be independent over time for both the pre- and post-change cases. Firstly, in the Bayesian setting where $\tau \sim \text{geo}(\rho)$, we compare DeepQCD with the optimal Shiryaev algorithm described through Eq. (4.15)-(4.17).

As an example, we choose $\rho = 0.001$, the data dimensionality as $p = 7$, $f_0 \sim \mathcal{N}(\mathbf{0}_p, \boldsymbol{I}_p)$, and $f_1 \sim \mathcal{N}(\mathbf{1}_p, \boldsymbol{I}_p)$, where $\mathbf{0}_p$ denotes a p-dimensional vector consisting of all zeros, $\mathbf{1}_p$ denotes a p-dimensional vector consisting of all ones, and $\boldsymbol{I}_p$ is a $p \times p$ identity matrix. For the given setup, we generate the training data as consisting of 3200 independent data streams. The length of each data stream is chosen to be 2000 where the observations are from the pre-change pdf up to time τ and from the post-change pdf afterwards. In other words, the training dataset is given by

$$\mathcal{D} = \{(\boldsymbol{x}_t^i, \hat{d}_t^i),\ t = 1, 2, \ldots, 2000,\ i = 1, 2, \ldots, 3200\},$$

and for the ith data stream in the training set (i.e., $\{\boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \ldots, \boldsymbol{x}_{2000}^i\}$), we have

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

where $\tau \sim \text{geo}(0.001)$. Note that if the random change-point τ is greater than the specified stream length 2000, then the corresponding data stream contains only the pre-change observations. Moreover, the ground truth labels are set as follows:

$$\hat{d}_t^i = \begin{cases} 0, & \text{if } t < \tau, \\ 1, & \text{if } t \geq \tau. \end{cases}$$

For DeepQCD, we choose a neural network architecture as shown in Fig. 4.3, where the network is composed of a recurrent layer and two regression layers. The recurrent layer contains an LSTM cell with 16 units (i.e., $\boldsymbol{s}_t \in \mathbb{R}^{16}$) whereas the regression layers are composed of two dense layers with 10 neurons and 1 neuron, respectively. In the first dense layer, we use the rectified linear unit (ReLu) activation function while we use the sigmoid activation in the second dense layer. Moreover, we use the binary cross entropy loss as the loss function and the adaptive moment (Adam) stochastic gradient descent method [59] with a learning rate of 0.001 as the optimizer during the training phase.

After the training phase, we evaluate the performance of DeepQCD and the Shiryaev algorithm in the Bayesian setting, where $\tau \sim \text{geo}(0.001)$. Fig. 4.4 shows the tradeoff between the ADD and PFA as the test threshold h varies for both algorithms. We observe through Fig. 4.4 that DeepQCD performs nearly the same as the optimal Shiryaev algorithm. Recall that the Shiryaev algorithm is designed with the complete knowledge of the observed data stream, namely the pre- and post-change pdfs f_0 and f_1, the geometric change-point model with the known parameter

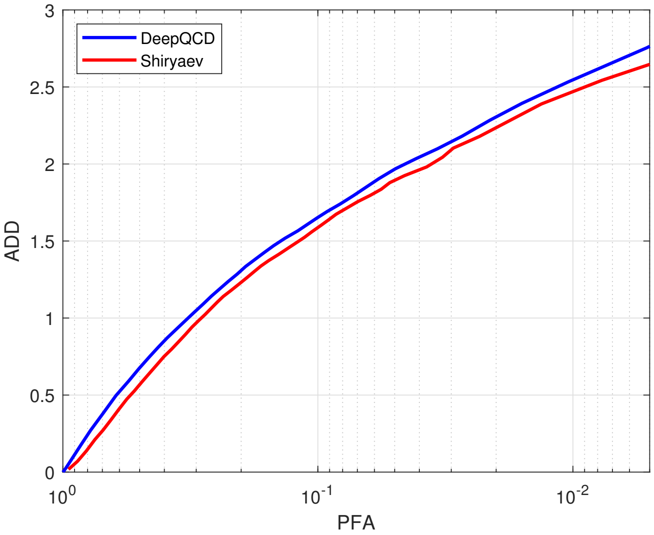


Figure 4.4: ADD vs. PFA curves for DeepQCD and the Shiryaev algorithm in the Bayesian setting where $\tau \sim \text{geo}(0.001)$.

ρ, and the i.i.d. observation model. Hence, Fig. 4.4 demonstrates the power of the proposed data-driven DeepQCD algorithm as it performs near-optimally without any prior information regarding the observed data stream.

Secondly, in the minimax setting, the CUSUM algorithm described through Eq. (4.11)-(4.13) is the optimal solution to the Lorden's problem [94] and it is asymptotically optimal for the Pollak's problem [133]. Furthermore, the SR procedure described through Eq. (4.18)-(4.20) is asymptotically optimal for the Pollak's problem [133]. Note that as the CUSUM and the SR procedures have their optimality properties in the minimax setting based on the pessimistic measures of the ADD, see Eq. (4.3) and Eq. (4.5), it is, in principle, possible to obtain better QCD procedures than the CUSUM and the SR procedures.

We evaluate DeepQCD, the CUSUM algorithm, and the SR procedure using the same experimental setup as in the Bayesian setting except for the change-point model. Firstly, assuming $\tau = \infty$, we compute the FAP of the algorithms for various test thresholds. Next, assuming $\tau = 1$, we compute the ADD of the algorithms for the

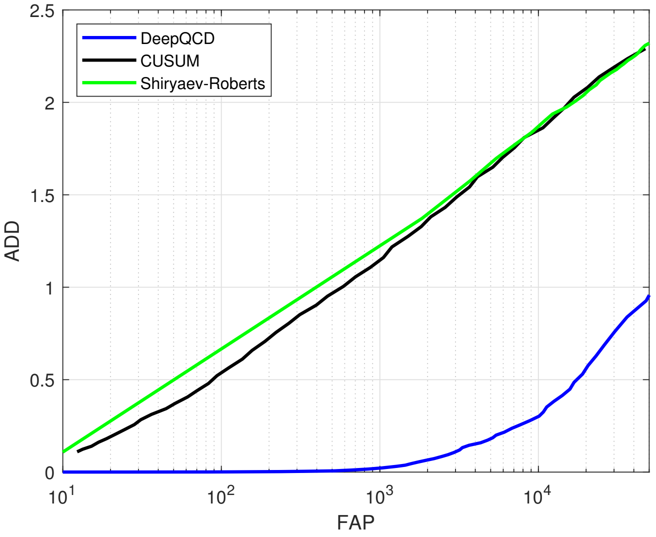


Figure 4.5: ADD vs. FAP curves for DeepQCD, the CUSUM algorithm, and the SR procedure.

same set of test thresholds. Fig. 4.5 illustrates that DeepQCD significantly outperforms the model-based CUSUM and the SR procedures. Note that in this experiment, we use the same trained network in the Bayesian setting and observe that even if the online data stream mismatches the training data in terms of the change-point model, the trained network still performs the QCD task very well. In fact, in this experiment, while generating the training data, we prefer to use random change-points to make sure that the neural network learns the transition from the pre-change to the post-change case over different change-points.

4.5.2 AR Observation Process

Consider a first-order AR observation process driven by temporally independent Gaussian noise:

$$x_t = \begin{cases} \mu_0 + \lambda_0 x_{t-1} + \epsilon_t, & \text{if } t < \tau, \\ \mu_1 + \lambda_1 x_{t-1} + \epsilon_t, & \text{if } t \geq \tau, \end{cases} \tag{4.29}$$

where $\epsilon_t \sim \mathcal{N}(0,1)$ and $|\lambda_0|, |\lambda_1| \leq 1$. Our goal is to timely detect possible changes in the drift of the AR(1) observation process (i.e., from μ_0 to μ_1) and its correlation coefficient (i.e., from λ_0 to λ_1). Notice that the observations are correlated over time except for the special case where $\lambda_0 = \lambda_1 = 0$.

In [106], the CUSUM and the SR procedures are adapted to the AR(1) observation process given in Eq. (4.29). Specifically, the LR is derived as [106]

$$\ell_t = e^{[x_t - 0.5(x_{t-1}(\lambda_0+\lambda_1)+\mu_0+\mu_1)]\,[x_{t-1}(\lambda_1-\lambda_0)+\mu_1-\mu_0]}, \tag{4.30}$$

and then the CUSUM and the SR procedures are extended from the i.i.d. observation setting to the AR(1) observation setting using the LR expression given in Eq. (4.30). It is shown that the modified versions of the CUSUM and the SR procedures are asymptotically minimax-optimal [106]. Hence, we compare DeepQCD with the modified CUSUM and the modified SR procedures. Further, we extend the Shiryaev algorithm to the AR(1) observation setting using the LR expression given in Eq. (4.30), and compare DeepQCD with the modified Shiryaev algorithm in the Bayesian setting.

As an example, let $\mu_0 = 0$, $\lambda_0 = -0.3$, $\mu_1 = 1$, and $\lambda_1 = 0.2$. Then, for the AR(1) observation process given in Eq. (4.29), we generate the training data consisting of 3200 independent data streams where each stream length is 2000 as in the previous subsection. As before, to obtain different change-points while generating the training data, we use a geometric change-point with the parameter $\rho = 0.001$. Finally, we use the same neural network structure (see Fig. 4.3) and the same training specifications given in the previous subsection.

After the training phase, firstly in the Bayesian setting where $\tau \sim \text{geo}(0.001)$, we evaluate the performance of DeepQCD and the modified Shiryaev algorithm (see Fig. 4.6). Next, we evaluate DeepQCD, the modified CUSUM and the modified SR procedures where $\tau = \infty$ for the FAP and $\tau = 1$ for the ADD calculations (see Fig. 4.7). The figures illustrate that DeepQCD performs nearly same or even better than the model-based benchmark algorithms, demonstrating the power of the

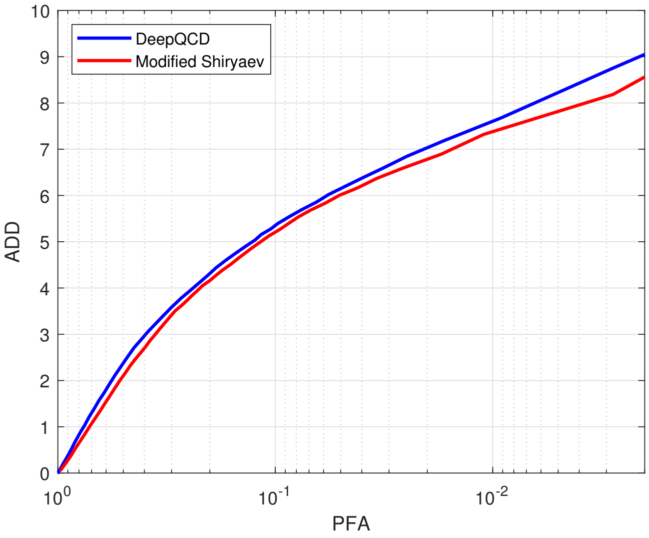


Figure 4.6: ADD vs. PFA curves for DeepQCD and the modified Shiryaev algorithm for detecting changes in an AR(1) observation process in the Bayesian setting where $\tau \sim \text{geo}(0.001)$.

proposed data-driven procedure in a temporally correlated AR(1) observation setting.

4.5.3 Transient Change Detection

In the QCD framework, the change is usually assumed to be persistent, that is, the post-change period is infinitely long after change happens, see Eq. (1.1) for example. However, in some applications such as radar, sonar, intrusion detection, and industrial monitoring [46], the change might be transient, that is, it occurs for a short period of time and then disappears. In such cases, we can write

$$\boldsymbol{x}_t \sim \begin{cases} f_0, & \text{if } t < \tau_1, \\ f_1 \neq f_0, & \text{if } \tau_1 \leq t < \tau_2, \\ f_0, & \text{if } t \geq \tau_2, \end{cases}$$

where the change lasts for the time period $[\tau_1, \tau_2)$. Assume that the observations are independent over time given τ_1 and τ_2.

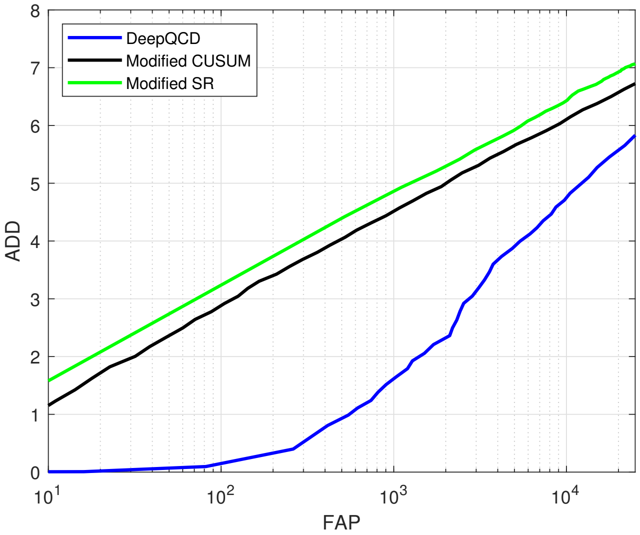


Figure 4.7: ADD vs. FAP curves for DeepQCD, the modified CUSUM algorithm, and the modified SR procedure for detecting changes in an AR(1) observation process.

In the transient QCD framework, a common objective is to detect the change before it disappears. Let Γ be the stopping time at which a change (from f_0 to f_1) is declared. Then,

$$\{\tau_1 \leq \Gamma < \tau_2\}, \tag{4.31}$$

$$\{\Gamma \geq \tau_2\}, \tag{4.32}$$

and

$$\{\Gamma < \tau_1\} \tag{4.33}$$

are considered as the detection, missed detection, and false alarm events, respectively. The goal is to maximize the probability of detection (PD) or equivalently to minimize the probability of missed detection (PMD) subject to an upper bound on the PFA. Let $\mathbb{P}_{\tau_1,\tau_2}$ be the probability measure when the change happens at time τ_1 and disappears at time τ_2. The transient QCD problem can then be written as follows:

$$\max_{\Gamma} \ \mathbb{P}_{\tau_1,\tau_2}(\tau_1 \leq \Gamma < \tau_2) \ \text{ subject to } \ \mathbb{P}_{\tau_1}(\Gamma < \tau_1) \leq \alpha, \tag{4.34}$$

where $\alpha \in (0,1)$ is the desired upper bound on the PFA. Notice that the detection, missed detection, and the false alarm events are disjoint and mutually exclusive, and hence

$$\mathrm{PD} + \mathrm{PMD} + \mathrm{PFA} = 1.$$

For the transient QCD problem given in Eq. (4.34), no optimal solutions exist except for the extreme case where only one observation is seen from f_1, that is, $\tau_2 = \tau_1 + 1$, for which the Shewhart algorithm is optimal in maximizing the worst-case PD [96]. Moreover, the well-known QCD algorithms such as the CUSUM and the SR procedures loose their optimality properties in the transient setup as they are designed for the persistent changes.

In [46], with the goal of minimizing the worst-case PMD, a suboptimal window-limited CUSUM algorithm is presented, where a certain transient change period $K \geq 1$ is assumed, that is, $\tau_2 = \tau_1 + K$, and the QCD procedure is designed accordingly. Specifically, the latest K observations are used to compute the decision statistic of the algorithm. Then, in terms of the generic QCD procedure (see Fig. 4.1), we can express the window-limited CUSUM algorithm as follows:

$$\boldsymbol{s}_t = [\boldsymbol{x}_{t-K+1}, \boldsymbol{x}_{t-K+2}, \dots, \boldsymbol{x}_t],$$

$$d_t = \max_{t-K+1 \leq k \leq t} \sum_{i=k}^{t} \log\left(\frac{f_1(\boldsymbol{x}_i)}{f_0(\boldsymbol{x}_i)}\right),$$

$$\Gamma = \inf\{t : d_t \geq h\}.$$

Notice that for transient changes, two consecutive transitions happen: the first from f_0 to f_1 at time τ_1 and the second from f_1 back to f_0 at time τ_2. Hence, we aim to train our network for quick detection of both the short signal of change and also moving back to the pre-change condition. Since we prefer to use random change-points in the training phase, as an example, we consider both change-points as geometric random variables while generating the training data, where $\tau_1 \sim \mathrm{geo}(\rho_1)$ and $\tau_2 - \tau_1 \sim \mathrm{geo}(\rho_2)$ with the parameters $\rho_1 = 0.001$ and $\rho_2 = 0.002$. Further, we assume $f_0 = \mathcal{N}(0,1)$ and $f_1 = \mathcal{N}(1,1)$. Then, we generate the training data accordingly consisting of 3200 independent data streams, where each stream length is 3000. In other words, the training dataset is given by

$$\mathcal{D} = \{(\boldsymbol{x}_t^i, \hat{d}_t^i),\ t = 1, 2, \ldots, 3000,\ i = 1, 2, \ldots, 3200\},$$

and for the ith data stream, we have

$$\boldsymbol{x}_t^i \sim \begin{cases} f_0, & \text{if } t < \tau_1, \\ f_1, & \text{if } \tau_1 \leq t < \tau_2, \\ f_0, & \text{if } t \geq \tau_2, \end{cases}$$

where $\tau_1 \sim \text{geo}(0.001)$ and $\tau_2 - \tau_1 \sim \text{geo}(0.002)$. Moreover, the ground truth labels are given by

$$\hat{d}_t^i = \begin{cases} 0, & \text{if } t < \tau_1, \\ 1, & \text{if } \tau_1 \leq t < \tau_2, \\ 0, & \text{if } t \geq \tau_2. \end{cases}$$

Furthermore, we use the same neural network architecture and the same training specifications as in the previous experiments.

After the training phase, assuming that $\tau_1 = 1000$, $K = 25$, and $\tau_2 = \tau_1 + K$, we evaluate the performance of the proposed DeepQCD algorithm and the window-limited CUSUM algorithm. Fig. 4.8 shows the PD versus the PFA for both algorithms in the given setup. Fig. 4.8 illustrates that although the window-based CUSUM algorithm assumes the complete knowledge of the observed data stream including the pre- and post-change pdfs, i.i.d. observation model, and the transient change period K, it still performs worse than DeepQCD.

Finally, note that although the transient QCD framework mainly focuses on the detection of the transient signal, that is, the transition from f_0 to f_1, the detection of moving back to pre-change conditions, that is, the transition from f_1 to f_0, is also important to infer the underlying statistical properties of the observed data stream

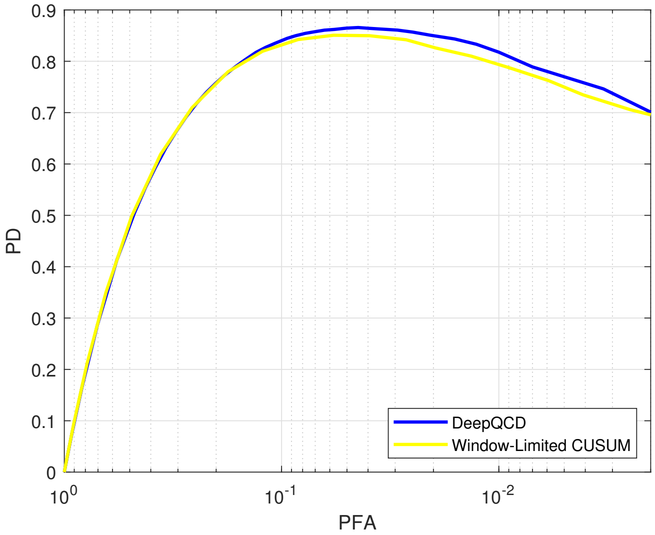


Figure 4.8: PD vs. PFA curves for DeepQCD and the window-limited CUSUM algorithm for detecting transient changes where $\tau_1 = 1000$ and $\tau_2 = 1025$.

at any given time. We argue that although the existing algorithms are not designed accordingly, the proposed data-driven procedure is naturally adapted to detect both transitions as it is trained with data streams containing both transitions. This can be listed as another advantage of DeepQCD over the existing alternatives in the transient QCD framework.

For an illustration of this additional feature, we compare DeepQCD with the window-limited CUSUM algorithm in detecting changes from f_1 back to f_0. For this purpose, we consider a short preheating period in which the internal states of both algorithms are initialized with the observations from f_1. As an example, let the preheating period be 50 and the performance evaluation begins after this period. Since in this experiment, we are interested in detecting the change from f_1 back to f_0, we modify the stopping times of both algorithms as follows:

$$\Gamma = \inf \{t : d_t \leq h\},$$

that is, the change from f_1 to f_0 is declared as soon as the decision statistic is

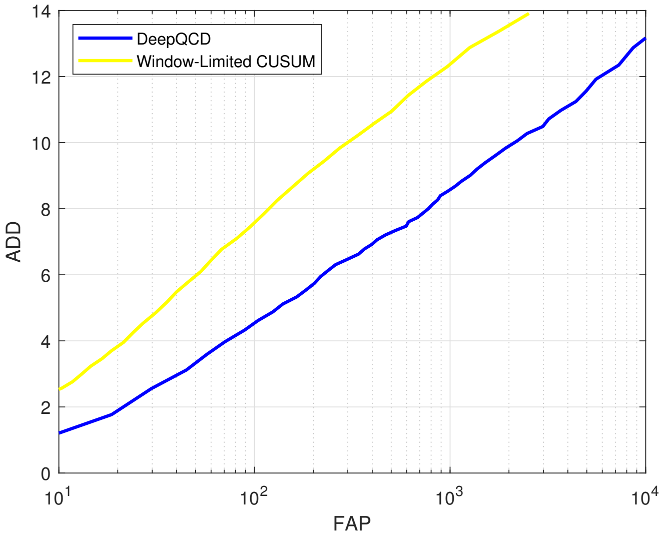
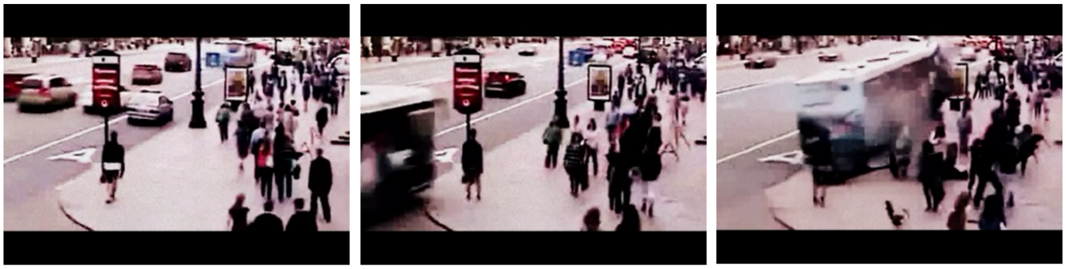


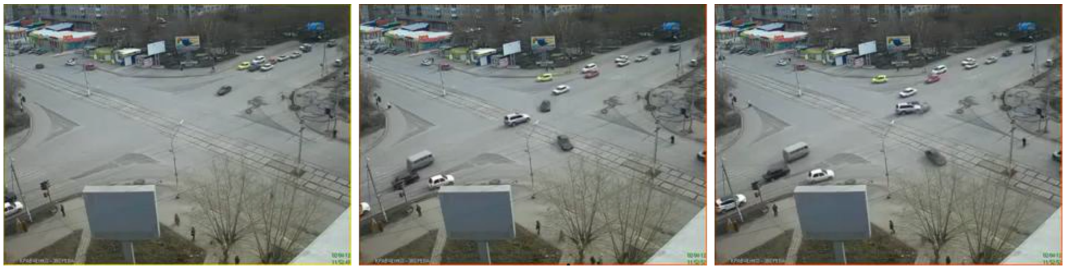
Figure 4.9: ADD vs. FAP curves in detecting changes from f_1 back to f_0 for DeepQCD trained under the transient change setup and the window-limited CUSUM algorithm.

sufficiently small. For the performance evaluation, firstly assuming $x_t \sim f_1, \forall t \geq 1$, that is, $\tau_1 = 1$ and $\tau_2 = \infty$, we compute the FAP of both algorithms. Here, the false alarm event is defined as declaring a change from f_1 to f_0 although the true pdf is still f_1. Next, assuming $x_t \sim f_0, \forall t \geq 1$, that is, $\tau_1 = \tau_2 = 1$, we compute the ADD of both algorithms for detecting the change from f_1 to f_0 (recall that in the preheating period, the observations are from f_1). Fig. 4.9 shows the corresponding performance curve, illustrating the superior performance of DeepQCD. Hence, we demonstrate that DeepQCD, trained under the transient change dynamics, effectively learns to detect both the transition from f_0 to f_1 and the one from f_1 back to f_0.



$t = 00{:}03$ $t = \tau = 00{:}07$ $t = 00{:}09$

Figure 4.10: An example car accident. A bus drives on the sidewalk.



$t = 00{:}02$ $t = \tau = 00{:}04$ $t = 00{:}05$

Figure 4.11: An example car accident. The white car hits the black car.

4.6 Experiments with Real-World Data

4.6.1 Online Anomaly Detection over Surveillance Videos

In [125], a real-world large-scale video surveillance dataset is provided, including normal activities and 13 different anomalies such as road accidents, burglary, and fighting, where each video frame can be resized to 240×320 pixels, that is, $\boldsymbol{x}_t \in \mathbb{R}^{76800}$. In this dataset, ground truth labels are provided at the video-level, that is, the videos are labeled as either nominal or anomalous, and for anomalous videos, it is not specified exactly where the anomaly happens. However, the training phase of our algorithm requires the frame-level ground truth, see Eq. (4.27) and Eq. (4.28). Moreover, in the testing phase, the ADD calculations require the knowledge of the change-point. Hence, for a subset of the available anomalous videos, we manually label the video frames and perform the training and the performance evaluation accordingly. In particular, we perform the manual labeling for the road accident videos because marking the anomalies is more clear for them and the majority of the available anomalous videos are the road accidents. For example, see Fig. 4.10 and Fig. 4.11. Moreover, as the nominal videos, we use the vehicle videos without any anomaly.

For DeepQCD, we use the neural network architecture shown in Fig. 4.12. For the initial data processing, we use the Two-Stream Inflated 3D ConvNet (I3D) [14] pre-trained over the ImageNet [27]. The I3D extracts 1024 features from the video frames. The extracted features are first normalized (subtracting the mean and dividing by the standard deviation of the nominal features) and then fed into the recurrent layers of the network. Finally, three dense layers are used to estimate the decision statistic. We use the GRU cells in the recurrent layers, where we use the dropout [123] with the rate of 0.5. Moreover, before each dense layer, we perform the batch normalization and before the last dense layer, we use the dropout with the rate of 0.5. For the first two dense layers, we use the ReLu activation and for the last dense layer, we use the sigmoid activation.

We use 320 nominal videos and 141 anomalous videos in total for the training and the performance evaluation. We split this dataset such that 0.4:0.2:0.4 is the training:validation:test ratio. For the training of the neural network, we stitch the available videos to obtain longer data streams. Specifically, in each training data stream, we include both nominal and anomalous videos where we keep the nominal:anomalous ratio approximately as 0.75:0.25. We use the mean squared error as the loss function and the Adam as the optimizer.

As a benchmark test, we use the algorithm provided in Liu et al. [81], which predicts the future frames and evaluates the prediction errors for anomaly detection. We use the sequential version of this algorithm, where an anomaly is declared as soon as the prediction error exceeds a certain threshold. Fig. 4.13 shows the conditional ADD (CADD) vs. FAP in terms of seconds for DeepQCD and the benchmark algorithm in [81], where we define the CADD as the ADD given that the anomaly is detected

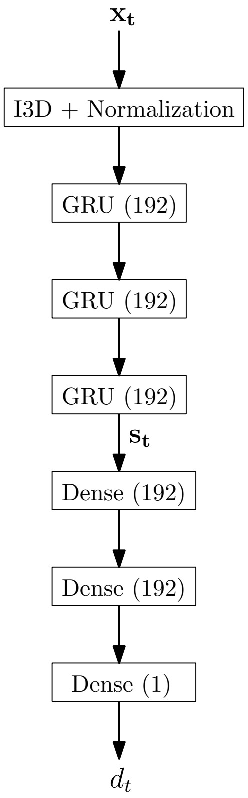


Figure 4.12: Neural network architecture for the online anomaly detection over surveillance videos.

before the video ends.

As an additional benchmark, we use the algorithm provided in Sultani et al. [125], which assigns an anomaly score to each video segment. It is a batch method and classifies the entire video as either nominal or anomalous. Although it is not possible to compute the ADD of this algorithm, we can compute the associated receiver operating characteristic (ROC) curve, showing the true positive rate (TPR) vs. false positive rate (FPR) as the test threshold varies. Fig. 4.14 shows the ROC curves for DeepQCD and the benchmark algorithms provided in [125] and [81]. Note that DeepQCD and the benchmark algorithm in [125] utilize both the nominal and anomalous data in their training phases, while the benchmark algorithm in [81] utilizes the nominal data only in its training phase.

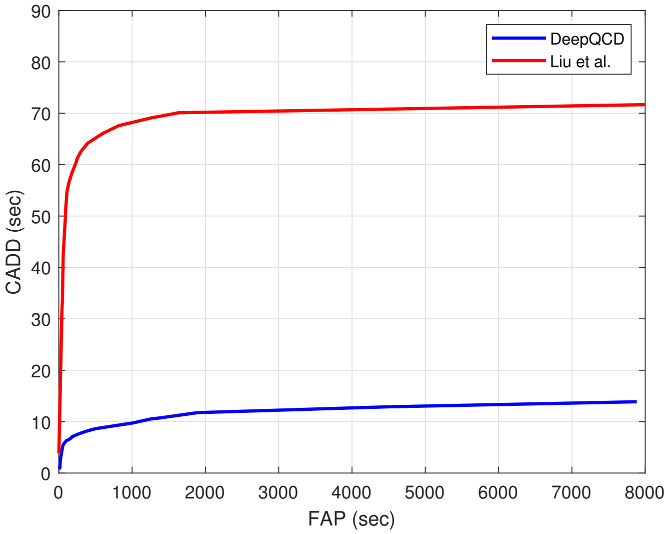


Figure 4.13: CADD vs. FAP curves for DeepQCD and the benchmark test in [81] in the online anomaly detection over surveillance videos.

4.6.2 Online Detection of IoT Botnet Attacks

Network-based detection of IoT botnet attacks (N-BaIoT) dataset [91] obtained from the UCI Machine Learning Repository [28] provides network traffic statistics for an IoT network under nominal system operation as well as under IoT botnet attacks. The network traffic statistics include time intervals between packet arrivals, packet sizes and counts, and so forth such that each data point is 115-dimensional, that is, $\boldsymbol{x}_t \in \mathbb{R}^{115}$. In case of botnet attacks, attackers inject malware to the IoT devices and use them to perform larger scale attacks over the entire network. As an example attack scenario, we consider that a spam attack is performed by the BASHLITE botnet [91] and we monitor the thermostat device for online attack detection.

For DeepQCD, we use the neural network architecture shown in Fig. 4.15. For the GRU layers, we use the dropout at the rate of 0.25. Moreover, before the dense layers, we perform the batch normalization. For the first dense layer, we use the ReLu activation and for the second dense layer, we use the sigmoid activation. Note that for both training and testing phases, we normalize the data before feeding them into

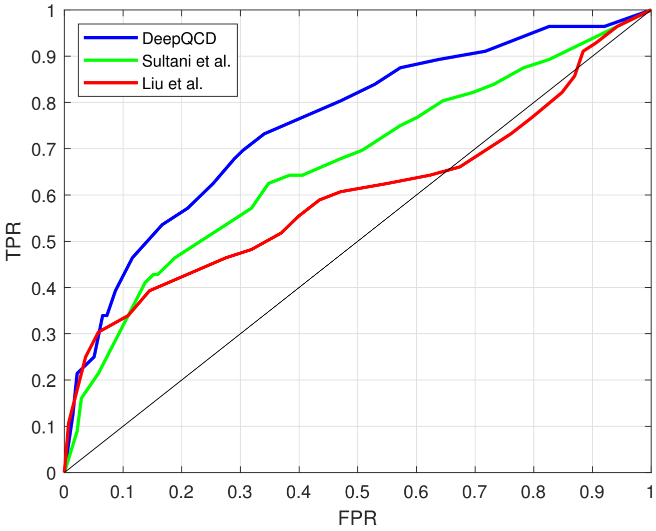


Figure 4.14: ROC curves for DeepQCD and the benchmark tests in [125] and [81] in the online anomaly detection over surveillance videos.

the neural network. Further, we use the mean squared error as the loss function and the Adam as the optimizer. We split the dataset for the thermostat device given in the N-BaIoT dataset such that 0.4:0.2:0.4 is the training:validation:test ratio. In the training phase, we uniformly sample from the nominal samples before the change-point and from the spam attack samples after the change-point, where the stream length is 2048 and the change-point is uniform between 128 and 1920.

Although the test data is limited, for the ADD vs. PFA performance curve, we obtain an infinite data stream by uniformly sampling from the nominal dataset and the attack dataset for the pre- and post-change cases, respectively. For the ADD calculations, we assume $\tau = 1$. As the benchmark algorithms, we use the principal component analysis (PCA)-based nonparametric CUSUM-like test in [68], QuantTree algorithm in [11], the Information Theoretic Multivariate Change Detection (ITMCD) algorithm in [37], and NN graph-based QCD algorithm in [18]. Fig. 4.16 shows that DeepQCD achieves significantly smaller ADDs for the same levels of FAP compared to the benchmark algorithms.

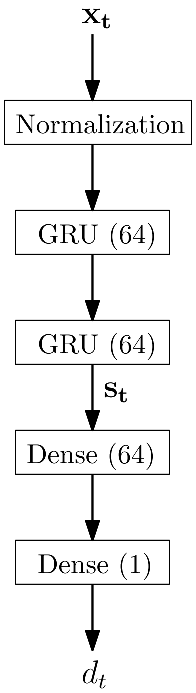


Figure 4.15: Neural network architecture for the online detection of IoT botnet attacks.

4.7 Concluding Remarks

This chapter has proposed a novel easy-to-implement data-driven QCD procedure, called DeepQCD, based on a generic neural network architecture composed of initial data processing, recurrent, and regression layers. The neural network discovers the change detection rule directly from the raw observations without requiring any prior knowledge on the observed data stream or any feature engineering. Specifically, the initial data processing layers filter out the noise and extract useful features from data. The recurrent layers build an internal state representation that summarizes the observation history for the QCD task. Finally, the regression layers map the internal state into the decision statistic, defined as the posterior probability that change has taken place given the data stream seen so far. The proposed DeepQCD algorithm has been shown to be very powerful in many scenarios where it performs nearly the

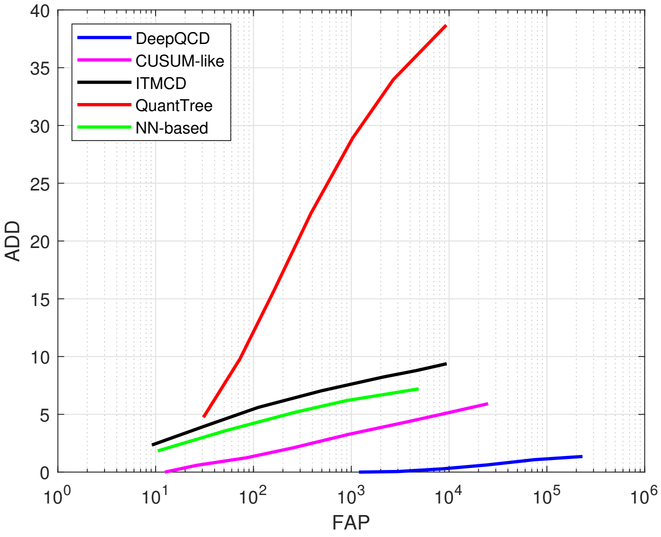


Figure 4.16: ADD vs. FAP curves for DeepQCD and the benchmark tests in the online detection of IoT botnet attacks.

same or even better than the existing model-based QCD procedures. Furthermore, experiments with real-world data have illustrated that DeepQCD outperforms the existing alternatives over challenging applications.

Chapter 5

Online Distributed Differentially Private Anomaly Detection over Networks

5.1 Introduction

5.1.1 Background

In real-time monitoring of safety-critical systems, deviations from the regular behavior, that is, anomalies, should be quickly identified for a timely and effective response based on the system data obtained in real time [5, 50, 66, 145]. Moreover, in distributed systems where each node/user/device holds privacy-sensitive data, the data-driven statistical inference process should not violate the confidentiality of data providers [33, 115]. Further, since the real-world data streams might have arbitrary statistical characteristics [68], this chapter aims for an effective online distributed privacy-preserving anomaly detection mechanism that is free of data model assumptions and hence applicable to a variety of practical settings. The corresponding application cases include but are not limited to vehicular networks [35], smart grid [47], Internet of Things (IoT) networks [130], and the cellular networks [44]. For example, in distribution smart grids, user electricity consumption patterns can be used to build sensitive user profiles for malicious purposes. This chapter aims to ensure security of such systems while effectively maintaining data privacy.

Building a data-driven mechanism comes together with the risk of violating the privacy of data providers. Hence, usually a balance between data utilization and data privacy needs to be sought. For instance, for the network-wide anomaly detection problem, a perfectly private regime requires that nodes share no information with respect to their data. In this case, however, the network-wide decision maker can only have an arbitrary decision about possible anomalies. Hence, the perfectly private regime corresponds to the worst-case scenario in terms of security. On the other end of the spectrum, in a completely non-private regime, the decision maker has access to all the network-wide data in its raw form, that is, the full data utilization, leading, in principle, to the optimal decision about anomalies (the best security performance). These two extremes illustrate the privacy-security tradeoff in the network-wide anomaly detection problem. This chapter aims to characterize this tradeoff analytically to enable a system designer in choosing the system parameters based on the desired performance levels.

Towards the objective of analyzing privacy-sensitive data while maintaining data privacy, various techniques have been developed in the literature such as homomorphic encryption, secure multi-party computation, federated learning, and differential privacy [115, 118]. Homomorphic encryption [15] transforms the data in such a way that arithmetic operations performed over the encrypted data correspond to the same arithmetic operations over the original data. This enables processing encrypted data directly without having access to the data in its raw form. In a distributed setting, this method requires a coordination among multiple parties, particularly a centralized key management authority. Moreover, the homomorphic encryption is computationally intensive, which might limit its practical use. Alternatively, the secure multi-party computation [148] enables a set of nodes to compute a desired function collaboratively without revealing their own sensitive data. Although promising, this method also requires a coordination among nodes usually via peer-to-peer communication, which might be costly in practical settings, especially over large-scale networks.

Federated learning is an iterative distributed learning procedure where a set of nodes is coordinated by a central node with the goal of learning a common model using the sensitive data at the nodes in a privacy-preserving manner [89, 116]. In this scheme, the central node sends the latest model parameters to the nodes and each node sends back an update computed through its own data. The central node then updates the model parameters by fusing the local updates. In [89], a deep neural network is iteratively trained where at each time the nodes compute the local gradient signals and then the central node updates the network weights via the average of the local gradients. In terms of privacy, the main concern is that the local update signals can still reveal sensitive information to the central node.

Differential privacy (DP) [33] is a probabilistic framework based on the notion of indistinguishability. In particular, observing an output of a differentially private algorithm, one cannot infer whether any specific individual/node/device contributed to the data. This ensures roughly the same level of privacy to each data provider. More specifically, change/removal of the data of any single data provider does not significantly change the output likelihood of a differentially private algorithm. In this framework, privacy is mainly achieved by randomizing the released statistics from a database, where the worst-case privacy risk can be quantified and calibrated with the level of randomization. The randomization can be achieved in many manners such as input perturbation (via additive noise), output perturbation, objective function perturbation, and exponential selection mechanism [115].

5.1.2 Related Work

There has been a growing research interest in differentially private machine learning and signal processing, for example, see [2, 12, 22, 26, 54, 75, 78, 115, 146]. For a differentially private algorithm, mainly the effect of privacy constraints on the algorithm performance should be analyzed, which allows the computation of achievable performance given the data and the desired privacy level. The DP literature commonly presumes a single fully-trusted centralized data collector that has access to all the system-wide data in its raw form and releases some statistics from this database privately.

Albeit to a lesser extent, distributed DP schemes with an untrusted data aggregator have also been studied. For instance, distributed implementations of privacy-preserving databases through distributed noise generation have been studied where each node generates a share of overall random noise [32]. This scheme requires cooperation and coordination among nodes, which might be costly over large-scale or dynamic networks. In [118], privacy-preserving aggregation of discrete time-series data is studied and a differentially private stream aggregation algorithm is presented where a group of users/nodes periodically send encrypted randomized messages to an untrusted data aggregator. This mechanism achieves aggregator obliviousness, that is, the aggregator is not able to obtain any unintended information about the individual nodes other than the intended aggregate statistic over the network. Similarly, in [76], a private stream aggregation algorithm is presented where, different from [118], neither coordination among nodes nor a centralized key management authority is required to achieve the aggregator obliviousness.

Privacy-preserving change and anomaly detection have also been studied in recent literature. A subset of these studies claim practical privacy benefits without rigorous privacy analysis. Such algorithms can be mainly motivated by the data processing inequality, that is, the processed or transformed form of data carries less information than the data in its raw form. For instance, in [57] a privacy-preserving anomaly detection scheme is presented where the sensitive data is firstly transformed for privacy concerns and then an anomaly detection algorithm is employed based on the Gaussian mixture model (GMM) and the Kalman filter. Particularly, a GMM is fit to the transformed data that is input to the Kalman filter to track its dynamics. In addition to making strong assumptions such as the GMM and linear dynamics (and hence the use of the Kalman filter), no theoretical privacy analysis is provided.

Another subset of studies provides provable privacy guarantees, such as DP, but usually with restrictive assumptions such as fully-known data models or bounded log-likelihood ratio. For instance, in [22] a window-based differentially private variant of the cumulative sum (CUSUM) algorithm is presented in a setting where the pre- and post-change data models are known. More particularly, the log-likelihood ratio is perturbed to provide a private estimate of the change-point. It is a centralized detection method as the entire data is assumed to be accessible in its row form by the decision maker. Similarly, in [12] the differentially private hypothesis testing problem is studied in a model-based centralized setting and a window-based private change-point detection algorithm is provided.

In [26], differentially private hypothesis testing is studied for an i.i.d. Gaussian sequence to decide whether the observed sequence has a given mean value. A generalized likelihood ratio test is used as a solution where the empirical mean of the observed data is perturbed for DP and the corresponding impact on the receiver operating characteristics and the decision threshold is characterized. In [38], a server is assumed to aggregate data from local devices and then send it to the third parties in a privacy-preserving manner (via perturbation) for an investigation about possible anomalies. However, no entity other than its owner should access the data in its raw form for an effective privacy protection. Further, in [42] the private stochastic gradient framework [121] is used for online training of an anomaly detector. Moreover, selective screening and active learning are used to reduce the required number of labeled data for the training process. In [146], a distributed intrusion detection scheme is proposed for vehicular networks based on distributed privacy-preserving machine learning and the alternating direction method of multipliers in which the vehicles collaborate via vehicle-to-vehicle communications for training of a network-wide classifier to detect attacks or intrusions while protecting the privacy of training data.

5.1.3 Contributions

We design and analyze a novel data-driven online anomaly detection algorithm for distributed systems based on the QCD and the private stream aggregation frameworks. The proposed mechanism provides early and reliable anomaly detection and in the meantime effectively protects the privacy of local sensitive data at each node. We propose a cooperative scheme where each node contributes to network-wide anomaly detection in a privacy-preserving manner. We list our main contributions as follows:

- We propose to extract and periodically release a minimal task-oriented univariate statistic from the observed high-dimensional data stream at every node, where the local data processing is free of data model assumptions.

- We propose a novel low-complexity differentially private network-wide anomaly detection algorithm.

- We derive an asymptotic approximation and an asymptotic lower bound for the average false alarm period (FAP) of the proposed algorithm.

- We derive an asymptotic approximation and an asymptotic upper bound for the average detection delay (ADD) of the proposed algorithm in cases where the anomaly signal can be specified. Additionally, we derive the worst-case asymptotic upper bound on the ADD of the proposed algorithm without needing the knowledge of the anomaly signal.

- We show the analytical privacy-security tradeoff, particularly the tradeoff between anomaly detection performance and the DP level, where the tradeoff is controlled via the variance of local perturbation noise.

5.1.4 Organization

The remainder of the chapter is organized as follows. Sec. 5.2 describes the problem. Sec. 5.3 explains the solution approach. Sec. 5.4 presents an analysis of the proposed solution scheme. Sec. 5.5 evaluates the performance of the proposed solution over real-world data. Finally, Sec. 5.6 concludes the chapter.

5.2 Problem Description

Consider a distributed network consisting of N nodes where each node $n \in \{1, 2, \ldots, N\}$ has a high-dimensional observation $\boldsymbol{x}_{t,n} \in \mathbb{R}^{m_n}$ at each time t, where $m_n \gg 1$ denotes the data dimensionality at node n. At an unknown time τ, called the change-point, an unexpected event (anomaly) happens over the network, such as a cyber-attack, a cyber-physical attack, or a random fault, and the network deviates from its nominal operation. Anomalies can be attributed to a change in the statistical properties of the data generating process and hence for the network-wide data $\boldsymbol{x}_t \triangleq [\boldsymbol{x}_{t,1}^{\mathrm{T}}, \boldsymbol{x}_{t,2}^{\mathrm{T}}, \ldots, \boldsymbol{x}_{t,N}^{\mathrm{T}}]^{\mathrm{T}}$, we can write

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

where $f_0^{\boldsymbol{x}}$ denotes the probability density function (pdf) of $\boldsymbol{x}_t$ under regular conditions and $f_1^{\boldsymbol{x}}$ denotes the pdf of $\boldsymbol{x}_t$ after the anomaly.

Our goal is to detect network-wide anomalies timely and reliably based on the observed data sequence, corresponding to a QCD problem [5, 65, 108]. In the QCD framework, the Lorden's minimax problem aims to minimize the worst-case ADD subject to a lower bound on the FAP [84]. In particular, letting Γ be the stopping time at which a change is declared and $J(\Gamma)$ be the worst-case ADD, given by

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

where $(\cdot)^+ = \max\{0, \cdot\}$, $\mathcal{F}_\tau$ denotes the set of observations up to the change-point τ, and $\operatorname{ess\,sup}$ denotes the essential supremum, the minimax problem can be stated as follows [84]:

$$\inf_{\Gamma} \; J(\Gamma) \;\; \text{subject to} \;\; \mathbb{E}_\infty[\Gamma] \geq \zeta, \tag{5.1}$$

where $\mathbb{E}_\infty[\Gamma]$ denotes the FAP and ζ denotes the desired lower bound on the FAP.

In cases where the probabilistic data models $f_0^{\boldsymbol{x}}$ and $f_1^{\boldsymbol{x}}$ are known and the network-wide data $\{\boldsymbol{x}_t\}_t$ is fully accessible to a decision maker, the CUSUM algorithm is the optimal solution to the minimax problem in Eq. (5.1) [94]. Furthermore, if the data models are known except for some unknown parameters, the generalized CUSUM algorithm, making use of the estimates of unknown parameters, has asymptotic optimality properties [5, Sec. 5.3]. However, for high-dimensional real-world data streams, usually the nominal pdf $f_0^{\boldsymbol{x}}$ might be difficult to model or intractable to estimate. Moreover, the anomalous pdf $f_1^{\boldsymbol{x}}$ might take arbitrary unknown forms depending on the type and cause of the anomalies [64, 69]. Hence, in this chapter, we assume both $f_0^{\boldsymbol{x}}$ and $f_1^{\boldsymbol{x}}$ are unknown and hence we look for a data-driven (model-free) solution approach.

In distributed settings where the data is privacy-sensitive to its owner, statistical inference should be performed without violating the confidentiality of nodes. Firstly, if the locally observed data $\{\boldsymbol{x}_{t,n}\}_t$ is sensitive for node n, no other entity should be allowed to access the data in its raw form. In such cases, since the network-wide anomaly detection is critical for the network safety, nodes may only be willing to disclose some minimal information aligned with the anomaly detection task in an encrypted form because of privacy concerns. Considering such a setting, let every

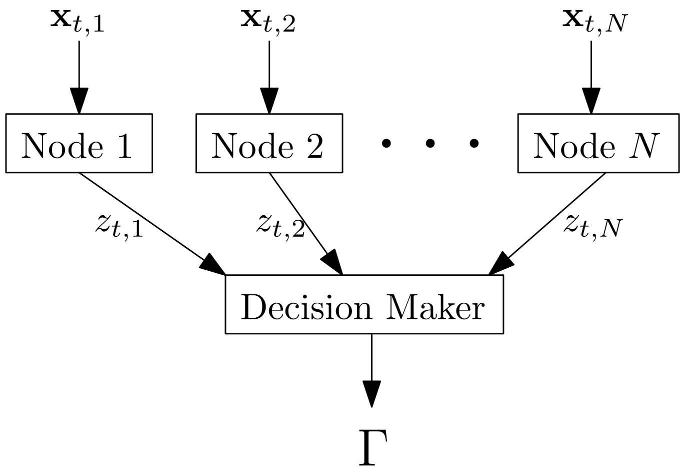


Figure 5.1: A graphical description of the problem.

node n, based on its observation $\boldsymbol{x}_{t,n}$, share a univariate signal $z_{t,n}$ with the network-wide decision maker at each time t, as illustrated in Fig. 5.1. The decision maker then receives $\boldsymbol{z}_t \triangleq [z_{t,1}, z_{t,2}, \ldots, z_{t,N}]$ from all nodes and decides on the anomaly based on the sequence of $\{\boldsymbol{z}_t\}_t$. Using this general architecture, we aim to design an effective solution scheme that achieves

- model-free online processing of the observed local data stream and disclosure of minimal task-oriented information at each node,
- differentially private aggregation of the node messages at the decision maker, and
- quick and reliable network-wide anomaly detection.

Moreover, we aim to analyze the properties of the designed solution scheme, particularly the privacy guarantees, the anomaly detection performance, and the effect of privacy guarantees on the anomaly detection performance.

5.3 Solution Approach

Our proposed solution consists of three functional modules: local data processing, private stream aggregation, and online anomaly detection (see Fig. 5.2). In the first module, sensitive data is analyzed and processed locally at the node it belongs, and some useful information is extracted for the anomaly detection task. Then, in the second module, for DP, instead of releasing the extracted information directly, it is first perturbed via additive noise and then a form of cryptographic communication is utilized between the nodes and the decision maker to ensure that the decision maker can only decrypt an aggregate statistic over the entire network but not the individual node information. Finally, in the third module, the decision maker performs online statistical inference for a network-wide anomaly based on the available information. The local data processing, perturbation, and encryption tasks are carried out at the nodes while the decryption and online anomaly detection tasks are performed at the decision maker. Next, we explain our solution steps in a greater detail.

5.3.1 Local Data Processing

In the proposed scheme, after observing $\boldsymbol{x}_{t,n}$ at time t, node n computes a local outlierness score $p_{t,n} \in [0, 1]$ corresponding to $\boldsymbol{x}_{t,n}$. For this purpose, nodes can use some distance-based, neural networks-based, or subspace-based method, provided that the method is nonparametric as the data models are unknown. In our design, we propose a generic data processing scheme, where each node can use a different method under this generic scheme based on the characteristics of its local data. As such, each node can keep its local processing method secret to make it difficult to interpret its raw data from the output of local processing. In addition, even if the nodes use the identical method, the corresponding algorithm parameters might still be different across the nodes.

Firstly, we assume each node n stores a historical dataset $\mathcal{X}_n \triangleq \{\boldsymbol{x}_{1,n}, \dots, \boldsymbol{x}_{W_n,n}\}$

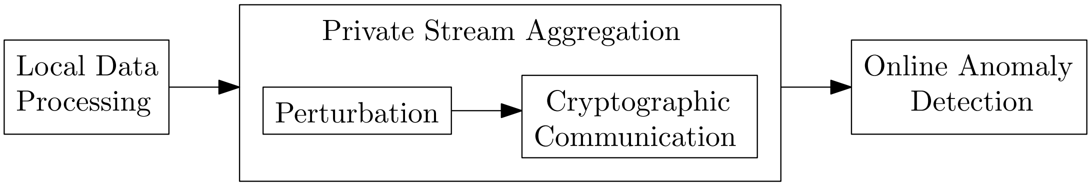


Figure 5.2: An overview of the proposed solution approach.

of size W_n consisting of nominal (anomaly-free) local samples. The proposed procedure at node n consists of an offline and an online phase. In the offline phase, a set of useful univariate summary statistics is extracted from $\mathcal{X}_n$ and stored locally. These statistics are used to represent the nominal behavior of node n. Then, in the online phase, as a new sample $\boldsymbol{x}_{t,n}$ is acquired, the online summary statistic corresponding to $\boldsymbol{x}_{t,n}$, denoted with $s_{t,n}$, is computed and the corresponding p-value estimate $p_{t,n}$ is computed as a local outlierness score based on how likely it is to observe $s_{t,n}$ under nominal conditions. Next, as a useful and illustrative example, we explain our local data processing method based on the principal component analysis (PCA).

If the observed high-dimensional data stream at node n exhibits a low intrinsic dimensionality, we can write:

$$\boldsymbol{x}_{t,n} = \boldsymbol{y}_{t,n} + \boldsymbol{r}_{t,n},$$

where $\boldsymbol{y}_{t,n}$ is the representation of $\boldsymbol{x}_{t,n}$ in a lower-dimensional submanifold and $\boldsymbol{r}_{t,n}$ is the residual term, mostly consisting of noise. The PCA is a well-known nonparametric method to learn linear submanifolds [8, Sec. 12.1]. Then, assuming the local nominal data can be well represented in a linear submanifold, we use the PCA to learn a nominal submanifold for node n using $\mathcal{X}_n$. Since anomalous data is expected to deviate from the nominal submanifold, the magnitude of the residual term, that is, $\|\boldsymbol{r}_{t,n}\|_2$, is expected to take higher values for anomalous data compared to the nominal data. We can then use the magnitude of the residual term as a summary statistic to make a distinction between anomalous and nominal data. First, in an offline phase, we employ the PCA to determine a representative nominal submanifold and compute a set of residual magnitudes. From this set of nominal summary statistics, we then form a nominal empirical distribution function (edf). In the online phase, we estimate the p-values corresponding to online residual magnitudes based on the nominal edf, as detailed next.

5.3.1.1 Offline Phase

We partition $\mathcal{X}_n$ into $\mathcal{X}_{n,1}$ and $\mathcal{X}_{n,2}$ with sizes $W_{n,1}$ and $W_{n,2}$, respectively. Using $\mathcal{X}_{n,1}$ and the PCA, we learn a representative nominal submanifold and then using $\mathcal{X}_{n,2}$, we determine a set of nominal summary statistics.

For $\mathcal{X}_{n,1}$, letting $\bar{\boldsymbol{x}}_n$ be its sample mean,

$$\bar{\boldsymbol{x}}_n \triangleq \frac{1}{W_{n,1}} \sum_{\boldsymbol{x}_{i,n} \in \mathcal{X}_{n,1}} \boldsymbol{x}_{i,n}, \tag{5.2}$$

and $\boldsymbol{Q}_n$ be its sample data covariance matrix,

$$\boldsymbol{Q}_n \triangleq \frac{1}{W_{n,1}} \sum_{\boldsymbol{x}_{i,n} \in \mathcal{X}_{n,1}} (\boldsymbol{x}_{i,n} - \bar{\boldsymbol{x}}_n)(\boldsymbol{x}_{i,n} - \bar{\boldsymbol{x}}_n)^{\mathrm{T}}, \tag{5.3}$$

we compute the eigenvalues $\{\lambda_{i,n} : i = 1, 2, \ldots, m_n\}$ and the eigenvectors $\{\boldsymbol{v}_{i,n} : i = 1, 2, \ldots, m_n\}$ of $\boldsymbol{Q}_n$. We then determine the dimensionality of the linear submanifold, denoted by r_n, for the desired fraction of data variance retained in the submanifold,

$$\gamma_n \triangleq \frac{\sum_{i=1}^{r_n} \lambda_{i,n}}{\sum_{i=1}^{m_n} \lambda_{i,n}} \leq 1, \tag{5.4}$$

where the r_n-dimensional subspace is spanned by the orthonormal eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\ldots$, $\boldsymbol{v}_{r_n}$ corresponding to the largest r_n eigenvalues λ_1, λ_2, $\ldots$, λ_{r_n}. Then, defining $\boldsymbol{V}_n \triangleq [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_{r_n}]$, we compute the residual term as follows:

$$\boldsymbol{r}_{t,n} = (\boldsymbol{I}_{m_n} - \boldsymbol{V}_n \boldsymbol{V}_n^{\mathrm{T}})(\boldsymbol{x}_{t,n} - \bar{\boldsymbol{x}}_n), \tag{5.5}$$

where $\boldsymbol{I}_{m_n}$ is an $m_n \times m_n$ identity matrix.

After computing $\bar{\boldsymbol{x}}_n$ and $\boldsymbol{V}_n$ as described above, we use Eq. (5.5) to compute the residual magnitudes for the set $\mathcal{X}_{n,2}$, that is, $\{\|\boldsymbol{r}_{j,n}\|_2 : \boldsymbol{x}_{j,n} \in \mathcal{X}_{n,2}\}$, as a set of nominal summary statistics. We then sort the nominal summary statistics in ascending order and store the sorted set in the local memory.

5.3.1.2 Online Phase

For each newly observed sample $\boldsymbol{x}_{t,n}$, node n computes $\boldsymbol{r}_{t,n}$ as in Eq. (5.5) and its magnitude. The node then estimates the corresponding p-value as follows [68]:

$$p_{t,n} = \frac{1}{W_{n,2}} \sum_{\boldsymbol{x}_{j,n} \in \mathcal{X}_{n,2}} \mathbb{1}\{\|\boldsymbol{r}_{j,n}\|_2 > \|\boldsymbol{r}_{t,n}\|_2\}, \tag{5.6}$$

which is simply the fraction of nominal summary statistics greater than the online summary statistic. The output of the local data processing module of node n is $p_{t,n}$ at time t.

Note that the p-value estimate given in Eq. (5.6) almost surely converges to the actual p-value as the size of the set of nominal summary statistics grows to infinity, that is, as $W_{n,2} \to \infty$. This is because the nominal edf formed by $\{\|\boldsymbol{r}_{j,n}\|_2 : \boldsymbol{x}_{j,n} \in \mathcal{X}_{n,2}\}$ pointwise almost surely converges to the actual nominal cumulative distribution function (cdf) of the summary statistic $\|\boldsymbol{r}_{j,n}\|_2$ as the sample size grows, by the Glivenko-Cantelli theorem [132]. Furthermore, under nominal conditions (no anomaly), the p-value is a uniform random variable taking values between 0 and 1, denoted by $\mathcal{U}[0,1]$. Our p-value estimate $p_{t,n}$ is hence asymptotically (as $W_{n,2} \to \infty$) uniform under regular conditions, that is, for $t < \tau$. Moreover, the p-value is expected to take smaller (close to zero) values for outliers and anomalous samples, and hence we expect smaller $p_{t,n}$ values for $t \geq \tau$.

The specified output of the local data processing module, that is, the p-value estimate, while useful to infer possible anomalies, is not quite as informative to interpret the raw local data. This is because, irrespective of the local data processing method and the observed data, the p-value estimate is always (asymptotically) a uniform random variable under regular conditions. Hence, in distributed anomaly detection, on the one hand, releasing such a task-oriented statistic from the nodes can be useful to preserve the data privacy. On the other hand, this is not considered as a perfect privacy solution since releasing such summary statistics may still lead to side information leakages and reconstruction attacks [33]. Therefore, we need additional techniques to improve privacy. Towards this objective, we next explain our private stream aggregation module.

5.3.2 Private Stream Aggregation

To achieve provable worst-case privacy guarantees, we aim for DP, for which we perturb the released statistic from nodes via additive noise and moreover, we utilize a form of cryptographic communication to ensure that the decision maker learns almost nothing about the individual nodes [76,118] while effectively performing network-wide anomaly detection.

5.3.2.1 Perturbation

Each node n perturbs its local p-value estimate $p_{t,n}$ via additive white Gaussian noise (AWGN), i.i.d. over time and between the nodes. Denoting the perturbed signal by $\tilde{p}_{t,n}$ yields

$$\tilde{p}_{t,n} = p_{t,n} + v_{t,n},$$

where $v_{t,n} \sim \mathcal{N}(0, \sigma^2)$ denotes zero-mean AWGN with variance σ^2. Higher levels of σ^2 increase uncertainty about the released statistic.

5.3.2.2 Cryptographic Communication

The purpose of cryptographic communication is such that the decision maker is able to only decrypt the noisy mean of the local p-value estimates, defined by

$$y_t \triangleq \frac{1}{N} \sum_{n=1}^{N} \tilde{p}_{t,n}. \tag{5.7}$$

To achieve this without any coordination between nodes in both static and dynamic network settings,[1] we make use of an auxiliary node [76], as illustrated in Fig. 5.3.

[1] In dynamic network settings, new nodes can join and leave over time, which is especially relevant to the case of node failures and to networks that are dynamic in nature such as mobile networks and vehicular networks.

In this mechanism, each node n generates a private key $k_{t,n} > 0$ at each time t and obtains the encrypted information to be sent to the decision maker as follows:

$$z_{t,n} = \tilde{p}_{t,n} + k_{t,n}.$$

The key generation process is specific to each node, kept secret, and independent from the other nodes. Ideally, it should be difficult to track the generated key pattern and hence to infer $\tilde{p}_{t,n}$ from $z_{t,n}$. Moreover, the generated keys should not carry any information relevant to the local data. As an example, nodes can obtain the keys as realizations of time-varying random variables. The generated keys are also sent to the auxiliary node that computes the negative average of the received keys from the nodes:

$$a_t \triangleq -\frac{1}{N}\sum_{n=1}^{N} k_{t,n},$$

and sends the result to the decision maker. The auxiliary node does not share any other information with the decision maker or any other node.

After receiving $\{z_{t,n}, n = 1, 2, \ldots, N\}$ from the nodes, and a_t from the auxiliary node, the decision maker takes the average of the node messages, then sums the average with a_t, and obtains y_t, see Eq. (5.7):

$$\begin{aligned} y_t &= a_t + \frac{1}{N}\sum_{n=1}^{N} z_{t,n} = a_t + \frac{1}{N}\sum_{n=1}^{N}(\tilde{p}_{t,n} + k_{t,n}) \\ &= \underbrace{a_t + \frac{1}{N}\sum_{n=1}^{N} k_{t,n}}_{0} + \frac{1}{N}\sum_{n=1}^{N} \tilde{p}_{t,n} = \frac{1}{N}\sum_{n=1}^{N} \tilde{p}_{t,n}. \end{aligned}$$

Remark 1: As an alternative to the presented cryptographic communication scheme, a secret key sharing mechanism can be designed among the nodes such that the decision maker can only decrypt the sum of the individual node signals, without needing an auxiliary node [76]. However, this mechanism requires a coordination and peer-to-peer communication between the nodes. Furthermore, it is not robust to node failures or dynamic network settings, as it needs to be redesigned after new nodes joining or

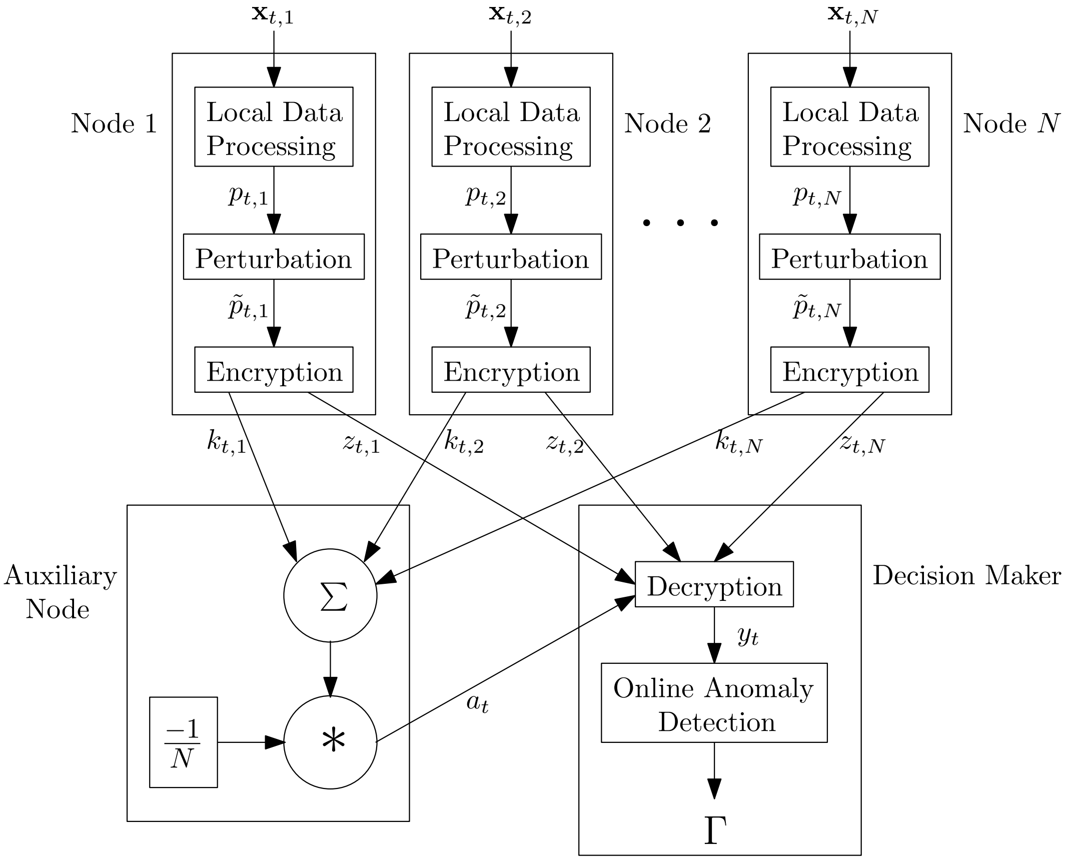


Figure 5.3: A graphical illustration of the proposed solution scheme.

leaving the network. Hence, the presented scheme has advantages in terms of robustness to nodes dynamically joining or leaving and it does not require coordination or communication among the nodes, that is, each node n can generate its own key $k_{t,n}$ independently from the other nodes.

5.3.3 Online Anomaly Detection

We firstly analyze the statistical properties of the information y_t obtained at the decision maker in both the nominal and anomalous cases and then explain the proposed online anomaly detection algorithm. Finally, we summarize the overall anomaly detection scheme and provide its time and space complexity analysis.

5.3.3.1 Distribution of y_t

The information obtained at the decision maker at time t can be rewritten as

$$\begin{aligned} y_t &= \frac{1}{N}\sum_{i=1}^{N} \tilde{p}_{t,n} = \frac{1}{N}\sum_{i=1}^{N}(p_{t,n} + v_{t,n}) \\ &= \underbrace{\frac{1}{N}\sum_{i=1}^{N} p_{t,n}}_{\bar{p}_t} + \underbrace{\frac{1}{N}\sum_{i=1}^{N} v_{t,n}}_{\bar{v}_t} \\ &= \bar{p}_t + \bar{v}_t, \end{aligned} \tag{5.8}$$

where $\bar{v}_t \sim \mathcal{N}(0, \sigma^2/N)$. Further, recalling that $p_{t,n} \sim \mathcal{U}[0,1], \forall n \in \{1, 2, \dots, N\}$ for $t < \tau$ and assuming $p_{t,n}$ is i.i.d. over time and space,[2] the central limit theorem yields, asymptotically

$$\bar{p}_t \sim \mathcal{N}\left(0.5, \frac{1}{12N}\right), \ t < \tau. \tag{5.9}$$

Then, since $\bar{p}_t$ and $\bar{v}_t$ are independent, we can write

$$y_t \sim \mathcal{N}\left(0.5, \frac{\sigma^2 + 1/12}{N}\right), \ t < \tau. \tag{5.10}$$

Further, if $\sigma^2 \gg 1/12$, it holds that approximately

$$y_t \sim \mathcal{N}(0.5, \sigma^2/N), \ t < \tau. \tag{5.11}$$

In case of an anomaly over the network, that is, for $t \geq \tau$, it is expected that anomalous nodes observe outliers more frequently, correspondingly smaller p-value estimates, leading to a decrease in the mean of y_t. Hence, we can argue that the mean of y_t is $0.5 - \gamma_t$ for $t \geq \tau$, where $\gamma_t \geq 0$ denotes the unknown and possibly time-varying mean decrease. Moreover, in case of an anomaly, we no longer have

[2] Obtaining an i.i.d. $p_{t,n}$ stream can be accomplished via the local data processing, for example, in the PCA-based method, if the linear submanifold well represents the observed data, the residual term will mostly correspond to noise, that can be assumed i.i.d. over time and space. Furthermore, in large-scale networks, data of any single node can be approximated nearly independent from the vast majority of other nodes, except the immediate neighborhood of the node.

$p_{t,n} \sim \mathcal{U}[0,1]$. In fact, the pdf of $p_{t,n}$ is unknown for $t \geq \tau$. Nevertheless, it is always true that $p_{t,n}$ takes values between 0 and 1, that is, $p_{t,n} \in [0,1]$, and for a random variable taking values in this range, its variance can at most[3] be 1/4. Hence, for $t \geq \tau$, each $p_{t,n}$ has a mean $\mu_{t,n} \in [0, 0.5]$ and a variance $\sigma^2_{t,n} \in [0, 1/4]$, $n \in \{1, 2, \dots, N\}$.

The central limit theorem has variants, ensuring convergence to the normal distribution for non-identical and/or dependent distributions under certain conditions. In large-scale networks (large N), we can consider $p_{t,n}$ as nearly independent but non-identically distributed across the nodes for $t \geq \tau$. Then, we can use the Lindeberg central limit theorem [7, p. 369] stating that given

$$s^2_{t,N} \triangleq \sum_{n=1}^{N} \sigma^2_{t,n},$$

if for every $\varepsilon > 0$, the condition

$$\lim_{N\to\infty} \frac{1}{s^2_{t,N}} \sum_{n=1}^{N} \mathrm{E}\left[(p_{t,n} - \mu_{t,n})^2\, \mathbb{1}\{|p_{t,n} - \mu_{t,n}| > \varepsilon\, s_{t,N}\}\right] = 0 \tag{5.12}$$

is satisfied, then

$$\frac{1}{s_{t,N}} \sum_{n=1}^{N} (p_{t,n} - \mu_{t,n})$$

converges to the standard normal distribution $\mathcal{N}(0,1)$. Equivalently stating, under the condition above, it asymptotically holds that

$$\bar{p}_t = \frac{1}{N} \sum_{n=1}^{N} p_{t,n} \sim \mathcal{N}\left(\frac{\sum_{n=1}^{N} \mu_{t,n}}{N}, \frac{s^2_{t,N}}{N^2}\right).$$

Noticing further that

$$0 \leq \sum_{n=1}^{N} \mu_{t,n} \leq N/2,$$

[3] For a random variable $x \in [0,1]$, its variance can be written as $\sigma^2_x \triangleq \mathbb{E}[x^2] - (\mathbb{E}[x])^2 \leq \mathbb{E}[x] - (\mathbb{E}[x])^2$, where the inequality is because of the fact that $x^2 \leq x$ for $x \in [0,1]$. Denoting $m \triangleq \mathbb{E}[x]$, we have $\sigma^2_x \leq f(m) \triangleq m - m^2$ where $m \in [0,1]$. Since the maximum value of the function $f(m)$ is 1/4 at $m = 1/2$, we have $\sigma^2_x \leq 1/4$.

and

$$0 \leq s_{t,N}^2 \leq N/4,$$

the mean of $\bar{p}_t$ is between 0 and 0.5 and the variance of $\bar{p}_t$ is between 0 and $\frac{1}{4N}$. Then, if $\sigma^2 \gg 1/4$, it approximately holds that, see Eq. (5.8),

$$y_t = \bar{p}_t + \bar{v}_t \sim \mathcal{N}(0.5 - \gamma_t, \sigma^2/N), \ t \geq \tau. \tag{5.13}$$

Note that the condition in Eq. (5.12) is satisfied in our case, as the indicator in Eq. (5.12) gets the value of 0 as $N \to \infty$. This is because the term $|p_{t,n} - \mu_{t,n}|$ is bounded due to $p_{t,n}, \mu_{t,n} \in [0,1]$, whereas $s_{t,N} \to \infty$ as $N \to \infty$.

Finally, if $\sigma^2 \gg 1/4$, we can write, see Eq. (5.11) and Eq. (5.13),

$$y_t \sim \begin{cases} \mathcal{N}(0.5, \sigma^2/N), & \text{if } t < \tau, \\ \mathcal{N}(0.5 - \gamma_t, \sigma^2/N), & \text{if } t \geq \tau. \end{cases} \tag{5.14}$$

Then, the high-dimensional network-wide anomaly detection problem reduces to the sequential detection of a mean change over a univariate Gaussian data stream. Hence, at this point, we make a transition from the originally nonparametric setting to a parametric setting for which we obtain an effective solution, as detailed next.

5.3.3.2 Online Detection

Denoting the nominal and anomalous pdfs of y_t by f_0^y and f_1^y, respectively, and defining $\theta \triangleq \sigma/\sqrt{N}$, we have $f_0^y \sim \mathcal{N}(0.5, \theta^2)$ and $f_1^y \sim \mathcal{N}(0.5 - \gamma_t, \theta^2)$, where f_1^y has an unknown and possibly time-varying parameter γ_t, see Eq. (5.14). As argued before, for the Lorden's minimax problem given in Eq. (5.1) with unknown parameters in the data models, the generalized CUSUM algorithm can be used as an effective solution. Then, we propose the following generalized CUSUM detector at the network-wide decision maker:

$$\Gamma = \inf\Big\{m \in \mathbb{N} : \underbrace{\max_{1 \leq j \leq m} \sum_{t=j}^{m} \underbrace{\log \frac{\sup_{\gamma_t \geq \eta} f_1^y(y_t \,|\, \gamma_t)}{f_0^y(y_t)}}_{\beta_t}}_{g_m} \geq h\Big\}, \tag{5.15}$$

	Space	Time
Offline	$O\left(m_n^2 + W_{n,2}\right)$	$O\left(m_n^3 + (W_{n,1} + W_{n,2})m_n^2 + W_{n,2}\log(W_{n,2})\right)$
Online	$O\left(m_n\right)$	$O\left(m_n^2 + \log(W_{n,2})\right)$

Table 5.1: Space and time complexity of the proposed PCA-based procedure at node n.

where η denotes the minimum change of interest, determining the detector sensitivity, and h denotes the test threshold. In fact, these two algorithm parameters together control the false alarm rate of the proposed detector. Further, β_t and g_t denote the generalized log-likelihood ratio (GLLR) and the decision statistic, respectively, at time t where the decision statistic can be written in the following recursive form [5, Sec. 2.2]:

$$g_t = (g_{t-1} + \beta_t)^+. \tag{5.16}$$

Moreover, the GLLR β_t can be computed as follows:

$$\begin{aligned}\beta_t &= \frac{1}{2\theta^2}\sup_{\gamma_t \geq \eta}(1-2y_t)\gamma_t - \gamma_t^2 \\ &= \begin{cases} \frac{1}{2\theta^2}(0.5 - y_t)^2, & \text{if } y_t \leq 0.5 - \eta, \\ \frac{1}{2\theta^2}(1-2y_t)\eta - \frac{\eta^2}{2\theta^2}, & \text{if } y_t > 0.5 - \eta. \end{cases}\end{aligned} \tag{5.17}$$

5.3.3.3 Summary of the Overall Scheme

We summarize the proposed procedures at node n (employing the PCA-based local data processing method), the auxiliary node, and the network-wide decision maker in Alg. 5.1, Alg. 5.2, and Alg. 5.3, respectively. Moreover, we provide time and space complexity of the proposed procedures in Table 5.1, Table 5.2, and Table 5.3, respectively. Note that both the auxiliary node and the decision maker are only employed in the online detection phase while the nodes additionally employ an offline phase before the online phase.

Remark 2: The proposed detection scheme provides network-wide anomaly detection. An alternative to this scheme could be that each node locally employs a

Algorithm 5.1 Proposed PCA-based procedure at node n

```
Offline Phase
Partition $\mathcal{X}_n$ into $\mathcal{X}_{n,1}$ and $\mathcal{X}_{n,2}$ with sizes $W_{n,1}$ and $W_{n,2}$, respectively.
Compute $\bar{\boldsymbol{x}}_n$ and $\boldsymbol{Q}_n$ over $\mathcal{X}_{n,1}$ using Eq. (5.2) and Eq. (5.3), respectively.
Compute the eigenvalues $\{\lambda_{i,n} : i = 1, 2, \dots, m_n\}$ and the eigenvectors $\{\boldsymbol{v}_{i,n} : i = 1, 2, \dots, m_n\}$ of $\boldsymbol{Q}_n$.
Based on a desired level of $\gamma_n$, see Eq. (5.4), determine $r_n$ and form the matrix $\boldsymbol{V}_n \triangleq [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots \boldsymbol{v}_{r_n}]$.
for $j : \boldsymbol{x}_{j,n} \in \mathcal{X}_{n,2}$ do
    $\boldsymbol{r}_{j,n} = (\boldsymbol{I}_{m_n} - \boldsymbol{V}_n \boldsymbol{V}_n^{\mathrm{T}})(\boldsymbol{x}_{j,n} - \bar{\boldsymbol{x}}_n)$.
    Compute $\|\boldsymbol{r}_{j,n}\|_2$.
end for
Sort the nominal summary statistics $\{\|\boldsymbol{r}_{j,n}\|_2 : \boldsymbol{x}_{j,n} \in \mathcal{X}_{n,2}\}$ in ascending order and store.
Online Phase
Initialization: $t \leftarrow 0$
while $t < \Gamma$ do
    $t \leftarrow t + 1$
    Obtain the new sample $\boldsymbol{x}_{t,n}$.
    $\boldsymbol{r}_{t,n} = (\boldsymbol{I}_{m_n} - \boldsymbol{V}_n \boldsymbol{V}_n^{\mathrm{T}})(\boldsymbol{x}_{t,n} - \bar{\boldsymbol{x}}_n)$ and compute $\|\boldsymbol{r}_{t,n}\|_2$.
    $p_{t,n} = \frac{1}{W_{n,2}} \sum_{\boldsymbol{x}_{j,n} \in \mathcal{X}_{n,2}} \mathbb{1}\{\|\boldsymbol{r}_{j,n}\|_2 > \|\boldsymbol{r}_{t,n}\|_2\}$.
    $\tilde{p}_{t,n} = p_{t,n} + v_{t,n}$.
    $z_{t,n} = \tilde{p}_{t,n} + k_{t,n}$.
    Send $z_{t,n}$ to the decision maker and $k_{t,n}$ to the auxiliary node.
end while
```

	Space	Time
Online	$O(1)$	$O(N)$

Table 5.2: Space and time complexity of the proposed procedure at the auxiliary node.

separate detector for its own data. However, in many anomaly cases, especially small deviations from the regular operation such as stealthily designed attacks [67], while single nodes might seem normal, the network as a whole could be anomalous. In such cases, statistical evidence collected over the network enables quicker and more reliable anomaly detection.

Remark 3: The proposed detector is cooperative where each node is assumed to be regular and trusted. In the case of misbehaving (hacked or faulty) nodes, the post-change model for y_t can be arbitrarily different from Eq. (5.13). Particularly, nodes with a malicious intent may perform stealthy attacks to bypass the proposed

Algorithm 5.2 Proposed procedure at the auxiliary node

```
Initialization: $t \leftarrow 0$
while $t < \Gamma$ do
    $t \leftarrow t + 1$
    Receive $\{k_{t,n}, n = 1, 2, \ldots, N\}$ from the nodes.
    $a_t = -\frac{1}{N}\sum_{n=1}^{N} k_{t,n}$
    Send $a_t$ to the decision maker.
end while
```

Algorithm 5.3 Proposed procedure at the decision maker

```
Initialization: $t \leftarrow 0$, $g_0 \leftarrow 0$
while $g_t < h$ do
    $t \leftarrow t + 1$
    Receive $\{z_{t,n}, n = 1, 2, \ldots, N\}$ from the nodes, and $a_t$ from the auxiliary node.
    $y_t = a_t + \frac{1}{N}\sum_{n=1}^{N} z_{t,n}$
    Compute the GLLR $\beta_t$ using Eq. (5.17).
    $g_t \leftarrow (g_{t-1} + \beta_t)^+$
end while
$\Gamma \leftarrow t$, declare a network-wide anomaly.
```

detection mechanism, for example, by reporting high p-values (close to 1) to intentionally increase the mean of y_t and accordingly to hide the network-wide anomaly from the proposed generalized CUSUM detector. Another anomaly case that does not comply with the given post-change model is a cyber-attack against the network communication channels between nodes and the decision maker or the auxiliary node. In this case, the decision maker cannot successfully decrypt the intended message y_t. For such cases, as a countermeasure, we can monitor deviations from the nominal case by evaluating whether the observed y_t sequence fits well to the pre-change model in Eq. (5.10). We can accomplish this via a nonparametric sliding-window goodness-of-fit test, see for example [67]. In this case, any deviation from the regular network operation can be detected since no model is assumed for the post-change case. However, it usually leads to larger detection delays since the post-change model is unknown and more samples are required for a reliable decision. For completeness, we briefly explain the nonparametric sliding-window chi-squared test in Sec. 5.5 and also use it as a benchmark detector.

	Space	Time
Online	$O(1)$	$O(N)$

Table 5.3: Space and time complexity of the proposed procedure at the decision maker.

5.4 Analysis

In this section, we firstly analyze the theoretical privacy guarantees, particularly the DP of the proposed scheme. We then analyze the anomaly detection performance, in particular the FAP and the ADD of the proposed scheme. Finally, we characterize the effect of privacy guarantees on the anomaly detection performance, showing the analytical privacy-security tradeoff.

5.4.1 Differential Privacy

Firstly, we state the definition of (ϵ, δ)-DP.

Definition: Let $\epsilon > 0$, $\delta < 1$, and $\phi(\cdot)$ be a randomized function taking a dataset as its input. Moreover, let $\text{im}(\phi)$ denote the image of ϕ, that is, the set of all output values it can take. The function $\phi(\cdot)$ is (ϵ, δ)-differentially private if

$$\mathbb{P}(\phi(D_1) \in \mathcal{S}) \leq e^{\epsilon}\, \mathbb{P}(\phi(D_2) \in \mathcal{S}) + \delta$$

for all datasets D_1 and D_2 differing in only a single entry and over all subsets $\mathcal{S}$ of $\text{im}(\phi)$.

The definition above mainly says that the outcome of a differentially private function does not vary significantly if any single entry in the database is changed, where the amount of significance is captured with the ϵ and δ parameters. Here, ϵ and δ represent the worst-case privacy loss where lower values of them implies stronger privacy guarantees. Furthermore, if $\delta = 0$, it is called ϵ-DP.

Next, we state the definition of sensitivity of a function to changes in the database.

Definition: Let D be a collection of datasets and d be a positive integer. The sensitivity of a function $\phi : D \to \mathbb{R}^d$ is defined by

$$\Delta\phi \triangleq \max_{\mathrm{dist}(D_1,D_2)=1} \|\phi(D_1) - \phi(D_2)\|_1,$$

over all datasets D_1 and D_2 in D differing in a single entry, denoted by $\mathrm{dist}(D_1, D_2) = 1$.

Computing the sensitivity of a function enables easy calibration of the level of randomization to achieve the desired privacy level. A common randomization technique is perturbation via additive noise. Particularly, the Gaussian mechanism achieves the DP by adding Gaussian noise to the output of a function operating on a database. The following lemma is useful to calibrate the amount of Gaussian perturbation noise to obtain (ϵ, δ)-DP [33]:

Lemma 5.1: Let the information released from a database D be

$$\tilde{\phi}(D) = \phi(D) + \omega_t,$$

where ω_t is the perturbation noise. If

$$\omega_t \sim \mathcal{N}\left(0, \frac{\Delta^2\phi \; 2\log(1.25/\delta)}{\epsilon^2}\right),$$

then (ϵ, δ)-DP is achieved.

As we observe through Lemma 5.1, stronger privacy level, equivalently lower ϵ and/or δ, requires higher level of perturbation noise. This is intuitive since the noise variance increases uncertainty about the released information from the database and hence improves privacy. The following theorem specifies the variance of the local perturbation noise so that the proposed online distributed anomaly detection scheme achieves (ϵ, δ)-DP.

Theorem 5.1: If at each node $n \in \{1, 2, \ldots, N\}$, the variance of the perturbation noise $v_{t,n}$ is set as

$$\sigma^2 = \frac{2\log(1.25/\delta)}{N\epsilon^2}, \tag{5.18}$$

then the proposed anomaly detection scheme is (ϵ, δ)-differentially private.

Proof. See Appendix 5.7.1. □

Theorem 5.1 shows the amount of local perturbation noise necessary to achieve the desired DP level. It further shows the relationship between the network size N and the required noise level. Particularly, for a smaller network (smaller N), to achieve the same level of DP, we need to add higher level of noise (higher σ^2). This is also intuitive because for a network consisting of a few nodes, it is usually more difficult to mask individual node information via an aggregate statistic over the network, compared to masking individual nodes over large-scale networks. Finally, since the desired DP level can be achieved by calibrating σ^2, we hereafter refer to σ^2 (equivalently θ^2) as the DP parameter.

Remark 4: In addition to being a minimal task-oriented statistic as discussed in Sec. 5.3.1, another advantage of using the p-value estimates in the proposed solution scheme is that even if the observed data stream $\boldsymbol{x}_t$ might be unbounded, the p-value estimates are always bounded in the range of $[0, 1]$. We use this boundedness property to show the DP of the proposed scheme (see Appendix 5.7.1).

5.4.2 Anomaly Detection Performance

The average run length (ARL), denoted by $\mathbb{E}_\tau[\Gamma]$, is the first time, on average, the decision statistic g_t exceeds the decision threshold h of the proposed detector. The ARL can be used for both FAP and ADD computations. In particular, if there is no change or anomaly at all ($\tau = \infty$), the ARL corresponds to the FAP, that is, $\mathbb{E}_\infty[\Gamma]$. Furthermore, setting $\tau = 1$, the ARL corresponds to the worst-case ADD, $\mathbb{E}_1[\Gamma]$. This is because the proposed detector is a CUSUM-type detector for which the decision statistic being zero just before the change-point essentially describes the worst-case scenario in terms of detection delays. In other words, for any $z \geq 0$ with $g_{\tau-1} = z$, and expressing the ADD as a function of z, we have $\text{ADD}(z) \leq \text{ADD}(0)$ for the proposed detector. Note that for $\tau = 1$, we always have $g_{\tau-1} = 0$ since $g_0 = 0$. In the following, we derive the Wald's approximations for both the FAP and the worst-case ADD of the proposed detector as well as some performance bounds, particularly a lower bound on the FAP and an upper bound on the ADD to obtain performance guarantees. Note that our approximations and bounds are asymptotic as we use both the central limit theorems and the Glivenko-Cantelli theorem to asymptotically characterize the statistical properties of the information obtained at the decision maker, that is, $\{y_t\}_t$, (see Sec. 5.3.3.1).

5.4.2.1 Average False Alarm Period

For a reliable anomaly detection, false alarm events should be infrequent, or equivalently, the FAP should be large. The following theorem states the Wald's approximation for the FAP of the proposed detector. This approximation can be useful to adjust the system and algorithm parameters to control the false alarm rate of the proposed detector.

Theorem 5.2: Let $\rho \triangleq \eta/\theta$. If $\rho > 0.61$, then the Wald's approximation to the FAP is given by

$$\mathbb{E}_\infty[\Gamma] \approx \frac{2h + 2(e^{-w_0 h} - 1)/w_0}{Q(\rho) - \rho^2\, Q(-\rho)}$$

where $Q(\cdot)$ denotes the Q-function, that is, $Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2/2}\, du$, and $-1 < w_0 < 0$ is the unique solution to

$$f(w_0) \triangleq \frac{1}{\sqrt{w_0 + 1}} Q(\rho) + Q(-\rho)\, e^{0.5\, \rho^2 (w_0 + w_0^2)} = 1. \tag{5.19}$$

Proof. See Appendix 5.7.2. □

In Theorem 5.2, w_0 is computed by solving Eq. (5.19) numerically, which is simple since we look for a unique w_0 in the range of $(-1, 0)$ for which $f(w_0) = 1$. The solution can be obtained by searching among a set of linearly spaced w_0 values in this range.

Next, the following theorem provides a lower bound on the FAP of the proposed detector.

Theorem 5.3: If $\rho > 0.61$, a lower bound for the FAP is given by

$$\mathbb{E}_\infty[\Gamma] \geq e^{-w_0 h},$$

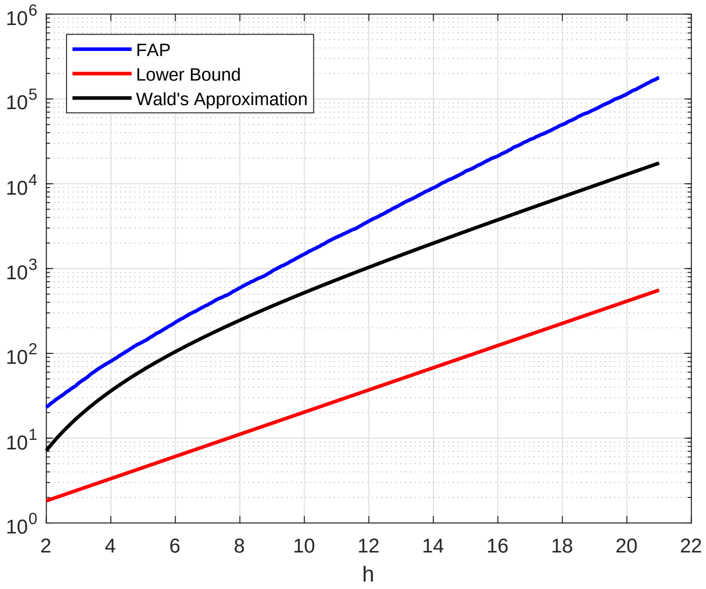


Figure 5.4: The FAP of the proposed detector, the theoretical lower bound on the FAP, and the Wald's approximation to the FAP for various levels of test threshold h.

where w_0 is as given in Eq. (5.19).

Proof. See Appendix 5.7.3. □

Next, to illustrate our theoretical results, we assume $y_t \sim \mathcal{N}(0.5, \theta^2)$ and plot the actual FAP of the proposed detector as well as the given lower bound and the Wald's approximation in Fig. 5.4, as the test threshold h varies. In this experiment, we choose $\eta = 0.06$ and $\theta = 0.08$. We observe through the figure that the Wald's approximation underestimates the FAP. This is because the Wald's approximation ignores the excess over the boundary for the random walk driven by the GLLR β_t, see Eq. (5.16), and in the pre-change case usually the lower threshold 0 is exceeded frequently during the random walk. Nevertheless, the derived approximation can still be useful, at least to achieve a performance guarantee (a lower bound on the FAP) in most cases.

5.4.2.2 Average Detection Delay

For a quick anomaly detection, detection delays should be as low as possible. Hence, the ADD analysis is useful to evaluate the expected performance of a detector. However, since anomalies can happen due to many reasons, it is usually difficult to fully specify the anomaly signal. In particular, for our proposed detector, it is difficult to characterize the mean decrease γ_t for $t \geq \tau$. But, in a special case where the mean decrease is constant after the change-point, that is,

$$\gamma_t = \gamma, \ t \geq \tau,$$

we can provide an analysis for the worst-case ADD of the proposed detector. Particularly, the following theorem states the Wald's approximation to the worst-case ADD of the proposed detector under this special condition.

Theorem 5.4: The Wald's approximation to the worst-case ADD is given by

$$\mathbb{E}_1[\Gamma] \approx \frac{h + \frac{e^{-w_1 h} - 1}{w_1}}{\frac{\gamma^2+\theta^2}{2\theta^2} Q\left(\frac{\eta-\gamma}{\theta}\right) + \frac{2\eta\gamma-\eta^2}{2\theta^2} Q\left(\frac{\gamma-\eta}{\theta}\right)},$$

where $w_1 > 0$ is the unique solution to

$$g(w_1) \triangleq Q\left(\frac{\eta-\gamma}{\theta}\right) \frac{e^{\frac{-w_1\gamma^2}{2\theta^2(w_1+1)}}}{\sqrt{w_1+1}} + Q\left(\frac{\gamma-\eta}{\theta}\right) e^{\frac{(\gamma^2-2\gamma\eta)w_1+\gamma^2 w_1^2}{2\theta^2}} = 1. \tag{5.20}$$

Proof. See Appendix 5.7.4. □

As before, we can compute w_1 by solving Eq. (5.20) numerically. Particularly, in the range of $w_1 > 0$, for a set of linearly spaced w_1 values, we look for the unique intersection point between the curves $g(w_1)$ and $\vartheta(w_1) = 1$.

Next, the following theorem provides an upper bound on the ADD.

Theorem 5.5: An upper bound on ADD is given by

$$\mathbb{E}_1[\Gamma] \leq \frac{h + Q\left(\frac{\eta-\gamma}{\theta}\right) \frac{\gamma^2+\theta^2}{2\theta^2} + Q\left(\frac{\gamma-\eta}{\theta}\right) \psi(a,b)}{\frac{\gamma^2+\theta^2}{2\theta^2} Q\left(\frac{\eta-\gamma}{\theta}\right) + \frac{2\eta\gamma-\eta^2}{2\theta^2} Q\left(\frac{\gamma-\eta}{\theta}\right)},$$

where

$$\psi(a,b) \triangleq a + \frac{\sqrt{b}\, e^{\frac{-a^2}{2b}}}{\sqrt{2\pi}\, Q(-a/\sqrt{b})},$$

$a \triangleq (2\gamma\eta - \eta^2)/(2\theta^2)$, and $b \triangleq \eta^2/\theta^2$.

Proof. See Appendix 5.7.5. □

Recall that our derivation for the theoretical upper bound on the ADD is based on the assumption that the post-change mean decrease satisfies $\gamma_t = \gamma$ for $t \geq \tau$. Using the result in Theorem 5.5, we can actually derive the worst possible upper bound on the ADD without needing this assumption. Particularly, we can use the fact that the Kullback-Leibler (KL) divergence is a measure of separability between the pre- and post-change data distributions and hence a measure of detectability in the QCD theory such that as the KL divergence between the pre- and post-change pdfs decreases, then the ADD increases. Hence, if $\gamma_t, t \geq \tau$ takes values minimizing the KL divergence between the pre- and post-change pdfs, then the ADD is maximized. Given that our generalized CUSUM detector is designed with the detector sensitivity parameter η such that $\gamma_t \geq \eta, t \geq \tau$, see Eq. (5.15), $\gamma_t = \eta, t \geq \tau$, minimizes the KL divergence between pre- and post-change pdfs, see Eq. (5.14), and hence maximizes the ADD of our detector. Then, we derive the worst-case upper bound on the ADD irrespective of the unknown and time-varying γ_t values by replacing γ in Theorem 5.5 with η, as presented in the Corollary 5.1 below.

Corollary 5.1: The worst-case upper bound on ADD is given by

$$\mathbb{E}_1[\Gamma] \leq \frac{2h + a^* + 0.5 + \psi(a^*, b^*)}{b^* + 0.5},$$

where $a^* \triangleq \eta^2/(2\theta^2)$, and $b^* \triangleq \eta^2/\theta^2$.

To illustrate our results, we assume $\tau = 1$, $y_t \sim \mathcal{N}(0.5 - \gamma, \theta^2)$, $\gamma = 0.1$, $\theta = 0.08$, and $\eta = 0.06$, and we plot in Fig. 5.5 the ADD of the proposed detector as well as the presented upper bounds and the approximation for the ADD, as the test threshold h varies. As we observe through the figure, the Wald's approximation well approximates

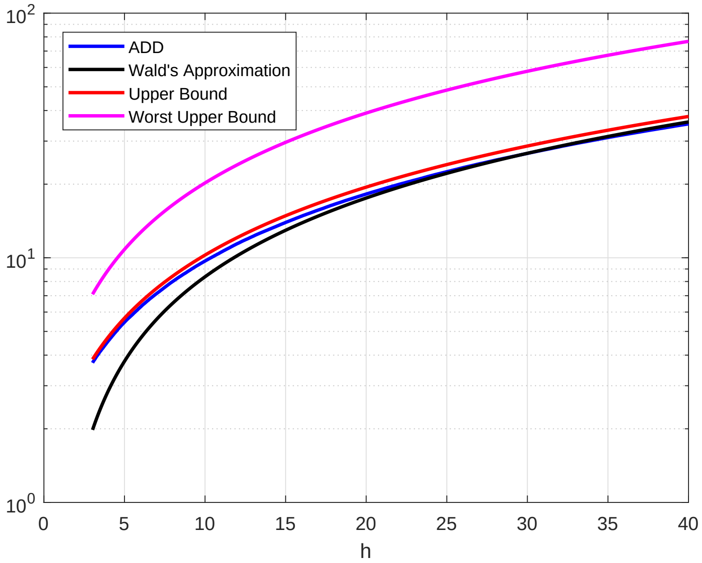


Figure 5.5: The worst-case ADD of the proposed detector, the Wald's approximation to the ADD, the theoretical upper bound on the ADD, and the worst-case upper bound on the ADD for various levels of test threshold h.

the ADD as the excess over the lower boundary 0 does not frequently happen in the post-change regime, unlike the pre-change regime leading to an underestimation of the FAP, as discussed in the previous subsection.

Finally, note that there is tradeoff between the ADD and the FAP and it is controlled by the decision threshold h and the detector sensitivity parameter η. Particularly, increasing h and/or η leads to a larger FAP (lower false alarm rate) but also a larger ADD, and vice versa.

5.4.3 Analytical Privacy-Security Tradeoff

We next show the analytical tradeoff between the DP level and the anomaly detection performance. If the proposed anomaly detection algorithm is employed in security applications such as cyber-attack detection or fraud detection over safety-critical systems, this tradeoff can also be termed as the privacy-security tradeoff. We obtain the

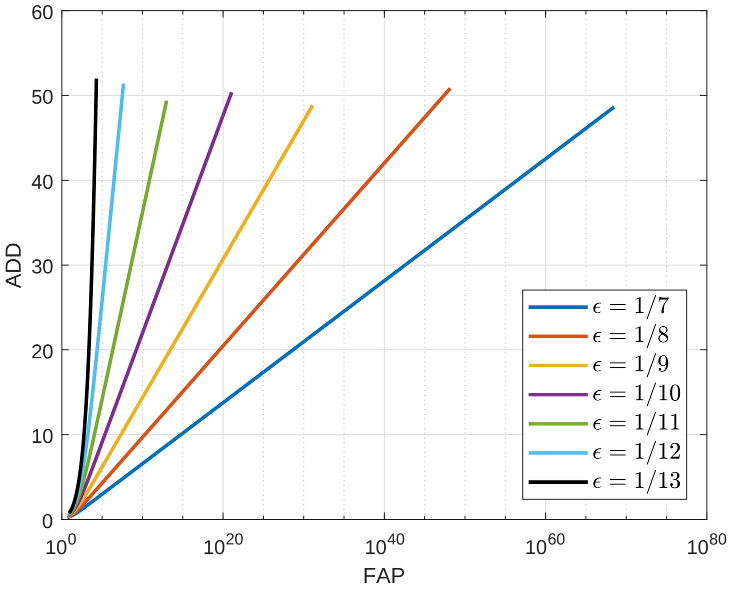


Figure 5.6: The Wald's approximation to the worst-case ADD vs. the Wald's approximation to the FAP of the proposed detector as the test threshold h varies, for various DP levels.

analytical privacy-security tradeoff based on the results derived in Sec. 5.4.2.

Considering a simple example where

$$y_t \sim \begin{cases} \mathcal{N}(0.5, \theta^2), & \text{if } t < \tau, \\ \mathcal{N}(0.5 - \gamma, \theta^2), & \text{if } t \geq \tau, \end{cases}$$

$\gamma = 0.2$, and $\eta = 0.08$, we plot in Fig. 5.6 the Wald's approximations to the ADD and the FAP, as the test threshold h varies and for various DP levels. We assume $N = 300$ and for a linearly spaced set of θ values in the range of $[0.07, 0.13]$, we obtain various DP levels as $\epsilon \approx 1/(100\,\theta)$ assuming $\delta = 0.0139$ and recalling that $\theta = \sigma/\sqrt{N}$, see Eq. (5.18). The figure illustrates that at the same levels of FAP, we obtain larger ADDs as ϵ decreases. That means that the anomaly detection performance degrades for stronger privacy guarantees, clearly showing the privacy-security tradeoff.

5.5 Performance Evaluation

In this section, we evaluate the performance of the proposed anomaly detection scheme against IoT botnet attacks over a real IoT network consisting of nine devices, namely a thermostat, a baby monitor, a webcam, two doorbells, and four security cameras. Particularly, we use the network-based detection of IoT botnet attacks (N-BaIoT) dataset [91] obtained from the UCI Machine Learning Repository [28], where the data dimensionality at each node n is 115, that is, $\boldsymbol{x}_{t,n} \in \mathbb{R}^{115}$. The data represent the network traffic statistics specific to each node, particularly, the number of data packets received and sent, time intervals between packet arrivals, packet sizes, and so forth. The dataset contains both nominal data obtained under nominal conditions and anomalous data obtained under several attack conditions. We use the PCA-based local data processing method (see Alg. 5.1) at each node since the nominal data can be well represented in a linear submanifold. Particularly, we observe that almost all the data variance can be retained in the 5-dimensional principal subspace, that is $r_n = 5$, for the nominal data of each node n.

We consider two attack cases, specifically, the UDP flooding attacks and the spam attacks by the BASHLITE botnet against the IoT network [91]. As the performance metrics, we use the worst-case ADD, $\mathbb{E}_1\big[(\Gamma - \tau)^+\big]$, the FAP, $\mathbb{E}_\infty[\Gamma]$, and the false alarm rate (FAR), which is simply the reciprocal of the FAP:

$$\text{FAR} \triangleq \frac{1}{\mathbb{E}_\infty[\Gamma]}.$$

We present the ADD vs. FAR curves of the proposed detector for various levels of the DP to illustrate the privacy-security tradeoff in this setup. Moreover, we compare the theoretical lower bound and the approximation to the FAP with the actual FAP of the proposed detector obtained with real-world data. Furthermore, we compare the theoretical worst-case upper bound on the ADD with the actual ADD of the proposed detector. Here, note that we compare the ADD only with the worst-case upper bound since in this experiment, $\gamma_t, t \geq \tau$ are unknown and possibly time-varying. Finally, we present the nonparametric sliding-window chi-squared test as a benchmark detector and compare its performance with the proposed CUSUM-based detector. In our experiments, since we set the DP parameter σ^2 to values comparable to 1/12, we use $\theta^2 = (\sigma^2 + 1/12)/N$, that is, we do not ignore the 1/12 term here. This is mainly because of the small network size ($N = 9$) in this setup. Moreover, although the maximum variance of each of the $p_{t,n}$ is 1/4 in the post-change regime as argued before, their actual variance can be much smaller than the maximum value, and hence we consider the same variance (i.e., $\theta^2 = (\sigma^2 + 1/12)/N$) for both the pre- and post-change cases in our experiments.

We choose the detector sensitivity parameter as $\eta = 0.08$. For the reliability of the proposed detector as well as the presented FAP lower bound/approximation to be valid, we need $\rho > 0.61$ (see Theorem 5.2), or equivalently, $0.1311 > \theta$ and

$$0.1311^2 > \frac{\sigma^2 + 1/12}{N},$$

that finally leads to $\sigma^2 < 0.0715$. Hence, we vary σ^2 in this range to obtain several different DP levels. Notice that the range of possible σ^2 values is, in fact, comparable to 1/12.

First, Fig. 5.7 shows the histogram of the average local p-value estimates, that is, $\bar{p}_t$, in the nominal case ($t < \tau$) and the pdf of the assumed nominal model in Eq. (5.9). Fig. 5.7 illustrates that our nominal model can be well justified over the N-BaIoT dataset even with a relatively small network size ($N = 9$) although the central limit theorem holds asymptotically as N gets large.

Next, we plot in Fig. 5.8 the FAP of the proposed network-wide anomaly detection scheme as well as the presented lower bound and the approximation. Here, we set the DP parameter as $\sigma^2 = 1/81$. Then, in case of the spam and the UDP flooding attacks over the network, we present ADD vs. FAR curves for various DP levels in Fig. 5.9 and Fig. 5.10, respectively. Particularly, we choose the local noise variance σ^2 from the set $\{0, 1/81, 1/36, 1/24, 1/16, 1/12, 1/9\}$. Then, assuming $\delta = 0.0139$, we obtain various DP levels as $\epsilon \approx 1/\sigma$, see Eq. (5.18). The figures illustrate that for the same

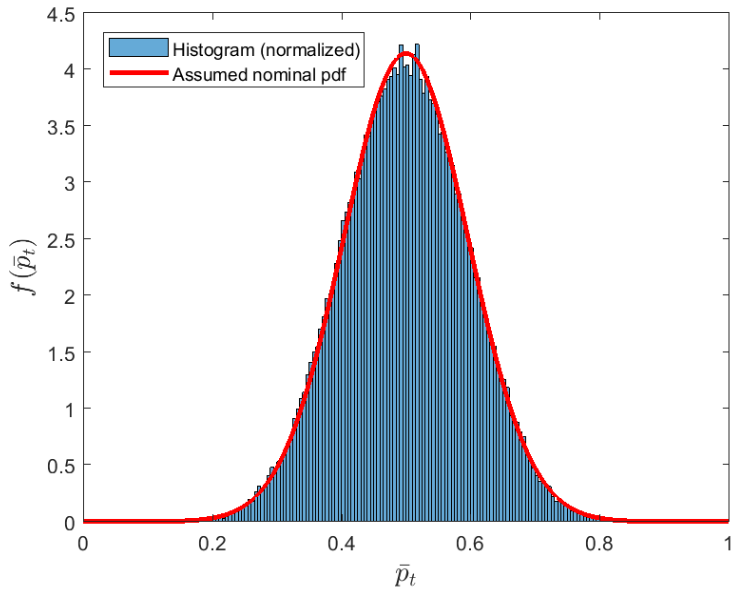


Figure 5.7: Histogram of $\bar{p}_t, t < \tau$, obtained over the N-BaIoT dataset and the pdf of the assumed nominal model in Eq. (5.9).

level of FARs, the ADDs are larger for stronger DP levels (lower ϵ). The implication is that stronger privacy guarantees worsen the anomaly detection performance, in compliance with the theoretical results in Sec. 5.4.3. Furthermore, in case of a spam attack and with the chosen DP parameter $\sigma^2 = 1/16$, we compare in Fig. 5.11 the ADD of the proposed generalized CUSUM detector with its theoretical worst-case upper bound presented in Corollary 5.1, as the test threshold h varies. The figure validates that even if the anomaly signal cannot be specified in this experiment, we are still able to provide a performance guarantee in terms of ADD of the proposed detector.

We next present the sliding-window chi-squared test as a benchmark. For this test, we firstly obtain a nominal statistic that is independent of the system and algorithm parameters so as to use exactly the same procedure regardless of the values of such parameters. By inspecting the pre-change model of y_t given in Eq. (5.10), we can

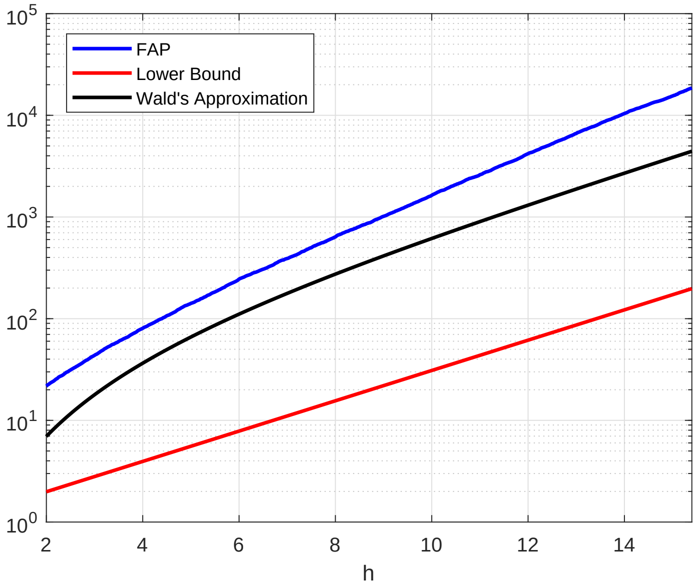


Figure 5.8: The FAP of the proposed network-wide anomaly detection scheme over the N-BaIoT dataset, the analytical lower bound and the approximation for the FAP.

write

$$q_t \triangleq \frac{(y_t - 0.5)^2}{\frac{\sigma^2+1/12}{N}} \sim \chi(1),\ t < \tau, \tag{5.21}$$

where $\chi(1)$ denotes a chi-squared random variable with 1 degree of freedom. Notice that $q_t \sim \chi(1)$ is true irrespective of the network size and the DP parameter. Then, using Eq. (5.21), we can evaluate whether the observed sequence of $\{q_t, t = 1, 2, \dots\}$ fits to its nominal model. For this purpose, goodness-of-fit tests such as the Kolmogorov-Smirnov test and the Anderson-Darling test can be used [124]. We propose to use an online version of the Pearson's chi-squared test, as in [67]. Particularly, we divide the range $[0, \infty)$ of q_t into L disjoint and mutually exclusive intervals $I_1, I_2, \dots, I_L$, and based on the density of $\chi(1)$, we compute the probabilities $p_1 = \mathbb{P}(q_t \in I_1)$, $p_2 = \mathbb{P}(q_t \in I_2)$, $\dots$, $p_L = \mathbb{P}(q_t \in I_L)$, where $\sum_{i=1}^{L} p_i = 1$. Let $\boldsymbol{W}_t \triangleq [q_{t-K+1}, q_{t-K+2}, \dots, q_t]$ be the online sliding window of size K. The expected number of window entries in each interval is then Kp_1, Kp_2, $\dots$, Kp_L, respectively. Hence, we have a multinomial distribution with the expected number of samples in

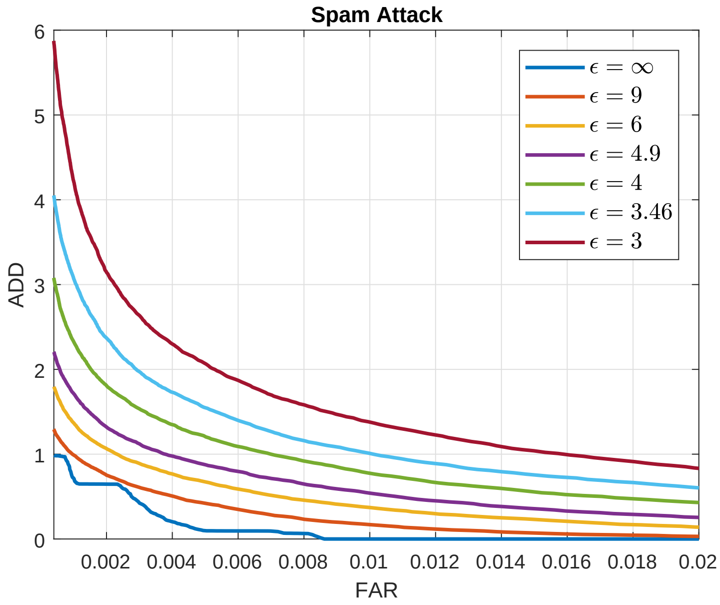


Figure 5.9: ADD vs. FAR of the proposed network-wide anomaly detection scheme in case of a spam attack over the network, for various DP levels.

the disjoint intervals as Kp_1, Kp_2, $\ldots$, Kp_L. For the observed sliding window $\boldsymbol{W}_t$ at time t, we then count how many of its entries reside in each interval. Let the number of entries of $\boldsymbol{W}_t$ residing in each interval be $N_{1,t}$, $N_{2,t}$, $\ldots$, $N_{L,t}$ at time t. The Pearson's chi-squared test is then given as follows:

$$\Gamma = \inf \left\{ t : d_t \triangleq \sum_{i=1}^{L} \frac{(N_{i,t} - Kp_i)^2}{Kp_i} \geq \varphi \right\},$$

where φ is the test threshold that controls the false alarm rate. Here, the decision statistic d_t is asymptotically (as $K \to \infty$) a chi-squared random variable with $L - 1$ degrees of freedom under the null hypothesis (no anomaly). Then, the decision threshold φ can be determined using the cdf of the chi-squared random variable with $L - 1$ degrees of freedom in order to achieve the desired false alarm rate.

Finally, Fig. 5.12 compares the proposed generalized CUSUM detector with the sliding-window chi-squared test in case of a spam attack where the DP parameter is chosen as $\sigma^2 = 1/16$. For the chi-squared test, we choose $L = 8$, $p_1 = p_2 = \cdots =$

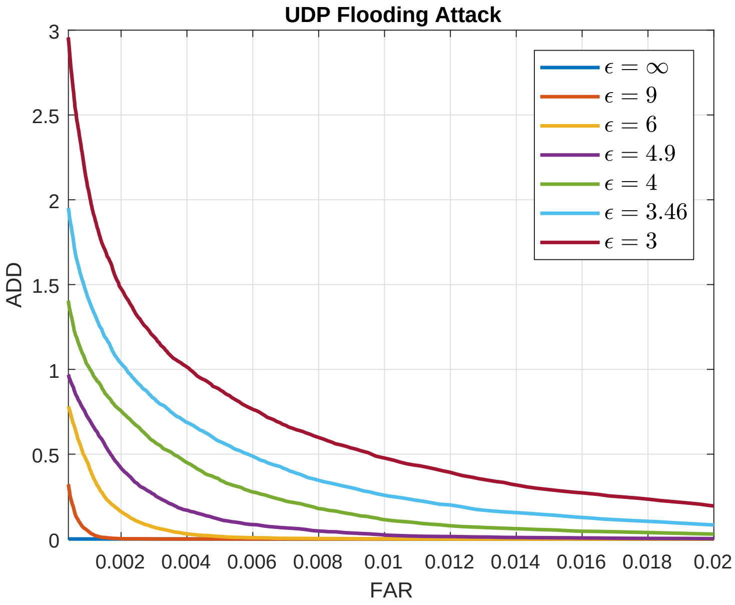


Figure 5.10: ADD vs. FAR of the proposed network-wide anomaly detection scheme in case of a UDP flooding attack over the network, for various DP levels.

$p_8 = 1/8$, and the window size as $K = 96$. Fig. 5.12 illustrates that the proposed detector outperforms the chi-squared test by achieving quicker attack detection (lower ADD) at the same levels of FAR. We obtain the tradeoff curves by varying the test thresholds of the detectors, particularly, h for the CUSUM-based detector and φ for the chi-squared test.

5.6 Concluding Remarks

In this chapter, we have studied the online data-driven anomaly detection over distributed systems while maintaining the privacy of local sensitive data. We have utilized the QCD framework for early and reliable anomaly detection. Moreover, we have combined practical privacy benefits with theoretical privacy guarantees for an effective privacy protection. In particular, our proposed approach consists of a differentially private generalized CUSUM-based distributed detection scheme that in-

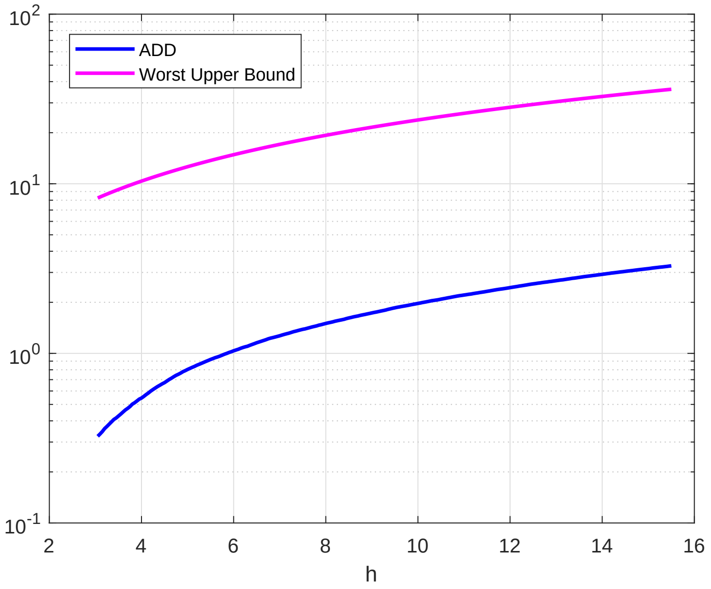


Figure 5.11: The ADD of the proposed network-wide anomaly detection scheme over the N-BaIoT dataset in case of a spam attack and the theoretical worst-case upper bound on the ADD.

fers network-wide anomalies based on the perturbed and encrypted statistics received from nodes. We have analyzed the anomaly detection performance of the proposed scheme in terms of system and algorithm parameters including the differential privacy parameter. In particular, we have derived a lower bound and an approximation for the average false alarm period (FAP) as well as an upper bound and an approximation for the average detection delay (ADD) of the proposed detector. Furthermore, we have used the derived FAP and ADD approximations to illustrate the analytical privacy-security tradeoff in the network-wide anomaly detection problem. Experiments over a real-world IoT dataset support our theoretical findings.

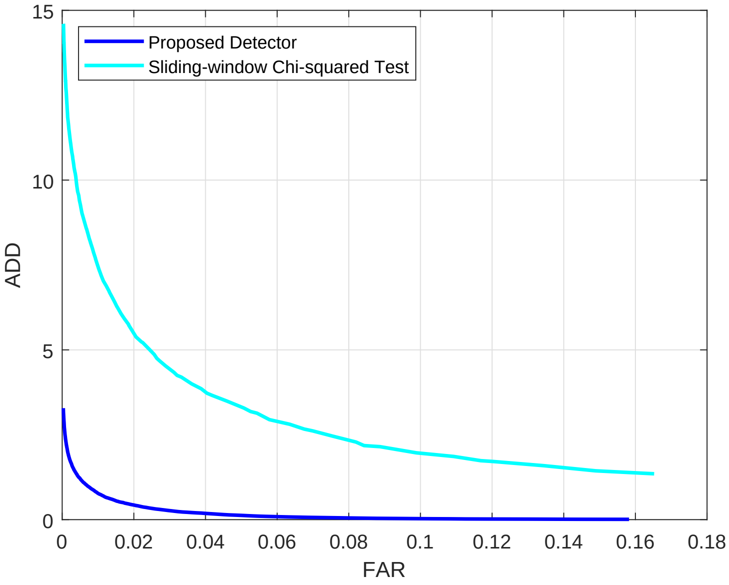


Figure 5.12: ADD vs. FAR of the proposed generalized CUSUM detector and the sliding-window chi-squared test in case of a spam attack over the network.

5.7 Appendix to Chapter 5

5.7.1 Proof of Theorem 5.1

Proof. In the proposed procedure, information released to the decision maker at time t can be written by, see Eq. (5.8),

$$y_t = \frac{1}{N} \sum_{i=1}^{N} p_{t,n} + \bar{v}_t,$$

where $\bar{v}_t \sim \mathcal{N}(0, \sigma^2/N)$ corresponds to the average perturbation noise over nodes. Defining a mean function

$$\phi(\{p_{t,n}\}_n) \triangleq \frac{1}{N} \sum_{i=1}^{N} p_{t,n},$$

we have

$$y_t = \phi(\{p_{t,n}\}_n) + \bar{v}_t.$$

Recalling that the p-value estimate $p_{t,n}$ takes values in the range of $[0, 1]$ irrespective of the nominal or anomalous cases, any single node n can change $\phi(\cdot)$ by at most $1/N$, for example, considering the change from $p_{t,n} = 0$ to $p_{t,n} = 1$. Hence, the sensitivity of the function $\phi(\cdot)$ is

$$\Delta\phi = \frac{1}{N}.$$

Since $\bar{v}_t$ is zero-mean AWGN, we can then use Lemma 5.1 to decide the required noise variance for $\bar{v}_t$ to achieve (ϵ, δ)-DP at time t, as follows:

$$\frac{\sigma^2}{N} = \frac{2\log(1.25/\delta)}{N^2\epsilon^2},$$

and hence

$$\sigma^2 = \frac{2\log(1.25/\delta)}{N\epsilon^2}.$$

If every node perturbs its local output via zero-mean AWGN with the given variance of σ^2 above, we obtain (ϵ, δ)-differentially private aggregation at time t. Moreover, since we process a data stream, at each time t, the newly incoming data $\{p_{t,n}\}_n$ is used, and then never used again. This, in fact, corresponds to a scheme where at each time, a disjoint subset of the dataset is used, considering that the data obtained over all nodes and at all times form our database. The parallel composition rule of the DP [90] states that if ϵ_i-differentially private mechanisms are employed over disjoint subsets of a database, then the overall mechanism achieves $\max_i \epsilon_i$-DP. Then, by invoking the parallel composition property and since at each time t we employ a (ϵ, δ)-differentially private mechanism over a disjoint subset of the entire database, the overall stream aggregation process achieves the (ϵ, δ)-DP.

The decision maker then employs the generalized CUSUM algorithm over the privately aggregated stream of $\{y_t\}_t$. The post-processing invariance rule of the DP [115] states that for an output v of an (ϵ, δ)-differentially private algorithm, any non-private function $\psi(v)$ of the output also achieves (ϵ, δ)-DP, as long as the post-processing does not use the original data. Then, since the generalized CUSUM algorithm can be considered as a non-private function, overall the proposed online anomaly detection scheme is (ϵ, δ)-differentially private.

□

5.7.2 Proof of Theorem 5.2

Proof. For the CUSUM-type detectors in the form of

$$\begin{aligned} \Gamma &= \inf\{t : g_t \geq h\}, \\ g_t &= (g_{t-1} + \beta_t)^+, \end{aligned} \tag{5.22}$$

where $g_0 = 0$, the Wald's approximation to the ARL is given by [5, Sec. 5.2.2]:

$$\mathbb{E}_\tau[\Gamma] \approx \frac{1}{\mathbb{E}[\beta_t]} \left(h + \frac{e^{-w_0 h} - 1}{w_0} \right), \tag{5.23}$$

where the equation

$$\mathbb{E}[e^{-w_0 \beta_t}] = 1 \tag{5.24}$$

has only one nonzero root w_0 such that

$$\begin{cases} w_0 > 0, & \text{if } \mathbb{E}[\beta_t] > 0, \\ w_0 < 0, & \text{if } \mathbb{E}[\beta_t] < 0. \end{cases}$$

The proposed generalized CUSUM detector can be expressed in the form of Eq. (5.22), see Eq. (5.15) and Eq. (5.16). Then, to derive the Wald's approximation for the ARL, we need to compute $\mathbb{E}[\beta_t]$ and also w_0 from Eq. (5.24). To this end, we first compute the pdf of β_t for the pre-change case ($t < \tau$) since in the FAP computations, the assumption is that no change happens at all, that is, $\tau = \infty$.

Let $\mathrm{E}_1 \triangleq \{y_t \leq 0.5 - \eta\}$ and $\mathrm{E}_2 \triangleq \{y_t > 0.5 - \eta\}$ be two complementary events. For $t < \tau$, we have, see Eq. (5.14),

$$\begin{aligned} \mathbb{P}(\mathrm{E}_1) &= \mathbb{P}(y_t - 0.5 \leq -\eta) \\ &= \mathbb{P}\left(\frac{y_t - 0.5}{\theta} \leq \frac{-\eta}{\theta} \right) = Q(\eta/\theta) \end{aligned}$$

and hence $\mathbb{P}(\mathrm{E}_2) = Q(-\eta/\theta)$. Moreover, if E_1 is true, we have, see Eq. (5.17),

$$\begin{aligned}\beta_t &= \frac{1}{2}\left(\frac{y-0.5}{\theta}\right)^2 \\ &\sim \frac{1}{2}\,\chi(1),\end{aligned}$$

where $\chi(1)$ denotes the chi-squared random variable with 1 degrees of freedom. Furthermore, if E_2 is true, we have, see Eq. (5.17),

$$\begin{aligned}\beta_t &= \frac{-\eta}{\theta}\left(\frac{y-0.5}{\theta}\right) - \frac{\eta^2}{2\theta^2} \\ &\sim \mathcal{N}\left(-\frac{\eta^2}{2\theta^2}, \frac{\eta^2}{\theta^2}\right).\end{aligned}$$

In summary, for $t < \tau$, we have

$$\beta_t \sim \begin{cases} \chi(1)/2, & \text{w.p. } Q(\eta/\theta) \\ \mathcal{N}\left(-\frac{\eta^2}{2\theta^2}, \frac{\eta^2}{\theta^2}\right), & \text{w.p. } Q(-\eta/\theta), \end{cases} \tag{5.25}$$

where w.p. denotes “with probability”. Then, using the linearity of the expectation, we can write

$$\mathbb{E}[\beta_t] = \frac{1}{2}Q(\eta/\theta) - \frac{\eta^2}{2\theta^2}Q(-\eta/\theta). \tag{5.26}$$

The Wald's approximation to the FAP requires $\mathbb{E}[\beta_t] < 0$, equivalently, after defining $\rho \triangleq \eta/\theta$, we need, see Eq. (5.26),

$$g(\rho) \triangleq Q(\rho) - \rho^2\, Q(-\rho) < 0.$$

Notice that since the Q-function is monotonically decreasing, the function $g(\rho)$ is monotonically decreasing in ρ and it takes the value of zero when $\rho \approx 0.61$. Hence, for $\rho > 0.61$, we have $\mathbb{E}[\beta_t] < 0$.

Next, we solve Eq. (5.24) to find w_0. Firstly, from Eq. (5.25), under E_1, $\beta_t \sim \chi(1)/2$ and hence the pdf of β_t in this case can be written as follows:

$$f(\beta) = \frac{1}{\sqrt{\pi\beta}}e^{-\beta},\ \beta \geq 0. \tag{5.27}$$

Then, using Eq. (5.25) and Eq. (5.27), we can rewrite Eq. (5.24) as

$$Q(\rho)\underbrace{\int_0^\infty e^{-w_0\beta}\frac{1}{\sqrt{\pi\beta}}e^{-\beta}d\beta}_{A_1}+Q(-\rho)\underbrace{\int_{-\infty}^\infty e^{-w_0\beta}\frac{1}{\sqrt{2\pi\rho^2}}e^{-\frac{1}{2\rho^2}(\beta+0.5\,\rho^2)^2}d\beta}_{A_2}=1, \quad (5.28)$$

where

$$\begin{aligned}A_1 &\triangleq \int_0^\infty e^{-w_0\beta}\frac{1}{\sqrt{\pi\beta}}e^{-\beta}d\beta\\ &= \int_0^\infty \frac{1}{\sqrt{\pi\beta}}e^{-(w_0+1)\beta}d\beta\end{aligned}$$

and letting $x \triangleq (w_0+1)\beta$, we can write

$$\begin{aligned}A_1 &= \frac{1}{\sqrt{w_0+1}}\underbrace{\int_0^\infty \frac{1}{\sqrt{\pi x}}e^{-x}dx}_{1}\\ &= \frac{1}{\sqrt{w_0+1}}, \qquad (5.29)\end{aligned}$$

provided that $w_0+1>0$, or equivalently $w_0>-1$. In the equation above, we use the fact that $\frac{1}{\sqrt{\pi x}}e^{-x}, x\geq 0$ represents a pdf (particularly, the pdf of $\chi(1)/2$).

Furthermore, we have

$$\begin{aligned}A_2 &= \int_{-\infty}^\infty e^{-w_0\beta}\frac{1}{\sqrt{2\pi\rho^2}}e^{-\frac{1}{2\rho^2}(\beta+0.5\,\rho^2)^2}d\beta\\ &= \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi\rho^2}}e^{-\frac{1}{2\rho^2}(\beta^2+2(0.5\,\rho^2+\rho^2w_0)\beta+\rho^4/4)}d\beta\\ &= e^{0.5\,\rho^2(w_0+w_0^2)}\underbrace{\int_{-\infty}^\infty \frac{1}{\sqrt{2\pi\rho^2}}e^{-\frac{1}{2\rho^2}(\beta-(-0.5\,\rho^2-\rho^2w_0))^2}d\beta}_{1}\\ &= e^{0.5\,\rho^2(w_0+w_0^2)}, \qquad (5.30)\end{aligned}$$

where by completing the square, we obtain a Gaussian pdf, whose under area curve is equal to 1.

Hence, we can rewrite Eq. (5.24) based on Eq. (5.28), Eq. (5.29), and Eq. (5.30) as follows:

$$f(w_0) \triangleq Q(\rho)\frac{1}{\sqrt{w_0+1}}+Q(-\rho)\,e^{0.5\,\rho^2(w_0+w_0^2)}=1, \qquad (5.31)$$

where there exists a unique $-1 < w_0 < 0$ solving the equation above.

□

5.7.3 Proof of Theorem 5.3

Proof. For the CUSUM-type detectors given in Eq. (5.22) and if $\mathbb{E}[\beta_t] < 0$, we have the following lower bound on the ARL [5, Sec. 5.2.2]:

$$\mathbb{E}_\infty[\Gamma] \geq e^{-w_0 h},$$

where $w_0 < 0$ is obtained from Eq. (5.24) and hence from Eq. (5.31).

□

5.7.4 Proof of Theorem 5.4

Proof. We can use the Wald's approximation to the ARL given in Eq. (5.23) to derive an approximation for the worst-case ADD of the proposed algorithm provided that $\mathbb{E}[\beta_t] > 0$. For this approximation, similar to Appendix 5.7.2, we need to compute $\mathbb{E}[\beta_t]$ and w_1 (for the ADD calculations, we use w_1 instead of w_0). To this end, we next determine the pdf of β_t for the post-change case, that is, for $t \geq \tau$.

Firstly, using the same event definitions E_1 and E_2 in Appendix 5.7.2 and assuming $\gamma_t = \gamma$ for $t \geq \tau$, we have, see Eq. (5.14),

$$\begin{aligned}\mathbb{P}(\mathrm{E}_1) &= \mathbb{P}(y_t - 0.5 + \gamma \leq \gamma - \eta)\\ &= \mathbb{P}\left(\frac{y_t - 0.5 + \gamma}{\theta} \leq \frac{\gamma - \eta}{\theta}\right) = Q\left(\frac{\eta - \gamma}{\theta}\right)\end{aligned}$$

and hence $\mathbb{P}(\mathrm{E}_2) = Q(\frac{\gamma - \eta}{\theta})$. Moreover, if E_1 is true, we have, see Eq. (5.17),

$$\beta_t = \frac{1}{2}\left(\frac{y - 0.5}{\theta}\right)^2 = \frac{1}{2}x^2,$$

where $x \sim \mathcal{N}(-\gamma/\theta, 1)$. Further, if E_2 is true, we have, see Eq. (5.17),

$$\begin{aligned}\beta_t &= \frac{-(y-0.5)\eta}{\theta^2} - \frac{\eta^2}{2\theta^2} \\ &= \frac{-\eta}{\theta}\left(\frac{y-0.5+\gamma}{\theta}\right) + \frac{2\eta\gamma-\eta^2}{2\theta^2} \\ &\sim \mathcal{N}\left(\frac{2\eta\gamma-\eta^2}{2\theta^2}, \frac{\eta^2}{\theta^2}\right)\end{aligned}$$

In summary, for $t \geq \tau$, we can write

$$\beta_t \sim \begin{cases} \frac{1}{2}\left(\mathcal{N}(-\gamma/\theta,1)\right)^2, & \text{w.p. } Q\left(\frac{\eta-\gamma}{\theta}\right) \\ \mathcal{N}\left(\frac{2\eta\gamma-\eta^2}{2\theta^2}, \frac{\eta^2}{\theta^2}\right), & \text{w.p. } Q\left(\frac{\gamma-\eta}{\theta}\right). \end{cases} \tag{5.32}$$

Then, we have

$$\mathbb{E}[\beta_t] = \frac{\gamma^2+\theta^2}{2\theta^2} Q\left(\frac{\eta-\gamma}{\theta}\right) + \frac{2\eta\gamma-\eta^2}{2\theta^2} Q\left(\frac{\gamma-\eta}{\theta}\right). \tag{5.33}$$

To use the Wald's approximation for the ADD, we need $\mathbb{E}[\beta_t] > 0$, for which a sufficient condition is

$$2\eta\gamma - \eta^2 > 0,$$

equivalently $\gamma > \eta/2$. This is because all the other terms in Eq. (5.33) are nonnegative. Moreover, since $\gamma \geq \eta$ by the definition of the proposed detector, see Eq. (5.15), the sufficient condition is satisfied.

Next, we solve Eq. (5.24) to determine w_1. We write Eq. (5.24) based on Eq. (5.32) as follows:

$$\begin{aligned} Q\left(\frac{\eta-\gamma}{\theta}\right) & \underbrace{\int_{-\infty}^{\infty} e^{-w_1 x^2/2} \frac{1}{\sqrt{2\pi}} e^{-(x+\gamma/\theta)^2/2} dx}_{B_1} \\ & + Q\left(\frac{\gamma-\eta}{\theta}\right) \underbrace{\int_{-\infty}^{\infty} e^{-w_1 \beta} \frac{1}{\sqrt{2\pi\eta^2/\theta^2}} e^{-\frac{1}{2\eta^2/\theta^2}\left(\beta - \frac{2\gamma\eta-\eta^2}{2\theta^2}\right)^2} d\beta}_{B_2} = 1, \end{aligned} \tag{5.34}$$

where for the first term in the summation, we use $\beta = x^2/2$ where $x \sim \mathcal{N}(-\gamma/\theta, 1)$. Then, we have

$$\begin{aligned} B_1 &\triangleq \int_{-\infty}^{\infty} e^{-w_1 x^2/2} \frac{1}{\sqrt{2\pi}} e^{-(x+\gamma/\theta)^2/2} dx \\ &= e^{-\frac{w_1\gamma^2}{2\theta^2(w_1+1)}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\sqrt{w_1+1}\,x + \frac{\gamma}{\theta\sqrt{w_1+1}}\right)^2} dx \\ &= \frac{1}{\sqrt{w_1+1}} e^{-\frac{w_1\gamma^2}{2\theta^2(w_1+1)}} \underbrace{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(y + \frac{\gamma}{\sqrt{w_1+1}}\right)^2} dy}_{1} \\ &= \frac{1}{\sqrt{w_1+1}} e^{\frac{-w_1\gamma^2}{2\theta^2(w_1+1)}}, \end{aligned} \tag{5.35}$$

where we again use the method of completing the square and $y \triangleq \sqrt{w_1+1}\,x$. Further, we have

$$B_2 \triangleq \int_{-\infty}^{\infty} e^{-w_1\beta} \frac{1}{\sqrt{2\pi\eta^2/\theta^2}} e^{-\frac{1}{2\eta^2/\theta^2}\left(\beta - \frac{2\gamma\eta-\eta^2}{2\theta^2}\right)^2} d\beta.$$

We determine B_2 by following the same methodology to find the A_2 in Appendix 5.7.2, that is, completing the square. Then, we obtain

$$B_2 = e^{\frac{(\gamma^2-2\gamma\eta)w_1+\gamma^2 w_1^2}{2\theta^2}} \tag{5.36}$$

Finally, based on Eq. (5.34), Eq. (5.35), and Eq. (5.36), we have

$$g(w_1) \triangleq Q\left(\frac{\eta-\gamma}{\theta}\right) \frac{e^{\frac{-w_1\gamma^2}{2\theta^2(w_1+1)}}}{\sqrt{w_1+1}} + Q\left(\frac{\gamma-\eta}{\theta}\right) e^{\frac{(\gamma^2-2\gamma\eta)w_1+\gamma^2 w_1^2}{2\theta^2}} = 1,$$

where there exists a unique $w_1 > 0$ satisfying the equation.

□

5.7.5 Proof of Theorem 5.5

Proof. For the CUSUM-type detectors given in the general form of Eq. (5.22), if $\mathbb{E}[\beta_t] > 0$ and the observation sequence is normally distributed, we have the following upper bound on the ARL [5, Sec. 5.2.2]:

$$\mathbb{E}_1[\Gamma] \leq \frac{1}{\mathbb{E}[\beta_t]} \left(h + \mathbb{E}[\beta_t | \beta_t > 0]\right). \tag{5.37}$$

Since the proposed detector fits to Eq. (5.22) and the observations y_t are normally distributed at the decision maker, we can derive an upper bound on the worst-case ADD of the proposed detector using Eq. (5.37). To this end, we next compute $\mathbb{E}[\beta_t|\beta_t > 0]$. Notice that $\mathbb{E}[\beta_t]$ is already computed and given in Eq. (5.33).

Firstly, based on Eq. (5.32), under E_1, since $\beta_t \geq 0$ is always true, we can easily compute

$$\mathbb{E}[\beta_t|\beta_t > 0, \mathrm{E}_1] = \frac{\gamma^2 + \theta^2}{2\theta^2}$$

Further, under E_2, we have $\beta_t \sim \mathcal{N}(a, b)$ where $a \triangleq \frac{2\eta\gamma - \eta^2}{2\theta^2}$ and $b \triangleq \frac{\eta^2}{\theta^2}$. Then,

$$\begin{aligned}\mathbb{E}[\beta_t|\beta_t > 0, \mathrm{E}_2] &= \frac{\mathbb{E}[\beta_t, \beta_t > 0|\mathrm{E}_2]}{\mathbb{P}(\beta_t > 0|\mathrm{E}_2)} \\ &= \frac{\mathbb{E}[\beta_t, \beta_t > 0|\mathrm{E}_2]}{Q(-a/\sqrt{b})},\end{aligned}$$

where

$$\begin{aligned}\mathbb{E}[\beta_t, \beta_t > 0|\mathrm{E}_2] &= \int_0^\infty \beta \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2b}(\beta-a)^2} d\beta \\ &= \underbrace{\int_0^\infty (\beta - a) \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2b}(\beta-a)^2} d\beta}_{C_1} + \underbrace{\int_0^\infty a \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2b}(\beta-a)^2} d\beta}_{C_2}\end{aligned}$$

Let $u \triangleq (\beta - a)^2$. Then,

$$\begin{aligned}C_1 &\triangleq \int_0^\infty (\beta - a) \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2b}(\beta-a)^2} d\beta \\ &= \int_{a^2}^\infty \frac{1}{2\sqrt{2\pi b}} e^{-\frac{u}{2b}} du \\ &= \frac{\sqrt{b}}{\sqrt{2\pi}} e^{-\frac{a^2}{2b}}.\end{aligned}$$

Moreover, letting $y \triangleq (x - a)/\sqrt{b}$, we have

$$\begin{aligned}C_2 &\triangleq \int_0^\infty a \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2b}(\beta-a)^2} d\beta \\ &= a \int_{-\frac{a}{\sqrt{b}}}^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy \\ &= a\, Q\left(-\frac{a}{\sqrt{b}}\right)\end{aligned}$$

Then, we obtain the following:

$$\begin{aligned}\mathbb{E}[\beta_t|\beta_t > 0, \mathrm{E}_2] &= \frac{\frac{\sqrt{b}}{\sqrt{2\pi}} e^{-\frac{a^2}{2b}} + a\,Q\left(-\frac{a}{\sqrt{b}}\right)}{Q(-a/\sqrt{b})} \\ &= a + \frac{\sqrt{b}\,e^{-\frac{a^2}{2b}}}{\sqrt{2\pi}\,Q(-a/\sqrt{b})} \\ &\triangleq \psi(a,b)\end{aligned}$$

Finally, we have

$$\begin{aligned}\mathbb{E}[\beta_t|\beta_t > 0] &= \mathbb{P}(\mathrm{E}_1)\,\mathbb{E}[\beta_t|\beta_t > 0, \mathrm{E}_1] + \mathbb{P}(\mathrm{E}_2)\,\mathbb{E}[\beta_t|\beta_t > 0, \mathrm{E}_2] \\ &= Q\left(\frac{\eta - \gamma}{\theta}\right)\frac{\gamma^2 + \theta^2}{2\theta^2} + Q\left(\frac{\gamma - \eta}{\theta}\right)\psi(a,b) \qquad (5.38)\end{aligned}$$

Then, in Eq. (5.37), by replacing $\mathbb{E}[\beta_t]$ and $\mathbb{E}[\beta_t|\beta_t > 0]$ with Eq. (5.33) and Eq. (5.38), respectively, we obtain an upper bound on the ADD.

□

Chapter 6

Conclusions and Future Work

This thesis has studied the data-driven quickest change detection (QCD). Effective and scalable QCD algorithms have been proposed for high-dimensional, complex, and temporally correlated real-world data streams with unknown statistical properties. The solution approaches include novel techniques from sequential analysis, reinforcement learning (RL), deep learning, and distributed privacy-preserving inference. The proposed algorithms have been evaluated through several real-world applications such as online cyber-attack detection over smart grids and online anomaly detection over Internet of Things (IoT) networks and surveillance videos. The following research directions can be considered as the future works:

- The RL-based QCD algorithm proposed in Chapter 2 can be further improved using more advanced methods. In particular, (i) compared to the finite-size sliding window approach, more sophisticated memory techniques can be developed, (ii) compared to discretizing the continuous observation space and using a tabular approach to compute the Q values, function approximation techniques (e.g., neural networks) can be used to compute the Q values, and (iii) deep RL algorithms can be useful to improve the QCD performance.
- In Chapter 2, a single-agent RL setting has been considered with the aim of

optimizing the policy of the defender, where the attacking strategies, such as attack types, magnitudes, set of attacked sensors, and so forth, do not alter the best policy of the defender. That is, once an attack is launched, the defender's best policy is *stop* and declare an attack. Moreover, in the considered setup, the attacker does not learn. As a future work, the current setup can be extended to a multi-agent RL setting where there is an underlying partially observable stochastic game with multiple states in which the payoffs and state transitions depend on joint actions taken by the attacker and the defender. In this setting, both agents aim to learn and adapt to the environment and also to each other's policies in order to maximize their payoffs.

- In Chapter 3, stationary big data streams have been studied. In practice, statistical properties of the observed data streams might also be nonstationary. In such cases, a common approach is assuming a slowly time-varying submanifold underlying the observed data stream [52,139,151]. The results of Chapter 3 can be extended for nonstationary big data streams by employing a subspace tracking algorithm [103] that dynamically estimates the underlying submanifold.

- In Chapter 4, the training dataset is assumed to contain both pre- and post-change samples. As a future work, robust deep QCD algorithms can be investigated where there is only a set of pre-change samples but no samples are available from the post-change, which is generally the case in online attack or anomaly detection problems.

- In Chapter 5, existence of a global decision maker is assumed for network-wide anomaly detection. As a future work, fully-distributed privacy-preserving online anomaly detection can be studied.

Bibliography

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

[2] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

[3] J Yu Angela. Optimal change-detection and spiking neurons. In *Advances in neural information processing systems*, pages 1545–1552, 2007.

[4] Satin Asri and Bernardi Pranggono. Impact of distributed denial-of-service attack on advanced metering infrastructure. *Wireless Personal Communications*, 83(3):2211–2223, 2015.

[5] Michèle Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes: Theory and Application.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[6] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. *Computer*, 50(2):76–79, 2017.

[7] P. Billingsley. *Probability and Measure.* Wiley series in Probability and Mathematical Statistics. Wiley, 1986.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[9] Rakesh B. Bobba, Katherine M. Rogers, Qiyan Wang, Himanshu Khurana, Klara Nahrstedt, and Thomas J. Overbye. Detecting false data injection attacks on dc state estimation. In *Preprints of the First Workshop on Secure Control Systems, CPSWEEK 2010*, 2010.

[10] Loïc Bontemps, James McDermott, Nhien-An Le-Khac, et al. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International Conference on Future Data and Security Engineering*, pages 141–152. Springer, 2016.

[11] Giacomo Boracchi, Diego Carrera, Cristiano Cervellera, and Danilo Macciò. Quanttree: Histograms for change detection in multivariate data streams. In *ICML*, 2018.

[12] Clément L Canonne, Gautam Kamath, Audra McMillan, Adam Smith, and Jonathan Ullman. The structure of optimal private tests for simple hypotheses. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 310–321. ACM, 2019.

[13] Yang Cao, Liyan Xie, Yao Xie, and Huan Xu. Sequential change-point detection via online convex optimization. *Entropy*, 20(2):108, 2018.

[14] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[15] Claude Castelluccia, Aldar CF Chan, Einar Mykletun, and Gene Tsudik. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(3):20, 2009.

[16] Charles W Champ and Steven E Rigdon. A a comparison of the markov chain and the integral equation approaches for evaluating the run length distribution of quality control charts. *Communications in Statistics-Simulation and Computation*, 20(1):191–204, 1991.

[17] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[18] Hao Chen. Sequential change-point detection based on nearest neighbors. *Ann. Statist.*, 47(3):1381–1407, 2019.

[19] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun. Evaluation of reinforcement learning-based false data injection attack to automatic voltage control. *IEEE Transactions on Smart Grid*, 10(2):2158–2169, 2019.

[20] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[21] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998:746–752, 1998.

[22] Rachel Cummings, Sara Krehbiel, Yajun Mei, Rui Tuo, and Wanrong Zhang. Differentially private change-point detection. In *Advances in Neural Information Processing Systems*, pages 10825–10834, 2018.

[23] A. M. L. da Silva, M. B. Do Coutto Filho, and J. M. C. Cantera. An efficient dynamic state estimation algorithm including bad data processing. *IEEE Transactions on Power Systems*, 2(4):1050–1058, Nov 1987.

[24] Tamraparni Dasu, Shankar Krishnan, Suresh Venkatasubramanian, and Ke Yi. An information-theoretic approach to detecting changes in multi-dimensional

data streams. In *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*. Citeseer, 2006.

[25] A. S. Debs and R. E. Larson. A dynamic estimator for tracking the state of a power system. *IEEE Transactions on Power Apparatus and Systems*, PAS-89(7):1670–1678, Sept. 1970.

[26] K. H. Degue and J. L. Ny. On differentially private gaussian hypothesis testing. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 842–847, Oct 2018.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[28] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[29] Adji B. Dieng, Chong Wang, Jianfeng Gao, and John Paisley. TopicRNN: A recurrent neural network with long-range semantic dependency. In *ICLR 2017 : International Conference on Learning Representations 2017*, 2017.

[30] Finale Doshi-Velez, David Wingate, Nicholas Roy, and Joshua Tenenbaum. Nonparametric bayesian policy priors for reinforcement learning. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 532–540, USA, 2010. Curran Associates Inc.

[31] Ricardo Dunia and S Joe Qin. Multi-dimensional fault diagnosis using a subspace approach. In *American Control Conference*, 1997.

[32] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[33] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[34] Zahra Ebrahimzadeh, Min Zheng, Selcuk Karakas, and Samantha Kleinberg. Deep learning for multi-scale changepoint detection in multivariate time series. *arXiv preprint arXiv:1905.06913*, 2019.

[35] D. Eckhoff and C. Sommer. Driving for big data? Privacy concerns in vehicular networking. *IEEE Security Privacy*, 12(1):77–79, Jan 2014.

[36] M. Esmalifalak, H. Nguyen, R. Zheng, and Zhu Han. Stealth false data injection using independent component analysis in smart grid. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 244–248, Oct 2011.

[37] L. Faivishevsky. Information theoretic multivariate change detection for multisensory information processing in Internet of Things. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6250–6254, March 2016.

[38] L. Fan and L. Xiong. Differentially private anomaly detection with a case study on epidemic outbreak detection. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 833–840, Dec 2013.

[39] C. Fuh and A. G. Tartakovsky. Asymptotic bayesian theory of quickest change detection for hidden markov models. *IEEE Transactions on Information Theory*, 65(1):511–529, Jan 2019.

[40] S. V. Georgakopoulos, S. K. Tasoulis, and V. P. Plagianakos. Efficient change detection for high dimensional data streams. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2219–2222, Oct 2015.

[41] S. Gezici, S. Bayram, M. N. Kurt, and M. R. Gholami. Optimal jammer placement in wireless localization systems. *IEEE Transactions on Signal Processing*, 64(17):4534–4549, Sept 2016.

[42] Mohsen Ghassemi, Anand D Sarwate, and Rebecca N Wright. Differentially private online active learning with applications to anomaly detection. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, pages 117–128. ACM, 2016.

[43] Amol Ghoting, Srinivasan Parthasarathy, and Matthew Eric Otey. Fast mining of distance-based outliers in high-dimensional datasets. *Data Mining and Knowledge Discovery*, 16(3):349–364, 2008.

[44] O. Gornerup, N. Dokoohaki, and A. Hess. Privacy-preserving mining of frequent routes in cellular network data. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 581–587, Aug 2015.

[45] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

[46] Blaise Kévin Guépié, Lionel Fillatre, and Igor Nikiforov. Sequential detection of transient changes. *Sequential Analysis*, 31(4):528–547, 2012.

[47] Muneeb Ul Hassan, Mubashir Husain Rehmani, Ramamohanarao Kotagiri, Jiekui Zhang, and Jinjun Chen. Differential privacy for renewable energy resources based smart metering. *Journal of Parallel and Distributed Computing*, 131:69–80, 2019.

[48] Alfred O Hero, III. Geometric entropy minimization (GEM) for anomaly detection and localization. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 585–592, Cambridge, MA, USA, 2006. MIT Press.

[49] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[50] K. Hsiao, K. S. Xu, J. Calder, and A. O. Hero. Multicriteria similarity-based anomaly detection using pareto depth analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1307–1321, June 2016.

[51] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

[52] X. J. Hunt and R. Willett. Online data thinning via multi-subspace tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[53] Tommi Jaakkola, Satinder P. Singh, and Michael I. Jordan. Reinforcement learning algorithm for partially observable markov decision problems. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'94, pages 345–352, Cambridge, MA, USA, 1994. MIT Press.

[54] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1, 2012.

[55] V. Jumutc and J. A. K. Suykens. Multi-class supervised novelty detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2510–2523, Dec 2014.

[56] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[57] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil. An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems. *IEEE Transactions on Sustainable Computing*, pages 1–1, 2019.

[58] Rasul A Khan. Wald's approximations to the average run length in cusum procedures. *Journal of Statistical Planning and Inference*, 2(1):63–77, 1978.

[59] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.

[60] J. Kittler, W. Christmas, T. de Campos, D. Windridge, F. Yan, J. Illingworth, and M. Osman. Domain anomaly detection in machine perception: A system architecture and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):845–859, May 2014.

[61] Mieczyslaw M Kokar and Spiridon A Reveliotis. Reinforcement learning: Architectures and algorithms. *International journal of intelligent systems*, 8(8):875–894, 1993.

[62] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. DDoS in the IoT: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[63] Yun-Long Kong, Qingqing Huang, Chengyi Wang, Jingbo Chen, Jiansheng Chen, and Dongxu He. Long short-term memory neural networks for online disturbance detection in satellite image time series. *Remote Sensing*, 10(3):452, 2018.

[64] M. N. Kurt, O. Ogundijo, C. Li, and X. Wang. Online cyber-attack detection in smart grid: A reinforcement learning approach. *IEEE Transactions on Smart Grid*, 10(5):5174–5185, Sep. 2019.

[65] M. N. Kurt and X. Wang. Multisensor sequential change detection with unknown change propagation pattern. *IEEE Transactions on Aerospace and Electronic Systems*, 55(3):1498–1518, June 2019.

[66] M. N. Kurt, Y. Yilmaz, and X. Wang. Distributed quickest detection of cyber-attacks in smart grid. *IEEE Transactions on Information Forensics and Security*, 13(8):2015–2030, Aug 2018.

[67] M. N. Kurt, Y. Yilmaz, and X. Wang. Real-time detection of hybrid and stealthy cyber-attacks in smart grid. *IEEE Transactions on Information Forensics and Security*, 14(2):498–513, Feb 2019.

[68] M. N. Kurt, Y. Yilmaz, and X. Wang. Real-time nonparametric anomaly detection in high-dimensional settings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[69] M. N. Kurt, Y. Yilmaz, and X. Wang. Secure distributed dynamic state estimation in wide-area smart grids. *IEEE Transactions on Information Forensics and Security*, 15:800–815, 2020.

[70] M. N. Kurt, Y. Yılmaz, and X. Wang. Sequential model-free anomaly detection for big data streams. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 421–425, Sep. 2019.

[71] Tze Leung Lai. Information bounds and quick detection of parameter changes in stochastic systems. *IEEE Transactions on Information Theory*, 44(7):2917–2929, 1998.

[72] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 219–230. ACM, 2004.

[73] Pier Luca Lanzi. Adaptive agents with reinforcement learning and internal memory. In *In*, pages 333–342. MIT Press, 2000.

[74] R. Laxhammar and G. Falkman. Online learning and sequential anomaly detection in trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1158–1173, June 2014.

[75] J. Le Ny and G. J. Pappas. Differentially private filtering. *IEEE Transactions on Automatic Control*, 59(2):341–354, Feb 2014.

[76] Iraklis Leontiadis, Kaoutar Elkhiyaoui, and Refik Molva. Private and dynamic time-series data aggregation with trust relaxation. In *International Conference on Cryptology and Network Security*, pages 305–320. Springer, 2014.

[77] S. Li, Y. Yilmaz, and X. Wang. Quickest detection of false data injection attack in wide-area smart grids. *IEEE Transactions on Smart Grid*, 6(6):2725–2735, Nov 2015.

[78] Tong Li, Jin Li, Zheli Liu, Ping Li, and Chunfu Jia. Differentially private naive bayes learning over multiple data sources. *Information Sciences*, 444:89–104, 2018.

[79] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong. A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid*, 8(4):1630–1638, 2017.

[80] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*, pages 157–163. Elsevier, 1994.

[81] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6536–6545, 2018.

[82] Yao Liu, Peng Ning, and Michael K. Reiter. False data injection attacks against state estimation in electric power grids. In *Proceedings of the 16th ACM Confer-*

ence on Computer and Communications Security, CCS '09, pages 21–32, New York, NY, USA, 2009. ACM.

[83] John Loch and Satinder P. Singh. Using eligibility traces to find the best memoryless policy in partially observable markov decision processes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 323–331, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[84] G. Lorden. Procedures for reacting to a change in distribution. *Ann. Math. Statist.*, 42(6):1897–1908, 1971.

[85] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[86] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89. Presses universitaires de Louvain, 2015.

[87] K. Manandhar, X. Cao, F. Hu, and Y. Liu. Detection of faults and attacks including false data injection attack in smart grid using kalman filter. *IEEE Transactions on Control of Network Systems*, 1(4):370–379, Dec 2014.

[88] Erik Marchi, Fabio Vesperini, Stefano Squartini, and Björn Schuller. Deep recurrent neural network-based autoencoders for acoustic novelty detection. *Computational intelligence and neuroscience*, 2017, 2017.

[89] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[90] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30. ACM, 2009.

[91] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Dominik Breitenbacher, Asaf Shabtai, and Yuval Elovici. N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders. *arXiv preprint arXiv:1805.03409*, 2018.

[92] Nicolas Meuleau, Leonid Peshkin, Kee-Eung Kim, and Leslie Pack Kaelbling. Learning finite-state controllers for partially observable environments. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 427–436, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[93] G. V. Moustakides. Detecting changes in hidden markov models. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 2394–2398, July 2019.

[94] George V. Moustakides. Optimal stopping times for detecting changes in distributions. *Ann. Statist.*, 14(4):1379–1387, 1986.

[95] George V. Moustakides. Sequential change detection revisited. *Ann. Statist.*, 36(2):787–807, 04 2008.

[96] George V Moustakides. Multiple optimality properties of the Shewhart test. *Sequential Analysis*, 33(3):318–344, 2014.

[97] C. Murguia and J. Ruths. Cusum and chi-squared attack detection of compromised sensors. In *2016 IEEE Conference on Control Applications (CCA)*, pages 474–480, Sept 2016.

[98] Ann Nowé, Peter Vrancx, and Yann-Michaël De Hauwere. Game theory and multi-agent reinforcement learning. In *Reinforcement Learning*, pages 441–470. Springer, 2012.

[99] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[100] D. Park, Y. Hoshi, and C. C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, July 2018.

[101] Theodore J. Perkins. Reinforcement learning for POMDPs based on action values and stochastic optimization. In *Eighteenth National Conference on Artificial Intelligence*, pages 199–204, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[102] Leonid Peshkin, Nicolas Meuleau, and Leslie Pack Kaelbling. Learning policies with external memory. *CoRR*, cs.LG/0103003, 2001.

[103] Delmas Jean Pierre. *Subspace Tracking for Signal Processing*, chapter 4, pages 211–270. Wiley-Blackwell, 2010.

[104] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[105] M. Pollak. Optimal detection of a change in distribution. *Ann. Statist.*, 13(1):206–227, Mar. 1985.

[106] Aleksey S. Polunchenko and Vasanthan Raghavan. Comparative performance analysis of the cumulative sum chart and the Shiryaev-Roberts procedure for detecting changes in autocorrelated data. *Applied Stochastic Models in Business and Industry*, 34(6):922–948, 2018.

[107] Aleksey S. Polunchenko and Alexander G. Tartakovsky. State-of-the-art in sequential change-point detection. *Methodology and Computing in Applied Probability*, 14(3):649–684, Sep 2012.

[108] H. V. Poor and O. Hadjiliadis. *Quickest Detection*. Cambridge University Press, 2008.

[109] Abdulhakim A Qahtan, Basma Alharbi, Suojin Wang, and Xiangliang Zhang. A PCA-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 935–944, 2015.

[110] G. Ratsch, S. Mika, B. Scholkopf, and K. . Muller. Constructing boosting algorithms from SVMs: an application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1184–1199, Sept 2002.

[111] D. B. Rawat and C. Bajracharya. Detection of false data injection attacks in smart grid communication systems. *IEEE Signal Processing Letters*, 22(10):1652–1656, Oct 2015.

[112] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016.

[113] Marion R Reynolds. Approximations to the average run length in cumulative sum control charts. *Technometrics*, 17(1):65–71, 1975.

[114] Stéphane Ross, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. A bayesian approach for learning and planning in partially observable markov decision processes. *J. Mach. Learn. Res.*, 12:1729–1770, July 2011.

[115] A. D. Sarwate and K. Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE Signal Processing Magazine*, 30(5):86–94, Sep. 2013.

[116] F. Sattler, S. Wiedemann, K. Müller, and W. Samek. Robust and communication-efficient federated learning from non-i.i.d. data. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[117] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[118] Elaine Shi, HTH Chan, Eleanor Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *Annual Network & Distributed System Security Symposium (NDSS)*. Citeseer, 2011.

[119] A. N. Shiryaev. *Optimal Stopping Rules*. Springer-Verlag, NY, 1978.

[120] N. R. Shivakumar and A. Jain. A review of power system dynamic state estimation techniques. In *2008 Joint International Conference on Power System Technology and IEEE Power India Conference*, pages 1–6, Oct 2008.

[121] S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248, Dec 2013.

[122] Kumar Srichanran and Alfred O. Hero. Efficient anomaly detection using bipartite k-NN graphs. In *Advances in Neural Information Processing Systems*, pages 478–486, 2011.

[123] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[124] Michael A Stephens. Edf statistics for goodness of fit and some comparisons. *Journal of the American statistical Association*, 69(347):730–737, 1974.

[125] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2018.

[126] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[127] S. Tan, D. De, W. Z. Song, J. Yang, and S. K. Das. Survey of security advances in smart grid: A data driven approach. *IEEE Communications Surveys Tutorials*, 19(1):397–422, Firstquarter 2017.

[128] A. G. Tartakovsky and V. V. Veeravalli. General asymptotic bayesian theory of quickest change detection. *Theory of Probability & Its Applications*, 49(3):458–497, 2005.

[129] A. L. Toledo and X. Wang. Robust detection of selfish misbehavior in wireless networks. *IEEE Journal on Selected Areas in Communications*, 25(6):1124–1134, August 2007.

[130] Samet Tonyali, Kemal Akkaya, Nico Saputro, A Selcuk Uluagac, and Mehrdad Nojoumian. Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled smart metering systems. *Future Generation Computer Systems*, 78:547–557, 2018.

[131] J. Unnikrishnan, V. V. Veeravalli, and S. P. Meyn. Minimax robust quickest change detection. *IEEE Transactions on Information Theory*, 57(3):1604–1614, March 2011.

[132] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge University Press, 1998.

[133] Venugopal V. Veeravalli and Taposh Banerjee. Chapter 6 - Quickest Change Detection. In Rama Chellappa Abdelhak M. Zoubir, Mats Viberg and Sergios Theodoridis, editors, *Academic Press Library in Signal Processing: Volume 3 Array and Statistical Signal Processing*, volume 3 of *Academic Press Library in Signal Processing*, pages 209 – 255. Elsevier, 2014.

[134] Q. Wang, S. R. Kulkarni, and S. Verdu. A nearest-neighbor approach to estimating divergence between continuous random vectors. In *2006 IEEE International Symposium on Information Theory*, pages 242–246, July 2006.

[135] Wenye Wang and Zhuo Lu. Cyber security in the smart grid: Survey and challenges. *Computer Networks*, 57(5):1344–1371, 2013.

[136] Michael Weinberg and Jeffrey S Rosenschein. Best-response multiagent learning in non-stationary environments. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 506–513. IEEE Computer Society, 2004.

[137] M. Wu and J. Ye. A small sphere and large margin approach for novelty detection using training data with outliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2088–2092, Nov 2009.

[138] L. Xie, Y. Mo, and B. Sinopoli. False data injection attacks in electricity markets. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 226–231, Oct 2010.

[139] Yao Xie, Jiaji Huang, and Rebecca Willett. Change-point detection for high-dimensional time series with missing data. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):12–27, 2013.

[140] J. Yan, H. He, X. Zhong, and Y. Tang. Q-learning-based vulnerability analysis of smart grid against sequential topology attacks. *IEEE Transactions on Information Forensics and Security*, 12(1):200–210, Jan 2017.

[141] Ye Yan, Yi Qian, Hamid Sharif, and David Tipper. A survey on cyber security for smart grid communications. *IEEE Communications Surveys & Tutorials*, 2012.

[142] Qingyu Yang, Liguo Chang, and Wei Yu. On false data injection attacks against kalman filtering in power system dynamic state estimation. *Security and Communication Networks*, 9(9):833–849, 2016.

[143] Y. Yilmaz. Online nonparametric anomaly detection based on geometric entropy minimization. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 3010–3014, June 2017.

[144] C. Yin, Y. Zhu, J. Fei, and X. He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017.

[145] D. Zambon, C. Alippi, and L. Livi. Concept drift and anomaly detection in graph streams. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5592–5605, Nov 2018.

[146] T. Zhang and Q. Zhu. Distributed privacy-preserving collaborative intrusion detection systems for vanets. *IEEE Transactions on Signal and Information Processing over Networks*, 4(1):148–161, March 2018.

[147] Yichi Zhang, Lingfeng Wang, Weiqing Sun, RC Green, and Mansoor Alam. Distributed intrusion detection system in a multi-layer network architecture of smart grids. *Smart Grid, IEEE Transactions on*, 2(4):796–808, 2011.

[148] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-an Tan. Secure multi-party computation: Theory, practice and applications. *Information Sciences*, 476:357–372, 2019.

[149] Manqi Zhao and Venkatesh Saligrama. Anomaly detection with score functions based on nearest neighbor graphs. In *Advances in neural information processing systems*, pages 2250–2258, 2009.

[150] R.D. Zimmerman, C.E. Murillo-Sanchez, and R.J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, Feb 2011.

[151] Ralf Zimmermann, Benjamin Peherstorfer, and Karen Willcox. Geometric subspace updates with applications to online adaptive nonlinear model reduction. *SIAM Journal on Matrix Analysis and Applications*, 39(1):234–261, 2018.