Online Distributed IoT Security Monitoring With Multidimensional Streaming Big Data

Fangyu Li, Rui Xie, Zengyan Wang, Lulu Guo, Jin Ye, Ping Ma, and Wenzhan Song

***Abstract*—Internet of Things (IoT) enables extensive connections between cyber and physical "things." Nevertheless, the streaming data among IoT sensors bring "big data" issues, for example, large data volumes, data redundancy, lack of scalability, and so on. Under big data circumstances, IoT system monitoring becomes a challenge. Furthermore, cyberattacks which threaten IoT security are hard to be detected. In this article, we propose an online distributed IoT security monitoring algorithm (ODIS). An advanced influential point selection operation extracts important information from multidimensional time-series data across distributed sensor nodes based on the spatial and temporal data dependence structure. Then, an accurate data structure model is constructed to capture the IoT system behaviors. Next, hypothesis testing is carried out to quantify the uncertainty of the monitoring tasks. Besides, the distributed system architecture solves the scalability issue. Using a real sensor network testbed, we commit cyberattacks to an IoT system with different patterns and strengths. The proposed ODIS algorithm demonstrates promising detection and monitoring performances.**

***Index Terms*—Big data, distributed, Internet of Things (IoT) security, online.**

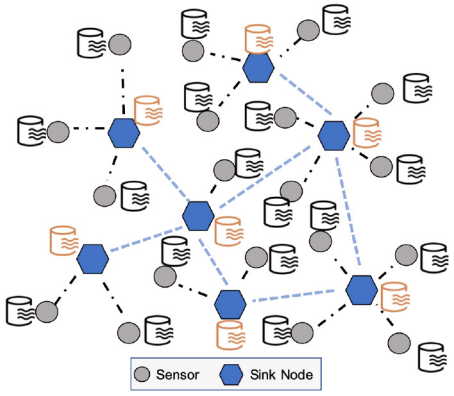

Fig. 1. Multidimensional streaming "big data" from IoT systems. In a WSN, there are sensor nodes (gray dots) and sink nodes (blue hexagons). Besides sensing data (black cylinders), sink nodes also process data and exchange information among sink nodes (orange cylinders).

I. Introduction

EXPLOSION of the Internet of Things (IoT) has been witnessed in diversified fields [1]. The number of IoT devices, as well as the generated data at different layers, are exponentially increasing. As reported, there will be more than 50 billion terminal devices worldwide, and the annual data generated will reach 847 Zebytes by 2021 [2]. "Big data" hereby becomes common in IoT applications [3], such as industrial manufacturing [4], smart cities [5], energy Internet [6], wireless sensor network (WSN) [7], etc. However, because the IoT paradigm enables various connections between cyber networks and physical devices, vulnerabilities become an important issue [8]. Data-driven-based IoT security solutions have been proposed, such as neural networks and deep-learning-based methods [9]. Nevertheless, the big data nature of IoT applications generates furthermore challenges, including the vast volume which will continuously grow [10], modeling complexity caused by large-scale processes [11], high-speed ubiquitous data collection [12], data redundancy introduced by multidimensional data collected asynchronously across distributed nodes [13], algorithm scalability [14], and so on. Thus, it is necessary to develop IoT security monitoring techniques in the big data era.

Our motivation is to effectively detect the IoT system anomalies caused by cyberattacks under the big data circumstances, especially in WSN where multidimensional streaming data are gathered from networked sensors in a high speed [15], as shown in Fig. 1. The important anomaly detection and diagnosis information for IoT monitoring are typically buried in the system metrics, such as energy consumption [8] and system resource usages [9]. Thus, extracting useful information from data, especially, unlabeled samples, is extremely important [16]. To fight against the data redundancy, finding the informative samples is highly desired for accelerating the computation and transmission processes of the high-speed streaming data. To effectively and efficiently extract informative samples, influential point selection (IPS) can be viewed as a data extraction approach to reduce the

Manuscript received September 23, 2019; revised December 16, 2019 and December 23, 2019; accepted December 25, 2019. Date of publication December 27, 2019; date of current version May 12, 2020. This work was supported in part by NSF under Grant 1663709, Grant DMS-1222718, Grant DMS-1903226, Grant DMS-1925066, and Grant ECCS-1946057, in part by NIH under Grant R01GM113242 and Grant R01GM122080, and in part by Southern Company. *(Corresponding author: Rui Xie.)*

Fangyu Li, Lulu Guo, Jin Ye, and Wenzhan Song are with the Center for Cyber-Physical Systems, University of Georgia, Athens, GA 30602 USA (e-mail: fangyu.li@uga.edu; lulu.guo@uga.edu; jin.ye@uga.edu; wsong@uga.edu).

Rui Xie is with the Department of Statistics and Data Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: rui.xie@ucf.edu).

Zengyan Wang is with the Department of Computer Science, University of Georgia, Athens, GA 30602 USA (e-mail: zengyan@cs.uga.edu).

Ping Ma is with the Department of Statistics, University of Georgia, Athens, GA 30602 USA (e-mail: pingma@uga.edu).

Digital Object Identifier 10.1109/JIOT.2019.2962788

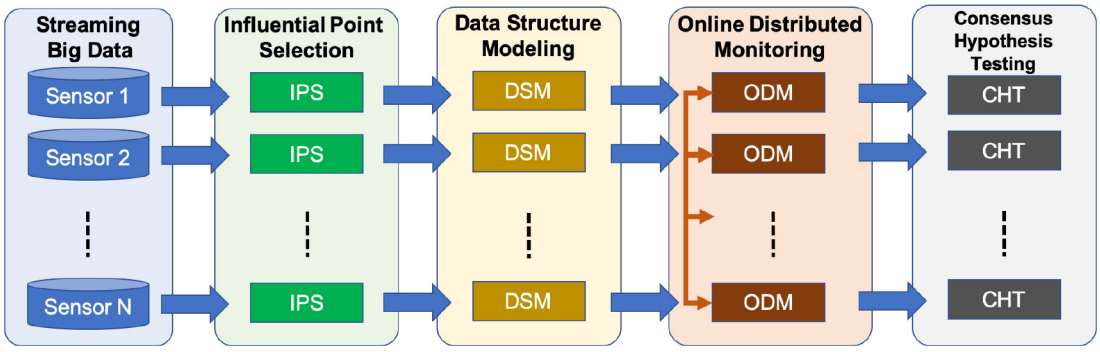

Fig. 2. Proposed ODIS algorithm with streaming big data. The detailed method descriptions and algorithms can be found in Section III.

unnecessary energy consumption in the IoT devices caused by redundant computations and system memory usages [17], [18]. Randomized data selection methods yield a high accuracy on model parameter estimation [17].

Based on the extracted influential points, how to understand the dynamic temporal and spatial/cross-sectional dependence structure of multidimensional streaming time series is still a challenge. The vector autoregressive (VAR) model, the most popular and fundamental time series models, provides a mechanism for capturing complex latent multidimensional dependency structures [13], [19], [20]. Since it is impossible to model the various unknown cyberattacks [8], we propose to model the normal system behaviors as an alternative solution. Thus, the system performance modeling accuracy is of great importance, and the accurate and robust data dependency modeling can facilitate this purpose.

In addition, the IoT system essentially has a distributed architecture [21], where each sensor node only observes partial local information (a smaller set of relevant variables and features are analyzed locally [22]), but in together forms the analysis of the whole network. Note that a distributed manner has additive benefits, such as more secure and enhanced robustness, since the attack behaviors are isolated. Thus, the IoT security monitoring procedure should be designed in a distributed manner and the computation tasks can be assigned to the individual nodes.

In this article, we propose an online distributed IoT security monitoring algorithm (ODIS) for multidimensional streaming big time-series data. The whole workflow is shown in Fig. 2. Based on the efficient streaming time-series data dynamic structure extraction through IPS, we can model the IoT data economically and accurately. In addition, thanks to the online distributed design, ODIS is suitable for the real-time large-scale multidimensional streaming IoT security monitoring. Finally, a complete algorithm is proposed so that the whole IoT system can have an end-to-end security solution. The contributions of this article can be summarized as follows.

1) We propose a novel online IoT security monitoring algorithm under the big data circumstances.
2) Data science techniques, such as IPS and streaming data modeling, are proposed to extract intrinsic data structures efficiently and effectively. The IoT application data are collected across time and space, so the proposed approach considers and models the spatio-temporal dependencies.
3) The proposed ODIS algorithm is scalabe and can be applied to real-time large-scale sensor network applications because of the online distributed streaming processing algorithm design.

The remainder of this article is organized as follows. Related works are introduced in Section II. In Section III, we describe the proposed ODIS algorithm in detail with the related theoretical principles of every important operation. Using a real IoT testbed in Section IV, we analyze the performances of our proposed algorithm explicitly with comprehensive and quantitative analysis. In the end, a conclusion section is enclosed in Section V.

II. Related Work

IoT vulnerabilities arise due to the connected cyber-physical infrastructure [23], [24]. To eliminate the IoT security threats, there is a high demand for solutions with a real time, scalable, and distributed monitoring infrastructure [13]. Thus, the previous resilient approaches, such as simple signal analytics-based [25], Kalman filter [26], generalized-likelihood ratio [27], the cumulative sum (CUSUM) [28], leverage score [29], Bayesian calibration [30], and machine learning [31], which need a center to monitor and control the entire system, are not suitable to be a distributed IoT security framework. In contrast, the required system should be distributed, where sensors only coordinate with its own neighbors within limited distances, and the multidimensional data could be collected asynchronously across distributed nodes [13].

In addition, typical data-driven approaches also suffer the lack of precise failure models in the device level as well as the system (network) level. For example, a resilient strategy was proposed to dispatch virtual power plant under cyber attacks in [32], where lower and upper bounds of the controller states are estimated in a distributed way. However, in the consensus-based energy management algorithm, false states, even minor ones within the given bounds, could cause large deviations [33], resulting in an ineffective resilient strategy.

III. Methodology and Algorithm

In this section, we briefly describe the principles of the ODIS algorithm. Every key step is introduced and the whole algorithm is summarized as well.

A. Symbol and Notation

The uppercase letters $\mathbf{A}$ and A are used for matrices and operators, and the curly capital letter $\mathcal{A}$ is for set or collection of sets. The vector is denoted by the lowercase bold letter $\mathbf{a}$, and the scalar is denoted by the lowercase letter a. We write the transpose of a matrix $\mathbf{A}$ as $\mathbf{A}'$, the determinant of a matrix $\mathbf{A}$ as $\det(\mathbf{A})$, and the matrix vectorization as $\text{vec}(\cdot)$. Specifically, $\mathbb{E}(\cdot)$ denotes the expectation, $\triangleq$ denotes *equal by definition*, $\|\cdot\|_2$ denotes the ℓ_2-norm for a vector, and $\|\cdot\|_F$ denotes the Frobenius norm for a matrix. Moreover, $\mathbb{Z}$ denotes the integers, $\mathbb{R}$ denotes the real numbers, $\mathbf{I}_n$ denote the identity matrix of dimension n, and $1_{\{\cdot\}}$ denotes the indicator function.

B. Multidimensional Time Series Modeling

The multidimensional time series modeling can effectively extract the temporal-dependent information from the streaming multidimensional data, which is the key to understand and monitor the status of the IoT system.

The VAR family, as the most important family of the time series models, is used to reveal the complex dependence structure in the streaming time-series data [19]. The VAR model class quantifies complex temporal and cross-sectional interrelationship among the multidimensional time series. At the same time, the VAR model is flexible enough to be easily integrated into the distributed IoT system [13], providing treatment for the big data scenario.

The K-dimensional VAR model of order p (VAR(p)) of K-dimensional streaming data $\mathbf{y}_t$ can be written as

$$\mathbf{y}_t = \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \boldsymbol{\varepsilon}_t \tag{1}$$

where $t \in \mathbb{Z}$, Φ_i's are the $K \times K$ model coefficient matrices, and $\boldsymbol{\varepsilon}_t$ is a sequence of independent and identically distributed (i.i.d.) random vectors with mean zero and finite nonsingular covariance matrix. The VAR(p) model in (1) encodes the temporal and cross-sectional dependence structure between sensors in the IoT system through the coefficient matrices Φ_i's, which are the key to understand the structural information from the streaming data. Learning these coefficient matrices can be done through ordinary least-squares (OLSs) estimate [19], [20]. However, twofold challenges need to be conquered for the streaming big data setting. First, for a given fixed-time period T, the computational cost of estimating the model coefficient matrices is $O(TK^2p^2)$. It will pose a computational challenge for the entire IoT system when the number of observed data T is huge or the dimension of the data K is high. Second, for streaming data analysis, an online algorithm is needed so that we can achieve the real-time monitoring of the IoT system. In the following sections, we will address these two issues.

C. Big Data Influential Point Selection

With the growing scale of the IoT streaming data, the huge volume of data challenges the computational and storage limits of the IoT system. For streaming the IoT data monitoring, selecting the influential points reduces the processing time, energy consumption, and thus be an effective road to the big data challenges. Data sketching and subsampling are the popular tools to reduce the size of the data, with applications in online streaming analysis [13].

Extend the linear VAR (p) model in (1) to the form of general streaming nonlinear model

$$\mathbf{y}_t' = f\left(\mathbf{y}_{t-1}', \mathbf{y}_{t-2}', \ldots, \mathbf{y}_{t-p}'\right)' \mathbf{B} + \varepsilon_t' \tag{2}$$

for observation up to time t, where $\mathbf{B} = [\Phi_1', \Phi_2', \ldots, \Phi_p']'$ is the $Kp \times K$ model coefficient matrix.

For a given function $f(\cdot)$, we denote $\mathbf{x}_t = f(\mathbf{y}_{t-1}', \mathbf{y}_{t-2}', \ldots, \mathbf{y}_{t-p}')'$, a column vector of length Kp. The model coefficient matrix can then be estimated as

$$\hat{\mathbf{B}}_{OLS} = \left(\sum_t \mathbf{x}_t \mathbf{x}_t'\right)^{-1} \sum_t \mathbf{x}_t \mathbf{y}_t'. \tag{3}$$

We value the importance of a data point $\mathbf{y}_t$ through its predicted value $\hat{\mathbf{y}}_t = h_{tt}\mathbf{y}_t$, where

$$h_{tt} = \mathbf{x}_t' \left(\sum_t \mathbf{x}_t \mathbf{x}_t'\right)^{-1} \mathbf{x}_t \tag{4}$$

is the tth diagonal element of Hat Matrix, denoting the Mahalanobis distance of the tth data [17], [34].

To effectively reduce the data size while maintaining the underlying data features, we select a subset $\mathcal{S}$ from time domain $\{1, \ldots, T\}$, and use the subset data $\{\mathbf{y}_t | t \in \mathcal{S}\}$ to efficiently estimate the model coefficient matrix. The least square estimator [35] then becomes

$$\hat{\mathbf{B}}_S = \left(\sum_{t\in\mathcal{S}} \mathbf{x}_t \mathbf{x}_t'\right)^{-1} \left(\sum_{t\in\mathcal{S}} \mathbf{x}_t \mathbf{y}_t'\right). \tag{5}$$

If the subset size $|\mathcal{S}|$ is much less than the data size T, i.e., $|\mathcal{S}| << T$, the IPS will greatly save the computational time and cost to $O(|\mathcal{S}|K^2p^2)$.

The IPS is defined according to the selection rule

$$\mathcal{S}_{\text{IPS}} = \left\{t \in \{1, \ldots, T\} : h_{tt} > r^2\right\} \tag{6}$$

where r is the selection threshold. The selection probability distribution follows the chi-squared distribution with degrees of freedom Kp, χ^2_{Kp}. Thus, the selection threshold r is chosen as a square root of the quantile of χ^2_{Kp} distribution, that is

$$P(t \in \mathcal{S}_{\text{IPS}}) = P\left(\chi^2_{Kp} > r^2\right) \tag{7}$$

where the selection threshold r is approximately proportional to the selection ratio, i.e., $r \propto |\mathcal{S}_{\text{IPS}}|/T$. The theoretical justification of the choice of r can be found in [13]. Alternatively, IPS can be described as, for data $\mathbf{y}_t$ observed at time t, if the Mahalanobis distance $\sqrt{h_{tt}} > r$, then we decide the data point $\mathbf{y}_t$ as the influential point and include t in subset $\mathcal{S}_{\text{IPS}}$. Fig. 3 visualizes the geometric property and corresponding Mahalanobis distance of the IPS procedure, where the data points outside the ellipse are selected as the influential points in the subset $\mathcal{S}_{\text{IPS}}$. IPS can be widely used to construct the important sampling in big data analytics to reduce the data size, and it can also be applied in regression diagnostics to

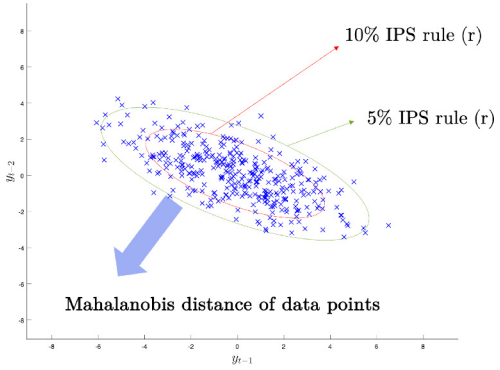

Fig. 3. IPS illustration: 1-D data y_t are plotted with axes lag-2 values y_{t-2} versus lag-1 values y_{t-1}. IPS selection rule r is proportional to selection ratio, i.e., $r \propto |\mathcal{S}_{\text{IPS}}|/T$. The Mahalanobis distances larger than the ellipses (red: 10%; green: 5%) will be selected as the influential points. The influential points only account for a small amount of the whole data set, e.g., 5% or 10%, but they represent the data structure.

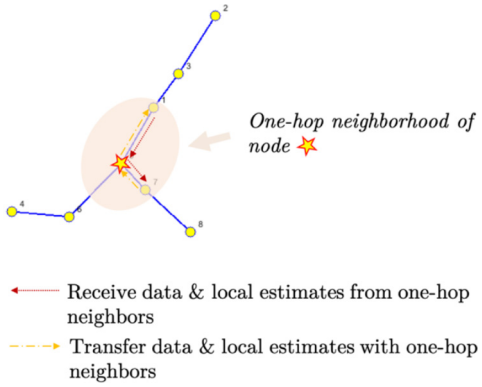

Fig. 4. Diffusion strategy of the distributed network. At every time t, node k collects a measurement $y_t^{(k)}$ and neighborhood data.

identify the outliers and the influential observations (see [36] and [37]).

D. Online Streaming IPS and Time Series Modeling

In the practical calculation of the streaming IPS, we need to tackle the computational bottleneck of the Mahalanobis distance in (4), the inverse of sample covariance matrix $\sum_t \mathbf{x}_t\mathbf{x}_t'$ and update it as the new data arriving. The benefit of calculating the Mahalanobis distance is that it provides the unitless and scale-invariant measurement to the influence of the data and takes into account the correlations of the data. Such bottleneck makes the IPS computation as expensive as solving the original least-squares problem in (3) in a streaming setting.

We propose an online streaming IPS adapting the single-pass streaming algorithm. As the new data arriving, we collect the first batch of data points as pilot sample to calculate a robust estimation on the sample covariance matrix $\Omega = (\sum_{t_0} \mathbf{x}_{t_0}\mathbf{x}_{t_0}')^{-1}$ with time range t_0 taking from the pilot sample batch. Then, for $t > t_0$, the streaming IPS and the corresponding selection rule are replaced as

$$\tilde{h}_{tt} \triangleq \mathbf{x}_t'\Omega\mathbf{x}_t > r^2, \quad \text{for } t > t_0. \tag{8}$$

Since $\mathbf{x}_t$ is constructed based on the VAR model, the streaming IPS procedure is a single-pass procedure that only requires linear computational time $O(Kp)$ with respect to the VAR model dimension. It makes the streaming IPS scalable in the big data setting. Online streaming time series modeling also calls for an online real-time method to periodically aggregate historical information from the previous data, updating the current estimation on the model coefficient matrix based on the arriving new data. More specifically, when the streaming data keep arriving sequentially, we update the estimate of the model coefficient matrix $\mathbf{B}$ adaptively.

The streaming IPS makes online real-time decision on selecting influential points $\mathcal{S}_{\text{IPS}}$ and IoT system monitoring. In other words, for each time stamp t and selected influential points, the online VAR modeling and optimization with respect to model matrix $\mathbf{B}$ become,

$$\mathbf{B}_t = \arg\min_{\mathbf{B}} \sum_{i \in \mathcal{S}_{\text{IPS}} \cap \{i \le t\}} \left\| \mathbf{y}_i' - \mathbf{x}_i'\mathbf{B} \right\|_2^2 \tag{9}$$

where $\mathbf{x}_i = f(\mathbf{y}_{i-1}', \mathbf{y}_{i-2}', \ldots, \mathbf{y}_{i-p}')'$. Note that the estimation of $\mathbf{B}_t$ in (9) is an online optimization in a standard linear form. It can be solved by various optimization algorithms, including the Kalman filter [38], recursive least squares [39], and gradient descent [40]. Our streaming IPS and time series modeling are independent from the choice of the solver to the optimization in (9). As long as the solver satisfies the one-pass property in online optimization and has the computational complexity linear in time t, the IPS monitoring and time series modeling will be scalable for the streaming big data setting.

E. Distributed Online Monitoring

Distributed computing infrastructure is intrinsic and necessary to the IoT tasks. Each sensor is observing a 1-D streaming data, and all sensors together form a network with a certain topological structure. The sensor network is observing the multidimensional streaming time series based on the topological structure. Such structure leads to a distributed computing environment. The streaming IPS and VAR modeling can be integrated into the asynchronous distributed computing environment. Fig. 4 illustrates the one-hop neighborhood diffusion strategy.

The streaming IPS defined in (8) and selection rule defined in (6) can be implemented on each marginal dimension asynchronously and independently under the distributed setting. The selection rule $\mathcal{S}_{lev}^{(k)}$ for node $k \in \{1, \ldots, K\}$ becomes

$$\tilde{h}_{\tau_k}^{(k)} = \mathbf{x}_{\tau_k}'\Omega\mathbf{x}_{\tau_k} > r^2 \tag{10}$$

where $\mathbf{x}_{\tau_k} = f(\mathbf{y}_{\tau_k-1}', \mathbf{y}_{\tau_k-2}', \ldots, \mathbf{y}_{\tau_k-p}')'$ is the kth marginal local copy of the streaming data at local time τ_k.

Given the assumption that the multidimensional streaming data arrive sequentially in communication restricted distributed streaming environment, we exploit the VAR model structure so that the VAR modeling in (9) can be decomposed to K subproblems. For simplicity, we assume the function $f(\cdot)$ as the linear form. We express the model coefficient matrix $\mathbf{B}$ as

Algorithm 1 Online Distributed Streaming IPS Monitoring

Input: From pilot sample batch: Ω , r, and initial values of $\mathbf{B}_0$ and $\mathbf{P}_0$.
Output: Model coefficient matrix estimate $\mathbf{B}_{\tau_k}$

while $t > 0$ **do**
  while node $k \in [1, \ldots, K]$ **do**
    Receive the local data $y_t^{(k)}$, and the one-hop neighborhood data
    Transmit the local data $y_t^{(k)}$ to one-hop neighbors
    Wait until $\mathbf{x}_{\tau_k}$ is complete for some $\tau_k \leq t$
    if $\tilde{h}_{\tau_k}^{(k)} > r^2$ **then**
      Update $\boldsymbol{b}_{\tau_k}^{(k)}$ and $\mathbf{P}_{\tau_k}$ according to recursive least squares according to Eq. (13) and Eq. (14)
    else
      $\boldsymbol{b}_{\tau_k}^{(k)} = \boldsymbol{b}_{\tau_k-1}^{(k)}$ **and** $\mathbf{P}_{\tau_k} = \mathbf{P}_{\tau_k-1}$
    end if
    Exchange the one-hop local estimate $\boldsymbol{b}^{(k)}$
    $\tau_k \leftarrow \tau_k + 1$
    return $\mathbf{B}_{\tau_k} = [\boldsymbol{b}_{\tau_k}^{(1)}, \ldots, \boldsymbol{b}_{\tau_k}^{(k)}, \ldots, \boldsymbol{b}_{\tau_k}^{(K)}]$
  end while node
end while t

a block matrix with column vectors

$$\mathbf{B} = \left[\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \ldots, \boldsymbol{b}^{(K)}\right] \tag{11}$$

with $\boldsymbol{b}^{(k)}$ being the kth column of $\mathbf{B}$ for $k \in \{1, \ldots, K\}$. For node k at local time τ_k, the kth subproblem becomes

$$\boldsymbol{b}_{\tau_k}^{(k)} = \arg\min_{\boldsymbol{b}^{(k)}} \sum_{\tau_k \in \mathcal{S}_{\text{IPS}}^{(k)} \cap \{1,\ldots,t\}} \left\| y_{\tau_k}^{(k)} - \mathbf{x}_{\tau_k}' \boldsymbol{b}^{(k)} \right\|_2^2 \tag{12}$$

where $y_{\tau_k}^{(k)}$ is the kth element of $\mathbf{y}_{\tau_j}$ at local time τ_k, for $k \in \{1, \ldots, K\}$. The estimation of $\boldsymbol{b}_{\tau_k}^{(k)}$ can be completed once all components of $\mathbf{x}_{\tau_k}$ is observed at local time τ_k. The estimation of different nodes $k \neq k'$ is calculated uncoordinately, which leads us the asynchronous algorithm in the IoT system.

Various distributed consensus optimization algorithms can be used to solve the subproblems (12), including distributed gradient descent [40], distributed ADMM [41], and distributed Kalman filter [38]. The framework of the distributed recursive least squares [39] is adapted to solve the distributed problem in (12) as an illustration. When data from its one-hop neighbors sequentially arrived with some delay (see Fig. 4), the local recursive least squares is to estimate the local model coefficient $\boldsymbol{b}_{\tau_k}^{(k)}$ for the kth node and local time $\tau_k \in \mathcal{S}_{\text{IPS}}$

$$\boldsymbol{b}_{\tau_k}^{(k)} = \boldsymbol{b}_{\tau_k-1}^{(k)} + \left[y_{\tau_k}^{(k)} - \mathbf{x}_{\tau_k}' \boldsymbol{b}_{\tau_k-1}^{(k)} \right] \mathbf{k}_{\tau_k} \tag{13}$$

where

$$\mathbf{P}_{\tau_k} = \mathbf{P}_{\tau_k-1} - \mathbf{k}_{\tau_k} \mathbf{x}_{\tau_k}' \mathbf{P}_{\tau_k-1} \tag{14}$$

$\mathbf{k}_{\tau_k} \triangleq \gamma_{\tau_k}^{-1} \mathbf{P}_{\tau_k-1} \mathbf{x}_{\tau_k}$, and $\gamma_{\tau_k} \triangleq 1 + \mathbf{x}_{\tau_k}' \mathbf{P}_{\tau_k-1} \mathbf{x}_{\tau_k}$ with $\mathbf{P}_{\tau_k}$ as the kth local estimate of the precision matrix. By transmitting the local estimation $\boldsymbol{b}_{\tau_k}^{(k)}$ to its neighborhood, each node will form a complete model coefficient matrix estimate $\mathbf{B}_{\tau_k}$, at time τ_k, by combining these column vectors according to (11). We summarize the algorithm in Algorithm 1.

Algorithm 2 Whole ODIS Algorithm

Input: K-dimensional Big Data Streaming in IoT System
Output: System monitoring, attack status decision.

while time $t > 0$ **do**
  Streaming IPS using Eq. (6) - Eq. (8) to reduce the data size and select influential points $\mathcal{S}_{IPS}$;
  Extract the model matrix B_t at time $t \in \mathcal{S}_{IPS}$ according to model Eq. (9);
  Distributed modeling according to Eq. (11) - Eq.(12);
  Structural weight update and online monitoring following Eq. (13) - Eq. (14);
  Hypothesis testing for attack status quantification based on $\mathbf{B}_t$. If null hypothesis H_0 is rejected, the attack status is detected.
end while t

F. Consensus Hypothesis Testing

For the monitoring purpose, we develop the statistical hypothesis testing strategy to provide the real-time status and uncertainty quantification based on the distributed monitoring results $\mathbf{B}_t$. In Algorithm 1, each node has been fused the consensus monitoring results $\mathbf{B}_t$ based on the diffusion strategy, see [38], which means that every node will have the same monitoring results $\mathbf{B}_t$ when t is large enough. The purpose of hypothesis testing is to distinguish the attack status from the normal status in a quantitative way. Based on the VAR modeling, we construct the Wald type statistics for hypothesis testing with null hypothesis $H_0 : \mathbf{B}_t = \mathbf{B}_{\text{Normal}}$ against alternative hypothesis $H_1 : \mathbf{B}_t \neq \mathbf{B}_{\text{Normal}}$, where we ignore the superscript (k) since the consensus results of $\mathbf{B}_t$. The Wald statistic has asymptotic χ^2 distribution with ρ degrees of freedom, where ρ is the rank of a matrix $\mathbf{B}_t$ [20]. Then, we provide a hypothesis testing with a p-value that quantifies the uncertainty of online attack monitoring states. If we reject the null hypothesis based on the p-value, the current data suggest that there is a significant pattern change to make the system deviates from its normal status. After the distributed consensus hypothesis testing, the system can obtain a unified decision.

G. Proposed ODIS Algorithm

We realize the ODIS monitoring through the mentioned key steps. While Fig. 2 shows the big picture of the whole workflow, Algorithm 2 shows the detailed comprehensive workflow of the proposed ODIS algorithm using the connections with key theories.

IV. Experiments and Evaluation

To evaluate the ODIS algorithm, we carry out cyberattack experiments using a real IoT system, where smart sensors are connected within a wireless network. Different cyber attack strengths are tested to demonstrate the performances of ODIS in cyber attack detection and monitoring.

A. Testbed Setup

We use the beaglebone black boards (BBBs) in our experiments[1] to implement a real IoT system consisting of wireless

[1]The hardware details are available at http://beagleboard.org/black (last access: June 25, 2019).

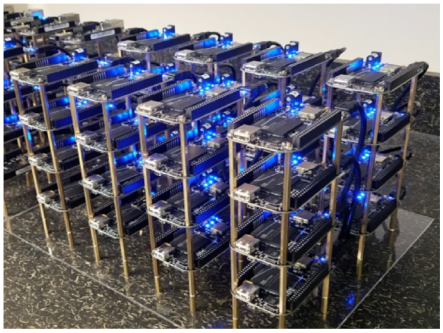

Fig. 5. Real IoT device testbed built by BBBs connected via a wireless network.

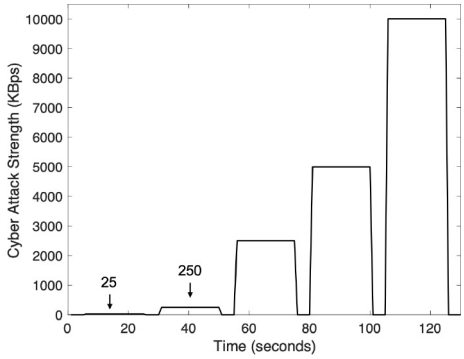

Fig. 6. DoS attack pattern in experiment 1.

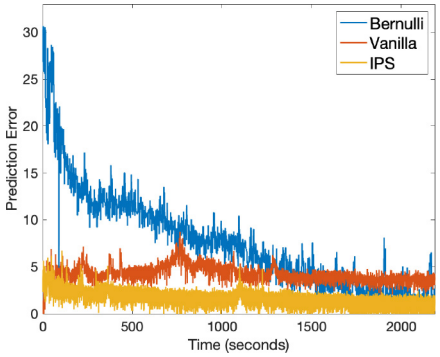

Fig. 7. Modeling errors using different methods in experiment 1.

network connected smart sensors (embedded system). Fig. 5 shows the testbed in our study. Note that there are 36 available BBBs in the same cluster sharing the same mesh network, where the distributed algorithms can be operated among nodes.

B. Cyberattack Detection and Monitoring

Denial-of-Service (DoS) Cyberattack: The IoT sensor networks are generally vulnerable to intrusion related to snooping, spoofing, masquerading, and DoS attacks. The DoS attacks impact the network communication partially or completely. As the sensor nodes of IoT are low powered and lossy, the impact of DoS attack is quite significant [42]. The DoS attack disrupts the communication between devices, making their unavailability. The DoS attack can be carried out externally or internally in IoT and is very hard to detect it unless the services have stopped working. For example, in the flooding attack, the attacker overflows the network through sending packets to disrupt the service of legitimate users. Its examples include domain name system (DNS) flood, Internet control message protocol (ICMP) flood, and user datagram protocol (UDP) flood. There are four common types of DoS attacks, volumetric, network transport, application, and multivector. As volumetric attacks are the most common DoS attacks, we simulate volumetric DoS attacks in our experiments, which consume available network bandwidth between the target and the Internet by overwhelming the target with a flood of data.

Data: To monitor the WSN, we adopt the energy consumption measurement [8] of every node, which represents the whole system activities of the node. The total energy consumption consists of energy consumption from individual subcomponents, such as CPU, RAM, storage, data transmission, etc. In addition, cyberattacks result in the abnormal system behaviors [9] that can be observed from the energy consumption of the subcomponents. For example, the DoS attacks significantly increase the amount of received data, resulting in the energy consumption of not only the network adapter but also the whole system increases. If the proposed ODIS algorithm can detect anomalies and monitor the attack variations based on the energy consumption auditing, the IoT monitoring system is fully functional.

Evaluation: We compare the VAR modeling accuracy using IPS with Vanilla and Bernoulli sampling methods. The Vanilla method uses all data points without data point selection for monitoring, which may result in delayed responses for the IoT system. In Bernoulli sampling, a Bernoulli trial is conducted to randomly select data points at each time with a fixed success probability ($p = 1/2$) [13].

Experiment 1 (DoS Detection): Fig. 6 shows the relative weak cyberattacks. The attack strengths vary from 25 kb/s to 10 mb/s. Every time, DoS attack happens 20 s then there is a 5 s interval. The time-series data contain 36 dimensions ($K = 36$), and each dimension has around 24 000 samples (23 985) with data interval 0.1 s.

As defined in (9), the VAR model should be accurately characterized using the streaming data. We compare the modeling performances in Fig. 7 using different sampling methods: IPS, Vanilla, and Bernoulli. Due to the sampling strategy used by the Bernoulli sampling, the modeling error is large. The Vanilla and IPS sampling methods generate relatively small modeling errors. Furthermore, due to the existence of inevitable noises in the real testbed, the modeling results from Vanilla could be influenced by the noise, whereas IPS has a better robustness as only informative points are extracted and used.

The estimated coefficient matrix Φ_1 under the DoS attacks 25 kb/s and 250 kb/s are shown in Fig. 8. It is observable that there are minor off-diagonal unusual patterns indicating that the IoT system is under attack even the attacks are not strong.

Experiment 2 (DoS Monitoring): Fig. 9 shows the relative strong cyber attacks. The attack strengths vary from 10 to 160 Mb/s. The time-series data contain 36 dimensions ($K =$

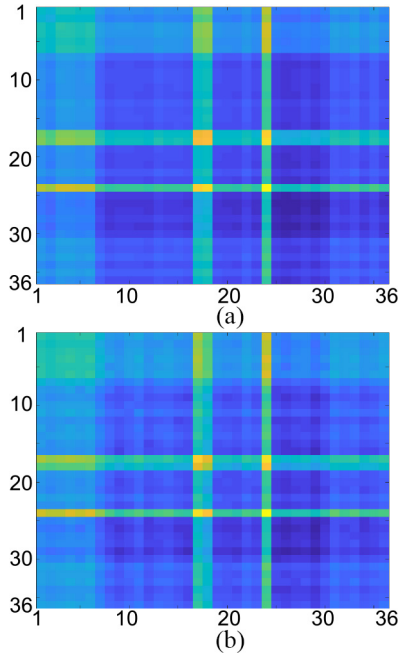

Fig. 8. Estimated parameter matrices Φ_1 under DoS attacks (a) 25 kb/s and (b) 250 kb/s.

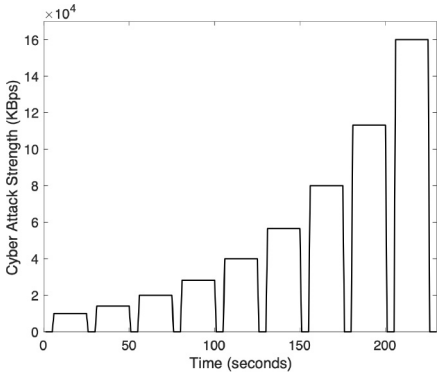

Fig. 9. DoS attack pattern in experiment 2.

36, and each dimension has around 24 000 samples (23 872) with data interval 0.1 s.

Fig. 10 shows the modeling accuracy errors using different sampling methods: IPS, Vanilla, and Bernoulli. Similar to Fig. 7, the Bernoulli sampling method generates larger errors than the other two methods. IPS is still the best approach. Note that because of the hardware design and system limitations, when the cyberattacks are strong, not only the network card performance is affected, the whole system does not behave normally. Then, there are more interference and noise mixed in the modeling process, so the modeling performances of Vanilla and IPS are not as good as those in experiment 1.

According to the consensus hypothesis testing in Section III-F, we use the Wald test [20] to monitor the streaming data structure variations. p-value is employed to reject the null hypothesis. We observe that p is close to 1 for the same attack strength, and when there is a system status change p value is small, for example, when the system changes from normal to under attack, p value can be as small

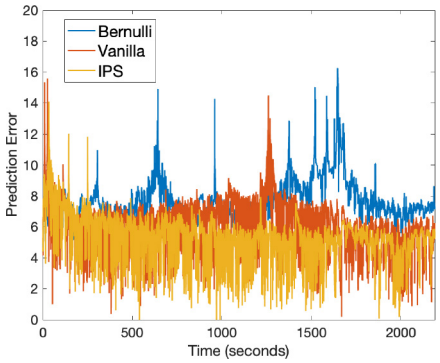

Fig. 10. Modeling errors using different methods in experiment 2.

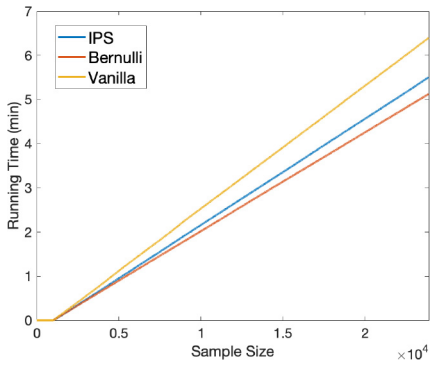

Fig. 11. Average elapsed time of IPS (blue), Bernoulli (red), and Vanilla (orange) in experiment 1.

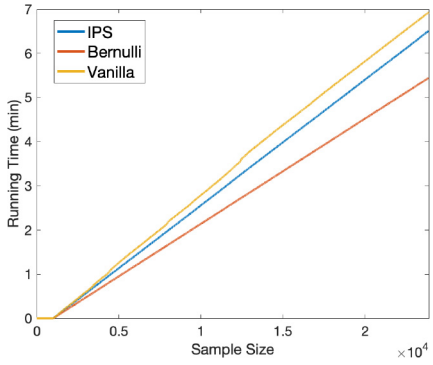

Fig. 12. Average elapsed time of IPS (blue), Bernoulli (red), and Vanilla (orange) in experiment 2.

as 0.0001, and when the DoS attack is strong, we observe p value's unit could be 10^{-7}.

1) Computation Efficiency: Besides the modeling error characterization, we also compare the computational efficiency of the mentioned methods. Since the data amounts are close in experiments 1 and 2, the total computation time consumption of two experiments does not vary significantly. It is clear that IPS is more efficient than Vanilla in both Figs. 11 and 12, as with less data the computation could be faster. However, because the modeling computation, storage, and data transmission also take time, even longer time, the time saving is limited. Nevertheless, the proposed approach is promising, as we can notice that more time saved for larger data, which is the very target for the big data processing. Compared with the Bernoulli sampling, IPS has additional computations, so it is

slower, but IPS can have a slightly better modeling accuracy even compared with the Vanilla method. Thus, the proposed approach is the most promising.

V. Conclusion

To deal with the big data issues in IoT security, we proposed an ODIS monitoring algorithm. The proposed algorithm handles the complex streaming multidimensional time series very well. The latent streaming data dependency on both time and space can be effectively and efficiently extracted from the multidimensional big data. In addition, the online distributed algorithm design enables the real-time IoT monitoring with affordable computation and communication burdens. Using the testbed with real IoT devices, we carry out experiments about cyberattacks (e.g., DoS) toward the IoT sensor networks. The proposed algorithm is a general IoT cybersecurity solution and shows promising performances in terms of cyberattack detection and monitoring.

References

[1] M. Marjani *et al.*, "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[2] *Cisco Global Cloud Index: Forecast and Methodology, 2015–2020. White Paper*, Cisco, San Jose, CA, USA, 2016.

[3] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, 2014.

[4] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and F. Bu, "An adaptive dropout deep computation model for industrial IoT big data learning with crowdsourcing to cloud computing," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2330–2337, Apr. 2019.

[5] M. V. Moreno *et al.*, "Applicability of big data techniques to smart cities deployments," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 800–809, Apr. 2017.

[6] K. Wang, H. Li, Y. Feng, and G. Tian, "Big data analytics for system stability evaluation strategy in the energy Internet," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1969–1978, Aug. 2017.

[7] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT urban scenarios," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3558–3567, Oct. 2013.

[8] F. Li, Y. Shi, A. Shinde, J. Ye, and W. Z. Song, "Enhanced cyber-physical security in Internet of Things through energy auditing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5224–5231, Jun. 2019.

[9] F. Li, A. Shinde, Y. Shi, J. Ye, X.-Y. Li, and W.-Z. Song, "System statistics learning-based IoT security: Feasibility and suitability," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6396–6403, Aug. 2019.

[10] K. Wang, Y. Shao, L. Shu, G. Han, and C. Zhu, "LDPA: A local data processing architecture in ambient assisted living communications," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 56–63, Jan. 2015.

[11] J. Zhu, Z. Ge, and Z. Song, "Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1877–1885, Aug. 2017.

[12] Z. Lv, H. Song, P. Basanta-Val, A. Steed, and M. Jo, "Next-generation big data analytics: State of the art, challenges, and future research topics," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1891–1899, Aug. 2017.

[13] R. Xie, Z. Wang, S. Bai, P. Ma, and W. Zhong, "Online decentralized leverage score sampling for streaming multidimensional time series," in *Proc. Mach. Learn. Res.*, vol. 89, 2019, pp. 2301–2311.

[14] M. Gharbieh, H. ElSawy, A. Bader, and M.-S. Alouini, "Spatiotemporal stochastic modeling of IoT enabled cellular networks: Scalability and stability analysis," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3585–3600, Aug. 2017.

[15] S. Rani, S. H. Ahmed, R. Talwar, and J. Malhotra, "Can sensors collect big data? An energy-efficient big data gathering algorithm for a WSN," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1961–1968, Aug. 2017.

[16] R. Iqbal, F. Doctor, B. More, S. Mahmud, and U. Yousuf, "Big data analytics: Computational intelligence techniques and application areas," *Technol. Forecast. Soc. Change*, Apr. 2018, Art. no. 119253. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0040162517318498?via%3Dihub

[17] P. Ma, M. W. Mahoney, and B. Yu, "A statistical perspective on algorithmic leveraging," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 861–911, 2015. [Online]. Available: http://jmlr.org/papers/v16/ma15a.html

[18] M. B. Cohen, C. Musco, and C. Musco, "Input sparsity time low-rank approximation via ridge leverage score sampling," in *Proc. 28th Annu. ACM-SIAM Symp. Discr. Algorithms*, 2017, pp. 1758–1777.

[19] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.

[20] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. New York, NY, USA: Springer, 2005.

[21] M. Valero, F. Li, J. Clemente, and W. Song, "Distributed and communication-efficient spatial auto-correlation subsurface imaging in sensor networks," *Sensors*, vol. 19, no. 11, p. 2427, 2019.

[22] Z.-K. Gao, Y.-X. Yang, P.-C. Fang, Y. Zou, C.-Y. Xia, and M. Du, "Multiscale complex network for analyzing experimental multivariate time series," *Europhys. Lett.*, vol. 109, no. 3, 2015, Art. no. 30005.

[23] W.-L. Chin, W. Li, and H.-H. Chen, "Energy big data security threats in IoT-based smart grid communications," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 70–75, Oct. 2017.

[24] B. Yang, L. Guo, F. Li, J. Ye, and W.-Z. Song, "Vulnerability assessments of electric drive systems due to sensor data integrity attacks," *IEEE Trans. Ind. Informat.*, to be published.

[25] B. Yang, F. Li, J. Ye, and W. Song, "Condition monitoring and fault diagnosis of generators in power networks," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2019, pp. 1–5.

[26] S. Dusmez, H. Duran, and B. Akin, "Remaining useful lifetime estimation for thermally stressed power MOSFETs based on on-state resistance variation," *IEEE Trans. Ind. Appl.*, vol. 52, no. 3, pp. 2554–2563, May/Jun. 2016.

[27] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011.

[28] J. Giraldo, A. Cárdenas, and N. Quijano, "Integrity attacks on real-time pricing in smart grids: Impact and countermeasures," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2249–2257, Sep. 2017.

[29] F. Li *et al.*, "Detection and identification of cyber and physical attacks on distribution power grids with PVS: An online high-dimensional data-driven approach," *IEEE J. Emerg. Sel. Topics Power Electron.*, to be published.

[30] M. Heydarzadeh, S. Dusmez, M. Nourani, and B. Akin, "Bayesian remaining useful lifetime prediction of thermally aged power MOSFETs," in *Proc. IEEE Appl. Power Electron. Conf. Expo. (APEC)*, 2017, pp. 2718–2722.

[31] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1644–1652, Sep. 2017.

[32] Y. Liu, H. Xin, Z. Qu, and D. Gan, "An attack-resilient cooperative control strategy of multiple distributed generators in distribution networks," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2923–2932, Nov. 2016.

[33] J. Duan and M.-Y. Chow, "A novel data integrity attack on consensus-based distributed energy management algorithm using local information," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1544–1553, Mar. 2019.

[34] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, "Fast approximation of matrix coherence and statistical leverage," *J. Mach. Learn. Res.*, vol. 13, pp. 3475–3506, Dec. 2012.

[35] J. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1994. [Online]. Available: https://books.google.com/books?id=B8_1UBmqVUoC

[36] D. C. Hoaglin and R. E. Welsch, "The hat matrix in regression and ANOVA," *Amer. Stat.*, vol. 32, no. 1, pp. 17–22, 1978.

[37] S. Chatterjee and A. S. Hadi, "Influential observations, high leverage points, and outliers in linear regression," *Stat. Sci.*, vol. 1, no. 3, pp. 379–393, 1986.

[38] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[39] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: Stability and performance analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3740–3754, Jul. 2012.

[40] S. Zheng *et al.*, "Asynchronous stochastic gradient descent with delay compensation," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 4120–4129.

[41] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1701–1709.

[42] S. Naik and N. Shekokar, "Conservation of energy in wireless sensor network by preventing denial of sleep attack," *Procedia Comput. Sci.*, vol. 45, pp. 370–379, Mar. 2015.