Distributed Real-time Anomaly Detection in Networked Industrial Sensing Systems

***Abstract*—Reliable real-time sensing plays a vital role in ensuring the reliability and safety of industrial Cyber-Physical Systems (CPSs) such as wireless sensor and actuator networks. For many reasons, such as harsh industrial environments, fault-prone sensors, or malicious attacks, sensor readings may be abnormal or faulty. This could lead to serious system performance degradation or even catastrophic failure. Current anomaly detection approaches are either centralized and complicated, or restricted due to strict assumptions, which are not suitable for practical large-scale Networked Industrial Sensing Systems (NISSs) where sensing devices are connected via digital communications, such as wireless sensor networks or smart grid systems. In this paper, we introduce a fully distributed general-anomaly-detection (GAD) scheme, which uses graph theory and exploits spatiotemporal correlations of physical processes to carry out real-time anomaly detection for general large-scale NISSs. We formally prove the scalability of our GAD approach and evaluate the performance of GAD for two industrial applications: building structure monitoring and smart grids. Extensive trace-driven simulations validate our theoretical analysis, and demonstrate that our approach can significantly outperform state-of-the-art approaches in terms of detection accuracy and efficiency.**

I. Introduction

Industrial Cyber-physical systems (CPSs) have been providing promising opportunities in many critical industrial segments such as energy, automotive, chemical, instrumentation, and industrial automation [1], [2]. Sensing is a key subsystem of industrial CPSs, which provides real-time measurements of physical process information, including temperature, humidity, illumination, vibration, chemical gas, smart power meter readings. In many industrial CPSs such as smart grids [3], the sensing devices can communicate with each other or with the central controller through information and communication technology (ICT) infrastructures such as wireless communications. We call such sensing systems Networked Industrial Sensing Systems (NISSs) in this paper.

In practice, sensor readings may be abnormal or faulty due to various unpredictable reasons such as harsh environments, inherently fault-prone sensors, or malicious attacks (e.g. false data injection attack in smart grid systems [4]). These anomalies could lead to significant system performance degradation or even catastrophic failure. Therefore, effective detection of sensing anomalies is highly important for the reliability and safety of the overall industrial CPS.

A. *Motivation*

In this paper, we focus on anomaly detection for NISSs. Our objective is to develop an anomaly detection algorithm that has the following three properties:

- **Real-time Detection.** Since sensor information is critical and even a single abnormal critical sensor reading may lead to a catastrophic cascade of failures throughout the whole system. Therefore abnormalities should be detected as early as possible to minimize the possibility of potential damage. To achieve this, an on-line scheme that provides real-time anomaly detections is needed. This scheme should be able identify the anomaly condition of each sensor observation, as soon as sensor observations are collected.
- **Distributed Solution.** Anomaly detection can be performed either at the central controller (i.e. centralized solution) or at local sensing devices (i.e. distributed solution). Centralized solutions require transmitting sensor readings to the central controller, which may result in data loss and delay to the detection decisions, especially in large-scale wireless NISSs. In contrast, distributed solutions are much more agile and robust to data transmission failures, and more importantly, scale to larger sizes.
- **General Solution.** For different NISSs, the system behaviors and dynamics could be very different. For instance, the stochastic behaviors of energy usage in smart grids could be quite different from that of chemical control processes. Therefore a general solution covering various NISSs is highly desirable. This self-tuning solution means that unrealistic assumptions or models related to specific industrial scenarios are not required.

B. *Our Approaches*

We propose General Anomaly Detection (GAD), a correlation-based anomaly detection algorithm for general NISSs that achieves all aforementioned properties. The contributions of this paper are summarized as follows:

1. We develop a Distributed Matching-based Grouping Algorithm (DMGA), the first correlation-aware algorithm that divides all sensing components into small strongly correlated groups in a fully distributed way. We then propose a novel approach to detect anomalies in real time, based on the spatiotemporal correlations among sensors within each correlation group.

2. We prove that the computation and storage complexity of GAD are of O(1)[1] with respect to the number of sensing devices, which means that it can be applied in large-scale industrial sensing systems such as smart grid and smart water systems.

3. The performance of GAD is evaluated in two NISSs: the sensing systems of buildings and smart grids. Extensive simulations using real building and smart grid data demonstrate that GAD achieves all its design objectives and outperforms

[1] This notation shows that GAD requires constant computational time and memory.

current approaches, in terms of detection accuracy, efficiency, and scalability.

C. Related Work

Industrial Sensing Systems. There exist a large body of research on networked sensing systems in industrial environments [5]–[7], such as building monitoring and control [8], smart water system monitoring [9], machine-condition monitoring and diagnostic [10], [11], and smart power grid systems [3]. None of the above examples consider distributed real-time anomaly detections. To improve the reliability of the overall system, [12], [13] study fault detection and for different types of sensors. However, they focus on specific sensor types and do not consider networked sensing systems.

Anomaly Detection Mechanisms. Anomaly detection schemes [14] can be broadly classified into *non-parametric (including semi-parametric)* and *parametric* approaches. On one hand, non-parametric mechanisms [15]–[17] such as statistical models and machine learning techniques, are capable of coping with changes and heterogeneities in the deployment environments. However, these solutions usually suffer from either low detection accuracy, or high computation complexity and poor scalability. Some non-parametric solutions [16], [18]–[20] tried to exploit other data-mining techniques (e.g. clustering, support vector machine (SVM), and kernel functions) to achieve a balanced solution. However, they either depend on static routing trees, or require accurately assigned thresholds to ensure their detection accuracy. Other approaches, such as [21]–[23], although they provide efficient anomaly-detection solutions, they do not focus on identifying anomalies with respect to each sensor observation, but the anomaly condition of samples (which is a sets of observations) and sensor devices.

On the other hand, parametric approaches [19], [24]–[28], that exploit *spatiotemporal correlations* between sensors, are lightweight and provide accuracy guarantees. However, these parametric approaches are normally based on quite specific assumptions which may not hold true in practice and need to be known in advance. This significantly restricts their application for many sensing systems. In summary, current approaches cannot achieve all the design objectives of anomaly detection in NISSs.

D. Organization

The remainder of this paper is organized as follows: Section II specifies our targeted sensing systems and, discusses the spatiotemporal correlations of physical phenomena. Section IV presents DMGA. The design of in-group anomaly detection are discussed in Section V. Evaluation of GAD are represented in Section VII, and we finally conclude the paper in Section VIII.

II. Preliminaries

In this section, we introduce the problem statement of our approach and the background information regarding the spatiotemporal correlation in physical phenomena.

A. Networked Industrial Sensing Systems

Our work focuses on identifying the anomaly-condition of each sensor observation in NISSs. These systems may contain single or multiple physical sources (e.g. boilers) that can simultaneously influence the observations of all of their nearby sensors. Specifically, we consider a set of sensors $\mathcal{S}$ that can communicate with other nearby sensors through wireless or wired communications. Each sensor $i \in \mathcal{S}$ is synchronised with others and monitors the same physical phenomenon (e.g. temperature and pressure), and periodically reports its measurement $r_i(t)$ at every time slot $t = \{0, 1, ...\}$.

B. Spatiotemporal Correlation in Physical Phenomena

Spatiotemporal correlation is a natural property in various physical phenomena [26], [29], including temperature, humidity, illumination, mechanical vibration, sound, gas concentration, radiation, and even human behaviors.

1) Spatial Correlation: Physical states pertaining to a given special area can simultaneously influence the sensor measurements observed in that specific area. For instance, a leakage of a water pipe can be detected by multiple nearby sensors. Specifically, consider two sensors i and j in a correlated sphere. At a given t, a *correlation mapping* $f_{i,j}^t$ from sensor i to j can be defined as:

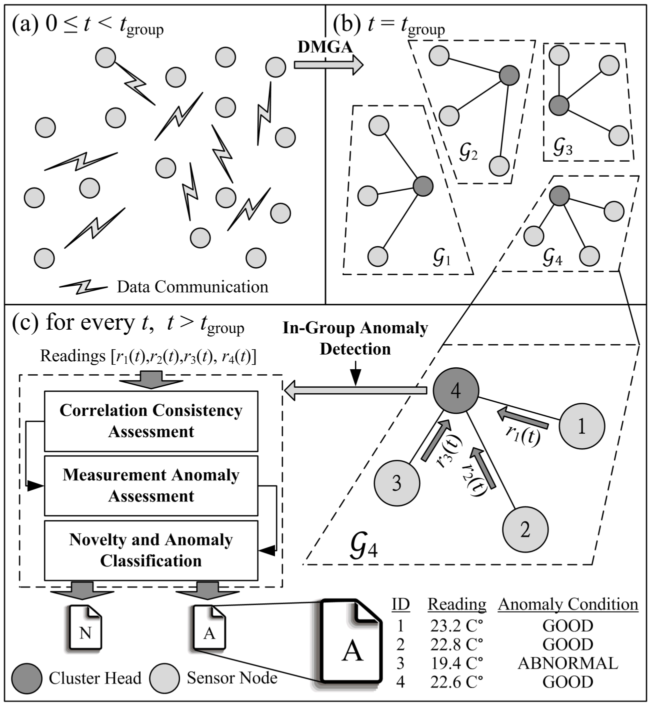
$$f_{i,j}^t : \mathcal{R}_i(t) \to \mathcal{R}_j(t) \qquad (1)$$

where $\mathcal{R}_i(t)$ and $\mathcal{R}_j(t)$ represent the set of all possible readings of i and j respectively at time t. In practice, $f_{i,j}^t$ depends on two key factors: the status of the physical phenomena in which we are interested (e.g. geographical distributions such as source values and sensor-source distances), and the surrounding environment (e.g. background noise). For instance, a correlation mapping between the readings of two temperature sensors should be affected by the nearby temperature sources (e.g. heaters), and the air temperature, which simultaneously influence the measurements of both sensors.

2) Temporal Correlation: Since physical phenomena is continuous, these spatial correlations should have a relationship with those measured previously. To be more specific, mapping $f_{i,j}^t$ should be *temporally* correlated to previous mappings $f_{i,j}^\tau$, $\tau \in [t - \triangle t, t-1]$, where sampling window size $\triangle t$ represents the period during within which the physical dynamic patterns are treated as stable.

III. The Outline of GAD

To summarise, the design of GAD can be mainly divided into two phases. In the first phase, we aim to group sensors in a sensing system $\mathcal{S}$ into multiple correlation groups, while maximize the total correlation in all correlation groups $\mathcal{G} \subseteq \mathcal{S}$. This guarantees that the sensors are highly correlated to others when they are in the same group. In the second phase, each group performs an on-line in-group anomaly-detection algorithm to tag each sensor observation with its anomaly condition in a real-time fashion. Fig. 1 illustrates the above operations of GAD algorithm.


(a) $0 \le t < t_{\text{group}}$
(b) $t = t_{\text{group}}$
DMGA
$\mathcal{G}_1$
$\mathcal{G}_2$
$\mathcal{G}_3$
$\mathcal{G}_4$
Data Communication
(c) for every t, $t > t_{\text{group}}$
In-Group Anomaly Detection
Readings $[r_1(t), r_2(t), r_3(t), r_4(t)]$
Correlation Consistency Assessment
Measurement Anomaly Assessment
Novelty and Anomaly Classification
4
1
2
3
$r_1(t)$
$r_2(t)$
$r_3(t)$
$\mathcal{G}_4$
N
A
A
Cluster Head
Sensor Node

ID	Reading	Anomaly Condition
1	23.2 C°	GOOD
2	22.8 C°	GOOD
3	19.4 C°	ABNORMAL
4	22.6 C°	GOOD

Fig. 1. The overview of GAD algorithm (a) The initial of GAD. (b) Sensors are grouped by DMGA at time t_{group}. (c) After applying DMGA, the anomaly condition of each sensor reading is identified at each time t.

- When GAD initials its algorithm (Fig.1(a)), each sensor node $i \in \mathcal{S}$ broadcasts its readings to its one-hop communication neighbours $\mathcal{N}_i^{1-hop}$. These readings will be stored and used by each sensor node $j \in \mathcal{N}_i^{1-hop}$ to calculate the Pearson correlation coefficient between sensor i and j.
- When acquire $|t_{group}|$ observations for the correlation calculation (Fig.1(b)), DMGA, a correlation-aware grouping algorithm, is performed. This algorithm exploits a matching technique [30] to group sensors into clusters, where the total correlations between sensor nodes are maximized. During this grouping, each sensor node i exchanges grouping-request messages with their neighbour nodes in $\mathcal{N}_i^{1-hop}$. This process terminates when the size of each group $\mathcal{G} \subseteq \mathcal{S}$ meets the minimum requirement N^{min}. The detailed design of DMGA can be found in Section IV.
- After grouping (Fig.1(c)), each group chooses a sensor node as its cluster head. This cluster head can be selected as per application requirements, including sensor energy budget, communication capability, or sensor ID. Once the cluster heads are selected, at each time slot t, each sensor sends its reading to its cluster head. After the cluster heads received all sensor readings in their groups, they perform an in-group anomaly detection algorithm to tag each reading with its anomaly condition. The details of this in-group algorithm will be given in Section V.

Note that, although GAD groups sensor nodes in to groups, in which sensor measurements have to be first routed to cluster heads, users can still apply any network topology, routing protocol, or data aggregation scheme, after sensor readings are tagged with their anomaly condition.

IV. Correlation-aware grouping

At the first stage of GAD, a grouping scheme is required to divide sensors into small correlation groups in a fully distributed way. The objective of grouping is to ensure strong spatiotemporal correlations among all sensors in each correlation group, while minimizing the overhead of GAD. To this end we develop distributed matching-based grouping algorithm (DMGA), the first correlation aware grouping algorithm for anomaly detection.

A. DMGA Design

For a given industrial sensing system $\mathcal{S}$, let $\mathcal{G} \subseteq \mathcal{S}$ represent a correlation group, and $\mathbb{G} \subseteq 2^{\mathcal{S}}$ represent the set of all correlation groups, i.e. a grouping solution. We define $\Pi \subset 2^{\mathcal{S}}$ as the set of all possible grouping solutions. DMGA aims to find a grouping solution $\mathbb{G} \in \Pi$, which maximizes the spatiotemporal correlations in all groups and guarantees strong spatiotemporal correlations among all sensors in each correlation group, while minimize the computational complexity of GAD. Formally, DMGA aim at solving the following problem:

$$\max_{\mathbb{G} \in \Pi} \quad \sum_{\{i,j\} \subset \mathcal{G}, \mathcal{G} \in \mathbb{G}} c_{i,j} \tag{2}$$

$$\textbf{subject to}$$

$$|\mathcal{G}| \ge N^{\min}, \ \forall \mathcal{G} \in \mathbb{G} \tag{3}$$

$$c_{i,j} \ge \mathrm{c}^{\min}, \ \forall i,j \in \mathcal{G} \in \mathbb{G} \tag{4}$$

$$\sum_{\mathcal{G} \in \mathbb{G}} |G|^2 \le \sum_{\mathcal{G} \in \mathbb{G}'} |G|^2, \ \forall \mathbb{G}' \ne \mathbb{G} \tag{5}$$

where $0 \le c_{i,j} \le 1$ is the standard *Pearson correlation coefficient*, which represents the spatiotemporal correlation between sensor i and j; $\mathrm{c}^{\min}$ is a predefined minimal correlation threshold; and $N^{\min}$ presents the minimal correlation group size to guarantee anomaly-detection accuracy [15], [25]. The objective (2) is to maximize total correlations between sensors in all groups. Constraint (3) states that each group should consist of at least $N^{\min}$ sensors. Constraint (4) ensures the strong correlations between each pair of sensors in each group. Constraint (5) ensures the grouping solution should also minimize the computational overhead of GAD algorithm, which is $O(\sum_{\mathcal{G} \in \mathbb{G}} |G|^2/|\mathcal{S}|)$ per sensor (This will be discussed in detail in next section).

DMGA solves problem (2)-(5) by utilizing distributed Maximum Weighted Matching (MWM) [30]. In graph theory, a matching is a set of links that do not share common node. A MWM is a matching with maximal aggregate weights (i.e. the maximal aggregated correlations in our context) over all other matchings for a given weight graph. The pseudocode of the DMGA is summarized in Fig. 2.

The idea of DMGA is based on the concept of hyper correlation graphs $G_c^n(\mathbb{G}^n, \mathbb{L}^n, \mathbb{W}^n)$ at the nth iteration of the **while** loop (lines 6-10), where $\mathbb{G}^n$ represents the set of hyper-nodes (i.e. non-overlapping correlation groups of the same

```
Input Parameters
$\mathcal{S}$: the set of all sensors in a industrial sensing system.
$c_{i,j}$: Correlation between every pair of sensors $i,\ j \in \mathcal{S}$.
$\mathrm{c}^{\min}$: Predefined minimal correlation threshold.
$N^{\min}$: the minimal number of nodes in a correlation group.
$\mathcal{N}_i^{1\text{-}hop}$: the set of one-hop communication neighbors of $i$.
Variables:
$n$: iteration round.
$G_c^n$: the $n$th Hyper Correlation Graph.
$\mathbb{G}^n$: the set of hyper-nodes in the $n$th iteration.
$\mathbb{L}^n$: the set of hyper-links in the $n$th iteration.
$\mathbb{W}^n$: the set of weights over hyper-links in the $n$th iteration.
Functions:
CON($\mathbb{G}, \mathbb{L}, \mathbb{W}$): construct a hyper-graph based on $\mathbb{G}, \mathbb{L}, \mathbb{W}$.
MWM($G$): compute the MWM of a graph $G$.

/* Initialization */
01: $n \leftarrow 0$;
02: $\mathbb{G}^0 \leftarrow \{\{i\} : i \in \mathcal{S}\}$;
03: $\mathbb{L}^0 \leftarrow \{(\{i\},\{j\}) : i \in \mathcal{S},\ j \in \mathcal{N}_i^{1\text{-}hop},\ c_{i,j} \geq \mathrm{c}^{\min}\}$;
04: $\mathbb{W}^0 \leftarrow \{c_{i,j} : (\{i\},\{j\}) \in \mathbb{L}^0\}$; $G_c^0 \leftarrow$ CON($\mathbb{G}^0, \mathbb{L}^0, \mathbb{W}^0$);

/*Establish nonoverlapping groups with same sizes*/
05: while $2^n < N^{\min}$ do
06:     $n \leftarrow n + 1$;
07:     $\mathbb{G}^n \leftarrow$ MWM($G_c^{n-1}$);
08:     $\mathbb{L}^n \leftarrow \{(\mathcal{G}_1, \mathcal{G}_2) : \forall i \in \mathcal{G}_1 \in \mathbb{G}^n, j \in \mathcal{G}_2 \in \mathbb{G}^n$
08:             s.t.$(\{i\},\{j\}) \in \mathbb{L}^0\}$;
09:     $\mathbb{W}^n \leftarrow \{W_{\mathcal{G}_1,\mathcal{G}_2} = \sum_{i,j \in \mathcal{G}_1 \cup \mathcal{G}_2} c_{i,j} : (\mathcal{G}_1, \mathcal{G}_2) \in \mathbb{L}^n\}$;
10:     $G_c^n =$ CON($\mathbb{G}^n, \mathbb{L}^n, \mathbb{W}^n$);

/* Insert individual sensors in established groups */
11: for each sensor $(i \in \mathcal{S}) \wedge (i \notin \mathcal{G}, \forall \mathcal{G} \in \mathbb{G}^n)$ do
12:     for each group $\mathcal{G} \in \mathbb{G}^n, |\mathcal{G}| = 2^n$ do
13:         if $\forall \mathcal{G}' \in \mathbb{G}^n$ s.t. $|\mathcal{G}'| = 2^n \wedge \sum_{j \in \mathcal{G}} c_{i,j} \geq \sum_{j \in \mathcal{G}'} c_{i,j}$ then
14:             $\mathcal{G} \leftarrow \mathcal{G} \cup \{i\}$;
15: return ($\mathbb{G}^n$);
```

Fig. 2. The pseudo code of DMGA.

size 2^n), $\mathbb{L}^n$ is the set of hyper-links between each pair of hyper nodes; and the weight of each hyper-link $(\mathcal{G}_1, \mathcal{G}_2) \in \mathbb{L}^n$ is computed as the sum of *Pearson correlation coefficients* between each pair of sensors in $\mathcal{G}_1 \cup \mathcal{G}_2$. At the nth iteration, the **while** loop first computes the MWM (discuss later) for the hyper-correlation graph $G_c^{n\text{-}1}(\mathbb{G}^{n\text{-}1}, \mathbb{L}^{n\text{-}1}, \mathbb{W}^{n\text{-}1})$ generated in the last iteration. Based on the computed MWM the nth hyper correlation graph $G_c^n(\mathbb{G}^n, \mathbb{L}^n, \mathbb{W}^n)$ is then constructed. According to Theorem 1, the **while** loop operates at most $\lceil \log_2 N^{\min} \rceil$ times, where $\lceil \log_2 N^{\min} \rceil$ represents the minimal integer that is larger than $\log_2 N^{\min}$. Eventually, if each sensor in $\mathcal{S}$ has been put into a correlation group in $\mathbb{G}^n$, the algorithm terminates after the **while** loop; otherwise, the algorithm inserts each sensor individually to an established group in $\mathbb{G}^n$ (lines 11–15), according to Theorem 1.

Fig. 3 illustrates an example to show the operation of DMGA. Assume $N^{\min} = 4$ in this example. Initially, we have the initial hyper-correlation graph G_c^0 consisting of 10 sensors (Step 0). In the first iteration of the **while** loop (Steps 1 and 2 in Fig. 3), the MWM is computed (Step 1) and the fisrt hyper-correlation graph G_c^1 is constructed (Step 2), where $\mathbb{G}^1 = \{\{a, c\},\ \{g, h\},\ \{e, d\},\ \{i, j\},\ \{f, b\}\}$.

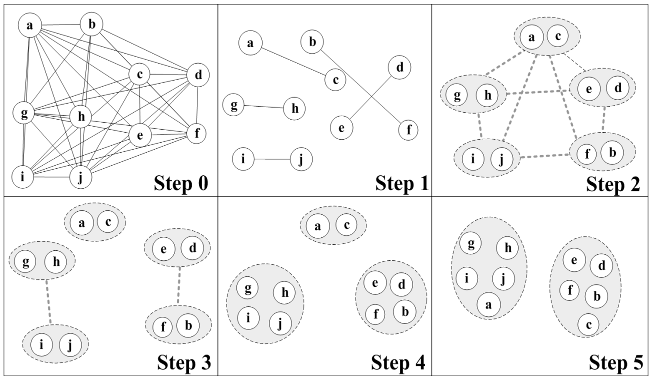


Fig. 3. An example to illustrate DMGA.

Similarly, the second **while** loop (steps 3 and 4 in Fig. 3) computes the MWM for G_c^1 (Step 3) and constructs G_c^2 (Step 4), where $\mathbb{G}^2 = \{\{g, h, i, j\},\ \{e, d, f, b\}\}$. Finally, in Step 5, the individual sensors a and c are inserted into the established two groups respectively, and then the algorithm terminates.

B. Distributed Operations of DMGA

DMGA is based on *locally greedy* optimal-link selections. Three one-hop control messages are used in this DMGA: Matching Apply (**MA**), Matching Reply (**MR**) and **drop**. To initialize the first hyper-correlation graph $G_c^0(\mathbb{G}^0, \mathbb{L}^0, \mathbb{W}^0)$ (Lines 1-4 in Fig. 2), every sensor $i \in \mathcal{S}$ broadcast its readings of a previous periods $\triangle t$ to all its 1-hop neighbors $j \in \mathcal{N}_i^{1\text{-}hop}$. Based on these sensor readings, the first hyper correlation graph $G_c^0(\mathbb{G}^0, \mathbb{L}^0, \mathbb{W}^0)$ can be established in a fully distributed manner.

Now we discuss the distributed operations of matching. For the nth hyper-correlation graph $G_c^n(\mathbb{G}^n, \mathbb{L}^n, \mathbb{W}^n)$, DMGA compute MWM as follows. Every hyper node G_k^n selects its locally heaviest-weighted and free (LHWF) link $(\mathcal{G}_k^n, \mathcal{G}_{l*}^n)$, where

$$(\mathcal{G}_k^n, \mathcal{G}_{l*}^n) = \arg \max_{(\mathcal{G}_k^n, \mathcal{G}_l^n) \in \mathbb{L}^n} \sum_{i,j \in \mathcal{G}_k^n \cup \mathcal{G}_l^n} c_{i,j} \tag{6}$$

Then $\mathcal{G}_k^n$ sends a **MA** message to $\mathcal{G}_{l*}^n$ to request the matching of link $(\mathcal{G}_k^n, \mathcal{G}_{l*}^n)$. If this link is also the LHWF link for $\mathcal{G}_{l*}^n$, then $\mathcal{G}_{l*}^n$ sends a **MR** message back to $\mathcal{G}_k^n$ to confirm this link is *matched*, and multicasts a **drop** message to its other neighbor groups, otherwise, $\mathcal{G}_l$ ignores the message. Alternatively, $(\mathcal{G}_k^n, \mathcal{G}_{l*}^n)$ is eventually *dropped*. If link $(\mathcal{G}_k^n, \mathcal{G}_{l*}^n)$ is dropped, then $\mathcal{G}_k^n$ selects a new LHWF link and sends another **MA** message. The above process repeats until every hyper node has either a *matched* link, or all its links are *dropped* and marked as **free**.

C. Performance Analysis

Theorems 1 below demonstrates three theoretical performance guarantees achieved by DMGA. For readability, the proofs of this theorem is presented in Appendix.

Theorem 1. *DMGA achieves the following performance guarantees:*

1) *DMGA minimizes the computational overhead of GAD algorithm, i.e. Constraint (5) is guaranteed.*
2) *The worst-case communication overhead of DMGA is O(1) per sensor, with respect to the industrial sensing system size $|\mathcal{S}|$.*
3) *DMGA achieves at least $1/N^{\min}$ performance of the optimal solution of problem (2)-(5).*

Theoretically, problem (2)-(5) may not have a feasible solution for some large $N^{\min}$ and $\mathrm{c}^{\min}$. From graph theoretical point of view, the setting of $\mathrm{c}^{\min}$ defines the topology of the initial hyper-correlation graph $G_c^0(\mathbb{G}^0, \mathbb{L}^0, \mathbb{W}^0)$. To ensure (2)-(5) has a feasible solution, $G_c^0(\mathbb{G}^0, \mathbb{L}^0, \mathbb{W}^0)$ must have at least $|\mathcal{S}|/N^{\min}$ non-overlapping cliques (i.e. complete subgraph) of size $N^{\min}$ [31].

In practice, when the certain setting of $N^{\min}$ and $\mathrm{c}^{\min}$ derives no feasible solution, the most straightforward solution is to reduce these two thresholds. However, to fulfill the redundancy and reliability requirements of GAD, it is necessary to have a reasonably high $\mathrm{c}^{\min}$. This is because that GAD exploits the spatiotemporal correlations between sensors, which are guaranteed by $\mathrm{c}^{\min}$, the lowest bond of sensor-correlation requirement during DMGA grouping. Therefore, if there is no feasible solution, users should try to reduce $N^{\min}$ rather $\mathrm{c}^{\min}$. It is worth noting that GAD only require each correlation group consists of more than 3 sensors to achieve its high detection accuracy.

V. In-Group Anomaly Detection

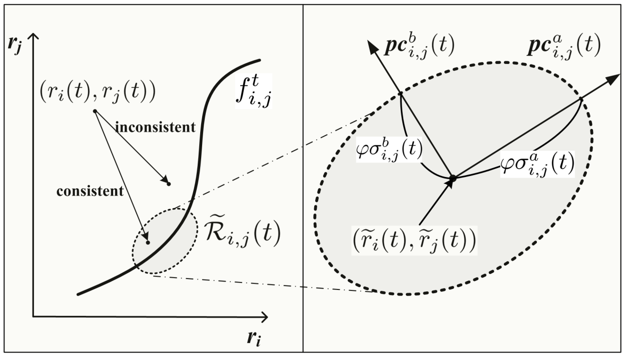
After DMGA, the following stages of anomaly detection will be performed within each correlation group $\mathcal{G}$: correlation consistency assessment, measurement anomaly assessment, and classification of novelty and anomaly. We assume that sensor reading errors follow Gaussian distribution, which is a well-accepted assumption and normally holds true in practice.

A. Correlation Consistency Assessment

Consider each pair of sensors i and j in a group $\mathcal{G}$. Let $r_i(t)$ and $r_j(t)$ be the readings of sensor $i \in \mathcal{G}$ at slot t, respectively. As we discussed in Subsection II-B, $r_i(t)$ and $r_j(t)$ should be temporally correlated to their the previous readings of sensor i and j. Therefore, based on previous sensor readings before time t, their consistency region $\widetilde{\mathcal{R}}_{i,j}(t)$ can be computed. Here, $\widetilde{\mathcal{R}}_{i,j}(t)$ represents the set of all possible potentially consistent reading pairs of $r_i(t)$ and $r_j(t)$ at current time t. If current sensor reading pair $(r_i(t), r_j(t)) \in \widetilde{\mathcal{R}}_{i,j}(t)$, we say the reading pair $r_i(t)$ and $r_j(t)$ are consistent; otherwise, they are inconsistent. Denote $C_{i,j}(t)$ as the *correlation consistency* of sensor readings $r_i(t)$ and $r_j(t)$ at slot t, i.e.

$$C_{i,j}(t) = \begin{cases} 1, & \text{if } (r_i(t), r_j(t)) \in \tilde{\mathcal{R}}_{i,j}(t) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Geometrically, $\widetilde{\mathcal{R}}_{i,j}(t)$ is a rotated ellipse area on the Cartesian coordinate formed with r_j and r_j, as shown in Fig. 4 (a). Here, the center of the ellipse $(\widetilde{r}_i(t), \widetilde{r}_j(t))$ is computed by using Exponential Weighted Moving Average (EWMA) of previous sensor readings as follows:


r_j
$(r_i(t), r_j(t))$
$f^t_{i,j}$
inconsistent
consistent
$\widetilde{\mathcal{R}}_{i,j}(t)$
r_i
$\boldsymbol{pc}^b_{i,j}(t)$
$\boldsymbol{pc}^a_{i,j}(t)$
$\varphi\sigma^b_{i,j}(t)$
$\varphi\sigma^a_{i,j}(t)$
$(\widetilde{r}_i(t), \widetilde{r}_j(t))$

(a) (b)

Fig. 4. Illustration of correlation consistency assessment.

$$\widetilde{r}_i(t) = 0.5 r_i(t-1) + 0.5 \widetilde{r}_i(t-1) \tag{8}$$

The major and minor radius of the ellipse $\tilde{\mathcal{R}}_{i,j}(t)$ are computed by using the Principle Component Analysis (PCA) [22], which is a mathematical procedure to convert observations of multiple observers into orthogonal variables called principle components (PCs). These PCs indicate the most representative variances in these observations. Consider the first step of PCA, Eigen decomposition on a data covariance matrix:

$$\boldsymbol{COV}_{i,j} = \begin{bmatrix} cov_{i,i}(t) & cov_{i,j}(t) \\ cov_{j,i}(t) & cov_{j,j}(t) \end{bmatrix} \tag{9}$$

where each entry $cov_{i,j}(t)$ is defined as

$$cov_{i,j} = \mathbb{E}^{\mathrm{GOOD}}_{t,\triangle t}[(r_i(t) - \mathbb{E}^{\mathrm{GOOD}}_{t,\triangle t}[r_i(t)])(r_j(t) - \mathbb{E}^{\mathrm{GOOD}}_{t,\triangle t}[r_j(t)])]$$

where the operator $\mathbb{E}^{\mathrm{GOOD}}_{t,\triangle t}[\cdot]$ returns the arithmetic average of previous *reliable* sensor readings during the sampling window $[t-\triangle t, t-1]$. The identification of reliable (i.e. **GOOD**) sensor readings will be discussed in detail in next subsection. With $\boldsymbol{COV}_{i,j}(t)$, we have

$$\boldsymbol{COV}_{i,j}(t)\boldsymbol{PC}_{i,j}(t) = \boldsymbol{\lambda}_{i,j}(t)\boldsymbol{PC}_{i,j}(t) \tag{10}$$

where the eigenvectors $\boldsymbol{PC}_{i,j}(t)$ contains two orthogonal principle components (vectors) $\boldsymbol{pc}^a_{i,j}(t)$ and $\boldsymbol{pc}^b_{i,j}(t)$; and the matrix of eigenvalues $\boldsymbol{\lambda}_{i,j}(t)$ consists of two variances $(\sigma^a_{i,j}(t)^2)$ and $(\sigma^b_{i,j}(t))^2$, where $\varphi\sigma^a_{i,j}(t)$ and $\varphi\sigma^b_{i,j}(t)$ are standard deviations related to the two principle components above.

With $\boldsymbol{PC}_{i,j}(t)$ and $\boldsymbol{\lambda}_{i,j}(t)$, the consistency region ellipse $\tilde{\mathcal{R}}_{i,j}(t)$ can be computed, as shown in Fig. 4 (b). Here, φ is a parameter that controls the probability margin of $\widetilde{\mathcal{R}}_{i,j}(t)$. For instance, $\varphi = 3$ can assure that 99.46% of normal observations lie in $\tilde{\mathcal{R}}_{i,j}(t)$.

B. Measurement Anomaly Assessment

In this stage, GAD exploits a trust-based voting algorithm to identify anomaly condition of the reading $r_i(t)$ of each sensor i in every correlation group $\mathcal{G}$ at time t. Here, the correlation consistencies $C_{i,j}(t)$ defined in (7), are considered as a measure of trust between each pair of sensors $i, j \in \mathcal{G}$.

Before introducing the voting, we first define the consistent neighbor set $\mathcal{N}_i^{cons}(t)$ for each sensor $i \in \mathcal{G}$ at time t as:

$$\mathcal{N}_i^{cons}(t) = \{j : C_{i,j}(t) = 1,\ j \in \mathcal{N}_i\}$$

where $\mathcal{N}_i = \mathcal{G} - \{i\}$ is the set of sensor i's all neighbors. This stage consists of two rounds of voting:

1) First Round Voting: In this voting, certain sensors are voted as trustworthy references. Sensor i is regarded as *trustworthy*, if the majority of i's neighbors are consistent neighbors (i.e. $|\mathcal{N}_i^{cons}(t)|/|\mathcal{N}_i| \geq 50\%$); otherwise, it is regarded as *untrustworthy*.

2) Second Round Voting: After the first round voting, only trustworthy sensors are qualified to involve in the second round voting. For each sensor i, denote $\mathcal{N}_i^{trust}(t) \subseteq \mathcal{N}_i$ as the set of its all trustworthy neighbors. The second round voting identifies one of the following three conditions of reading $r_i(t)$ of each sensor in $\mathcal{G}$:

- **GOOD**, if the majority of i's neighbors are trustworthy ($|\mathcal{N}_i^{trust}(t)|/|\mathcal{N}_i| \geq 50\%$), and the majority of its consistent neighbors are trustworthy ($|\mathcal{N}_i^{cons}(t) \cap \mathcal{N}_i^{trust}(t)|/|\mathcal{N}_i^{trust}(t)| \geq 50\%$).
- **ABNORMAL.** if the majority of i's neighbors are trustworthy, and the minority of its consistent neighbors are trustworthy ($|\mathcal{N}_i^{cons}(t) \cap \mathcal{N}_i^{trust}(t)|/|\mathcal{N}_i^{trust}(t)| < 50\%$).
- **UNKNOWN**, if the minority of i's neighbors are trustworthy ($|\mathcal{N}_i^{trust}(t)|/|\mathcal{N}_i| < 50\%$). In this case, anomaly condition of reading $r_i(t)$ cannot be determined, due to the lack of trustworthy references.

C. Novelty and Anomaly Classification

In this stage, GAD aims to distinguish novelties from anomalies. Unlike anomalies that are faults or errors, *novelties* are the emerging patterns in the physical process that were previously unobserved. Novelties represent the real dynamics of physical phenomena, which may be critical for industrial CPS applications. For example, the occurrence of sudden heightened temperature (e.g. a fire) is a novelty to temperature monitoring systems rather than an anomaly.

In a correlated group $\mathcal{G}$, the probability that the measurements of all sensors are unreliable simultaneously is close to zero. Therefore, we assume that the novelty can be identified when the majority of spatial correlations between sensor measurements changes, i.e. the readings of more than 50% sensors in a correlation group are declared **UNKNOWN**. This assumption is reasonable, because the fact that spatial varying physical phenomena should influence nearby sensor readings at the same time. Therefore, when genuine environmental changes occur, the correlation mappings between each pair of sensors in a correlation group should begin to vary.

D. Performance Analysis

Theorem 2 and 3 below demonstrate the scalability of GAD.

Theorem 2. *The per sensor computational complexity of GAD is O(1) with respect to the industrial sensing system size $|\mathcal{S}|$.*

Proof. It can be seen only the consistency assessment stage of GAD require heavy computations, i.e. Eigen decomposition requires $\mathrm{O}(|\mathcal{G}|^2)$ matrix-multiplication operations for each correlation group $\mathcal{G} \in \mathbb{G}$, where $\mathbb{G}$ is the grouping solution computed by DMGA. Therefore the per sensor computational complexity is $\mathrm{O}(\sum_{\mathcal{G}\in\mathbb{G}} |\mathcal{G}|^2)/|\mathcal{S}|)$. Consider DMGA, we have

$$\begin{aligned}\sum_{\mathcal{G}\in\mathbb{G}} |\mathcal{G}|^2/|\mathcal{S}| &\leq \frac{|\mathcal{S}|}{N^{\min}}(N^{\min}+1)^2\frac{1}{|\mathcal{S}|}\\ &= (N^{\min}+1)^2/N^{\min}\end{aligned}$$

This proves Theorem 2. □

Theorem 3. *The per sensor storage complexity of GAD is O(1) with respect to the industrial sensing system size $|\mathcal{S}|$.*

Proof. According to Section IV and Subsection V-A, each sensor requires $\mathrm{O}(|\mathcal{N}^{1-hop}| \times t_{group})$ and $\mathrm{O}(|\mathcal{G}| \times |\triangle t|)$ memory storage to perform DMGA and to compute $\widetilde{\mathcal{R}}_{i,j}(t)$, respectively. Since $|t_{group}|$ and $|\triangle t|$ are constant parameters, the storage complexities of these two operations become $\mathrm{O}(|\mathcal{N}^{1-hop}|)$ and $\mathrm{O}(|\mathcal{G}|)$. Typically, $|\mathcal{G}| \leq |N^{min}| + 1 \leq |\mathcal{N}^{1-hop}| = D$,where D is the degree of $G_c(\mathbb{G}^0, \mathbb{L}^0, \mathbb{W}^0)$; therefore, the per sensor storage complexity of GAD is $\mathrm{O}(D)$, which is independent of $|S|$. □

VI. Deploy GAD in Real-World NISSs

Theorem 2, 3 in Subsection V-D and Part 2 of Theorem 1 in Subsection IV-C have demonstrated that GAD is a lightweight distributed anomaly-detection algorithm that has a great potential in large-scale industrial sensing systems. In this section, we further discuss the feasibility of deploying GAD on real-world NISSs.

- **Storage**. According to the analysis in Theorem 3, DMGA is typically the most storage-intense operation in GAD. For example, when a sensor node has 20 1-hop neighbours (i.e. $|\mathcal{N}^{1-hop}| = 20$), and needs 50 2-bytes readings of each neighbour ($t_{group} = 50$) to calculate their Person correlation coefficients, it requires at most 2kB memory to perform GAD. This storage requirement can be easily fulfilled by current resource-constrained wireless sensors platforms, such as TelosB (10kB memory) or iMote2 (256kB memory).
- **Computation**. In GAD, the PCA operation in Section V-A is the most computational-intense operation. This PCA operates on 2×2 covariance matrixes. For example, in correlation group having 4 sensor nodes, the cluster head only has to compute 10 covariance values, to form the covariance matrixes needed by GAD. This make GAD be suitable for resource-restrained micro control units (MCUs), such as MSP430 on TelosB, to perform.
- **Communication** When apply GAD on sensing systems, the main communication overhead comes from the in-group data exchanging between sensors. This communication overhead can be minimized by assigning the most communication effective sensor nodes as the cluster heads defined in Section III. For example, when a NISS is constructed under a tree topology, users can select the sensor nodes that are closer to the root as cluster heads.

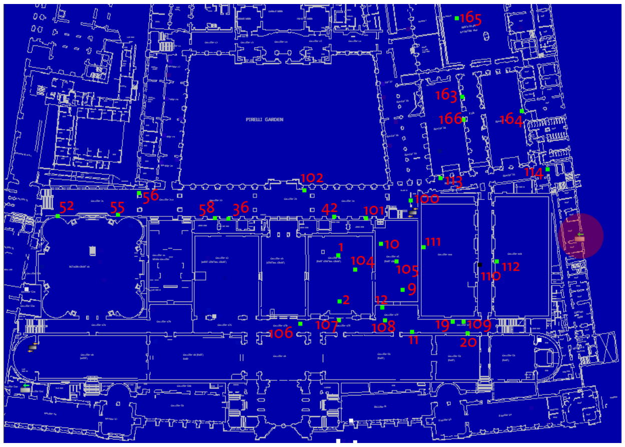


Fig. 5. Floor plan and sensor deployment in the ground floor of the building.

VII. Evaluation

This section presents simulation studies to evaluated the performance of GAD, and to demonstrate how to apply GAD in various industrial scenarios. All simulations used data sets from real-world industrial environments, and ran on a PC with Intel 4-core CPU, and 8 GB memory.

A. Case Study 1: Environmental Monitoring in a Building

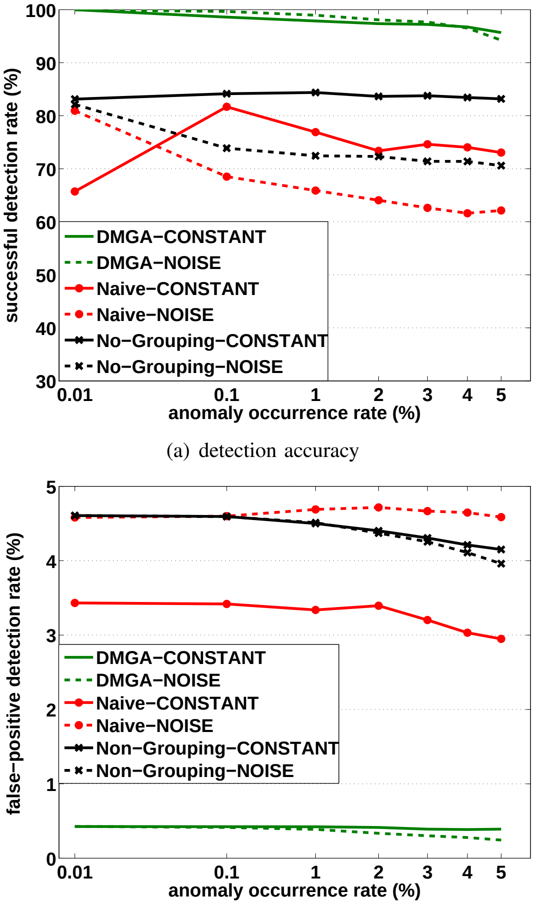
As one of the most important industrial sensing applications, structural monitoring aims to guarantee the healthy conditions of buildings, by deploying ambient sensors at the specific assigned positions [8], [32]. Due to massive deployments, these sensors are usually low-cost low-end components and typically do not provide reliability guarantee. Therefore, anomaly detection is required to ensure the reliability of overall sensing systems.

We constructed simulations based on a real data set obtained from a building (the floor plan of which is illustrated in Fig. 5) was performed. This data set contains of temperature and humidity data from 72 sensors (with same sensing rate of one reading per every 15 seconds) during one year (from 20/Oct/2008 to 19/Oct/2009). Due to the lack of ground-truth-anomalous measurements, anomalies were simulated by the two abnormal sensor behaviours: CONSTANT fault and NOISE fault [33], [34]. To create discernable impacts on sensor measurements, we simulated CONSTANT and NOISE faults by increasing sensor readings by 20% and the background noises by 600%, respectively. Note that since SHORT faults [33], [34] show similar short-term behaviours (i.e. a sudden transients in sensor readings) to the other two types of faults, and they are relatively easier to detect [34], we did not include this type of faults in our simulation.

To measure the detection accuracy of GAD, we use two different metrics, successful detection rate (SDR) and false-positive detection rate (FPDR), which are defined as follows:

$$SDR = \frac{\text{number of successful detections}}{\text{number of anomalous measurements}} \times \%$$

$$FPDR = \frac{\text{number of false detections}}{\text{number of normal measurements}} \times \%$$



(a) detection accuracy

(b) false-positive detection rate

Fig. 6. The performance of GAD with different grouping criteria in the build structure monitoring application.

Also, to demonstrate the effectiveness of DMGA, two non-standard GAD was performed on the same data set, including the GAD that adopted a naive grouping algorithm (with which sensors are randomly grouped) and the GAD without grouping (where 72 sensors are regarded as a single group). Both DMGA and naive grouping clustered the 72 sensors into 9 different groups consisting of 8 sensors.

As shown Fig 6(a) and (b), by using DMGA, GAD can identify more than 95% anomalies with only around 0.4% false-positives, which are much better than no-grouping (75%–85% SDR and 4%– 4.6% FPDR) and naive grouping (61% – 81% SDR and 4% –4.6% FPDR). This is because naive grouping cannot provide spatiotemporal correlation guarantees, and strongly spatial correlations do not exist between all the sensors in the entire building. The lack of correlations resulted in many unexpected false-positive events and erroneous devices.

B. Case Study 2: FDI Attack Detection in Smart Grids

Smart grids systems use information and communication technology (ICT) to provide reliable and efficient electricity transmission and distribution of power grids [3], [35]. Here, sensors such as smart meters are connected via both power lines and ICT infrastructure as shown in Fig. 7. However, the heterogeneity, diversity, and complexity of the smart grid

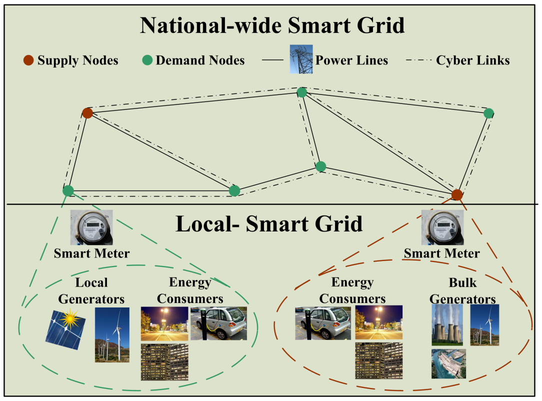


Fig. 7. The conceptual illustration of a smart grid system.

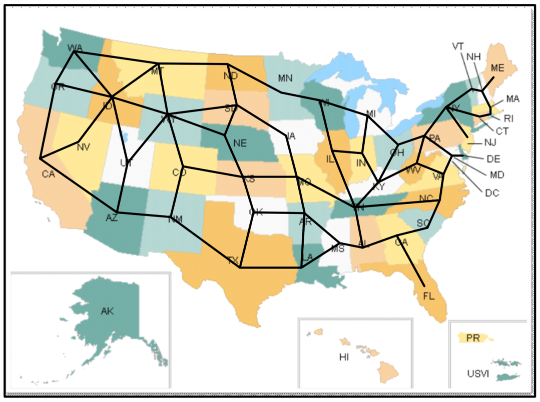


Fig. 8. US smart grid topology.

components pose critical challenge in ensuring overall cyber security [36]. Among various emerging security issues, false data injection (FDI) attacks (i.e. maliciously modify sensor readings) have a substantial cost in terms of the energy distribution process [4]. To demonstrate how GAD can minimize the impacts of FDI attacks, simulations based on a simplified version of the US smart grid (as illustrated in Fig. 8 (http://www.oe.energy.gov/smartgrid.htm)) was conducted in this subsection. In this set of simulations, each state contains 10 energy suppliers and 10 energy consumers to simulate local energy generations and consumptions within the state. Each of these suppliers/consumers contains generates 365 daily energy generation/consumption profiles to simulate the annual behaviour of the smart-grid system. All these profiles used in this simulation are based on 2009 US Energy Information Administration State Electricity Profiles (available at http://www.eia.gov/).

Two different types FDI attacks were considered in this set of simulations: The first one aims to increase the metric $Cost_{total}$ defined in [4] (i.e. the total energy transmission cost over all power lines in the smart grid system). The demands of the energy consumers were maliciously increased by 100% and 200% while the supplies of the energy suppliers were falsely decreases by 10% and 20%. The second one aim to incur significant power-supply outage rate (i.e. the percentage of outage states over all energy demanding states), by falsely increasing the energy supply-demand value by 15% and 30% in our simulations.

Fig. 9 shows the simulation results when applying our FDI-detection approach and the state-of-the-art distanced-based (DB) solution [16], [37] proposed in 2013. Both approaches referenced the past 30 days data during evaluation (i.e. $\triangle t = 30$). Also, the user-defined parameters of DB solution was set as r=0.5 and D=0.5 [37].

As shown in Fig. 9 (a) and (c), GAD can significantly reduce the adverse impact caused by FDI attacks with nearly zero false-positive detections. The total energy transmission cost reduces from 6.35 to 5.21 million US dollars when FDI-attack probability is 1%. In contrast, although the DB-detection scheme achieves comparable cost reduction, it suffers from extremely high false-positive detection rate. For instance, when the FDI-attack probability is equal to 1%, the DB-detection scheme treats about 40% correct advertised energy supplying and demanding as FDI attacks. Such false-positive detections would result in significant confusion in decision making and would incur additional costs, such as extra labour to understand the problem where the integrity of the information is required to be checked manually.

Fig. 9 (b) and (c) show similar results to our observation in the aforementioned simulation. GAD manages to reduce the user outage rate from 40.5% to 18% with almost no false-positive detections, while the distance-based solution suffered from false alarms, especially when the attack probability is 1%. Furthermore, as shown in Fig. 9(d), GAD only introduces about 15% additional computational overhead to the system, while the DB-approach leads to more than 4000% additional computational complexity, which demonstrate the high efficiency of GAD.

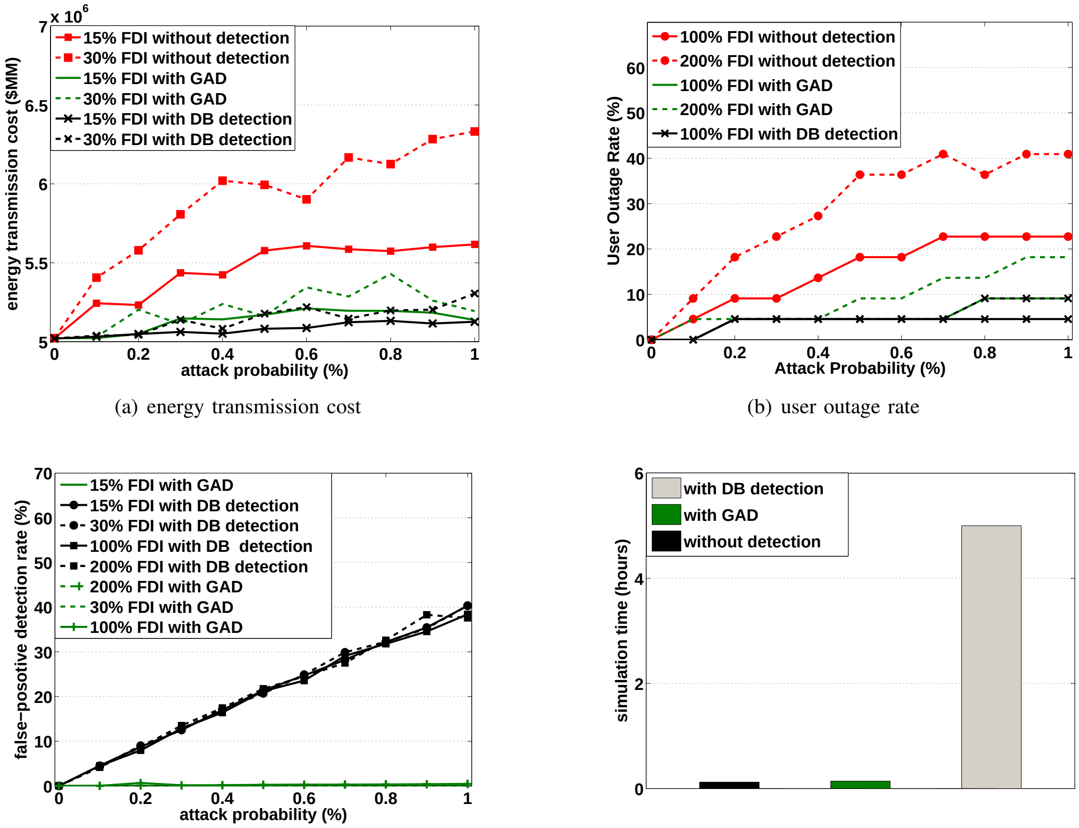
VIII. Conclusion

In this paper, we propose GAD, a novel distributed real-time approach for anomaly detection in general large-scale networked industrial sensing systems. Unlike current anomaly-detection approaches that make stringent assumptions about physical phenomenon being sensed and anomaly models, GAD assumes that spatiotemporal correlations exist in the physical system and that measurement errors follow Gaussian distributions. Both are well-accepted assumptions and normally hold true in practice. We prove the scalability and efficiency of GAD, by computing its worst-case complexity bounds. The performance of GAD is then evaluated using real data from two industrial sensing systems: building structural monitoring and smart grids. Simulation results demonstrate that GAD can be used in different industrial sensing systems and outperforms state-of-the-art approaches in terms of scalability, detection accuracy, and efficiency.

Appendix

A. *Proof of Part 1*

We can write the overhead of the GAD algorithm for a correlation group $\mathcal{G}$ as $\alpha|\mathcal{G}|^2, \alpha > 0$. To minimize computation overhead of GAD, we have following problem



(a) energy transmission cost

(b) user outage rate

(c) false positive detection rate

(d) false positive detection rate

Fig. 9. The performance of GAD in the smart grid FDI attack detection application.

$$\min_{\mathbb{G}\in\Pi} \qquad \alpha\sum_{\mathcal{G}\in\mathbb{G}}|\mathcal{G}|^2 \tag{11}$$

subject to

$$\sum_{\mathcal{G}\in\mathbb{G}}|\mathcal{G}|=|\mathcal{S}|,\quad |\mathcal{G}|\geq N^{\min},\forall\mathcal{G}\in\mathbb{G} \tag{12}$$

Due to its convexity of the objective (11), it is clear to see that the objective (11) is minimized when every group size is identical and minimized (i.e. $N^{\min}$). DMGA achieves this according to lines 1–10 of shown Fig. 2. Let $N^{rest}=|\mathcal{S}|$ mod $N^{\min}$ and $N^g=\lfloor|\mathcal{S}|/N^{\min}\rfloor$. Now the N^{rest} sensors can be divided into $K\leq N^{rest}$ subgroups. Denote I_k as the size of the kth subgroup, i.e. $\sum_{1\leq k\leq K}I_k=N^{rest}$, where each subgroup are inserted into a established group with size $N^{\min}$. We have the overall overhead of the GAD algorithm:

$$\alpha((N^g-K)(N^{\min})^2+\sum_{k=1}^{K}(N^{\min}+I_k)^2) \tag{13}$$

Obviously, (13) is minimized when $I_k=1$, $\forall k\leq K$, which is achieved by DMGA (Lines 11–14 shown in Fig. 2) □

B. Proof of Part 2

According to the distributed operations of DMGA, at most two messages are transmitted over a hyper-link in each hyper-correlation graph (i.e. a MA and a MR messages, or a MA and a drop messages). For any correlation graph $G_c(\mathbb{G}^n,\mathbb{L}^n,\mathbb{W}^n), 0\leq n\leq\lceil\log_2 N^{\min}\rceil$, we have the number of groups $|\mathbb{G}^n|\leq|\mathcal{S}|/2^n$ and the number of hyper links $|\mathbb{L}^n|=D|\mathbb{G}^n|/2\leq D|\mathcal{S}|/2^{n+1}$, where D is the degree of $G_c(\mathbb{G}^1,\mathbb{L}^1,\mathbb{W}^1)$. Therefore, the per sensor messages produced by DMGA algorithm for all hyper correlation graphs is at most

$$2\sum_{n=0}^{log_2 N^{\min}}|\mathbb{L}^n|/|\mathcal{S}|\leq\sum_{n=0}^{\infty}\frac{D|S|}{2^n}/|\mathcal{S}|=2D$$

which is independent of $|\mathcal{S}|$. □

C. Proof of Part 3

Distributed matching at each hyper correlation graph can achieve at least $1/2$ of the total weights of the optimal [30]. Since DMGA compute the distributed matching for $(\log_2\lceil N^{\min}\rceil$ hyper correlation graphs, at can achieve at least $1/N^{\min}$ of the optimal. □

References

[1] A. Fisher, C. Jacobson, E. Lee, R. Murray, A. Sangiovanni-Vincentelli, and E. Scholte, "Industrial cyber-physical systems - icyphy," in *Complex Systems Design & Management*. Springer International Publishing, 2014, pp. 21–37.

[2] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. IEEE ISORC*, 2008, pp. 363–369.

[3] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3557–3564, 2010.

[4] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” *ACM Transactions on Information and System Security*, vol. 14, no. 1, p. 13, 2011.

[5] V. C. Gungor and G. P. Hancke, “Industrial wireless sensor networks: Challenges, design principles, and technical approaches,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, 2009.

[6] M. Chen, “Reconfiguration of sustainable thermoelectric generation using wireless sensor network,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 2776–2783, 2014.

[7] S.-e. Yoo, P. K. Chong, D. Kim, Y. Doh, M.-L. Pham, E. Choi, and J. Huh, “Guaranteeing real-time services for industrial wireless sensor networks with ieee 802.15. 4,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3868–3876, 2010.

[8] X. Cao, J. Chen, Y. Xiao, and Y. Sun, “Building-environment control with wireless sensor and actuator networks: centralized versus distributed,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3596–3605, 2010.

[9] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, C. A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan *et al.*, “The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms,” *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 556–568, 2008.

[10] B. Lu and V. C. Gungor, “Online and remote motor energy monitoring and fault diagnostics using wireless sensor networks,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4651–4659, 2009.

[11] O. Kreibich, J. Neuzil, and R. Smid, “Quality-based multiple-sensor fusion in an industrial wireless sensor network for mcm,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4903–4911, 2014.

[12] G. Foo, X. Zhang, and D. Vilathgamuwa, “A sensor fault detection and isolation method in interior permanent-magnet synchronous motor drives based on an extended kalman filter,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3485–3495, 2013.

[13] K. Rothenhagen and F. W. Fuchs, “Current sensor fault detection, isolation, and reconfiguration for doubly fed induction generators,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4239–4245, 2009.

[14] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, no. 3, p. 15, 2009.

[15] B. Krishnamachari and S. Iyengar, “Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks,” *IEEE Trans.Comput.*, vol. 53, no. 3, pp. 241–250, 2004.

[16] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, “Online outlier detection in sensor data using non-parametric models,” in *Proc. VLDB*, 2006, pp. 187–198.

[17] O. Obst, X. R. Wang, and M. Prokopenko, “Using echo state networks for anomaly detection in underground coal mines,” in *Proc. IEEE/ACM IPSN*, 2008, pp. 219–229.

[18] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, “Distributed anomaly detection in wireless sensor networks,” in *Proc. IEEE ICCS*, 2006, pp. 1–5.

[19] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta, “In-network outlier detection in wireless sensor networks,” *Knowledge and information systems*, vol. 34, no. 1, pp. 23–54, 2013.

[20] S. Rajasegarar, C. Leckie, and M. Palaniswami, “Hyperspherical cluster based distributed anomaly detection in wireless sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1833–1847, 2014.

[21] S. Chan, H. Wu, and K. Tsui, “Robust recursive eigendecomposition and subspace-based algorithms with application to fault detection in wireless sensor networks,” *IEEE Trans. Instrum. Meas.*, vol. 61, no. 6, pp. 1703–1718, 2012.

[22] R. Jiang, H. Fei, and J. Huan, “A family of joint sparse pca algorithms for anomaly localization in network data streams,” *IEEE Trans. Knowledge Data Eng.*, vol. 25, no. 11, pp. 2421–2433, 2013.

[23] Y. Yao, A. Sharma, L. Golubchik, and R. Govindan, “Online anomaly detection for sensor systems: A simple and efficient approach,” *Performance Evaluation*, vol. 67, no. 11, pp. 1059–1075, 2010.

[24] J. Chen, S. Kher, and A. Somani, “Distributed fault detection of wireless sensor networks,” in *Proc. ACM DIWANS*, 2006, pp. 65–72.

[25] P. Jiang, “A new method for node fault detection in wireless sensor networks,” *Sensors*, vol. 9, no. 2, pp. 1282–1294, 2009.

[26] E. W. Dereszynski and T. G. Dietterich, “Spatiotemporal models for data-anomaly detection in dynamic environmental monitoring campaigns,” *ACM Trans. Sens. Netw. (TOSN*, vol. 8, no. 1, p. 3, 2011.

[27] R. Fontugne, J. Ortiz, N. Tremblay, P. Borgnat, P. Flandrin, K. Fukuda, D. Culler, and H. Esaki, “Strip, bind, and search: a method for identifying abnormal energy consumption in buildings,” in *Proc. ACM SenSys*, 2013, pp. 129–140.

[28] S. Guo, Z. Zhong, and T. He, “Find: faulty node detection for wireless sensor networks,” in *Proc. ACM SenSys*, 2009, pp. 253–266.

[29] J. O. Berger, V. De Oliveira, and B. Sansó, “Objective bayesian analysis of spatially correlated data,” *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1361–1374, 2001.

[30] J.-H. Hoepman, “simple distributed weighted matchings,” *http://arxiv.org/PS_cache/cs/pdf/0410/0410047v1.pdf*, 2004.

[31] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[32] T. Torfs, T. Sterken, S. Brebels, J. Santana, R. van den Hoven, V. Spiering, N. Bertsch, D. Trapani, and D. Zonta, “Low power wireless sensor network for building monitoring,” *IEEE Sensors J.*, vol. 13, no. 3, pp. 909–915, 2013.

[33] I. Urteaga, K. Barnhart, and Q. Han, “Redflag: A run-time, distributed, flexible, lightweight, and generic fault detection service for data-driven wireless sensor applications,” *Pervasive and Mobile Computing*, vol. 5, no. 5, pp. 432–446, 2009.

[34] A. B. Sharma, L. Golubchik, and R. Govindan, “Sensor faults: Detection methods and prevalence in real-world datasets,” *ACM Trans. Sen. Netw. (TOSN)*, vol. 6, no. 3, p. 23, 2010.

[35] H. Farhangi, “The path of the smart grid,” *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18–28, 2010.

[36] Y. Mo, T.-H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, “Cyber–physical security of a smart grid infrastructure,” *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, 2012.

[37] L. Yang and F. Li, “Detecting false data injection in smart grid in-network aggregation,” in *Proc. IEEE SmartGridComm*, 2013, pp. 408–413.