sensors

MDPI

*Article*

# A Framework for Detecting False Data Injection Attacks in Large-Scale Wireless Sensor Networks

**Jiamin Hu [1], Xiaofan Yang [1,*] and Lu-Xing Yang [2]**

[1] School of Big Data & Software Engineering, Chongqing University, Chongqing 400044, China; jiaminhu@cqu.edu.cn
[2] College of Information Technology, Deakin University, Melbourne, VIC 3125, Australia; y.luxing@deakin.edu.au
* Correspondence: xfyang1964@cqu.edu.cn

**Abstract:** False data injection attacks (FDIAs) on sensor networks involve injecting deceptive or malicious data into the sensor readings that cause decision-makers to make incorrect decisions, leading to serious consequences. With the ever-increasing volume of data in large-scale sensor networks, detecting FDIAs in large-scale sensor networks becomes more challenging. In this paper, we propose a framework for the distributed detection of FDIAs in large-scale sensor networks. By extracting the spatiotemporal correlation information from sensor data, the large-scale sensors are categorized into multiple correlation groups. Within each correlation group, an autoregressive integrated moving average (ARIMA) is built to learn the temporal correlation of cross-correlation, and a consistency criterion is established to identify abnormal sensor nodes. The effectiveness of the proposed detection framework is validated based on a real dataset from the U.S. smart grid and simulated under both the simple FDIA and the stealthy FDIA strategies.

**Keywords:** large-scale sensor networks; false data injection attacks; detection framework; distributed solution

check for updates

**Citation:** Hu, J.; Yang, X.; Yang, L.-X. A Framework for Detecting False Data Injection Attacks in Large-Scale Wireless Sensor Networks. *Sensors* **2024**, *24*, 1643. https://doi.org/10.3390/s24051643

Academic Editors: Rogério Dionísio, Pedro M. B. Torres and Carlos Cambra

Received: 10 January 2024
Revised: 25 February 2024
Accepted: 29 February 2024
Published: 2 March 2024

**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Wireless sensor networks (WSNs) consist of spatially dispersed sensors connected via wireless communication protocols [1]. These sensors are equipped with sensing capabilities to collect data on environmental parameters and physical quantities, which are transmitted to a central server or data center for further analysis and decision-making. WSNs are widely employed in various fields, including military affairs, agriculture, healthcare, industrial automation, and intelligent transportation [2].

Typically, the sensors in WSNs are resource-constrained devices in unprotected environments that are vulnerable to physical tampering [3–5]. The behavior of an attacker who physically tampers with sensor data is known as a false data injection attack (FDIA). As a result of the FDIA, the tampered sensors provide misleading data to the central server, leading the system to make incorrect judgments. The FDIA undermines the authenticity of sensor data, which can seriously impact systems that rely on sensor data for decision-making or monitoring, culminating in economic loss or even a life crisis. As a result, it is critical to develop detection mechanisms to ensure that WSNs are resistant to FDIAs [6,7].

### 1.1. Motivation

The focus of this paper is on detecting FDIAs in large-scale WSNs. Our goal is to provide a detection framework for FDIAs with the following properties:

- **Stealthy FDIA detection**. The attacker's purpose is to use resourceful and sophisticated strategies to minimize the risk of being identified. Stealthy FDIAs may be employed, i.e., making the injected false data look as close to the genuine data as

possible, such as by mimicking genuine data distributions and time series patterns. Since stealthy FDIAs are typically not easily observed, the detection framework should take this into account to reduce the likelihood of potential harm.

- **Distribute detection**. The detection process might be centralized or distributed. In centralized detection, all sensor data are sent to a central node for thorough processing. In distributed detection, sensor data are evaluated separately by local sensors or edge devices, making it more responsive to data changes than centralized detection. More significantly, given the widely dispersed sensors and enormous data volumes in large-scale WSNs, distributed detection might be more straightforward to scale.
- **General detection**. Large-scale WSNs are employed in various fields, and the physical behavior of such systems is diverse. Electric power systems, for example, can be defined using circuit equations, whereas thermodynamic systems can be represented using thermodynamic laws. Therefore, a detection framework based only on a measurement that does not require domain-specific a priori knowledge is necessary, which makes the detection method more general and allows for similar detection methods to be applied to sensors in different domains without much adaptation.

### *1.2. Main Contributions*

We propose a correlation-based framework for detecting FDIAs, and our main contributions are sketched below.

- We first develop a grouping approach based on the temporal correlation of the cross-correlation between the time-series signals of pairwise sensors. All sensors are categorized into multiple correlated groups, and subsequent detection methods are performed separately within the groups.
- We build an autoregressive integrated moving average (ARIMA) model for predicting future data from each sensor using historical time-series signals, which is used to learn the normal temporal correlation of the cross-correlation between data reported by pairwise sensors.
- Based on the comparison of the normal and actual temporal correlation of the cross-correlation within each group, the basis for determining the consistency of the pairwise sensor data is established. Then, majority voting is executed within each group to identify the abnormal sensors.
- To verify the performance of the detection framework, we construct simple FDIAs and stealthy FDIAs in a genuine sensor dataset. The effectiveness of our proposed detection framework is verified through extensive simulation experiments.

The subsequent materials are organized in this fashion: Section 2 reviews the related works. In Section 3, sensor data and correlation definitions are introduced. The detection framework is described in detail in Section 4, and the performance of the detection framework is corroborated through simulation experiments in Section 5. Finally, this work is summarized in Section 6.

## 2. Related Work

In this section, we make comments on the previous work that is related to the present paper, aiming to highlight the novelty of our work. Detecting FDIAs in sensors has received considerable attention. In this section, we categorize the existing related works into three research directions related to FDIA detection: FDIA detection methods and FDIA types.

### *2.1. FDIA Detection Methods*

Recent studies have been conducted to detect FDIAs on sensors by modeling the physical behavior of the system. In general, the physical behavior of the system is established based on physical equations (fluid dynamics, electromagnetic laws, etc.) to predict the sensor data, and then the predicted data are compared to the actual data [8]. Some attempts have been made to build predictive models to detect FDIAs through the dynamical equations of smart grids [9,10], unmanned aerial vehicles [11], water distribution systems [12],

and cyber–physical systems [13–15]. However, this detection approach requires appropriate predictive models for specific domains and relies on a priori knowledge of specific physical behaviors, which allows for limited scalability.

Subsequent studies have explored techniques for detecting FDIAs from sensor measurements, with the majority of these works based on exploring inter-measurement correlations. Illiano et al. [16] presented an approach to detecting FDIAs in WSNs that combines measurement checks and authentication strategies. Aboelwafa et al. [17] addressed an approach to detecting FDIAs in the industrial Internet of Things that exploits sensor data correlation in time and space. Martovytskyi et al. [18] explored the method of FDIA detection, which is based on spatiotemporal correlation in smart grids. Berjab et al. [19] presented a method for detecting FDIAs in WSNs, which uses observed spatiotemporal and multivariate attribute sensor correlations. Huang et al. [20] addressed the problem of detecting FDIAs in dynamic WSNs based on spatial correlation. Based on the spatiotemporal correlation, Hu et al. [21] explored the idea of fault diagnosis to detect collusive FDIAs in WSNs. However, these efforts depend on centralized detection, increasing the complexity and cost of detection systems in the face of increasing data volumes.

In contrast, distributed detection methods can be more easily scaled to large-scale sensor networks. Chen et al. [22] built distributed real-time detection algorithms based on spatiotemporal correlation to detect FDIAs in large-scale networked industrial sensing systems. Islam et al. [23] utilized distributed algorithms based on spatiotemporal correlation to detect data anomalies in large-scale intelligent transportation systems. Lai et al. [24] suggested a distributed approach to detecting FDIAs in WSNs using temporal, spatial, and event-based correlation. In this paper, our framework is based on a distributed approach, where detection methods can be executed at separate edge devices to reduce the network pressure associated with processing data generated by large-scale sensors.

### *2.2. FDIA Types*

Another crucial consideration in FDIA detection is the type of attack. An adversary may employ simple attacks, such as randomly injecting high outliers and injecting false data with a common strategy. An adversary may employ stealthy attacks, such as constructing coherent attack signals. Most of the works [17–20,22–24] mentioned based on sensor measurements themselves are effective in detecting simple FDIAs, but not stealthy ones. For instance, in [22], based on the spatiotemporal correlation of sensor data, the authors used exponential weighted moving average and principal component analysis to establish a rotated ellipse area for each pair of sensors in a correlation group and detected FDIAs by determining whether the current sensor readings for each pair of sensors were located within the corresponding area of the rotated ellipse. Assuming that an attacker employs a collusive strategy whereby the current anomalous readings of a pair of sensors are also located within the corresponding area of the rotated ellipse, this may result in a false alarm.

While some works [16,21] have considered the collusive scenario, further development is needed for when an attacker employs stealthy attacks that construct coherent attack signals (mimicking genuine data distributions and time series patterns). Therefore, in this paper, we propose a generalized detection framework that can be used to detect FDIAs in large-scale WSNs, including stealthy FDIAs. The approach we propose in this paper to meet these requirements, together with the previously mentioned works, is summarized in Table 1.

**Table 1.** Comparison between approaches.

| Research Works | FDIA Detection Methods | | FDIA Types | | |
|---|---|---|---|---|---|
| | General Detection | Distributed Detection | Simple FDIAs | Collusive FDIAs | Stealthy FDIAs |
| Illiano et al. [16] | yes | no | yes | yes | no |
| Aboelwafa et al. [17] | yes | no | yes | no | no |
| Martovytskyi et al. [18] | yes | no | yes | no | no |

**Table 1.** *Cont.*

| Research Works | FDIA Detection Methods | | FDIA Types | | |
|---|---|---|---|---|---|
| | General Detection | Distributed Detection | Simple FDIAs | Collusive FDIAs | Stealthy FDIAs |
| Berjab et al. [19] | yes | no | yes | no | no |
| Huang et al. [20] | yes | no | yes | no | no |
| Hu et al. [21] | yes | no | yes | yes | no |
| Chen et al. [22] | yes | yes | yes | no | no |
| Islam et al. [23] | yes | yes | yes | no | no |
| Lai et al. [24] | yes | yes | yes | no | no |
| Our approach | yes | yes | yes | yes | yes |

## 3. Preliminaries

We extract information from the sensor data itself to detect FDIAs. In this section, we discuss the definition of sensor data and the correlation between sensor data.

### *3.1. Sensor Data*

Consider a set of sensors $V = \{v_1, \cdots, v_N\}$ distributed over a geographic area, where each sensor $v_i \in V$ collects one type of environmental data in synchronization with the other sensors. Let $r_i(t)$ denote the sensor measurement reported by $v_i$ at time $t$, as follows:

$$r_i(t) = \tilde{r}_i(t) + \epsilon(t), \tag{1}$$

where $\tilde{r}_i(t)$ is the true value and $\epsilon(t)$ is an error at time $t$. The error $\epsilon(t)$ can be caused by either a random error or a systematic error. A random error is an uncertainty in the measurement result caused by various random factors (e.g., noise), and a systematic error is an uncertainty in the measurement result due to inherent defects or biases (e.g., faults, FDIAs). Since our work focuses on detecting FDIAs on sensors, we only consider systematic errors caused by FDIAs. The collection of $r_i(t)$ from $v_i$ over a period of time is a time-series signal [25]. A time-series signal consisting of $t$ successive sensor measurements $r_i(1), r_i(2), \cdots, r_i(t)$ can be expressed as follows:

$$R_i(t) = \{r_i(1), r_i(2), \cdots, r_i(t)\}. \tag{2}$$

### *3.2. Spatiotemporal Correlation between Sensor Data*

Spatiotemporal correlation is a combination of spatial and temporal correlation, referring to the simultaneous existence of correlations in space and time. The correlation of sensor data exists because sensors are distributed in space and measure time-dependent physical phenomena. The anomalous data generated when an FDIA occurs can go so far as to cause this correlation to be disrupted, so we can identify false data injection attacks by analyzing the correlation of sensor data [26].

#### 3.2.1. Spatial Correlation

Spatial correlation between sensor data over a fixed time interval reveals the degree of association between events or phenomena at adjacent or discrete locations in space. For example, in a smart grid, neighboring industrial facilities may belong to similar industries and, thus, have a similar electricity demand, resulting in a strong spatial correlation between meter data in industrial areas, but there may be a weak spatial correlation between meter data in industrial areas and meter data in residential areas.

#### 3.2.2. Temporal Correlation

The temporal correlation of sensor data reveals the degree of association between events or phenomena over time. For example, in a smart grid, due to differences between day and night, seasonal factors, etc., by observing hourly, daily, weekly, or seasonal data

from meters, it is possible to find repeating patterns or regularities in the use of electrical energy on different time scales.

## 4. FDIA Detection Framework

In this section, this paper proposes a framework for FDIA detection. This framework consists of three phases: correlation grouping, correlation prediction, and correlation testing, stated as follows:

- *Phase I: Correlation grouping*. The purpose of this phase is to group $V$ in a large-scale WSN based on historical sensor data so that sensors in the same group are highly correlated with other sensors.
- *Phase II: Correlation prediction*. The purpose of this phase is to predict the normal temporal correlation of the cross-correlation between pairwise sensor measurements in the same group over a short period of time in the future.
- *Phase III: Correlation testing*. The purpose of this phase is to test the actual sensor data based on the predicted normal temporal and spatial correlations.

The flow diagram for FDIA detection in large-scale WSNs is shown in Figure 1. Next, let us discuss the three phases in detail.

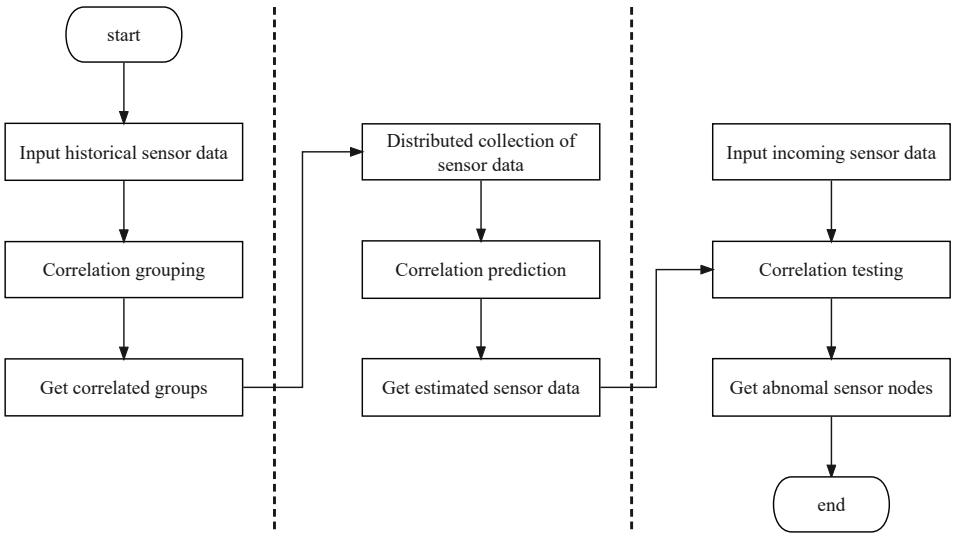



**Figure 1.** The flow diagram for FDIA detection in large-scale WSNs.

### *4.1. Correlation Grouping*

Collect sensor data, ensuring that the data are collected at the same or similar frequencies, and pre-process the data if necessary, including de-noise, filling in missing values, interpolating, and other operations to facilitate analysis. Standardize the sensor data (e.g., min-max normalization, z-score normalization) to ensure that the measurements from different sensors are similarly scaled so that the magnitude of the change in one sensor does not affect the cross-correlation results.

Let $R_i(T) = \{r_i(1), r_i(2), \cdots, r_i(T)\}$ denote the Historical Time-series Signal (HTS) of $v_i$ obtained after data processing, where $T$ denotes the length of the HTS. The cross-correlation of any two full HTSs is usually calculated to determine the spatial correlation between $R_i(T)$ and $R_j(T)$, expressed as follows:

$$C_{ij} = \frac{cov(R_i(T), R_j(T))}{\delta(R_i(T))\delta(R_j(T))}, \tag{3}$$

where

$$cov(R_i(T), R_j(T)) = \frac{1}{T}\sum_{t=1}^{T}(r_i(t) - \bar{r_i})(r_j(t+\tau) - \bar{r_j}) \tag{4}$$

denotes the covariance between $R_i(T)$ and $R_j(T)$, $\delta(R_i(T)) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(r_i(t) - \bar{r}_i)^2}$, and $\delta(R_j(T)) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(r_j(t+\tau) - \bar{r}_j)^2}$ denote the standard deviations of $R_i(T)$ and $R_j(T)$, respectively. $C_{ij}$ denotes the correlation coefficients of $v_i$ and $v_j$ at lag $\tau$; $\bar{r}_i$ and $\bar{r}_j$ represent the average values of two full HTSs from $v_i$ and $v_j$. The lag $\tau$ represents the delay of one HTS with respect to the other, and by analyzing the peak of cross-correlation, it is determined at which lag value the correlation between the two HTSs is greatest. $C_{ij}$ has a value between 1 and $-1$, where 1 means perfect positive correlation, $-1$ means perfect negative correlation, and 0 means the signals are uncorrelated [27].

However, this paper's goal is to extract the temporal correlation of the cross-correlation between any two HTSs, so the sliding window cross-correlations need to be computed.

First, let the size of the sliding window be $k$. The $w$th sub-signal, consisting of $k$ successive sensor measurements $r_i(w), \cdots, r_i(w+k-1)$ within $[1, t]$, can be defined as

$$R_i(t, w) = \{r_i(w), \cdots, r_i(w+k-1)\}. \tag{5}$$

Therefore, the HTS of $v_i$ is segmented into multiple *historical sub-signals*, denoted as $\mathbb{R}_i = \{R_i(T,1), R_i(T,2), \cdots, R_i(T,W)\}$, where $W$ denotes the number of historical sub-signals.

Second, for $\mathbb{R}_i$ and $\mathbb{R}_j$, the cross-correlation is computed within each sliding window, denoted as

$$\mathbb{C}_{ij}(w) = \frac{cov(R_i(T,w), R_j(T,w))}{\delta(R_i(T,w))\delta(R_j(T,w))}, \quad w = 1, 2, \cdots, W. \tag{6}$$

Here, $cov(R_i(T,w), R_j(T,w))$ represents the covariance between $R_i(T,w)$ and $R_j(T,w)$; $\delta(R_i(T,w))$ and $\delta(R_j(T,w))$ represent the standard deviations of $R_i(T,w)$ and $R_j(T,w)$, respectively. Then, the time series of the cross-correlation of $v_i$ and $v_j$ can be represented by

$$\mathbb{C}_{ij} = \{\mathbb{C}_{ij}(1), \mathbb{C}_{ij}(2), \cdots, \mathbb{C}_{ij}(W)\}, \quad w = 1, 2, \cdots, W. \tag{7}$$

Finally, we pick $\mathbb{C}_{ij}$ with a positive correlation for $K$-means clustering, which is one of the most widely used parameter selection methods. After $K$-means clustering, the sensors can be categorized into multiple correlation groups.

For a dataset with $M$ time series of cross-correlation, we represent $\mathbb{C}_{ij}$ with a positive correlation as a feature vector. We extract relevant features that capture the characteristics of the time series; commonly used features include mean, standard deviation, slope, etc. Each $\mathbb{C}_{ij}$ with a positive correlation is represented as a feature vector $\mathbf{v}_p = [f_1, f_2, \cdots, f_k]$ $(p = 1, \cdots, M)$, where $k$ is the number of features and $\mathbf{v}_p$ includes all necessary extracted features (mean, standard deviation, slope, etc.). The random cluster centers $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K$ are first selected, and then the $K$-means objection function is defined as follows:

$$J = \sum_{p=1}^{M}\sum_{q=1}^{K} x_{pq} \cdot \|\mathbf{v}_p - \mathbf{u}_q\|^2, \tag{8}$$

where $x_{pq}$ is an indicator function indicating if the time series $p$ belongs to cluster $q$ $(q = 1, 2, \cdots, K)$, and $\|\cdot\|^2$ denotes the squared Euclidean distance.

We update the centroids of the clusters by calculating the mean feature vector for each cluster:

$$\mathbf{u}_q = \frac{1}{|C_q|}\sum_{p \in C_q} \mathbf{v}_p, \tag{9}$$

where $|C_q|$ is the number of time series in cluster $q$. We repeat the centroid's update and minimization of $J$ until convergence. Then,

**Definition 1.** *Let $V_i^q = \{v_j | \mathbb{C}_{ij} \in clustr\ q, j \neq i\}$ be the set of sensors consistent with sensor $v_i$ in cluster $q$ obtained according to HTSs, and let $V^q = \{v_i | \frac{|V_i^q|}{N-1} > 50\%\}$ be the set of sensors that are grouped in $q$ according to HTSs.*

**Remark 1.** *Figure 2 illustrates the correlation grouping of four sensor nodes. After correlation grouping, each group's sensor data can be sent to a separate edge device for distributed processing to reduce network pressure and improve processing efficiency [28]. The following stages are performed within each group: correlation prediction and correlation testing.*

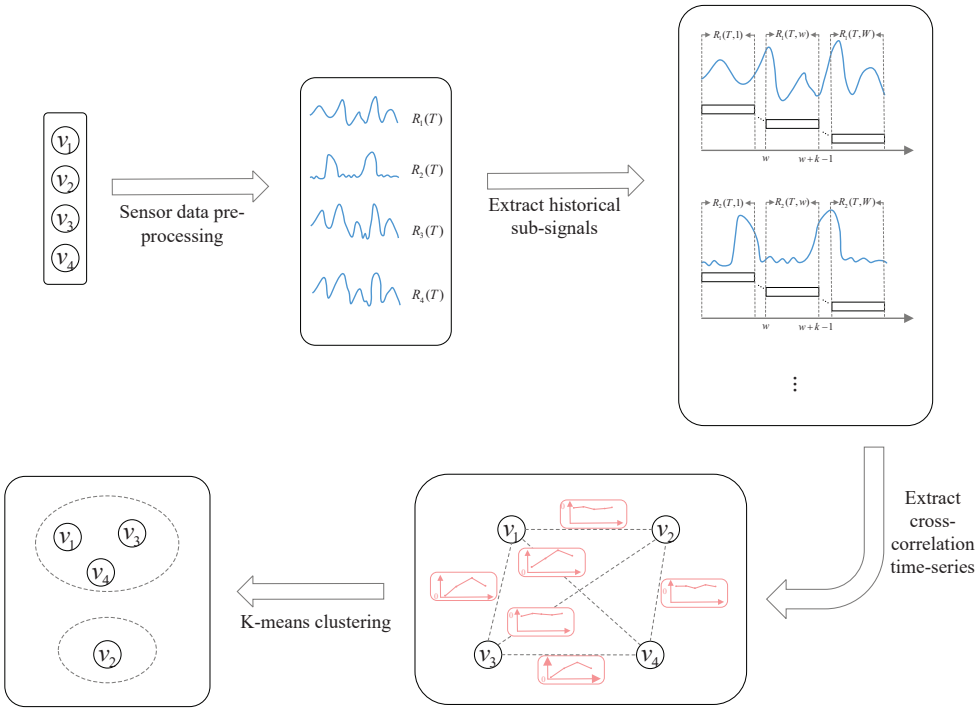



**Figure 2.** The correlation grouping of four sensor nodes.

### *4.2. Correlation Prediction*

Next, we predict the normal temporal correlation of cross-correlation between pairwise sensor measurements in each group over a short period of time in the future.

Consider pairwise sensors $v_i$ and $v_j$ in a group. As we discussed in the previous subsection, the measurements of $v_i$ and $v_j$ should be temporally correlated with their previous measurements. Therefore, this subsection uses the Autoregressive Integrated Moving Average (ARIMA) model to predict the future time-series signal of each sensor based on the HTS, which is referred to as the Estimated Time-series Signal (ETS). ARIMA is used as a time series predictive analysis method, which requires only historical data to make predictions and has the ability to be widely applied to a wide range of time series data.

ARIMA combines the concepts of autoregression (AR), moving average (MA), and the operation of differencing the time series signals. Specifically, the autoregressive part represents the relationship between the current value of a variable and its value at $p'$ previous moments, where $p'$ denotes the autoregressive order. The moving average part represents the relationship between the current value and the error (white noise) at $q'$ previous moments, where $q'$ denotes the moving average order. The $d$-order differencing operation is performed to remove trends and seasonality from HTSs. Therefore, an ARIMA model is used to fit the trend and periodicity of the HTS by choosing appropriate parameters $(p', d, q')$ to make forecasts of the ETS [29].

First, a suitable $d$ is chosen using the following difference method:

$$\Delta^d r_i(t) = \Delta(\Delta^{d-1} r_i(t)), \tag{10}$$

where $\Delta r_i(t) = r_i(t) - r_i(t-1)$ denotes the first-order difference at time point $t$. The suitable value is $d$ when the sequence after $d$-order differencing of the HTS passes the Augmented Dickey–Fuller (ADF) test [30].

Second, for all possible combinations of $p'$ and $q'$, an ARIMA model is fitted using the information criterion (AIC) to select the best combination of $p'$ and $q'$ as the one with the smallest AIC value. The formula for calculating AIC is as follows:

$$AIC = -2ln(L) + 2l, \tag{11}$$

where $L$ is the maximum likelihood estimate of the model, and $l$ is the number of parameters of the model.

Third, the HTS is fitted using an ARIMA model with order $(p', d, q')$, which is formulated as follows:

$$\Delta^d r_i(t) = \mu + \sum_{l=1}^{p'} \phi_l \Delta^d r_i(t-l) + \sum_{l=1}^{q'} \psi_l \epsilon(t-l) + \epsilon(t), \tag{12}$$

where $\mu$, $\phi_l$, and $\psi_l$ are model parameters, and $\epsilon(t)$ stands for the value of the independent error at time $t$, which follows a Gaussian distribution with a zero mean. The fitted model is tested to see if it matches the characteristics of the data, including the autocorrelation and partial autocorrelation of the residuals and normality of the residuals.

Finally, assuming that the fitted model is used to predict future data, the ETSs can be made by the difference restoration of predicted data. An estimated time-series signal consisting of $t$ successive sensor measurements $r'_i(1), r'_i(2), \cdots, r'_i(t)$ can be expressed as follows:

$$R'_i(t) = \{r'_i(1), r'_i(2), \cdots, r'_i(t)\}. \tag{13}$$

Then, let $R'_i(S) = \{r'_i(1), r'_i(2), \cdots, r'_i(S)\}$ denote the ETS of $v_i$, where $S$ denotes the length of the ETS.

The ETS and HTS are concatenated into a new time-series signal $X_i(t)$, which consists of $t$ successive sensor measurements and can be expressed as follows:

$$X_i(t) = \begin{cases} R_i(t), & t \leq T, \\ \{R_i(T), R'_i(t-T)\}, & t > T. \end{cases} \tag{14}$$

Similar to Equation (6), the $w$th *estimated sub-signal* consisting of $k$ successive sensor measurements within $[1, t]$ can be defined as $X_i(t, w)$. $\mathbb{C}'_{ij} = \{\mathbb{C}_{ij}(1), \mathbb{C}_{ij}(2) \cdots, \mathbb{C}_{ij}(S)\}$ is calculated in the same manner as in Equation (7) based on $X_i(t, w)$ and $X_j(t, w)$, where $w = 1, 2, \cdots, S$.

Therefore, $\mathbb{C}'_{ij}$ represents the normal temporal correlation of cross-correlation between pairwise sensor measurements in a group. The diagram of correlation prediction within a group is shown in Figure 3.

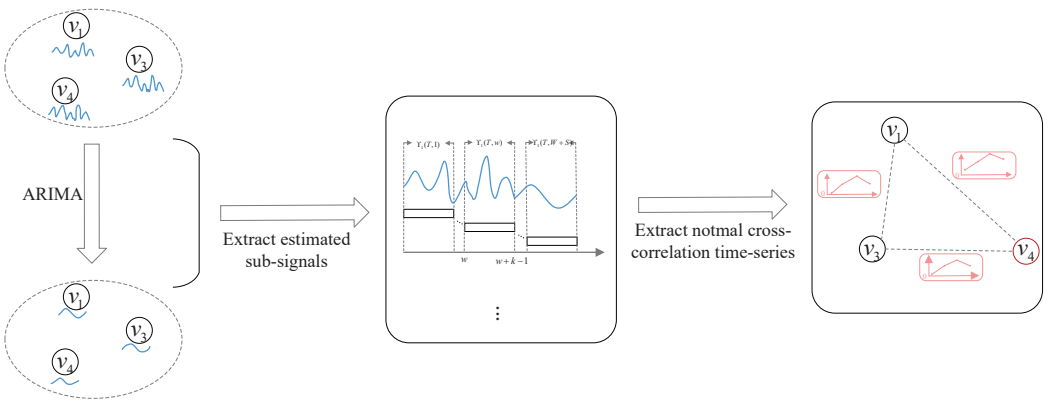



**Figure 3.** The diagram of correlation prediction within a group.

*4.3. Correlation Testing*

After correlation prediction, we compare $\mathbb{C}'_{ij}$ with the actual ones to detect FDIAs in this subsection.

An actual time-series signal consisting of $t$ successive sensor measurements $r_i^*(1), r_i^*(2), \cdots, r_i^*(t)$ can be expressed as follows:

$$R_i^*(t) = \{r_i^*(1), r_i^*(2), \cdots, r_i^*(t)\}. \tag{15}$$

Then, let $R_i^*(S) = \{r_i^*(1), r_i^*(2), \cdots, r_i^*(S)\}$ denote the Actual Time-series Signal (ATS) of $v_i$, where $S$ denotes the length of the ATS.

The ATS and HTS are concatenated into a new time-series signal $Y_i(t)$, which consists of $t$ successive sensor measurements and can be expressed as follows:

$$Y_i(t) = \begin{cases} R_i(t), & t \leq T, \\ \{R_i(T), R_i^*(t-T)\}, & t > T. \end{cases} \tag{16}$$

Similar to Equation (6), the $w$th *actual sub-signal*, which consists of $k$ successive sensor measurements within $[1, t]$, can be defined as $Y_i(t, w)$. Consider each pair of $v_i$ and $v_j$ in a group; $\mathbb{C}_{ij}^* = \{\mathbb{C}_{ij}(1), \mathbb{C}_{ij}(2), \cdots, \mathbb{C}_{ij}(S)\}$ is calculated in the same manner as in Equation (7) based on $Y_i(t, w)$ and $Y_j(t, w)$, where $w = 1, 2, \cdots, S$.

Therefore, $\mathbb{C}_{ij}^*$ represents the actual temporal correlation of cross-correlation between pairwise sensor measurements in a group.

Consider each pair of $v_i$ and $v_j$ in the group $q \subseteq V$. Based on $\mathbb{C}'_{ij}$ and $\mathbb{C}_{ij}^*$, we have

$$\rho_{ij} = \frac{cov(\mathbb{C}'_{ij}, \mathbb{C}_{ij}^*)}{\delta(\mathbb{C}'_{ij})\delta(\mathbb{C}_{ij}^*)}, \tag{17}$$

where $cov(\mathbb{C}'_{ij}, \mathbb{C}_{ij}^*)$ is the covariance of $\mathbb{C}'_{ij}$ and $\mathbb{C}_{ij}^*$; $\delta(\mathbb{C}'_{ij})$ and $\delta(\mathbb{C}_{ij}^*)$ are the standard deviations of $\mathbb{C}'_{ij}$ and $\mathbb{C}_{ij}^*$, respectively. Then, we have

$$e_{ij} = \begin{cases} 1, & if \rho_{ij} > \theta, \\ 0, & otherwise, \end{cases} \tag{18}$$

where $e_{ij}$ is for the *consistency criterion*, and $e_{ij} = 1 (resp. 0)$ denotes that $v_i$ and $v_j$ are consistent (resp. inconsistent).

**Remark 2.** *The choice of threshold $\theta$ depends on the experience of $\mathbb{C}_{ij}$ in Phase I, and the performance of the model on the test set can be observed by trying different thresholds and selecting the one with the best performance.*

**Definition 2.** *Let $N_i^a = \{v_j | e_{ij} = 1 \text{ or } e_{ij} = 0, i \neq j\}$ be the set of all neighbors of $v_i$, and $N_i^c = \{v_j | e_{ij} = 1, i \neq j\}$ be the set of consistent neighbors of $v_i$ obtained according to the comparison of ETSs and ATSs.*

**Definition 3.** *Let $N^t = \{v_i \in q | \frac{|N_i^c|}{|N_i^a|} \geq 50\%\}$ be the set of all trusted neighbors.*

So, let $N^F = \{v_i \in q | \frac{|N^t \bigcap N_i^a|}{|N_i^a|} \geq 50\% \text{ and } \frac{|N^t \bigcap N_i^a \bigcap N_i^c|}{|N^t \bigcap N_i^a|} < 50\%\}$ be the set of abnormal nodes in group $q$. The diagram of correlation testing within a group is shown in Figure 4.

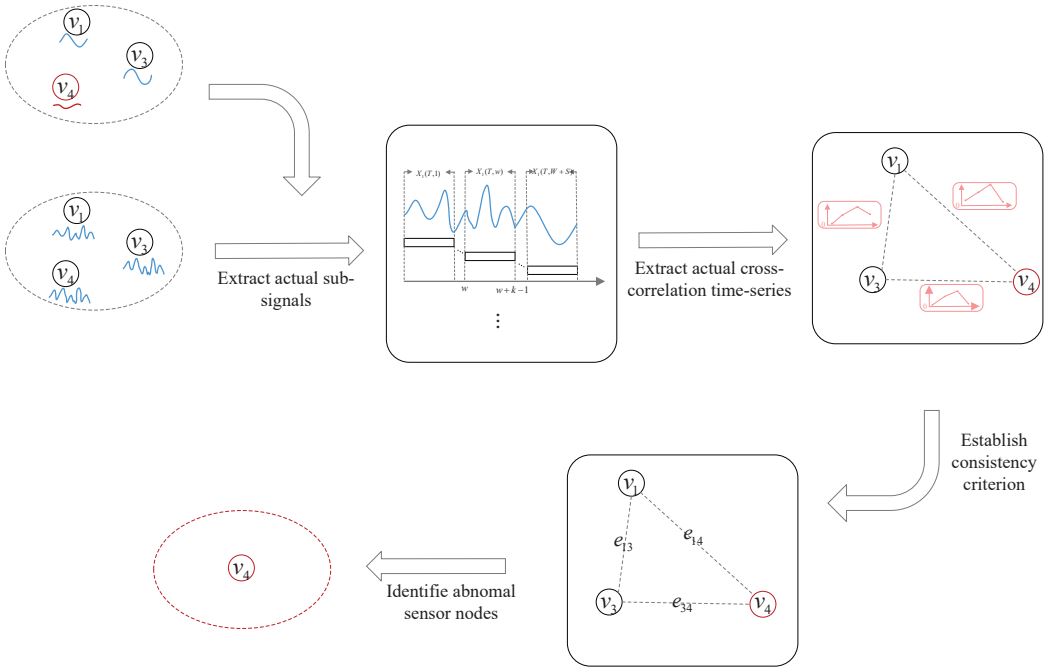



**Figure 4.** The diagram of correlation testing within a group.

## 5. Effectiveness of the Proposed Framework

This section is devoted to investigating the effectiveness of the framework through simulation experiments.

### *5.1. Experiment Preparation*

We applied the detection framework to an hourly electricity demand dataset by subregion, which was based on the 2020 US Energy Information Administration State Electricity Profiles (available at http://www.eia.gov/, accessed on 2 June 2023). This dataset was chosen because it was derived from widely distributed smart meters to evaluate the effectiveness of our proposed framework in detecting FDIAs.

By visualizing the dataset, we found that the time series data show a pronounced periodicity with a period length of 24. Therefore, the model parameters used for this dataset were obtained through observation and manual grid search, as shown in Table 2.

**Table 2.** Model parameters used on the dataset.

| Parameters | Value |
|---|---|
| The HTSs' size $T$ | 4246 h |
| The ETSs' size $S$ | 120 h |
| Sliding window size $k$ | 720 h |
| Threshold $\theta$ | 0 |

### *5.2. Experiments and Analysis of Experimental Results*

Figure 5 illustrates the results for one of the groups, consisting of a set of sensors $\{v_1, v_2, \cdots, v_7\}$, after correlation grouping and data fitting for HTSs. In Figure 5, we visualize only the data points with a step size of 24 to display the fitting results clearly. As can be seen from the figure, there is a strong correlation between the HTSs within a group, and our approach effectively fits the HTSs.

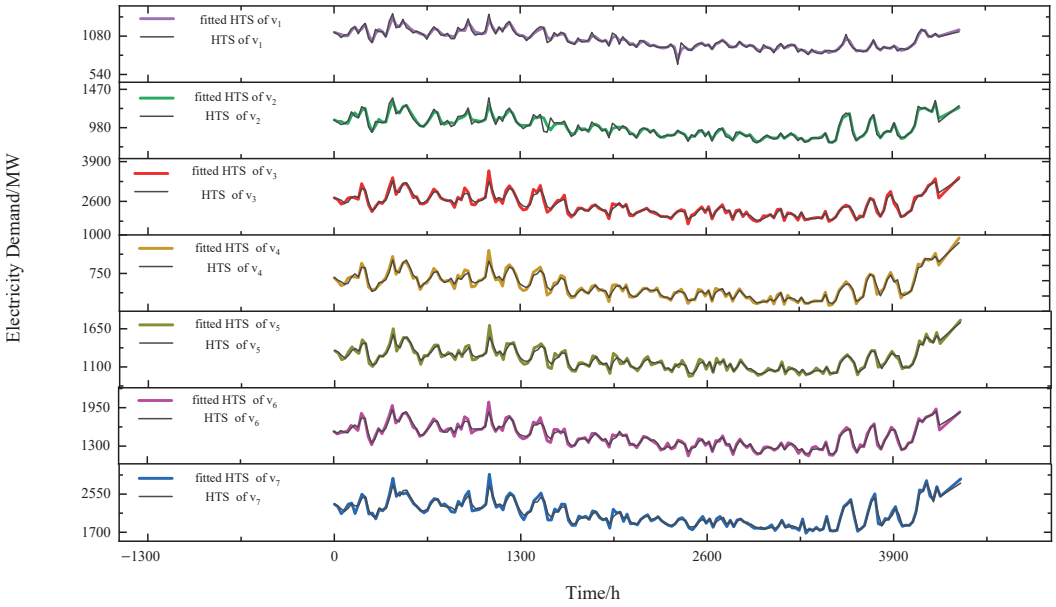



**Figure 5.** The results for one of the groups after correlation grouping and data fitting for HTSs.

Figure 6 illustrates the comparison results of ETSs and ATSs for the group after correlation prediction. It is seen that our approach can effectively predict future data.

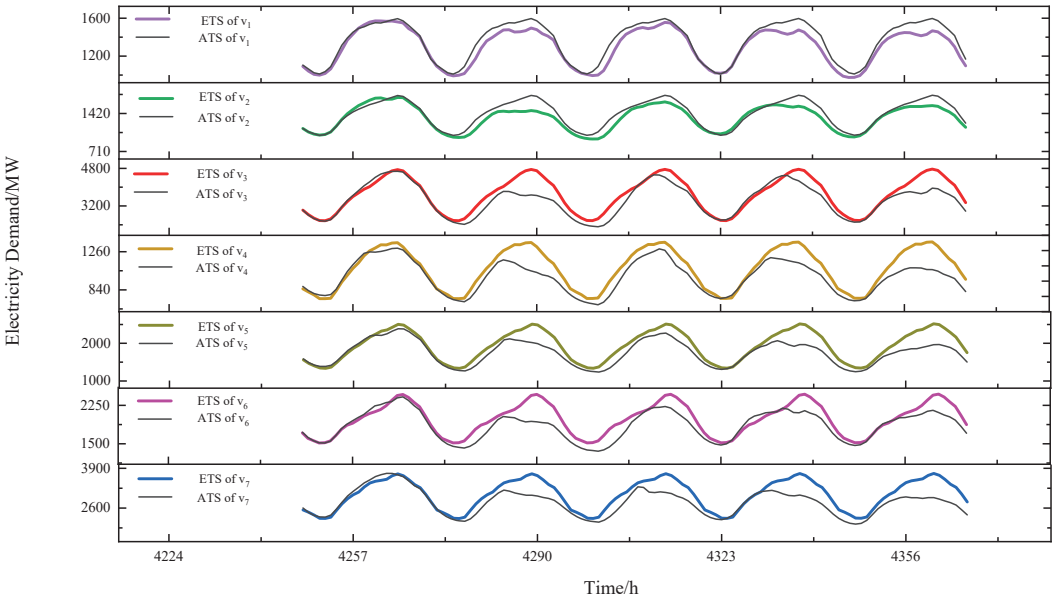



**Figure 6.** The comparison results of ETSs and ATSs for the group after correlation prediction.

To further validate the effectiveness of our framework for detecting FDIAs, we performed correlation testing for various FDIA strategies on target signals. Moreover, we compared our approach with the SCCR solution given in previous work [18], where the SCCR is a consistent ellipse area formed by spatiotemporal correlations. In our experiments, the confidence degree of the consistency ellipse was set to 95%.

In addition, we used three different metrics: successful detection rate, false-negative detection rate, and false-positive detection rate. The successful detection rate is the proportion of actual abnormal nodes that are correctly identified; the false-negative detection rate is the proportion of actual abnormal nodes that are incorrectly identified as normal; and the false-positive detection rate is the proportion of actual normal nodes that are incorrectly identified as abnormal nodes.

5.2.1. The Simple FDIA

A simple FDIA means randomly generating an attack signal. Assuming that $r_3(t)$ is chosen as the target of the attack of the group, the power demand of $v_3$ is randomly increased by 50%, as shown in Figure 7.

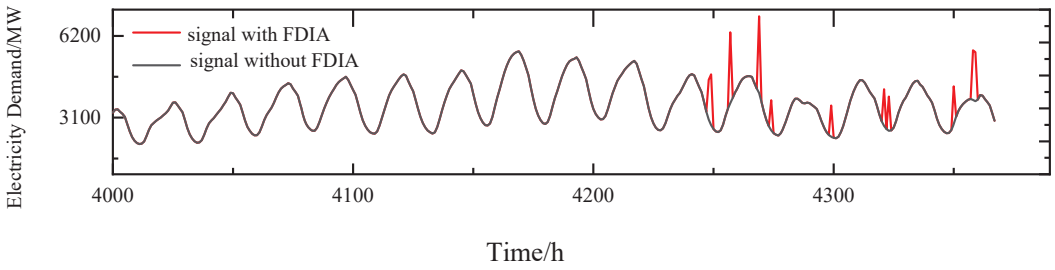



**Figure 7.** The simple FDIA on $v_3$.

In our solution, Figure 8 and the second line of Table 3 show the results of comparing the normal temporal correlation of cross-correlation with the actual temporal correlation of cross-correlation of $v_3$ within the group in a simple FDIA. As shown in Figure 8, from the start of the FDIA, the change in the trend of $\mathbb{C}^*_{ij}$ relative to the trend of $\mathbb{C}'_{ij}$ is clearly inconsistent. In the SCCR solution, we can also observe the inconsistency of $v_3$ with other nodes. The proposed framework and SCCR solution are able to accurately detect the simple FDIA on $v_3$.

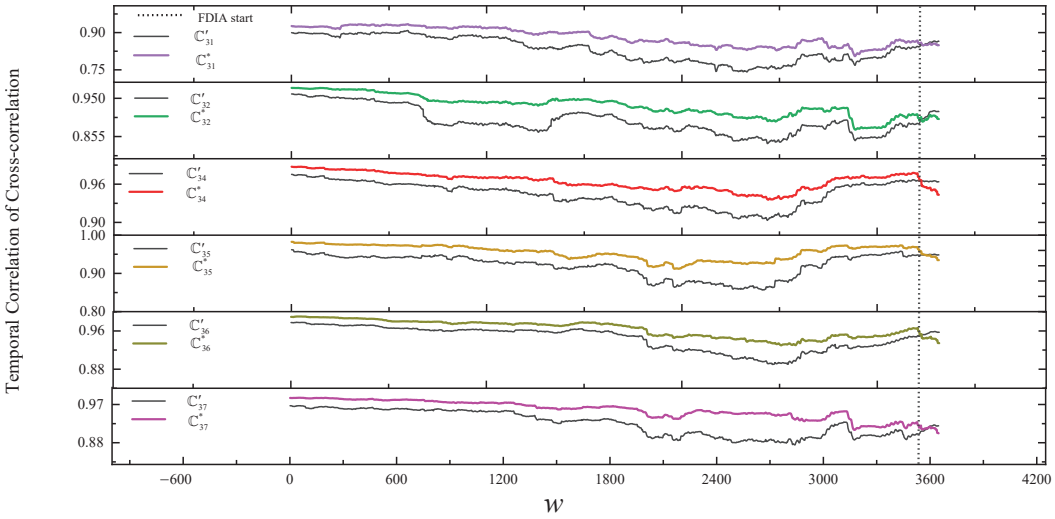



**Figure 8.** The results of comparing the normal temporal correlation of cross-correlation with the actual temporal correlation of cross-correlation of $v_3$ within the group in a simple FDIA.

**Table 3.** The comparison results of the temporal correlation of cross-correlation.

| **Fdia Type** | $\rho_{31}$ | $\rho_{32}$ | $\rho_{34}$ | $\rho_{35}$ | $\rho_{36}$ | $\rho_{37}$ |
|---|---|---|---|---|---|---|
| Simple FDIA | −0.18 | 0.15 | 0.10 | −0.48 | −0.66 | −0.44 |
| Stealthy FDIA | −0.60 | −0.67 | 0.03 | −0.21 | −0.79 | −0.71 |
| Stealthy and collusive FDIA | −0.60 | 0.98 | 0.03 | −0.21 | −0.79 | −0.71 |

We conducted a total of 100 similar experiments in all groups, in which the framework proposed in this paper and the SCCR solution were able to detect at least 99% of FDIAs (Figure 9a), and the false-negative detection rate (Figure 9b) and false-positive detection rate (Figure 9c) were almost zero. Therefore, we conclude that, in general, the framework proposed in this paper performs well in detecting simple FDIAs.

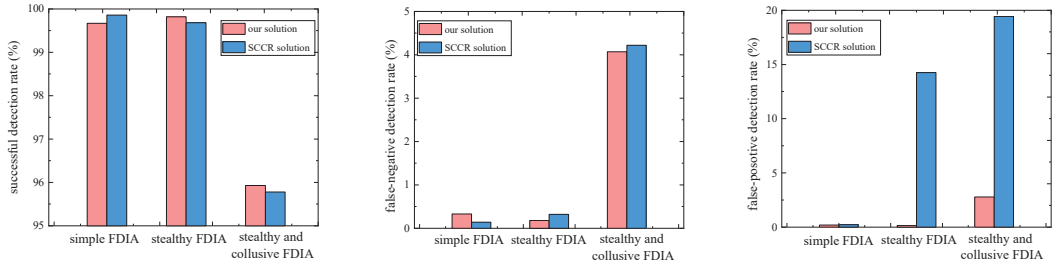



**Figure 9.** The comparison results of three metrics: (**a**) successful detection rate; (**b**) false-negative detection rate; (**c**) false-positive detection rate.

#### 5.2.2. The Stealthy FDIA

The stealthy FDIA means the attacker injects in a well-designed way that is generally not easily observable. Assuming that $r_3(t)$ is chosen as the target of the attack and that the attacker is able to learn the time series of $v_3$ of the group, in this case, the power demand of $v_3$ slowly increases within the detected threshold (boiling frog attack [31]) and also exhibits periodicity from $t = 4247$ h, as shown in Figure 10.

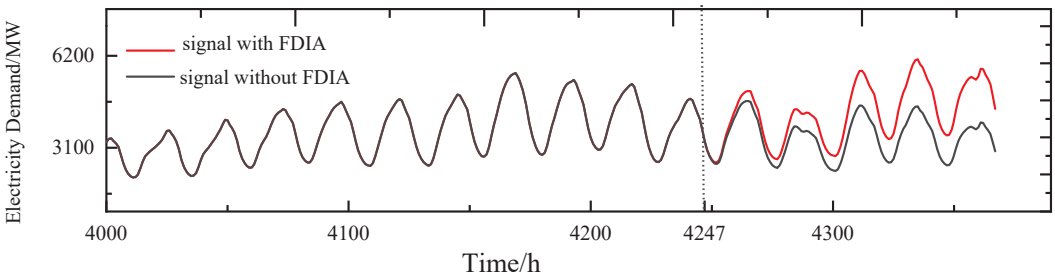



**Figure 10.** The stealth FDIA on $v_3$.

In our solution, Figure 11 and the third line of Table 3 show the results of comparing the normal temporal correlation of cross-correlation with the actual temporal correlation of cross-correlation of $v_3$ within the group in stealthy FDIA. As shown in Figure 11, from the start of the FDIA, the change in the trend of $\mathbb{C}^*_{ij}$ relative to the trend of $\mathbb{C}'_{ij}$ is gradually inconsistent. However, in the SCCR solution, we do not identify any outliers during the first 66 h of the FDIA, after which the abnomal nodes $v_3$, $v_6$, and $v_7$ are identified. This result is caused by the fact that at the beginning of the FDIA, the outliers are within the detection threshold of the SCCR solution, leading to unrecognized anomalies, which are then considered normal to build the consistency ellipse, resulting in a high rate of false positives.

We conducted a total of 100 similar experiments in all groups, in which both the framework proposed in this paper and the SCCR solution were able to detect at least 99% of FDIAs (Figure 9a), and the false-negative detection rate was almost zero (Figure 9b); the false-positive detection rate for the framework proposed in this paper was almost zero, while the false-positive detection rate for the SCCR solution was up to 14% (Figure 9c). In addition, we observed that long-term attack signals resulted in stronger inconsistencies than short-term attack signals. Therefore, we conclude that, in general, the framework proposed in this paper performs well for detecting stealthy FDIAs on a single sensor, and our approach is superior compared to the SCCR solution in detecting long-term and stealthy attack signals. However, the inconsistency was not obvious from the beginning of the FDIA. Therefore, it is necessary to choose a suitable ETS size or sliding window size when detecting stealthy FDIAs.

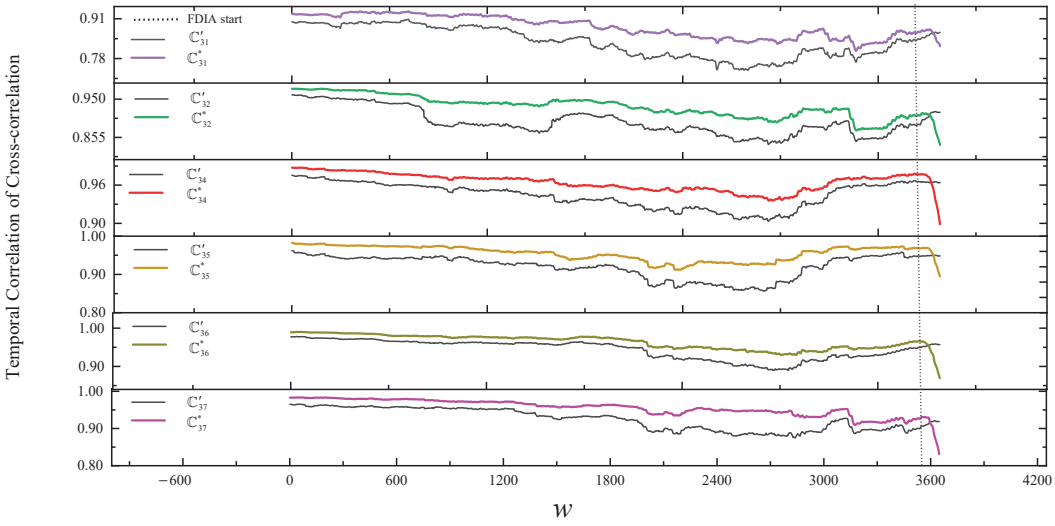



**Figure 11.** The results of comparing the normal temporal correlation of cross-correlation with the actual temporal correlation of cross-correlation of $v_3$ within the group in a stealthy FDIA.

In addition, assuming there is a collision, the attacker chooses the next node whose data are consistent with node $v_3$ as the next attack target to work in concert. With $v_2$ chosen as the next attack target, the same FDIA strategy is used to construct an attack signal for $v_2$ after the attacker learns the cross-correlation between $v_2$ and $v_3$. Figure 12 shows the signals of $v_2$ and $v_3$ with FDIA and without FDIA, where the FDIA starts at $t = 4247$ h.

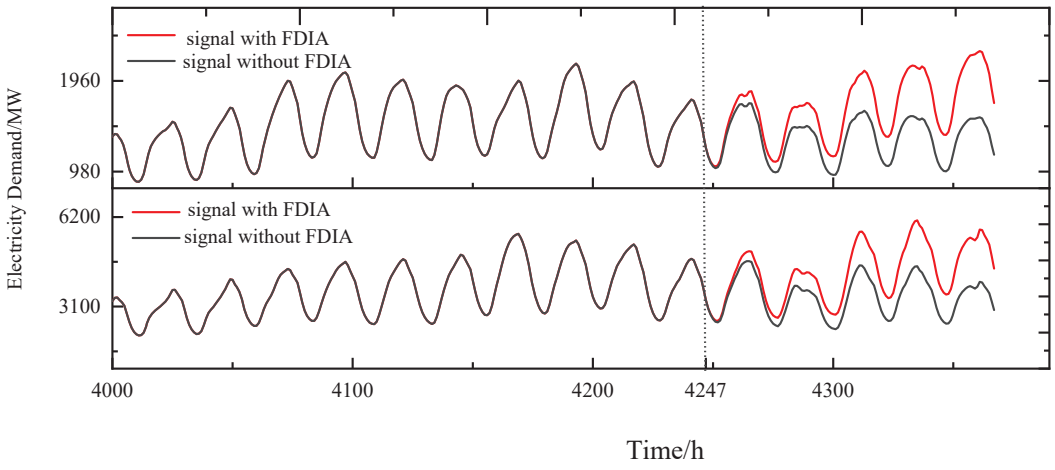



**Figure 12.** The stealthy and collusive FDIA on $v_2$ and $v_3$.

In our solution, Figure 13 and the fourth line of Table 3 show the results of comparing the normal temporal correlation of cross-correlation with the actual temporal correlation of cross-correlation of $v_2$ and $v_3$ within the group in stealthy and collusive FDIAs. As shown in Figure 13, from the FDIA's start, the change in the trend of $\mathbb{C}^*_{32}$ relative to the trend of $\mathbb{C}'_{32}$ is relatively consistent, and $\rho_{32} = 0.98$ also indicates that the readings of the collusive nodes are consistent. Due to the proposed voting algorithm, the framework can still detect the stealthy and collusive FDIAs on $v_2$ and $v_3$. However, in the SCCR solution, we similarly do not identify any outliers during the first 66 h of the FDIA, after which the abnormal nodes $v_2$, $v_3$, $v_6$, and $v_7$ are identified.

We conducted a total of 100 similar experiments in all groups, in which both the framework proposed in this paper and the SCCR solution were able to detect at least 95% of FDIAs (Figure 9a) with no more than a 5% false-negative detection rate (Figure 9b); the false-positive detection rate for the framework proposed in this paper was, again, no more than 3%, while the false-positive detection rate for the SCCR solution was as

high as 19% (Figure 9c). Furthermore, we observed that long-term attack signals result in stronger inconsistencies than short-term attack signals. Therefore, we conclude that, in general, the framework proposed in this paper performs well for detecting stealthy FDIAs in two collusive sensors, and our approach is superior compared to the SCCR solution in detecting long-term, stealthy, and collusive attack signals. However, as the number of collusive sensors increased, we observed a performance degradation. The proposed detection algorithm fails when the number of collusive sensors exceeds 50%. This is due to the fact that the detection algorithm uses majority voting, and more than 50% of the sensors must be normal to ensure the performance of the detection.

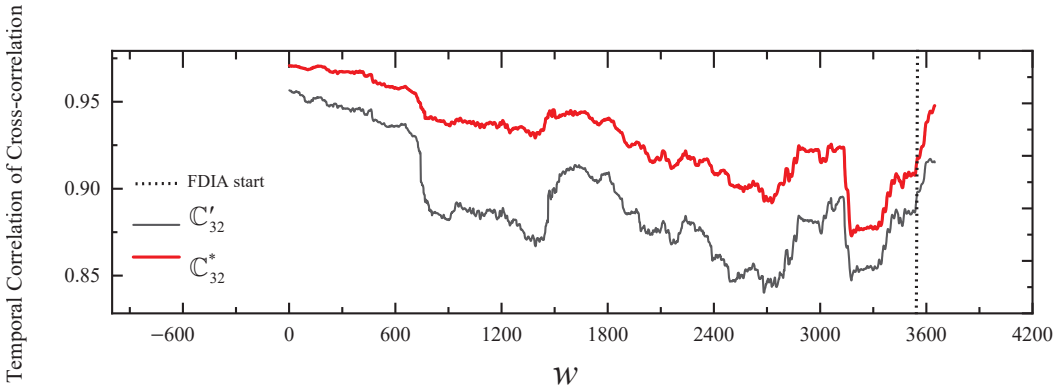



**Figure 13.** The results of comparing the normal temporal correlation of cross-correlation with the actual temporal correlation of cross-correlation of $v_2$ and $v_3$ in stealthy and collusive FDIAs.

Overall, the framework proposed in this paper performs best in detecting simple and stealthy FDIAs in single-sensor scenarios and is relatively effective in detecting stealthy FDIAs in multi-sensor scenarios.

## 6. Conclusions and Future Works

This paper presents a novel detection framework for FDIAs on large-scale WSNs. The framework consists of three phases. The first stage groups the sensors, which is based on the temporal correlation of the cross-correlation between the pairwise sensors. The second phase proposes a model for learning the temporal correlation of the cross-correlation. The third stage establishes consistency criteria within each group and votes out the abnormal nodes. We validated the performance of the framework by simulating simple FDIAs and stealthy FDIAs on a real dataset.

However, the detection framework also has some limitations. First, this paper only considers the scenario where FDIAs exist, and the framework is not designed to distinguish between FDIAs and natural anomalies, disruptive events, etc. Second, ARIMA is usually more suitable for forecasting problems with one-dimensional time series data, while for more complex problems, especially when multidimensional data are involved, the method needs to be further optimized. In addition, the voting algorithm fails to detect FDIAs on more than 50% of the sensors, and there is merit in exploring detection methods in the collusion-tolerant anomaly. Thus, there is value in further research on an anomaly score aggregation that tolerates collusion, and future work on the detection framework can be optimized by exploring other techniques to distinguish between FDIAs and natural anomalies. In addition, using a distributed detection framework that takes into account the trade-off between cost and criticality, the work can be conducted in the context of an optimization problem, such as the allocation of defense resources [32,33]. Finally, the framework proposed in this paper can be generalized to other correlation-based problems, such as advanced persistent threat detection [34,35], DDoS detection [36,37], and event-triggered state estimation [38,39].

**Author Contributions:** Conceptualization, J.H. and X.Y.; methodology, J.H., X.Y. and L.-X.Y.; software, J.H.; validation, J.H., X.Y. and L.-X.Y.; formal analysis, J.H., X.Y. and L.-X.Y.; investigation, J.H. and L.-X.Y.; resources, X.Y.; data curation, J.H. and X.Y.; writing—original draft preparation, J.H.; writing—review and editing, X.Y.; visualization, J.H.; supervision, X.Y.; project administration, X.Y.; funding acquisition, X.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by National Natural Science Foundation of China (Grant No. 61572006).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are available upon request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Forster, A. *Introduction to Wireless Sensor Networks*; Wiley-IEEE Press: Hoboken, NJ, USA, 2016.
2. El Emary, I.M.M.; Ramakrishnan, S. (Eds.) *Wireless Sensor Networks: From Theory to Applications*; CRC Press: Boca Raton, FL, USA, 2013.
3. Faquih, A.; Kadam, P.; Saquib, Z. Cryptographic techniques for wireless sensor networks: A survey. In Proceedings of the 2015 IEEE Bombay Section Symposium (IBSS), Mumbai, India, 10–11 September 2015; pp. 1–6. [CrossRef]
4. Oreku, G.S.; Pazynyuk, T. *Security in Wireless Sensor Networks*; Springer International Publishing: Cham, Switzerland, 2016.
5. Rani, A.; Kumar, S. A survey of security in wireless sensor networks. In Proceedings of the 3rd International Conference on CICT, Ghaziabad, India, 9–10 February 2017; pp. 1–5.
6. Ahmed, M.; Pathan, A.-S.K. False data injection attack (FDIA): An overview and new metrics for fair evaluation of its countermeasure. *Complex Adapt. Syst. Model.* **2020**, *8*, 4. [CrossRef]
7. Illiano, V.P.; Lupu, E.C. Detecting malicious data injections in wireless sensor networks: A survey. *ACM Comput. Surv. (CSUR)* **2015**, *48*, 1–33. [CrossRef]
8. Urbina, D.I.; Urbina, D.I.; Giraldo, J.; Cardenas, A.A.; Valente, J.; Faisal, M.; Tippenhauer, N.O.; Ruths, J.; Candell, R.; Sandberg, H. *Survey and New Directions for Physics-Based Attack Detection in Control Systems*; National Institute of Standards and Technology, US Department of Commerce: Gaithersburg, MD, USA, 2016.
9. Liu, Y.; Cheng, L. Relentless false data injection attacks against Kalman-filter-based detection in smart grid. *IEEE Trans. Control Netw. Syst.* **2022**, *9*, 1238–1250. [CrossRef]
10. Hegazy, H.I.; Tag Eldien, A.S.; Tantawy, M.M.; Fouda, M.M.; TagElDien, H.A. Real-time locational detection of stealthy false data injection attack in smart grid: Using multivariate-based multi-label classification approach. *Energies* **2022**, *15*, 5312. [CrossRef]
11. Gu, Y.; Yu, X.; Guo, K.; Qiao, J.; Guo, L. Detection, estimation, and compensation of false data injection attack for UAVs. *Inf. Sci.* **2021**, *546*, 723–741. [CrossRef]
12. Moazeni, F.; Khazaei, J. Formulating false data injection cyberattacks on pumps' flow rate resulting in cascading failures in smart water systems. *Sustain. Cities Soc.* **2021**, *75*, 103370. [CrossRef]
13. Ren, X.X.; Yang, G.H. Adaptive control for nonlinear cyber-physical systems under false data injection attacks through sensor networks. *Int. J. Robust Nonlinear Control* **2020**, *30*, 65–79. [CrossRef]
14. Padhan, S.; Turuk, A.K. Design of false data injection attacks in cyber-physical systems. *Inf. Sci.* **2022**, *608*, 825–843. [CrossRef]
15. Miao, B.; Wang, H.; Liu, Y.-J.; Liu, L. Adaptive security control against false data injection attacks in cyber-physical systems. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2023**. [CrossRef]
16. Illiano, V.P.; Steiner, R.V.; Lupu, E.C. Unity is strength! Combining attestation and measurements inspection to handle malicious data injections in WSNs. In Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Boston, MA, USA, 18–20 July 2017; pp. 134–144.
17. Aboelwafa, M.M.; Seddik, K.G.; Eldefrawy, M.H.; Gadallah, Y.; Gidlund, M. A machine-learning-based technique for false data injection attacks detection in industrial IoT. *IEEE Internet Things J.* **2020**, *7*, 8462–8471. [CrossRef]
18. Martovytskyi, V.; Ruban, I.; Lahutin, H.; Ilina, I.; Rykun, V.; Diachenko, V. Method of detecting FDI attacks on smart grid. In Proceedings of the 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 6–9 October 2020; pp. 132–136.
19. Berjab, N.; Le, H.H.; Yokota, H. A spatiotemporal and multivariate attribute correlation extraction scheme for detecting abnormal nodes in WSNs. *IEEE Access* **2021**, *9*, 135266–135284. [CrossRef]
20. Huang, D.-W.; Liu, W.; Bi, J. Data tampering attacks diagnosis in dynamic wireless sensor networks. *Comput. Commun.* **2021**, *172*, 84–92. [CrossRef]
21. Hu, J.; Yang, X.; Yang, L. A novel diagnosis scheme against collusive false data injection attack. *Sensors* **2023**, *23*, 5943. [CrossRef]
22. Chen, P.-Y.; Yang, S.; McCann, J.A. Distributed real-time anomaly detection in networked industrial sensing systems. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3832–3842. [CrossRef]

23. Islam, J.; Talusan, J.P.; Bhattacharjee, S.; Tiausas, F.; Vazirizade, S.M.; Dubey, A.; Yasumoto, K.; Das, S.K. Anomaly based incident detection in large scale smart transportation systems. In Proceedings of the 2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS), Milano, Italy, 4–6 May 2022; pp. 215–224.
24. Lai, Y.; Tong, L.; Liu, J.; Wang, Y.; Tang, T.; Zhao, Z.; Qin, H. Identifying malicious nodes in wireless sensor networks based on correlation detection. *Comput. Secur.* **2022**, *113*, 102540. [CrossRef]
25. Hamilton, J.D. *Time Series Analysis*; Princeton University Press: Princeton, NJ, USA, 2020.
26. Rassam, M.A.; Zainal, A.; Maarof, M.A. Advancements of data anomaly detection research in wireless sensor networks: A survey and open issues. *Sensors* **2013**, *13*, 10087–10122. [CrossRef]
27. Shiavi, R. *Introduction to Applied Statistical Signal Analysis: Guide to Biomedical and Electrical Engineering Applications*; Elsevier: Amsterdam, The Netherlands, 2010.
28. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
29. Choi, B. *ARMA Model Identification*; Springer Science & Business Media: New York, NY, USA, 2012.
30. Mushtaq, R. *Augmented Dickey Fuller Test*; SSRN-Elsevier: Rochester, NY, USA, 2011.
31. Chan-Tin, E.; Feldman, D.; Hopper, N.; Kim, Y. The frog-boiling attack: Limitations of anomaly detection for secure network coordinate systems. In Proceedings of the Security and Privacy in Communication Networks: 5th International ICST Conference (SecureComm 2009), Athens, Greece, 14–18 September 2009; Revised Selected Papers 5, 2009; pp. 448–458.
32. Hao, W.; Yao, P.; Yang, T.; Yang, Q. Industrial cyber–physical system defense resource allocation using distributed anomaly detection. *IEEE Internet Things J.* **2021**, *9*, 22304–22314. [CrossRef]
33. Sun, H.; Yang, X.; Yang, L.-X.; Huang, K.; Li, G. Impulsive artificial defense against advanced persistent threat. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3506–3516. [CrossRef]
34. Wang, X.; Liu, Q.; Pan, Z.; Pang, G. APT attack detection algorithm based on spatio-temporal association analysis in industrial network. *J. Ambient. Intell. Humaniz. Comput.* **2020**, 1–10. [CrossRef]
35. Yang, L.-X.; Huang, K.; Yang, X.; Zhang, Y.; Xiang, Y.; Tang, Y.Y. Defense against advanced persistent threat through data backup and recovery. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 2001–2013. [CrossRef]
36. Cao, Y.; Jiang, H.; Deng, Y.; Wu, J.; Zhou, P.; Luo, W. Detecting and mitigating ddos attacks in SDN using spatial-temporal graph convolutional network. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 3855–3872. [CrossRef]
37. Khan, M.A.; Nasralla, M.M.; Umar, M.M.; Khan, S.; Choudhury, N. An efficient multilevel probabilistic model for abnormal traffic detection in wireless sensor networks. *Sensors* **2022**, *22*, 410. [CrossRef]
38. Akrami, A.; Mohsenian-Rad, H. Event-Triggered Distribution System State Estimation: Sparse Kalman Filtering with Reinforced Coupling. *IEEE Trans. Smart Grid* **2023**, *15*, 627–640. [CrossRef]
39. Ponnarasi, L.; Pankajavalli, P.; Lim, Y.; Sakthivel, R. Optimization Based Event-Triggered State Estimation Algorithm for IoT-Based Wind Turbine Systems. *IEEE Internet Things J.* **2023**, *early access*. [CrossRef]


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.