IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 23, 2020, accepted January 4, 2021, date of publication January 8, 2021, date of current version January 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049837

Lightweight Fault Detection Strategy for Wireless Sensor Networks Based on Trend Correlation

XIUWEN FU[1,2], (Member, IEEE), YE WANG[1], WENFENG LI[2], (Senior Member, IEEE), YONGSHENG YANG[1], AND OCTAVIAN POSTOLACHE[3], (Senior Member, IEEE)

[1]Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China
[2]School of Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China
[3]ISCTE-Lisbon University Institute, 1070 Lisbon, Portugal

Corresponding author: Xiuwen Fu (xwfu@shmtu.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61902238.

ABSTRACT In this work, we propose a fault detection strategy for wireless sensor networks (WSNs) called Trend Correlation based Fault Detection strategy (TCFD). This strategy can detect the faulty sensor nodes through analyzing the trend correlation and the median value of neighboring nodes. On this basis, aiming to avoid the excessive routing overhead caused by over-frequent fault detection, a fault detection self-starting mechanism is designed based on the cubic exponential smoothing method. Since the detection results in TCFD are determined by historical data at consecutive times and do not rely on the comparison of instantaneous sensed values at a single moment, it can significantly reduce the impact of fault detection time on detection accuracy. The simulation results have indicated that compared with referenced strategies, the proposed TCFD can obtain better fault detection accuracy for four common fault types of sensor nodes; in the case where the real fault rate of the network reaches 0.5, at least 70% of the faulty nodes can be detected by TCFD and the false alarm rate can still be kept below 30%; with the help of fault detection self-starting mechanism, the response time of sensor nodes to faults can be significantly shortened.

INDEX TERMS Wireless sensor networks, fault detection, trend correlation, self-starting mechanism, fault type, exponential smoothing method.

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of sensor nodes. Each of these individual sensor nodes in a WSN has sensing and processing capability. Sensor nodes are low power devices with limited computational power, memory, battery and storage [1]–[4]. Since sensor nodes are usually deployed in hostile and harsh environments, they are susceptible to frequent and unexpected faults. The occurrence of faults during normal operation may results in severer consequences [5]–[7]. For example, in fire-alarming WSNs, if a sensor node gives incorrect readings, it might result in false alarming or loss of lives due to the absence of warning. Therefore, fault detection strategies have always been a hot spot in the research field of WSNs, and some gratifying progress has been made. According to the differences in detection executors, existing fault detection strategies for WSNs can be divided into two categories: centralized strategies and distributed strategies [8]–[10]. In centralized fault detection strategies, the sink node is responsible for fault detection. In most actual WSNs, as the fusion center, the sink node has powerful computational resources and storage resources, so it supports the fault detection strategies with high algorithmic complexity. However, the centralized strategies need to rely on the sensor nodes in the network to send all the raw data to it to complete the fault detection, which will generate excessive routing overhead. In distributed fault detection strategies, the sensor nodes can complete the fault detection by means of neighboring cooperation, and do not need to forward data to the sink node, so it can quickly respond to the fault and significantly reduce the routing overhead. However, compared with the sink node, the computing resources and storage resources owned by sensor nodes are quite limited, which makes them unable to support detection strategies with high algorithmic complexity. Therefore, distributed fault detection strategies are usually lightweight from the perspective of algorithmic complexity.

In the actual WSNs, we generally assumed that readings of sensor nodes belonging to nearby regions at a given point in time or readings by a sensor node within a time window

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Shu.

are correlated and these correlations in measured data are called spatial and temporal correlations respectively. Based on this assumption, most distributed fault detection strategies in WSNs use whether the sensing data of neighboring sensor nodes are similar at the same time as the detection basis. If the difference between the sensing data of neighboring nodes is smaller than the similarity threshold, the two nodes are considered ''similar''. In the case where a node is similar to the majority of its neighboring nodes, this node can be regarded as a normal node. This type of fault detection strategy uses the instantaneous value difference between neighboring nodes as the basis for fault detection, making the detection results very sensitive to the detection time. That is to say, for the same faulty node, at different times, the fault detection strategy may obtain different detection results.

Due to this reason, in this work, we propose a Trend Correlation based Fault Detection strategy (TCFD). Since the trend between neighboring nodes is jointly determined by historical data at consecutive times and does not rely on the comparison of instantaneous values at a single moment, the proposed TCFD can effectively reduce the impact of fault detection time on detection accuracy. The major contributions of this work are summarized as follows:

- To reduce the impact of fault detection time on detection accuracy, a distributed fault detection strategy TCFD based on the trend correlation and the median value of neighboring nodes is proposed;
- In order to improve the response speed of network to node faults, a fault detection self-starting mechanism is developed based on the cubic exponential smoothing method;
- The detection accuracy of the proposed detection strategy for four common fault types is verified. The simulation results have indicated that the proposed TCFD can obtain better fault detection accuracy than the referenced strategies.

The reminder of the paper is organized as follows: Section 2 reviews the related work; Section 3 elaborates the detection strategy TCFD in detail; Section 4 presents the self-starting mechanism in TCFD; Section 5 gives the experiment results. Finally, the paper is concluded.

II. RELATED WORK

In recent years, with the rapid development of WSN technologies, gratifying research progress has been made in the field of fault detection. Existing fault detection strategies for WSNs can roughly divided into two categories: centralized and distributed. The centralized strategies use the sink node to monitor the status of the entire network [11]–[18]. The centralized strategies increase the routing overhead in the network, which leads to the emergence of the distributed solutions. In the distributed strategies, monitoring tasks are distributed among each sensor nodes in the network. Since in this work, our focus is on distributed fault detection strategies, we will only review the research progress in this field.

Distributed detection strategies use the spatial-temporal correlation between sensor nodes to detect faults in sensor reading. To achieve this, most of them use a ''majority is right'' mechanism. In this mechanism, the node to be detected compares its state value with the majority of state values of its neighboring nodes. If the comparison result is within the allowable error range, the node is deemed to be in a normal status. In [19], Yarinezhad *et al.* proposed a faulty node detection method for WSNs based on cellular learning automata. In this method, the cellular learning automata at each node determines the status of the node based on its hardware condition. In [20], Chen *et al.* proposed a localized fault detection algorithm for WSNs based on median value. In this algorithm, if the sensed data of a sensor node deviates significantly from the median value of the sensed data from neighboring nodes, they will consider this node to be a faulty node. In [21], Jiang *et al.* proposed a Distributed Fault Detection strategy (DFD). This strategy includes two rounds. The first round is to find the possible faulty nodes by self-diagnose, and the second round is to determine the final fault status by analyzing the fault information from the neighboring nodes. In [22], Saihi *et al.* proposed a Distributed Fault Detection strategy based on Error Functions (DFDEF). In this strategy, two error functions were developed using Gaussian distribution and reduced centered normal distribution, respectively. In [23], Feng *et al.* proposed a distributed fault detection algorithm based on weighted distance. In this algorithm, the weighted sensed value of a node is compared to original sensed value to judge faulty nodes. In [24], Panda *et al.* proposed a Distributed Fault Detection Strategy (DFDS). In this algorithm, the mean of neighboring nodes is computed to check whether faulty sensor node is present or not. In [25], Yuan *et al.* proposed a distributed Bayesian algorithm to detect node faults. The fault probability of sensor nodes is calculated using Bayesian networks and is improved by exploiting the border nodes to increase the fault detection accuracy. In [26], Mo *et al.* proposed a fault detection strategy based on time domain features of sensed data. This strategy used one-dimensional Gabor transform to analyze features of the sensed data and determine the fault status by comparing the feature values of neighboring nodes. In [27], Gharamaleki *et al.* proposed a probabilistic fault detection strategy for WSNs. In this strategy, the probabilistic and deterministic status of sensor nodes is determined by analyzing the differences between sensed data of neighboring nodes. For ease of understanding, Table 1 compares the the above-mentioned distributed fault detection strategies from four aspects (*i.e.,* approach, topology dependent, trigger mechanism and double check).

According to the literature review, the existing distributed fault detection strategies determine the fault state of sensor nodes by comparing the spatial eigenvalues of adjacent nodes, but these eigenvalues are extracted from the instantaneous sensor readings obtained in a single time period, which makes the detection accuracy very sensitive to the detection time. In addition, the existing strategies rarely involve fault

TABLE 1. Comparison of distributed fault detection strategies in WSNs.

Strategies	Approach	Topology dependent	Threshold based	Trigger mechanism	Double check
Yarinezhad *et al.* [19]	Cellular learning automata	Yes	No	Not mentioned	No
Chen *et al.* [20]	Median value	Yes	Yes	Not mentioned	No
Jiang *et al.* [21]	Neighboring voting	Yes	Yes	Not mentioned	Yes
Saihi *et al.* [22]	Error function	No	No	Timer	Yes
Feng *et al.* [23]	Weighted voting	Yes	Yes	Not mentioned	No
Panda *et al.* [24]	Mean value	Yes	Yes	Not mentioned	Yes
Yuan *et al.* [25]	Bayesian judgement	No	No	Timer	No
Mo *et al.* [26]	Gabor transform	No	Yes	Not mentioned	No
Gharamaleki *et al.* [27]	Probabilistic comparison	No	Yes	Timer	No

detection trigger mechanism, which makes sensor nodes unable to respond to fault events in the first time. Therefore, in the design of fault detection strategy, it is necessary to make the following two improvements: 1) the spatial-temporal correlation features extracted from continuous times should be used in fault detection, so as to reduce the impact of detection time on detection accuracy; 2) the self-starting detection mechanism should be applied to improve the response speed of sensor nodes to fault events.

III. FAULT DETECTION STRATEGY

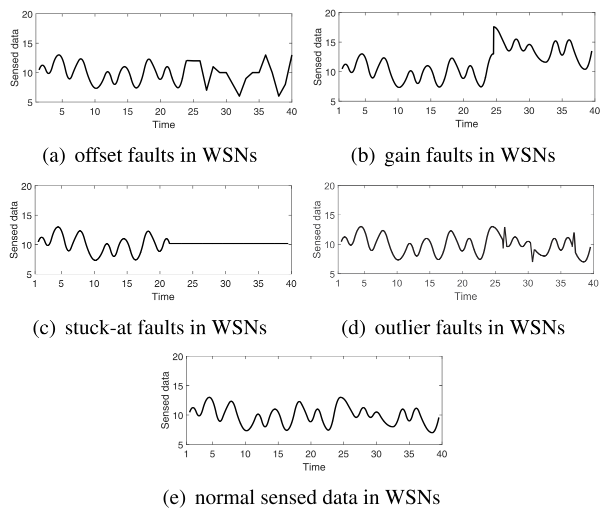
In this section, we first introduce the common fault types of WSNs from a data-centric perspective, then define the trend correlation and the median value of sensed data, finally present the fault detection strategy TCFD.

A. FAULT TYPES

From a data-centric perspective, faults in WSNs can be divided into four categories: offset fault, gain fault, stuck-at fault and outlier fault.

- Outlier fault refers to the fault in which a few discrete data points in a series of sensed data deviate significantly from the expected data. Sensors subject to short-term vibration and electromagnetic interference are the main causes of outlier faults. An outlier fault can be modeled as $x' = \omega + x$.
- Gain fault refers to the fault if the rate of change of sensed data fails to match with expectation over an extended period of time. In gain fault, a constant value gets multiplied to the non-fault sensed data. The gain fault might occur due to the excessive signal noise inside the sensor module. A gain fault can be modeled as $x' = \beta x + \eta$, where β is the gain coefficient that gets multiplied to the normal reading x;
- Stuck-at fault refers to the fault that the sensed data is constant and cannot respond to environmental changes. The stuck-at fault might occur when the power supply of the sensor module is interrupted. A stuck-at fault can be modeled as $x' = \omega$;
- Offset fault refers to the addition of deterministic bias from the original measurement. This might occur due to improper calibration of the sensor or the reading drifts away from original calibration formulas. Therefore, in many literatures [11], [28], offset fault is also called drift fault. An offset fault can be modeled as $x' = \omega + x + \eta$, where x' is the faulty reading, x is the normal reading, ω is a constant value, η is the permitted measurement error. It should be noted that in this fault type, ω is much greater than η;

To facilitate the understanding of the above-mentioned four types of faults in WSNs, Fig.1 shows the comparison between normal sensed data and the four types of faulty sensed data.



(a) offset faults in WSNs
(b) gain faults in WSNs
(c) stuck-at faults in WSNs
(d) outlier faults in WSNs
(e) normal sensed data in WSNs

FIGURE 1. Comparison between normal sensed data and faulty sensed data.

B. NETWORK ASSUMPTIONS

In this work, we generally assume that a WSN is deployed on a target area to monitor specific physical phenomena and all sensed data is forwarded from the general sensor nodes to the sink node. All sensor nodes are fixed and can keep their clock synchronized. The batteries of sensor nodes cannot be charged. The wireless links between sensor nodes are symmetric and fault-free. We recognize that local processing can be done to reduce overall communication costs. Moreover, we assume that all sensor nodes are homogeneous in terms of energy, communication, and processing capabilities. To facilitate the understanding of TCFD, Table 2 summarizes all the parameters in the strategy.

C. TREND CORRELATION OF SENSED DATA

In this part, we propose a metric to measure the trend correlation of sensed data. We first give three definitions related to the trend correlation of sensed data.

TABLE 2. Parameters in TCFD.

Items	Descriptions
$x_i(t)$	sensed data obtained by sensor node i at time t
m	length of time window for fault detection
$X_i(t)$	data series obtained by sensor node i within the time window $[t-m, t]$
$var(i)$	variance of data series $X_i(t)$
$\overline{X_i(t)}$	mean of data series $X_i(t)$
$cov(i,j)$	covariance between $X_i(t)$ and $X_j(t)$
$\chi_{i,j}(t)$	trend correlation coefficient between $X_i(t)$ and $X_j(t)$
$\Omega(i)$	set formed by neighboring nodes of node i
$M_i(t)$	sensed data of neighboring nodes of node i at time t
$N_i(t)$	sorted set of $M_i(t)$ in descending order of sensed values
$y_k(t)$	kth element in set $N_i(t)$
n_i	number of neighboring nodes of node i
$med_i(t)$	median value of node i at time t
x_{max},x_{min}	maximum normal reading and minimum normal reading of sensor nodes, respectively
θ	trend correlation threshold
P_i	number of neighboring nodes that have correlated trend with node i
$S_i(t)$	preliminary detection status of node i at time t
α	smoothing coefficient
δ	median threshold
γ	detection trigger threshold
a, b, c	weighted coefficients for cubic exponential s-moothing prediction
$x_i^*(t+t_1)$	predicted sensed value of node i at time t+t_1
$S_t^{(1)}$, $S_t^{(2)}$, $S_t^{(3)}$	single, quadratic and cubic exponential smoothing values

Definition 1: Suppose $x_i(t)$ is the sensed data obtained by sensor node i at time t, then we can define the data series $X_i(t) = [x_i(t-m), x_i(t-m+1), \ldots, x_i(t)]$ within the sliding time window $[t-m, t]$.

Definition 2: Suppose sensor node i obtains the data series $X_i(t)$ within the time window $[t-m, t]$, then we can define the variance of $X_i(t)$ as

$$var(i) = \frac{\sum_{k=1}^{m+1} \left[x_i(t-k+1) - \overline{X_i(t)}\right]^2}{m}, \tag{1}$$

$$\overline{X_i(t)} = \sum_{k=1}^{m+1} \frac{x_i(t-k+1)}{m+1}, \tag{2}$$

where $\overline{X_i(t)}$ refers to the mean of $X_i(t)$.

Definition 3: Suppose $X_i(t)$ and $X_j(t)$ are the data series obtained by node i and node j within the time window $[t-m, t]$, then we can define the covariance between $X_i(t)$ and $X_j(t)$ as

$$cov(i,j) = \frac{\sum_{k=1}^{m+1} [x_i(t-k+1) - \overline{X_i(t)}][x_j(t-k+1) - \overline{X_j(t)}]}{m}. \tag{3}$$

Based on Pearson's correlation coefficient [29], we define the trend correlation coefficient $\chi_{i,j}(t)$ between $X_i(t)$ and $X_j(t)$ at time t as

$$\chi_{i,j}(t) = \frac{cov(i,j)}{\sqrt{var(i)var(j)}} = \frac{\sum_{k=1}^{m+1} [x_i(t-k+1) - \overline{X_i(t)}][x_j(t-k+1) - \overline{X_j(t)}]}{\sqrt{\sum_{k=1}^{m+1} \left[x_i(t-k+1) - \overline{X_i(t)}\right]^2 \sum_{k=1}^{m+1} \left[x_j(t-k+1) - \overline{X_j(t)}\right]^2}}. \tag{4}$$

The trend correlation coefficient $\chi_{i,j}(t)$ is between $[-1, 1]$. The larger the absolute value of $\chi_{i,j}(t)$, the higher the correlation between $X_i(t)$ and $X_j(t)$. For example, if the sensed data series $X_i(t)$ and $X_j(t)$ obtained by neighboring nodes i and j at time t are [1,1.1,1.2,1.1] and [1.2,1.3,1.4,1.3], we can conclude that the trend of these two nodes are perfectly correlated, and the difference in their readings is due to their different positions. In this case, we can get $\chi_{i,j}(t) = 1$ according to (4), which perfectly characterizes the trend correlation between $X_i(t)$ and $X_j(t)$.

D. MEDIAN VALUE OF SENSED DATA

According to Section III-A, for stuck-at faults and offset faults, because the data trend in the continuous time segment has been destroyed, these two types of faults can be easily detected with the help of trend correlation. However, for gain faults and outlier faults, because the data sensed by nodes can still respond to changes in the surrounding environment, if only the trend correlation is used as the basis for fault detection, there is a high probability that such faults will not be detected. In order to avoid this problem, the median value of the sensed data from neighboring nodes is introduced into the fault detection process to ensure that the proposed TCFD has good detection accuracy for all these four fault types.

Definition 4: Suppose $\Omega(i) = \{j, \ldots, h\}$ is the set formed by neighboring nodes of node i, we can get the sensed data of neighboring nodes of node i at time t as $M_i(t) = \{x_j(t), \ldots, x_h(t)\}$. We sort the set $M_i(t)$ in descending order of sensed values, and get the sorted set $N_i(t)$. We use $y_k(t)$ to represent the kth element in set $N_i(t)$.

The median value of node i at time t can be defined as

$$med_i(t) = \begin{cases} y_{(n_i+1)/2} & n_i \text{ is odd} \\ \dfrac{y_{n_i/2} + y_{(n_i+2)/2}}{2} & n_i \text{ is even}, \end{cases} \tag{5}$$

where n_i is the number of neighboring nodes of node i. Unlike many fault detection strategies that select the average sensed value of neighbor nodes [24], [28], the proposed TCFD uses the median value to support fault detection. Although both the median and mean can be used to represent the center of the neighboring node's sensed data, their tolerance for faulty nodes has a significant difference. For the mean of the neighborhood, if there are a few faulty nodes in the neighborhood, the error data of these faulty nodes will lead to a certain degree of deviation between the mean value of the neighborhood and the expected normal value. For the median of the neighborhood, as long as the number of faulty nodes does not exceed half of the total number of neighboring nodes, the median value of the neighborhood will not deviate from the expected normal value.

E. FAULT DETECTION PROCESS OF TCFD

The basic idea of TCFD is: if the trend of the node to be tested is consistent with most of its neighboring nodes, and its sensed data is close to the median center of the neighborhood, then the node is considered as a normal node, otherwise it is determined as a faulty node. The fault detection steps of TCFD are as follows:

Step 1: For node i, if the fault detection mechanism inside it is triggered, it will send a request message to its neighboring nodes. Then, each neighboring node $j \in \Omega(i)$ will send back the data encapsulated with $X_j(t)$ after receiving the request message;

Step 2: Node i sequentially calculates the trend correlation coefficient $\chi_{i,j}(t)$ between itself and its neighboring nodes according to (4), and turns on the counter P_i. The initial value of the counter P_i is 0. If $\chi_{i,j}(t) > \theta$, the counter P_i increases by 1, otherwise the P_i remains unchanged. θ is the trend correlation threshold. In this work, When $\chi_{i,j}(t)$ is greater than θ, we assume that node i and node j have similar trends. Therefore, after completing the trend correlation calculation of all neighboring nodes, the value of P_i is the number of neighboring nodes that have correlated trend with node i;

Step 3: Node i calculates the neighboring median value $med_i(t)$ according to the received data series from all neighboring nodes;

Step 4: If node i satisfies the conditions: $P_i \geq n_i/2$ and $\left|\frac{x_i(t)-med_i(t)}{x_{\max}-x_{\min}}\right| \leq \delta$, then the preliminary detection status of node i is possibly normal (PN), otherwise the status of node i is possibly faulty (PF). δ is the median threshold. x_{max} and x_{min} are the maximum normal reading and the minimum normal reading, respectively;

Step 5: All nodes in the neighboring set $\Omega(i)$ of node i perform steps 1 to 4. Then each node j in $\Omega(i)$ can obtain the preliminary detection status $S_j(t)$, and return this status to node i;

Step 6: Node i counts the number of possible normal nodes G_{PN} and the number of possible faulty nodes G_{PF} in the neighboring nodes. If $G_{PN} \geq G_{PF}$, the possible status of node i is confirmed. For the node i whose possible status is PN, its status will become normal (NR). For the node whose possible state is PF, its status will become faulty (FT). If $G_{PN} < G_{PF}$, we will invert the possible status of node i. For the node i whose possible status is PN, its status will become FT. For the node whose possible status is PF, its status will become NR;

Step 7: If the final fault status of node i is confirmed as FT, it will send an alarm message to the sink node. The sink node will report this message to the user.

Through the above description, it is not difficult to find that TCFD includes two rounds. The first round is to obtain the possible fault status of the node by analyzing the trend correlation and the median value, and the second round is to conduct a second confirmation of the possible fault status by comparing the possible fault status of neighboring nodes. In the actual WSNs, there may be a situation that when the number of neighboring nodes of a normal node i is small and the faulty nodes are the majority in its neighboring nodes, it is possible to misjudge node i as a faulty node only through the first round of detection. Therefore, we need to double confirm the possible fault status of node i through checking possible fault status of its neighboring nodes. In this way, we can effectively reduce the false alarm rate.

To facilitate the understanding, the flow chart of the detection process of TCFD is shown in Fig.2.

IV. SELF-STARTING MECHANISM OF TCFD

Existing fault detection strategies for WSNs mostly set up timers inside the sensor nodes, and use timed wake-up mechanism to trigger the fault detection process. However, in the actual fault detection process, this mechanism has some obvious limitations. If the detection time interval T_D set by the timer is too short, the fault detection mechanism in the network will be activated frequently, which intensifies the network communication load and generates a lot of extra energy consumption; on the contrary, if the detection time interval T_D is set too long, it will cause the network's response speed to node faults to decrease. Therefore, in TCFD, aiming to avoid the above limitations, we design a fault detection self-starting mechanism based on the cubic exponential smoothing prediction method.

As a lightweight data forecasting method, exponential smoothing has the advantages of high computational efficiency and low algorithmic complexity compared with time series autoregressive methods and artificial neural networks [30], [31]. In most actual WSNs, sensor nodes are only equipped with limited computing resources, it is quite challenging for their own processors to perform a large number of complex calculations in a short period of time. The exponential smoothing method can achieve rapid and accurate data predication with only a small amount of computing resources. The basic idea of the exponential smoothing method is to use the time series correlation characteristics of the data to obtain the predicted value by weighting the historical data. The exponential smoothing method can be further subdivided into: single exponential weighting, quadratic exponential weighting and cubic exponential weighting. The high-order exponential weighting is obtained by repeating the smoothing weighting operation on the basis of the low-order exponential weighting. Both the primary and secondary exponential smoothing methods are only suitable for the prediction of linear data series, while the cubic exponential smoothing method can be used for the prediction of nonlinear data series. For sensor nodes, although the fluctuation of sensed data is usually not dramatic in a fixed time window, the time-series

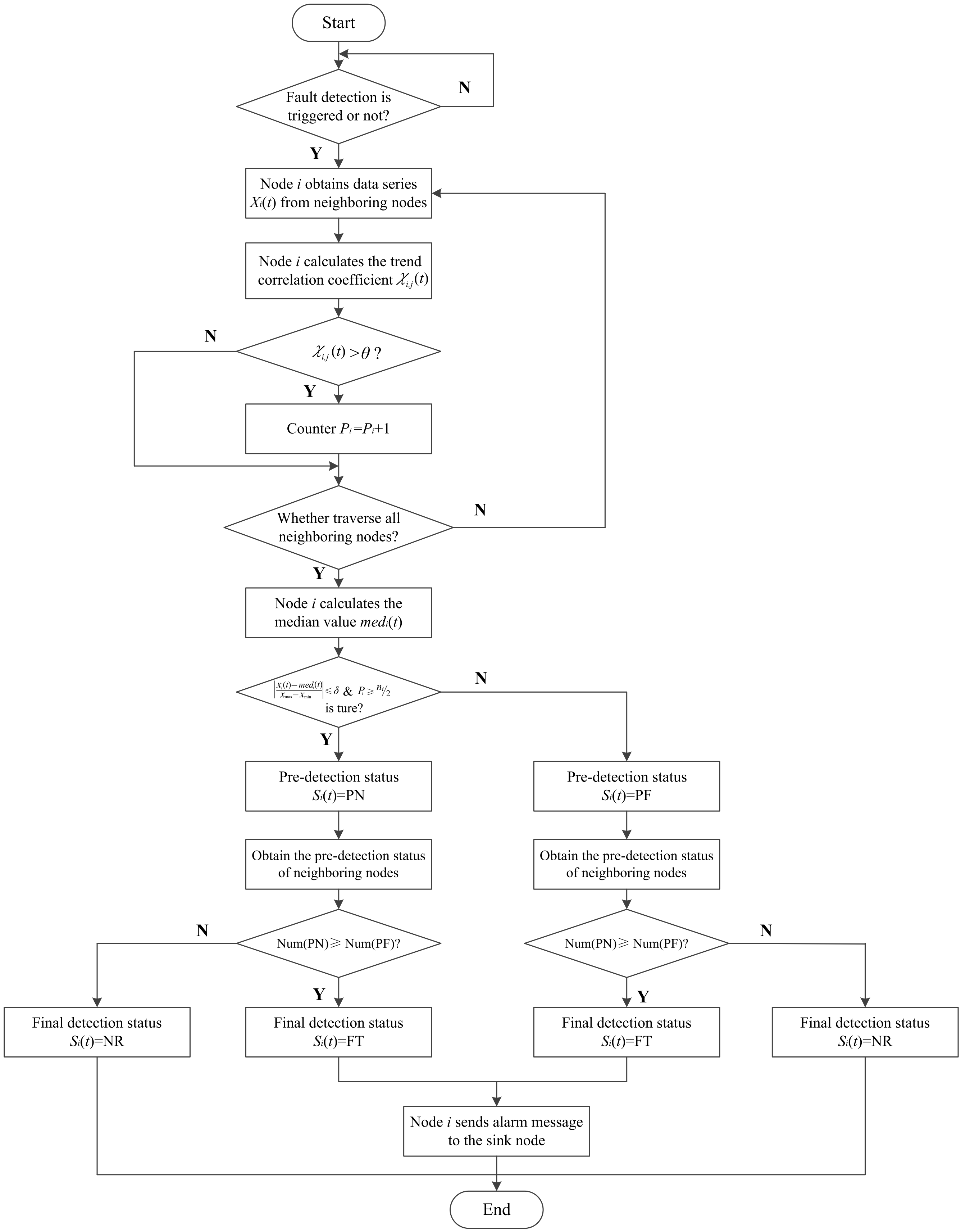


FIGURE 2. Detection process of TCFD.

data they collect usually demonstrates some nonlinear characteristics due to the interference of fault and noise [32]–[34]. Therefore, we use the cubic exponential smoothing method to predict the sensed data of sensor nodes at the next moment.

Suppose $X_i(t)$ is the data series obtained by node i within the time window $[t-m, t]$, then the the cubic exponential smoothing prediction model is defined as

$$x_i^*(t+t_1) = a + bT + cT^2, \tag{6}$$

where $x_i^*(t+t_1)$ is the predicted sensed value of node i at time $t+t_1$; a, b and c are the weighted coefficients, which can be calculated by

$$a = 3S_t^{(1)} - 3S_t^{(2)} + S_t^{(3)}, \tag{7}$$

$$b = \frac{\alpha}{2(1-\alpha)^2}\left[(6-5\alpha)S_t^{(1)} - 2(5-4\alpha)S_t^{(2)} + (4-3\alpha)S_t^{(3)}\right], \tag{8}$$

$$c = \frac{\alpha^2}{2(1-\alpha)^2}\left(S_t^{(1)} - 2S_t^{(2)} + S_t^{(3)}\right). \tag{9}$$

α is the smoothing coefficient. $S_t^{(1)}$, $S_t^{(2)}$ and $S_t^{(3)}$ are the single, quadratic and cubic exponential smoothing values, which can be calculated by

$$S_t^{(1)} = \alpha x_i(t) + (1-\alpha)S_{t-1}^{(1)}, \tag{10}$$

$$S_t^{(2)} = \alpha S_t^{(1)} + (1-\alpha)S_{t-1}^{(2)}, \tag{11}$$

$$S_t^{(3)} = \alpha S_t^{(2)} + (1-\alpha)S_{t-1}^{(3)}. \tag{12}$$

The key to the prediction accuracy of the cubic exponential smoothing method is the selection of smoothing coefficient α. The larger the smoothing coefficient α, the more obvious the forecast data $x_i^*(t+t_1)$ is affected by recent data change. In practical applications, the value of α can be determined according to the fluctuation feature of the data series. If the data series fluctuates relatively smoothly, the value of α should be kept in a small range; if the data series fluctuates sharply, the value of α should be a relatively large value. Considering the correlation characteristics of the data series obtained by sensor nodes in most cases, we set α to 0.45. We use the average value of the first three data in the data series $X_i(t)$ as the initial smoothing value, which is

$$S_0^{(1)} = S_0^{(2)} = S_0^{(3)} = \frac{\sum_{k=1}^{3}[x_i(t-m+k-1)]}{3}. \tag{13}$$

According to the above steps, node i can obtain the predicted value $x_i^*(t+1)$ and the actual sensed value $x_i(t+1)$ at time $t+1$. Here we use the detection trigger threshold γ to determine whether the predicted value meets the expectation. If $\left|\frac{x_i^*(t+1)-x_i(t+1)}{x_{\max}-x_{\min}}\right| \leq \gamma$, we can assume that the actual sampled value $x_i(t+1)$ meets expectations, and there is no need to trigger the fault detection process, otherwise the fault detection process will be triggered at time $t+2$. With the help of the fault detection self-starting mechanism, sensor nodes can complete the fault pre-detection in each sampling period, so as to ensure that they can respond to faulty events in the first time.

V. PERFORMANCE EVALUATION

A. SIMULATION SETUP

The experiments are based on NS-2. The original dataset used in this study is based on an existing dataset published by researchers at the University of North Carolina in 2010 [35]. Using TelosB motes, the researchers collected data from simple multi-hop WSNs. This dataset include humidity and temperature measurements measured every five seconds for six hours. The researchers introduced an environmental change event into the network to increase the variability of the data. They used hot water steam to increase humidity and temperature. In this study, we only used the temperature data in this original dataset. With different real fault rates (10%, 20%, 30%, 40% and 50%) and different fault types (*i.e.,* outlier, gain, stuck-at and offset), we prepared our experimental dataset according to the method presented in [11]. In the original method, the experimental dataset composed of a set of measurement vectors and each vector contains measurement values at 3 consecutive moments (t_0, t_1, t_2). In our experiments, we set the measurement vector as the measurement values at 10 consecutive moments, which can help the proposed TCFD strategy to better perceive the trend change of sensing data. In addition, in the experimental setting of [11], different values of β (2, 4, 6, 8, 10) are used to test the accuracy of the fault detection strategy in the face of gain faults. β is the gain coefficient. The higher the β value, the more obvious the deviation between the faulty data and the actual data. In our experiments, in order to better test the performance of TCFD, we set the value of β as a random value between 1 and 2. Compared with the setting of β in [11], the dataset used in our experiment is more challenging. More details of the methods for simulating the four fault types are shown in Table 3.

TABLE 3. Methods to simulate the four fault types.

Fault types	Simulation method
Outlier faults	Randomly extract 10% discrete data samples from the normal data set, and replace them with random numbers between 15 and 30
Gain faults	Randomly extract 300 continuous data samples from the normal data set, and replace them according to the gain formula $x'=\beta x+\eta$ (β is a random value between 1 and 2, η is a random value between 0 and 2)
Stuck-at faults	Randomly extract 300 continuous data samples from the normal data set, and replace them with the real reading collected just before the fault occurs
Offset faults	Randomly extract 400 continuous data samples from the normal data set, and replace them according to the offset formula $x'=\omega+x+\eta$ (ω is a random value between 5 and 10, η is a random value between 0 and 2)

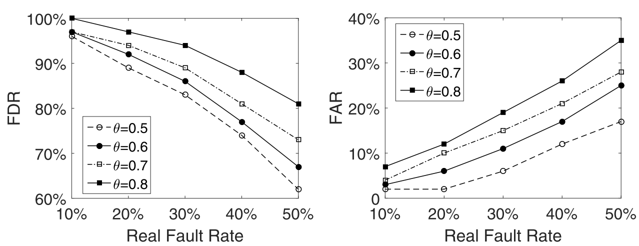
We use two metrics to evaluate the performance of the proposed scheme: 1) the fault detection rate (FDR): the ratio of the number of correctly detected faulty nodes to the total number of faulty nodes; 2) the false alarm rate (FAR) refers to the ratio of normal nodes that are misjudged as faulty nodes to all normal nodes. In the experiments, we observe the fault detection accuracy under different real fault rate (*i.e.,* proportion of real faulty nodes in the network).

B. SIMULATION RESULTS

1) KEY PARAMETERS IN TCFD

In this experiment, the impact of key parameters on the performance of TCFD is explored. For each group of parameter settings, a total of 20 tests are performed, and the proportions of the four fault types in the tests are the same.

From Fig.3, we can easily observe that as a threshold to judge whether the trends of two nodes are similar, θ has an important impact on the detection accuracy of the TCFD strategy. When the value of θ is small, the FDR is low. At the same time, the FAR is also at a relatively low level. With the increase of θ, both the FDR and the FAR will increase. It is not difficult to understand that when θ is a small value, the conditions for judging that the trends of two nodes are similar are quite loose. In this case, there is a high probability that a faulty node will not be detected because its data trend is judged to be similar to the normal nodes, and the probability of a normal node being misjudged as a faulty node due to different location will also be reduced. As θ increases, the sensitivity of the nodes to the trend correlation gradually increases, and the probability that the faulty node is determined to be similar in trend to the normal node also decreases, which promotes the increase of the FDR. The increase of θ will also reduce the probability of the data trend difference between normal nodes being tolerated, which will lead to an increase in the FAR. Considering the overall performance of TCFD in a balanced way, we set θ to 0.7 in the following experiments.

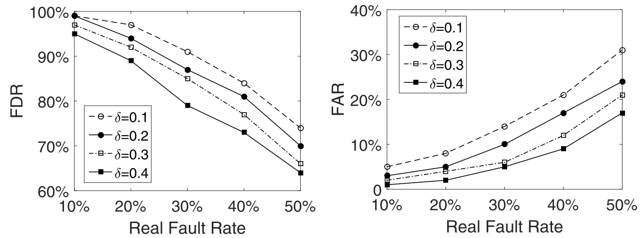


(a) fault detection rate (b) false alarm rate

FIGURE 3. Detection accuracy of TCFD varying the trend correlation threshold θ ($\delta = \gamma = 0.2$).

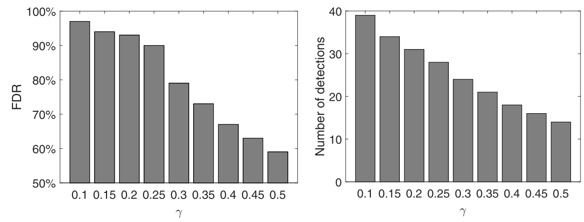
Fig.4 depicts the relationship between the median threshold δ and the detection accuracy of the proposed strategy. As can be seen, the value of δ has an important impact on the detection accuracy of the TCFD strategy. When the value of δ is small, both the FDR and the FAR are at a relatively high level. As the value of δ increases, the FDR and FAR gradually decrease. This is because the smaller the value of δ, the lower the probability that the node to be detected satisfies the judgment condition about the median threshold of the neighborhood. Considering the overall performance of TCFD in a balanced way, we set δ to 0.2 in this work.

Fig.5 shows the change of FDR and trigger times of fault detection varying γ. We can easily observe that as the value of γ increases, the FDR and detection trigger times decrease significantly. It is not difficult to understand that for any node



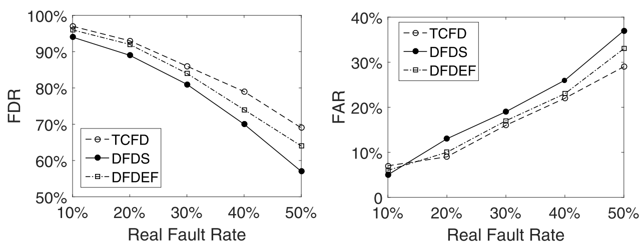
(a) fault detection rate (b) false alarm rate

FIGURE 4. Detection accuracy of TCFD varying the median threshold δ ($\theta = 0.7$, $\gamma = 0.2$).



(a) fault detection rate (b) number of detections

FIGURE 5. Change of FDR and trigger times of fault detection varying γ ($\theta = 0.7$, $\delta = 0.2$).



(a) fault detection rate (b) false alarm rate

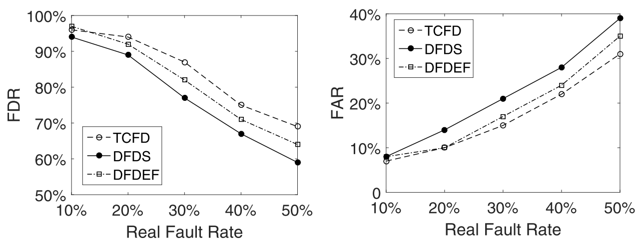
FIGURE 6. Detection accuracy of different strategies for outlier faults.

in the network, when there is a small deviation between the actual sensed data and the predicted value, the smaller the value of γ, the higher the possibility of triggering the fault detection mechanism, which can reduce the probability of missing faulty nodes. However, the increase in FDR is at the cost of an increase in the number of detections. Too frequent detections will increase the network routing overhead. Therefore, considering the tradeoff between detection accuracy and network routing overhead, this study takes $\gamma = 0.2$.

2) DETECTION ACCURACY

Fig.6 depicts the detection accuracy of different strategies when only outlier faults occur in the network. The three detection strategies all have high FDR for outlier faults, but with the increase in the real fault rate of the network, the FAR of the DFDS strategy decreases significantly. In contrast, the FDR of the TCFD strategy and the DFDEF strategy has not decreased significantly. This is because in these two strategies, in order to avoid misjudgments caused by the excessively high proportion of faulty neighboring nodes, the possible status of neighboring nodes is estimated. If more than half of the neighboring nodes are considered to be likely to be faulty, the original detection result will be corrected to reduce the risk of misjudgment.

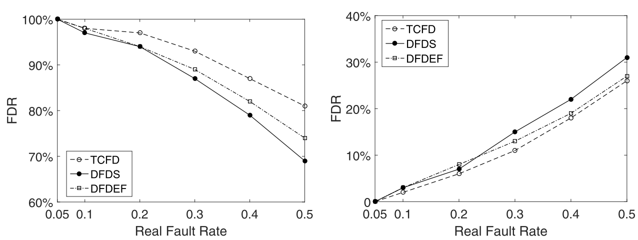
Fig.7 shows the difference in detection performance of each strategy when only gain faults occur. Compared with outlier faults, the detection accuracy of the three strategies for gain faults has declined to a certain extent. This is because in the gain faults, the degree of damage to the data trend in the continuous time segment is not obvious, and the difference in the sensed values of neighboring nodes is small, making the fault detection relatively difficult. Nevertheless, due to the introduction of the neighborhood median as the fault judgment condition, the detection performance of the TCFD strategy for offset faults is still better than the other two strategies.



(a) fault detection rate (b) false alarm rate

FIGURE 7. Detection accuracy of different strategies for gain faults.

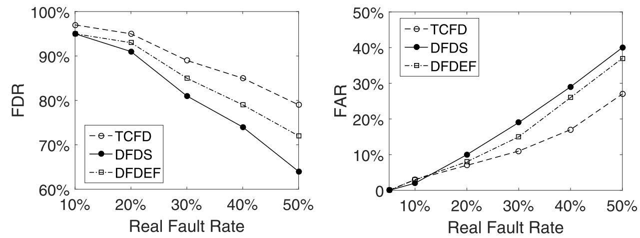
Fig.8 shows the performance of the three detection strategies when only stuck-at faults occur. When the real fault rate is 0.1, the FDR of the TCFD strategy is 97%, which is slightly better than the other two strategies. In the case where the real fault rate rises to 50%, the performance advantage of the TCFD strategy becomes more prominent. In this case, the FDR of the TCFD strategy is 81%. In contrast, the FDR of the DFDEF strategy and the DFDS strategy are only 73% and 67%, respectively. Although the FAR of the three strategies have increased significantly with the increase in the real fault rate, the increase of FAR in the TCFD strategy is still much lower than the other two strategies.



(a) fault detection rate (b) false alarm rate

FIGURE 8. Detection accuracy of different strategies for stuck-at faults.

Fig.9 shows the performance of different detection strategies when only offset faults occur. The detection accuracy of the TCFD strategy is much better than the other two strategies. Even if the real faulty rate in the network reaches 0.5, the FDR of the TCFD strategy is still higher than 75%, and the FAR is lower than 30%.



(a) fault detection rate (b) false alarm rate

FIGURE 9. Detection accuracy of different strategies for offset faults.

FIGURE 10. Fault response time before and after using the fault detection self-starting mechanism.

In summary, we can easily conclude that for outlier faults and gain faults, the difference in detection accuracy of the three strategies is not so significant. This is because outlier faults and gain faults do not change the general trend of sensed data, which makes these two types of faults can only be detected by comparing with the instantaneous sensed value of neighboring nodes. For stuck-at faults and offset faults, the data trend in the continuous time segment has been destroyed. With the help of the trend correlation judgment, TCFD can obtain better detection accuracy than the other two strategies.

3) RESPONSE TIME FOR FAULT DETECTION

In this experiment, the fault response time of the TCFD strategy before and after using the fault detection self-starting mechanism is tested. The fault response time is the time interval from when a fault occurs to when it is successfully detected. Before the fault detection self-starting mechanism is used, the node still uses the traditional timing trigger method to start the fault detection process. Considering the influence of the fault detection frequency on the detection performance of the strategy, the detection time interval T_D set by the timer is set to 1 min.

As shown in Fig.10, using the fault detection self-starting mechanism can significantly shorten the fault response time of the TCFD algorithm. For outlier faults and offset faults, the fault response time is less than 10s. It is not difficult to understand that these two types of faults will cause a serious deviation between the actual value and the expected value at the next moment, so the fault detection mechanism can be easily triggered. For gain faults and fixed value faults, the difference between the actual value and the expected value is relatively small, which makes it more difficult for the fault detection self-starting mechanism to be triggered. This will lead to a longer fault response time.

C. COMPLEXITY ANALYSIS

In this part, the time and message complexity of TCFD are analyzed. The time complexity and message complexity refer to the maximum time and maximum number of messages required to detect a faulty node, respectively.

1) MESSAGE COMPLEXITY

The proposed fault detection strategy TCFD runs in two stages: preliminary detection and final detection. Without loss of generality, we assume that node i is the node to be detected. During the stage of preliminary detection, node i broadcasts a request message to its neighboring nodes and obtains their sensor readings. The total number of messages generated in this stage depends upon the number of neighbors that node i has, and thus this stage has the message complexity of $O(l)$. l is the number of neighbors owned by node i. In the stage of final detection, the neighboring nodes of node i broadcasts a request message to their neighboring nodes for double-check. The number of messages generated in this stage depends upon the number of neighbors that the neighbors of node i have, and thus this stage has the message complexity of $O(lk)$. k is the average number of neighbors owned by the neighbors of node i. Considering the time complexity required for these two stages, we can easily get the overall message complexity of TCFD is $O(lk)$.

2) TIME COMPLEXITY

The time required to complete a fault detection equals to the sum of times spent on neighbor discovery, nodes ranking, status message exchange and data processing. The neighbor discovery can be negligible in fault detection since this step can be done in the network initialization phase. Since TCFD does not need complicated computing in nodes ranking and processing, the times for these two steps can also be negligible. Therefore, the main time is mainly spent on data transmission. The data transmission of TCFD includes four stages. We still assume that node i is the node to be detected. The first stage is the data transmission between node i and its neighboring nodes for preliminary detection, and we use the time T_{PR} to represent the time spent in this stage. The second stage is the data transmission between neighbors of node i and their neighbors, and we use the time T_{NN} to represent the time spent in this stage. The final stage is the data transmission from the neighbors of node i to node i for double-check, and we use the time T_{DC} to represent the time spent in this stage. As a result, the time complexity of TCFD is $T_{PR}+T_{NN}+T_{DC}$.

VI. CONCLUSION AND FUTURE WORK

In this work, we propose a distributed fault detection strategy TCFD based on trend correlation. By introducing trend correlation judgment and neighborhood median judgment in the strategy, the impact of the trigger time of fault detection on the detection accuracy can be alleviated. Based on the cubic exponential smoothing method, a fault detection self-starting mechanism is designed to reduce the response time of nodes to faults. Experimental results have shown that compared with existing strategies DSFD and DFDEF, the proposed TCFD can obtain better fault detection accuracy for four common fault types of sensor nodes; in the case where the real fault rate of the network reaches 0.5, at least 70% of the faulty nodes can be detected by TCFD and the false alarm rate can still be kept below 30%; with the help of fault detection self-starting mechanism, the response time of sensor nodes to faults can be significantly shortened. Since the proposed TCFD does not require complex calculations and can have good detection accuracy through the comparison of sensed values in the neighborhood, it can be regarded as a efficient lightweight fault detection solution for WSNs.

In the next step of our work, we plan to improve the TCFD strategy so that it can be used in clustered WSNs. Because the election of cluster heads is dynamic in clustering WSNs, how to reduce the impact of dynamic clustering on detection accuracy will be the key problem we need to solve.

REFERENCES

[1] S. Lata, S. Mehfuz, S. Urooj, and F. Alrowais, "Fuzzy clustering algorithm for enhancing reliability and network lifetime of wireless sensor networks," *IEEE Access*, vol. 8, pp. 66013–66024, 2020.
[2] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, "A survey on fault diagnosis in wireless sensor networks," *IEEE Access*, vol. 6, pp. 11349–11364, 2018.
[3] D. Ciuonzo, P. S. Rossi, and P. Willett, "Generalized Rao test for decentralized detection of an uncooperative target," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 678–682, May 2017.
[4] X. He, X. Fu, and Y. Yang, "Energy-efficient trajectory planning algorithm based on multi-objective PSO for the mobile sink in wireless sensor networks," *IEEE Access*, vol. 7, pp. 176204–176217, 2019.
[5] R. Niu and P. K. Varshney, "Performance analysis of distributed detection in a random sensor field," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 339–349, Jan. 2008.
[6] X. Fu, G. Fortino, W. Li, P. Pace, and Y. Yang, "WSNs-assisted opportunistic network for low-latency message forwarding in sparse settings," *Future Gener. Comput. Syst.*, vol. 91, pp. 223–237, Feb. 2019.
[7] M. Safaei, S. Asadi, M. Driss, W. Boulila, A. Alsaeedi, H. Chizari, R. Abdullah, and M. Safaei, "A systematic literature review on outlier detection in wireless sensor networks," *Symmetry*, vol. 12, no. 3, p. 328, Feb. 2020.
[8] X. Fu, G. Fortino, P. Pace, G. Aloi, and W. Li, "Environment-fusion multipath routing protocol for wireless sensor networks," *Inf. Fusion*, vol. 53, pp. 4–19, Jan. 2020.
[9] D. Ciuonzo and P. Salvo Rossi, "Distributed detection of a non-cooperative target via generalized locally-optimum approaches," *Inf. Fusion*, vol. 36, pp. 261–274, Jul. 2017.
[10] X. Fu, Y. Yang, and O. Postolache, "Invulnerability of clustering wireless sensor networks against cascading failures," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1431–1442, Jun. 2019.
[11] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through SVM classifier," *IEEE Sensors J.*, vol. 18, no. 1, pp. 340–347, Jan. 2018.
[12] Y. Gao, F. Xiao, J. Liu, and R. Wang, "Distributed soft fault detection for interval Type-2 Fuzzy-Model-Based stochastic systems with wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 334–347, Jan. 2019.
[13] R. R. Swain and P. M. Khilar, "Composite fault diagnosis in wireless sensor networks using neural networks," *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 2507–2548, Aug. 2017.

[14] H. S. Z. Kazmi, N. Javaid, M. Awais, M. Tahir, S. Shim, and Y. B. Zikria, "Congestion avoidance and fault detection in WSNs using data science techniques," *Trans. Emerg. Telecommun. Technol.*, vol. 19, no. 10, p. 3756, Oct. 2019.

[15] L. Yang, Y. Lu, S. X. Yang, T. Guo, and Z. Liang, "A secure clustering protocol with fuzzy trust evaluation and outlier detection for industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, early access, Aug. 25, 2020, doi: 10.1109/TII.2020.3019286.

[16] T. Goelles, B. Schlager, and S. Muckenhuber, "Fault detection, isolation, identification and recovery (FDIIR) methods for automotive perception sensors including a detailed literature survey for lidar," *Sensors*, vol. 20, no. 13, p. 3662, Jun. 2020.

[17] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors J.*, early access, Oct. 7, 2020, doi: 10.1109/JSEN.2020.3029459.

[18] X. Fu, H. Yao, and Y. Yang, "Modeling and optimizing the cascading robustness of multisink wireless sensor networks," *IEEE Trans. Rel.*, early access, Nov. 17, 2020, doi: 10.1109/TR.2020.3024797.

[19] R. Yarinezhad and S. N. Hashemi, "Distributed faulty node detection and recovery scheme for wireless sensor networks using cellular learning automata," *Wireless Netw.*, vol. 25, no. 5, pp. 2901–2917, Jul. 2019.

[20] M. Ding, D. Chen, K. Xing, and X. Chen, "Localized fault-tolerant event boundary detection in sensor networks," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies*, Mar. 2005, pp. 902–913.

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

[22] M. Saihi, B. Boussaid, A. Zouinkhi, and M. N. Abdelkrim, "Decentralized fault detection in wireless sensor network based on function error," in *Proc. 10th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Mar. 2013, pp. 1–5.

[23] Z. Feng, J. Q. Fu, and Y. Wang, "Weighted distributed fault detection for wireless sensor networks based on the distance," in *Proc. 33rd Chin. Control Conf.*, Jul. 2014, pp. 322–326.

[24] M. Panda and P. M. Khilar, "Distributed self fault diagnosis algorithm for large scale wireless sensor networks using modified three sigma edit test," *Ad Hoc Netw.*, vol. 25, pp. 170–184, Feb. 2015.

[25] H. Yuan, X. Zhao, and L. Yu, "A distributed Bayesian algorithm for data fault detection in wireless sensor networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2015, pp. 63–68.

[26] L. Mo, J. Li, G. Wang, and L. Chen, "Passive diagnosis for WSNs using time domain features of sensing data," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 6, Jun. 2015, Art. no. 590430.

[27] M. M. Gharamaleki and S. Babaie, "A new distributed fault detection method for wireless sensor networks," *IEEE Syst. J.*, vol. 14, no. 4, pp. 4883–4890, Dec. 2020.

[28] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 78, pp. 267–287, Jan. 2017.

[29] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction speech Processing*. New York, NY, USA: Springer, 2009, pp. 1–4.

[30] H.-A. Li, M. Zhang, K. Yu, J. Zhang, Q. Hua, B. Wu, and Z. Yu, "Combined forecasting model of cloud computing resource load for energy-efficient IoT system," *IEEE Access*, vol. 7, pp. 149542–149553, 2019.

[31] W. Hussain and O. Sohaib, "Analysing cloud QoS prediction approaches and its control parameters: Considering overall accuracy and freshness of a dataset," *IEEE Access*, vol. 7, pp. 82649–82671, 2019.

[32] G. B. Tayeh, A. Makhoul, C. Perera, and J. Demerjian, "A spatial-temporal correlation approach for data reduction in cluster-based sensor networks," *IEEE Access*, vol. 7, pp. 50669–50680, 2019.

[33] X. Fu and Y. Yang, "Modeling and analyzing cascading failures for Internet of Things," *Inf. Sci.*, vol. 545, pp. 753–770, Feb. 2021.

[34] X. Fu, H. Yao, and Y. Yang, "Modeling and analyzing cascading dynamics of the clustered wireless sensor network," *Rel. Eng. Syst. Saf.*, vol. 186, pp. 1–10, Jun. 2019.

[35] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *Proc. 6th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process.*, Dec. 2010, pp. 269–274.

XIUWEN FU (Member, IEEE) received the Ph.D. degree in mechatronics from the Wuhan University of Technology, China, in 2016. He was a Visiting Researcher with the University of Calabria. He is currently a Lecturer with the Shanghai Maritime University, China. He has authored more than 40 articles in international publications. His research interests include opportunistic networks, wireless sensor networks, and complex network systems.

YE WANG received the B.S. degree in electrical engineering and automation from Liaoning Technical University, China. He is currently pursuing the M.S. degree in control theory and control engineering with Shanghai Maritime University. His research interests include wireless sensor networks, complex network systems, and network modeling.

WENFENG LI (Senior Member, IEEE) received the Ph.D. degree from the Wuhan University of Technology, China, in 2000. He worked with the Royal Institute of Technology of Sweden, as a Visiting Scholar, in 2003. He was a Visiting Professor with the New Jersey Institute of Technology, in 2012, and a Visiting Professor with the CUSP, New York University, in 2015. He set up the Institute of Logistics and Robotics, the Center of IoT and Logistics Technologies, and the international joint Laboratory of Internet of Things. He is currently a Professor with the School of Logistics Engineering, Wuhan University of Technology. He holds more than 20 patents and has published more than 200 academic articles and six academic books. His current research interests include the IoT and robotics, modeling, and simulation of logistics systems.

YONGSHENG YANG received the Ph.D. degree in mechanical engineering from the Nanjing University of Aeronautics and Astronautics, in 1998. He is currently a Professor with the Institute of Logistics Science and Engineering, Shanghai Maritime University, China. He has authored more than 70 articles in international publications. His research interests include distributed network systems and optimization of logistics systems.

OCTAVIAN POSTOLACHE (Senior Member, IEEE) received the degree in electrical engineering and the Ph.D. degree from the Gh. Asachi Technical University of Iasi, Romania, in 1992 and 1999, respectively. He is currently a Professor with the ISCTE-Lisbon University Institute, Portugal. He is the author or coauthor of ten patents, ten books, 18 book chapters, and 320 articles.