A Reliable Information Fusion Algorithm for Reputation Based Wireless Sensor Networks

Teng Ma 1, Yun Liu 1,*, Junsong Fu 1 and Ya Jing 2

1 School of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing, 100044, China
2 State Grid Jibei Electric Power Co., Ltd. Material Branch (North China Power Equipment & Material General Corp.), State Grid, Beijing, 100075, China

1 [08111003, liuyun*, 12120067]@bjtu.edu.cn
2 jing.ya@nc.sgcc.com.cn

Abstract

In wireless sensor networks (WSNs), cryptographic primitives alone cannot provide a sufficient solution to the secure information fusion problem, therefore reputation systems have been introduced into WSNs. In a cluster, each sensor node has a single reputation value that is evaluated by the other sensor nodes in the same cluster. In this paper, we propose a novel, reliable information fusion algorithm, called reputation-driven information fusion (RDIF). In this work, a clustering algorithm is employed to divide all of the sensor nodes into many clusters. Then, a reputation system is established for each cluster, and an information fusion algorithm driven by reputation values is performed by the cluster head. In addition to the sensor nodes’ reputation values, we also consider the values of the readings collected by the sensor nodes and eliminate the outliers before fusing information. The simulation results show that RDIF can improve the reliability and accuracy of fusion results significantly when some compromised nodes appear in the WSNs.

Keywords: Information fusion; wireless sensor networks; reputation systems

1. Introduction

Wireless Sensor Networks, i.e., networks of wirelessly interconnected devices that are able to ubiquitously gather information from the environment, consist of large numbers of resource-constrained sensor nodes. Lifetime is one of the most important parameters studied extensively in WSNs. A property of WSNs is that neighboring sensor nodes, especially the sensor nodes in a cluster, often have overlapping sensing ranges, and, therefore, some data generated by neighboring sensor nodes are redundant. It takes a lot of energy to transmit all the data collected by the sensor nodes to the users of WSNs. In addition, the large number of sensor nodes generates an enormous amount of data for the base station to process. Fortunately, information fusion techniques can be used to obtain high-quality data and limit energy consumption. Information fusion is one of in-network data processing techniques, and it can combine the data from several sources to eliminate redundant information. Only the fusion results are provided to the base stations, which are bridges from WSNs to the users.

Many WSNs have been deployed in remote regions to conduct critical tasks, and, therefore, the security of the information fusion results is vital for these networks. However, cryptographic primitives alone cannot provide a sufficient solution to secure the reliability of information fusion [1-3]. As a result, the reputation concept [4], which originated from sociology, can be used in the field of WSNs to overcome the...
shortcomings of cryptographic primitives. What is more, introducing the reputation into WSNs also can improve the accuracy of the information fusion results.

Traditionally, to improve the security and accuracy of fusion results, many information fusion algorithms considered only the values of monitoring data and neglected the reliability of the data. In [5], the authors presented a modified Bayesian approach that can automatically identify the inconsistency of sensors’ readings by the values of the readings only. When fusing two readings of sensor nodes, it has the effect of increasing the variance of the posterior distribution when reading from one of the sensors is inconsistent with respect to the other. Then, they find the spurious readings and eliminate them from data fusion process. However, in that approach, the probability that whether a reading is spurious is estimated based on the value of reading only.

To further improve the reliability and accuracy of information fusion results, in this work, we propose the RDIF algorithm which focuses attention both on the values of the readings collected by the sensor nodes and on the reputations of the sensor nodes. In RDIF, we use the reputation of a sensor node to act as the reputation of the sensor node’ reading. This is reasonable, because if a sensor node has a high reputation means that the quality of this sensor node is good and it is high likely that the readings of this sensor node is reliable and accurate. On the contrary, if a sensor node’s reputation value is very low, it is high likely that the sensor node has some defects, such as poor quality, large measurement errors or being compromised, and the readings of this sensor node is incredible. As the basement of RDIF, all of the sensor nodes in a WSN need to be grouped into clusters by their geographic locations. Many clustering algorithms have been presented in the literature [6-11]. LEACH-C algorithm is employed in this work to cluster all the sensor nodes [10]. To save the energy and reduce the transmission cost, a reputation system is automatically built and periodically updated for each cluster [4] rather than the whole network. In the process of building and updating the reputation system, a ‘watch-dog’ is used to monitor and record the neighbors’ behaviors. Then, the Beta function [12] is used to calculate the reputation value. When fusing the data collected by the sensor nodes in a cluster, the data provided by high-reputation sensor nodes have a greater weight, and a modified data clustering algorithm introduced from data mining is used to eliminate the outliers of the readings. At last, we use Bayesian information fusion algorithm to fuse the credible readings and get the information fusion result. The simulation results show that RDIF can improve the accuracy and reliability of the fusion results significantly compared to Bayesian information fusion.

The rest of the paper is organized as follows. In Section 2, the related work is surveyed. Section 3 presents the RDIF in detail. Performance evaluation of RDIF is presented in Section 4, and concluding remarks are made in Section 5.

2. Related Work

In this section, we present some important work on clustering algorithms for WSNs [6-11], decentralized reputation systems that can be used in WSNs [13-16, 17-21] and information fusion algorithms [22-26].

Clustering in WSNs involves selecting cluster heads (CHs) and assigning cluster members to them for efficient data relay, and it is an important method for managing the sensor nodes in a large WSN. Generally speaking, it is very difficult to manage large WSNs that contain thousands of sensor nodes. Clustering is an efficient method of controlling such large populations. As an example, we consider a large WSN that is composed of densely-deployed sensor nodes. Building a global reputation system to evaluate all the sensor nodes’ reputations is impossible, because every sensor node requires a lot of energy to communicate with the other sensor nodes, and a large space is required to store the other nodes’ reputation values. However, if we divide the WSNs into many clusters, it is easier to build a local reputation system, because each cluster consists
of a small number of sensor nodes. There are several other popular targets for network clustering, such as load balancing, fault-tolerance, increased connectivity, and reduced delay [6]. Many clustering algorithms for WSNs have been proposed. According to the convergence rate, clustering algorithms are divided into two groups. The group of variable convergence time algorithms includes the Linked Cluster Algorithm [7-8] and Random Competition Based Clustering [9]. The group of constant convergence time algorithms includes Low Energy Adaptive Clustering Hierarchy [10], Fast Local Clustering Service [11]. Based on clustering algorithms for WSNs, a large-scale WSN can be grouped into many small clusters, and it is easy to manage such clusters. The RDIF algorithm proposed in this paper will be operated in the small clusters rather than in a large WSN.

Primitive cryptographic primitives alone cannot provide a sufficient solution to a secure information fusion problem. The reputation concept, which originated in the field of sociology, can be used to overcome the shortcomings of cryptography-based information fusion systems. There are several centralized reputation systems for the Internet [13-14]. However, we need a decentralized reputation system in this paper. In [15], the authors proposed a reputation-based framework for high-integrity sensor networks. The framework employs a Bayesian formulation, specifically a beta reputation system, for reputation representation, updates, and integration. In [16], the Distributed Reputation-based Beacon Trust System (DRBTS) was proposed, and it is a distributed model that uses both first-hand and second-hand information. DRBTS was developed for secure localization by enabling sensor nodes to exclude location information from malicious beacon nodes by using a majority voting scheme. Many reputation systems have been proposed in the literature. In [17], the authors surveyed the trust and reputation management systems in wireless communications and briefly introduced a large number of reputation systems in the fields of mobile ad hoc networks (MANETs), WSNs, and cognitive radio networks (CRNs). The paper [18] studies the trust in WSNs and proposes a solution to estimate the trust based on communal reputation and individual trust (CRIT) in sensor nodes. A Gaussian trust and reputation system is introduced in [19]. There also are some other reputation-based frameworks and systems [20-21]. For different reputation systems, the reputation values of the sensor nodes are not all the same, for example, the reputation for a sensor node can be an integer or a decimal range from 0 to 1. For the sake of convenience, we assumed that the reputations of all the sensor nodes can be transformed to an integer range from 0 to 10.

Information fusion has been researched extensively and information fusion systems are used extensively in many areas, such as sensor networks, robotics, and multi-media processing. To date, various types of information fusion algorithms have been proposed, such as probability theory-based information fusion algorithms, Dempster-Sharfer (D-S) evidence theory-based information fusion algorithms, and fuzzy set theory-based information fusion algorithms [22]. There are dozens of mature information fusion algorithms, and we will pay attention to information fusion in reputation-based WSNs. Data integrity, confidentiality, and freshness are security issues that are required in many information fusion applications, particularly in the military [22]. Some protocols for secure information fusion have been proposed in the literature. A secure information fusion framework named Blind Information Fusion Framework (BIFF) was proposed in [23]. In BIFF, the sensor nodes are not aware of the actual information they are processing, but the Framework converges to the intended result(s). First, the data are transformed from normal space to anonymous space in which the data, once fused, cannot be deduced. In [24], the Random Offset Method (ROM) was proposed to ensure mutual privacy in distributed fusion systems based on a consensus averaging method. The secure problem also has been researched actively in the field of data aggregation [25-26].
3. RDIF Algorithm

In this section, we introduce RDIF in detail. As described in Section 1, RDIF is an in-cluster information fusion algorithm and performed in the cluster heads. Therefore, this work focuses on a cluster composed of some sensor nodes rather than the whole network. In the initial phase of RDIF, the whole network is divided into several clusters by clustering algorithms, such as LEACH-C [10]. In each cluster, the cluster head collects the monitoring readings of the sensor nodes and each sensor node’s reputation value. Note that, the cluster head don’t monitor the behaviors of a sensor node and get the reputation value by averaging the evaluation of the neighbors of the sensor node. Based on the monitoring readings and the reputation values of the sensor nodes in a cluster, we can get a reliable information fusion result by RDIF.

3.1. Background and Basic concepts

In a cluster, the sensor nodes upload their monitoring data to the cluster head periodically. Cluster heads obtain the reputation values of each sensor node in the cluster by the reputation system [4]. All of the monitoring data uploaded by sensor nodes and the reputation values of the sensor nodes are stored in the cluster node. The data structure of each object stored in the cluster head is shown as follows, and we call each piece of the information ‘metadata’.

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>Reputation</th>
<th>Monitoring data</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Rep</td>
<td>$X = {x_1, ..., x_n}$</td>
<td>{UNCLASSIFIED, OUTLIER or CLUSTERID}</td>
</tr>
</tbody>
</table>

In order to identify the sensor nodes in a cluster, each sensor node has a unique ID. The cluster head gets the reputation values of all of the nodes by the reputation system. Monitoring data are denoted by $X$ and their dimensions are decided by the property of the sensor nodes. For example, in order to monitor a forest fire, we need information about the temperature and humidity in the forest. We can scatter two kinds of sensor nodes, i.e., temperature sensor nodes and humidity sensor nodes. In this way, the dimension of $X$ is one, and $X$ represents the temperature or humidity. An alternative method is to scatter only one kind of sensor nodes that can detect both temperature and humidity. In this way, the dimension $X$ is two, and the two dimensionalities are temperature dimensionality and humidity dimensionality. In this paper, we assume that $X$ is two-dimensional. Note that RDIF also can be used in high-dimensional datasets. In order to delete the outliers and improve the security and accuracy of the information fusion results, we embedded a modified clustering algorithm in our algorithm. Then, we used ‘Type’ to record an object’s status. UNCLASSIFIED objects are the objects that have not been checked to determine whether they belong to a cluster or not. An object with Type CLUSTERID means that the object belongs to a cluster and that CLUSTERID is the cluster’s ID. If the Type of an object is OUTLIER, the object is an outlier, and it will be eliminated when objects are fused. The initial values of all of the objects’ Type are UNCLASSIFIED.

It is very difficult to maintain and update a reputation and trust system for the whole network, therefore, in this paper, we first divide all the sensor nodes into several clusters by LEACH-C algorithm [10]. After clustering, we need to build a reputation and trust system for each cluster. Reputation and trust systems are utilized extensively in various research fields for WSNs, such as sensing, sharing, routing, information modeling, decision making, and dissemination. The systems use nodes to watch and observer neighboring nodes’ action and evaluate nodes’ activities so
that they assess whether the nodes are compromised or not. Then, each sensor node uploads the reputations of its neighbors to the cluster head and the cluster head calculate the reputation value of each sensor node by averaging the neighbors’ evaluation.

In this paper, we mainly pay our attention on the beta reputation system. In beta reputation systems, reputation $R_{ij}$ is computed by sensor node $N_i$ using beta density function of sensor node $N_j$’s previous actions. For example, sensor node $N_i$ counts the number of good and bad actions of $N_j$ as $r_{ij}$ and $s_{ij}$. Then, $N_i$ records the reputation $R_{ij}$ about node $N_j$ as $R_{ij} = \text{Beta}(p|r_{ij}+1, s_{ij}+1)$ and the reputation value is $E(R_{ij}) = \frac{r_{ij}+1}{r_{ij}+s_{ij}+2}$, where Beta represents the Beta distribution which can be expressed by the gamma function $\Gamma$ as: $\text{Beta}(p|r_{ij}+1, s_{ij}+1) = \frac{\Gamma(r_{ij}+s_{ij}+2)}{\Gamma(r_{ij}+1)\Gamma(s_{ij}+1)} p^{r_{ij}}(1-p)^{s_{ij}}$, where $0 \leq p \leq 1$, $r_{ij}$, $s_{ij} \geq 0$. In a cluster, every node records and updates the reputation values of the neighbor nodes and sends the reputation values to the cluster head periodically. In this way, the cluster head that performs the information fusion programs can have all the nodes’ reputation values clearly. In fact, the reputation is a small number ranging from 0 to 1. However, for convenience sake, we transform the reputation values of a sensor node into an integer from 0 to 10 by linear transformation.

3.2. Selecting and Fusing Credible Readings

There are two steps in the RDIF algorithm. First, we select the credible data from the dataset that was generated by the sensor nodes by a call of SelectCredibleData(). The credible data are selected mainly based on their reputations and numerical values. Second, we fuse the credible data by Bayesian information fusion algorithm. We present the RDIF algorithm, which is designed to get credible and reliable fusion results from the monitoring data, in the following:

$$\text{RDIF}(\text{m pieces of metadata, Rad, MinRep})$$

1. SelectCredibleData (m pieces of metadata, Rad, MinRep)
2. Fuse the credible data through Bayesian Information Fusion Algorithm

As discussed previously, the dimension of $X$ changes with the type of sensors. Note that our algorithm applies to 2D or 3D Euclidean space and also to some high-dimensional feature space. However, for convenience, we assumed that the data points are two-dimensional in our simulation. Assume that there are dozens of pieces of metadata and that the $X$ of these metadata is a set of data points with a weight, i.e., the Reputation. That is to say, different points have different degrees of importance. For example, if a data point has a reputation 8, then its weight is 8.

In order to distinguish the credible data and outliers clearly, some basic definitions modified from [27] are given in the following:

**Definition 1:** (Rad-Neighbor of a data point, p) The Rad-Neighbor of a data point, $p$, denoted by $\text{Ne}_\text{Rad}(p)$, is defined by $\text{Ne}_\text{Rad}(p) = \{ q | S(\text{dist}(p,q) \leq \text{Rad}) \}$, where Rad is a parameter.

**Definition 2:** (directly density-reachable) A data point, $p$, is directly density-reachable from a point, $q$, with regard to $\text{Rad}$ and $\text{MinRep}$ if:

1) $p \in \text{Ne}_\text{Rad}(q)$ and
2) $\sum \text{Rep}(\text{Ne}_\text{Rad}(q)) \geq \text{MinRep}$,

where $\sum \text{Rep}(\text{Ne}_\text{Rad}(q))$ is the sum reputation of $q$’s neighbors.
Definition 3: (density-reachable) A data point, \( p \), is density-reachable from a point, \( q \), with regard to \( \text{Rad} \) and \( \text{MinPts} \) if there is a chain of points \( p_1, \ldots, p_n \). \( p_1 = q \). \( p_n = p \) such that \( p_{i+1} \) is directly density-reachable from \( p_i \).

Definition 4: (density-connected) A point \( p \) is density-connected to a point \( q \) with regard to \( \text{Rad} \) and \( \text{MinPts} \) if there is a point \( o \) such that both \( p \) and \( q \) are density-reachable from \( o \) with regard to \( \text{Rad} \) and \( \text{MinPts} \).

Definition 5: (cluster) Let \( S \) be a set of monitoring data \( X \). A cluster \( C \) with regard to \( \text{Rad} \) and \( \text{MinPts} \) is a non-empty subset of \( S \) satisfying the following conditions:

1. \( \forall p, q: \) if \( p \in C \) and \( q \) is density-reachable from \( p \) with regard to \( \text{Rad} \) and \( \text{MinPts} \), then \( q \in C \).
2. \( \forall p, q \in C: \) \( p \) is density connected to \( q \) with regard to \( \text{Rad} \) and \( \text{MinPts} \).

Definition 6: (target cluster) There are several clusters in the set of data, and each cluster contains some data points with different reputations. The cluster with the highest total reputation is the target cluster.

Definition 7: (credible data) Data in target cluster are credible data.

We present the algorithm as follows:

We will choose the credible data based on the definitions above, it is easy to develop an algorithm to select the credible data. We will choose the credible data based on their reputation and values. An algorithm named SelectCredibleData is designed to select credible data according to definitions. Algorithm SelectCredibleData starts with an arbitrary data point \( p \) and retrieves all points that are density-reachable from \( p \) with regard to \( \text{Rad} \) and \( \text{MinRep} \).

We present the algorithm as follows:

SelectCredibleData (m pieces of metadata, \( \text{Rad} \), \( \text{MinRep} \))

```java
1. MetaData = {metadata(1), ..., metadata(m)}
2. FOR i FROM 1 TO m
3. seed = MetaData.get(i)
4. IF seed.type == UNCLASSIFIED
5. seeds = MetaData.regionQuery(seed, Rad)
6. IF MetaData.size(seeds) < MinRep
7. MetaData.changeType(seed, OUTLIER)
8. ELSE
9. ClusterID = i
10. MetaData.changeType(seeds, ClusterID)
11. seeds.delete(seed)
12. WHILE (seeds.isNotEmpty == TRUE)
13. currentSeed = seeds.first();
14. result = MetaData.regionQuery (currentSeed, Rad)
15. IF MetaData.size(result) ≥ MinRep
16. FOR j FROM 1 TO result.size
17. currentResult = result.get(j)
18. IF currentResult.TYPE ==
19. (UNCLASSIFIED OR OUTLIER)
20. IF currentResult.TYPE ==
21. UNCLASSIFIED
22. seeds.append(currentResult)
23. END IF
24. MetaData.changeType
25. (currentResult, ClusterID)
26. END IF
27. END IF
28. END FOR
29. END IF
30. seeds.delete(currentSeed)
```
This algorithm will divide the data collected by the sensor nodes into several groups. We can distinguish the groups by parameter Type in each piece of metadata. The metadata that have a Type of OUTLIER are outliers. If one piece of metadata has a Type of an integer i, the piece of metadata is an object in cluster i. Generally speaking, there is only one cluster, and its core is made up of high-reputation data objects. However, if there are several clusters, there must be something wrong, because the monitoring environments are not all the same for all of the sensor nodes in a WSN cluster. Note that, if two data clusters \( C_1 \) and \( C_2 \) are very close to each other, it might happen that some point \( p \) belongs to both \( C_1 \) and \( C_2 \). Then \( p \) must be a border point in both cluster, and it will be assigned to the cluster discovered first.

The sub-program \( \text{MetaData.regionQuery}(seed, \text{Rad}) \) is shown as follows. A call of \( \text{MetaData.regionQuery}(seed, \text{Rad}, \text{MetaData}) \) returns the Rad-Neighbor of seed as a list of metadata. The run-time of our algorithm was \( O(n^2) \), which was longer than \( R^* \)-tree-, \( M^- \)-tree-, or \( X^- \)-tree-based region query algorithms. However, considering that the quantity of sensor nodes in a cluster is not that large, we need not to construct the complicated tree structure.

\[
\text{MetaData.regionQuery}(metadata, \text{Rad})
\]

1. FOR \( i \) FROM 1 TO \( \text{MetaData.size} \)
2. \( \text{IF } (\text{distance}(meta\text{data.X}, \text{MetaData.get(i).X}) \leq \text{Rad}) \)
3. \( \text{Neighbor.append(Meta\text{Data.get(i)})} \)
4. END IF
5. END FOR
6. RETURN \( \text{Neighbor} \)

A call of \( \text{MetaData.size}(metadata) \) returns the total reputation of a set of metadata. Note that, \( \text{MetaData.size}(metadata) \) is different from \( \text{MetaData.size} \), which is the number of metadata in \( \text{MetaData} \).

\[
\text{MetaData.size}(metadata)
\]

1. FOR \( i \) FROM 1 TO \( metadata.size \)
2. \( |\text{TotalRepofNet}| += \text{metadata.get(i).Rep} \)
3. END FOR
4. RETURN \( |\text{TotalRepofNet}| \)

There will be feedback to the reputation system after we divide the data into two excluded sets, i.e., credible data and outliers. The reputation of the sensor nodes remains
unchanged if the data uploaded by them are credible. If the data uploaded by a sensor are not credible, there will be negative feedback to the sensor’s reputation. Through the parameter ‘Type’ in metadata, we searched the most credible cluster and then fused the metadata in the most credible cluster.

In this work, each sensor node is assumed to have a measure error and we assume that the measurement value is a Gaussian distribution with a mean of the true physical phenomenon and a variance based on the reliability of the sensor node. Bayesian information fusion algorithm [5, 22, 28] can fuse pieces of information and lies at the core of probability based information fusion algorithms. We assume a state-space representation X which is the range of the fusion results, the Bayesian information fusion algorithm provides a method for computing the posterior conditional probability distribution/density of the hypothetical state $x_k$ given the set of measurements $Z = \{z_1, z_2, ..., z_n\}$ obtained from n independent sensor nodes and the prior distributions, as following:

$$p(x_k|Z) = \frac{p(x_k) \cdot p(Z|x_k)}{p(Z)},$$

where $Z$ is the set of the readings of the sensor nodes needed to be fused, $p(x_k)$ is the priori probability of $x_k$, $p(Z|x_k)$ is the decided by the sensor measurement model, $p(Z)$ is a constant value and it is merely a normalization term to ensure that the probability density function is integrate to 1. At last, we can use the rule of maximum a posteriori (MAP) or minimum mean square error (MMSE) [5] to get $x_k$.

### 3.3. Determining the Parameters Rad and MinRep

For different applications, the parameters Rad and MinRep are often different from each other, and an effective method is to set the parameters by experience. Even so, we need a universal method to roughly estimate the parameters. In the literature [27], the authors proposed an artificial way to determine the parameters Rad and MinRed. This method requires a lot of prior knowledge about the percentage of malicious data points, or the users can select a data point as the threshold point. However, for most of the users, it is very difficult the set the parameters. Therefore, in order to overcome these shortcomings of the artificial method, we design a more intelligent algorithm to set the parameters automatically.

In a cluster, the sensor nodes have different reputations, and we treat the reputation values as the weights of readings collected by the sensor nodes. A sensor node with a weight $w$ is equal to $w$ sensor nodes with a weight of 1. In this way, we can transform the sensor nodes with different weights to sensor nodes with no weight, i.e., all of their weights are 1. Let $d$ be the distance of a point $p$ to its $k$th nearest neighbor; then, the $d$-neighbor of $p$ contains approximately $k$ sensor nodes. For a given $k$ we define a function $k$-dist from MetaData to the real numbers, mapping each point to the distance from its $k$th nearest neighbor. When sorting the data points of the MetaData in descending order of their $k$-dist values, the graph of this function gives some hints concerning the density distribution in the MetaData. We call it the sorted $k$-dist graph.
As in [27], the threshold point is the first point in the first “valley” of the sorted k-dist graph (Figure 1). In general, it is difficult to find the “valley” automatically, but it is easy for users to see the valley in a graph. In [27], the authors proposed an interactive approach for determining the “valley.” In this paper, we proposed a more intelligent algorithm to set the parameters of $Rad$ and $MinRep$.

Figure 2 shows that the k-dist graphs for $k > 45$ are not significantly different from the 45-dist graph for 2-dimensional data points and the reputations of the data points randomly distributed from 1 to 10. Therefore, we set the parameter $MinRep$ to 45 for all 2-dimensional data points, and, then, we can determine the parameter $Rad$ by the algorithm in the following.

DetermineParameter(MetaData)

1. FOR $i$ FROM 1 TO $MetaData.size$
2.     $metadata = MetaData.get(i)$
3.     Sort the $MetaData$ by the distance to $metadata$
4.     Get $\{metadata\}_1, ..., metadata_{MetaData.size}$
5.     FOR $i$ FROM 1 TO $MetaData.size$
6.         WHILE($neighbors += metadata_{i} \geq 45$)
7.             $Distance.append(dist(metadata, metadata_{i}))$
8.     END WHILE
9. END FOR
10. END FOR
11. Sort $Distance$ in descending order to $y$
   \[
   \{ Distance_1, \ldots, Distance_{Distance.size}\}
   \]
12. $x = [1 \ 2 \ 3 \ \ldots \ Distance.size]$
13. $y' = \text{ployfit}(x, y)$
14. FOR $i$ FROM 1 TO $Distance.size$
15. IF $(y'_i - y'_{i+1} \leq (Distance_1 - Distance_{Distance.size})/Distance.size)$
16. thresholdpoint $= (i, y'_i)$
17. $Rad = y_i$
18. BREAK
19. END IF
20. END FOR
21. RETURN $Rad$

A call of $\text{ployfit}(x, y)$ returns the secondary-order polynomial fitting result of a set of data points. Our experiments showed that the performance of second-order polynomial fitting is the best compared to other fitting methods. The simulation result of $\text{DetermineParameter(MetaData)}$ is shown in Figure 4.

![Figure 3. Threshold Point Detected by the Algorithm of DetermineParameter](image1)

We first find the threshold point as shown in Figure 3 and then get the parameter $Rad$. The result of $\text{DetermineParameter(MetaData)}$ is shown in Figure 3 and $Rad$ is equal to 4.2791.

![Figure 4. Sorted k-dist Graph for 3-dimensional Data Points](image2)
For the three-dimensional data points, Figure 4 shows that the results of our experiments were similar to the two-dimensional data points shown in Figure 2. So the parameter MinRep for 3-dimensional data points also can be set to 45, and Rad also can be obtained by DetermineParameter. In fact, our conjecture is that the k-dist graph has an important relationship with the number of data points and the distances between them rather than the dimensions of the data points. We will do more research on this issue in the future.

4. Simulations

The RDIF algorithm is simulated using the ns-3 (version 3.21) [29]. Though, RDIF is a universal information algorithm, in this work, we design a WSN is to monitor the temperature and humidity of an interesting terrain and further, to monitor fire. The WSN is a reputation-based WSN consist of 500 homogeneous sensor nodes. The sensor nodes in the network were randomly distributed in a square area in which the base station was located on one corner, and each sensor node are assumed to be capable of storing and processing data. As discussed previously, the RDIF algorithm is operated in a cluster rather than the whole network, since the processes of information fusion between clusters are independent. Therefore, in the simulation, we focused our attention only on one of the WSN clusters that was composed of 50 sensor nodes. Each of the sensor nodes in the cluster can monitor both the temperature and humidity of the environment simultaneously.

In the simulation, most of the sensor nodes (about 80%) are credible, and a small percentage of the sensor nodes are not credible (about 20%). The non-credible sensor nodes in the network send false data to the cluster head, inject false data during information fusion, temper the data, and drop the packets with a constant probability of 0.5. In this work, for the sake of convenience, we assumed that the reputation system only monitored the actions of tempering with data and dropping packets. The reputations of the sensor nodes, which were generated by the reputation system, are integers from 1 to 10. The accurate temperature of the area that we monitored was 25 oC and the relative humidity was 40%. Each sensor has measure error and we assume that the measurement value is a Gaussian distribution with a mean and a variance. The means of all the sensor nodes are the true value of the environment and the variance for a credible sensor nodes is set to 2 and for a malicious sensor nodes is set to five 5.

As presented in 3.2, we employed the modified DBSCAN algorithm to find the suspicious readings of the sensor nodes. As a result, the similarity between the readings needs to be defined first. In the simulation, Euclidean distance is used to measure the similarity between two data points. A problem of Euclidean distance is that some features with large amplitude often cover the effect of other features with small amplitude. Therefore, a preprocessing is used to normalize the original n data points in the data set

\[ x_i^j = \frac{x_i^j - x_{\text{min}}^j}{x_{\text{max}}^j - x_{\text{min}}^j} \tag{2} \]

where \( x_i^j \) is the jth feature of data point \( x_i \), \( x_{\text{min}}^j = \min(x_1^j, x_2^j, \ldots, x_n^j) \), \( x_{\text{max}}^j = \max(x_1^j, x_2^j, \ldots, x_n^j) \). The distance between data points \( x_i \) and \( x_j \) is defined by

\[ \text{dist}(x_i, x_j) = \sqrt{\sum (x_i^d - x_j^d)^2} \tag{3} \]

where \( d \) is the dimension of the data points. After detecting the outliers, the normalized data points need to be restored by

\[ x_i^j = x_i^j * (x_{\text{max}}^j - x_{\text{min}}^j) + x_{\text{min}}^j \tag{4} \]

The original data are shown in Fig. 5(a) and the normalized data are shown in Fig. 5(b).
Obviously, the monitoring data of the environment are 2-dimensional, which is shown in Figure 5(a), and the data after reshaping are shown in Figure 5(b). In Figure 5(b), we find a clear phenomenon that the data points that have high reputations are usually close to the center, vice versa. This phenomenon suggests that the sensor nodes that have low reputations have a high probability to generate some false data. Intuitively, the false data should be eliminated in the process of information fusion. We selected the credible data by the SelectCredibleData algorithm, and the results are shown in Figure 6.

We determined the parameters Rad and MinPts by the method that was introduced in Section 3.3. The parameter MinPts was set to 45, and the parameter Rad was set to 0.1. In Figure 6, the data points that are covered by green circles are credible. The other data points are outliers, and they will be eliminated in the process of information fusion.

We used the Bayesian information algorithm discussed in Section 3.2 to fuse the data. In our simulation, Z is the set of the temperature and humidity readings of the sensor nodes, \( p(Z|x_k) \) is set to be a Gaussian distribution as discussed previously. In addition, \( p(x_k) \) is set to be a uniform distribution in the range of temperature and humidity and the rule of MAP is use to get \( x_k \). We conducted the experiment 50 times, and compare the results of RDIF with that of Bayesian information fusion and the method in [5].
As Figure 7 and 8 show, in most cases, RDIF outperforms Bayesian information fusion algorithm and the method in [5]. Only in a few cases, the RDIF made the results of information fusion even worse. However, we can ignore the exceptional cases because they make up a very small proportion of the total number of cases.

### Table 2. Simulation Results

<table>
<thead>
<tr>
<th></th>
<th>Temperature (℃)</th>
<th>Humidity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Variance</td>
</tr>
<tr>
<td>Bayesian</td>
<td>24.8565</td>
<td>0.8151</td>
</tr>
<tr>
<td>The method</td>
<td>25.0687</td>
<td>0.2611</td>
</tr>
<tr>
<td>in [5]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDIF</td>
<td>25.0262</td>
<td>0.1295</td>
</tr>
<tr>
<td>True value</td>
<td>25.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
In order to compare the approaches more clearly, we present the means and the variances of the fifty results. In Table 2, we can find that RDIF’s performance is slightly better than the method in [5] which performs better than the Bayesian information fusion algorithm.

Another important parameter that affects the accuracy of the fusion result is the percentage of the compromised sensor nodes. In our simulation, the accuracy of information fusion is defined as follows:

\[
\text{accuracy} = 1 - \frac{|f - r|}{r},
\]

(5)

where \(f\) is the result of information fusion, and \(r\) is the true value of the environment. There was no ambiguity if the data is one-dimensional. However, in this paper, the data is two-dimensional including temperature and humidity. For \(d\)-dimensional (\(d \geq 2\)) data, we defined \(f\) and \(r\) as follows:

\[
f = \sqrt{f_1^2 + f_2^2 + \cdots + f_d^2},
\]

(6)

\[
r = \sqrt{r_1^2 + r_2^2 + \cdots + r_d^2},
\]

(7)

where \(f_i\) and \(r_i\) are the \(i\)th component of \(f\) and \(r\), respectively. We defined the distance of two vectors by the Euclidean distance in this paper. However, there are several other types of distance, such as Mahalanobis distance and Manhattan distance. The proper type of distance should be chosen for different cases. To reduce the randomness, for each percentage of the credible nodes, we conduct the experiment for 10 times and the average result of them is presented. As Figure 9 shows, the percentage of compromised sensor nodes has significant influence on the three approaches.

![Figure 9. Accuracy of Information Fusion Results](image)

Figure 9 shows that the accuracy of the fusion result increased for all the three approaches. However, the performance of RDIF algorithm was the best. We did not simulate our algorithm when the percentage of credible sensor nodes was less than 60%. In fact, if half or more of the sensor nodes in a cluster are compromised sensor nodes, the cluster is useless and the reputation system will also be paralyzed, because the non-credible sensor nodes make up a larger proportion than the credible sensor nodes.
5. Conclusions and Future Work

Multi-sensor information fusion in WSNs has been researched extensively in recent years. In this paper, we propose a novel information fusion algorithm based on outlier detection to extract credible data by considering both the reputation of the sensor nodes and the values of the readings. Then, we fuse the credible data using the Bayesian fusion algorithm. Simulation shows that the RDIF performs well in accuracy; this can be explained by the fact that RDIF can identify non-credible outliers. Obviously, because of the reputation systems, the security of WSNs has also been improved. As the future work, we plan to design a more suitable reputation system for WSNs and a more lightweight multi-sensor information fusion algorithm.

Our results indicate that the algorithm we developed for determining the parameters Rad and MinPts performed better than the method in [27]. We focused mainly on 2-dimensional and 3-dimensional data points. We also conjectured in this work that the k-dist graph has an important relationship with the number of data points and the distances between them rather than the dimensions of the data points. In future work, we will do more research on high-dimensional problems.

Acknowledgements

This research is supported by National Natural Science Foundation under Grant 61371071, Beijing Natural Science Foundation under Grant 4132057, Beijing Science and Technology Program under Grant Z121100007612003, Academic Discipline and Postgraduate Education Project of Beijing Municipal Commission of Education.

References


Authors

Teng Ma, is currently pursuing the PhD degree with Department of Information and Communication Engineering in Beijing Jiaotong University. His research interests include wireless sensor network, Internet of Things and green networking.

Yun Liu, is a Professor of Communication and Information Systems, Beijing Jiaotong University; Dean of Communication Engineering Department, Beijing Jiaotong University; Director of key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education; Director of Institute of Network Consensus Security, Beijing Jiaotong University; Vice-chair of Teachers and Staff Representative Committee, Beijing Jiaotong University. She is currently a Fellow of IET, UK and specialist enjoying special government allowance. In addition, she is an evaluation expert of State Scientific and Technological reward, State Natural Sciences Fund in communication, National High
Technology Research and Development Program (HTRDP), an advanced counselor of China Tietong Company, and an advanced counselor of Beijing Municipal Office of Internet Propaganda and Management.

**Jun Song Fu** received his bachelor’s degree in communication engineering in 2012 from Beijing Jiaotong University, China. Currently, he is a doctoral student in Beijing Jiaotong University. He has been doing research in the field of wireless sensor networks, reputation systems, data fusion and data mining algorithms.

**Ya Jing** received her Master Degree in Communication and Information System in 2009 from Beijing Jiaotong University. She is in charge of enterprise data security & informationization construction since 2009 in State Grid Jibe Electric Power CO.LTD. Material Branch (North China Power Equipment & Material General Corp.), State Grid, Beijing.