

Research Article

VSMURF: A Novel Sliding Window Cleaning Algorithm for RFID Networks

He Xu,^{1,2} Weiwei Shen,^{1,2} Peng Li,^{1,2} Daniele Sgandurra,³ and Ruchuan Wang^{1,2}

¹School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

²Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China

³Information Security Group, Royal Holloway, University of London, Surrey TW200EX, UK

Correspondence should be addressed to He Xu; xuhe@njupt.edu.cn

Received 9 February 2017; Revised 24 May 2017; Accepted 22 June 2017; Published 27 July 2017

Academic Editor: Nashwa El-Bendary

Copyright © 2017 He Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Radio Frequency Identification (RFID) is one of the key technologies of the Internet of Things (IoT) and is used in many areas, such as mobile payments, public transportation, smart lock, and environment protection. However, the performance of RFID equipment can be easily affected by the surrounding environment, such as electronic productions and metal appliances. These can impose an impact on the RF signal, which makes the collection of RFID data unreliable. Usually, the unreliability of RFID source data includes three aspects: false negatives, false positives, and dirty data. False negatives are the key problem, as the probability of false positives and dirty data occurrence is relatively small. This paper proposes a novel sliding window cleaning algorithm called *VSMURF*, which is based on the traditional SMURF algorithm which combines the dynamic change of tags and the value analysis of confidence. Experimental results show that VSMURF algorithm performs better in most conditions and when the tag's speed is low or high. In particular, if the velocity parameter is set to 2 m/epoch, our proposed VSMURF algorithm performs better than SMURF. The results also show that VSMURF algorithm has better performance than other algorithms in solving the problem of false negatives for RFID networks.

1. Introduction

The Internet of Things (IoT) [1] is defined as the set of objects that can communicate over the Internet. Cloud computing is an Internet-based computing service that can provide task allocation [2] and secure data services [3] to IoT devices on demand [4]. This has led IoT to be a global technology used in many fields, such as ubiquitous cities [5]. With the development of these systems, data process technology has attracted many researchers' interest. As an example, to meet the secure requirements of ubiquitous cities, Shen et al. have proposed a sharing framework based on attribute-based cryptography to support dynamic operations for urban data [5]. Radio Frequency Identification (RFID) is considered one of the key technologies to realize IoT and is widely used in many areas, such as mobile payments, public transportation, health monitoring, environment protection, and smart city [6, 7]. More and more big data are generated by those types of RFID and Cloud-based systems, which make the

spatial-temporal data management a current hot topic [8]. RFID uses radio frequency signals through wireless communications to achieve noncontact transmission of information and to perform automatic identification of objects attached with RFID tags. The advantage lies in the fact that the RFID tags and readers can perform identification without physical contact. RFID system can be divided into the following three components: readers, tags, and back-end computer system. The reader and tag communicate through an antenna. To this end, the reader firstly emits electronic signals through an antenna, and then the tag emits identification information of internal storage after receiving the signal. Secondly, the reader receives and identifies the information sent back from the tag via an antenna. Finally, the reader sends the identification results to the computer system [9]. The working principle of RFID system is shown in Figure 1.

In early RFID applications, the reader is directly connected to the application, and RFID data will be processed as logical data by the application. This data processing

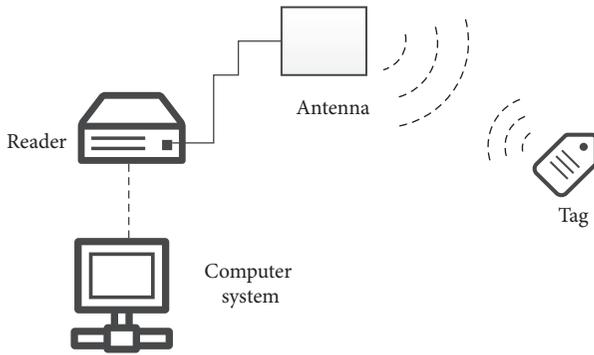


FIGURE 1: Working principle of RFID system.

approach meets the needs of earlier RFID systems, but the system's design is complex and the efficiency is low. There is an important problem with these legacy systems; in particular reusability is difficult. To reduce the complexity and meet the requirements of the rapid development of RFID technology, RFID middleware [10] systems were introduced. Middleware system works as a separate level which can perform some of the data processing and shield the reader hardware and the upper application system. In these scenarios, the program's scalability and applicability are greatly improved. RFID middleware can be used to clean, filter, and format the data collected by the readers and transfer the processed data to the back-end applications [11]. This paper mainly studies the data preprocessing method, which is also referred to as *data cleaning* method in RFID middleware system.

Traditional data cleaning technology [12] is mainly applied in three fields: (1) data warehouse (DW or DWH); (2) Data Discovery or Knowledge Discovery in Data (KDD) [13]; (3) Total Data Quality Management (TDQM) [14]. In these areas, data cleaning is an integral and essential part for data processing. However, traditional data cleaning technology is only suitable for relational database or data warehouse technology. The RFID data flow is different from the traditional relational database and the data stream generated in data warehouse. The interaction between the reader and the tag in the RFID system determines the following characteristics of RFID data [15, 16]: the source data has a simple structure that holds the characteristics of flow, batch, magnanimity, temporal, dynamic change, correlation, and unreliability. The way in which the data is generated by the RFID device determines that the source data is often seriously unreliable [17, 18]. The main reasons for unreliability of data in RFID system are as follows [19]:

- (1) *False positives*: tags' data which should not be recognized by the reader for some reason (noise, electromagnetic interference, and the like) have been read.
- (2) *False negatives*: tags' data that should be recognized by the reader have not been correctly read.
- (3) *Dirty data*: reader detects the tag that exists in its reading range, but the data which is read by the reader contains errors.

False positive and dirty data are fortuity, and the probability of their occurrence is relatively small. Instead, the false negative phenomenon is more common and is the main reason of causing unreliability in RFID system. Therefore, in order to improve the quality of data and ensure that the upper application works effectively, RFID source data cleaning operation is needed. Previous researches aimed at RFID data cleaning technology have done extensive and in-depth study and proposed some classic data cleaning algorithms. This paper proposes a novel data cleaning algorithm (named as *VSMURF*), which is based on the traditional SMURF algorithm in RFID networks, which is aimed at reducing the number of false negatives.

The paper is organized as follows: Section 2 describes the related work and the SMURF algorithm. Section 3 presents the proposed VSMURF algorithm. Section 4 describes simulation and experimental results. Finally, Section 5 summarizes the paper.

2. Related Work and SMURF Algorithm

In this section we describe the related work on data cleaning technology and recap the SMURF algorithm.

2.1. Data Cleaning Technology. Data cleaning [20] refers to a process that can detect and correct the identifiable error in data files. Some classical data cleaning algorithms exist in the literature, which will be introduced in this section.

The data cleaning method based on sliding window is a typical and commonly used method. For example, Bai et al. proposed a method based on a fixed sliding window [21], where the window is fixed and moves forward over time. The cleaning process is shown in Figure 2. Here, the raw data is generated by reader.

Jeffery et al. proposed a scalable data cleaning framework [22] based on Extensible receptor Stream Processing (ESP), which is a declarative query processing tool that is easily pipelined to deploy the configuration to each recipient. The framework is capable of cleaning data from various features of different readers. ESP pipelined structure is processed into the following five consecutive programmable stages: Point, Smooth, Merge, Arbitrate, and Virtualize. Gonzalez et al. proposed a dynamic data cleaning method based on Dynamic Bayesian Networks (DBNs) [23], which dynamically adjusts the trustworthiness state (which is the probability of tag existence). Song of Liaoning University proposed a three-tier structure Kalman Filter-Based RFID Cleaning (KFBC) [24] to address the problem of false negative in RFID system. As shown in Figure 3, this is a Kalman filter update process, which consists of time update and measurement update, and it is an autoregressive process.

2.2. SMURF Algorithm. Here, we firstly recall the basic concepts of the SMURF algorithm [25], and then we describe the algorithm in detail.

Interrogation Cycle. It is an inquiry and answer-response process between a reader and a tag, which is the basic reader's protocol that tries to detect all tags by the reader.

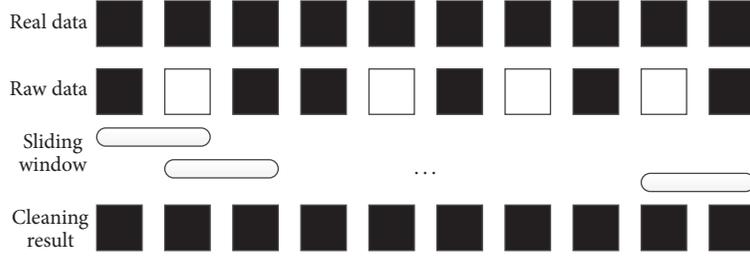


FIGURE 2: Data cleaning method based on fixed sliding window.

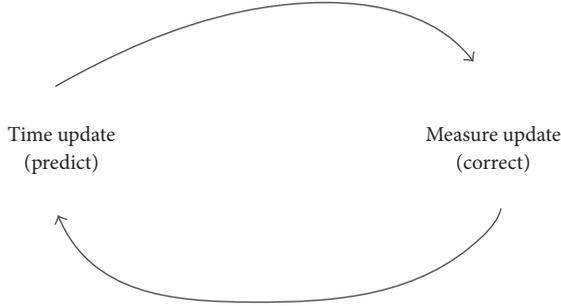


FIGURE 3: The updating process of Kalman filter.

Reading Cycle. It is a set of multiple interrogation cycles. It is also named as epoch, the value of which is from 0.2 seconds to 0.25 seconds [25]. In each epoch, the reader keeps the recording of how many and which tags are identified.

Read Rate $p_{i,t}$. It is the probability of tag i to be read by reader at t epoch, which is calculated as

$$p_{i,t} = \frac{\text{Responses}}{N_t}, \quad (1)$$

where N_t is the number of interrogation cycles at t epoch, and Responses is the number of responses of tag i at t epoch in the reader's reading range.

Sliding Window Size W_i . The value denotes w_i epochs in which tags can be identified by readers. It can be expressed by $W_i = (t - w_i, t]$.

S_i . It denotes the subset of tags observed by the reader during that epoch, which can be expressed by $S_i \subseteq W_i$. In the general case, it is assumed that tag i is seen in a subset of all the epochs in W_i .

The SMUR algorithm works as follows. In each epoch, the reader detects all tags in its reading range and stores the information of tags into the tag list, including *TagID*, *Responses*, and *Timestamp*, as shown in an example in Table 1. The list can be represented by a triple (*TagID*, *Responses*, and *Timestamp*), and all those pieces of information are transmitted to the reader's client at regular intervals.

Each epoch is treated as an independent Bernoulli experiment. In detail, the probabilities of tag i appearing in the

TABLE 1: List of tags.

TagID	Responses	Timestamp
8576 4432 1345 5343	9	11:05:09
8576 9533 2334 2846	1	11:05:09
8576 9864 3453 8642	7	11:05:10

window size (W_i) at each epoch are all the same and can be calculated using the formula:

$$p_i^{\text{avg}} = \left(\frac{1}{|S_i|} \right) \cdot \sum_{t \in S_i} p_{i,t}. \quad (2)$$

In addition, the following two formulas are the most important ones in the SMURF algorithm:

$$W_i \geq \left\lceil \frac{1}{p_i^{\text{avg}}} \ln \left(\frac{1}{\delta} \right) \right\rceil, \quad (3)$$

$$\left| |S_i| - W_i \cdot p_i^{\text{avg}} \right| \geq 2 \cdot \sqrt{W_i \cdot p_i^{\text{avg}} (1 - p_i^{\text{avg}})}, \quad (4)$$

where δ denotes *confidence* in formula (3). Confidence method can be used to process and analyze various data, such as image reconstruction. In the literature [26], authors use the confidence method to remove the noise from image according to the confidence of the depth to compute the range map. Experimental results show that the confidence method obtains good effectiveness in data processing. In SMURF algorithm, confidence is used to meet the integrity requirements of the data and ensure that the tag is successfully read in the sliding window W_i according to formula (3). Formula (4) is obtained by the central limit theorem, which is the control condition for setting the current sliding window size and the conditions of the tag's dynamic change. A reasonable sliding window size needs to balance the integrity requirements with the tag dynamic changes. Determining whether to output data is decided according to the data in the window in SMURF algorithm. In addition, determining how to adjust the window size is done according to the requirements of integrity and dynamic detection. Therefore, SMURF is a type of single tag's data cleaning algorithm.

SMURF algorithm is composed of the following steps.

Step 1. The initial setting of W_i is 1. T is the tag's pool and W_i^* is calculated according to formula (3).

Step 2. If $W_i^* > W_i$, adjust the window size according to the following formula:

$$W_i \leftarrow \max \{ \min \{ W_i + 2, W_i^* \}, 1 \}. \quad (5)$$

Step 3. If $W_i^* \leq W_i$, the dynamic change state of tag is detected according to formula (4). If the tag's state is changed, the window size needs to be adjusted according to the following formula:

$$W_i \leftarrow \max \left\{ \min \left\{ \frac{W_i}{2}, W_i^* \right\}, 1 \right\}. \quad (6)$$

2.3. Limitations of SMURF Algorithm. In the RFID middleware system, the sliding window filtering method is the key technology to reduce false negatives. However, there are some shortcomings in SMURF algorithm: firstly, the size of sliding window is difficult to set in the case of dynamic movement of tags. If the window is set too large, it will generate false positives in the process of filling data; and if the window is set too small, it will not be able to completely fill the data which results in false negative. The selection of the optimal value of sliding window size directly affects the effectiveness of the algorithm. In addition, the effect of this cleaning algorithm is remarkable only when the RFID data stream is in an ideal condition, which means that the RFID tags move at a uniform speed. However, this situation does not happen in some conditions, for example, when the external environmental factors change (such as the tag moves in or out at the reader's detection range rapidly). In these cases, the performance of the cleaning effect is greatly reduced. In addition, SMURF algorithm's confidence is just based on empirical evaluation; that is, it is not the result of a specific analysis. If the tag quickly leaves the reader's detection range, the value δ has a great impact on the results [20].

Specifically, formula (3) shows how the sliding window size (W_i) is related to the two parameters p_i^{avg} and δ . In fact, W_i is inversely proportional to p_i^{avg} , which means that the performance of the SMURF algorithm decreases rapidly when the tag moves out or into the reader's detection range. Assuming a set of raw data streams (0.5, 0.6, 0.6, 0.6, 0.5, 0.8, 0.8, 0.0, 0.4, and 0.8), the initial window size (W_i) is set to 5 epochs, where each epoch contains 10 interrogation cycles, and δ is set to 0.05. Hence, after calculation, p_i^{avg} is 0.56. Because the sliding window size (W_i) is 5 epochs, the processing tag has been moved out from the reader's detection range in the third epoch of the second window. In the end, the tag is still in reader's detection range according to SMURF algorithm, which results in false positives in RFID network. Therefore, SMURF algorithm fails under this condition. For these reasons, we have designed a novel algorithm to take into account the limitations of SMURF, which is described in the next section.

3. VSMURF: A Novel Sliding Window Cleaning Algorithm for RFID Networks

Based upon the previous analysis, we propose a novel algorithm to address SMURF algorithm's limitations, which

is called VSMURF. In the following, we firstly describe the dynamic detection mechanism of tags and analysis of confidence. Then, we detail our adaptive sliding window cleaning algorithm.

3.1. The Dynamic Detection Mechanism of Tags. Assume that the current window size of tag i is $w_i = (t - w_i, t]$, w_i is divided into two parts, where the first part is denoted as $w_{1i} = (t - w_i, t - w_i/2]$ and the second part is denoted as $w_{2i} = [t - w_i/2, t]$, and the binomial distribution samples are, respectively, denoted as $|S_{1i}|$ and $|S_{2i}|$.

The dynamic detection mechanism of a tag's movement is divided into three parts.

Mechanism 1. In order to determine when the tag leaves the reader's detection range, the algorithm introduces the sliding window w_{2i} ($|S_{2i}|$) based on formula (4). The Least Squares Method [27] is used to fit the slope of the curve to determine whether the tag is moving out. If the slope is negative, the tag is moving out from the detection range. In addition, our proposed algorithm reduces two epochs of sliding window size. The purpose of this is to reduce the occurrence of false negatives.

Mechanism 2. The sliding window will be reduced by half if the tag is moving out and cannot be detected in the sliding window w_{2i} ($|S_{2i}| = 0$).

Mechanism 3. The sliding window size is increased if w_i^* , which is calculated according to formula (3), is larger than the current window w_i ($w_i^* > w_i$) and the actual number of readings is greater than the expected number of readings ($|S_i| > w_i \cdot p_i^{\text{avg}}$).

The above three mechanisms are used to adaptively adjust the size of the tag's sliding window. The tag's dynamic detection still uses the central limit theorem (CLT) [28]. In the initial state, all tags which need to be detected are set to 1 epoch and the window size is adjusted to 3 epochs. The goal is to balance the efficiency of the proposal algorithm and reduce false positives. Similar to the SMURF algorithm, the proposed VSMURF algorithm outputs the results at the middle of window and slides once at each epoch.

3.2. Confidence δ Analysis. In the SMURF algorithm, the formula $(1 - p_i^{\text{avg}})^{w_i} < \delta$ is the integrity requirement to ensure that the RFID data stream is completely covered, but it does not cover the particular value of δ that should be selected. The SMURF algorithm only provides an empirical value. When the value of δ is less than 0.5, it has a little effect in the sliding window size [25]. In fact, the SMURF algorithm cleans the data significantly when the tag moves slowly and does not move in and out frequently in the reader's detection range. But when the tag moves quickly, the algorithm's error rate becomes high. The reason is that the sliding window size becomes larger when the confidence δ is smaller, which results in an increase in the number of false positive readings and a corresponding increase in reading error rate.

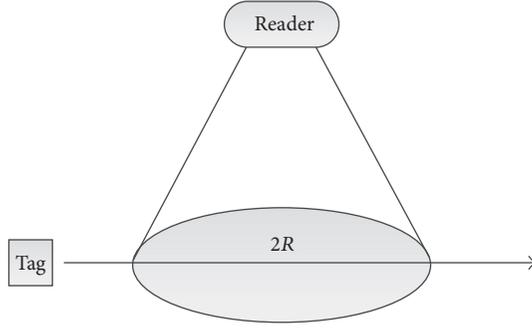


FIGURE 4: Reader detection range.

In order to solve the above problems, this paper takes into account the following factors that affect the cleaning results: the reader's detection range (R), reading frequency (f), the tag's speed (V), and confidence (δ). All these factors dynamically affect the sliding window size's adjustment. The maximum number of epochs is determined by $2R/(V/f) = 2Rf/V$ when the tag passes through the center of the circle (as shown in Figure 4).

When the tag is very close to the antenna of the reader reading range, the number of readings is $F = Rf/V$. To improve the efficiency of data cleaning, VSMURF algorithm dynamically adjusts the parameter δ ($\delta = p_i^{\text{avg}}/F$). Therefore, we obtain the following formula:

$$(1 - p_i^{\text{avg}})^{W_i} < \delta = \frac{p_i^{\text{avg}}}{F} = \frac{p_i^{\text{avg}}V}{Rf}. \quad (7)$$

By applying the natural logarithm on both sides of the equation, we get

$$W_i \ln(1 - p_i^{\text{avg}}) < \ln\left(\frac{p_i^{\text{avg}}V}{Rf}\right). \quad (8)$$

This is because

$$\ln(1 - x) \leq -x, \quad x \in (0, 1). \quad (9)$$

Finally, we obtain

$$W_i \geq \frac{\ln(Rf/p_i^{\text{avg}}V)}{p_i^{\text{avg}}}. \quad (10)$$

Thus, the integrity requirement is adjusted to formula (10). The maximum threshold ν of V can ensure that the error rate is less than 10% [20].

3.3. Adaptive Sliding Window Cleaning Algorithm VSMURF. VSMURF is an improved single tag's sliding window cleaning algorithm based on SMURF algorithm and is based on the adjustment process of tag i at each epoch. The algorithm is composed of seven steps, which are described in detail in the following.

Step 1. Initialize the reader's detection range (R), reading frequency (f), confidence (δ), and threshold of tag's maximum speed and set the initial window to 1 epoch. When $V > \nu$, δ is increased appropriately ($\delta < 0.25$). Otherwise, δ is reduced appropriately.

Step 2. Detect whether reading cycle (epochs) ends, and if it ends, then the algorithm ends.

Step 3. Calculate the minimum value that satisfies the integrity requirement according to formula (3) and determine whether the tag is removed by using least squares.

Step 4. If the tag is moving out and $|S_{2i}| = 0$, the sliding window size is adjusted to the half according to Mechanism 2. Then adjust the window size according to formula (11) and enter the next epoch:

$$W_i \leftarrow \max \left\{ \min \left\{ \frac{W_i}{2}, W_i^* \right\}, 3 \right\}. \quad (11)$$

Step 5. Check whether the window is modified according to formula (4). If the window transforms, adjust the window size according to formula (12) and enter the next epoch:

$$W_i \leftarrow \max \{ \min \{ W_i - 2, W_i^* \}, 3 \}. \quad (12)$$

Step 6. If the current window size does not meet the integrity requirement ($W_i^* > W_i$) and the window does not convert ($|S_i| > W_i \cdot p_i^{\text{avg}}$), according to Mechanism 3 and to formula (13) increase the window size and then enter the next epoch:

$$W_i \leftarrow \min \{ W_i + 2, W_i^* \}. \quad (13)$$

Step 7. If the current window satisfies the integrity requirement ($W_i \geq W_i^*$) and the window transforms ($|S_i| \leq W_i \cdot p_i^{\text{avg}}$), the window size will not change.

Figure 5 shows the data process flow of VSMURF algorithm.

The pseudocode of the VSMURF algorithm is given in Pseudocode 1.

4. Simulation and Experimental Results

4.1. Reader's Detection Model and Evaluation Mechanism. In our experiments, the detection range of the reader is divided into the major detection range and the minor detection range [29] (as shown in Figure 6).

The reader detection model has the following characteristics:

- (1) The reader has a high detection probability and the reading rate is higher than 95% in the major detection range, which is near to the reader.
- (2) The area that extends from the end of the major detection range to the end of the reader's detection range is called minor detection range. Here, the reading rate is linearly reduced to 0.

The model uses the following parameters to capture the behavior of readers under different conditions:

- (i) *Detection range*: the distance between the reader and the boundary of the reader
- (ii) *Percentage of major detection range (MajorPercentage)*: the major detection range of the reader accounting for the percentage of the entire detection range

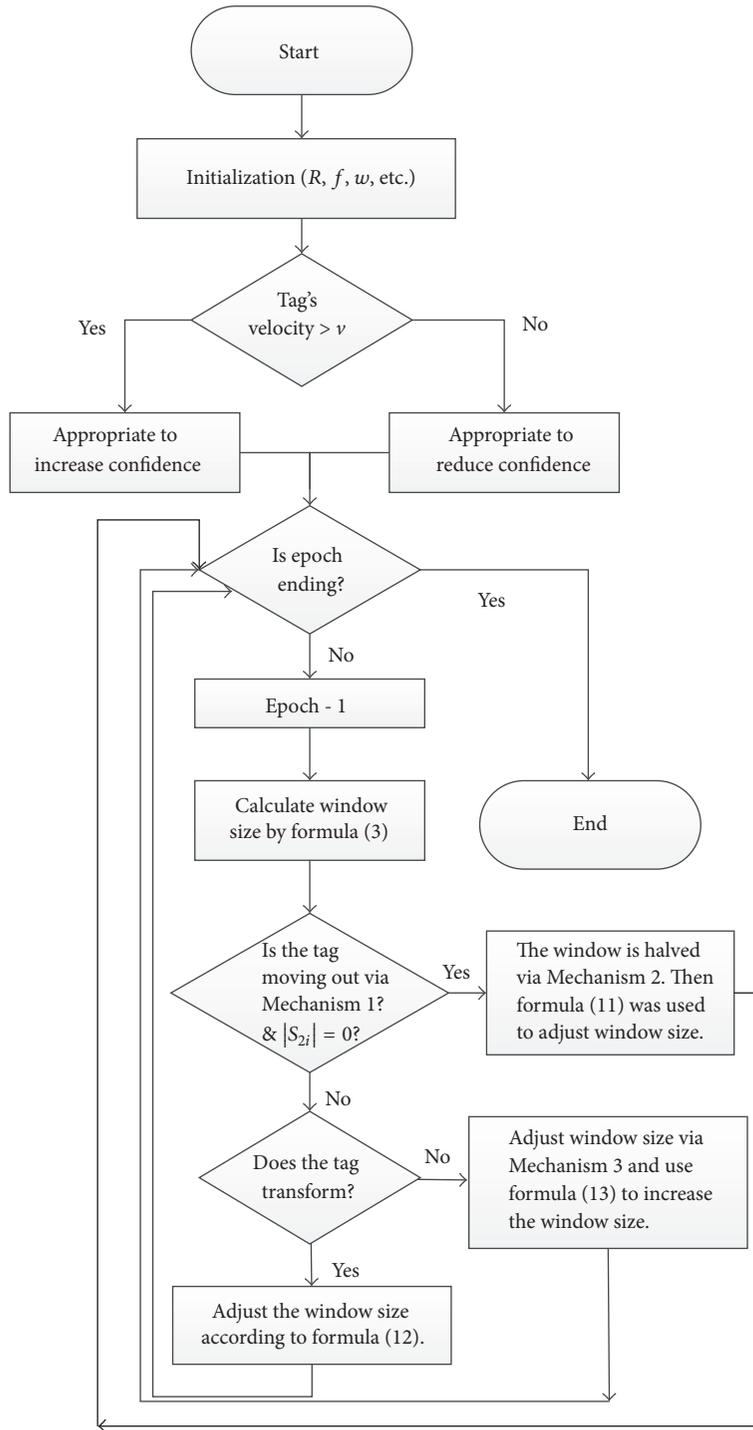


FIGURE 5: Data process flow of VSMURF algorithm.

(iii) Major reading rate (*MajorReadRate*): reading rate of tags in the major detection range

(iv) *Error rate*: error data divided by the total number of data

In addition, evaluation mechanism of RFID data cleaning algorithm mainly uses the average error rate (AvgErrors), the average false positive readings (AvgPositives), and the average

false negative readings (AvgNegatives). They are calculated as follows:

AvgErrors

$$= \frac{\sum_j^{\text{NumEpochs}} (\text{FalsePositives}_j + \text{FalseNegatives}_j)}{\text{NumEpochs}}$$

```

Input:  $T$  = set of all observed tag IDs
 $\delta$  = required completeness confidence
 $v$  = the maximum threshold of tag's speed
 $R$  = the radius of reader's detection range
 $f$  = the frequency of reader's detection
Output:  $t$  = set of all present tag IDs
Initialize:  $\forall i \in T, W_i \leftarrow 1, \delta = 0.1$ 
While ( $getNextEpoch()$ ) do
  For ( $i$  in  $T$ ) do
     $processWindow(W_i) \rightarrow p_{i,t}, p_i^{avg}, |S_i|, V$ 
    if ( $(V > v) \wedge (V < 0.25)$ )
       $\delta+ = 0.05$ 
    Else
       $\delta- = 0.05$ 
    End if
    If ( $tagExist(|S_i|)$ )
      Output  $i$ 
    End if
     $W_i^* \leftarrow completeWindowSize(p_i^{avg}, \delta)$ 
     $tagleaving \leftarrow moveDetection(p_{i,t}, S_i)$ 
    If ( $tagleaving \wedge (|S_{2i}| = 0)$ )
       $W_i \leftarrow \max\{\min\{W_i/2, W_i^*\}, 3\}$ 
    Else if ( $detectTransition(|S_i|, W_i, p_i^{avg})$ )
       $W_i \leftarrow \max\{\min\{W_i - 2, W_i^*\}, 3\}$ 
    Else if ( $(W_i^* > W_i) \wedge (|S_i| > W_i \cdot p_i^{avg})$ )
       $W_i \leftarrow \min\{W_i + 2, W_i^*\}$ 
    Else
       $W_i \leftarrow \min\{W_i, W_i^*\}$ 
    End if
  End for
End while

```

PSEUDOCODE 1: Adaptive sliding window cleaning algorithm.

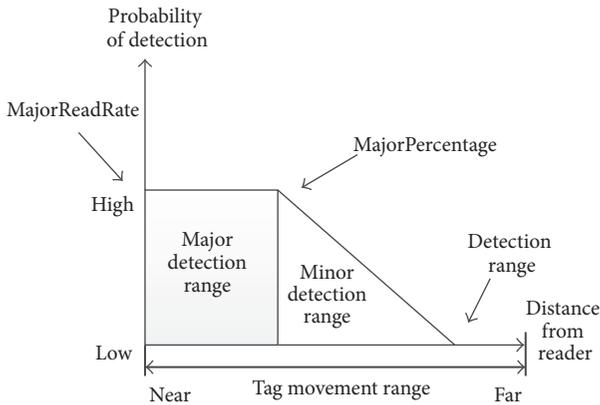


FIGURE 6: Reader detection model.

$$\begin{aligned}
 AvgPositives &= \frac{\sum_j^{NumEpochs} (FalsePositives_j)}{NumEpochs}, \\
 AvgNegatives &= \frac{\sum_j^{NumEpochs} (FalseNegatives_j)}{NumEpochs},
 \end{aligned}
 \tag{14}$$

where NumEpochs is the number of reading cycles; FalsePositives and FalseNegatives are the numbers of false positive readings and false negative readings, respectively. The evaluation mechanism includes two types of errors, which can effectively evaluate the performance of various algorithms.

The reading rate p is defined as in the following formula:

$$p(D) = \begin{cases} 1 - \frac{1 - p_h}{r_2} * D, & D \in (0, r_1), \\ \frac{p_l}{r_1 - r_2} * D - \frac{p_l * r_2}{r_1 - r_2}, & D \in (r_1, r_2), \end{cases} \tag{15}$$

where D is the distance between the tag and the reader, r_1 and r_2 are the reader's major and minor detection range, and p_h and p_l are the reading rates of the major and minor detection range, respectively.

4.2. Settings of Simulation. This section describes the experimental hardware and software environment used in all the simulations.

4.2.1. Hardware Environment. The hardware is composed of the following equipment:

- (i) PC is HP Intel Core 2 Duo notebook. The speed of CPU is 2.33 GHz and the memory is 2 G.
- (ii) RFID Reader is Speedway R420 reader.
- (iii) Tag is EPC G2 UHF RFID tag.

Figure 7 shows the RFID reader, reader's antenna, and tags used in our experiments.

4.2.2. Software Environment. The software environment includes the following:

- (i) Operating system: Windows 7
- (ii) Programming language: Java
- (iii) Programming environment: Java JDK 1.8.0_40; Eclipse IDE for Java Developers Luna Service Release 2 (4.4.2)

VSMURF algorithm belongs to the single tag cleaning algorithm; hence, in the experiments, only one reader is used and the number of tags is set to 25. The experimental model of VSMURF algorithm is shown in Figure 8. At some point, when the tag moves into the reader's detection range, p is calculated on the current coordinate point. The value of p is determined by the distance between the tag and the reader using formula (15). In addition, the system generates a random number n (the value of n between 0 and 1), where n is used to determine whether the tag's information is generated. If $p \geq n$, the tag is read; if $p < n$, then the tag is false negative reading.

4.3. Results and Discussion. In this section we describe the experiments that we have performed.

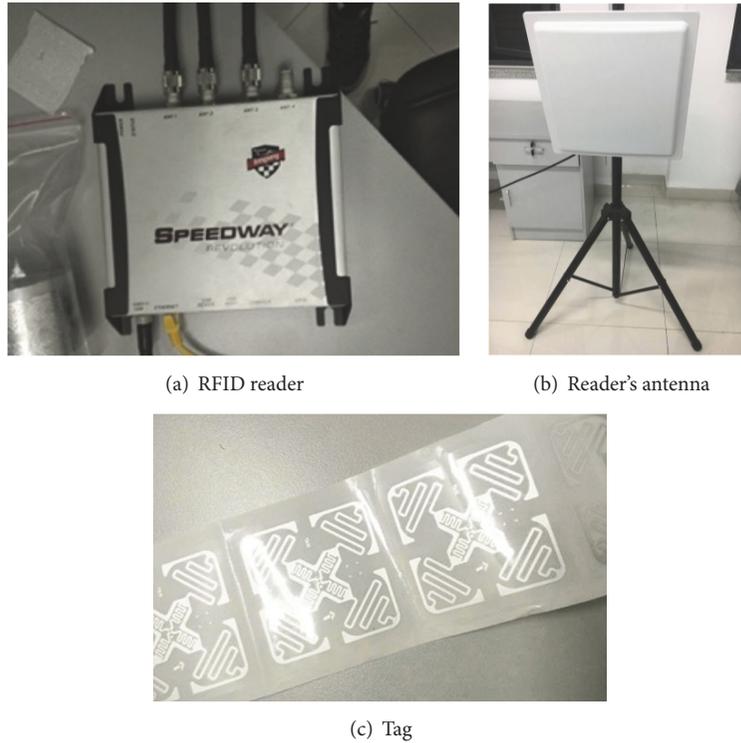


FIGURE 7: RFID reader (a), reader's antenna (b), and tag (c).

TABLE 2: Experiment 1 parameter setting.

Parameters	Value
Detection range	20 m
MajorPercentage	Variable (0, 1)
MajorReadRate	0.8
Number of tags	25
Speed	2 m/epoch
Reading cycles	1000

Experiment 1. 25 tags are selected, where each tag randomly changes its state, the MajorPercentage is changed from 0 to 1, and the reading cycles are set as 1000 epochs. We adopt and compare the following three algorithms: fixed window cleaning algorithm, SMURF cleaning algorithm, and VSMURF algorithm. The average error rate per epoch of above algorithms is compared. The experiment parameters are listed in Table 2.

The results are shown in Figure 9. In detail, when MajorPercentage is 0 (such as in a very noisy environment), the large window performs better, which shows about 4 errors per epoch. As MajorPercentage increases, the reliability of the original data and the accuracy of each algorithm are improved. The small window performs better when MajorPercentage reaches 1. For example, the major detection range covers the entire detection range of the reader. In this experiment, the VSMURF cleaning algorithm has a noticeable efficiency improvement compared with the

TABLE 3: The execution time of SMURF and VSMURF algorithm.

Algorithm	Execution time
SMURF	16.33 s
VSMURF	20.76 s

SMURF algorithm and has the lowest error rate of the whole detection range at each epoch.

In order to simulate realistic conditions in terms of tag and reader behavior, we have implemented the SMURF and VSMURF algorithm using R420 RFID readers. In the experiments, we have collected 12 sets of real data. Due to the number of reading cycle, which is set to 1000, the time of data cycles is equal 1000. In the experiments, each set of data needs to record the execution time. The time result is shown in Table 3. From Table 3 and Figure 9, we can see that when the tag speed is large and the tag is moved out of the reader detection range frequently, the execution time of VSMURF is slightly higher than that of the SMURF algorithm because the VSMURF algorithm adds the confidence judgment. However, the average error rate of VSMURF is lower than that of SMURF algorithm.

In addition, the overhead of VSMURF algorithm with respect to traditional fixed window and SMURF algorithm is minimal because our VSMURF algorithm only utilizes simple mathematical operations.

Experiment 2. 25 tags are selected, and the tag's speed is changed randomly, the MajorPercentage is kept at 0.5,

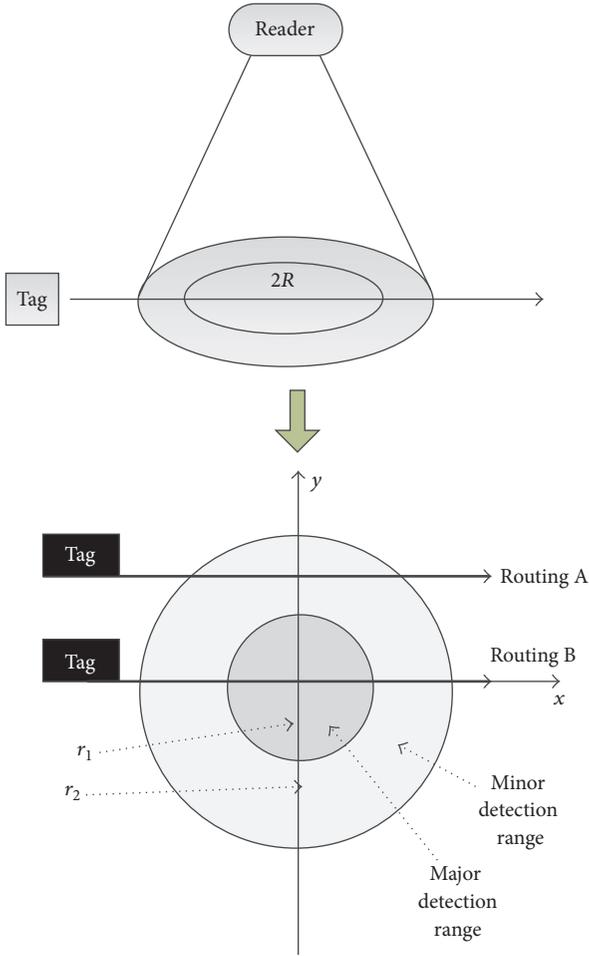


FIGURE 8: The experimental model of VSMURF algorithm.

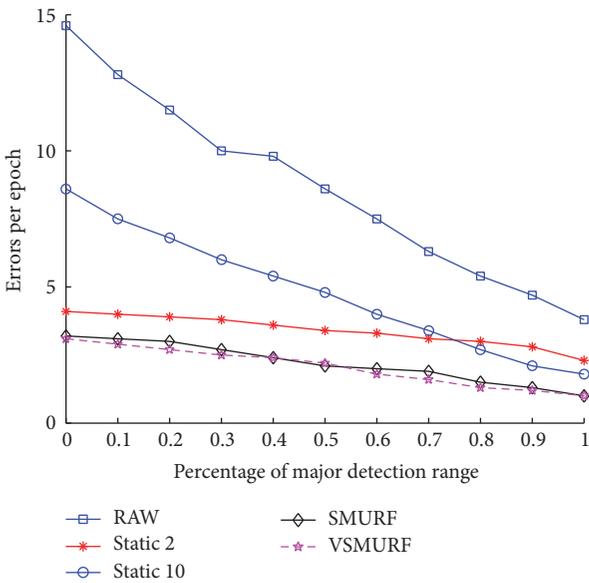


FIGURE 9: Results of average errors per epoch of compared algorithms (MajorPercentage is changed from 0 to 1).

TABLE 4: Experiment 2 parameter settings.

Parameters	Value
Detection range	20 m
Major range	10 m
MajorReadRate	0.8
Number of tags	25
Speed	Variable
Reading cycles	1000

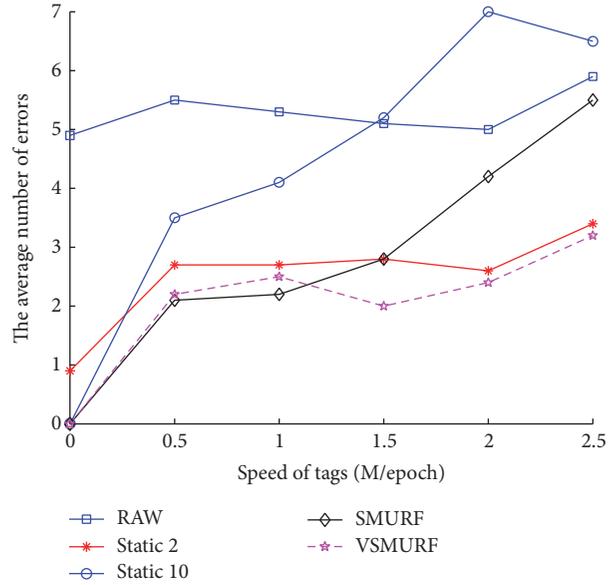


FIGURE 10: Results of average number of errors of compared algorithms when tags' speed changes.

MajorReadRate is 0.8, and the number of reading cycles is 1000 epochs. The fixed window cleaning algorithm, SMURF algorithm, and VSMURF algorithm are compared in this experiment. The average error rate of each algorithm is compared at different speeds. The parameter settings are shown in Table 4.

As shown in Table 5 and Figure 10, all algorithms work well when the tags' speed is less than 0.5, because the tag is in a stable environment at this speed. *Static 10*, which selects the large window to eliminate more false negative readings, obtains the best performance when the tag is static, and the number of errors per epoch is less than 1. With the tag's speed increase, the average number of errors for *static 10* starts to increase and the efficiency begins to decline. This is because the large window causes more false positive readings. In addition, the small window *static 2* cannot completely make up for false negative reading errors. However, it will not cause a lot of false positive readings, and the overall performance is better. SMURF performs well when the tag's speed is small. As the speed increases, that is, the tag moves in or out the detection range frequently, the efficiency of the algorithm is seriously degraded. VSMURF algorithm works better in most conditions and, regardless of the fact that a tag is at

TABLE 5: Comparison of average errors in different speeds.

Speed	Raw errors	Static 2	Static 10	SMURF	VSMURF
0	4.98	0.98	0	0	0
0.5	5.57	2.77	3.5	2.1	2.2
1.0	5.32	2.7	4.1	2.2	2.5
1.5	5.16	2.85	5.2	2.8	2
2.0	5.02	2.6	6.05	4.2	2.4
2.5	5.95	3.4	7	5.5	3.2

low or high speed, the average number of errors is the lowest in all the compared algorithms, as confirmed by the results reported in Table 5 and Figure 10.

5. Conclusion

This paper has presented a new sliding window cleaning algorithm VSMURF, which is based on the tag's dynamic property and confidence. The paper is based on the observation that the SMURF algorithm performs well only when the tag's speed is slow. However, if the tag's speed is increasing and the tag moves into or out the detection range frequently, the efficiency of the SMURF algorithm declines dramatically. To address this limitation, we have introduced VSMURF algorithm and we have shown that it performs better in most conditions and whether the tag's speed is low or high. In particular, if the velocity parameter is set to 2 m/epoch, which represents a typical situation, our proposed VSMURF performs better than SMURF. The same results apply when the tag is moving fast. The experimental results show that the proposed method is better suited than other similar algorithms in reducing the problem of false negatives. As a future work, we intend to implement VSMURF also in RFID middleware system.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work is financially supported by Jiangsu Government Scholarship for Overseas Studies, the National Natural Science Foundation of China (no. 61373017, no. 61572260, no. 61572261, no. 61672296, and no. 61602261), the Natural Science Foundation of Jiangsu Province (no. BK20140886, no. BK20140888), Jiangsu Natural Science Foundation for Excellent Young Scholar (BK20160089), Scientific & Technological Support Project of Jiangsu Province (nos. BE2015702, BE2016185, and BE2016777), China Postdoctoral Science Foundation (no. 2014M551636, no. 2014M561696), Jiangsu Planned Projects for Postdoctoral Research Funds (no. 1302090B, no. 1401005B), Postgraduate Research and Practice Innovation Program of Jiangsu Province (SJLX15_0381, SJLX16_0326, and KYCX17_0798), Project of Jiangsu High Technology Research Key Laboratory for Wireless Sensor

Networks (WSNLBZY201509), and NUPTSF (Grants no. NY214060, no. NY214061).

References

- [1] T. T. Mulani and V. P. Pingle, "Internet of Things," *International Research Journal of Multidisciplinary Studies*, vol. 2, no. 1, pp. 1–4, 2016.
- [2] Y. Kong, M. Zhang, and D. Ye, "A belief propagation-based method for task allocation in open and dynamic cloud environments," *Knowledge-Based Systems*, vol. 115, pp. 123–132, 2017.
- [3] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*.
- [4] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing," *Journal of Network and Computer Applications*, vol. 67, pp. 99–117, 2016.
- [5] J. Shen, D. Liu, J. Shen, Q. Liu, and X. Sun, "A secure cloud-assisted urban data sharing framework for ubiquitous-cities," *Pervasive and Mobile Computing*.
- [6] S. F. Wamba, A. Anand, and L. Carter, "RFID applications, issues, methods and theory: a review of the AIS basket of TOP journals," *Procedia Technology*, vol. 9, pp. 421–430, 2013.
- [7] J. Zhang, G. Tian, A. Marindra, A. Sunny, and A. Zhao, "A review of passive RFID tag antenna-based sensors and systems for structural health monitoring applications," *Sensors*, vol. 17, no. 2, article 265, 2017.
- [8] Q. Liu, W. Cai, J. Shen, Z. Fu, X. Liu, and N. Linge, "A speculative approach to spatial-temporal efficiency with multi-objective optimization in a heterogeneous cloud environment," *Security and Communication Networks*, vol. 9, no. 17, pp. 4002–4012, 2016.
- [9] C. Zhao, C. Wu, J. Chai et al., "Decomposition-based multi-objective firefly algorithm for RFID network planning with uncertainty," *Applied Soft Computing*, vol. 55, pp. 549–564, 2017.
- [10] R. Derakhshan, M. F. Orłowska, and X. Li, "RFID data management: challenges and opportunities," in *Proceedings of the IEEE International Conference on RFID*, pp. 175–182, Grapevine, Tex, USA, March 2007.
- [11] J. Zhu, J. Huang, D. He, Y. Leng, and S. Xiao, "The design of RFID middleware data filtering based on coal mine safety," in *Proceedings of the Second International Conference on Communications, Signal Processing, and Systems (CSPS)*, vol. 246 of *Lecture Notes in Electrical Engineering*, pp. 859–867, Springer, Tianjin, China, 2013.
- [12] B. Gao, *The Research of Algorithm for Uncertain RFID Data Cleansing [Master's, thesis]*, Nanjing University of Information Science and Technology, Nanjing, China, 2015.

- [13] F. Abdul Rahman, M. Desa I, A. Wibowo, and N. Abu Haris, "Antoni wibowo and norhaidah abu haris, data cleaning in knowledge discovery database (KDD)-data mining," in *Proceedings of the International Conference on Advance in Computer Science and Electronics Engineering (CSEE '14)*, pp. 4–10, New Delhi, India, February 2014.
- [14] M. A. Ramadan, A. I. El Seddawy, and A. S. El Rawas, "The impact of total quality management using data mining on the integrated innovation management in smart education," *International Journal of Scientific and Research Publications*, vol. 5, no. 5, pp. 1–6, 2015.
- [15] C. C. Aggarwal and J. Han, "A survey of RFID data processing," in *Managing and Mining Sensor Data*, pp. 349–382, Springer-Verlag, New York, NY, USA, 2013.
- [16] Y. Liu, H. Zhang, L. Zhang, and X. Xu, "Measuring the uncertainty of RFID data streams," *Journal of Computational Information Systems*, vol. 11, no. 6, pp. 2181–2187, 2015.
- [17] Z. Zhao and W. Ng, "A model-based approach for RFID data stream cleansing," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*, pp. 862–871, New York, NY, USA, November 2012.
- [18] Y. Gu, G. Yu, X.-L. Hu, and Y. Wang, "Efficient RFID data cleaning model based on dynamic clusters of monitored objects," *Journal of Software*, vol. 21, no. 4, pp. 632–643, 2010.
- [19] H. Mahdin and J. Abawajy, "An approach to filtering RFID data streams," in *Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks (I-SPAN '09)*, pp. 742–746, Kaohsiung, Taiwan, December 2009.
- [20] Z. Hongsheng, T. Jie, and Z. Zhiyuan, "Limitation of RFID data cleaning method—SMURF," in *Proceedings of the IEEE International Conference on RFID-Technologies and Applications (RFID-TA '13)*, 4, 1 pages, Johor Bahru, Malaysia, September 2013.
- [21] Y. Bai, F. Wang, and P. Liu, "Efficiently filtering RFID data streams," in *Proceedings of the 1st International VLDB Workshop on Clean Databases (CleanDB '11)*, pp. 1–8, Seoul, Korea, September 2006.
- [22] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, "A pipelined framework for online cleaning of sensor data streams," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pp. 1–12, Atlanta Georgia, Ga, USA, April 2006.
- [23] H. Gonzalez, J. Han, and X. Shen, "Cost-conscious cleaning of massive RFID data sets," in *Proceedings of the 23rd International Conference on Data Engineering (ICDE '07)*, pp. 1268–1272, Istanbul, Turkey, April 2007.
- [24] Y. Wang, B.-Y. Song, H. Fu, and X.-G. Li, "Cleaning method of RFID data stream based on kalman filter," *Journal of Chinese Computer Systems*, vol. 32, no. 9, pp. 1794–1799, 2011.
- [25] S. R. Jeffery, M. Garofalakis, and M. J. Franklin, "Adaptive cleaning for RFID data streams," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06)*, pp. 163–174, Seoul, Korea, September 2006.
- [26] Y. Chen, C. Hao, W. Wu, and E. Wu, "Robust dense reconstruction by range merging based on confidence estimation," *Science China Information Sciences*, vol. 59, no. 9, Article ID 092103, 2016.
- [27] A. F. C. Errington, B. L. F. Daku, and A. F. Prugger, "Initial position estimation using RFID tags: a least-squares approach," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 11, pp. 2863–2869, 2010.
- [28] J. Dou, F. Zou, and F. Chen, "Application of the central limit theorem," *Aisa-Pacific Education*, vol. 3, no. 1, pp. 138–139, 2016.
- [29] N. Dhama and M. Sharma, "An improved algorithm for reducing false and duplicate readings in RFID data stream based on an adaptive data cleaning scheme," *International Journal of Computer Trends and Technology*, vol. 4, no. 4, pp. 944–950, 2013.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

