Inter-Data Commonality Detection for Spectrum Monitoring in Wireless Sensor Networks

Zhijuan Hu, Danyang Wang*, Chenxi Li, Tingting Wang

School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
* The corresponding author, email: dywang@xidian.edu.cn

Abstract: Cooperative spectrum monitoring with multiple sensors has been deemed as an efficient mechanism for improving the monitoring accuracy and enlarging the monitoring area in wireless sensor networks. However, there exists redundancy among the spectrum data collected by a sensor node within a data collection period, which may reduce the data uploading efficiency. In this paper, we investigate the inter-data commonality detection which describes how much two data have in common. We define common segment set and divide it into six categories firstly, then a method to measure a common segment set is conducted by extracting commonality between two files. Moreover, the existing algorithms fail in finding a good common segment set, so Common Data Measurement (CDM) algorithm that can identify a good common segment set based on inter-data commonality detection is proposed. Theoretical analysis proves that CDM algorithm achieves a good measurement for the commonality between two strings. In addition, we conduct an synthetic dataset which are produced randomly. Numerical results shows that CDM algorithm can get better performance in measuring commonality between two binary files compared with Greedy-String-Tiling (GST) algorithm and simple greedy algorithm.

Keywords: spectrum monitoring; wireless sensor network; inter-data commonality detection; measurement; CDM algorithm

Received: Jun. 18, 2019
Revised: Aug. 21, 2019
Editor: Zan Li

I. INTRODUCTION

Radio spectrum is limited-sharing natural resources, which is important in wireless communication applications, such as mobile communication, electronic reconnaissance, and electromagnetic spectrum warfare [1-4]. Reasonable and efficient utilization and management of spectrum resources have received great attention in the field of civil and military [5]. Spectrum monitor is the premise foundation of spectrum management and information construction in wireless sensor networks. Since the current demand for spectrum monitoring expands from a single frequency element to frequency, time, space, signal, power and other multi-dimensional elements, the spectrum monitoring has changed from the traditional data collection to data analysis [6,7], signal process [8-10], signal localization and tracking [11-13], etc. With the proliferation of wireless sensor networks, multi-sensor based cooperative spectrum monitoring system [14] has become the inevitable trend and main research direction of spectrum monitoring in the future [15]. Because of the correlation of time, there exists redundancy among the spectrum data collected within a data collection period by a sensor node. In order to obtain valuable spectral information efficiently, the inter-data commonality detection seems especially urgent. A good method of inter-data commonality detection can find more common segments

so as to compress data compactly and improve the data transmission rate effectually.

In the exist literatures, three typical algorithms have been studied to find out the common segments exactly of two data [16-19] in byte level. Greedy-String-Tiling (GST) algorithm [20] is usually used in commonality detection systems to find the matching segments, which consists of two phases [21]. In the first phase, the longest contiguous common segments between two file are searched. In the second phase, all the common segments longer than or equal to the length of segment we want are marked. Then the marked segments are forbidden using in the next iteration. If an unmarked segment is repetitive with part of a marked segment, it will be ignored, so GST algorithm can not produce a good common segment set. Running Karp-Rabin Greedy String Tiling (RKR-GST) [22] is an improvement of GST as it imports a rolling hash function. Although the computational complexity of RKR-GST algorithm has reduced dramatically compared to GST algorithm, both of the two algorithms have the same ability in finding common segments. A simple greedy algorithm of differential compression [23, 24], based on block move model [25-27], can also detect common segments. Assume D_i^t and D_i^{t+1} are the data collected by sensor number i at time t and $t+1$. We aim to find common segments in D_i^{t+1} refer to D_i^t. The simple greedy algorithm first computes footprints of D_i^t and stores in a hash link-table. Then the footprint at offset zero in D_i^{t+1} is computed. The footprint is the hash value of a data segment with fixed length and offset is the relative start position of a data segment in the data. If the footprint is exist in hash link-table, it will find the longest common segment of D_i^t and D_i^{t+1}. Then it computes the footprint at the offset following the matching in D_i^{t+1}, and the process continues. Since the simple greedy algorithm pays more attention to the longest common segments and discards the short ones, it is not good at determining the commonality of two data.

In this paper, we propose an algorithm which modified from simple greedy algorithm to construct a good common segment set from D_i^t and D_i^{t+1}. Firstly, two segments in D_i^t(or D_i^{t+1}) are separated, contained or overlapped according to the offsets. Based on which, two pairs of common segments in a common segment set fall into one of six cases, i.e., separate, partially overlapping, partially containing, fully overlapping, fully containing and overlapping with containment. Secondly, for all pairs of common segments in a common segment set, if only separate holds, the common segment set is fully translative, otherwise, it is weakly translative or hybridly. These are three level at which we measure the commonality between D_i^t and D_i^{t+1}. Let S denote a common segment set, we define $L(S)$ as the sum length of all common segments in D_i^t (or D_i^{t+1}) to measure a common segment set. Because there usually are not only one common segment set, $L(S)$ may have various values. A good common segment set is that whose $L(S)$ get the maximum. Then an algorithm named Common Data Measurement (CDM) algorithm to extract a good common segment set from D_i^t and D_i^{t+1}. The CDM algorithm can be divided into three parts: 1) finding a common segment set that the common segments are separate or weakly overlapped in D_i^{t+1}, 2) extracting all the cascading sequences in the common segment set obtained and computing a best representatives of each cascading sequences and 3) replacing all the cascading sequences with the best representatives. We prove that a quite good commonality between D_i^t and D_i^{t+1} can be achieved by leveraging the CDM algorithm. At last, we conduct an experiment by randomly producing D_i^t and D_i^{t+1}. By comparing the result from the CDM algorithm with that from the GST and the simple greedy algorithm, we get the conclusion that the CDM algorithm can construct a better common segment set based on commonality between two data.

The main contributions in this paper are

In this paper, we have studied the inter-data commonality detection between two data for spectrum monitoring in a sensor network.

summarized as follows:

1) We define common segment set rigorously, and categorize it into 6 cases so that we can understand common segment set better.

2) The CDM algorithm is proposed to identify a good common segment set between two binary data. The measurement for a common segment set is conducted by extracting commonality between two files. In order to produce fewer collisions and be computed incrementally, we use the Karp-Rabin function to create hash link-table. Theoretical analysis reveals that the CDM algorithm performs well in terms of measuring file commonality.

3) At last, the CDM algorithm is implemented by programming. The synthetic dataset for experiment is produced randomly. Numerical results indicate that the CDM algorithm can find a better common segment set than the GST and the simple greedy algorithm.

The rest of this paper is organized as follows. Section II presents some preliminaries of spectrum monitoring in sensor network and the measurement of the inter-data commonality detection algorithm. Section III proposes the outline of the CDM algorithm and then analyses it in theory. Section IV presents parts of the CDM algorithm and gets the result from experiment carrying on synthetic D_i^t and D_i^{t+1}.

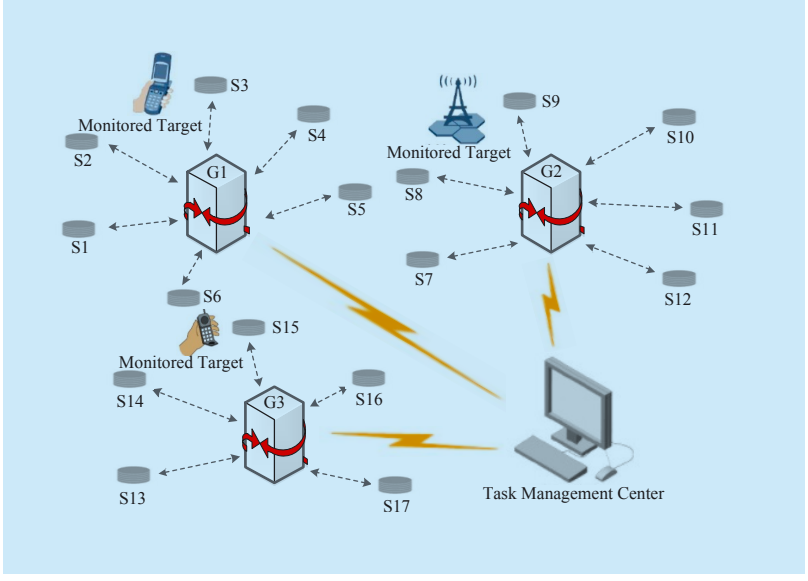


Fig. 1. *System network structure.*

Finally, we draw our conclusions and present some perspective in section V.

II. Preliminaries

Before introduce our inter-data commonality detection method, a system structure for spectrum monitoring in sensor network is described. This structure is used to show an application scene of the inter-data commonality detection, and other conditions like that can introduce our method. Then an evaluation for the inter-data commonality detection algorithm is presented, and based on which we compare our method with the GST and the simple greedy algorithm.

2.1 System structure

A system network structure for electromagnetic spectrum monitoring is exhibited in figure 1.

It consists of three layers two-stage network. The three layers are task management center, gateway nodes and sensor nodes. The first stages network is between sensor nodes and gateway nodes while the second is between gateway nodes and task management center. The sensor node collects spectrum data and transmits them to gateway node while the gateway node receives spectrum data and detects common data and then emits to task management center. At last, the management center realizes the human-machine interaction and displaying. This system network structure accesses to the net flexible using a variety of network interconnection. As a result, it works cooperatively in different network at the same time and is not limited to a particular network paralysis.

The inter-data commonality detection is carried out at the gateway node as figure 2 showing. Usually, the data perceived by the same sensor node in the previous moment and the current have a lot of correlation. That is, let D_i^t and D_i^{t+1} are the data collected by sensor number i at time t and $t+1$, there is many common pattern between D_i^t and D_i^{t+1}.

After a gateway node receiving the D_i^t and D_i^{t+1}, a good inter-data commonality detection algorithm can find more common segments between D_i^t and D_i^{t+1}. This provides convenient conditions for compact data compression. And as a result, the data transmission rate from gateway node to task management center is improved.

Our purpose is to construct a good inter-data commonality detection algorithm which can find a good common segment set including the all pairs of common segment between D_i^t and D_i^{t+1}.

2.2 Definition and category

Before describing the evaluation method, we give several definitions and theorems in assist. Convenient to clarify our idea clearly, the data is represented with binary string and its segment is the substring. And the segment we mention in the following is larger than or equal to a fixed length β. Definition 1 shows three relationship of two segments in a data D_i^t, where k and m are the beginning and end offsets of d_1^t while p and q are that of d_2^t.

Definition 1: Given $d_1^t = D_i^t[k,\cdots,m]$ and $d_2^t = D_i^t[p,\cdots,q]$ are two segments of D_i^t where D_i^t is the data collected by a sensor at time t, there is

1) If $[k,\cdots,m]\bigcap[p,\cdots,q] = \{\}$, d_1^t and d_2^t are separate in D_i^t. We denote this case by $d_1^t \leftrightarrow d_2^t$.

2) If $[k,\cdots,m]$ and $[p,\cdots,q]$ satisfy $k \le p < q \le m$, we say d_1^t contains d_2^t in D_i^t and denote this case by $d_1^t \copyright d_2^t$.

3) If $max(m,q) - min(k,p) + 1 < len(d_1^t) + len(d_2^t)$, we say d_1^t and d_2^t overlap in D_i^t and denote this case by $d_1^t \propto d_2^t$.

Given a data D_i^t and two segments $d_1^t = D_i^t[k,\cdots,m]$ and $d_2^t = D_i^t[p,\cdots,q]$, we get that one and only one of the three cases above can hold. It is easy to proof.

Definition 2: Given two data D_i^t and D_i^{t+1}, and d^t and d^{t+1} are the segments of D_i^t and D_i^{t+1}. If there is a set $S = \{(d^t, d^{t+1}) \mid d^t = d^{t+1}\ for\ each\ (d^t, d^{t+1})\}$, S is a common segment set.

Definition 2 clarifies the common segment set. Usually, there are more than one S between D_i^t and D_i^{t+1}, and each S can present the commonality of two data to certain degree. Hence, a nature question arises: how to find a set S which is the best representative? To answer it, we categorize S with Theorem 1 as follow.

Theorem 1: Given D_i^t, D_i^{t+1}, and S. $\forall(d_1^t, d_1^{t+1})$ and $(d_2^t, d_2^{t+1}) \in S$, one of following cases holds:

1) If $(d_1^t \leftrightarrow d_2^t) \wedge (d_1^{t+1} \leftrightarrow d_2^{t+1}) = 1$, we say (d_1^t, d_1^{t+1}) and (d_2^t, d_2^{t+1}) are separate with respect to S.

2) We say (d_1^t, d_1^{t+1}) and (d_2^t, d_2^{t+1}) are partially overlapping with respect to S if $[(d_1^t \leftrightarrow d_2^t) \wedge (d_1^{t+1} \propto d_2^{t+1}) = 1] \oplus [(d_1^t \propto d_2^t) \wedge (d_1^{t+1} \leftrightarrow d_2^{t+1}) = 1]$.

3) We say (d_1^t, d_1^{t+1}) and (d_2^t, d_2^{t+1}) are partially containing with respect to S if $[(d_1^t \leftrightarrow d_2^t) \wedge (d_1^{t+1} \copyright d_2^{t+1}) = 1] \oplus [(d_1^t \copyright d_2^t) \wedge (d_1^{t+1} \leftrightarrow d_2^{t+1}) = 1]$.

4) If $(d_1^t \propto d_2^t) \wedge (d_1^{t+1} \propto d_2^{t+1}) = 1$, we say (d_1^t, d_1^{t+1}) and (d_2^t, d_2^{t+1}) are fully overlapping with respect to S. When the relative offsets of overlapped segments are the same, it is a triv-

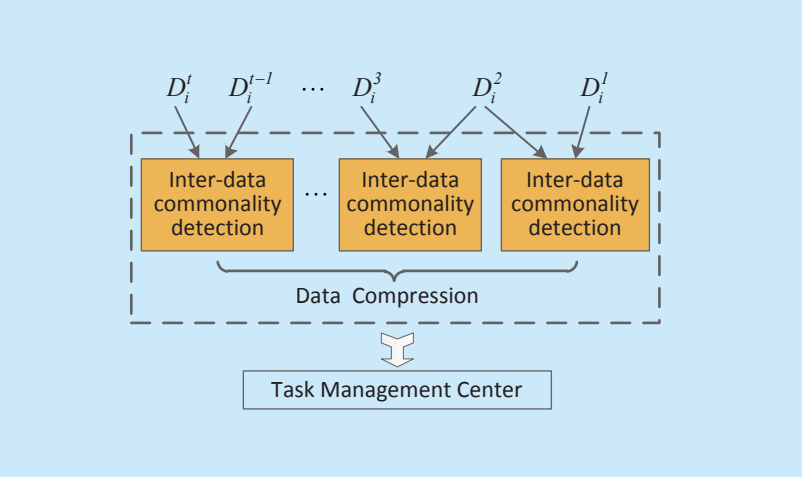


Fig. 2. *Data preprocess at gateway node.*

ial case, otherwise nontrivial. We consider in nontrivial cases only.

5) If $(d_1^t \copyright d_2^t) \wedge (d_1^{t+1} \copyright d_2^{t+1}) = 1$, we say (d_1^t, d_1^{t+1}) and (d_2^t, d_2^{t+1}) are fully containing with respect to S. When the relative offsets of overlapped segments are the same, it is a trivial case, otherwise nontrivial. We consider in nontrivial only.

6) If $[(d_1^t \propto d_2^t) \wedge (d_1^{t+1} \copyright d_2^{t+1}) = 1] \oplus [(d_1^t \copyright d_2^t) \wedge (d_1^{t+1} \propto d_2^{t+1}) = 1]$, We say (d_1^t, d_1^{t+1}) and (d_2^t, d_2^{t+1}) are overlapping with containment respect to S.

2.3 Evaluation method

Now we discussing how to measure a set S based commonality by using the categories. We give Definition 3 and Definition 4 as follow and based on which the evaluation method is described in Definition 5.

Definition 3: Given D_i^t, D_i^{t+1}, and S. If $\forall(d_1^t, d_1^{t+1})$ and $(d_2^t, d_2^{t+1}) \in S$ there is $(d_1^t, d_1^{t+1}) \neq (d_2^t, d_2^{t+1})$, let us describe the cases from the Theorem 1:

1) If only case 1 holds, that is $(d_1^t \leftrightarrow d_2^t)$ and $(d_1^{t+1} \leftrightarrow d_2^{t+1})$, we say that D_i^t and D_i^{t+1} are fractured with respect to S.

2) If one of case 1, 2 and 3 holds and $[(d_1^t \leftrightarrow d_2^t) \vee (d_1^t \propto d_2^t) \vee (d_1^t \copyright d_2^t)] \wedge (d_1^{t+1} \leftrightarrow d_2^{t+1}) = 1$, we say that D_i^{t+1} is fractured with respect to S.

3) If one of case 1, 2 and 3 holds and $(d_1^t \leftrightarrow d_2^t) \wedge [(d_1^{t+1} \propto d_2^{t+1}) \vee (d_1^{t+1} \copyright d_2^{t+1}) \vee (d_1^{t+1} \leftrightarrow d_2^{t+1})] = 1$, we say that D_i^t is fractured with respect to S.

Definition 4: Given D_i^t, D_i^{t+1}, and S. If $\forall(d_1^t, d_1^{t+1})$ and $(d_2^t, d_2^{t+1}) \in S$ there is $(d_1^t, d_1^{t+1}) \neq (d_2^t, d_2^{t+1})$, we have:

1) If only case 1 holds, S is fully translative. Let $\Psi_1(D_i^t, D_i^{t+1}) = \{S \mid S \text{ is a fully translative}\}$.

2) If one of case 1, 2, 3 holds, S is weakly translative. Let $\Psi_2(D_i^t, D_i^{t+1}) = \{S \mid S \text{ is a weakly translative}\}$.

3) If one of case 1, 2, 3, nontrivial case 4, 5 and case 6 holds, S is hybridly translative. Let $\Psi_2(D_i^t, D_i^{t+1}) = \{S \mid S \text{ is a hybridly translative}\}$.

Given D_i^t and D_i^{t+1}, if there are segments d^t and d^{t+1} such that $d^t = d^{t+1}$ and $len(d^{t+1}) \geq \beta$, obviously we have $\Psi_k(D_i^t, D_i^{t+1}) \neq \{\}$ for $k \in \{1,2,3\}$, otherwise there is $\Psi_k(D_i^t, D_i^{t+1}) = \{\}$ for $k \in \{1,2,3\}$.

Definition 5: Given D_i^t, D_i^{t+1}, and S. To measure the set S, we define $L(S)$ as:

$$L(\mathrm{S}) = \sum_{(d^t, d^{t+1}) \in \mathrm{S}} len(d^{t+1}).$$

For $k \in \{1,2,3\}$, we apply $L_k(D_i^t, D_i^{t+1}) = max\{L(\mathrm{S}) \mid S \in \Psi_k(D_i^t, D_i^{t+1})\}$ to measure commonality between D_i^t and D_i^{t+1} at three levels.

If $\Psi_k(D_i^t, D_i^{t+1}) = \{\}$, $L_k(D_i^t, D_i^{t+1}) = 0$. Otherwise, we have $S_k \in \Psi_k(D_i^t, D_i^{t+1})$ such that $L_k(D_i^t, D_i^{t+1}) = L(S_k)$. Also $\forall k \in \{1,2,3\}$, $L_k(D_i^t, D_i^{t+1}) = L_k(D_i^{t+1}, D_i^t)$. These facts are easy to prove.

Theorem 2: Given D_i^t and D_i^{t+1}, we have:

1) $L_1(D_i^t, D_i^{t+1}) \leq L_2(D_i^t, D_i^{t+1}) \leq L_3(D_i^t, D_i^{t+1})$.

2) If either D_i^t or D_i^{t+1} is fractured, we have $L_2(D_i^t, D_i^{t+1}) = L_3(D_i^t, D_i^{t+1})$.

3) If both D_i^t and D_i^{t+1} are fractured, we have $L_1(D_i^t, D_i^{t+1}) = L_2(D_i^t, D_i^{t+1}) = L_3(D_i^t, D_i^{t+1})$.

Proof:

1) Since $\Psi_1(D_i^t, D_i^{t+1}) \subseteq \Psi_2(D_i^t, D_i^{t+1}) \subseteq \Psi_3(D_i^t, D_i^{t+1})$, we get $L_1(D_i^t, D_i^{t+1}) \leq L_2(D_i^t, D_i^{t+1}) \leq L_3(D_i^t, D_i^{t+1})$. This is trivial.

2) If either D_i^t or D_i^{t+1} is fractured, trivial cases of case 4 and 5 do not exist. Hence we have $\Psi_2(D_i^t, D_i^{t+1}) = \Psi_3(D_i^t, D_i^{t+1})$ so that $L_2(D_i^t, D_i^{t+1}) = L_3(D_i^t, D_i^{t+1})$.

3) If both D_i^t and D_i^{t+1} are fractured, only case 1 exists. Then we have that $L_1(D_i^t, D_i^{t+1}) = L_2(D_i^t, D_i^{t+1}) = L_3(D_i^t, D_i^{t+1})$.

We use the concept fractured to describe randomness of a data. The third result of Theorem 2 shows us that if both strings are random, the three level measurements of commonality are actually the same.

III. Inter-Data Commonality Detection Method

In this section, we propose a CDM algorithm to extract a good common segment set, which modified from the simple greedy algorithm based on the block move model proposed by Tichy.

3.1 CDM algorithm

Definition 6: Given D^t is the data collected by a sensor at time t, $U =< d_1,\cdots,d_m >$ is a sequence of segment where d_i is a segment of D^t.

1) If U satisfies two properties: (a) the offset of $d_1,\cdots,d_m$ are strictly increasing; (b) $d_i \propto d_{i+1}$, U is a cascading sequence.

2) Let $Q =< q_1,\cdots,q_k >$ where q_i is a segment of D^t. The offsets are increasing and $q_i \leftrightarrow q_j$ for $i \neq j$. If $\forall i \in \{1,\cdots,k\}$ there is $j \in \{1,\cdots,m\}$ such that t_j contains q_j, we say Q is a representative of U. Let $V(Q) = len(q_1) +\cdots+ len(q_k)$. Additionally, we define $|Q| = k$.

3) Let P be a representative of U. If $V(P)=max\{V(Q)|Q \text{ is a representative of } U\}$, we say P is a good representative of U. Let $G(U)=max\{Q|Q \text{ is a representative of } U\}$.

4) Given $P \in G(U)$, If $|P| = min\{|w| \mid w \in G(U)\}$, we say P is a best representative of U. Let $B(U) = \{P \mid P \text{ is a best representative of } U\}$.

From the definition above, we can easily get that if P is a best representative of U, there must be: (a) Each d_j contains at most one $k \le m$; (b) $k \le m$.

Given a D_i^{t+1}, for any cascading sequence U, how to construct a best representative is an interesting problem. When $m = 2$ or 3, the solution is simple. When $F = GetRep(U)$, it is getting complicated that may be worthy of another paper. In this paper, we assume an algorithm exists at our disposal, and we denote the algorithm as $F = GetRep(U)$. We can assume that $GetRep(< d_1,\cdots,d_m >) = < p_1,\cdots,p_m >$ where p_i is either *NIL* or contained in d_j.

In Table 1, we define some functions and quantities to help describing CDM algorithms.

After input data D_t and D_{t+1}, we can get a set D_i^{t+1} which is a good set by eight steps. The basic outline of CDM algorithm is described as algorithm 1.

3.2 Algorithm analysis

Let us analyse the result of the CDM algorithm.

Definition 7: Given D_i^t and D_i^{t+1}, let $\Psi_h(D_i^t, D_i^{t+1}) = \{S \mid \forall(d_1^t, d_1^{t+1}) \text{ and } (d_2^t, d_2^{t+1}) \in S, [(d_1^t \leftrightarrow d_2^t) \vee (d_1^t \propto d_2^t) \vee (d_1^t \copyright d_2^t)] \wedge (d_1^{t+1} \leftrightarrow d_2^{t+1}) = 1\}$ and $L_h(D_i^t, D_i^{t+1}) = max\{S \in \Psi_h(D_i^t, D_i^{t+1})\}$.

Through observation, there are $\Psi_h(D_i^t, D_i^{t+1}) \cap \Psi_h(D_i^{t+1}, D_i^t) = \Psi_1(D_i^t, D_i^{t+1})$ and $\Psi_h(D_i^t, D_i^{t+1}) \cup \Psi_h(D_i^{t+1}, D_i^t) = \Psi_2(D_i^t, D_i^{t+1})$ so that $\Psi_1(D_i^t, D_i^{t+1}) \subseteq \Psi_h(D_i^t, D_i^{t+1}) \subseteq \Psi_2(D_i^t, D_i^{t+1})$. Therefore, we can get $L_1(D_i^t, D_i^{t+1}) \le L_h(D_i^t, D_i^{t+1}) \le L_2(D_i^t, D_i^{t+1})$.

Lemma 1: Given D_i^t and D_i^{t+1}, we can obtain $S' \neq \{\}$ from CDM algorithm. Based on which, we have $L(S') = L_h(D_i^t, D_i^{t+1})$.

Proof:

By construction of the CDM algorithm, we have $L(S') \in \Psi_h(D_i^t, D_i^{t+1})$. Hence $L(S') \le L_h(D_i^t, D_i^{t+1})$. $\forall W \in \Psi_h(D_i^t, D_i^{t+1})$, we

Table I. *Quantities and description.*

I.D.	Description
los	Length of segment over which a hash value is calculated.
p	Offset in D_t when getting a match.
q	Offset in D_i^t when getting the longest match.
k	Current offset in D_i^{t+1}.
L	Length of a match.
M	Length of the longest match.
SS	Set which is hybridly translative.
S_T	Set of substrings merged from cascading sequences.
S_2	Set from the best representatives of S_T.
S_1	Set divided from SS, with respect to which D_i^{t+1} is fractured.
S'	Common segment set merged from S_1 and S_2.
HT	HashLinkList of D_i^{t+1}.

Algorithm 1. CDM algorithm.

Input: D_i^t and D_i^{t+1}

Output: S'

1. Set $SS = \{\}$ and $S' = \{\}$
2. Use a rolling hash function $H(x)$ to slide a *los* width window along the data D_i^t for generating $m+1-los$ hash values. We store them in a hash table *HT* with chaining of linked lists to resolve collisions. The nodes in the linked-list of the hash table *HT* save the offsets where hash values are created.
3. To start with the first offset of D_i^{t+1}, let $k = 1$.
4. If $k > n+1-los$, go to STEP 6.
5. If $H\left(T\left[k,\cdots,k+los-1\right]\right)$ does not match any value h in *HT*, let $k = k+1$, go to STEP 4. Otherwise, we get that $H\left(T\left[k,\cdots,k+los-1\right]\right)$ matches a value h in *HT*. For each offset h in the linked-list, we compare $D_i^t[p,\cdots, p+los-1]$ with $D_i^{t+1}[k,\cdots,k+los-1]$. If they are identical, we extend both segment as long as possible to get the longest match ending at length L, i.e., $D_i^t[p,\cdots, p+L-1] = D_i^{t+1}\left[k,\cdots,k+L-1\right]$. We may have multiple nodes in the linked list, do this for each one and get the longest match from all, say $D_i^t[p,\cdots,p+M-1] = D_i^{t+1}[p,\cdots,p+M-1]$. Add $(D_i^t[p,\cdots,p+M-1], D_i^{t+1}[p,\cdots, p+M-1])$ to *SS*. Let $k = k+M+1-los$, go to step 4.
6. Let $SS = \{(d_1^t, d_1^{t+1}),\cdots,(d_m^t, d_m^{t+1})\}$. Merge all weakly overlapped consecutive segment of $\{d_1^{t+1},\cdots,d_m^{t+1}\}$ into one segment. For example, if $< d_j^{t+1},\cdots,d_k^{t+1} >$ is a cascading sequence, we merge them into one segment g and add g into S_T. And put the rest of *SS* into S_1.
7. For each $t \in S_T$, we assume that t was merged from $< d_j^{t+1}, \cdots, d_k^{t+1} >$ of $SS = \{(d_1^t, d_1^{t+1}),\cdots,(d_m^t, d_m^{t+1})\}$. By using $F = GetRep(t)$, we have a best representative $< z_j^{t+1},\cdots, z_k^{t+1} >$ of $< d_j^{t+1},\cdots,d_k^{t+1} >$ where z_p may be NIL without loss of generality. For each $z_p \neq NIL$ as a segment of d_p^{t+1}, we get its associated segment y_p as a segment of d_p^t where $(y_p^t, z_p^{t+1}) \in SS$. Now, for $p \in \{j,\cdots,k\}$, if $z_p \neq NIL$, we add $< y_p^t, z_p^{t+1} >$ into S_2.
8. S_2 plus S_1 is S'.

need to prove $L(W) \leq L(S')$. Let $W' = \{< y_1^t, z_1^{t+1} >,\cdots,< y_k^t, z_k^{t+1} >\}$ and $Z = \{z_1^{t+1},\cdots,z_k^{t+1}\}$, we need to prove $L(W') \leq L(S_2)$. From step 6, we have $S_T = \{g_1^{t+1},\cdots,g_m^{t+1}\}$ where each $g^{t+1} \in S_T$ is formed by merging a cascading sequence U into one segment of D_{t+1}. $\forall z_i^{t+1}$ and $z_j^{t+1} \in Z$, it exists $z_i^{t+1} \leftrightarrow z_j^{t+1}$ since $\forall W' \in \Psi_h(D_i^t, D_j^t)$. $\forall z_i^{t+1} \in Z$, we have three cases which exclude each other:

1) $\forall g^{t+1} \in S_T$, we have $z_i^{t+1} \leftrightarrow g^{t+1}$;

2) There exists $g^{t+1} \in S_T$ such that $z_i^{t+1} \propto g^{t+1}$;

3) There exists $g^{t+1} \in S_T$ such that $z_i^{t+1} \copyright g^{t+1}$.

The first case can be excluded, otherwise the algorithm should include a $d^{t+1} \in S_T$ which contains z_i^{t+1}, it is a contradiction. With proof by construction, we can also exclude the second case by the construction of S_T. Hence there exist $g^{t+1} \in S_T$ such that $z_i^{t+1} \copyright g^{t+1}$. If $z_i^{t+1} = g^{t+1}$, g^{t+1} contains z_i^{t+1}. When $z_i^{t+1} \neq g^{t+1}$, z_i^{t+1} can not contain g^{t+1}, by the construction of S_T again. Hence,g^{t+1} contains z_i^{t+1}.Therefore, we can conclude that $\forall z_i^{t+1} \in Z$, there is $g^{t+1} \in S_T$ such that g^{t+1} contain z_i^{t+1}. For each $g^{t+1} \in S_T$, let $< z_i^{t+1},\cdots,z_l^{t+1} >$ be the whole subset of Z that g^{t+1} contains and $c =< d_g^{t+1},\cdots,d_k^{t+1} >$ be the cascading sequence that merges into g^{t+1} at step 6. Then $< z_i^{t+1},\cdots,z_l^{t+1} >$ is a representative of c. If P is the representative of c, we have $V(Q) \leq V(P)$. For each $g_i^{t+1} \in S_T$, we denote the best representative as P_i. Hence $L(W') = len(z_1)+\cdots+len(z_k) \leq V(P_1)+\cdots+V(P_m) = L(S_2)$. Because all the pairs of common segments in S_1 are separated to each other and separated to the common segments in S_2, $L(S_1)$ has nothing to do with $L(W')$. Hence, we have $L(W) = L(W') + L(S_1) \leq L(S_2) + L(S_1) = L(S')$. Now we can conclude $L(S') = L_h(D_i^t, D_i^{t+1})$.

The set *SS* obtained in step 5 is not a good common segment set because some longest common segments in D_i^{t+1} are weakly overlapped, so a best representatives of each cascading sequence are needed. Algorithm $F = GetRep(t)$ finds the best representatives and put them into S_2. Combine with the separate common segments in S_1, we get a good common segment set S'.

The following theorem assures that we can achieve quite "good" measurement for the commonality between two strings with the *S* obtained from the CDM algorithm.

Theorem 3: Given D_i^t and D_i^{t+1}, we ob-

tain S' from CDM algorithm. If $S'=\{\}$, $L_1(D_i^t, D_i^{t+1}) = L_h(D_i^t, D_i^{t+1}) = L_2(D_i^t, D_i^{t+1}) = 0$. If $S' \neq \{\}$, we conclude:

1) S' is a weakly translative S;

2) $L_1(D_i^t, D_i^{t+1}) \le L(S') \le L_2(D_i^t, D_i^{t+1})$;

3) If D_i^{t+1} is fractured, $L_2(D_i^t, D_i^{t+1}) = L(S')$;

4) If D_i^t is fractured, $L_1(D_i^t, D_i^{t+1}) = L(S')$;

5) If D_i^t and D_i^t are fractured, $L_1(D_i^t, D_i^{t+1}) = L(S')$.

Proof:

1) According to Definition 6, it is trivial to prove that is a weakly translative cross-sharing.

2) Since $L_1(D_i^t, D_i^{t+1}) \le L_h(D_i^t, D_i^{t+1}) \le L_2(D_i^t, D_i^{t+1})$ and $L(S') = L_h(D_i^t, D_i^{t+1})$, we conclude $L_1(D_i^t, D_i^{t+1}) \le L(S') \le L_2(D_i^t, D_i^{t+1})$.

3) If D_i^{t+1} is fractured, $\Psi_h(D_i^t, D_i^{t+1}) = \Psi_2(D_i^t, D_i^{t+1})$ so that we have $L_h(D_i^t, D_i^{t+1}) = L_2(D_i^t, D_i^{t+1})$, we conclude $L(S') = L_2(D_i^t, D_i^{t+1})$.

4) If D_i^t is fractured, $L_h(D_i^t, D_i^{t+1}) = L_1(D_i^t, D_i^{t+1})$ so that we have $L_h(D_i^t, D_i^{t+1}) = L_1(D_i^t, D_i^{t+1})$, we conclude $L(S') = L_1(D_i^t, D_i^{t+1})$.

5) If D_i^{t+1} and D_i^t are fractured, $L(S') = L_1(D_i^t, D_i^{t+1}) L_2(D_i^t, D_i^{t+1}) = L_3(D_i^t, D_i^{t+1})$ by 3), 4) and Theorem 2.

When D_i^{t+1} is fractured, $L_h(D_i^t, D_i^{t+1})$ is the only case existing in $L_2(D_i^t, D_i^{t+1})$. Meanwhile, if D_i^t is fractured, $L_1(D_i^t, D_i^{t+1})$ is the only case existing in $L_h(D_i^t, D_i^{t+1})$. So the result of CDM algorithm provides a good measurement of commonality.

IV. Algorithm Implement and Experiment

4.1 Algorithm implement

Our CDM algorithm is implemented in Python 2.7.5. It contains four functions.

First of all, we can use the Karp-Rabin hash function or other efficient rolling hash functions. Karp-Rabin hash algorithm is not the fastest, but it produced a very uniform distribution with fewer collisions. An alternative is the "rolling" version Rsync is also known as a tool of remote differential compression.

Secondly, csALL function, including step 4 and 5, designed to find a set S respect to which SS is fractured, and returns set SS is used to store the pairs of common segment we discovered.

The third function is in step 6, aiming to distinguish all the cascading sequence from the separate common segments and merge it into a segment.

The last function $F = GetRep(t)$ is used to pick out a best representative of $< d_j^{t+1}, \cdots, d_k^{t+1} >$. It is accomplished in two cases: $m = 2$ and $m = 3$. To save the run time, we select a best representative directly by analysis instead of exhaustive method. We use $d_i^{t+1}[0]$ and $d_i^{t+1}[1]$ to store the offset of the beginning and the end of d_i^{t+1} in D^{t+1} when

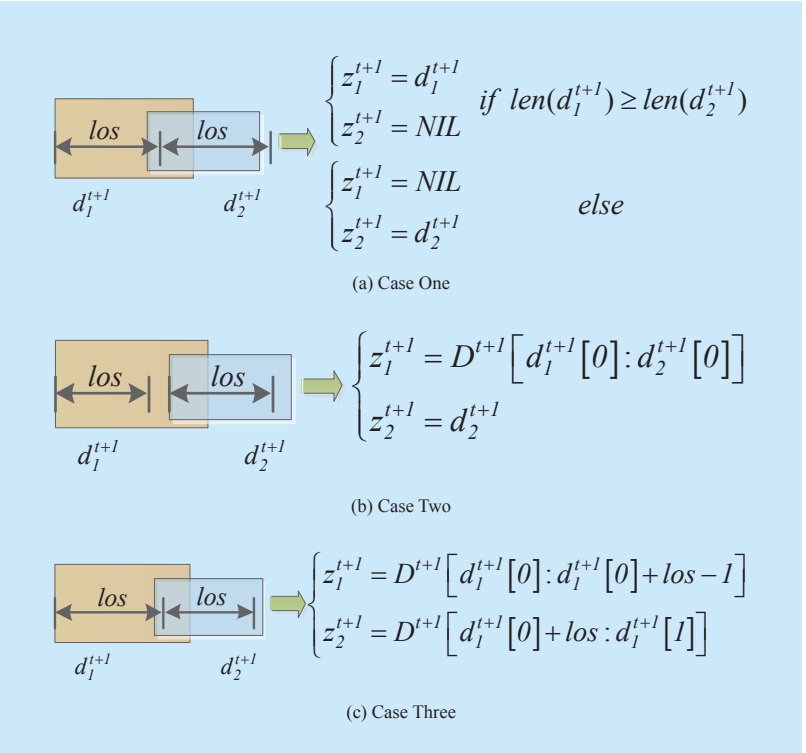


Fig. 3. *A best representative when m=2 (a)* $(d_2^{t+1}[1] - d_1^{t+1}[0] + 1) \le 2los$*; (b)* $d_1^{t+1}[0] + los - 1 < d_2^{t+1}[0]$*; (c)* $(d_2^{t+1}[0] \le d_1^{t+1}[0] + los - 1)$ *and* $(2los < d_2^{t+1}[1] - d_1^{t+1}[0] + 1)$.

$1 \le i \le k$, and $< z_j^{t+1}, \cdots, z_k^{t+1} >$ is a best representative where z_i^{t+1} may be $m = 2$ without loss of generality. When $m = 2$, a best representative can be picked out easily since it is possible that d_1^{t+1} and d_2^{t+1} are weakly overlapped. Fig. 3 illustrates the three cases when $m = 2$.

The case $m = 3$ is more complex than $m = 2$. we should consider two exclusive case as Fig. 4 show.

When d_1^{t+1} and d_3^{t+1} are weakly overlapped, it has:

1) If $d_1^{t+1}[0] + 2los - 1 > d_3^{t+1}[1]$, we compare the length of d_1^{t+1}, d_2^{t+1} and d_3^{t+1}. If d_1^{t+1} is the longest, z_1^{t+1} is equal to d_1^{t+1} while d_2^{t+1} and d_3^{t+1} are both NIL. Anyway the versa.

2) If $d_1^{t+1}[0] + 2los - 1 \le d_3^{t+1}[1]$, z_2^{t+1} is NIL, meanwhile d_3^{t+1} and d_3^{t+1} construct a cascade sequence. In this case, we refer to case (b) and (c) in figure 3.

When d_1^{t+1} and d_3^{t+1} are separated, the method to get a best representative is the same with the case $m = 2$, and we do not give more detail here. If $m \ge 3$, it is complicated and may be worthy of another paper. In this paper, we assume an algorithm exists at our disposal and the output in case of $m \ge 3$ is same as simple greedy algorithm.

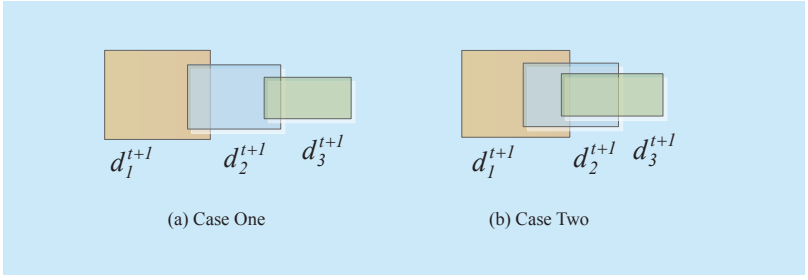


Fig. 4. *Two exclusive cases when m=3 (a) d_1^{t+1} and d_3^{t+1} are separated; (b) d_1^{t+1} and d_3^{t+1} are overlapped.*

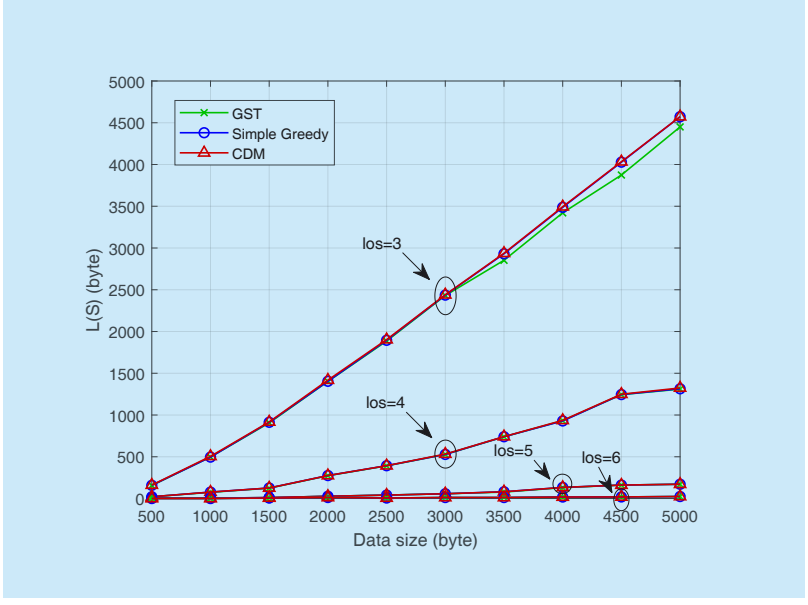


Fig. 5. *Performance of algorithms when there is no correlation between two data.*

4.2 Results and analysis

In this section, we compare our CDM algorithm with GST and simple greedy algorithm in python 2.7.5. The experiment runs on a machine with 3.30GHz Intel Core i5-4590 CPU, 4GB main memory, and a Windows operating system.

We construct the experiment using the experimental data sets. The alphabet consists of 10 Arabic numbers. First, the D_i^t and D_i^{t+1} are produced in random and their size is from 500 to 5000 bytes. The program is ran 1000 times and the average value is computed. Figure 5 shows the result when *los* is between 3 and 6 if there is no correlation between D_i^t and D_i^{t+1}. We can get that the performance of the CDM algorithms is about the same as the simple greedy, and slightly better than the GST when *los* is 3.

Figure 6 shows the result when two data are correlative. The D_i^{t+1} is generated randomly while D_i^t is constructed by two operations. One is to copy segments from D_i^{t+1} and another is to add segments directly. The size of D_i^t and D_i^{t+1} are basically same. The alphabet in case (a) consists of 10 Arabic Numbers and 5 English characters while that in case (b) only uses10 Arabic Numbers, which makes the D_i^t and D_i^{t+1} in case (b) more correlative than that in case (a). The program is also ran 1000 times and the average value is computed. In Fig. 6 (a), the CDM algorithm works well when *los* is 4. In this case, the CDM finds 4.1% common segments in D_i^{t+1} more than GST and 15.9% more than simple greedy algorithm.

Fig. 6 (b) gives the result when the D_i^t and D_i^{t+1} are higher correlative. When *los* is 5, the CDM algorithm performs significantly better than the GST and the simple greedy algorithm. In this case, the CDM finds 4.0% common segments in D_i^{t+1} more than GST and 9.9% more than simple greedy algorithm.

Since the best representative in case of $m \geq 3$ is same as simple greedy algorithm, the CDM algorithm does not behave well enough. However, we can get that the CDM algorithm find a better S than simple greedy algorithm as well as GST as long as the *los* is chosen properly.

V. Conclusion

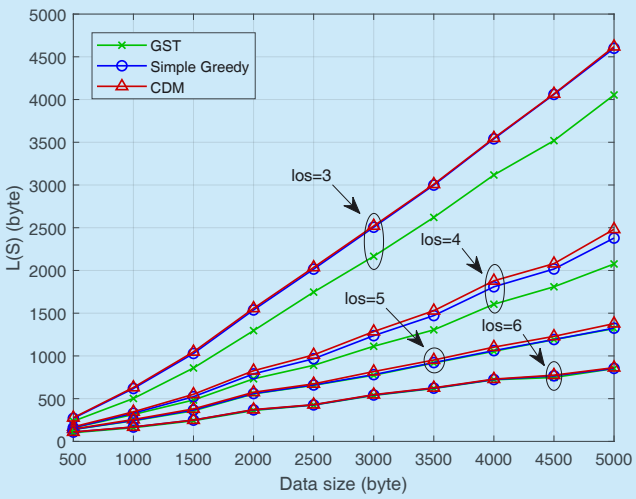
In this paper, we have detailed studied the inter-data commonality detection between two data for spectrum monitoring in sensor network. After the system network structure of the multi-point cooperative electromagnetic spectrum monitoring is exhibited, the definition, category and measurement of cross-sharing have been provided, based on which, an approach to measure the commonality of two strings has been introduced. Then we propose CDM algorithm to find a good common segment set which could represent how much two data have in common. Theoretical analysis has revealed that the common segment set that CDM algorithm had picked out is a good representative for measuring commonality between two data. Additionally, we have programmed to realize CDM algorithm on test data. Numerical results have shown that CDM algorithm performed better in finding a good common segment set than GST algorithm as well as simple greedy algorithm. For further study, given D_i^t and D_i^{t+1}, we will research on both theoretic analysis and algorithmic solutions for a few more questions:

1) If neither D_i^t nor D_i^{t+1} is not repetitive, when do we have $L_1(D_i^t, D_i^{t+1}) \leq L_h(D_i^t, D_i^{t+1}) \leq L_2(D_i^t, D_i^{t+1})$?

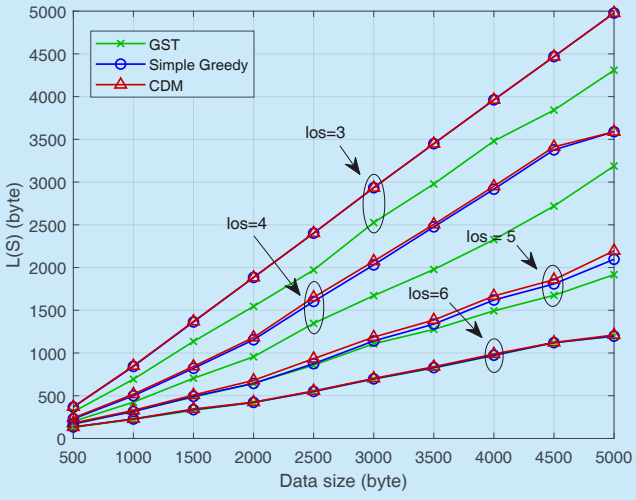
2) How to calculate $L_1(D_i^t, D_i^{t+1})$ and $L_2(D_i^t, D_i^{t+1})$?

3) How to estimate $L_3(D_i^t, D_i^{t+1})$?

4) Given D_i^{t+1}, for any cascading sequence U of D_i^{t+1}, how to construct a best representative of U?



(a) The Alphabet Consists of 10 Arabic Numbers and 5 English Characters



(b) The Alphabet Consists of 10 Arabic Numbers

Fig. 6. *Performance of algorithms when two data are correlative.*

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China

(No.61901328), the China Postdoctoral Science Foundation (No. 2019M653558), the Fundamental Research Funds for the Central Universities (No. CJT150101) and the Key project of National Natural Science Foundation of China (No. 61631015).

References

[1] S. W. Boyd, J. M. Frye, M. B. Pursley, and T. C. Royster IV, "Receiver statistics for spectrum monitoring while communicating,"*Proc. IEEE Global Telecommunications Conferenc*e, 2009, pp. 1-6.

[2] I. F. Akyildiz, W. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks,"*IEEE Communications Magazine*, vol. 46, no. 4, 2008, pp. 40-48.

[3] A. Kliks et al., "Spectrum Management Application for Virtualized Wireless Vehicular Networks: A Step Toward Programmable Spectrum Management in Future Wireless Networks," *IEEE Vehicular Technology Magazine*, vol. 13, no. 4, pp. 94-105, Dec. 2018.

[4] Y. Yang, Q. Zhang, Y. Wang, T. Emoto, M. Akutagawa and S. Konaka, "Multi-strategy dynamic spectrum access in cognitive radio networks: Modeling, analysis and optimization," *China Communications*, vol. 16, no. 3, pp. 103-121, March 2019.

[5] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohantly, "Next generation /dynamic spectrum access/cognitive radio wireless network: a survey,"*Computer Networks*, vol. 50, no. 9, 2006, pp. 2127-2159,.

[6] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: a large-scale spectrum measurement study,"*IEEE Transactions on Mobile Computing*, vol. 11, no. 6, 2012, pp. 1033-1046.

[7] M. Islam, C. Koh, S. W. Oh, X. Qing, Y. Lai, and C. Wang, "Spectrum survey in Singapore: occupancy measurements and analyses," *Proc. 2008 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, 2008, pp. 1-7.

[8] Y. Pei, Y. C. Liang, K. C. Teh, and K. H. Li, "How much time is needed for qideband spectrum sensing?,"*IEEE Transactions on Wireless Communications*, vol. 8, no. 11, 2009, pp. 5466-5471.

[9] F. Zhou, N. C. Beaulieu, Z. Li, and J. Si, "Feasibility of maximum eigenvalue cooperative spectrum sensing based on Cholesky factorisation,"*IET Communications*, vol. 10, no. 2, 2016, pp. 199-206.

[10] P. H. Qi, Z. Li, J. B. Si, and T. Y. Xiong, "A Two-Stage Spectrum Sensing Scheme Based on Energy detection and a Novel Multitaper Method,"*Chinese Physics B*, vol. 24, no. 4, 2015.

[11] N. Patwari, A. O. Hero, and M. Perkins, "Relative location estimation in wireless sensor networks,"*IEEE Transactions on Signal Processing*, vol. 51, no. 8, 2003, pp. 2137-2148.

[12] M. Sun and K. C. Ho, "An asymptotically efficient estimator for TDOA and FDOA positioning of multiple disjoint sources in the presence of sensor location uncertainties,"*IEEE Transactions on Signal Processing*, vol. 59, no. 7, 2011, pp. 3434-3440.

[13] B. J. Hao, Z. Li, J. B. Si, and L. Guan, "Joint source localisation and sensor refinement using time differences of arrival and frequency differences of arrival,"*IET Signal Processing*, vol. 8, no. 6, 2014, pp. 588-600.

[14] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks,"*IEEE Communications Magazine*, vol. 40, no. 8, 2002, pp. 102-114.

[15] Weilong Hu, Zan Li, and Linlin Liang, "Wide Area and Wide Band Spectrum Monitoring System Based on Sensor Networks,"*Proc. 2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2016, pp. 1-5.

[16] X. Benavent, A. Garcia-Serrano, R. Granados, J. Benavent, and E. de Ves, "Multimedia information retrieval based on late semantic fusion approaches: Experiments on a wikipedia image collection,"*IEEE Transactions on Multimedia*, vol. 15, no. 8, 2013, pp. 2009-2021.

[17] T. Li, and M. Ogihara, "Toward intelligent music information retrieval,"*IEEE Transactions on Multimedia*, vol. 8, no. 3, 2006, pp. 564-574.

[18] C. Zhang, and T. Chen, "An active learning framework for content-based information retrieval,"*IEEE Transactions on Multimedia*, vol. 4, no. 2, 2002, pp. 260-268.

[19] M. Raffinot, "Flexible Pattern Matching in Strings" *Cambridge University Press Cambridge*, 2002.

[20] L. Prechelt, G. Malpohl, and M. Phlippsen, "Finding plagiarisms among a set of programs with JPlag," *Journal of Universal Computer Science*, vol. 8, no. 11, 2002, pp. 1016-1038.

[21] M J. Wise, "Running Karp-Rabin Matching and Greedy String Tiling,"*Basser Department of Computer Science Technical Report, Sydney University*, 1993.

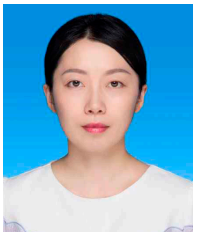
[22] M. J. Wise, "String similarity via greedy string tiling and running Karp-Rabin matching,"*Basser Department of Computer Science Technical Report, Sydney University*, 1993.

[23] M. Ajtai, R. Burns, R. Fagin, D. D. E. Long, and L. Stockmeyer, "Compactly Encoding Unstructured Inpouts With Differential Compression,"*Journal of the ACM*, vol. 49, no. 3, 2002, pp. 318-367.

[24] W.F. Tichy The string-to-string correction problem with block move, "*ACM Transactions on Computer Systems*, vol. 2, no. 4, 1984, pp.309-321.

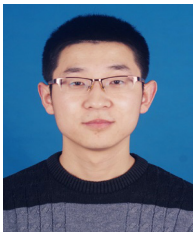
[25] D. E. Knuth, J. H. Morris, JR. and V. R. Pratt, "Fast Pattern Matching in String,"*SIAM Journal of Computing*, vol. 6, no. 2, 1977, pp.323-350.
[26] W. Obst, "Delta Technique and String-to-String Correction,"*Proc. The 1th European SE-Conference*, 1987, pp.64-68.
[27] Christoph Reichenberger, "Delta Storage for Arbitrary Non-Text Files,"*Proc. The 3rd international workshop on Software configuration management*, 1991, pp. 144-152.

Biographies



Zhijuan Hu, received her B.S in Electronic information engineering and M.S degree in Testing technology and automation device from Xi'an Polytechnic University, Xian, China, in 2005 and 2008, respectively. She is currently pursuing the Ph.D. degree in Military Communication from Xidian University. Her research interests focus on cognitive networks and data compression.

Chenxi Li, received the B.S. degree in communications engineering from Zhengzhou University, Zhengzhou, China, in 2015. She is currently working toward the Ph.D. degree in Military Communication from Xidian University. Her current research interests include frequency hopping systems, frequency hopping sequences, spectrum sensing networks and software defined cognitive radio.



Danyang Wang, received the B.S. degree in communications engineering and Ph.D. degrees in Military Communication from Xidian University in 2012 and 2017, respectively. He is now a post-doctoral researcher in information and communications engineering in Xidian University. His research interests include cognitive radio networks, cooperative spectrum sensing, and NOMA.

Tingting Wang, received the B.S. degree in communications engineering from Xidian University in 2016. She is currently pursuing the Ph.D. degree in information and communications engineering in Xidian University. Her research interests include physical layer security and covert communications.