IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 5, 2019, accepted February 14, 2019, date of publication February 18, 2019, date of current version March 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899831

A Sanitization Approach to Secure Shared Data in an IoT Environment

JERRY CHUN-WEI LIN[1,2], JIMMY MING-TAI WU[3], PHILIPPE FOURNIER-VIGER[4], YOUCEF DJENOURI[5], CHUN-HAO CHEN[6], AND YUYU ZHANG[1]

[1]School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China
[2]Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences, 5063 Bergen, Norway
[3]College of Computer and Engineering, Shandong University of Science and Technology, Qindao 266590, China
[4]School of Humantities and Social Sciences, Harbin Institute of Technology (Shenzhen), Shenzhen, China
[5]Department of Computer Science, Norweigan University of Science and Technology, 7491 Trondheim, Norway
[6]Department of Computer Science and Information Engineering, Tamkang University, New Taipei City 25137, Taiwan

Corresponding author: Jimmy Ming-Tai Wu (wmt@wmt35.idv.tw)

This work was supported by the Shenzhen Technical Project under Grant JCYJ20170307151733005 and Grant KQJSCX20170726103424709.

ABSTRACT Internet of Things (IoT) supports high flexibility and convenience in several applications because the IoT devices continuously transfer, share, and exchange data without human intervention. During shared or exchanged progress of data, security and privacy threats result because the published or shared data mainly corresponds to a raw dataset, and an attacker can easily obtain details on the shared data in an IoT environment. In the paper, we present a sanitization approach by adopting the hierarchical-cluster method to hide confidential information while still discovering useful and meaningful information in the sanitized dataset. The multi-objective particle swarm optimization framework and an algorithm termed as HCMPSO are utilized to balance four side effects, namely, *hiding failure*, *missing cost*, *artificial cost*, and database dissimilarity (*Dis*), and thereby provide optimized solutions for data sanitization. The experiments are performed to compare the performance of the designed HCMPSO with that of the single-objective cpGA2DT and multi-objective NSGA-II-based approaches. As shown in the results, the designed HCMPSO exhibits good performance in terms of *hiding failure*, and thus the most confidential information is hidden after the sanitization process. The shared or published data in IoT is secured. Furthermore, the designed sanitization algorithm achieves reasonable results in terms of *missing cost*, *artificial cost*, and *Dis*.

INDEX TERMS IoT security, PPDM, hierarchical cluster, data sanitization, multi-objective PSO.

I. INTRODUCTION

The Internet of Things (IoT) [7], [41] refers to devices/nodes that are physically connected and used to transfer, exchange, or share published information. Thus, IoT provides high flexibility as information can be easily obtained and shared among devices without human intervention. This results in convenience in decision-making. However, it can also cause security threats [8] especially when shared or published data is not sanitized. Several works [33], [47], [48], [51] regarding to the privacy-preserving and security issues of IoT devices and environment have discussed in recent decades and it has become a critical topic in recent years. To prevent security risks while still obtaining useful information for decision-making, privacy-preserving data mining (PPDM) [4], [6], [45], [49] is an option to sanitize confidential information while providing useful and meaningful information after the sanitization process. Thus, the published or shared data is secured and security risks are significantly reduced.

Data sanitization is a method of PPDM and is used to hide confidential information via perturbation technology. However, the process leads to side effects including *hiding failure*, *missing cost*, and *artificial cost* during the sanitization process. The *hiding failure* indicates that the confidential information is supposed to be hidden although it still exists after the sanitization process and is discovered during the mining process. *Missing cost* indicates that it is possible to miss the already discovered information after the sanitization process, and the *artificial cost* states that meaningless or unnecessary information is not discovered but is mined after the sanitization process. The side effect problem also corresponds to an NP-hard problem [4], [49] because more confidential information is hidden, and thus more loss occurs or new information appears. Several algorithms have been presented to minimize the three side effects during the

The associate editor coordinating the review of this manuscript and approving it for publication was SK Hafizul Islam.

sanitization process including straightforward (conventional) or optimization processes. Lindell and Pinkas presented an ID3 algorithm [37] to solve the problem of PPDM based on a decision-tree. Clifton *et al.* [13] designed a software to solve the problem of PPDM. Dwork *et al.* [17] presented several approaches that are used to handle the published noisy statistics to the vertically partitioned databases. Wu *et al.* [50] designed several algorithms to reduce support/confidence, thereby hiding the sensitive information by decreasing the support/confidence values. Hong *et al.* [28] utilized the TF-IDF method and presented a SIF-IDF algorithm to evaluate the score of each transaction for data sanitization. Several studies related to PPDM were examined, and most of them are based on a deletion procedure to hide sensitive information [15], [18], [28], [38].

The PPDM is non-trivial and is considered an NP-hard problem. Thus, it is difficult to determine the optimized solutions between the side effects. Several algorithms based on the evolutionary computation have been designed to obtain optimal solutions. For example, Lin *et al.* [39], [40] utilized genetic algorithms (GAs) to sanitize the database and presented the cpGA2DT and pGA2DT algorithms. It exhibited good results with respect to three side effects when compared to a greedy algorithm. Although the aforementioned algorithms achieve lower side effects when compared to the traditional algorithms, they rely on the pre-defined weight values of three side effects in the pre-defined fitness function. Thus, the results of side effects are significantly affected by the weighted values, the results are occasionally not optimized, and a-priori knowledge or expert is required to set up the weighted values. To handle the problem, Cheng *et al.* [14] developed an EMO-based algorithm to consider ''data distortion'' and ''knowledge distortion'' in the sanitization process via item deletion. Although the approach involves multi-objective functions, it can lead to incomplete knowledge for decision-making because it directly deletes the attributes from the databases. This is not applicable in the sequential dataset.

Specifically, NSGA-II [16] is a method that involves multi-objective functions rather than a single-objective function. Lin *et al.* [42] developed an algorithm by adapting the NSGA-II model for data sanitization in PPDM. The multi-objective particle swarm optimization (MOPSO) algorithm [12] is extended from the conventional particle swarm optimization (PSO) algorithm [34] although it handles the multi-objective problems to determine a set of Pareto solutions. However, the MOPSO framework cannot be utilized in a straightforward manner to handle the PPDM problem because dominant relationships should be utilized to obtain optimized transactions for deletion. In the study, we utilized the MOPSO framework and presented a hierarchical-cluster algorithm termed as HCMPSO. The major contributions of the study are summarized below.

- The study involves designing the MOPSO-based framework in PPDM by adopting the hierarchical-cluster method, and this exhibits better performance in terms of side effects when compared to the conventional single-objective approach and the NSGA-II-based model.
- Two updating strategies of the *gbest* (also termed as global best) and *pbest* (also termed as personal best) solutions in the updating process of MOPSO framework are utilized to obtain higher diversity of the solutions when compared to the NSGA-II-based model.
- To accelerate the evolutionary process the pre-large concept is utilized to accelerate the progress of the evaluated solution, and thus this significantly avoids a multiple database scan.
- The experiments demonstrated the performance of time cost and four side effects of the designed HCMPSO when compared to that of the traditional single-objective algorithm and NSGA-II-based approach.

Based on the aforementioned contributions, we believe that the designed HCMPSO can secure data in an IoT environment and prevent security threats especially in terms of shared and published data. The rest of the study is structured as follows. The literature review is presented in Section II. The preliminary problem statement is examined in Section III. The developed HCMPSO with two updating strategies is developed in Section IV. Several experiments are performed for various datasets in Section V. The conclusions and future work are discussed in Section VI.

II. LITERATURE REVIEW

Extant studies involving evolutionary computation, PPDM, and pre-large concepts are described as follows.

A. EVOLUTIONARY COMPUTATION

Evolutionary computation is used to solve the NP-hard problem by providing the optimal solutions in different applications and domains, and this was originally inspired by biological evolution. Holland applied Darwin's theory of natural selection and survival of the fittest to develop genetic algorithms (GAs) [26] that are widely used in computational intelligence to solve the NP-hard problem. Specifically, a GA consists of several operations including selection, crossover, and mutation to iteratively evaluate the solutions in the evolutionary process. Kennedy and Eberhart [34] presented particle swarm optimization (PSO), which is inspired by bird flocking activities, to search for optimal solutions. Thus, each particle in the PSO corresponds to a potential solution. In PSO, each bird exhibits its own velocity, and this is used to represent the direction for the other solutions. Furthermore, each particle is iteratively updated with its own *pbest* and *gbest* based on the pre-defined fitness function. Each particle is updated by the *gbest*, *pbest*, and its own velocity at each iteration. The processes are as follows [34]:

$$v_i(t+1) = w \times v_i(t) + c_1 \times r_1 \times (pbest_i - x_i(t)) + c_2 \times r_2 \times (gbest - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

As shown in the aforementioned equations, w is considered a factor and is utilized to balance between the global and

the local search. Specifically, v_i denotes the velocity of the i-th particle in a population where t denotes the t-th iteration. Furthermore, c_1 and c_2 denote constant values, and r_1 and r_2 are both represented as random numbers using a uniform distribution to retrieve values between 0 and 1 ([0, 1]). The velocity of the particle is updated by eq. (1), and its position is updated by eq. (2). Several meta-heuristic algorithms were presented and applied in several realistic problems and situations to determine optimal solutions such as the ant colony optimization (ACO) [9] or artificial bee colony (ABC) [36].

The aforementioned algorithms mainly deal with single-objective optimization in a fitness function, and thus the derived solutions cannot be optimized because more than two objectives can be combined in real-world applications to obtain solutions. Multi-objective genetic algorithms (MOGAs) [19], non-dominated sorting genetic algorithm (NSGA) [46], and NSGA-II model [32] were designed to merge more objectives to obtain Pareto solutions. Each solution in the Pareto solutions exhibits a non-dominated relation with each other. Extensions of the multi-objective algorithms were examined, such as the strength Pareto evolutionary algorithm (SPEA) [52] and the Pareto archived evolution strategy (PAES) [35]. Coello and Lechuga [12] introduced the multi-objective particle swarm optimization (MOPSO) framework that applies the adaptive grid method to maintain an external archive, change the direction of particles when they flow out of the search space, and maintain particles within a boundary.

B. PRIVACY-PRESERVING DATA MINING

As the rapid growth of IoT devices and environment, the security and privacy issues in IoT have discussed in recent decades. Song *et al.* [47] discussed the smart home systems and presented an energy-efficient, secure, and privacy-preserving communication protocol for it. From the results, it showed that the designed system achieves good performance in terms of computational complexity, memory cost, and communication overhead compared to the previous works. Togan *et al.* [48] applies authentication service for smart-home devices using a smart-phone as security anchor, QR codes and attribute based cryptography for the security of IOT devices. In recent years, cloud environment is used to keep the large scale data, thus the privacy issue in cloud has become a critical issue. Jayaraman *et al.* [33] mentioned the IoT privacy preservation problem and presented innovative techniques for privacy preservation of IoT data by introducing a privacy preserving IoT architecture. The designed architecture utilizes multiple IoT cloud data stores to protect the privacy of data collected from IoT. Yang *et al.* [51] presented the e-health system in IOT environment. The system consists of the non-interactive and authenticated key distribution procedure to enable flexible access policy updating without privacy leakage.

In the past two decades, data mining [2], [3], [10], [11], [21], [23]–[25], [29], [44] has been an efficient method to discover potential/useful information from an extremely large database. Specifically, relationships between products cannot be easily discovered and visualized. Given that information can be revealed from the databases, confidential/secure information is also discovered during the mining procedure, thereby leading to privacy and security threats to users. Currently, PPDM is a critical issue because it exhibits the ability to determine useful information for decision-making and also hide confidential/secured data using a sanitization procedure. Following the sanitization pf PPDM, confidential information is hidden to maintain data security. Agrawal and Srikant [5] presented a new reconstruction algorithm that accurately estimated the distribution of original data. The classifiers were also constructed to compare the accuracy between original data and sanitized data. Verykios *et al.* [49] developed hierarchical classification techniques that are utilized in PPDM. Dasseni *et al.* [15] designed an approach that was based on the hamming-distance mechanism to decrease the support or confidence of the sensitive information (i.e., association rules) for sanitization. Oliveira and Zaïane [45] developed several sanitization approaches that were utilized to hide frequent itemsets by the developed heuristic method. The developed algorithms used the item-restriction approach to avoid noise addition and limited the removal of real dataset. Islam and Brankovic [31] presented a framework via the noise addition method to protect and hide individual privacy while maintaining high data quality. Hong et al. developed the SIF-IDF method [28] that utilized the TF-IDF concepts to assign the weight of each transaction. Subsequently, the developed sanitization algorithm was implemented to iteratively sort transactions from the transaction with the highest score to that with the lowest score.

The aforementioned algorithms mainly focus on hiding the sensitive or confidential information in a straightforward manner by their developed approaches, and thus it is not possible to optimize the obtained results. The progress of the PPDM also involves an NP-hard problem [4], [49], and thus it is better to provide the meta-heuristic approaches to determine the optimal solutions. Han and Ng [30] designed a secure protocol that was used to discover a better set of rules without disclosing own private data via genetic algorithms (GAs) [22], [26], and the result of true positive rate times true negative rate was calculated to evaluate each decision rule. Lin et al. developed several GA-based algorithms including sGA2DT, pGA2DT [40], and cpGA2DT [39] to hide confidential information via transaction deletion for data sanitization. The encoded chromosome was considered as a set of solutions, and the transaction of the gene within chromosome corresponded to the victim for subsequent deletion. A fitness function was also developed to consider three side effects for evaluation with pre-defined weights to demonstrate the goodness of the chromosome. Although the aforementioned algorithms are efficient in terms of determining the optimal transactions for deletion, they still require the pre-defined weights of side effects. The mechanism significantly affects the final results of the designed approaches.

Cheng *et al.* [14] designed an EMO-RH approach by adopting the EMO for data sanitization. Although the method is based on the multi-objective framework, it produces incomplete transactions; and this can lead to misleading decisions especially in the treatment of hospital diagnoses. Lin *et al.* [42] presented a meat-heuristic approach that was based on the NSGA-II framework for data sanitization, and this exhibited better side effects when compared to the single-objective algorithms.

III. PRELIMINARY AND PROBLEM STATEMENT

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a finite set of r distinct items that appear in database D. Additionally, D denotes a set of transactions such as $D = \{T_1, T_2, \ldots, T_n\}$, and $T_q \in D$. Each T_q denotes a subset of I and corresponds to a unique identifier q and is termed as *TID*. If a support of the itemset exceeds the minimum support count ($minsup \times |D|$, then *minsup* is defined as minimum support threshold), and it is considered as a frequent itemset and is put into the set of *FI*. The set of confidential information is denoted as $CI = \{c_1, c_2, \ldots, c_k\}$, and this is defined by user experience or preferences. Furthermore, each confidential information is also a subset of *FI*, and this corresponds to $c_i \in FI$.

Definition 1: For each $c_i \in CI$, the size of Deleted Transactions for hiding c_i is denoted as $DT(c_i)$, and this is defined as:

$$DT(c_i) = \frac{sup(c_i) - minsup \times |D|}{1 - minsup}, \tag{3}$$

where $sup(c_i)$ is defined as the support count of c_i in the database.

Definition 2: The maximal number of deleted transactions of all confidential information in *CI* is denoted as *MDT*, and this is defined as:

$$MDT = max\{DT(c_1), DT(c_2), \ldots, DT(c_k)\}. \tag{4}$$

In the designed HCMPSO algorithm, *MDT* is considered as the size of a particle in the MOPSO model. To completely hide the confidential information in the database, the set of *CI* corresponds to *null* after the sanitization process. However, this process can lead to significant side effects in terms of missing and artificial costs because the three side effects exhibit a trade-off relationship. Therefore, an optimization process is required to determine the balance between the three side effects. The details of the three side effects are described below.

Definition 3: The *hiding failure* denotes the number of confidential information that is not hidden after the sanitization process, and this is denoted as α and defined as:

$$\alpha = |CI \cup FI'|, \tag{5}$$

where FI' is defined as the set of FIs (frequent itemsets) after data sanitization.

Definition 4: The *missing cost* denotes the number of itemsets that are discovered as large itemsets before sanitization but are hidden after data sanitization. This is denoted as β and defined as:

$$\beta = |FI - CI - FI'|, \tag{6}$$

where *FI* is used to maintain the FIs in the original database before sanitization.

Definition 5: The *artificial cost* denotes the number of itemsets that arise and are not discovered as the large itemset before sanitization but appear as the frequent itemset after the sanitization process. This is denoted as γ and defined as:

$$\gamma = |FI' - FI|. \tag{7}$$

The relationships between α, β, and γ are shown in Fig. 1.

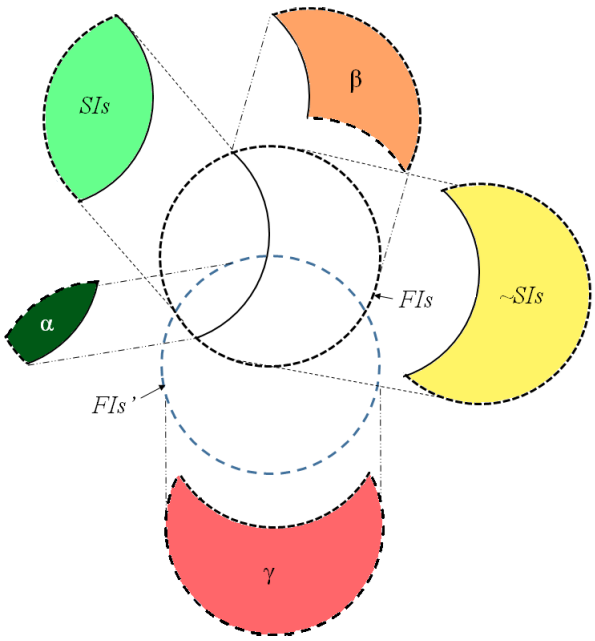

FIGURE 1. Three side effects of the sanitization process.

To determine more optimized solutions related to considering more objectives, the similarity between the original database and sanitized database (*Dis*) [40] is explored as a side effect of the optimization and is discussed as follows.

Definition 6: Database dissimilarity is used to measure the number of deleted transactions between the original database and sanitized database, and this is denoted as *Dis* and defined as:

$$Dis = |D - D'|, \tag{8}$$

where *D* denotes the original database before sanitization and *D'* denotes the database after sanitization process.

From the perspective of PPDM optimization, this corresponds to an NP-hard problem, and thus it is critical to determine the trade-off relationships among the four side effects. In the study, we utilize the MOPSO-based framework in the developed HCMPSO algorithm. We consider the hierarchical clustering method to determine the groups of the particles and assign the probability of the solutions for subsequent selection. The approach provides a higher diversity of derived

solutions when compared to the NSGA-II-based model. Based on the approach, the designed HCMPSO significantly hides confidential information when compared to the traditional single-objective approach and the NSGA-II-based model. The problem statement of the study is defined below.

Problem Statement: The designed sanitization algorithm involves deleting the most relevant transactions to hide confidential information based on the multi-objective particle swarm optimization (MOPSO) framework. Furthermore, the designed model should consider four side effects and minimize them to the maximum possible extent. This especially holds for the hiding failure because the aim of PPDM involves hiding the confidential information as the first priority. Thus, the problem statement of the study is defined as:

$$minf(p) = [f_1(p), f_2(p), f_3(p), f_4(p)], \tag{9}$$

where f_1 corresponds to α, f_2 corresponds to β, f_3 corresponds to γ, and f_4 corresponds to Dis. Furthermore, p denotes a solution or a particle in the designed HCMPSO algorithm.

IV. PROPOSED SANITIZATION FRAMEWORK

In this section, a sanitization framework based on the multi-objective particle swarm optimization in PPDM is presented. It exhibits two main phases. With respect to the first phase, the frequent itemsets are discovered and placed into the set of *FIs* for later progress. Furthermore, the transactions with any confidential information are subsequently projected in a new database for later progress. With respect to the second phase, two updating strategies of *gbest* and *pbest* and a designed algorithm are developed to iteratively update the particles in the evolutionary process. The details are described below.

A. PRE-PROCESSING

Before the sanitization process, a user has to set the confidential information that is required to be hidden, and the itemsets are placed in the set of *CI*. The frequent itemsets are discovered relative to the minimum support count and placed into the set of *FIs*. To obtain better transactions for deletion in PPDM, the database is first processed to project the transactions with any of the confidential information appearing in the set of *CI*. The projected database is set as D^*. Each transaction in D^* consists of at least one itemset within the set of *CI*, and this is also considered as a candidate of the particles for subsequent deletion in the evolutionary process. The detailed algorithm is given in Algorithm 1.

In the pre-processing phase, D denotes the original database, and *CI* denotes the set of confidential itemsets, respectively. The outputs D^* and *FIs* denotes the projected database and set of frequent itemsets, respectively. First, the original database is scanned to obtain the *FIs* by the *minsup* (Line 1). Subsequently, if a transaction consists of any of the itemset within CI, then the transaction is projected as D^* (Lines 2 to 4). The size of each particle is calculated (Line 5) for later evolutionary process. Finally, the projected database (D^*), and the frequent itemsets (*FIs*) are returned as the outputs (Line 6) for the next phase.

Algorithm 1 Pre-Processing Phase

Input: D, the original database; *CI*, the set of confidential information; *minsup*, the minimum support threshold.

Output: D^*, the projected database; *FIs*, the set of frequent itemsets.

```
1 find FIs by minsup;
2 for q:= 1, n; i:= 1, k do
3     if c_i ⊆ T_q then
4         D* = D* ∪ T_q;
5 calculate the size of particle by eq. (3) and (4);
6 return D*, FIs;
```

B. EVOLUTION PROGRESS

In the second phase, the particles are evaluated to obtain better solutions for next iteration. In the developed sanitization framework, each particle exhibits a possible solution and the size of the particle is defined as *MDT* vectors. Each vector in a particle is defined as the *TID* (transaction ID), and this indicates that the transactions should be potentially deleted for data sanitization. The formulas to update the velocity and its position in the designed model are defined as follows:

$$v_i(t+1) = (pbest - x_i(t)) \cup (gbest - x_i(t)) \tag{10}$$

$$x_i(t+1) = rand(x_i(t), null) + v_i(t+1) \tag{11}$$

With respect to the updating progress, the *TIDs* within the elder particle or *null* value are randomly selected for next iteration if the size of the particle does not achieve *MDT*. The position of the updated particle of next iteration is summed up with the updated velocity of the particle. The aforementioned equations exhibit high randomization and exploration in the evolutionary process.

Given that the designed HCMPSO is based on the multi-objective PSO framework for utilizing the sanitization progress, it is not possible to directly apply traditional updating strategies of *pbest* and *gbest* in PSO to the developed approach. Thus, the un-dominated relation is utilized for the updating progress. The LUS is utilized here to update the *pbest* value as follows:

Pruning Strategy 1 (Local Updating Strategy, LUS):

$$pbest \leftarrow \begin{cases} x(t+1) & \text{if } f(x(t+1)) \succ f(pbest) \\ rand(x(t+1), pbest) & otherwise. \end{cases} \tag{12}$$

Thus, if the current particle dominates its last *pbest*, then the *pbest* is replaced by the current particle; otherwise, a random selection is performed to select a particle as the *pbest* for next iteration.

To update the *gbest*, the GUS is utilized to obtain a better solution in the evolutionary progress. In the designed HCMPSO, a hierarchical clustering method is used to group the solutions. Each particle is assigned its own probability

Algorithm 2 Proposed HCMPSO Sanitization Algorithm

Input: D^*, the projected database; CI, the confidential information; FIs, the set of frequent itemsets; N, the size of populations; $minf(x) = [f_1(p), f_2(p), f_3(p), f_4(p)]$, the multi-objective fitness function.
Output: PS^*, a set of Pareto solutions.

1 set t:= 0;
2 initial N populations with MDT ;
3 put generated particle p into the set of POP_t;
4 set $Pool \leftarrow null$;
5 **while** *termination criteria is not achieved* ($t < N$) **do**
6 **for** *each* $p \in POP_t$ **do**
7 evaluate $minf(p) = [f_1(p), f_2(p), f_3(p), f_4(p)]$;
8 **if** $Pool \neq null$ **then**
9 **for** *each* $c \in Pool$ **do**
10 **if** $p \succ c$ **then**
11 remove c from $Pool$;
12 $Pool \leftarrow \cup p$;
13 **else**
14 $Pool \leftarrow \cup p$;
15 **if** t:= 0 **then**
16 set nc:= $\frac{|Pool|+1}{2}$;
17 initialize $|Pool|$-clusters;
18 **HCProb**($Pool$, nc);
19 update $pbest$ and $gbest$;
20 generate POP_{t+1};
21 t++;
22 return PF^*;

based on the number of clusters and the number of the particles within a cluster. A random selection is performed to select a candidate particle as the *gbest* based on their assigned probability. The GUS is expressed as follows.

Pruning Strategy 2 (Global Updating Strategy, GUS):

$$gbest \leftarrow rand(p_{prob}), \tag{13}$$

With respect to the designed sanitization HCMPSO algorithm, the details are described in Algorithm 2.

The D^*, CI, FIs, and N correspond to the projected database, set of confidential information, set of large itemsets, and number of populations, respectively. The f_1, f_2, f_3, and f_4 represent four fitness functions corresponding to the number of *hiding failure*, number of *missing cost*, number of *artificial cost*, and number of dissimilar databases (*Dis*), respectively. First, N particles are randomly initialized based on the size of each particle as MDT (Line 2). The generated particles are placed into the set of POP_0 as the initial populations. The candidate set of the Pareto front is set as *null* (Line 4). Subsequently, the iteratively progress is performed until the termination criteria is achieved such as the size of iteration ($t < N$, Lines 5 to 21). Each particle in the candidate set of

Algorithm 3 HCProb ($Pool$, nc)

Input: $Pool$, the set of particles.
Output: $prob(p)$, the probability of each particle in PF.

1 set $iter$:= $|Pool|$;
2 **while** $iter > nc$ **do**
3 merge nearest two clusters c_n and c_m ;
4 $iter$:= $iter$ - 1;
5 **for** *each cluster* c **do**
6 **for** *each particle* $p \in c$ **do**
7 $prob(p) := \frac{1}{nc} \times \frac{1}{|c|}$;
8 return $prob(p)$;

TABLE 1. Parameters of the used datasets.

$\#\|\mathbf{D}\|$	Total number of transactions
$\#\|\mathbf{I}\|$	Number of distinct items
AvgLen	Average transaction length
MaxLen	Maximal length transactions
Type	Dataset type

Pareto front is evaluated by four fitness functions (Line 7) to determine the non-dominated solutions (Lines 8 to 14). The satisfied solutions are performed by the **HCProb** function to assign the probability of each particle based on the hierarchical clustering method. The pseudo-code of the **HCProb** function is illustrated in Algorithm 3.

In Algorithm 3, the size of the $Pool$ is used to set the k-clusters for the particles, and the nearest to clusters are merged together (Lines 2 to 4). Subsequently, the particles within a cluster are assigned with a probability (Lines 5 to 7), and this is used to select the *gbest* with high diversity of the solutions.

V. EXPERIMENTAL RESULTS

Several experiments are conducted to demonstrate the effectiveness and efficiency of the designed HCMPSO when compared to the single-objective cpGA2DT [39] and multi-objective NSGA-II-based approach [42]. All the compared algorithms are implemented in Java language and are accessed from PPSF website [43]. Experiments are performed on a personal computer (PC) with an Intel Core i7-6700 Quad-Core Processor and 8GB main memory that is run on a 64-bit Microsoft Windows 10 operating system. Three real-world datasets [20], namely chess, mushroom, and foodmat, and a synthetic dataset of T10I4D100K [1] are used in the experiments. The corresponding minimum support thresholds for four datasets are set as 90%, 45%, 0.32%, and 2.5% and are adjusted based on user preferences. Parameters and characteristics of the datasets used in the experiments are shown in Tables 1 and 2. The results in terms of runtime and four side effects are discussed and analyzed as follows.

A. RUNTIME

In the experiments, the execution time under varying sensitive percentages with a fixed minimum support threshold is conducted in the four datasets. We also applies the pre-large

TABLE 2. Characteristics of used datasets.

Dataset	#|D|	#|I|	AvgLen	MaxLen	Type
chess	3,196	74	37	37	dense
mushroom	8,124	119	23	23	dense
foodmart	21,556	1,559	4	11	sparse
T10I4D100K	100,000	870	10.1	29	sparse

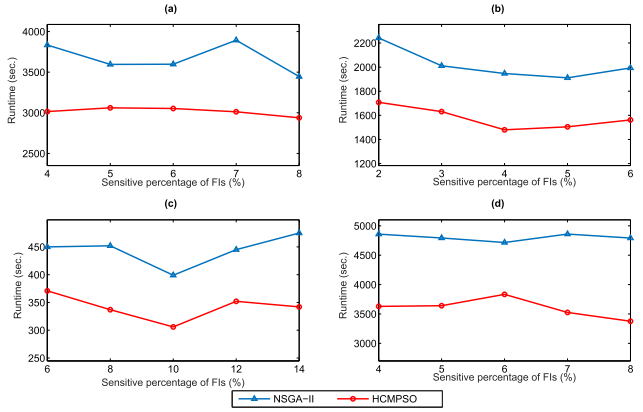

FIGURE 2. Runtime under varying sensitive percentages of frequent itemsets. (a) Chess (minsup: 90%). (b) Mushroom (minsup: 45%). (c) Foodmart (minsup: 0.32%). (d) T10I4D100K(minsup: 2.5%).

concept [27] to speed up computation in the evolutionary progress. The pre-large concepts also help to reduce the multiple database scans since the *artificial cost* can be easily evaluated and updated. In this section, only the NSGA-II-based model [42] is compared to the designed HCMPSO because it is not reasonable to compare the single-objective algorithm with the multi-objective algorithms. Two compared algorithms are also utilized by the pre-large concepts to speed up computations. The results of the two multi-objective algorithms are shown in Fig. 2.

It is observed that the developed HCMPSO consumes significantly less runtime than the NSGA-II-based model under varying sensitive percentages of frequent itemsets. This is because the NSGA-II-based model consumes significant time in terms of generating new populations via crossover and mutation operations. The sorting strategy of the Pareto solutions in NSGA-II-based model also involves considerable computational time. However, the presented HCMPSO is not required to perform the crossover and mutation operations to generate next populations. Thus, the HCMPSO requires less runtime than that of the NSGA-II-based model. The results in terms of four side effects are discussed next.

B. SIDE EFFECTS

In this section, the state-of-the-art single-objective cpGA2DT [39] and multi-objective NSGA-II-based model [42] are compared with the designed HCMPSO in terms of *hiding failure* (α), *missing cost* (β), *artificial cost* (γ), and database dissimilarity (*Dis*). The population for all evolutionary algorithms is set as 50. Given that both the designed HCMPSO and the NSGA-II-based approach generate a set of Pareto solutions, we evaluate the side effects as an average of the generated solutions. The results are discussed and described as follows.

1) HIDING FAILURE

The results of *hiding failure* for the four datasets are shown in Fig. 3.

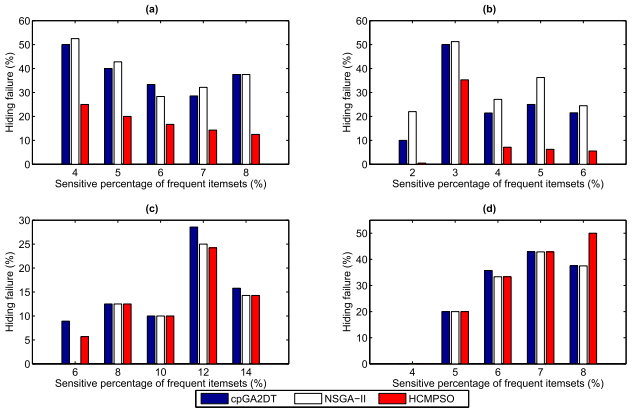

FIGURE 3. Hiding failure under varied sensitive percentages of frequent itemsets. (a) Chess (minsup: 90%). (b) Mushroom (minsup: 45%). (c) Foodmart (minsup: 0.32%). (d) T10I4D100K(minsup: 2.5%).

As shown in Fig. 3, the designed HCMPSO evidently reaches the lowest *hiding failure* in most cases. The results indicate that confidential information is mainly hidden after the sanitization process when compared to the other two approaches. For example, when the sensitive percentage of the frequent itemsets is set as 4%, 5%, 6%, 7%, and 8% for the chess dataset, the *hiding failures* of the designed HCMPSO under varied sensitive percentages of frequent itemsets are lower than 30%, whereas those of the other two approaches exceed 30%, and the NSGA-II-based model exhibits more than 50% *hiding failure* when the sensitive percentage of frequent itemsets is set as 4% for the chess dataset. Furthermore, it is also observed that the hiding failure of the developed HCMPSO is almost zero when the sensitive percentage of frequent itemset is set as 2% for the mushroom dataset. With respect to the foodmart and T10I4D100K datasets, the developed HCMPSO still exhibits good performance when compared to the other two approaches. Generally, we conclude that the designed HCMPSO exhibits good results in terms of *hiding failure* when compared to the cpGA2DT or NSGA-II-based model.

2) MISSING COST

We discuss the results of *missing cost* as shown in Fig.4.

As shown in Fig. 4, the HCMPSO occasionally obtains higher *missing cost* under chess and mushroom datasets. This is because the two datasets belong to the dense dataset, and thus the contents of most transactions exhibit high overlapping. Thus, when more confidential information is deleted from the dataset, more discovered information is deleted together. As shown in Figs. 3(a) and 3(b), the HCMPSO achieves lowest *hiding failure* than the other algorithms. Thus, the HCMPSO occasionally produces higher *missing cost* in extremely dense datasets, and this is reasonable. However, for spare datasets, such as foodmart and T10I4D100K that are shown in Figs. 4(c) and 4(d), respectively,

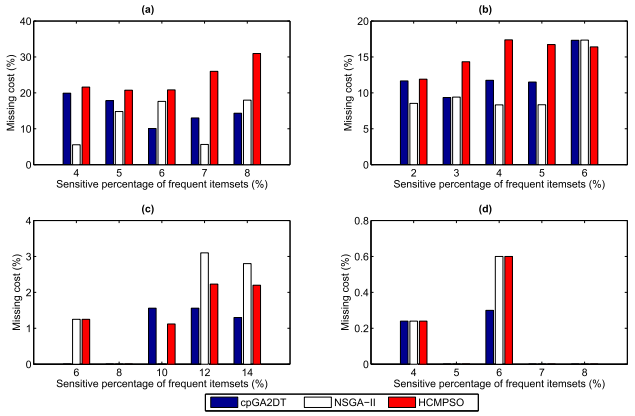

FIGURE 4. Missing cost under varied sensitive percentages of frequent itemsets. (a) Chess (minsup: 90%). (b) Mushroom (minsup: 45%). (c) Foodmart (minsup: 0.32%). (d) T10I4D100K(minsup: 2.5%).

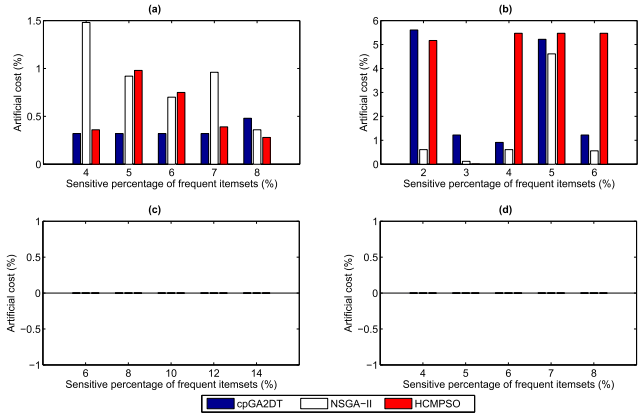

FIGURE 5. Artificial cost under varied sensitive percentages of frequent itemsets. (a) Chess (minsup: 90%). (b) Mushroom (minsup: 45%). (c) Foodmart (minsup: 0.32%). (d) T10I4D100K(minsup: 2.5%).

the designed HCMPSO occasionally does not produce *missing cost* under varied sensitive percentages of frequent itemsets. For example, all the compared algorithms exhibit no side effect of *missing cost* when the sensitive percentage of frequent itemsets is set as 8% for the foodmart dataset and when the sensitive percentages of frequent itemsets are respectively set as 5%, 7%, and 8% for the T10I4T100K dataset. Generally, the *missing cost* of the developed HCMPSO is acceptable and especially most confidential information is hidden by the designed HCMPSO.

3) ARTIFICIAL COST

Results of *artificial cost* are shown in Fig. 5.

As shown in Fig. 5, the HCMPSO generates higher *artificial cost* when compared to the cpGA2DT and NSGA-II-based models for extremely dense datasets such as the mushroom shown in Fig. 5(b). This is because the size of the database is subsequently reduced when more *hiding failure* is prohibited (for example, the results shown in Fig. 3(b)). Thus, more unexpected rules subsequently arise as new information after the sanitization process because the minimum support count is changed. This is reasonable because it exhibits a trade-off relationship between *hiding failure*

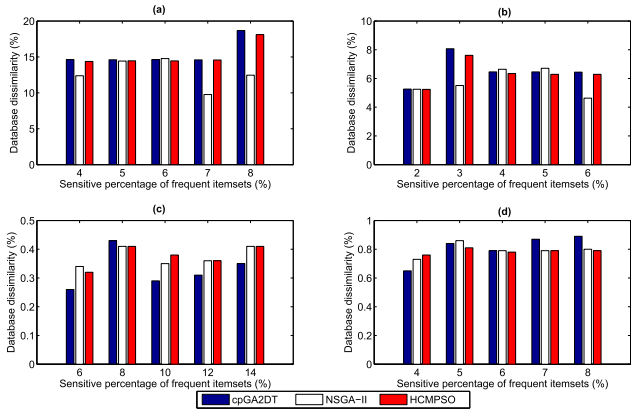

FIGURE 6. Database dissimilarity under varied sensitive percentages of frequent itemsets. (a) Chess (minsup: 90%). (b) Mushroom (minsup: 45%). (c) Foodmart (minsup: 0.32%). (d) T10I4D100K(minsup: 2.5%).

and *artificial cost*. When compared to the NSGA-II-based model shown in 5(a), the designed HCMPSO still exhibits lower *artificial cost* than that of the NSGA-II-based model. With respect to sparse datasets (such as foodmart and T10I4D100K), all the algorithms do not generate any of the *artificial cost*, and the developed HCMPSO as well as the three compared algorithms obtain the optimized results of *artificial cost* for the sparse datasets.

4) DATABASE DISSIMILARITY

Results of the database similarity (*Dis*) are discussed and shown in Fig. 6.

As shown in Fig. 6, the NSGA-II-based model mainly exhibits the optimal *Dis* for the dense datasets such as chess and mushroom when compared to the other two algorithms. This is because as shown in Figs. 3(a) and 3(b), the NSGA-II-based model exhibits worse results in terms of *hiding failure*, less information is deleted, and thus the side effect of *Dis* is not high. As observed, the proposed HCMPSO exhibits good performance in terms of *hiding failure* shown in Figs. 3(a) and 3(b); and thus more confidential information is deleted and the *Dis* value increases. With respect to the sparse datasets (such as foodmart and T10I4D100K shown in Figs. 6(c) and 6(d)), the *Dis* for cpGA2DT, NSGA-II-based model, and the HCMPSO are not higher, and this corresponds to less than 1% difference. Therefore, the HCMPSO exhibits good *Dis* when compared with that of the other two approaches. In summary, the designed HCMPSO exhibits a good performance in terms of four side effects and obtains optimized solutions when compared to the other two approaches. Furthermore, the designed HCMPSO exhibits higher flexibility when compared with that of the single-objective cpGA2DT algorithm because the transactions for deletion are selected by user preferences.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a hierarchical clustering method to find better particles as the Pareto solutions based on the

multi-objective particle swarm optimization framework. The HCMPSO algorithm is designed here as the sanitization approach to hide the confidential information. Two updating strategies are also utilized here to find the better diversity of the obtained solutions. From the results, we can observe that the designed HCMPSO has good *hiding failure* compared to the other two approaches. Moreover, in terms of *missing cost*, *artificial cost*, and database dissimilarity (*Dis*), the designed HCMPSO still obtain good performance compared to the other approaches, and in some cases, it can reach the optimized results. Thus, we then can conclude that the designed HCMPSO can secure the confidential information of the shared or published data, which is quiet suitable for the IoT environment. Since each item/product may have its own specific threshold to identify whether it is a frequent or useful item, it would be an interesting topic to consider more specific thresholds of different items in privacy-preserving data mining. Moreover, the utility factor can be also considered to handle the issues of privacy-preserving utility mining as our future work.

REFERENCES

[1] R. Agrawal and R. Srikant. (1994). Quest Synthetic Data Generator. IBM Almaden Research Center. [Online]. Available: http://www.Almaden.ibm.com/cs/quest/syndata.html

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. Int. Conf. Very Large Data Base*, 1994, pp. 487–499.

[3] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. Int. Conf. Data Eng.*, 1995, pp. 3–14.

[4] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios, "Disclosure limitation of sensitive rules," in *Proc. Workshop Knowl. Data Eng. Exchange*, 1999, pp. 45–52.

[5] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, 2000.

[6] C. C. Aggarwal, J. Pei, and B. Zhang, "On privacy preservation against adversarial data mining," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 510–516.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[8] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Secur. Appl.*, vol. 38, pp. 8–27, Feb. 2018.

[9] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proc. Eur. Conf. Artif. Life*, 1991, pp. 134–142.

[10] D. W. Cheung, J. Han, V. T. Ng, and C. Y. Wong, "Maintenance of discovered association rules in large databases: An incremental updating technique," in *Proc. Int. Conf. Data Eng.*, 1996, pp. 106–114.

[11] M. S. Chen, J. Han, and P. S. Yu, "Data mining: An overview from a database perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 6, pp. 866–883, Dec. 1996.

[12] C. A. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, May 2002, pp. 1051–1056.

[13] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM SIGKDD Explor.*, vol. 4, no. 2, pp. 28–34, 2003.

[14] P. Cheng, I. Lee, C. W. Lin, and J. S. Pan, "Association rule hiding based on evolutionary multi-objective optimization," *Intell. Data Anal.*, vol. 20, no. 3, pp. 495–514, 2016.

[15] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding association rules by using confidence and support," in *Proc. Int. Workshop Inf. Hiding*, 2001, pp. 369–383.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[17] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptography Conf.*, vol. 3876, 2006, pp. 265–284.

[18] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 217–228.

[19] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. Int. Conf. Genetic Algorithms*, 1993, pp. 416–423.

[20] P. Fournier-Viger, J. C. W. Lin, A. Gomariz, T. Gueniche, A. Soltani, and Z. Deng, "The SPMF open-source data mining library version 2," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2016, pp. 36–40.

[21] P. Fournier-Viger, J. C. W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Sci. Pattern Recognit.*, vol. 1, no. 1, pp. 54–77, 2017.

[22] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

[23] W. Gan, J. C. W. Lin, H. C. Chao, and J. Zhan, "Data mining in distributed environment: A survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 7, no. 6, pp. 1–19, 2017.

[24] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, T. P. Hong, and H. Fujita, "A survey of incremental high-utility itemset mining," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 2, pp. 1–23, 2018.

[25] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, and P. S. Yu, "HUOPM: High utility occupancy pattern mining," *IEEE Trans. Cybern.*, to be published. doi: 10.1109/TCYB.2019.2896267.

[26] J. H. Holland, "*Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[27] T. P. Hong, C. Y. Wang, and Y. H. Tao, "A new incremental data mining algorithm using pre-large itemsets," *Intell. Data Anal.*, vol. 5, no. 2, pp. 111–129, 2001.

[28] T. P. Hong, C. W. Lin, K. T. Yang, and S. L. Wang, "Using TF-IDF to hide sensitive itemsets," *Appl. Intell.*, vol. 38, no. 4, pp. 502–510, 2012.

[29] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining Knowl. Discovery*, vol. 8, no. 1, pp. 53–87, 2004.

[30] S. Han and W. K. Ng, "Privacy-preserving genetic algorithms for rule discovery," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*, 2007, pp. 407–417.

[31] M. Z. Islam and L. Brankovic, "Privacy preserving data mining: A noise addition framework using a novel clustering technique," *Knowl.-Based Syst.*, vol. 24, no. 8, pp. 1214–1223, 2011.

[32] S. Jeyadevi, S. Baskar, C. K. Babulal, and M. W. Iruthayarajan, "Solving multiobjective optimal reactive power dispatch using modified NSGA-II," *Int. J. Elect. Power Energy Syst.*, vol. 33, no. 2, pp. 219–228, 2011.

[33] P. P. Jayaraman, X. Yang, A. Yavari, D. Georgakopoulos, and X. Yi, "Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation," *Future Gener. Comput. Syst.*, vol. 76, pp. 540–549, Nov. 2017.

[34] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Dec. 1995, pp. 1942–1948.

[35] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 1999, pp. 98–105.

[36] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, pp. 459–471, 2007.

[37] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Proc. Annu. Int. Cryptol. Conf. Adv. Cryptol.*, 2000, pp. 6–54.

[38] C. W. Lin, T. P. Hong, C. C. Chang, and S. L. Wang, "A greedy-based approach for hiding sensitive itemsets by transaction insertion," *J. Inf. Hiding Multimedia Signal Process.*, vol. 4, no. 4, pp. 201–227, 2013.

[39] C. W. Lin, B. Zhang, K. T. Yang, and T. P. Hong, "Efficiently hiding sensitive itemsets with transaction deletion based on genetic algorithms," *Sci. World J.*, vol. 2014, Sep. 2014, Art. no. 398269.

[40] C. W. Lin, T. P. Hong, K. T. Yang, and S. L. Wang, "The GA-based algorithms for optimizing hiding sensitive itemsets through transaction deletion," *Appl. Intell.*, vol. 42, no. 2, pp. 210–230, 2015.

[41] S. Li, L. Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.
[42] J. C. W. Lin, Y. Zhang, P. Fournier-Viger, Y. Djenouri, and J. Zhang, "A metaheuristic algorithm for hiding sensitive itemsets," in *Proc. Int. Conf. Database Expert Syst. Appl.*, 2018, pp. 492–498.
[43] J. C. W. Lin, P. Fournier-Viger, L. Wu, W. Gan, Y. Djenouri, and J. Zhang, "PPSF: An open-source privacy-preserving and security mining framework," in *Proc. IEEE Int. Conf. Data Mining Workshop*, Nov. 2018, pp. 1459–1463.
[44] J. C. W. Lin, L. Yang, P. Fournier-Viger, and T. P. Hong, "Mining of skyline patterns by considering both frequent and utility constraints," *Eng. Appl. Artif. Intell.*, vol. 77, pp. 229–238, Jan. 2019.
[45] S. R. M. Oliveira and O. R. Zaiane, "Privacy preserving frequent itemset mining," in *Proc. IEEE Int. Conf. Privacy, Secur. Data Mining*, Dec. 2002, pp. 43–54.
[46] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Dec. 1994.
[47] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A privacy preserving communication protocol for IoT applications in smart homes," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1844–1852, Dec. 2017.
[48] M. Togan, B.-C. Chifor, I. Florea, and G. Gugulea, "A smart-phone based privacy-preserving security framework for IoT devices," in *Proc. Int. Conf. Electron., Comput. Artif. Intell.*, 2017, pp. 1–7.
[49] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *ACM SIGMOD Rec.*, vol. 33, no. 1, pp. 50–57, 2004.
[50] Y. H. Wu, C. M. Chiang, and A. L. P. Chen, "Hiding sensitive association rules with limited side effects," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 1, pp. 29–42, Jan. 2007.
[51] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving fusion of IoT and big data for e-health," *Future Gener. Comput. Syst.*, vol. 86, pp. 1437–1455, Sep. 2018.
[52] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

JERRY CHUN-WEI LIN received the Ph.D. degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, in 2010. He is currently an Associate Professor with the Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 250 research papers in referred journals and international conferences. He is the Funder of the PPSF Project (http://ppsf.ikelab.net). His research interests include data mining, artificial intelligence, soft computing, privacy preserving data mining and security, social networks, and cloud computing. He is the Editor-in-Chief of *Data Science and Pattern Recognition* Journal.

JIMMY MING-TAI WU received the Ph.D. degree from the Department of Computer Science and Engineering, National Sun Yat-sen University, Taiwan. He is currently an Assistant Professor with the College of Computer and Engineering, Shandong University of Science and Technology, Qindao, Shandong, China. His research emphases are based on artificial intelligence, data mining, fuzzy theory, evolutionary computation, deep learning, big data, and cloud computing.

PHILIPPE FOURNIER-VIGER received the Ph.D. degree. He is currently a Full Professor with the Harbin Institute of Technology, Shenzhen, China, and National Talent in China. He has participated in more than 190 research papers, which have received more than 2,800 citations, including more than 800 citations in the last year. His research interests include data mining, algorithm design, pattern mining, sequence mining, sequence prediction, itemset mining, graph mining, big data, and applications. He is the Founder of the popular SPMF data mining library, which offers more than 150 algorithms, and has been cited in more than 650 research papers, since 2010. He is the Co-Editor-in-Chief of the *Data Science and Pattern Recognition* journal.

YOUCEF DJENOURI received the Ph.D. degree in computer engineering from the University of Science and Technology Houari Boumediene, Algiers, Algeria, in 2014. From 2014 to 2015, he was a Permanent Teacher-Researcher with the University of Blida, Algeria, where he was a member of LRDSI Lab. He was granted a Postdoctoral Fellowship from the Ulsan National Institute of Science and Technology (Unist), South Korea, and he worked on BPM project supported by Unist, in 2016. In 2017, he was a Postdoctoral Research with Southern Denmark University, where he was working in urban traffic data analysis. He is currently granted a Postdoctoral Fellowship from the European Research Consortium on Informatics and Mathematics, and he worked with the Norwegian University of Science and Technology, Trondheim, Norway. He is working on topics related to artificial intelligence and data mining, with focus on pattern mining, parallel computing, swarm and evolutionary algorithms. He has published more than 45 international conference and journal papers, two book chapters, and one tutorial paper in the areas of data mining, parallel computing, and artificial intelligence. Current information can be found at https://sites.google.com/site/youcefdjenouri/.

CHUN-HAO CHEN received the Ph.D. degree with major in computer science and information engineering from National Cheng Kung University, Taiwan, in 2008. After that, he joined the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung City, Taiwan, as a Postdoctoral Fellow, in 2009. From 2010 to 2013 and from 2013 to 2017, he served as an Assistant Professor and Associate Professor with the Department of Computer Science and Information Engineering, Tamkang University, Taiwan, respectively, where he is currently a Professor. He has published more than 100 research papers in referred journals and international conferences. He has a wide variety of research interests covering data mining, time series, machine learning, evolutionary algorithms, and fuzzy theory. Research topics cover portfolio selection, trading strategy, business data analysis, and time series pattern discovery.

YUYU ZHANG is currently pursuing the master's degree with the School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China. His research interests include data mining, and privacy-preserving data mining and data security.