IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 28, 2020, accepted May 20, 2020, date of publication May 25, 2020, date of current version June 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997409

Approximation of Probabilistic Maximal Frequent Itemset Mining Over Uncertain Sensed Data

SHENG CHEN[1], LIHAI NIE[1], XIAOYI TAO[2], ZHIYANG LI[2], (Member, IEEE), AND LAIPING ZHAO[1], (Member, IEEE)

[1]College of Intelligence and Computing, Tianjin University, Tianjin 300000, China
[2]School of Information Science and Technology, Dalian Maritime University, Dalian 116000, China

Corresponding authors: Lihai Nie (nlh3392@tju.edu.cn), Zhiyang Li (lizy0205@dlmu.edu.cn), and Laiping Zhao (laiping@tju.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61672379, in part by the Liaoning Provincial Nature Science Foundation of China under Grant 2019-SM-028, in part by the National Key Research and Development Program of China under Grant 2019QY1302, in part by the NSFC-General Technology Basic Research Joint Funds under Grant U1836214, in part by the NSFC under Grant 61872265, and in part by the new Generation of Artificial Intelligence Science and Technology Major Project of Tianjin under Grant 18ZXZNGX00190 and Grant 19ZXZNGX00010.

ABSTRACT Event detection by discovering frequent itemsets is very popular in sensor network communities. However, the recorded data is often a probability rather than a determined value in a really productive environment as sensed data is often affected by noise. In this paper, we study to detect events by finding frequent patterns over probabilistic sensor data under the *Possible World Semantics*. This is technically challenging as probabilistic records can generate an exponential number of possible worlds. Although several efficient algorithms are proposed in the literature, it is still difficult to mine probabilistic maximal frequent items (PMFIs) in large uncertain database due to the high time complexity. To address this issue, we employ approximate idea to further reduce the time complexity from $O(nlogn)$ to $O(n)$ and propose a two-step solution (Aproximation Probabilistic Frequent Itemset-MAX, APFI-MAX in short) including PMFI candidates generation and PMFIs confirmation. We also provide the necessary proofs of our approximation method to make APFI-MAX more solid and convincing. Finally, extensive experiments have been conducted on synthetic and real databases, demonstrating that the proposed APFI-MAX always running faster than state-of-art methods under different parameter settings.

INDEX TERMS Uncertain database, probabilistic maximal frequent patterns, event detection.

I. INTRODUCTION

Event detection or monitoring is a key application for the environmental surveillance in sensor networks [2], [3]. For example, the sensor system should report a "on fire" to the base station to alarm the fire and trigger quick response [4]. One efficient approach for detecting such events is to mine maximal frequent items (FIs). The key idea of those solutions is to calculate the all the frequent itemsets and defines events, whose attributions are beyond the frequent itemset description, as anomalous/dangerous observations, since the less frequent attributions indicate a lower occurrence chance.

However, raw data from devices is often affected by noise in many sensor applications due to the dynamics of physical environments and the possible faults of the seining nodes, such as underground coal mine monitoring [5] and moving object search [6], etc. We name this noise-affected sensed data as probabilistic or uncertain data. An example of uncertain sensed noise data is illustrated in Table 1. Attributes A, B and C represent the events that the amount of gas, dust and water are under normal condition reported by a mine wireless sensor network system. Then corresponding associated probability indicates the possibility of the current records, e.g., the item $A(0.7)$ indicates the probability of the normal gas amount is 0.7.

TABLE 1. An example of uncertain database.

ID	Attributes
T1	A(0.7),B(0.6),C(0.3)
T2	B(0.8),C(0.6)
T3	A(0.3),B(0.8)

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

In general, comparing to exact data, the main difference of uncertain data is that the existence of an item is associated with a likelihood measure or existential probability.

To interpret uncertain data, *Possible World Semantics* (PWS) is often adopted. Even if the PWS is intuitive and useful, the query evaluation and data mining is hard work. There is exponential number of possible worlds in PWS, for instance, Table 3 has $2^3 = 8$ possible worlds, and which is technically challenging for managing the big data. Thus, many algorithms are proposed to address the issue such as Dynamic-Programming Algorithm (DP) [7], Divide-and-Conquer Algorithm (DC) and Top-Down Inheritance of Support Algorithm (TODIS) [8], etc. However, with the size of data becomes larger, these algorithms are still not so efficient and with huge time complexity. In this paper, we study how to extract the FMPIs over uncertain database efficiently and employ approximate idea to further reduce the time complexity.

Our method generally consists of two steps: PMFI candidates generation and PMFIs confirmation. In the first step, To generate smaller candidates set, we improve Apriori algorithm by introducing support expectation of an itemset, which can reduce the candidates set effectively.

Moreover, we give and prove the bound for the support expectation of real PMFIs. In the second step, we present a top-down PMFIs confirmation framework APFI-MAX, which is proved more efficient than state-of-art framework TODIS-MAX [8]. Since probability mass function (pmf) of itemsets must be calculated in TODIS-MAX which is time consuming, we propose an approximation of pmf but high efficient for calculation inspired by the Central Limit Theorem. We also prove the inheritance of our approximation method which is the core of the top-down confirmation framework.

The main contributions of this paper are listed as follows:

- We introduce support expectation of the probabilistic frequent itemsets in candidate generating to reduce the candidate set, thus a novel PMFI generation method is obtained.
- An approximation for probability mass function (pmf) is proposed, which is high efficient in computation and suitable for the top-down PMFIs confirmation framework, due to the proved inheritance.
- We conduct experiments in both real databases and synthetic databases, showing that our algorithm can mine PMFIs with high efficiency, and perform better than the state-of-art approaches in many aspects.

II. RELATED WORK

Many researches focus on the discovery of frequent items (FIs) in exact database. And a set of efficient algorithms, such as the well-known Apriori and FP-growth are proposed. But it is difficult to measure the frequency of a queried itemset in uncertain database, because the support of an itemset in uncertain database is a random variable rather than a fixed occurrence counting. To address this issue, many methods in this area are presented and generally categorized into two groups by their distinct definitions on frequent items in uncertain database.

The methods in first group employ the expectation of the support to measure frequency and define frequent items, referred as the expected support-based frequent itemset [9]–[13]. In this definition, they compute the expectation of support and compare the result with a given threshold like U-Apriori [14], UFP-growth [15], etc. Those algorithms mostly extend from the approaches for mining FIs in exact database.

Point that the approaches based on expected support model may miss the important information of the frequent items in uncertain database. Moreover, they propose a new definition of frequent items in uncertain database, referred as the probabilistic frequent itemset (denote as PFIs) [16]–[24]. This definition introduces the sum of the frequent probabilities in Possible World Semantics (PWS) to capture more inherent information. A set of approaches are subsequently developed to mine PFIs such as DP [7] and DC [8].

However, these approaches are still not so efficient due to the exponential number of possible worlds. Recent research are focused on how to improve the efficiency. As the maximal frequent itemsets can efficiently represent all the frequent itemsets by their subset, some methods propose to only mine maximal frequent itemsets, which can certainly reduce the computing cost and memory size [8]. For example, Li and Zhang *et al.* [25] propose a new tree structure to mine the PMFIs more efficiently with a novel pruning strategy and Bai *et al.* [26] proposes a SelPMiner based on selective partitioning to mine maximal frequent itemset. However, their method in practice is a little time-consuming and can not been applied in very big data limited by the tree structure.

Meanwhile, it is worthy to notice that the above methods need computing probability mass function (*pmf*) whose time complexity is $O(nlogn)$. As the time complexity of the whole mining algorithm is at least $O(nlogn)$, it will be greatly time costing when dealing with very large data. Leung and Hayduk [27] presents a distributed computing method to make it possible to discover the frequent patterns in big data. Another promising solution on this problem is computing *pmf* in an approximation manner.

Several estimation-based methods are developed to settle this problem. In the probabilistic view, when the scale of data is large enough, the support of an itemset can be viewed as random variable following Poisson Binomial distribution [28], [29]. They make use of the properties of Poisson Binomial distribution to extract PFIs rather than getting the precise probability distribution. Although the set of results is imprecise, the algorithms like Poisson Distribution-based UApriori [28] and Normal Distribution-based UH-Mine can be evaluated in $O(n)$, which is much faster than the precise ones.

Moreover, learning-based methods, e.g., transfer learning [30], [31], meta learning [32] and representation learning [33] show great advantages over tradition methods

when dealing with high dimensional data with complex pattern. However, we deem that learning-based methods can hardly applied in mining frequent itemset. One key issue of exploiting learning based methods is that they require inputs with fixed dimension while we can observe that dimension varies over different itemsets. Finally, heuristic and meta-heuristic algorithms [34]–[36] can obviously improve the efficiency of dealing with big data. Our further work will concern the methodology to accelerate process of mining frequent itemsets.

In this paper, we propose a further improved method to reduce the time in computing *pmf* by approximate way with $O(nlogn)$ time complexity. Thus the proposed method can work faster than precise solutions. To further reducing the time for measuring the frequency, we provide a novel candidate algorithm to obtain more precise candidates with a smaller memory costing. Besides, we provide a variance and expectation estimation strategy in frequency measuring process. Owing to those technique, the proposed can spend less than estimation based methods in mining all PFMIs.

III. PROBLEM DEFINITION

A. UNCERTAIN TRANSACTION DATABASE AND POSSIBLE WORLD SEMANTICS

Table 2 shows a simple example of uncertain transaction database.[1] The data is abstracted from the records of environment monitoring in coal mines. The records are usually recorded by wireless monitoring sensor [5] and important to protect safe working conditions in coal mines.

TABLE 2. An example of uncertain database.

ID	Attributes	Pro
T1	ABCD	0.5
T2	BCD	0.6
T3	ABD	0.7

To interpret the uncertain database better, the Possible World Semantics (PWS) is often adopted. Conceptually, an uncertain database can be considered as a set of several (zero is included) exact database. Each line contains zero or more tuples in uncertain database. Table 3 shows a PWS generated from Table 2. $PW5$, for example, represents the occurrence of $T2$ and the absences of $T1$, $T3$ and its existing probability equals $(1-0.5)\times 0.3\times(1-0.6)=0.14$.

B. PROBABILISTIC FREQUENT ITEMSETS

As discussed, there are two definitions of the frequent itemsets on uncertain database. In this paper, we define an itemset as a Probabilistic Frequent Itemset (PFI) by the sum of its frequent probability in PWS [7]. Its formal definition is given in Definition 1.

[1]Table 2 is slightly different from Table 1 in associated probability. In Table 1, each item is associated with an individual probability (attribute uncertainty). Reversely in Table 2, each tuple has an unique probability (tuple uncertainty), in this paper we focus on tuple-uncertainty data.

TABLE 3. A simple example of PWS generated from Table 2.

PWid	Transactions	Sets of Items	Probability
PW0	null	null	0.06
PW1	T1	ABCD	0.06
PW2	T1,T2	ABCD,BCD	0.14
PW3	T1,T3	ABCD,ABD	0.09
PW4	T1,T2,T3	ABCD,BCD,ABD	0.21
PW5	T2	BCD	0.14
PW6	T2,T3	BCD,ABD	0.21
PW7	T3	ABD	0.09

Definition 1: An itemset is frequent if and only if the sum of the frequent probabilities of this itemset in PWS is larger than or equal to the given probabilistic threshold.

$$P(sup(X)\geq minsup)\geq minpro \tag{1}$$

More intuitively, $P(sup(X)\geq minsup)$ measures the probability of the occurrences of item X are larger than a given minimum support *minsup*.

An observation is that if an item X is a Probabilistic Frequent Itemset (PFI), all its subsets are PFIs too.

Definition 2: For a maximal probability frequent itemset X, it will satisfy that if Y is probabilistic frequent and $X\subseteq Y$, then $X=Y$.

TABLE 4. Summary of notations.

notation	meaning
UD	Uncertain Database X
sup(X)	the support of itemset X
minsup	minimum support threshold
minpro	minimum probabilistic frequent threshold
minconf	probabilistic ARs threshold
sup(X)	the support of itemset X
C	Candidates of PMFIs
$Var(X)$	support variance
$E(X)$	support expectation
T	support threshold
T_2	low bound for E(X)

C. TODIS-MAX

Sun *et al.* [8] introduce an efficient algorithm for mining PMFIs named TODIS-MAX. The top-down framework is utilized in TODIS-MAX. Firstly, the candidates are generated by Apriori. Then the candidates are examined from size-1 to size-n. The advantage of TODIS-MAX over traditional method is that the long items which are potentially PFMIs can be quickly yielded in finding candidates. However, the cost to get a probability mass function of an itemset is at least $O(nlogn)$, which makes it hard to do such work in big data. To settle this problem, we make an attempt to estimate the probability mass function and reduce the time complexity in this paper.

IV. ALGORITHM DESIGN

In this section, we will design our PMFIs mining algorithm, namely APFI-MAX. Two steps are included in APFI-MAX, candidates generation and PMFIs conforming. Firstly,

we exploit the bound of expectation in the process of yielding PMFI candidates. Moreover, we provide the proof for the expectation bound. Second, an algorithm is designed to extract all the PMFIs from candidates. In this algorithm, a method is presented to measure the frequency of candidates. Then, we prove the inheritance in the measuring method. At last, an estimation method of expectation and variance is given to further improve the efficiency of APFI-MAX. The advantage of the proposed APFI-MAX is time efficiency since APFI-MAX can reduce time consuming in candidates generation, frequency measurement and computation of variance and expectation. Consequentially, the mining precision is discounted as part of information loses when estimating frequency. Fortunately, the APFI-MAX's impact on accuracy is trivial from experimental results.

A. GENERATE CANDIDATES

The same as the priori work [8], the traditional Apriori algorithm suffer from large candidate set, resulting inefficiency in later processing. This is because Apriori select candidates on the support of itemsets. According to Eq.1, X is a PFI if and only if $P(sup(x) > T1)$ is larger than a given threshold. This property can be utilized to further accelerate the process of candidates generating. However, we find that computing $P(sup(x) > T1)$ is non-trivial in practice.

To this aim, we discover a useful bound for $sup(x)$ for all PMFIs. Specifically, for any PMFI X, we find that there exists a low bound T_2 for $E(X)$. Owing to T_2, it is obvious that we can get a novel criterion for selecting PMFI candidates i.e., for a PMFI candidate, its support expectation must be large than T_2. Here, we reformulate the two judging criteria for PFIs candidates.

Algorithm 1 (CGEB) implements the procedure of generating the candidates for probabilistic maximal frequent itemsets based on expectation bound (based on Theorem1). This method improves Apriori by utilizing the low bound of expectation. In particular, line 8−10 calculates the expectation and variance for X to measure the frequency when we obtian all PMFIs. Lines 12−14 indicates that the scanning process is terminated once the expectation is larger than the low boundof expectation and support is no smaller than support threshold. Line 19 refers that the procedure will stop once there is no candidates generating.

The Bound of the Support Expectation of PMFIs: In the following, we will discuss how to lower bound T_1, and give a formal formulation of the lower bound and upper bound of the support expectation of PMFIs and present some proofs to make it more convincing.

Theorem 1: For an itemset X in uncertain database UD, given the probabilistic support is T_1 and probabilistic frequent threshold equals τ, X is a PMFIs candidate if and only if it meets the following two criteria at the same time,

$$\begin{cases} S(X) > T_1 \\ E(X) > T_2. \end{cases} \tag{2}$$

Algorithm 1 CGEB

```
INPUT: Uncertain Database UD,minsup T1,minpro τ.
OUTPUT: The candidates C for frequent itemsets.
Begin
i = 1
Put all single attributes into L
while True do
  for every item X in L do
    E = 0, Var = 0,count=0
    for every transaction Tj(j = 1 to sizeof UD) in UD
    do
      if X in Tj then
        E=E+pj
        Var=Var+pj ∗ (1 − pj)
        count++
      end if
      if E ≥ lb(E(X))&& count≥ T1 then
        Put (X,E,Var,j) into Ci
        Break
      end if
    end for
  end for
  Put Ci into C
  i++
  Update L according to Ci−1
  if L == null then
    Return C
  end if
end while
End
```

Here, $S(X)$ is the support of X, and T_2 is the low bound for the support expectation of the PMFIs.

Theorem 2: For an itemset X in uncertain database UD, given the probabilistic support is T and probabilistic frequent threshold equals τ, we can get the lower bound and upper bound of the expectation, denoted lb(E(X)) and ub(E(X) as follows.

$$\begin{cases} lb(E(X)) = \dfrac{2T - ln\tau - \sqrt{ln^2\tau - 8tln\tau}}{2} \\ ub(E(X)) = T - ln(1-\tau) + \sqrt{ln^2(1-\tau) - 2tln(1-\tau)} \end{cases} \tag{3}$$

Proof 1: For an itemset X, we use ε to denote its expectation. According to Definition 1, we can get the following inequations.

$$\begin{cases} P_r(sup(X) > T) \le \tau \\ P_r(sup(X) \ge T) > \tau \end{cases} \tag{4}$$

If we set $T = (1-\xi)\varepsilon$, i.e.,where $0 < \xi \le 1$, then $T \ge \varepsilon$, according to the Chernoff Bound,

$$P_r(sup(X) > T) = P_r(sup(X) > (1-\xi)\varepsilon) > 1 - e^{-\frac{\xi^2\varepsilon}{2}}$$

Based on the first inequality in Eq.4, we can get

$$\tau > 1 - e^{-\frac{\xi^2\varepsilon}{2}}$$

Thus,

$$\varepsilon < T - ln(1-\tau) + \sqrt{ln^2(1-\tau) - 2tln(1-\tau)} \quad if\ \varepsilon \geq T \quad (5)$$

If we set $T = (1+\xi)\varepsilon$, i.e., where $\xi > 0$, then $\varepsilon < T$, according to the Chernoff Bound,

$$P_r(X \geq T) = P_r(X \geq (1+\xi)\varepsilon) \leq e^{-\frac{\xi^2\varepsilon}{2+\xi}}$$

Based on the second inequality in Eq.4, we can get

$$\tau < e^{-\frac{\xi^2\varepsilon}{2+\xi}} = e^{-\frac{(T-\varepsilon)^2}{T+\varepsilon}}$$

Thus,

$$\frac{2T - ln\tau - \sqrt{ln^2\tau - 8tln\tau}}{2} < \varepsilon \quad if \quad \varepsilon \leq T \quad (6)$$

From Eq.5 and Eq.6, we can obtain the range of ε is

$$\left(\frac{2T - ln\tau - \sqrt{ln^2\tau - 8tln\tau}}{2}, T - ln(1-\tau) + \sqrt{ln^2(1-\tau) - 2tln(1-T)}\right)$$

Obviously, when given minsup $= T$ and minpro$= \tau$, the expectation of an probabilistic frequent itemset is surely larger than the lb(E(X)).

B. OBTAIN PMFIs

Upon obtaining the PMFIs candidates set C, the next step is to confirm which itemsets in set C are real PMFIs by whether these frequencies are larger than a given threshold or not. However, to compute an itemset's frequency, the probability mass function (*pmf*) of the itemset must be calculated, which is time-consuming due to the exponential number of possible worlds.

To check the PMFIs candidates in a more efficient way, we propose novel way to estimate the frequency inspired by the Central Limit Theorem. In practice, we find the estimation decreases only little checking accuracy but greatly reduces the checking time. Furthermore, we adopt a top-down PMFIs checking framework like TODIS-MAX [8], which means checking the candidates ordered by their length. We also prove that our estimation method has inheritance characteristic. That is to say, when a candidate is confirmed as a PMFI, its subsets in the candidates set C are PMFIs directly and need no checking.

To make it more clear, we give our PMFIs confirmation algorithm using frequency estimation and the top-down framework, named APFI-MAX in the following algorithm 2.

Algorithm 2 (APFI-MAX) shows the procedure to extract the probabilistic maximal frequent itemsets in the candidate

Algorithm 2 APFI-MAX

```
INPUT Uncertain Database UD,minsup T,minpro τ.
OUTPUT A Container RES for PMFI.
Begin
Obtain the candidates C with algorithm 1
Fre_Pre = null and Fre_Cur=null
for i←N to 1 do
  for j← 1 to sizeof(C_i) do
    if C_i(j) ⊂ X&&X ∈ Fre_Pre then
      Put it in Fre_Cur
      Continue
    end if
    Call algorithm 3 to measure its frequency
    if X is frequent then
      Put it into RES
      Put it in Fre_Cur
    end if
  end for
  Fre_Pre = Fre_Cur
  Fre_Cur = null
end for
Return RES
End
```

set C. Like others do, an top-down framework is employed. It is noticed that *Fre_Pre* records all the PFIs (PMFIs and non-PMFIs) in the previous step. Therefore, in line 6−8, if an itemset is a subset of the PFIs in *Fre_Pre*, it is surely frequent. If not, line 9 measures its frequency in an estimation way. Thus, in the following, we will discuss how to estimate the frequency in Section IV-B1 and prove the inheritance characteristic of this kind of estimation in Section IV-B2.

1) FREQUENCY ESTIMATION

To measure the frequency, convolution computation is needed in $O(nlogn)$ to get the probability mass function (*pmf*) for each itemset. This is greatly time costing. A motivation here is that we can find an efficient way to estimate the frequency rather than accurately computing it through pmf.

Given an itemset X, according to Central Limit Theorem, it is known that $\frac{X-E(X)}{\sqrt{Var(X)^2}}$ converges to the standard normal distribution in probability for sufficiently large databases [29]. Inspired by the above fact, we can obtain the following equation:

$$P_r(sup(X) \geq T) = 1 - \phi\left(\frac{T - E(X)}{\sqrt{Var(X)^2}}\right)$$
$$\begin{cases} E(X) = \sum_{n=1}^{N} p_n \\ Var(X)^2 = \sum_{n=1}^{N} p_n(1-p_n) \end{cases} \quad (7)$$

It is not hard to understand the Eq.7. We can consider the every transaction which contains X as a single coin toss. A little difference is the probability in every toss is not

Algorithm 3 FM

INPUT Itemset X,minsup T,minpro τ,Expectation E, Variable σ.
OUTPUT The Boolean Value of the Frequency for X.
Begin
if E $\geq$ ub(E(X)) **then**
  Retrun ture
else
  Retrun the Frequency According to Eq.6
end if
End

the same. Thus in every transaction $X \sim B(1, p)$, the expectation and variance in every transaction are p and $(1-p)p$. As all the transactions in database are disjoint, the expectation and variance of $\sum_{n=1}^{N} X_n$ are $\sum_{n=1}^{N} p_n$ and $\sum_{n=1}^{N} p_n(1-p_n)$. According to the Central Limit Theorem, $\sum_{n=1}^{N} X_n$ follows a normal distribution with expectation equals $\sum_{n=1}^{N} p_n$ and variance equals $\sum_{n=1}^{N} p_n(1-p_n)$.

Algorithm 3 (FM) implements the method of frequency measurement. Line 2−3 shows that an item is frequent if its expectation is larger than the up bound. Line 5 estimates the frequency according to Eq.6. Base on the algorithm 3, we do not need to compute the convolution for the probability mass function (*pmf*) [8] with a little loss of accuracy.

The proposed method reduces the time complexity in mining PMFIs from two perspectives, candidate generation and PMFIs confirmation. In candidate generating stage, assuming that the comparison solution, i.e., Apriori, produces m candidates. Then the proposed CGEB only yield $m * \eta$ candidates owing to the proposed pruning technique, here η varies from 0 to 1 since the part of candidates can be filtered by pruning strategy. Moreover, in PMFIs confirmation stage, the proposed APFI-Max only need $O(n)$ time to confirm a real APFI by approximating the frequency via Central Limit Theorem, here n is the support of the queried candidate. However, TODIS-Max need to compute the probability mass function (*pmf*) to confirm the queried candidate, which consumes $O(nlogn)$ time. As a summary, PMFI-Max can mine all the PMFIs in $O(m * \eta * n)$ time, while the state-of-the-art method need $O(m * n * logn)$ time.

2) INHERITANCE IN FREQUENCY MEASUREMENT

Obviously, APFI-MAX needs the frequency inheritance between subset and superset when we measure the frequency with FM. In particular, when the expectation of subset is larger than up bound, the expectation of superset is surely larger than the up bound. Thus, we only need to discuss the inheritance of Eq. 7.

Theorem 3: If itemset X and Y are two probabilistic frequent items, and Y$\subseteq$ X, then $P_r(sup(X) \geq T) \leq P_r(sup(Y) \geq T)$.

Proof 2: Set minsup = T and minpro = τ, assuming the probability arraies associated with X and Y are $[p_1, p_2, p_3, \ldots, p_n]$, and $[p_1, p_2, p_3, \ldots, p_n, p_{n+1}, \ldots, p_{n+k}]$ respectively. Here $k = sup(Y) - sup(X)$.

Because

$$\begin{cases} P_r(sup(X) \geq T) = 1 - \phi(\dfrac{T - E(X)}{\sqrt{Var(X)^2}}) \\ P_r(sup(Y) \geq T) = 1 - \phi(\dfrac{T - E(Y)}{\sqrt{Var(Y)^2}}), \end{cases}$$

and $\phi(.)$ is an increasing function, it is equivalent to prove

$$\frac{T - E(X)}{\sqrt{Var(X)^2}} > \frac{T - E(Y)}{\sqrt{Var(Y)^2}}. \tag{8}$$

Recall that

$$\begin{cases} E(Y) = E(X) + \sum_{j=1}^{k} p_{n+j} \\ Var(Y) = Var(X) + \sum_{j=1}^{k} (p_{n+j} - p_{n+j}^2) \end{cases} \tag{9}$$

If $T \geq E(X)$, it is obvious that Eq. 8 is correct. In the other case $T < E(X)$, the correctness of Eq. 8 can also be proved by using Eq. 9. The details will not be given here due to the space limitation.

In the following, we give the inheritance in frequency measurement in Theorem 4.

Theorem 4: If itemset X is probabilistic frequent obtained by Eq.6, then for any Y$\subseteq$ X, Y will surely be satisfied to Eq.6.

Proof 3: Theorem 4 can be obtained directly by Theorem 3.

C. EXPECTATION AND VARIANCE ESTIMATION

Given an itemset X, its expectation and variance are needed to estimate its frequency according to Eq. 7. So we will discuss how to calculate the expectation and variance of X in the following.

Here, we give two choices to achieve this. Firstly, since the preceding Apriori algorithm has scanned part of the database, the expectation and variance of itemset X in the scanned database can be calculated during the scanning process. As to the whole database, a reasonable estimation of $E(X)$ and $Var(X)$ are obtained by Eq. 10. This estimation will be used directly in frequency estimation and save a lot of time.

$$\begin{cases} \hat{E(X)} = \dfrac{E(X|D') * |D|}{|D'|} \\ \hat{Var(X)} = \dfrac{Var(X|D') * |D|}{|D'|}. \end{cases} \tag{10}$$

where D' is the scanned database, $|.|$ is the size of the database. However, the scanned part generally can be considered as a sample of the population. It is not accurate if the associated probabilities of the item are skewly distributed. In experiments, we find that the accuracy of expectation and variance will greatly influence the estimation of frequency. So for the applications which require high accuracy, we recommend the second choice. That is scanning the whole database to get precise expectation and variance. Although it is somehow time-consuming, it will increase the accuracy

of obtained PMFIs. Thus, we adopt precise expectation and variance in this paper unless specifically stated. We also give some comparisons of these two strategies in experiments parts.

V. EXPERIMENTAL EVALUATION

In this section, we design and conduct experiments to evaluate the performance of the presented algorithms. To make experiments, we generate uncertain transaction database like other papers by making each tuple in exact database associated with a probability. The algorithms are evaluated on two synthetic datasets (T10I4D320K and T40I10D100K) generated by the IBM synthetic data generator and two real-life datasets (POWERC and KOSARAK) [18]. The detailed characteristics of these datasets are shown in Table 5.

TABLE 5. Certain dataset characteristics.

Dataset	size	average	min	max	#
T25I15D320K	320002	26	1	67	994
T40I10D100K	100000	39	4	77	1000
POWERC	1040002	7	7	7	121
KOSARAK	990002	8	1	2498	41270

Parameter Descriptions: There are three key hyper-parameters that can significantly affect the APFI-max's performance, including data size (k), minsup (T) and minpro (τ). For selecting data size (k), we firstly count the number (denoted as K_{real}) of itemsets in each dataset (K_{real} = 100k, 100k, 1000k, 1000k in T10I4D100K,T40I10D100K, POWERC and KOSARAK, respectively). When evaluating the impacts of data size (FIGURE 1, 3, 6, 7), we select the top 20%, 40%, 60%, 80% and 100% itemsets from each dataset, and then consider them as individual datasets. When evaluating the impacts of minsup (T) (FIGURE 2, 4) and minpro (τ) (FIGURE 5), we fix $k = 100\% * K_{real}$ since the richest information are included with largest data size. For setting minsup (T), we first take the empirical settings recommended by TODIS-Max [8] as default values when varying data size (k) and varying minpro (τ). For evaluating the impact of minsup (T), we increase with two kinds of interval. In particular, if minsup is smaller than 1%, the interval is 0.1% (T10I4D100K and KOSARAK). Otherwise, the increasing interval is 0.5% (T40I10D100K and POWERC). The default minpro is set as 0.6 empirically, recommended by reference [8]. To assess the impact of minpro, we increase the minpro with a fixed interval (0.1) in four datasets.

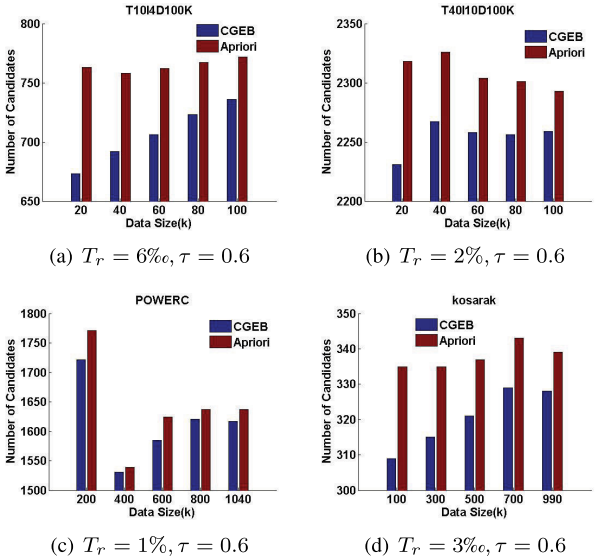
The rest of this section is organized as follows. In subsection V-A, we give the evaluation of the CGEB by comparing with Apriori, a very classical candidates generation algorithm. Then, the advantage of our proposed APFI-Max algorithm over TODIS algorithm is shown in subsection V-B. At last, we evaluate the accuracy of the result set produced by APFI-Max in subsection V-C.

A. CGEB VS. APRIORI

This section provides a performance evaluation of our candidates generation algorithm CGEB under varying data size k and minsup T_r. According to Theorem 1, CGEB will further reduce the candidates size than the classic Apriori algorithm, since CGEB introduces a novel judge criterion. That is to say, CGEB has a greater pruning ability than Apriori algorithm. Experiments in FIGURE.1 and FIGURE.2 also show the similar results under different parameters choices.

1) EFFECT OF DATA SIZE

In FIGURE. 1, we perform CGEB and Apriori algorithms with different data size to evaluate their pruning effects. The relative minimum support is set to a fixed value. As shown in the FIGURE. 1, the pruning effect of these two algorithms grows worse with the increasing of the data size. The reason is that when the data size is large, the expectations of frequent itemsets are mainly concentrated between the low bound and up bound in Eq.3, resulting in the bad effect on pruning. While it is worthy to mention that the proposed CGEB algorithm performs better than Apriori algorithm in all the cases.



(a) $T_r = 6‰, \tau = 0.6$ (b) $T_r = 2\%, \tau = 0.6$

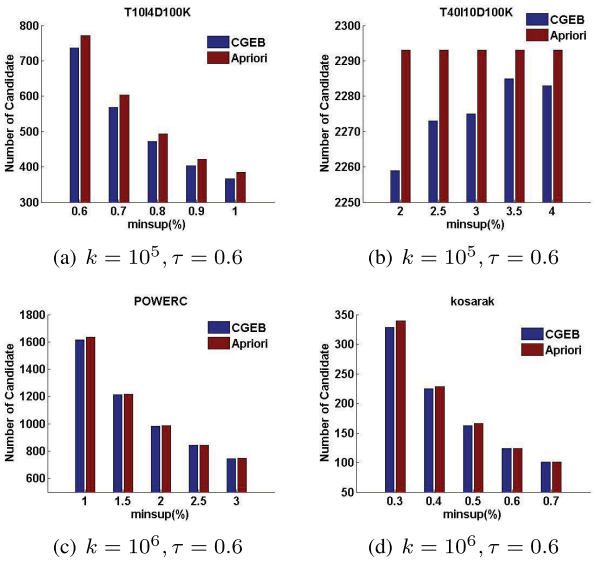
(c) $T_r = 1\%, \tau = 0.6$ (d) $T_r = 3‰, \tau = 0.6$

FIGURE 1. # of candidates vs. datasize.

2) EFFECT OF MINIMUM SUPPORT

In FIGURE.2, we perform CGEB and Apriori algorithms with different relative minsup to evaluate their pruning effects. To make fair comparison, the data size is set to a fixed value. As shown in FIGURE.2, the pruning effect of these two algorithms grows worse with the increasing of the relative minsup. The reason is similar to the above situation. The larger minsup will make expectations of frequent itemsets mainly concentrated between the low bound and up bound in Eq.3, resulting in a bad effect on pruning.

Meanwhile, these pruning effects greatly depend on the database. From the FIGURE.2, The pruning effects over the databases 'T10I4D100K' and 'T40I10D100K' are better than the effects over the other two databases. We find that the itemsets in 'POWERC' and 'kosarak' have a strong inheritance.



(a) $k = 10^5, \tau = 0.6$ (b) $k = 10^5, \tau = 0.6$

(c) $k = 10^6, \tau = 0.6$ (d) $k = 10^6, \tau = 0.6$

FIGURE 2. # of candidates vs. minsup.

In other words, the support difference between subsets and supersets is very little, which may make a bad effect on pruning.

B. APFI-MAX ALGORITHM

This section provides a performance evaluation of APFI-MAX in running time when data size k, relative minsup T_r and minpro τ varies respectively. To make comparisons with TODIS-MAX, a state-of-art for mining PMFIs algorithm, we also give its results in the same cases as APFI-MAX.

1) EFFECT OF DATA SIZE

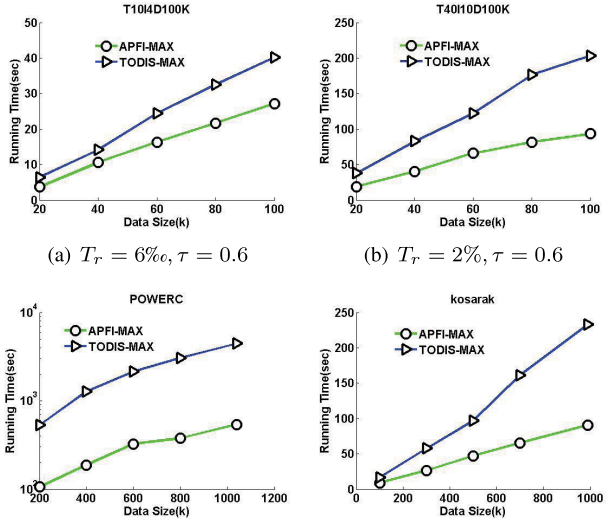
In FIGURE. 3, we perform two algorithms with different data size to evaluate the scalability of the algorithms. Both of the relative minimum support and the minimum probabilistic threshold are set to fixed values. Note that even though the relative minimum support is fixed, the minimum support changes because the data size is different. As shown in the FIGURE.3, the running time of both algorithms increases linearly with the increasing of data size.

From FIGURE. 3, we can see that APFI-MAX always performs better than TODIS-MAX under varying data size in all the four databases. Moreover, it is worth noticing that, the gap between the two lines is growing larger with the increasing of the data size. The reason may be that the time complexity of APFI-MAX is $O(n)$ and lower than TODIS-MAX whose time complexity is $O(nlogn)$. Thus, APFI-MAX algorithm will be much more efficient and scalable than TODIS-MAX when the dataset becomes larger.

2) EFFECT OF MINIMUM SUPPORT

To evaluate the effect of the relative minimum support, the two mining algorithms are conducted with fixed data size and minimum probabilistic threshold. As seen in FIGURE. 4, the running time for two algorithms reduces linearly with the increasing of the relative minimum support, and the proposed APFI-MAX outweighs TODIS-MAX in running time in all the cases in this figure.

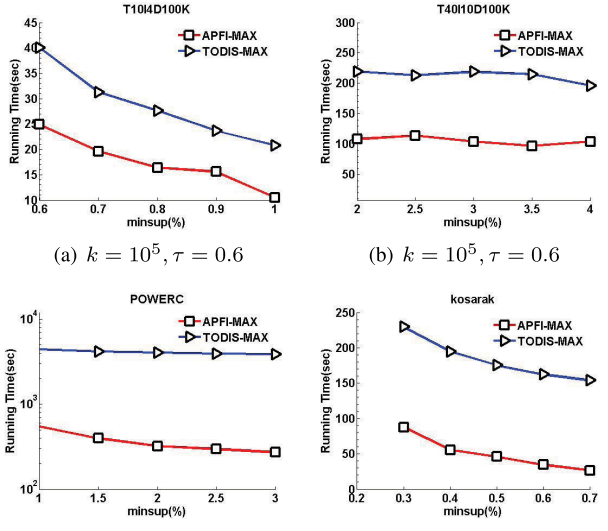
Similarly, the gap between the two lines is getting smaller with the increasing of T_r. The reasons are as follows. On the one hand, the average support of the obtained frequent itemset is bigger when the minsup becomes larger which will make the gap becomes wider as discussed. On the other hand, the number of probabilistic frequent itemsets becomes smaller with the increasing of minsup, which makes the gap become narrower instead. As the gap between the two lines is getting narrow, we can conclude that the influence of minsup weighs more than the influence of frequent itemsets number.



(a) $T_r = 6‰, \tau = 0.6$ (b) $T_r = 2\%, \tau = 0.6$

(c) $T_r = 1\%, \tau = 0.6$ (d) $T_r = 3‰, \tau = 0.6$

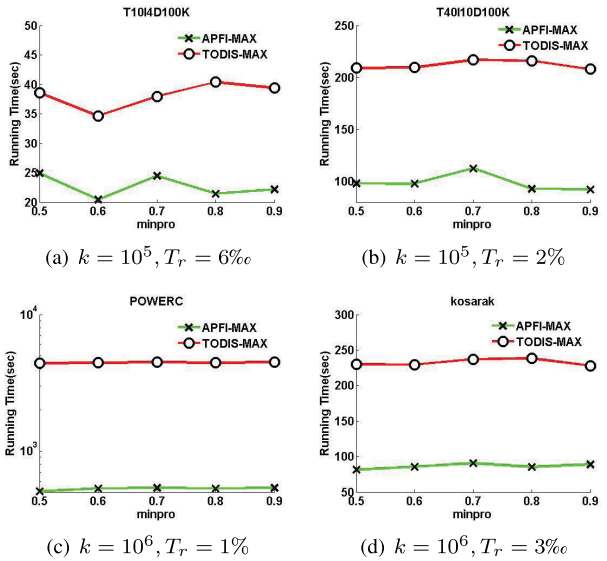
FIGURE 3. Running time vs. datasize.



(a) $k = 10^5, \tau = 0.6$ (b) $k = 10^5, \tau = 0.6$

(c) $k = 10^6, \tau = 0.6$ (d) $k = 10^6, \tau = 0.6$

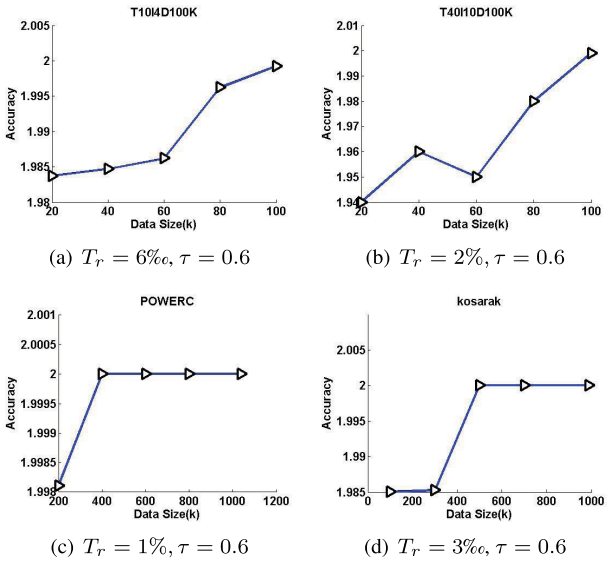
FIGURE 4. Running time vs. minsup.



(a) $k = 10^5, T_r = 6‰$ (b) $k = 10^5, T_r = 2\%$

(c) $k = 10^6, T_r = 1\%$ (d) $k = 10^6, T_r = 3‰$

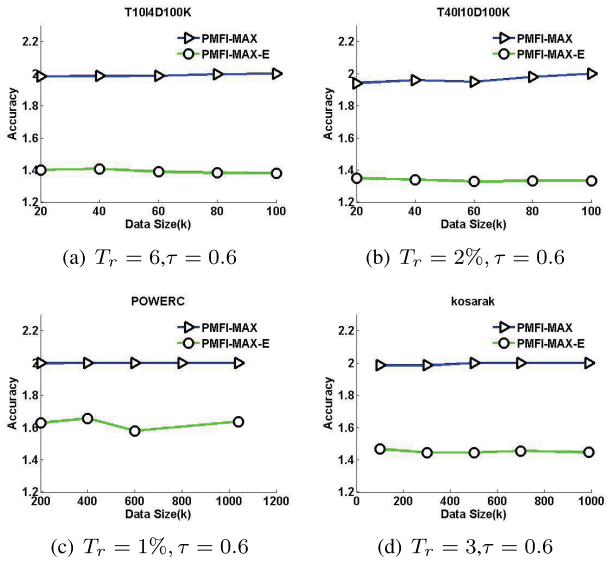
FIGURE 5. Running time vs. minpro.



(a) $T_r = 6‰, \tau = 0.6$ (b) $T_r = 2\%, \tau = 0.6$

(c) $T_r = 1\%, \tau = 0.6$ (d) $T_r = 3‰, \tau = 0.6$

FIGURE 6. Accuracy vs. datasize.



(a) $T_r = 6, \tau = 0.6$ (b) $T_r = 2\%, \tau = 0.6$

(c) $T_r = 1\%, \tau = 0.6$ (d) $T_r = 3, \tau = 0.6$

FIGURE 7. Accuracy vs. datasize.

3) EFFECT OF MINIMUM PROBABILISTIC THRESHOLD

We evaluate the effect of minimum probabilistic threshold with fixed minsup and datasize in different databases. The results are shown in FIGURE. 5. APFI-MAX performs better in all cases. It is easy to understand higher minimum probabilistic threshold will result in less probabilistic frequent itemsets, which shortens the running time.

C. ACCURACY EVALUATION

In this section, since PMFI-MAX is an approximation of true PMFIs set, we want to give an accuracy evaluation of the proposed PMFI-MAX method. To evaluate the accuracy, we design a similarity measure in the following.

Suppose that A is an approximate set of the true PMFI set B. The similarity of A and B (denoted as $S(\frac{A}{B})$) is defined as Eq.11, e.i., the average of *precision* (the term $\frac{A \cap B}{A}$) and *recall* (the term $\frac{A \cap B}{B}$).

$$S(\frac{A}{B}) = \frac{A \cap B}{A} + \frac{A \cap B}{B}. \tag{11}$$

Effect of Data Size: In FIGURE. 6, to evaluate the effect of data size, algorithms are conducted with fixed relative minsup and minpro. It is shown that the accuracy increases with the data size growing larger. The reason may be that as the data size becomes larger, the minsup becomes larger, resulting in a more accurate frequency estimation by the central limit theorem (Eq. 7). Meanwhile, from the FIGURE. 6, it can be seen that no matter how large the data size, the accuracy is pretty high.

As we mentioned in Section IV-C, the expectation and variance can be rapidly estimated by Eq. 10. However, it will reduce the accuracy of obatained PMFIs. To illustrate it better, we give some comparisons on two mining strategies APFI-MAX-E and APFI-MAX in FIGURE. 7. APFI-MAX-E estimates the expectation and variance by Eq. 10, and APFI-MAX estimates the expectation and variance using all the records in the database. Please notice that blue lines in FIGURE. 6 is a local enlarged view of these lines in FIGURE. 7.

From FIGURE. 7, we can see that APFI-MAX undoubtedly outweighs APFI-MAX-E regardless of the size of the database. The reason is that APFI-MAX uses more precise expectation and variance which are important to the accuracy of the detected PMFIs. On the other hand, as the accuracy of APFI-MAX-E is also acceptable, APFI-MAX-E will be a choice if the application is not so focused on the accuracy but the efficiency.

VI. CONCLUSION

In this paper, we study the problem of how to efficiently detect dangerous event by mining probabilistic maximal

frequent itemsets (PMFIs) over noisy sensor data and present a two-step algorithm APFI-MAX which mines PMFIs in an approximation manner. In the first step, according to Chernoff Bound, we present a tight bound of the support expectation of an itemset, which can be utilized to generate more accurate PMFI candidates and largely reduce the size of the candidates set. In the second step, the probability distribution of a candidate is considered as the Normal Distribution and the frequency can be estimated rapidly without computing the probability mass function which is the most time-consuming step in the classic PMFIs mining algorithms. Then, we prove the inheritance of the frequency estimation which can be used to further accelerate the whole mining algorithm. Extensive experiments on different databases and parameters settings are designed and conducted, showing that the presented APFI-MAX always performs better than the state-of-art method TODIS-MAX.

ACKNOWLEDGMENT

This is an extended revision of the early version appeared in [1].

REFERENCES

[1] L. Nie, Z. Li, H. Qi, W. Liu, and W. Qu, "Probabilistic frequent itemsets mining based on expectation bound over uncertain database," in *Proc. 14th Int. Symp. Pervas. Syst., Algorithms Netw. 11th Int. Conf. Frontier Comput. Sci. Technol. 3rd Int. Symp. Creative Comput. (ISPAN-FCST-ISCC)*, Jun. 2017, pp. 14–19.

[2] W. Qian, G. Zhi-Peng, Q. Xue-Song, and W. Xing-Bin, "Frequent itemset mining-based spatial subclustering algorithm," *J. Beijing Univ. Posts Telecommun.*, vol. 38, no. 2, pp. 20–23, Jun. 2015, doi: 10.13190/j.jbupt.2015.s1.005.

[3] W. Xue, Q. Luo, L. Chen, and Y. Liu, "Contour map matching for event detection in sensor networks," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*. New York, NY, USA: Association for Computing Machinery, 2006, pp. 145–156, doi: 10.1145/1142473.1142491.

[4] Y. Lai and J. Xie, "Frequent itemset based event detection in uncertain sensor networks," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Social Comput.*, Aug. 2013, pp. 1037–1043.

[5] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 2, pp. 1–29, Mar. 2009, doi: 10.1145/1498915.1498916.

[6] R. Cheng, S. Prabhakar, and D. V. Kalashnikov, "Querying imprecise data in moving object environments," in *Proc. 19th Int. Conf. Data Eng.*, 2003, pp. 723–725.

[7] T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, and A. Zuefle, "Probabilistic frequent itemset mining in uncertain databases," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 119–128.

[8] L. Sun, R. Cheng, D. W. Cheung, and J. Cheng, "Mining uncertain data with probabilistic guarantees," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: Association for Computing Machinery, 2010, pp. 273–282, doi: 10.1145/1835804.1835841.

[9] C. K.-S. Leung and S. K. Tanbeer, "Fast tree-based mining of frequent itemsets from uncertain data," in *Proc. 17th Int. Conf. Database Syst. Adv. Appl. (DASFAA)*, vol. 7238. Berlin, Germany: Springer-Verlag, 2012, pp. 272–287, doi: 10.1007/978-3-642-29038-1_21.

[10] K. S. Leung and R. K. Mackinnon, "Blimp: A compact tree structure for uncertain frequent pattern mining," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*, 2014, pp. 115–123.

[11] C. K.-S. Leung and D. A. Brajczuk, "Efficient algorithms for the mining of constrained frequent patterns from uncertain data," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 2, pp. 123–130, May 2010, doi: 10.1145/1809400.1809425.

[12] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and V. S. Tseng, "Mining high-utility itemsets with both positive and negative unit profits from uncertain databases," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2017, pp. 434–446.

[13] H. Li, Y. Wang, N. Zhang, and Y. Zhang, "Fuzzy maximal frequent itemset mining over quantitative databases," in *Proc. Asian Conf. Intell. Inf. Database Syst.*, 2017, pp. 476–486.

[14] C.-K. Chui, B. Kao, and E. Hung, "Mining frequent itemsets from uncertain data," in *Advances in Knowledge Discovery and Data Mining*, Z. H. Zhou, H. Li, and Q. Yang, Eds. Berlin, Germany: Springer, 2007, pp. 47–58.

[15] C. K.-S. Leung, M. A. F. Mateo, and D. A. Brajczuk, "A tree-based approach for frequent pattern mining from uncertain data," in *Proc. 12th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining (PAKDD)*. Berlin, Germany: Springer-Verlag, 2008, pp. 653–661.

[16] Q. Zhang, F. Li, and K. Yi, "Finding frequent items in probabilistic data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2008, pp. 819–832.

[17] L. Wang, D. W. Cheung, R. Cheng, S. D. Lee, and X. Yang, "Efficient mining of frequent item sets on large uncertain databases," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 12, pp. 2170–2183, Dec. 2012. [Online]. Available: https://doi.org/10.1109/TKDE.2011.165

[18] Y. Tong, L. Chen, Y. Cheng, and P. S. Yu, "Mining frequent itemsets over uncertain databases," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1650–1661, Jul. 2012, doi: 10.14778/2350229.2350277.

[19] P. Tang and E. A. Peterson, "Mining probabilistic frequent closed itemsets in uncertain databases," in *Proc. 49th Southeast Regional Conf.*, 2011, pp. 86–91.

[20] E. A. Peterson and P. Tang, "Fast approximation of probabilistic frequent closed itemsets," in *Proc. 50th Annu. Southeast Regional Conf.*, 2012, pp. 214–219.

[21] C. Liu, L. Chen, and C. Zhang, "Mining probabilistic representative frequent patterns from uncertain data," in *Proc. SIAM Int. Conf. Data Mining*, May 2013, pp. 1–9.

[22] W. Tong, C. K. Leung, D. Liu, and J. Yu, "Probabilistic frequent pattern mining by PUH-mine," in *Proc. Asia–Pacific Web Conf.*, 2015, pp. 768–780.

[23] A. Farhat, M. S. Gouider, and L. B. Said, "New algorithm for frequent itemsets mining from evidential data streams," *Procedia Comput. Sci.*, vol. 96, pp. 645–653, Oct. 2016.

[24] Y. Tong, X. Zhang, and L. Chen, "Tracking frequent items over distributed probabilistic data," *World Wide Web*, vol. 19, no. 4, pp. 579–604, Jul. 2016.

[25] H. Li and N. Zhang, "Probabilistic maximal frequent itemset mining over uncertain databases," in *Proc. Int. Conf. Database Syst. Adv. Appl. (DASFAA)*, 2016, pp. 149–163.

[26] A. Bai, M. Dhabu, V. Jagtap, and P. S. Deshpande, "An efficient approach based on selective partitioning for maximal frequent itemsets mining," *Sādhanā*, vol. 44, no. 8, p. 183, Aug. 2019.

[27] K. S. Leung and Y. Hayduk, "Mining frequent patterns from uncertain data with mapreduce for big data analytics," in *Proc. Int. Conf. Database Syst. Adv. Appl. (DASFAA)*, 2013, pp. 440–455.

[28] L. Wang, R. Cheng, S. D. Lee, and D. Cheung, "Accelerating probabilistic frequent itemset mining: A model-based approach," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 429–438.

[29] T. Calders, C. Garboni, and B. Goethals, "Approximation of frequentness probability of itemsets in uncertain data," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 749–754.

[30] G. Wang, J. Qiao, J. Bi, W. Li, and M. Zhou, "TL-GDBN: Growing deep belief network with transfer learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 874–885, Apr. 2019.

[31] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, no. C, pp. 228–235, Apr. 2017, doi: 10.1016/j.neucom.2017.01.018.

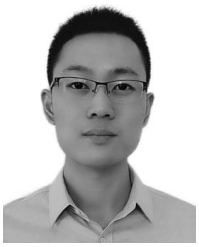
[32] A. S. Qureshi, A. Khan, A. Zameer, and A. Usman, "Wind power prediction using deep neural network based meta regression and transfer learning," *Appl. Soft Comput.*, vol. 58, pp. 742–755, Sep. 2017.

[33] W. Dong, Y. Wang, and M. Zhou, "A latent space-based estimation of distribution algorithm for large-scale global optimization," *Soft Comput.*, vol. 23, no. 13, pp. 4593–4615, Jul. 2019.

[34] X. Guo, M. Zhou, S. Liu, and L. Qi, "Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints," *IEEE Trans. Cybern.*, pp. 1–11, 2019.

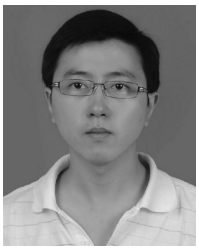
[35] J. Bi, H. Yuan, M. Zhou, and Q. Liu, "Time-dependent cloud workload forecasting via multi-task learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2401–2406, Jul. 2019.
[36] Z. Zhao, S. Liu, M. Zhou, X. Guo, and L. Qi, "Decomposition method for new single-machine scheduling problems from steel production systems," *IEEE Trans. Autom. Sci. Eng.*, early access, Dec. 30, 2019, doi: 10.1109/TASE.2019.2953669.

SHENG CHEN received the bachelor's degree from Dalian Maritime University, in 2011, and the master's degree from Dalian University of Technology, in 2017. He is currently pursuing the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, China. His research interests include data center networks, edge computing, wireless sensing, and indoor localization.

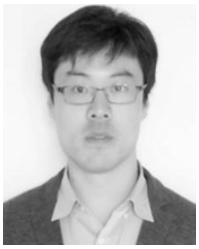


LIHAI NIE received the bachelor's and master's degrees from Dalian Maritime University, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, China. His research interests include cloud computing and machine learning.

XIAOYI TAO received the B.S. and Ph.D. degrees from the School of Software Engineering, Dalian University of Technology, China, in 2011 and 2019, respectively. She is currently a Lecturer with the School of Information Science and Technology, Dalian Maritime University, China. Her research interests include data center networks, SDN networks, and cloud computing.



ZHIYANG LI (Member, IEEE) received the Ph.D. degree in computation mathematics from the Dalian University of Technology, China, in 2011. He is currently an Associate Professor with the Information Science and Technology College, Dalian Maritime University, China. His research interests include computer vision, cloud computing, and data mining. He has published more than 50 articles in international journals and conferences.



LAIPING ZHAO (Member, IEEE) received the B.S. and M.S. degrees from the Dalian University of Technology, China, in 2007 and 2009, respectively, and the Ph.D. degree from the Department of Informatics, Kyushu University, Japan, in 2012. He is currently an Associate Professor with the School of Computer Software, Tianjin University, China. His research interests include cloud computing and software defined networking.