IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received July 8, 2020, accepted July 26, 2020, date of publication July 30, 2020, date of current version August 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013157

DPARM: Differentially Private Association Rules Mining

YAO-TUNG TSOU[1], (Member, IEEE), HAO ZHEN[2], XIYU JIANG[2], YENNUN HUANG[3], (Fellow, IEEE), AND SY-YEN KUO[2]

[1]Department of Communications Engineering, Feng Chia University, Taichung 407, Taiwan
[2]Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan
[3]Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan

Corresponding author: Yao-Tung Tsou (yaodong1014@gmail.com)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2221-E-035-020-MY3 and Grant MOST 109-2923-E-035-001-MY3.

ABSTRACT Association analysis is critical in data analysis performed to find all co-occurrence relationships (*i.e.*, frequent itemsets or confident association rules) from the transactional dataset. An association rule can improve the ability of users to discover patterns and develop corresponding strategies. The data analysis process can be summarized as a set of queries, where each query is a real-valued function of the dataset. However, unless restrictions and protections are implemented, accessing the dataset to answer the queries may lead to the disclosure of the private information of individuals. In this paper, we propose an original differentially private association rules mining (DPARM) algorithm, which uses multiple support thresholds to reduce the number of candidate itemsets while reflecting the real nature of the items and uses random truncation and uniform partition to reduce the dimensionality of the dataset. Both of these elaborated approaches can aid in reducing the sensitivity of the queries, and this dramatically reduces the scale of the required noise and improves the utility of the mining results. We significantly stabilize the noise scale by adaptively allocating the privacy levels and bound the overall privacy loss. Through a series of experiments, we prove that our DPARM algorithm outperforms the literature in the accuracy of data mining while satisfying differential privacy. To the best of our knowledge, our work is the first DPARM algorithm to adopt multiple support thresholds while using a set of elaborated approaches to bound the overall privacy loss of the mining process.

INDEX TERMS Privacy-preserving data analysis, differential privacy, association analysis, association rules mining, frequent itemset mining.

I. INTRODUCTION

With the development of information technology and the popularity of smart devices, the volume of data generated by humans has substantially grown in scale. International Data Corporation [30] has forecasts that by 2025, the global datasphere will grow to 163 ZB, which is approximately 10 times the 16.1 ZB of data generated in 2016. As Professor Viktor Mayer-Schönberger [25] opined, "Data's true value is like an iceberg floating in the ocean. Only a tiny part of it is visible at first sight, while much of it is hidden beneath the surface." Data analysis techniques, such as data mining and machine learning, are powerful tools for exploring the iceberg. Decision-making is becoming increasingly automated and thus no longer relies on the subjective judgments of individuals.

Data mining and machine learning are two common data analysis techniques, but their application scenarios differ somewhat: Data mining emphasizes the discovery of useful knowledge, whereas machine learning focuses on predicting unknown entities on the basis of associations. Associations, which are co-occurrence relationships in transactional datasets, constitute an essential class of laws in data. Association analysis is a fundamental data mining task that was first applied to analyze the contents of shopping baskets in a market to discover purchasing habits. One of the most famous examples is "diapers and beer." Any relationship between these two items is not readily apparent, but after analysis of associations, the market manager determined that shoppers who buy diapers are more likely to buy beer. On the basis of

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

this result, the market manager could adopt effective commercial strategies; for example, the market manager could publish promotional advertisements for diapers while increasing the price of beer. Then, when customers come to buy cheap diapers, they may not notice the increased in beer prices or may not want to go to another market to find cheaper beer, and thus, the market would profits.

Associations can more specifically be divided into frequent itemsets and association rules. By definition, the frequency with which an itemset appears in a transactional dataset is called support. If the support is larger than the minimum support, then the item set is called a frequent itemset. An association rule is represented as "{coronavirus disease 2019 (COVID-19)}→{fever, cough}," which means that if one person gets the COVID-19, then the person "possibly" has fever and cough. The support of the association rule is equal to the support of the union {COVID-19, fever, cough}. The possibility is portrayed through confidence concerning the proportion of all individuals' information containing {COVID-19} that also contain {fever, cough}. The proportion is equal to the ratio of the support of the association rule to the support of {COVID-19}. If the confidence is higher than the minimum confidence, then the association rule is called a confident association rule.

Data analysis has made business more effective and is changing many aspects of personal life. However, it is likely that as the volume of data concerning all individuals continues to increase, the documentation of individual personalities and behaviors will become increasingly detailed, an outcome which entails risk to individual privacy if due caution is not exercised. Members of the general public acting as data providers thus face a tradeoff: they can enjoy more personalized services and a higher quality of life, but excessive data collection and misuse threaten their privacy. Among companies acting as data collectors and processors, those that can collect massive amounts of data and extract more information from them can gain a competitive advantage, but the more data they hold and the more analysis they perform, the greater is their data protection responsibility; otherwise, they may violate regulations such as the European Union's General Data Protection Regulation [9].

Thus far, privacy-preserving data analysis has been receiving increased attention by those pursuing the goal of protecting individual privacy while maintaining data utility. In the data analysis domain, there is an urgent need for a reasonable and feasible definition of privacy as well as a need to develop mechanisms satisfying this definition and address various types of attack strategies (for different adversarial targets).

An intuitively appealing privacy protection measure is data anonymization, which specifies the form of datasets to be released. However, data anonymization is considered a weak privacy definition because an adversary who knows auxiliary information about certain individuals in the dataset can initiate a re-identification attack. For instance, the anonymized Netflix dataset can be linked with the Internet Movie Database (IMDb), and accordingly, almost all subscribers in the Netflix dataset can be uniquely identified [28].

Many attempts have been made to refine anonymization, such as k-anonymity [31] and its variants l-diversity [21] and t-closeness [14]. An extreme approach is to release only exact statistics for the dataset. However, an adversary can exploit the relationships between certain pairs of statistics to obtain private information of certain individuals through a "difference attack." For example, the difference between the answers to "How many people have cancer?" and "How many people have cancer and are not Alice?" can reveal A's cancer status. Note that a query audit is useless because such malicious query pairs are usually difficult to discover.

In general, all the aforementioned privacy definitions are syntactic (*i.e.*, they specify how a privacy-preserving output should look) and do not reflect the meaning of privacy protection with respect to individuals. By contrast, semantic security entails a semantic definition of privacy.

Semantic security compares the extent of change that occurs when the adversary's inferences about an individual made before and after statistics are released, whereas differential privacy compares the extent of change that occurs when the adversary's inferences about an individual made when statistics containing and excluding the individual's data are released. Differential privacy is a mathematical definition tailored to data analysis and equipped with a metric of privacy loss. The output distribution of a privacy-preserving data analysis remains "stable" under any possible change to a single individual's data. In particular, differential privacy can (1) provide meaningful privacy protection against adversaries with arbitrary auxiliary information and arbitrary attack strategies, (2) bind the privacy risk of an individual participating in data analysis, and (3) create a complex system using several simple building blocks. Thus far, differential privacy is considered the strongest privacy definition and has become the formal privacy standard in both academia and industry.

Many differentially private algorithms for frequent itemset mining and confident association rule mining have been proposed, such as the score perturbation-based FPM algorithm [3], the PrivBasis algorithm [18], the DiffFIM algorithm [41], and the HCRMine algorithm [26], [27]. Most of these are based on the nonprivate Apriori algorithm [2]. However, these algorithms remain subject to the detrimental effect of large or unstable noise. Large noise occurs when high-dimensionality datasets are analyzed or too many candidate itemsets are generated in using a single support threshold. Unstable noise results from improper allocation of the privacy budget for each substep of the mining process. Both compromise the utility of the mining results.

The key element that makes association rule mining practical is the support threshold, used to prune the search space and limit the number of rules generated. However, using only a single support threshold implicitly assumes that all items in the data are of the same nature, have similar frequencies in the database, or both. This is often not the case in real-life applications [13]. In many applications, some items appear highly frequently in the data, whereas others rarely appear.

How to set one appropriate support threshold for all items is a difficult question. In the retailing business, customers buy some items very frequently but other items very rarely. Usually, the necessities, consumables and low-price products are bought frequently, while the luxury goods, electric appliance and high-price products infrequently. If the support threshold is set too high, we might consider those luxury goods to be less important items; however, those luxury goods represent hundreds if not thousands of times more revenue to the store than the commonly purchased low-price products do. By contrast, if the threshold is too low, the computational cost becomes expensive, and the mining result causes a combinational explosion; the result is many meaningless itemsets. The same difficulty may occur when we are about to mine medical records. Mining medical records is a vital real-life application because it can reveal which symptoms are related to which diseases. However, many critical symptoms and diseases are infrequent in medical records. For example, influenza (flu) occurs much more frequently than does the recent coronavirus disease 2019 (COVID-19), and both have symptoms of fever and persistent cough. If the value of the support threshold is set high, the rule "flu→fever, cough" can be found, but we would never find the rule "COVID-19→fever, cough." To find this COVID-19 rule, we must set the value of the support threshold very low. However, this causes many meaningless rules to be found at the same time.

The dilemma described in the preceding paragraph is the "rare item problem." In real-life applications, some items tend to naturally have more weights than other items. Researchers have addressed this problem by allowing users to use "multiple support thresholds." In brief, rare itemset mining is a more advanced setting of frequent itemset mining that allows user to apply different thresholds to each item.

These efforts motivated us to develop a novel association rules algorithm to guarantee both privacy and utility by using multiple support thresholds. In this article, we propose the DPARM algorithm, which meets these challenges through a group of well-elaborated techniques, namely the use of multiple support thresholds, sensitivity control, and adaptive allocation of privacy budgets. In particular, our major contributions are as follows:

- We originally use multiple support thresholds and assign the support threshold for each distinct item in both a data-driven and differentially private manner.
- We dramatically reduce the noise scale through sensitivity control by lowering the dimensionality of the transactional dataset and decreasing the number of candidate itemsets.
- We significantly stabilize the noise scale through adaptive allocation of privacy budgets, which changes the privacy budget in proportion to the query sensitivity.
- We perform formal privacy analysis and bound the overall privacy loss through ex post differential privacy. We also verify utility and demonstrate that our method outperforms the literature through a series of experiments.

The remainder of the article is organized as follows. Section II reviews the background knowledge concerning association rule mining and differential privacy. Section III presents the related work. Section IV formalizes the computational model and key challenges. Section V describes the DPARM algorithm in detail. Section VI proves that the DPARM algorithm satisfies differential privacy. Section VII presents the results of experiments in which the DPARM algorithm is applied to five real-world datasets. Section VIII presents the conclusion of the work.

II. PRELIMINARIES

In this section, we review the basic knowledge of association rule mining and differential privacy.

A. ASSOCIATION RULE MINING

Let $\mathbf{T} = \{t_1, \ldots, t_n\}$ be a transactional dataset where each transaction t_j is a subset of an item universe $I = \{i_1, \ldots, i_m\}$, *i.e.*, $t_j \subseteq I$. An association rule r is an implication of the form $X \rightarrow Y$, in which the itemsets $X, Y \subseteq I$ are called antecedent and consequent, and $X \cap Y = \varnothing$. The support of r is the proportion of transactions containing $X \cup Y$, which is the same as the support of the itemset $X \cup Y$. The confidence of r is the proportion of transactions containing X which also contain Y.

When mining association rules, we must first specify the support and confidence thresholds, which are the minimum support and the minimum confidence that any output association rules must reach. In the traditional setting [2], there is only one minimum support *minsup* and minimum confidence *minconf*, and these are selected in advance. By contrast, in the advanced setting [13], there are multiple minimum supports. Each distinct item is specified a minimum item support (MIS), and the minimum support of an association rule is the lowest MIS among the items in it. A single minimum confidence *minconf* is used in both settings. We apply the advanced setting because it is superior in simulating the real world and entails a less number of candidate itemsets. Without losing generality, we define the minimum support in terms of the itemset instead of the association rule.

Definition 1 (Minimum Support [13]): For $\rho, \lambda \in [0, 1]$ and an item j with support $sup_{\mathbf{T}}(j)$, the MIS of j is

$$MIS(j) = \max\{\rho \times sup_{\mathbf{T}}(j), \lambda\}. \quad (1)$$

For an itemset $X \subseteq I$, the minimum support of X is $\min_{j \in X}\{MIS(j)\}$.

The constants $\rho, \lambda \in [0, 1]$ are called the support relevance and the lowest allowed MIS, which control the dependence of MIS on support and the lower bound of MIS, respectively. Note that when $\rho = 0$, all MISs converge to λ, equivalent to the single *minsup* in the traditional setting.

In general, we want strong association rules both frequent and confident. As in the Apriori algorithm, we can generate confident association rules from frequent itemsets. The frequent itemsets and the confident association rules are defined as follows.

Definition 2 (Frequent Itemset [41]): For an itemset $X \subseteq I$ and a dataset $\mathbf{T}$ with n transactions, the support of X is

$$sup_{\mathbf{T}}(X) = \frac{1}{n}|\{t : X \subseteq t\}|. \tag{2}$$

X is a frequent itemset if and only if its support is higher than the minimum support, i.e., $sup_{\mathbf{T}}(X) \geq \min_{j \in X}\{MIS(j)\}$.

Definition 3 (Confident Association Rule [13]): For an association rule $r : X \to Y$ and a dataset $\mathbf{T}$ with n transactions, the support and confidence of r is

$$sup_{\mathbf{T}}(r) = sup_{\mathbf{T}}(X \cup Y), \tag{3}$$

$$conf_{\mathbf{T}}(r) = \frac{sup_{\mathbf{T}}(r)}{sup_{\mathbf{T}}(X)}. \tag{4}$$

r is a confident association rule if and only if its confidence is higher than the minimum confidence minconf, i.e., $conf_{\mathbf{T}}(r) \geq minconf$.

B. DIFFERENTIAL PRIVACY

Let $D = \{d_1, \ldots, d_n\}$ be a dataset where each d_j denotes the data of a single individual and is an element of the data universe $\mathcal{D}$. Neighbors are a pair of datasets that differ only with respect to the data of a single individual.

Lemma 1 (Neighbors): Datasets D and D' are neighbors if D' can be obtained from D by removing or adding the data of a single individual.

These neighbors are also called unbounded neighbors, in contrast to bounded neighbors [12]. Differential privacy that is constrained only by unbounded neighbors obtains superior properties [19].

Definition 4 (Differential Privacy [5]): A randomized mechanism M satisfies ε-differential privacy if for any neighbors D and D' and for any potential output $\hat{q} \in \mathrm{range}(M)$ and then we have

$$\mathbb{P}[M(D, q) = \hat{q}] \leq e^{\varepsilon} \cdot \mathbb{P}[M(D', q) = \hat{q}]. \tag{5}$$

The query q is a real-valued function of D. The constant $\varepsilon > 0$ is called the privacy budget, which bounds the privacy loss. Note that M must be randomized, and randomness might come from adding noise (or resampling, *etc*.). When answering q, $M(D, q)$ [abbreviated as $M(D)$ when q is fixed] can be seen as a random variable, and we need to calibrate the noise scale according to the global sensitivity of q.

Definition 5 (Global Sensitivity [5]): For any neighbors D and D', the global sensitivity of a query $q \in \mathbb{R}$ is

$$\Delta q = \max_{D,D'} |q(D) - q(D')|. \tag{6}$$

The Laplace mechanism perturbs q by adding the noise generated from the Laplace distribution. The probability density of the Laplace distribution is $Lap(z|b) = \frac{1}{2b}\exp\{-\frac{|z|}{b}\}$, where the scale parameter b is related to both the global sensitivity Δq and privacy budget ε.

Definition 6 (Laplace Mechanism [5]): For a transactional dataset D, a query $q \in \mathbb{R}$, and a privacy budget ε, the Laplace mechanism M_{Lap} satisfies ε-differential privacy:

$$M_{\mathrm{Lap}}(D) = q(D) + Z, \tag{7}$$

where Z is drawn from $Lap(\frac{\Delta q}{\varepsilon})$.

Definitions 5 and 6 can be extended to the vector form. For $\mathbf{q} = [q_1, q_2 \ldots, q_J]$, the global sensitivity $\Delta\mathbf{q} = \max_{D,D'} ||\mathbf{q}(D) - \mathbf{q}(D')||_1 = \max_{D,D'} \sum_{j=1}^{J} |q_j(D) - q_j(D')|$. Similarly, the Laplace mechanism $M_{\mathrm{Lap}}(D) = \mathbf{q}(D) + \mathbf{Z} = [q_1 + Z_1, q_2 + Z_2, \ldots, q_J + Z_J]$, where Z_j is $i.i.d.$ drawn from $Lap(\frac{\Delta\mathbf{q}}{\varepsilon})$.

A mechanism that analyzes the differentially private results or consists of several differentially private building blocks still satisfies differential privacy.

Definition 7 (Closure Under Postprocessing [7]): For any algorithm A and a randomized mechanism M, if M satisfies ε-differential privacy, then $A \circ M$ satisfies ε-differential privacy.

Definition 8 (Sequential Composition [7]): For D and a sequential mechanism $M_{\mathrm{seq}}(D) = [M_1(D), M_2(D), \ldots, M_s(D)]$, if each M_j $(1 \leq j \leq s)$ satisfies ε_j-differential privacy, then M_{seq} satisfies $(\sum_{j=1}^{s} \varepsilon_j)$-differential privacy.

Definition 9 (Parallel Composition [7]): For a random partition $\{D_1, D_2, \ldots, D_p\}$ of D and a parallel mechanism $M_{\mathrm{par}}(D) = [M_1(D_1), M_2(D_2), \ldots, M_K(D_p)]$, if each M_j $(1 \leq j \leq p)$ satisfies ε_j-differential privacy, then M_{par} satisfies $(\max_{j=1}^{p} \varepsilon_j)$-differential privacy.

Definition 10 (Privacy Loss [5]): The privacy loss of a randomized mechanism M for any neighbors D and D' and for any potential output $\hat{q} \in \mathrm{range}(M)$ is

$$\mathrm{Loss}(\hat{q}) = \max_{D,D'} \log \frac{\mathbb{P}[M(D) = \hat{q}]}{\mathbb{P}[M(D') = \hat{q}]} \tag{8}$$

However, commonly, the privacy budgets cannot be fixed a priori. For instance, in adaptive data analysis, the privacy budgets may be a function of the outputs of previous computations. Nevertheless, we can obtain an a posteriori privacy guarantee provided by ex post differential privacy.

Definition 11 (Ex Post Differential Privacy [7]): A randomized mechanism M satisfies $\mathcal{E}(\hat{q})$-ex post differential privacy if for any potential output $\hat{q} \in \mathrm{range}(M)$ and then we have $\mathrm{Loss}(\hat{q}) \leq \mathcal{E}(\hat{q})$.

Note that ex post differential privacy is equivalent to differential privacy once the outputs of the mechanism are known. In other words, if $\mathcal{E}(\hat{q}) \leq \varepsilon$ for all $\hat{q}$, then M is also differentially private.

III. RELATED WORK

To ensure data security and user privacy, privacy-preserving data mining (PPDM) has emerged as an increasingly important concern. Approaches based on sanitization and anonymity, such as those in [23], [31], [37], and [35], were proposed to generate a protected dataset for general purposes of data mining using privacy-preserving technologies, such as k-anonymization, differential privacy, and cryptograph-based technologies. These methods are data-driven solutions and used to generate protected dataset for statistic or machine learning analysis. Nevertheless, the aforementioned workarounds were not designed for association rule mining of transactional datasets.

The problem of association rule mining was first identified in [1], which received widespread attention. Several well-known algorithms for association rule mining have been developed, such as Apriori [2], Eclat [39], FP-Growth [10], and NR-HARs [22] algorithms. Some well-konwn studies [15], [16], [33], [36], [38] have attempted to elaborate efficient frequent itemset/pattern mining algorithms for transactional datasets or the data from the Internet of Things. These works focus on designing efficient algorithms for mining frequent itemsets/patterns based on novel data structures and mining techniques while maintaining high utility. However, these methods do not focus on guarantee of data security and user privacy. These methods of data mining without providing protection of data security and privacy are orthogonal to our work. We are more interested in PPDM.

As a specific type of PPDM, privacy-preserving association rule mining (PPARM) has been widely researched in several areas, such as market basket analysis, e-health, and wireless sensor networks. The purpose of PPARM is to identify relationships of interest among sets of items in a transaction database while protecting the data security and user privacy. Reference [42] classified workarounds related to PPARM into six categories of methods, namely those based on data obfuscation, heuristics, reconstruction, metaheuristics, cryptographs, and secure multiparty computation. However, knowing how to distinguish between sensitive and insensitive information relies heavily on experience. Because adversaries are assumed to have any auxiliary information, meeting the requirements of privacy protection under such a division is difficult.

Differential privacy [6] is a powerful semantic privacy model that transforms a dataset through bounding the probability of an adversary learning whether a particular individual is present in the dataset. Differential privacy has become the de facto standard in the privacy-preserving release and processing of sensitive data and has been considered in the data mining domain.

Our work focuses on mining association rules in a differentially private manner. The existing DPARM algorithms can be roughly divided into two categories, one of which needs to mine frequent itemsets as a fundamental step in association rule mining; examples include the DiffFIM [41], PrivBasis [18], and PrivSuper [34] algorithms. The other category mines association rules directly, and examples of such algorithms are the HCRMine algorithm [26] and its variants the HCRBins and HCRPlus algorithmx [27]. These algorithms are discussed in further detail in the subsequent paragraphs.

The DiffFIM algorithm [41] smartly truncates $\mathbf{T}$ by a truncation cardinality θ once $\mathbf{C}_k$ is generated, which limits the maximum cardinality of transactions from m to θ. For each transaction $t \in \mathbf{T}$, the smart truncation first selects c, where $c \in \mathbf{C}_k$ and $c \subseteq t$ with the highest weight, and then updates the weight of the remaining $c \subseteq t$ and continues to select c unless θ is reached. The initial weight of c is defined as the summation of the noisy support of all its $(k-1)$-subsets, and the updated increment is the product of the average weight and number of items that has already been selected. The DiffFIM algorithm tries to estimate K through binary search and averagely separates the overall privacy budget into K parts. It first computes $\mathbf{y} = [y_1, \ldots, y_m]$, where y_j is the maximum support among the itemsets with cardinality k. It then finds the $j^\star$ such that $y_{j^\star}$ is the smallest integer that exceeds the support threshold λ through differentially private binary search because $\mathbf{y}$ is nonincreasing. However, it cannot offset the effect of $\Delta\mathbf{q}$ on noise because the privacy budgets are allocated averagely.

The PrivBasis algorithm [18] projects $\mathbf{T}$ onto a λ-basis set $\mathbf{B} = \{B_1, B_2, \ldots, B_w\}$, where $B_j \subseteq I$ and $|B_j| < m$, and any itemset with support higher that λ is a subset of some basis B_j. For each B_j, such projection means removing all the items that are not in B_j from every transaction, which is equivalent to generating $2^{|B_j|} - 1$ candidate itemsets. The PrivBasis algorithm specifies K and only mines the most K frequent itemsets rather than all frequent itemsets above a single support threshold. Some of overall privacy budget is used to construct the λ-basis set $\mathbf{B} = \{B_1, B_2, \ldots, B_w\}$, and the other is averagely separated into w parts for computing the noisy support of each B_j. Nevertheless, it works well only when the *minsup* is very large.

The PrivSuper algorithm [34] applies the concept of maximal frequent itemsets to detect the final results; consequently, the corresponding subsets are added into the results without privacy budget consumption. Furthermore, the sequence exponential mechanism that combines the Laplace mechanism and exponential mechanism is designed to extend the current itemset. Specifically, the sequence exponential mechanism is applied to extend a given itemset S by adding one more item x to S that maximizes the frequency of the resulting itemset $S' = S \cup \{x\}$. The sequence exponential mechanism consumes zero privacy budget if the resulting itemset S' is no longer frequent; otherwise, the sequence exponential mechanism consumes the same amount of budget as the plain exponential mechanism.

Rather than truncation or projection, the HCRMine algorithm [26] uses a novel representation of the association rule space and voids to compute the support of $c \in \mathbf{C}_k$ for $k > 1$. It only mines the set $\mathbf{F}_1$ of frequent 1-itemsets and sorts them in descending order of noisy supports. Then, it uses a sliding window of length l on the sorted $\mathbf{F}_1$ and directly generates all confident association rules in the window. Similarly, the HCRMine algorithm only mines the most K confident association rules and averagely separates the overall privacy budget into K parts. However, if K is reached, then it still can generate new association rules and uses weighted reservoir sampling wrapped with an exponential mechanism to determine whether a new generated association rule should be retained by discarding a saved association rule. It isolates items inside and outside the sliding window and hides the $sup(\cdot)$ and $conf(\cdot)$ information.

In summary, all existing solutions are only suitable for a single support threshold *minsup*, and some of them only work well for the very large *minsup*. The average allocation of privacy budgets after estimating K is not ideal because it cannot offset the effect of the sensitivities on noise, and the top-K methods may compromise the integrity of the mining results because the original goal requires finding all strong association rules.

Moreover, the HCRMine algorithm is problematic because the items inside and outside the sliding window cannot exist in the same association rule, and we cannot judge the importance of the association rules because their confidence and support are hidden by the exponential mechanism. The HCRBins algorithm [27] tries to improve on the utility of the HCRMine algorithm by using the parallel composition theorem. It decomposes the item universe I into J disjoint subsets, where each subset $j \in J$ generates $\frac{K}{J}$ confident association rules by using the overall privacy budget ε. However, the HCRBins algorithm may violate differential privacy because the parallel composition theorem should be applied to disjoint subsets of a dataset (such as $\mathbf{T}$) rather than to disjoint subsets of a dictionary (such as I). For simplicity, we consider the case when $J = 2$ (*i.e.*, $I = I_1 \cup I_2$, where $I = I_1 \cap I_2 = \varnothing$). For a transaction $t \in \mathbf{T}$ s.t. $t \cap I_1 \neq \varnothing$ and $t \cap I_2 \neq \varnothing$, if some items are used to generate association rules in I_2, the rest of the items remain available for generating association rules in I_1. This is a reaccess of $\mathbf{T}$ that causes more privacy loss and fails to satisfy the precondition of the parallel composition theorem.

Although the feasibility and efficiency of these methods have been demonstrated, they were all designed based on single support, which tends to suffer from the rare item problem. By contrast, our proposed technique can extract high-confidence rules with adaptive supports. To the best of our knowledge, our work is the first DPARM algorithm to adopt multiple support thresholds.

IV. COMPUTATIONAL MODEL AND KEY CHALLENGES

Before we describe the DPARM algorithm in detail, we present the formalized computational model. We also identify the key challenges to achieving the goal.

A. COMPUTATIONAL MODEL

The computational model is shown in Fig. 1. A trusted server holds a transactional dataset $\mathbf{T} = \{t_1, t_2, \ldots, t_n\}$ of n individuals. Each individual contributes a transaction $t_j \subseteq I$, where $I = \{i_1, i_2, \ldots, i_m\}$ is an item universe containing m distinct items. Note that $\mathbf{T}$ can be seen as a compact representation of a relational dataset $\mathbf{R} = [r_1, r_2, \ldots, r_n]$, where $r_j \in \{0, 1\}^m$. A data analyst who is also assumed to be an adversary interacts with the trusted server by posing a statistical query $q \in [0, 1]$ to obtain information about $\mathbf{T}$, and q is given a response only through the randomization mechanism M.

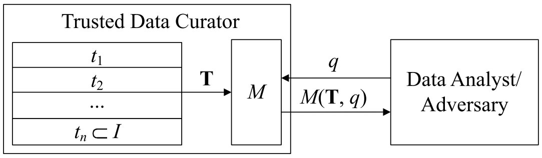


FIGURE 1. Computational model.

B. KEY CHALLENGES

A DPARM algorithm mainly involves two steps:

- Mine all frequent itemsets from the candidate itemsets.
- Mine all confident association rules from the frequent itemsets.

Two key challenges affect the first step. The first challenge is that the high sensitivity of the queries leads to large-scale noise. Let $\mathbf{q} = [q_1, \ldots, q_{|\mathbf{C}_k|}]$ be a vector of counting queries where each q_j computes the support of each candidate $c \in \mathbf{C}_k$, and $\mathbf{C}_k$ is the set of candidate itemsets with cardinality k. The sensitivity of $\mathbf{q}$ can be easily computed as $\Delta\mathbf{q} = \min\{\binom{m}{k}, |\mathbf{C}_k|\}/n$ because the addition or removal of a single transaction can increase or reduce the support of $\binom{m}{k}$ itemsets by at most $\frac{1}{n}$ simultaneously. Clearly, the dimensionality m of the transactional dataset and the amount $|\mathbf{C}_k|$ of the candidate k-itemsets determine the sensitivity of the queries.

Consider an example. For the Laplace mechanism M_{Lap}, if the maximum cardinality K of frequent itemsets is 4 and the overall privacy loss is bounded by 2, the scale of noise will be unacceptably large relative to the support, which results in no utility at all. When mining the frequent 1-itemsets with $m = 2000$ and $n = 50000$, the scale parameter of Laplace noise is $\frac{\Delta\mathbf{q}}{\varepsilon} = \frac{2000/50000}{2/4} = \frac{2}{25}$ and the standard deviation is $\sqrt{2} \times \frac{2}{25} \approx 0.1131$. However, the typical support threshold λ is 0.01, which is one order of magnitude smaller than the noise.

The second challenge is that the improper allocation of privacy budgets leads to unstable noise. It is difficult to know the maximum cardinality K of the frequent itemsets in advance and in a differentially private manner because the mining process utilizes the breadth-first search strategy; that is, the algorithms first mine $\mathbf{F}_k$ from $\mathbf{C}_k$ and then constructs $\mathbf{C}_{k+1}$ according to $\mathbf{F}_k$. Even if we can obtain K, it is also difficult to allocate privacy budget for each submining process with cardinality k because the sensitivity is significantly changed for different k, which makes the noise severely vary.

V. DPARM ALGORITHM

In this article, we propose a DPARM algorithm that resolves the challenges mentioned in Section IV-B by adopting a group of techniques, namely random truncation and partition of a transactional dataset and adaptive choice of privacy budgets. Notation is listed in Table 1, and the bold letters are used to denote both vectors and nested sets.

A. OVERVIEW OF THE DPARM ALGORITHM

We plot the flow of the DPARM algorithm in Fig. 2. The steps requiring privacy protection are presented in red, and the steps involving multiple support thresholds are shown in blue.

TABLE 1. Notation.

Notation	Description
$\mathbf{T}$	Transactional dataset
$\mathbf{T}^{(a)}, \mathbf{T}^{(b)}$	Truncated and split transactional dataset
θ	Truncation length
r	Split rate
I	Item universe
S	Sorted item universe
λ	Lowest allowed MIS
ρ	Support relevance
φ	Maximum support difference
$\mathbf{C}_k$	Set of candidate k-itemsets
$\mathbf{F}_k$	Set of frequent k-itemsets
$\varepsilon, \varepsilon^{(a)}, \varepsilon_k^{(b)}$	Privacy budgets

We first randomly truncate the long transactions in $\mathbf{T}$ and uniformly partition $\mathbf{T}$ into two disjoint subsets $\mathbf{T}^{(a)}$ and $\mathbf{T}^{(b)}$ for different tasks. In particular, $\mathbf{T}^{(a)}$ is used to obtain the sorted item universe S, which stores the minimum support $i.MIS$ for each item $i \in I$ in ascending order. $\mathbf{T}^{(b)}$ is used together with S to mine frequent itemsets and further strong association rules through a breadth-first search strategy. Note that we use purple (which can be made by blending red and blue) to highlight this step because it should be not only differentially private but also the basis for assigning multiple support thresholds.

We must scan truncated transactional datasets many times and obtain support information, and each time should be differentially private. We only need to select the initial privacy budget, which is used to count the support $i.sup$ for each item $i \in I$. When $k \geq 2$ and counting the support $c.sup$ for each candidate itemset $c \in \mathbf{C}_k$, the privacy budget is adaptively computed after $\mathbf{C}_k$ are generated. Hence, we do not need to estimate or specify the maximum cardinality K in advance.

We use multiple support thresholds to better simulate the real world, which yields fewer frequent itemsets because the concept of "frequent" is no longer absolute but is rather related to the nature of the items in the itemset. This not only makes the mining results more useful but also reduces the number of candidate itemsets. As stated in [13], when $\rho = 0.25$ and $\lambda = 0.01$, the number of frequent itemsets is only 8.5% of the result yielded by algorithms using a single support threshold, and the number of candidate itemsets is further reduced to less than 4%. The utility of $i.MIS$ is critical because it determines the sorted item universe S and the minimum support $c.MS$ for each candidate itemset $c \in \mathbf{C}_k$ (when $k \geq 2$).

B. INITIALIZATION

Before beginning the mining process, we first need to initialize the transactional dataset. The algorithm, shown in Algorithm 1, can be divided into three parts: identifying the distinct items, estimating the cardinality distribution, and truncating and partitioning the transactional dataset.

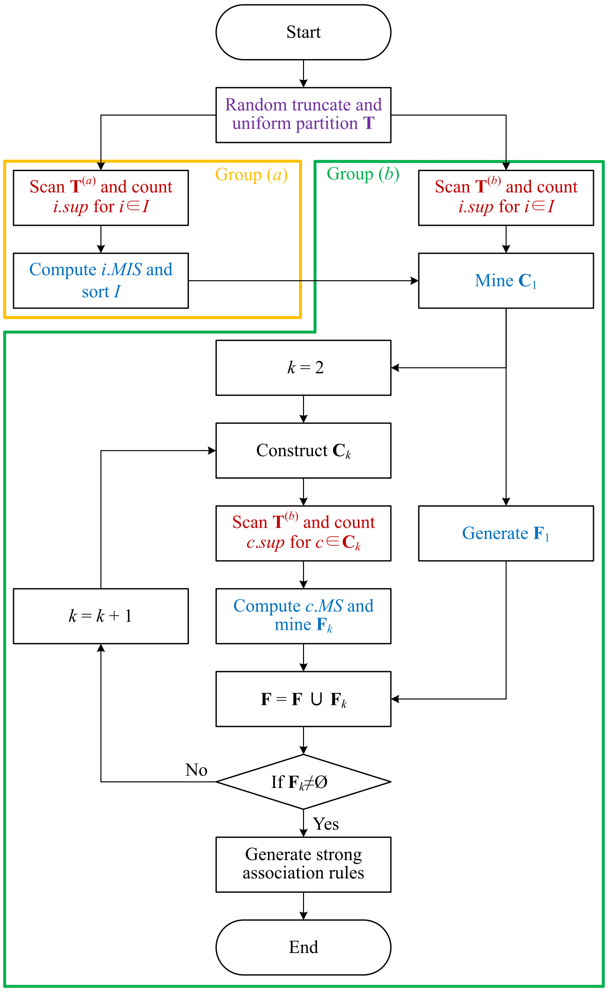

Start
Random truncate and uniform partition T
Group (a)
Group (b)
Scan T^(a) and count i.sup for i∈I
Compute i.MIS and sort I
Scan T^(b) and count i.sup for i∈I
Mine C1
k = 2
Construct Ck
Scan T^(b) and count c.sup for c∈Ck
Compute c.MS and mine Fk
Generate F1
k = k + 1
F = F ∪ Fk
No
If Fk≠Ø
Yes
Generate strong association rules
End

FIGURE 2. Flowchart of the DPARM algorithm.

In lines 4 to 8, the algorithm finds the item universe I from the transactional dataset $\mathbf{T}$, which is trivial but necessary for all mining algorithms. The maximum cardinality of transactions is equal to the number of distinct items $m = |I|$.

In lines 9 to 10, the algorithm computes the noisy cardinality distribution $\mathbf{d}$. It divides $\mathbf{T}$ into m disjoint subsets according to the cardinality j and counts the number d_j of transactions in each subset through the Laplace mechanism. The privacy analysis is shown in Theorem 1. In line 11, the algorithm computes the transaction cardinality θ through $\mathbf{d}$, and θ is the smallest integer that makes the summation of d_j for $j = 1$ to θ larger than a p-th percentile of n. The typical value of p is 0.85 [41]. However, as shown in Fig. 3, the true cardinality distributions of variant datasets (*e.g.*, Retail, BMS1, BMS2, Action, and Kosarak) are different when $j \leq 10$. Therefore, we view p as a function of $\mathbf{T}$ with range (0, 1) rather than a fixed value. Note that p can also be optimized using the method introduced in [8].

In lines 12 to 16, the algorithm truncates the transactions in $\mathbf{T}$. For each transaction $t \in \mathbf{T}$ with cardinality larger than θ,

Algorithm 1 Algorithm for Initializing the Transactional Dataset

```
1  Input: Transactional dataset $\mathbf{T}$, split rate $r$, and privacy
          budget $\varepsilon$
2  Output: Item Universe $I$, truncation cardinality $\theta$, and
           truncated transactional dataset $\mathbf{T}^{(a)}$, $\mathbf{T}^{(b)}$
3  Function(InitDataset ($\mathbf{T}, r, \varepsilon$))
4  $I = \varnothing$
5  for each transaction $t \in \mathbf{T}$ do
6      for each item $i \in t$ do
7          if $i \notin I$ then
8              $I = I \cup \{i\}$
           end
       end
   end
9  $\mathbf{T} = \{\mathbf{T}_1, \ldots, \mathbf{T}_m\}$            // $\mathbf{T}_j$ contains
   transactions with cardinality $j$
10 $\mathbf{d} = [d_1, \ldots, d_m] = [n_1 + Z_1, \ldots, n_m + Z_m]$
   // $n_j = ||\mathbf{T}_j||_1$ and $Z_j \overset{iid}{\sim} Lap(\frac{1}{\varepsilon})$
11 $\theta$ = The minimum $\theta$ such that $\sum_{j=1}^{\theta} d_j \geq p(\mathbf{T}) \times n$
   // $p \in (0, 1)$
12 $\mathbf{T}^{(a)}, \mathbf{T}^{(b)} = \varnothing, \varnothing$
13 for $(j = 1; j \leq m; j{+}{+})$ do
14     for each transaction $t \in \mathbf{T}_j$ do
15         if $|t| > \theta$ then
16             $t$ = Randomly select $\theta$ items from $t$
           end
       end
17     $\mathbf{T}_j^{(a)}$ = Randomly select $r \times d_j$ transactions from $\mathbf{T}_j$
18     $\mathbf{T}_j^{(b)} = \mathbf{T}_j \setminus \mathbf{T}_j^{(a)}$
19     $\mathbf{T}^{(a)} = \mathbf{T}^{(a)} \cup \mathbf{T}_j^{(a)}$; $\mathbf{T}^{(b)} = \mathbf{T}^{(b)} \cup \mathbf{T}_j^{(b)}$
   end
20 return $I, \theta, \mathbf{T}^{(a)}, \mathbf{T}^{(b)}$
```

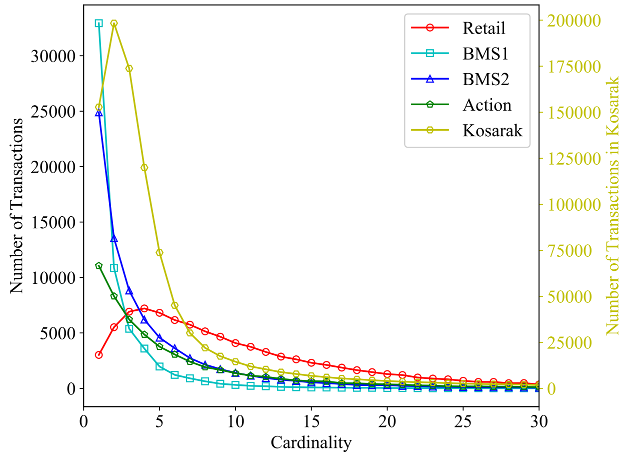


FIGURE 3. True cardinality distribution.

we randomly select θ items in t and remove other items. In lines 17 to 19, the algorithm partitions $\mathbf{T}$ in a uniformly random manner. Let $r \in (0, 1)$ be the split rate of partitioning $\mathbf{T}$ into two parts $\mathbf{T}^{(a)}$ and $\mathbf{T}^{(b)}$ with size $r \times n$ and $(1 - r) \times n$. For $j = 1$ to m, we randomly pick $r \times d_j$ transactions and put them into $\mathbf{T}^{(a)}$ and put the other transactions into $\mathbf{T}^{(b)}$.

C. MIS ASSIGNMENT AND SUPPORT COUNTING

During the mining process, we need to repeatedly access the transactional dataset to obtain support information. This is done not only in MIS assignment but also in generating the set $\mathbf{C}_1$ of candidate 1-itemsets from S and mining the set $\mathbf{F}_k$ of frequent k-itemsets from $\mathbf{C}_k$ for $k > 1$. The NoisyCount algorithm is shown in Algorithm 2, which is an interface between $\mathbf{T}$ and the algorithms requiring the support information.

Algorithm 2 Algorithm for Differentially Private Computation of Supports

```
1  Input: Transactional dataset $\mathbf{T}$, general set $\mathbf{G}$, sensitivity
          $\Delta$, and privacy budget $\varepsilon$
2  Output: Set $\mathbf{G}$ of candidate itemsets
3  Function(NoisyCount ($\mathbf{T}, \mathbf{G}, \Delta, \varepsilon$))4 for each
   transaction $t \in \mathbf{T}$ do
5      for each itemset $g \in \mathbf{G}$ do
6          if $g \subseteq \mathbf{T}$ then
7              $g.sup{+}{+}$
           end
       end
   end
8  for each itemset $g \in \mathbf{G}$ do
9      $g.sup = g.sup + Lap(\frac{\Delta}{\varepsilon})$
10     $g.sup$ = Round($g.sup$)
   end
11 return G
```

In lines 4 to 9, the algorithm computes the noisy support of each element $g \in \mathbf{G}$ through the Laplace mechanism, where $\mathbf{G}$ is a general set representing item universe I, sorted item universe S, or the set of candidate k-itemsets $\mathbf{C}_k$ in different scenarios. The privacy analysis is presented in Theorem 2. In line 10, the algorithm normalizes the noisy supports through the Round function. In brief, if the noisy support is smaller than 0 or larger than 1, then we round it to 0 or 1. An alternative method is to add or subtract a single standard deviation [17].

At the start of the mining process, we need to assign the MIS for each item $i \in I$. Rather than specifying the support thresholds in advance, specification of MIS is closely related to the support of i. The algorithm for assigning MIS is shown in Algorithm 3. Each item $i \in I$ has two attributes, *i.e.*, $i.MIS$ and $i.sup$, which denote MIS and support of i.

In line 4, the algorithm calls the NoisyCount algorithm and obtains the support information of each item $i \in I$. In lines 5 to 7, the algorithm assigns MIS for each item $i \in I$ according to Lemma 1 and sorts I in ascending order of $i.MIS$.

Algorithm 3 Algorithm for Assigning MIS

```
Input: Transactional dataset $\mathbf{T}$, item universe $I$, support
       relevance $\rho$, lowest allowed MIS $\lambda$, sensitivity $\Delta$,
       and privacy budget $\varepsilon$
Output: Sorted item universe $S$
Function (AssignMIS ($\mathbf{T}, I, \rho, \lambda, \Delta, \varepsilon$))
$I$ = NoisyCount($\mathbf{T}^{(a)}, I, \Delta, \varepsilon$)   // Update $i.sup$
for each item $i \in I$ do
    $i.MIS = \max\{\rho \times i.sup, \lambda\}$
end
$S = \mathrm{Sort}(I)$   // Sort $I$ in ascending order of $i.MIS$
return $S$
```

Algorithm 4 The DPARM Algorithm

```
Input: Transactional dataset $\mathbf{T}$, split ratio $r$, support
       relevance $\rho$, lowest allowed MIS $\lambda$, maximum
       support difference $\varphi$, and privacy budgets
       $\varepsilon_1, \varepsilon^{(a)}, \varepsilon_1^{(b)}$
Output: The set $\mathbf{F}$ of frequent itemsets and the privacy
        loss $\mathcal{E}$
Function (DPARM ($\mathbf{T}, r, \rho, \lambda, \varphi, \varepsilon_1, \varepsilon^{(a)}, \varepsilon_1^{(b)}$))
$I, \theta, \mathbf{T}^{(a)}, \mathbf{T}^{(b)}$ = InitDataset($\mathbf{T}, r, \varepsilon_1$)
$\Delta^{(a)} = \theta / n^{(a)}$                    // $n^{(a)} = |\mathbf{T}^{(a)}|$
$S$ = AssignMIS($\mathbf{T}^{(a)}, I, \rho, \lambda, \Delta^{(a)}, \varepsilon^{(a)}$)
$\Delta_1^{(b)} = \theta / n^{(b)}$                    // $n^{(b)} = |\mathbf{T}^{(b)}|$
$\mathbf{C}_1, \mathbf{F}_1$ = Mine1($\mathbf{T}^{(b)}, S, \Delta_1^{(b)}, \varepsilon_1^{(b)}$)
// Algorithm 5
for ($k = 2; \mathbf{F}_{k-1} \neq \varnothing; k++$) do
    if $k == 2$ then
        $\mathbf{C}_k$ = GenCandidate2($\mathbf{C}_1, \varphi$)
        // Algorithm 6
    end
    else
        $\mathbf{C}_k$ = GenCandidate($\mathbf{F}_{k-1}, \varphi$)
        // Algorithm 7
    end
    $\Delta_k^{(b)} = \min\{\binom{\theta}{k}, |\mathbf{C}_k|\} / n^{(b)}$
    $\varepsilon_k^{(b)} = \varepsilon_1^{(b)} \cdot \Delta_k^{(b)} / \Delta_1^{(b)}$
    $\mathbf{F}_k$ = MineFrequenter($\mathbf{T}^{(b)}, \mathbf{C}_k, \Delta_k^{(b)}, \varepsilon_k^{(b)}$)
    // Algorithm 8
end
$\mathbf{F} = \cup_k \mathbf{F}_k$
return $\mathbf{F}$
```

D. MAIN ALGORITHM

Now we are ready to explain our main algorithm, DPARM, which is shown in Algorithm 4. The DPARM algorithm consists of the following steps.

(1) For $\mathbf{T}$ and the privacy budget ε_1, initialize $\mathbf{T}$ and obtain the item universe I, the truncation length θ, and the truncated and partitioned transactional datasets $\mathbf{T}^{(a)}$ and $\mathbf{T}^{(b)}$.

(2) For $\mathbf{T}^{(a)}$ and the privacy budget $\varepsilon^{(a)}$, assign the MIS and obtain the sorted item universe S with support information.

(3) For $\mathbf{T}^{(b)}$, $k = 1$, and the privacy budget $\varepsilon_1^{(b)}$:

   a) Generate the set $\mathbf{C}_k$ of candidate k-itemsets from S (if $k = 1$), $\mathbf{C}_{k-1}$ (if $k = 2$), or $\mathbf{F}_{k-1}$ (if $k > 2$).

   b) Adaptively select the privacy budget $\varepsilon_k^{(b)}$ (if $k \geq 2$).

   c) Mine the set $\mathbf{F}_k$ of frequent k-itemsets from $\mathbf{C}_k$. If $\mathbf{F}_k$ is not empty, then go back to (3-a) with $k+1$. Otherwise, publish all the frequent itemsets with their noisy support.

We use multiple support thresholds to improve simulation of the real world, which results in fewer frequent itemsets because the concept of "frequent" is no longer absolute but is related to the nature of the items in the itemset. This not only makes the mining results more useful but also reduces the number of candidate itemsets. As stated in [13], when $\rho = 0.25$ and $\lambda = 0.01$, the number of frequent itemsets is only 8.5% of the number of results yielded by the algorithms using a single support threshold, and the number of candidate itemsets is further reduced to less than 4% of the number yielded by the other algorithms. Note that the utility of MIS is critical because it determines the sorted item universe S and the minimum support of candidate itemsets.

We refer to the random truncation in [41] to reduce the dimensionality m of $\mathbf{T}$, but differ from [55] in implementing the step of uniformly random partition, which is a follow-up work of random truncation without any privacy consumption. Before completing such partition, we must assign the MIS with the same privacy budget as mining $\mathbf{C}_1$ and $\mathbf{F}_1$, which may compromise the utility of the MIS. However, after partitioning the dataset uniformly at random, we can use the same privacy budget as the entire mining process and also obtain almost the same MIS as the result in the original dataset.

We also adaptively select the privacy budgets to stabilize the noise scale. The intuition is to maintain the scale parameter of Laplace noise, which requires the privacy budget be changed synergistically with the sensitivity, *i.e.*, $\frac{\Delta_1^{(b)}}{\varepsilon_1^{(b)}} = \ldots = \frac{\Delta_K^{(b)}}{\varepsilon_K^{(b)}}$. By utilizing the notation of ex post differential privacy [20] and sequential composition theorems under adaptive settings [29], we can finally bound the overall privacy loss.

E. MISCELLANEOUS

The algorithm for generating the set $\mathbf{C}_k$ of candidate itemsets does not entail privacy concerns because its seed is already produced through a differentially private approach. The only exception is when $k = 1$ because $\mathbf{C}_1$ is mined from S, which relies on support information. In addition, the algorithm for mining the set $\mathbf{F}_k$ of frequent itemsets also needs support information, which is obtained by calling the `NoisyCount`

Algorithm 5 Algorithm for Mining Candidate 1-Itemsets and Frequent 1-Itemsets

```
Input: Truncated transactional dataset $\mathbf{T}^{(b)}$, sorted item
        universe $S$, sensitivity $\Delta_1^{(b)}$, and privacy budget
        $\varepsilon_1^{(b)}$
Output: Set $\mathbf{C}_1$ of candidate 1-itemsets and set $\mathbf{F}_1$ of
         frequent 1-itemsets
Function (Mine 1($\mathbf{T}^{(b)}, S, \Delta_1^{(b)}, \varepsilon_1^{(b)}$))
$S$ = NoisyCount($\mathbf{T}^{(b)}, S, \Delta_1^{(b)}, \varepsilon_1^{(b)}$)
$\mathbf{C}_1 = \varnothing$
for each item $i \in S$ do
    if $\mathbf{C}_1 == \varnothing$ then
        if $i.sup \geq i.MIS$ then
            $\gamma = i.MIS$
            $\mathbf{C}_1 = \mathbf{C}_1 \cup \{\{i\}\}$
        end
    end
    else
        if $i.sup \geq \gamma$ then
            $\mathbf{C}_1 = \mathbf{C}_1 \cup \{\{i\}\}$
        end
    end
end
$\mathbf{F}_1 = \varnothing$
for each itemset $\{i\} \in \mathbf{C}_1$ do
    $\{i\}.MS = i.MIS$; $\{i\}.sup = i.sup$
    if $\{i\}.sup \geq \{i\}.MS$ then
        $\mathbf{F}_1 = \mathbf{F}_1 \cup \{\{i\}\}$
    end
end
return $\mathbf{C}_1, \mathbf{F}_1$
```

algorithm. To avoid duplication, we simply list the algorithms separately in Algorithms 5, 6, 7, and 8. Note that each itemset $c \in \mathbf{C}_k$ has two attributes, *i.e.*, $c.MS$ and $c.sup$, which denote the minimum support and the support of $\mathbf{c}$.

VI. PRIVACY ANALYSIS

In this section, we analyze the proposed method for privacy-preserving data mining that combines the Laplace mechanism of differential privacy. Specifically, we show that: (i) the proposed method of computing the noisy cardinality distribution $\mathbf{d}$ that satisfies ε-differential privacy, (ii) the proposed method of computing the supports of all $g \in \mathbf{G}$ that satisfy ε-differential privacy, and (iii) for the maximum cardinality K of all the candidate itemsets, the DPARM algorithm satisfies $\mathcal{E}$-ex post differential privacy.

Initially, our algorithm needs to compute the noisy cardinality distribution $\mathbf{d}$. It divides $\mathbf{T}$ into m disjoint subsets according to the cardinality j and counts the number d_j of transactions in each subset through the Laplace mechanism. To prove the noisy cardinality distribution $\mathbf{d}$ that satisfies ε-differential privacy, we analyze it as follows:

Algorithm 6 Algorithm for Generating Candidate 2-Itemsets

```
Input: Set $\mathbf{C}_1$ of candidate 1-itemsets and maximum
        support difference $\varphi$
Output: Set $\mathbf{C}_2$ of candidate 2-itemsets
Function (GenCandidate 2($\mathbf{C}_1, \varphi$)4 $\mathbf{C}_2 = \varnothing$
for each itemset $\{i\} \in \mathbf{C}_1$ do
    if $\{i\}.sup \geq \{i\}.MS$ then
        for each itemset $\{j\} \in \mathbf{C}_1$ after $\{i\}$ do
            if $|j.sup - i.sup| \leq \varphi$ then
                $\mathbf{C}_2 = \mathbf{C}_2 \cup \{\{i, j\}\}$
            end
        end
    end
end
return $\mathbf{C}_2$
```

Algorithm 7 Algorithm for Generating k-Itemsets for $k > 2$

```
Input: Set $\mathbf{F}_{k-1}$ of frequent $(k-1)$-itemsets and
        maximum support difference $\varphi$
Output: Set $\mathbf{C}_k$ of candidate $k$-itemsets
Function (GenCandidate ($\mathbf{F}_{k-1}, \varphi$)4 $\mathbf{C}_k = \varnothing$
for each pair of itemsets $f_1, f_2 \in \mathbf{F}_{k-1}$ do
    // $f_1 = \{i_1, \ldots, i_{k-2}, i_{k-1}\}$ and
    // $f_2 = \{i_1, \ldots, i_{k-2}, j_{k-1}\}$ such that
    // $i_{k-1} \leq j_{k-1}$ and $|j_{k-1}.sup - i_{k-1}.sup| \leq \varphi$
    $c = \{i_1, \ldots, i_{k-2}, i_{k-1}, j_{k-1}\}$
    $\mathbf{C}_k = \mathbf{C}_k \cup \{c\}$
    for each $(k-1)$-subset $s$ of $c$ do
        if $i_1 \in s$ or $i_2.MIS == i_1.MIS$ then
            if $s \notin \mathbf{F}_{k-1}$ then
                $\mathbf{C}_k = \mathbf{C}_k \setminus \{c\}$
            end
        end
    end
end
return $\mathbf{C}_k$
```

Theorem 1: Compute the noisy cardinality distribution $\mathbf{d}$ *that satisfies* ε*-differential privacy.*

Proof: Let $\mathbf{q} = [q_1, \ldots, q_m]$ where each q_j counts the number n_j of transactions in $\mathbf{T}_j$. We can compute that $\Delta\mathbf{q} = 1$ because the addition or removal of a single transaction can increase or decrease a single n_j by at most 1. Therefore, for a privacy budget ε, answering $\mathbf{q}$ through the Laplace mechanism with the scale parameter $\frac{1}{\varepsilon}$ satisfies ε-differential privacy. □

During the mining process, we need to repeatedly access the transactional dataset to obtain support information. For protecting the privacy of the transactional dataset, our algorithm accesses the noisy support of each element $g \in \mathbf{G}$

Algorithm 8 Algorithm for Mining Frequent k-Itemsets for $k > 1$

```
Input: Truncated transactional dataset $\mathbf{T}^{(b)}$, sorted item
       universe $S$, sensitivity $\Delta_k^{(b)}$, and privacy budget
       $\varepsilon_k^{(b)}$
Output: Set $\mathbf{F}_k$ of frequent $k$-itemsets
Function(MineFrequenter($\mathbf{T}^{(b)}, \mathbf{C}_k, \Delta_k^{(b)}, \varepsilon_k^{(b)}$))
$\mathbf{C}_k$ = NoisyCount($\mathbf{T}^{(b)}, \mathbf{C}_k, \Delta_k^{(b)}, \varepsilon_k^{(b)}$)
$\mathbf{F}_k = \varnothing$
for each itemset $c \in \mathbf{C}_k$ do
    $c.MS = \min_{i \in c}\{i.MIS\}$
    if $c.sup \geq c.MS$ then
        $\mathbf{F}_k = \mathbf{F}_k \cup c$
    end
end
return $\mathbf{F}_k$
```

through the Laplace mechanism. The privacy guarantee can be proven as follows:

Theorem 2: Compute the supports of all $g \in \mathbf{G}$ that satisfy ε-differential privacy.

Proof: Let $\mathbf{q} = [q_1, \ldots, q_{|G|}]$, where each q_j computes the support of $g \in \mathbf{G}$ in $\mathbf{T}$. We can compute that $\Delta\mathbf{q} = \frac{1}{n}\min\{\binom{\theta}{k}, |\mathbf{G}|\}$ because the addition or removal of a single transaction can at most increase or decrease the support of $\binom{\theta}{k}$ itemsets by $\frac{1}{n}$ simultaneously. Therefore, for a privacy budget ε, answering $\mathbf{q}$ through the Laplace mechanism with the scale parameter $\frac{1}{\varepsilon n}\min\{\binom{\theta}{k}, |\mathbf{G}|\}$ satisfies ε-differential privacy. □

We conclude this section by resulting from Theorems 1 and 2. Here, we prove that for the maximum cardinality K of all the candidate itemsets, the DPARM algorithm satisfies $\mathcal{E}$-ex post differential privacy.

Theorem 3: For the maximum cardinality K of all the candidate itemsets, the DPARM algorithm satisfies $\mathcal{E}$-ex post differential privacy for $\mathcal{E} = \varepsilon_1 + \max\{\varepsilon^{(a)}, \varepsilon^{(b)}\}$, where $\varepsilon^{(b)} = \sum_{k=1}^{K} \varepsilon_k^{(b)}$.

Proof: After the mining process is finished, both the maximum cardinality K of all the candidate itemsets and the total privacy loss $\mathcal{E}$ are fixed. For $\mathbf{T}$ and the privacy budget ε_1, `InitDataset` satisfies ε_1-differential privacy according to Definition 7 and Theorem 1. For $\mathbf{T}^{(a)}$ and the privacy budget $\varepsilon^{(a)}$, `AssignMIS` satisfies $\varepsilon^{(a)}$-differential privacy according to Definition 7 and Theorem 2. For $\mathbf{T}^{(b)}$ and the privacy budget $\varepsilon_k^{(b)}$, 1 times call of `Mine` 1 and $k-1$ times call of `MineFrequenter` satisfies $\varepsilon^{(b)}$-ex post differential privacy for $\varepsilon^{(b)} = \sum_{k=1}^{K} \varepsilon_k^{(b)}$ according to Definitions 7 and 8. $\{\mathbf{T}^{(a)}, \mathbf{T}^{(b)}\}$ is a random partition of $\mathbf{T}$ because $\mathbf{T}^{(a)}$ is uniformly sampled from $\mathbf{T}$, and $\mathbf{T}^{(a)} \cap \mathbf{T}^{(b)} = \varnothing$ and $\mathbf{T}^{(a)} \cup \mathbf{T}^{(b)} = \mathbf{T}$. Therefore, steps (2) and (3) satisfy $\max\{\varepsilon^{(a)}, \varepsilon^{(b)}\}$-ex post differential privacy such that the DPARM algorithm satisfies $\mathcal{E}$-ex post differential privacy for $\mathcal{E} = \varepsilon_1 + \max\{\varepsilon^{(a)}, \varepsilon^{(b)}\}$ according to Definitions 7 and 8. □

VII. EXPERIMENTS

A. ENVIRONMENT AND DATASETS

We implemented the DPARM algorithm in Python 3.6, and conducted experiments on a computer running Windows 10 with a 3.6 GHz Intel Core i7-4790 CPU and 16 GB of RAM. Each experiment was run 10 times, and the mean and standard deviation are reported. Because [13] is a nonprivate algorithm using multiple minimum supports to mine high-confidence association rules, it was assigned to be the ground truth in our experiments.

We ran the DPARM algorithm on the following datasets.

- The *Retail* dataset [32] contains the basket records provided by an anonymous Belgian retail supermarket. Each item is a stock-keeping unit (SKU), and each transaction contains all the SKUs that a customer purchased during an instance of shopping.
- The *BMS-WebView-1* and *BMS-WebView-2* [40] (abbreviated as BMS1 and BMS2) datasets contain several months of click-stream records from two e-commerce sites. Each item is a web page with product details, and each transaction contains all the web pages that a visitor viewed in one session.
- The *Action* dataset contains all the five star–rated action movies screened from the MovieLens 10M Dataset [11]. Each item is an action movie, and each transaction contains all the action movies to which an audience gave five stars.
- The *Kosarak* dataset [4] contains click-stream records of a Hungarian online news portal. Each item is a news page, and each transaction contains all the web pages that a visitor viewed from the news portal.

The characteristics of these datasets, particularly the number of transactions, number of distinct items, average cardinality of transactions, and maximum cardinality of transactions, are summarized in Table 2.

TABLE 2. Characteristics of datasets.

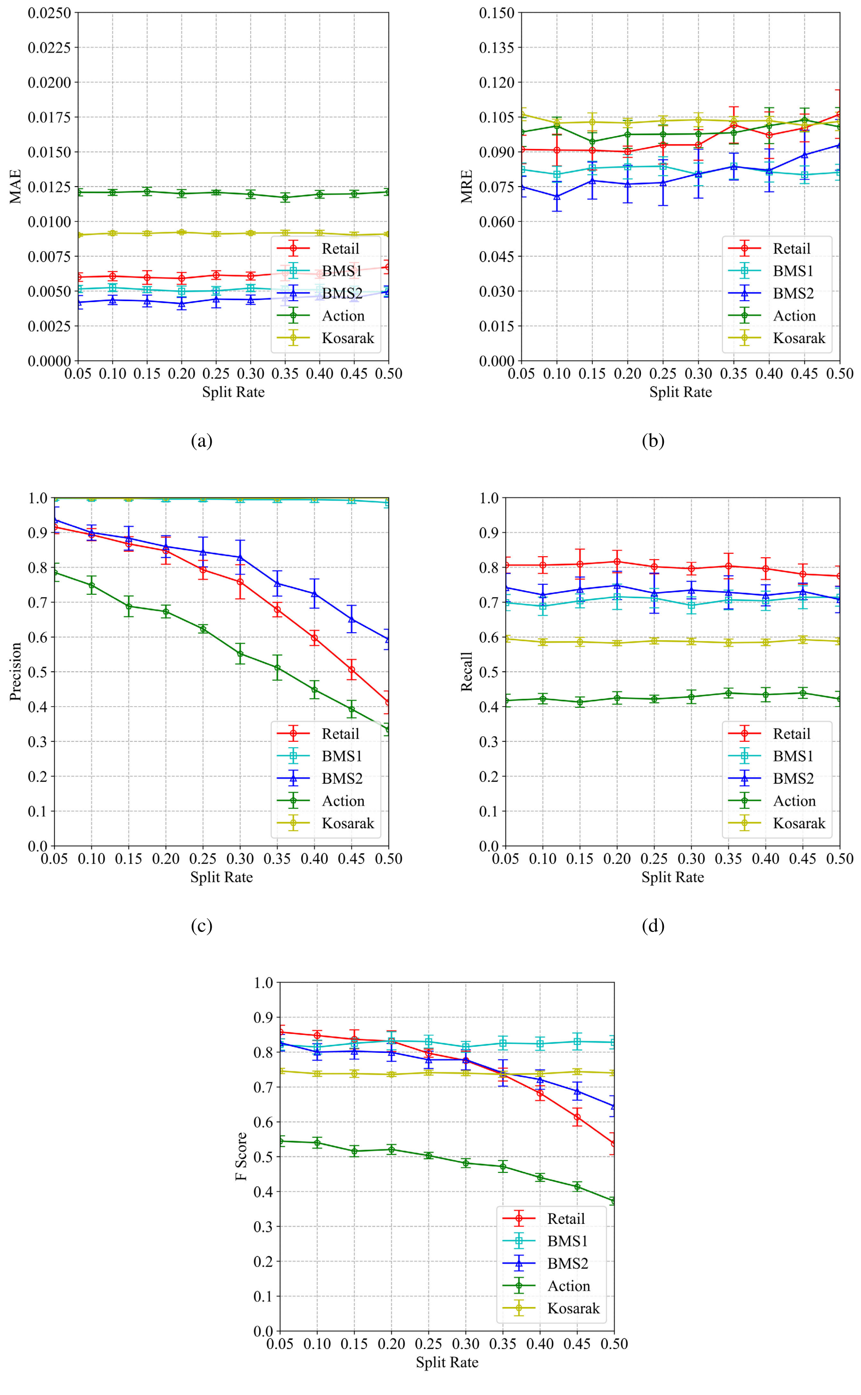
Dataset	n	m	avg$\vert t\vert$	max $\vert t\vert$
Retail	88162	16470	10	76
BMS1	59602	497	3	267
BMS2	77512	3340	5	161
Action	69878	1263	5	165
Kosarak	990002	41270	8	2498

B. UTILITY METRICS

We measured the performance of the DPARM algorithm according to the following utility metrics.

Definition 12 (Mean Error [18]): For all frequent itemset $f \in \mathbf{F}$, the mean absolute error (MAE) measures the mean error between the noisy truncated support and the original support:

$$MAE = \frac{1}{|\mathbf{F}|} \sum_{f \in \mathbf{F}} |nsup_{\mathbf{T}_\theta}(f) - sup_{\mathbf{T}}(f)|. \tag{9}$$



(a)

(b)

(c)

(d)

(e)

FIGURE 4. Split rate vs. utility.

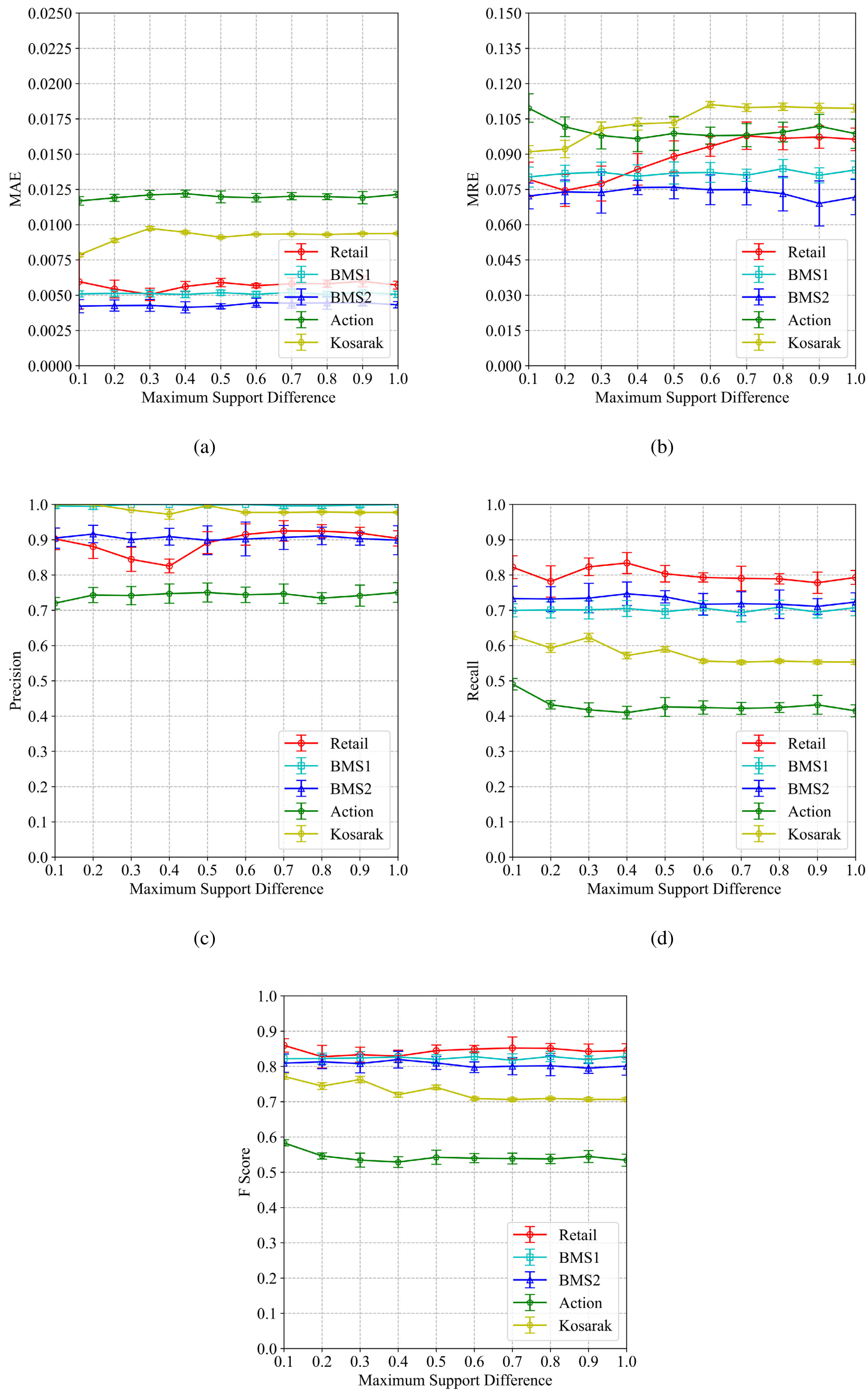
Similarly, the mean relative error (MRE) measures the mean error between the noisy truncated support and the original support relative to the original support:

$$MRE = \frac{1}{|\mathbf{F}|}\sum_{f\in\mathbf{F}} \frac{|nsup_{\mathbf{T}_\theta}(f) - sup_{\mathbf{T}}(f)|}{sup_{\mathbf{T}}(f)}. \tag{10}$$

Definition 13 (F Score [34]): For the set $\mathbf{F}'$ *of frequent itemsets mined by the DPARM algorithm and the set* $\mathbf{F}$ *of frequent itemsets mined by the nonprivate algorithm that serves as a ground truth in this paper, precision and recall are the proportions of common frequent itemsets* $\mathbf{F}' \cap \mathbf{F}$ *in* $\mathbf{F}'$ *and* $\mathbf{F}$, *respectively*:

$$precision = \frac{|\mathbf{F}' \cap \mathbf{F}|}{|\mathbf{F}'|}, \tag{11}$$

$$recall = \frac{|\mathbf{F}' \cap \mathbf{F}|}{|\mathbf{F}|}. \tag{12}$$



(a)

(b)

(c)

(d)

(e)

FIGURE 5. Maximum support difference vs. utility.

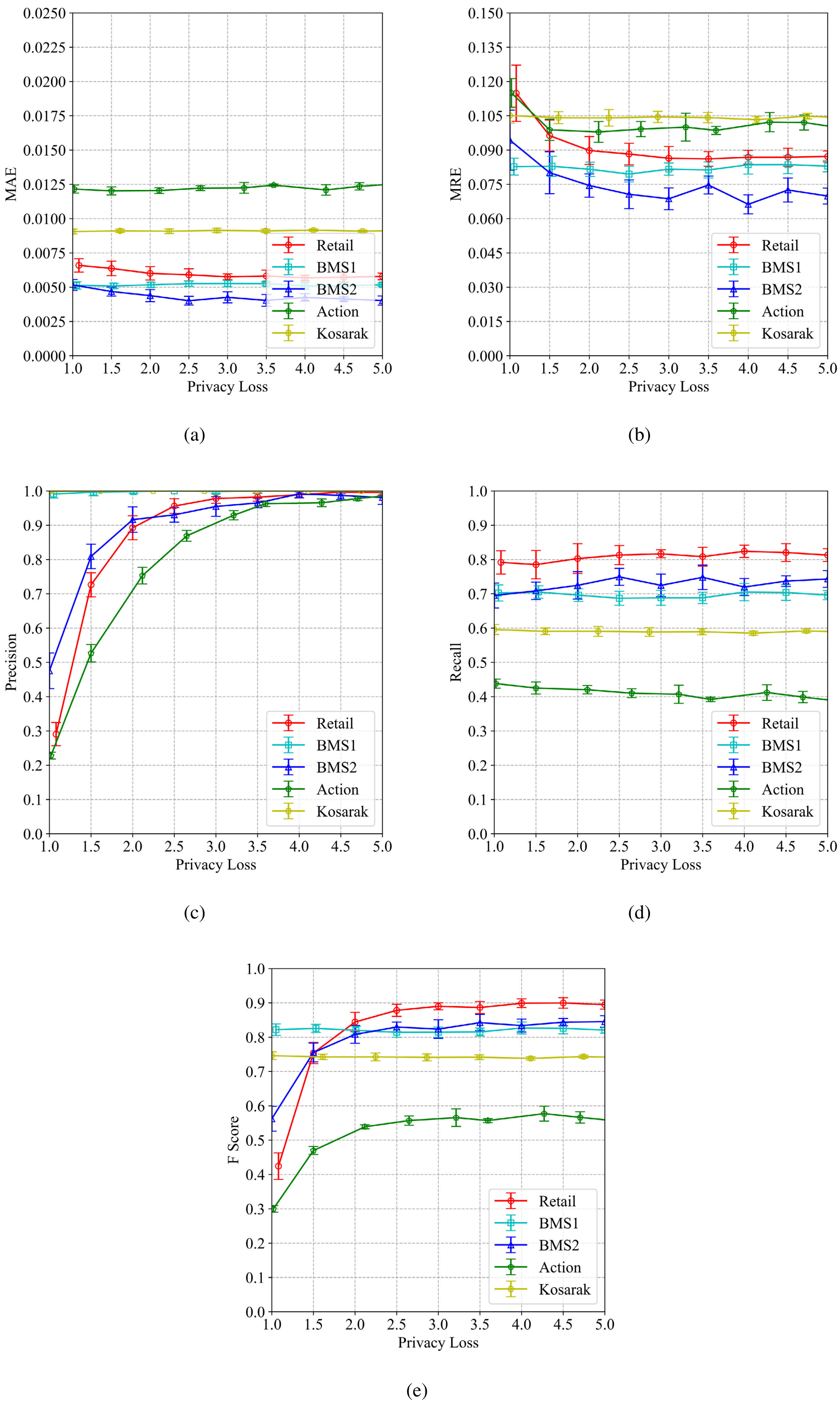
The F Score is the harmonic mean of precision and recall and is defined as

$$F\text{-}score = (\frac{precision^{-1} + recall^{-1}}{2})^{-1} = 2 \times \frac{precision \times recall}{precision + recall}. \quad (13)$$

C. SPLIT RATE VERSUS UTILITY

The experimental results are shown in Fig. 4. We bounded the overall privacy loss $\mathcal{E}$ to 2 and fixed the support relevance $\rho = 0.25$, the lowest allowed MIS $\lambda = 0.01$, and the maximum support difference $\varphi = 0.5$.

We evaluated the effect of the split rate r on utility by varying the split rate r from 0.05 to 0.5 with a step length



(a)

(b)

(c)

(d)

(e)

FIGURE 6. Privacy vs. utility.

of 0.05, which entails the number of transactions in $\mathbf{T}^{(a)}$ increasing until it equals $\mathbf{T}^{(b)}$. The larger the r, the lower is the utility of the mining results; this is because we must involve as many transactions as possible in the mining process. However, r cannot be set too low. This is because even if $\mathbf{T}^{(a)}$ is a uniform sample of $\mathbf{T}$, some items with extremely low support may still be missed. We compensated for this by setting the MIS of the missing items to the lowest allowed MIS λ. When r is too small, most of the items are missing, and thus, the use of multiple support thresholds is meaningless.

D. MAXIMUM SUPPORT DIFFERENCE VERSUS UTILITY

The experimental results are shown in Fig. 5. We bounded the overall privacy loss $\mathcal{E}$ to 2, and fixed the split rate $r = 0.05$, the support relevance $\rho = 0.25$, and the lowest allowed MIS $\lambda = 0.01$.

We evaluated the effect of the maximum support difference φ on utility by varying φ from 0.1 to 1 with a step length of 0.1. The utility of the mining results for most datasets did not significantly fluctuate as a consequence of changes in φ and always remained at a relatively high level. This indicates that the DPARM algorithm can perform well for mining frequent itemsets containing not only items whose support lies in a certain interval but also both low- and high-support items.

E. OVERALL PRIVACY LOSS VERSUS UTILITY

We then evaluated the effect of privacy parameters, *i.e.*, overall privacy loss $\mathcal{E}$, on utility. We fixed the split rate $r = 0.05$, the support relevance $\rho = 0.25$, the lowest allowed MIS $\lambda = 0.01$, and the maximum support difference $\varphi = 0.5$. We also fixed the privacy budget $\varepsilon_1 = 0.05$ and varied $\varepsilon^{(a)}$ from 1 to 5 with a step length of 0.5. The experimental results are shown in Fig. 6. Similar to most differentially private algorithms, with an increase in the overall privacy loss, the utility of the mining results improved. When the overall privacy budget was small, *e.g.*, approximately 2, the utility of the mining results was still acceptable.

F. COMPARISONS

Our work is closest to the literature PrivBasis [18], which focuses on generating frequent itemsets in a differentially private way. Using the noisy frequent itemset counts, they could in theory determine high confidence rules, but only rules with very high frequency are retained, whereas numerous moderately-frequent rules are discarded. In contrast, our proposed technique is able to extract high confidence rules that have adaptive supports. As shown in Figs. 7 and 8, we evaluated the effect of the privacy loss on *precision* and *MRE* by varying $\varepsilon^{(a)}$ from 1 to 5 with a step length of 0.5, in which the different values of k are set according to the size of each dataset and used to mine frequent itemsets. The experimental results are the average results after 10 runs. Our DPARM algorithm can outperform PrivBasis for mining frequent itemsets. Notably, using the accurate frequent itemsets, we can further determine high confidence rules.

In Fig. 7, for example, the values of *precision* for DPARM on Action, Retail, Kosarak, BMS2, and BMS1 are about 0.841, 0.956, 0.997, 0.949, and 0.998, respectively, while the values of *precision* for PrivBasis are about 0.694, 0.388, 0.996, 0.375, and 0.543 at privacy loss = 2.5. In Fig. 8, for example, the values of *MRE* for DPARM on Action, Retail, Kosarak, BMS2, and BMS1 are about 0.098, 0.091, 0.104, 0.070, and 0.080, respectively, while the values of MAE for PrivBasis are about 0.112, 0.048, 0.002, 0.111, and 0.221 at privacy loss = 2.5. Generally, the higher the k, the more frequent itemsets are mined. After almost all frequent itemsets in the ground truth are hit, the remaining mined itemsets that are not hit to the ground truth will increase the value of *MRE*. In contrast, when the frequent itemsets are not fully excavated, the higher the k, the more frequent itemsets are excavated. Under this situation, the more hits will be achieved. As a result, the value of *MRE* can be low.

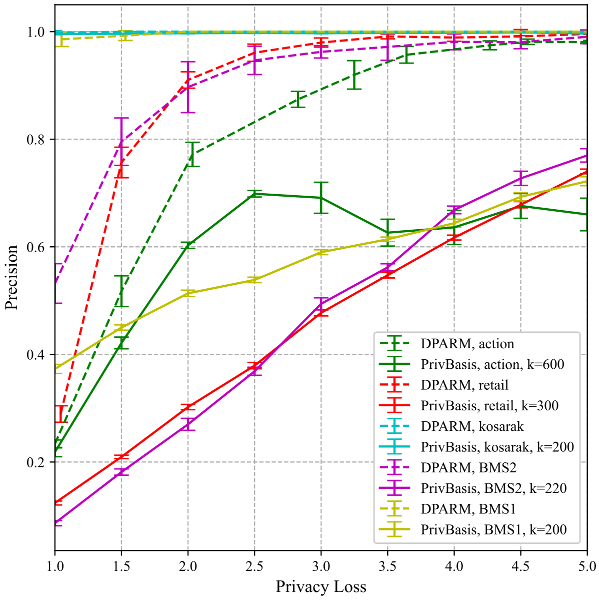


FIGURE 7. Effect of the privacy loss on *precision*.

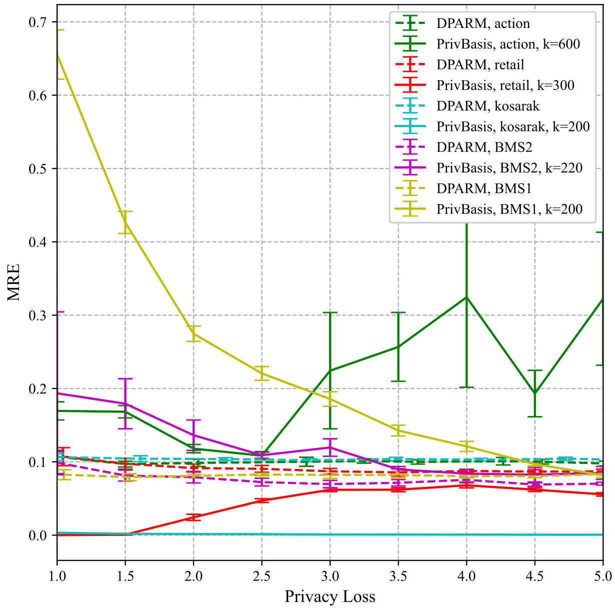


FIGURE 8. Effect of the privacy loss on *MRE*.

As shown in Table 3, when privacy loss is 2.5, Privbasis using single support greater than or equal to MIS (= 0.01) results in mining the number of frequent itemsets (which is 561 from Action dataset), while DPARM using multiple supports results in mining the number of frequent itemsets (which is 288 from Action dataset). Compared with Privbasis, DPARM uses less number of frequent itemsets, but it can still achieve higher accuracy of data mining. Intuitively, because the number of frequent itemsets mined by DPARM

TABLE 3. Privbasis using a single support greater than or equal to MIS (= 0.01) to mine the number of frequent itemsets, while DPARM using multiple supports to mine the number of frequent itemsets at privacy loss = 2.5.

	Privbasis			DPARM		
Dataset	# of Frequent Itemsets	Precision	MRE	# of Frequent Itemsets	Precision	MRE
Action ($K = 600$)	561	0.694	0.112	288	0.841	0.098
BMS1 ($K = 200$)	64	0.543	0.221	53	0.998	0.080
BMS2 ($K = 220$)	42	0.375	0.111	41	0.949	0.070
Kosarak ($K = 200$)	200	0.996	0.002	100	0.997	0.104
Retail ($K = 300$)	85	0.388	0.048	80	0.956	0.091

is less than Privbasis, in terms of runtime and memory usage, DPARM can be superior to Privbasis to use the generated frequent itemsets for association rule mining.

VIII. CONCLUDING REMARKS

Herein, we proposed an original differentially private association rule mining algorithm — the DPARM algorithm, which meets the challenges that had not been solved in existing works. More precisely, we dramatically reduced the noise scale by adjusting two aspects closely related to the sensitivity; in particular, we lowered the dimension of the transactional dataset as well as reducing the number of candidate itemsets. The lowering of the dimension was achieved through random truncation and uniform partition of the transactional dataset, whereas the number was reduced through a benefit achieved through the multiple support thresholds. When assigning the MIS, we applied the parallel composition theorem of differential privacy to obtain more utility. We also significantly stabilized the noise scale through adaptive allocation of the privacy budgets and used the sequential composition theorem of differential privacy to compute the overall privacy loss of the mining process, which is bounded by ex post differential privacy. Through a series of experiments, we verified the utility of the DPARM algorithm and demonstrated that our method outperforms the literature over several typical datasets in the real world.

REFERENCES

[1] R. Agrawal, T. Imieliáski, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, Jun. 1993.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. Int. Conf. Very Large Data Bases*, vol. 1215, 1994, pp. 487–499.

[3] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering frequent patterns in sensitive data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 503–512.

[4] F. Bodon, "A fast APRIORI implementation," in *Proc. FIMI*, vol. 3, 2003, pp. 1–63.

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

[6] C. Dwork, "Differential Privacy," in *Proc. Automata, Lang., Program.*, 2006, pp. 1–12.

[7] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[8] W.-Y. Day and N. Li, "Differentially private publishing of high-dimensional data using sensitivity control," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur.*, 2015, pp. 451–462.

[9] (2016). *New Regulation of The European Union on The Protection of Personal Data*. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj

[10] 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, Jan. 2004.

[11] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Jan. 2016.

[12] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proc. Int. Conf. Manage. Data*, 2011, pp. 193–204.

[13] B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 1999, pp. 337–341.

[14] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-Anonymity and l-Diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 106–115.

[15] G. Lee, U. Yun, H. Ryang, and D. Kim, "Approximate maximal frequent pattern mining with weight conditions and error tolerance," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 30, no. 6, Jul. 2016, Art. no. 1650012.

[16] G. Lee and U. Yun, "A new efficient approach for mining uncertain frequent patterns using minimum data structure without false positives," *Future Gener. Comput. Syst.*, vol. 68, pp. 89–110, Mar. 2017.

[17] Lei, Jing, "Differentially Private M-stimators," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 361–369.

[18] N. Li, W. Qardaji, D. Su, and J. Cao, "PrivBasis: Frequent itemset mining with differential privacy," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1340–1351, Jul. 2012.

[19] N. Li, M. Lyu, D. Su, and W. Yang, "Differential privacy: From theory to practice," *Synth. Lectures Inf. Secur., Privacy, Trust*, vol. 8, no. 4, pp. 1–138, Oct. 2016.

[20] S. Wu, A. Roth, K. Ligett, B. Waggoner, and S. Neel, "Accuracy first: Selecting a differential privacy level for accuracy-constrained ERM," *J. Privacy Confidentiality*, vol. 9, no. 2, pp. 2566–2576, Sep. 2019.

[21] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "*L*-diversity: Privacy beyond *k*-anonymity," in *Proc. ACM Trans. Knowl. Discovery From Data*, 2006, pp. 1–24.

[22] T. Mai, L. T. T. Nguyen, B. Vo, U. Yun, and T.-P. Hong, "Efficient algorithm for mining non-redundant high-utility association rules," *Sensors*, vol. 20, no. 4, p. 1078, Feb. 2020.

[23] R. Mendes and J. P. Vilela, "Privacy-preserving data mining: Methods, metrics, and applications," *IEEE Access*, vol. 5, pp. 10562–10582, 2017.

[24] C. N. Modi, U. P. Rao, and D. R. Patel, "Maintaining privacy and data quality in privacy preserving association rule mining," in *Proc. 2nd Int. Conf. Comput., Commun. Netw. Technol.*, Jul. 2010, pp. 1–6.

[25] V. Mayer-Schonberger and K. Cukier, "Big data: The essential guide to work, life and learning in the age of insight," in *Proc. Hachette*, 2013, pp. 1–256.

[26] M. Maruseac and G. Ghinita, "Differentially-private mining of moderately-frequent high-confidence association rules," in *Proc. 5th ACM Conf. Data Appl. Secur. Privacy*, 2015, pp. 13–24.

[27] M. Maruseac and G. Ghinita, "Precision-enhanced differentially-private mining of high-confidence association rules," *IEEE Trans. Dependable Secure Comput.*, early access, Sep. 3, 2018, doi: 10.1109/TDSC.2018.2868190.

[28] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. IEEE Symp. Secur. Privacy*, May 2008, pp. 111–125.

[29] R. M. Rogers, A. Roth, J. Ullman, and S. Vadhan, "Privacy odometers and filters: Pay-as-you-go composition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1921–1929.

[30] D. Reinsel, J. Gantz, and J. Rydning, "Data age 2025: The evolution of data to life-critical," in *Proc. Focus Big Data*, Apr. 2017, pp. 1–25.

[31] L. Sweeney, "*k*-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness, Knowl. Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[32] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets, "Using association rules for product assortment decisions: A case study," in *Proc. Knowl. Discovery Data Mining*, 1999, pp. 254–260.

[33] T. Truong, H. Duong, B. Le, P. Fournier-Viger, and U. Yun, "Efficient high average-utility itemset mining using novel vertical weak upper-bounds," *Knowl.-Based Syst.*, vol. 183, Nov. 2019, Art. no. 104847.

[34] N. Wang, X. Xiao, Y. Yang, Z. Zhang, Y. Gu, and G. Yu, "PrivSuper: A superset-first approach to frequent itemset mining under differential privacy," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 1–12.

[35] T. Wang, Z. Zheng, M. H. Rehmani, S. Yao, and Z. Huo, "Privacy preservation in big data from the communication Perspective—A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 753–778, 1st Quart., 2019.

[36] T. Wu, J. Lin, Y. Chun-Wei, C. Unil, S. Chun-Hao, B. Gautam, and X. Lv, "An efficient algorithm for fuzzy frequent itemset mining," *J. Intell. Fuzzy Syst.*, vol. 38, no. 5, pp. 5787–5797, 2020.

[37] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, "DPPro: Differentially private high-dimensional data release via random projection," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 3081–3093, Dec. 2017.

[38] U. Yun, G. Lee, and K.-M. Lee, "Efficient representative pattern mining based on weight and maximality conditions," *Expert Syst.*, vol. 33, no. 5, pp. 439–462, Oct. 2016.

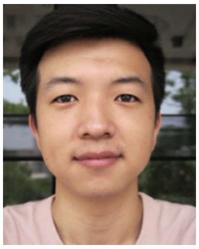
[39] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 3, pp. 372–390, Oct. 2000.

[40] Z. Zheng, R. Kohavi, and L. Mason, "Real world performance of association rule algorithms," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 401–406.

[41] C. Zeng, J. F. Naughton, and J.-Y. Cai, "On differentially private frequent itemset mining," *Proc. VLDB Endowment*, vol. 6, no. 1, pp. 25–36, Nov. 2012.

[42] L. Zhang, W. Wang, and Y. Zhang, "Privacy preserving association rule mining: Taxonomy, techniques, and metrics," *IEEE Access*, vol. 7, pp. 45032–45047, 2019.

YAO-TUNG TSOU (Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan. He was a Research Assistant with the Institute of Information Science, Academia Sinica, Taipei, from 2009 to 2012. He joined the Research Center for Information Technology Innovation (CITI), Academia Sinica, in 2013, as a Research Assistant. Then, he joined CITI, Academia Sinica, in 2015, as a Postdoctoral Research Fellow. He was a Visiting Scholar with CITI, Academia Sinica, in 2018. He is currently an Associate Professor with the Department of Communications Engineering, Feng Chia University, Taichung, Taiwan. He is also a Consultant at Lin Dan Technology Inc. and Swiss Innovation Valley. His research interests include data privacy and security, physical unclonable function, embedded systems, and STT-MRAM applications. He received the Magnetism Research Prize from the Taiwan Association for Magnetic Technology, in 2019.



HAO ZHEN received the B.S. degree from the School of Electronic and Information Engineering, Beijing Jiaotong University, in 2017, and the M.S. degree in electrical engineering from National Taiwan University, in 2019. His research interests include data privacy and security, and cryptography.

XIYU JIANG received the B.S. degree from the Department of Communications Engineering, Feng Chia University, Taichung, Taiwan, in 2018. He is currently pursuing the degree in electrical engineering with National Taiwan University. His research interests include data privacy and security, and the Internet of Things.

YENNUN HUANG (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Maryland. He Joined AT&T Bell Labs, in 1989. He worked on Software Implemented Fault Tolerance (SwiFT) tools, which have been applied to tens of telecommunication systems in AT&T, and SwiFT was named one of the ten major technology breakthroughs in Bell Laboratories, in 1992. He became a Distinguished Member of Technical Staff at Bell Labs, in 1996. He started the Dependable Computing Research Department, AT&T, in 1999, and was the Department Head of the organization to ensure the high dependability of all AT&T services. He became the VP of engineering of PreCache Inc., a Sony subsidiary, in 2001, to create a multi-media content delivery platform. In late 2004, he returned to AT&T and later became the Executive Director of the Dependable Distributed Computing and Communication Research Department to lead research on Digital Content Management and IPTV research programs. In 2007, he went to Taiwan and became the Executive Vice President of the Institute for Information Industry, a government funded research and development organization with more than 1800 employees. From 2008 to 2011, he was the President of VeeTIME Company to build quadruple-play telecom services, including cable TV, FTTx, NGN voice, and 4G Wimax using an all-IP network in central and South Taiwan. He joined the Research Center for Information Technology Innovation (CITI), Academia Sinica, in 2011, as the CEO of the Security Research Center. He is currently the Deputy Executive Secretary of the Taiwan Office of Science and Technology of Executive Yuan, helping Taiwan government on the Information and Communication Technology (ICT) Research and Development policy and funding allocation. He has coauthored the first paper on software aging and rejuvenation in the 1995 Fault Tolerant Computing Symposium (FTCS95). He was elected as an IEEE Fellow, in 2012, for his contributions on fault tolerant and fault avoidance software.

SY-YEN KUO received the B.S. degree in electrical engineering from National Taiwan University (NTU), Taipei, Taiwan, in 1979, the M.S. degree in electrical and computer engineering from the University of California at Santa Barbara, in 1982, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, in 1987. He was a Faculty Member with the Department of Electrical and Computer Engineering, The University of Arizona, from 1988 to 1991, and an Engineer at Fairchild Semiconductor and Silvar-Lisco, CA, USA, from 1982 to 1984. In 1989, he was also a Summer Faculty Fellow at the Jet Propulsion Laboratory, California Institute of Technology. He spent his sabbatical years as a Visiting Professor at The Hong Kong Polytechnic University, from 2011 to 2012, and at The Chinese University of Hong Kong, from 2004 to 2005, and as a Visiting Researcher with AT&T Labs-Research, NJ, USA, from 1999 to 2000, respectively. He was the Dean of the College of Electrical Engineering and Computer Science, NTU, from 2012 to 2015, and the Chairman of the Department of Electrical Engineering, NTU, from 2001 to 2004. He also took a leave from NTU and served as a Chair Professor and the Dean of the College of Electrical Engineering and Computer Science, National Taiwan University of Science and Technology, from 2006 to 2009. He is currently the Pegatron Chair Professor at the Department of Electrical Engineering, National Taiwan University. He has published more than 400 articles in journals and conferences and also holds 21 U.S. patents, 19 Taiwan patents, and 10 patents from other countries. His current research interests include dependable systems and networks, mobile computing, cloud computing, and quantum computing and communications. He received the Distinguished Research Award and the Distinguished Research Fellow award from the National Science Council, Taiwan. He was also a recipient of the Best Paper Award in the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/ACM Design Automation Conference (DAC), the National Science Foundation's Research Initiation Award, in 1989, and the IEEE/ACM Design Automation Scholarship, in 1990 and 1991.