Privacy-Preserving Joint Distribution Analysis for Set-Valued Data via Local Differential Privacy

Yilin Wang, Xiong Li, *Senior Member, IEEE*, Shuai Shang, *Student Member, IEEE*, Wei Liang, Jinjun Chen, *Fellow, IEEE*, Xiaosong Zhang, *Senior Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—Set-valued data, an important data form expressing diversity and uncertainty, is widely used in fields such as recommendation systems and social network analysis. However, such data often contains fine-grained records, which may lead to the leakage of users' sensitive information. To this end, some privacy-preserving set-valued data analysis schemes have been proposed. This paper first proposes a joint shift inference attack against the cyclic shift-based local differential privacy (LDP) protocol introduced by Huang et al., which exploits deterministic cyclic-shift patterns and significant frequency differences to infer the user's original data. Experimental results demonstrate that the user's original data can be inferred with a probability exceeding 96%. Theoretically, the cyclic-shift mechanism violates the core requirement of local differential privacy due to its non-surjective output space. To overcome the limitations of existing schemes, we propose a privacy-preserving joint distribution analysis scheme for set-valued data via LDP (SVJDA). It employs the Sparse Vector Mean Estimation (SVME) mechanism and utilizes a sign-based hashing function to compress data, allowing privacy-preserving joint distribution analysis while introducing only minimal additional noise. Theoretical analysis shows that SVJDA satisfies ϵ-LDP with minimal estimation error. The experimental results confirm that SVJDA achieves higher accuracy in joint distribution estimation while ensuring the accuracy of frequent itemset identification. For $\epsilon \in [0.4, 1]$, the L_∞ error of SVJDA is only 7.208% – 21.725% of SVSM and 2.821% – 7.279% of LDP-RM, while its MSE is 0.00364% – 2.338% of SVSM and 0.017% – 0.068% of LDP-RM, demonstrating its superior performance.**

***Index Terms*—Local differential privacy, joint distribution analysis, set-valued data, privacy-preserving, inference attack.**

I. Introduction

SET-VALUED data is a type of data in which each element is represented by a set of multiple possible values rather than a single value, reflecting variability or uncertainty. It is widely used in fields such as recommendation systems and social network analysis that require modeling diverse or ambiguous information. For sensitive data in areas such as health, finance, or social media, ensuring privacy protection while enabling effective data utilization has become a critical research challenge. Local Differential Privacy (LDP) [1] offers a practical paradigm in which users locally perturb their data before sharing, thereby preventing the data collector from accessing sensitive personal information. Numerous LDP-based protocols have been proposed across diverse application domains, including social networks [2], [3], [4], data mining [5], [6], [7], location-based services [8], [9], edge computing [10], [11], [12], and machine learning [13]. However, most existing LDP studies primarily focus on basic statistical metrics that do not require consideration of correlations between data, such as frequency estimation, heavy hitter identification, and range queries.

For set-valued data, traditional differential privacy (DP) mechanisms face significant challenges in simultaneously ensuring privacy and maintaining the intrinsic relationships within the data. Consequently, most existing LDP techniques for set-valued data primarily concentrate on frequent item or itemset mining. In 2016, Qin et al. [14] proposed a two-phase framework utilizing a "padding and sampling" technique to reduce communication overhead and utility loss when processing set-valued data. However, it cannot perform frequent itemset mining. Subsequently, Wang et al. [15] introduced the Padding-and-Sample-based Frequency Oracle protocol (PSFO) and proposed the Set Value Item Mining (SVIM) and Set Value ItemSet Mining (SVSM) protocols to improve the efficiency of frequent itemset mining. More recently, Huang et al. [16] proposed a cyclic shift-based LDP protocol for joint distribution analysis of set-valued data while preserving data correlations, but its security has not yet been thoroughly examined.

Although several LDP protocols for set-valued data analysis have been proposed, they still have certain limitations. First, most studies focus on heavy-hitter identification (or frequent-item mining), which is only applicable to scenarios involving the computation of frequent-item frequencies. Second, most existing approaches are confined to single-item frequency estimation and do not support joint distribution estimation. Furthermore, the security of the joint distribution estimation protocol proposed by Huang et al. [16] has not been well validated, posing potential risks of user privacy leakage. Therefore, achieving accurate and privacy-preserving joint

Received 15 February 2025; revised 3 December 2025; accepted 12 January 2026. Date of publication 16 January 2026; date of current version 26 January 2026. This work was supported in part by the National Natural Science Foundation of China under Grant U25A20429 and Grant 62332018 and in part by Sichuan Provincial Natural Science Foundation under Grant 2025ZNSFSC0497. The associate editor coordinating the review of this article and approving it for publication was Dr. Jun Zhao. *(Corresponding author: Xiong Li.)*

Yilin Wang, Xiong Li, Shuai Shang, and Xiaosong Zhang are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: wangyilin1221@gmail.com; lixiong@uestc.edu.cn; shshang180@gmail.com; johnsonzxs@uestc.edu.cn).

Wei Liang is with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: wliang@hnust.edu.cn).

Jinjun Chen is with the Swinburne University of Technology, Melbourne, VIC 3122, Australia (e-mail: jinjun.chen@gmail.com).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TIFS.2026.3654799

distribution estimation for set-valued data under LDP remains a significant challenge.

A. Related Work

Since there are few studies on the joint distribution analysis of set-valued data under LDP, we review the work most relevant to this research topic from the following four aspects.

1) Heavy Hitters of Set-Valued Data: In 2016, Qin et al. [14] introduced the first solution for identifying heavy hitters in set-valued data, known as LDPMiner. This approach addresses the challenge of varying item counts among users by padding each user's items to a uniform length. However, it is only suitable for frequent item mining. Building on LDPMiner, Wang et al. [15] formally defined PSFO and proposed two protocols, SVIM and SVSM, for mining frequent items and frequent itemsets in set-valued LDP scenarios, respectively. Subsequently, Ma et al. [17] enhanced frequent itemset mining by integrating the FP-tree structure with the Hadamard response mechanism, thereby reducing communication costs and improving accuracy. However, this method requires generating a uniform Hadamard matrix for each user, which increases memory and space complexity. To further improve accuracy and utility in top-k frequency estimation, Wang et al. [18] proposed the EPS^2 framework utilizing the Shuffle model. Additionally, Zhu et al. [19] developed PemSet, which perturbs and reports only the prefix of user data, thereby reducing computational costs while efficiently identifying heavy hitters in large-domain set-valued data.

2) Frequency Estimation of Set-Valued Data: For frequency estimation, it is necessary not only to identify frequent items but also to estimate the frequencies of all items. In 2018, Wang et al. proposed PrivSet [20], which perturbs the entire set-valued data as a whole to avoid splitting the privacy budget. However, its experiments were limited to small item domains. In 2020, Wang et al. introduced the wheel mechanism [21], designed for high-dimensional set-valued data and adaptable to large-scale real-world applications. Nonetheless, these studies remain restricted to frequency estimation for individual items. In 2022, Zhou et al. [22] studied the k-sparse version of the vector mean estimation problem and developed a mechanism that achieves asymptotically optimal error and efficient communication under user-level or event-level LDP. This mechanism is referred to in this paper as Sparse Vector Mean Estimation (SVME).

3) Key-Value Collecting Under LDP: In key-value pair data, correlations between keys and values should be preserved during the perturbation process. In 2019, Ye et al. [23] proposed three methods, PrivKV, PrivKVM, and PrivKVM$^+$, which maintain key-value correlations while satisfying the LDP privacy guarantees. In 2020, Gu et al. [24] introduced PCKV to address the challenges posed by large key domains in PrivKVM by leveraging the padding-and-sampling protocol. However, selecting an optimal padding length is infeasible. In 2021, Ye et al. [25] introduced PrivKVM*, which employs adaptive sampling techniques to effectively mitigate the sampling challenges in large key domains, achieving higher estimation accuracy than existing schemes. These studies mainly focus on preserving the correlation between the key and the value within a single record, whereas set-valued data consist of multiple elements and thus require maintaining the internal correlations among them. Therefore, although key-value collection schemes offer useful insights into correlation preservation, their mechanisms cannot be directly extended to the joint analysis of set-valued data.

4) Relation Mining Under DP: Some studies in this area have been conducted under centralized differential privacy constraints rather than LDP. In 2012, Li et al. [26] utilized a trusted third-party publisher to perturb multiple high-support itemsets, from which the top-k subsets were extracted. In 2020, Maruseac et al. [27] proposed a high-confidence relationship mining approach that enables association rule mining on low to moderate-support itemsets. However, these schemes are based on centralized differential privacy, which requires centralized processing of user data and poses a risk of data leakage. In 2024, Dong et al. [28] proposed LDP-RM, the first relation mining approach under LDP, leveraging singular value decomposition and low-rank approximation to reduce the number of estimates, thereby enhancing the accuracy of final estimates.

B. Motivation

Although several LDP protocols for data analysis have been proposed, none are capable of simultaneously achieving the following goals: 1) *Joint Distribution Analysis*: Most existing schemes support only heavy hitter identification or frequency estimation of individual items, lacking the capability to comprehensively estimate the joint frequency of multiple items, which limits their applicability in complex data analysis. 2) *High Accuracy*: The number of joint distribution estimates increases quadratically with the number of items, amplifying the impact of noise and significantly degrading estimation accuracy. 3) *Privacy-Preserving*: Some studies, such as Huang et al.'s work [16], exhibit security vulnerabilities, failing to strictly satisfy the requirements of LDP and effectively protect user privacy.

C. Main Idea and Contributions

To address the aforementioned challenges, we propose a privacy-preserving joint distribution analysis protocol for set-valued data via LDP. The core ideas are as follows: First, we estimate the frequency of each item in the set-valued data and construct an initial joint distribution matrix accordingly. Next, we refine the frequency estimation for items with higher frequencies and update the initial joint distribution matrix based on the newly obtained estimates. To update the matrix, the aggregator collects the perturbation results from users. During the collection process, the set-valued data owned by users are treated as sparse vectors, and the SVME protocol [22] is employed to perturb and aggregate user data, thereby enhancing the accuracy of frequency estimation. Specifically, this paper has the following contributions:

1) We propose a joint shift inference attack against cyclic shift-based LDP protocol introduced by Huang et al. [16], which utilizes significant frequency differences to infer the user's original data. Experiments show that the attack can infer the user's original data with a probability

TABLE I
NOTATIONS

Notation	Description
$x, \boldsymbol{x}, X, d$	Item, itemset, set of all items, domain size of X
$\epsilon, \Psi(\cdot), \Phi(\cdot)$	Privacy budget, perturbation algorithm, aggregation algorithm
U^1, U^2	Two distinct sets within the item domain of user data
n, n_1, n_2	Number of total users, users in the $1^{st}, 2^{nd}$ group
u_i	The i-th user
$\boldsymbol{v}_i^k$	The set-valued data (binary vector) of user i, from set k
$\hat{\boldsymbol{s}}_i^k$	The perturbed set-valued data of user i, from set k
c	The cardinality of the set
l	The parameter of the cycle shift function
p, q	Perturbation probabilities for choosing the original and shifted data
S^k	The perturbed data matrix of set k
S^*	The aggregated perturbed data matrix
$f^*_{i,j}, \hat{\boldsymbol{f}}_{i,j}, f_{i,j}$	The perturbed/estimated/true joint frequency of item i in the first set and item j in the second set
$f_i^*, \hat{f}_i, f_i$	The perturbed/estimated/true frequency of item i in the set
$\hat{\boldsymbol{f}}, \boldsymbol{f}^*$	The calibrated/perturbed frequency matrix
M	The calibration matrix
$\hat{\boldsymbol{v}}_i$	The estimated set-valued data of user i
$\ell, L, u(L)$	Padding length, sparsity parameter, and update factor
b	Number of hash bins
$\eta_{i,j}, \Delta_{i,j}$	The clipping range and noise parameter for the j-th phase of task i
$B_i, \bar{B}_i, \widetilde{B}_i$	The true/clipped/perturbed bucket value of i-th user
h_i	Hash function of i-th user
$\hat{\boldsymbol{v}}, \bar{\boldsymbol{v}}$	The estimated/true mean of user's set-valued data
Adv, Usr	Adversary and User
I	Adv's auxiliary information
τ	Confidence threshold
S, IS	Candidate set, candidate set of itemsets
$\boldsymbol{v}_S, \boldsymbol{v}_{IS}$	The set-valued data corresponding to the sets S and IS
S_t, S_e	The true/estimated frequent itemsets

of more than 96%, proving that the cyclic shift-based LDP protocol is not effective in ensuring data security.

2) We design a privacy-preserving set-valued joint distribution analysis protocol (SVJDA) via LDP. The protocol leverages the two-phase framework of SVSM [15] for joint distribution estimation and adopts the SVME protocol [22] for data perturbation to add smaller noise. SVJDA can accurately identify frequent itemsets when performing joint distribution analysis.
3) Theoretical analysis and experimental results demonstrate that SVJDA achieves the minimal estimation errors. For privacy budget $\epsilon \in [0.4 - 1]$, SVJDA's L_∞ error is only $7.208\% - 21.725\%$ of SVSM [15] and $2.821\% - 7.279\%$ of LDP-RM [28], while its MSE is $0.00364\% - 2.338\%$ of SVSM [15] and $0.017\% - 0.068\%$ of LDP-RM [28].

The remainder of this paper is structured as follows: Section II introduces the preliminaries used in our scheme. Section III briefly reviews Huang et al.'s scheme and outlines the corresponding attack. Section IV introduces our proposed scheme, SVJDA. Sections V and VI focus on the security analysis and performance evaluation of SVJDA, respectively. Finally, Section VII concludes the paper.

II. PRELIMINARIES

This section introduces some preliminaries used in this work, including local differential privacy, set-valued joint distribution analysis, and sparse vector aggregation under LDP. The notations used throughout this paper are summarized in Table I.

A. Local Differential Privacy (LDP)

The LDP [29] is a variant of DP that perturbs data locally on the client side, preventing the server from directly accessing the user's original data. This mechanism provides stronger privacy protection, making it particularly suitable for decentralized data collection scenarios. Below we review the formal definition of LDP.

Definition 1 (ϵ-LDP): An algorithm $\Psi(\cdot)$ satisfies ϵ-local differential privacy (ϵ-LDP), where $\epsilon \geq 0$ is the privacy budget, iff for any two inputs $x, x' \in X$, we have

$$\forall y \in \text{Range}(\Psi) : \Pr[\Psi(x) = y] \leq e^{\epsilon} \Pr[\Psi(x') = y],$$

where Range(Ψ) *denotes the output range of the algorithm* Ψ.

The core idea of LDP is that any output y should be possible regardless of the algorithm's inputs. The degree of "whatever" is controlled by the privacy budget ϵ [16].

B. Set-Valued Joint Distribution Analysis

Set-valued data refers to data records where each instance consists of a set of items, rather than a single numerical value or scalar [15]. Let $U = \{x_1, x_2, \ldots, x_c\}$ represent the universal set of items, and user i's set-valued data v_i is denoted as a subset of U, i.e., $v_i \subseteq U$. Different users may have different numbers of items, thus their set-valued data can be different subsets of the universal set. Set-valued data analysis aims to estimate statistical information from the set-valued data collected from all users. In this paper, the purpose of the set-valued joint distribution analysis is to analyze the frequency of combining any two items in the set, and the joint distribution is defined as:

$$f_{i,j} = \frac{|\{u_k \mid v_k[i] = 1 \wedge v_k[j] = 1\}|}{n},$$

here, $f_{i,j}$ denotes the proportion of users who have both item i and item j at the same time, and $v_k[i]$ and $v_k[j]$ denote the i-th and j-th bits of user k's data vector v_k, respectively, indicating whether items i and j are included in user k's set-valued data.

C. Sparse Vector Aggregation Under LDP

The user's vector $v_i \in [-1, 1]^d$ is usually sparse in many practical applications. A d-dimensional vector v_i is said to be L-sparse if the non-zero coordinates of the vector do not exceed L. Different users may have different sparse patterns, i.e., the location of the non-zero coordinates can be different [22]. The task of sparse vector aggregation under LDP can be formalized as an LDP protocol $\mathcal{S}$, which consists of a pair of algorithms defined as $\mathcal{S}(\epsilon) \triangleq \langle \Psi, \Phi \rangle$, here each user locally disturbs the input values using the perturbation algorithm Ψ, and the aggregator extracts useful information from the disturbed data using the aggregation algorithm Φ.

The SVME protocol [22] introduces a mechanism applicable to L-sparse by integrating the 1-sparse mechanism proposed by Wang et al. [15] with the random binning method, and proves that the optimal number of hash bins b is $b = 1$ under user-level LDP. Specifically, under user-level LDP, each user perturbs and uploads only one bucket value, and increasing b linearly amplifies the variance of the injected

Laplace noise without reducing the aggregation error. Hence, fixing $b = 1$ minimizes the overall mean-square error while maintaining the same asymptotic utility bound. The SVME protocol is specified by the following parameters: the sparsity parameter L, the cropping range η, the noise parameter Δ, and the privacy budget ϵ.

In SVME [22], each user u_i has the input vector $\boldsymbol{v}_i$ and perturbs $\boldsymbol{v}_i$ using the perturbation algorithm $\Psi_{\text{SVME}(L,\eta,\Delta,\epsilon)}(\boldsymbol{v}_i)$:

1) Randomly select a hash function $h_i : [d] \to \{-1, 1\}$, where $[d] = \{1, 2, \cdots, d\}$ denotes the set of coordinate indices of the d-dimensional input vector v_i. This hash function assigns a random sign to every coordinate.
2) Each user maps their data to a bucket, and the coordinate values of all data mapped to the bucket are summed up with sign-based weighting:
$$B_i = \sum_{j \in [d]} h_i(j) v_i[j].$$
3) Restrict B_i to the range $[-\eta, \eta]$:
$$\bar{B}_i = \text{clip}_{[-\eta,\eta]}(B_i).$$
4) Add Laplace noise to the clipped bucket value:
$$\widetilde{B}_i = \bar{B}_i + \text{Lap}\left(\frac{\Delta}{\epsilon}\right).$$
5) Send hash functions h_i and bucket values $\widetilde{B}_i$ to the aggregator.

The aggregator performs mean estimation based on the data sent by all clients, and the aggregation algorithm Φ_{SVME} is defined as follows:

1) For each $x \in [d]$, estimate the mean value of the coordinate x using the bucket values sent by the clients and the corresponding hash functions:
$$\hat{v}[x] = \frac{1}{n} \sum_{i \in [n]} h_i(x) \cdot \widetilde{B}_i. \tag{1}$$
2) Calculate the vector mean $\hat{\boldsymbol{v}}$.

III. Review and Attack on Huang's Scheme

In this section, we briefly review Huang et al.'s [16] joint distribution analysis protocol for set-valued data. Subsequently, we introduce our Joint Shift Inference Attack and evaluate its performance through experimental analysis.

A. Review and Analysis of Huang's Scheme

1) Review: Huang et al. [16] utilized a client-aggregator architecture consisting of an aggregator and a group of users. Each client represents a user who possesses set-valued data belong to two distinct sets U^1 and U^2, where $U^1 = \{x_1^1, x_2^1, \ldots, x_c^1\}$ and $U^2 = \{x_1^2, x_2^2, \ldots, x_c^2\}$ are two different item domains. The data of user u_i denoted as two subsets, $s_i^1 \subseteq U^1$ and $s_i^2 \subseteq U^2$. Their scheme employs a cyclic shift-based LDP protocol to protect privacy during the joint distribution estimation. We review this scheme in three stages.

a) Encoding: U_i encodes the set-valued data s_i^1 and s_i^2 into binary bit strings $\boldsymbol{v}_i^1$ and $\boldsymbol{v}_i^2$ with a length equal to the cardinality of sets U^1 and U^2, respectively. Specifically, for each item j in the set U^k, $k = 1, 2$, if U_i owns the j-th item, $v_i^k[j] = 1$; otherwise, $v_i^k[j] = 0$. Finally, u_i's set-valued data can be encoded as two binary vectors $\boldsymbol{v}_i^1$ and $\boldsymbol{v}_i^2$, respectively.

b) Perturbation: For each set-valued data $\boldsymbol{v}_i^k$ of the user u_i, the perturbed set-valued data $\boldsymbol{s}_i^k$ is generated by applying the following perturbation rule:

$$\boldsymbol{s}_i^k = \begin{cases} \boldsymbol{v}_i^k, & \text{with probability } p = \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}, \\ \text{circshift}(\boldsymbol{v}_i^k, c - l), & \text{with probability } q = \frac{1}{e^{\epsilon/2}+1}. \end{cases} \tag{2}$$

Here, p and q are the perturbation probabilities that determine whether the user uploads the original set-valued vector or its cyclically shifted version, respectively.

The circular shift operation, which shifts the binary bit string to the left or right by a fixed length l, is used to perturb the set-valued data while preserving its internal correlation. The shift parameter l is uniformly determined by the aggregator and satisfies $0 < l < c$, $l \nmid c$, where c is the length of the bit string, i.e., the cardinality of the set.

c) Aggregation: The perturbed data are stored as matrices S^1 and S^2, where each matrix contains the users' perturbed results corresponding to the two item sets. A union matrix S^* is then derived from S^1 and S^2 through cyclic shifting and Boolean AND operations. Each element at position (i, j) in the matrix represents the number of users in the perturbed set-value data who simultaneously possess item i in set U_2 and item j in set U_1. Dividing this user count by the total number of users n yields the observed perturbed joint frequency $f_{i,j}^*$. Using the perturbed data, the joint frequency estimate is adjusted by the following formula:

$$f_{i,j}^* = \hat{f}_{i,j} \cdot p^2 + (\hat{f}_{i,j+l} + \hat{f}_{i+l,j}) \cdot p \cdot q + \hat{f}_{i+l,j+l} \cdot q^2, \tag{3}$$

where $\hat{f}_{i,j}$ and $f_{i,j}^*$ represent the estimated joint frequency and the perturbed joint frequency of itemset (i, j), respectively. Since the above equation can be interpreted as $\boldsymbol{f}^* = M \cdot \hat{\boldsymbol{f}}$, where the aggregator applies matrix division to solve the recursive equation outlined in Eq. (3). The calibration matrix M is constructed from the coefficients corresponding to each true joint frequency on the right-hand side. Subsequently, the matrix left division is performed to obtain the calibrated joint distribution $\hat{\boldsymbol{f}}$.

2) Analysis: Through careful analysis of Huang et al.'s protocol [16], we derive the following two key observations.

(1) *The Shift Operation Lacks Sufficient Randomization*. The protocol relies on cyclic shifts, resulting in a strong correlation between the perturbed and original data. This correlation allows an attacker to infer users' original data within a domain far smaller than the full data space, thereby leading to potential privacy breaches. For instance, if the perturbed data is $\{1, 0, 0, 1\}$, the original data can only be one of the following: $\{1, 0, 0, 1\}$, $\{1, 1, 0, 0\}$, $\{0, 1, 1, 0\}$, or $\{0, 0, 1, 1\}$, instead of the complete set of 2^4 random permutations. Specifically, when the shift parameter l is publicly known, the user's original data is only two possible candidates, regardless of the cardinality of the set. This significantly increases the server's ability to infer the original data with high probability.

(2) *Failure to Satisfy LDP*. The perturbed outputs generated by Huang et al.'s protocol [16] fail to cover all possible outputs of the perturbation. This violates the fundamental requirement of DP, which mandates that for any two inputs x and x', and any possible output y, the probability of producing y under the perturbation algorithm Ψ should satisfy: $\Pr[\Psi(x) = y] \leq e^{\epsilon} \Pr[\Psi(x') = y]$. However, due to the constraints of the shift operation, the flipping probability of each bit in the binary vector depends on whether the shifted data remains identical to the original data. If they are identical, the flip probability is 0. Consequently, there exists an output y that cannot be generated from the perturbed input x, resulting in a probability of 0 for such cases. Even if the shift parameter l is not publicly known and is instead dynamically negotiated between the user and the server, the deterministic structure of the cyclic shift still causes certain input–output pairs (x, y) to have zero transition probability. This implies that the protocol can only satisfy differential privacy when the privacy budget ϵ approaches infinity. Therefore, the shift-based LDP protocol fails to meet the requirements of ϵ-LDP.

B. Shift Inference Attack of Huang Et Al.'s Scheme

Based on the above observations, we propose a Joint Shift Inference Attack on Huang et al.'s protocol [16] using frequency analysis, where significant frequency variations in certain cases allow an attacker to leverage the frequency distribution after cyclic shifting to infer the original data. We first introduce an attack model and then provide a detailed description of the attack.

1) Formalizing the Shift Inference Attack Model: Formally, the shift inference attack model is defined as a game between an adversary (Adv) and a target user (Usr), where Usr perturbs its data using an LDP protocol A.

Shift Inference Game.

- Step 1. Usr perturbs the original set-valued data $\boldsymbol{v}_i$ using the cyclic shift-based LDP protocol A to generate the perturbed set-valued data s_i.
- Step 2. Usr sends the perturbed set-valued data s_i to Adv.
- Step 3. Adv applies the attack method and outputs an estimated set-valued data, denoted as $\hat{v}_i$, which represents the adversary's inferred original data of Usr.

Adv wins the game if $\hat{v}_i = v_i$.

To further clarify the attack model, we distinguish between two canonical adversaries:

- **Passive Adversary (Honest-but-Curious Adversary):** This adversary strictly follows the protocol during data collection and aggregation but attempts to infer the original data by analyzing the perturbed results. This model corresponds to an *honest-but-curious server* in practical deployments and represents the primary adversary type considered in this work.
- **Active Adversary:** Beyond the capabilities of a passive adversary, this adversary may forge inputs, issue repeated queries, or manipulate subsets of clients to induce additional information leakage.

This work mainly focuses on the passive adversary assumption, demonstrating that even under correct protocol execution, the cyclic-shift mechanism exhibits structural vulnerabilities that can be exploited.

Adversary's Background Knowledge. The sole knowledge Adv possesses about Usr is the perturbed data s_i, called *observations*. Furthermore, we assume that Adv knows the universal set U, the privacy budget ϵ, and any internal parameters of the protocol A, a standard assumption in attack models where system parameters are typically considered public.

2) Joint Shift Inference Attack: We instantiate the above attack model for Huang et al.'s scheme [16] to infer whether a user has perturbed the data by estimating the joint probability of the user's set-valued data.

For honest-but-curious servers, the distribution estimates for each item can be aggregated to compute the overall distribution after collecting perturbed data from users. Assuming that items are independent of each other, the server can compute the occurrence probability of the received user data based on the independence assumption. If the user data has been perturbed, the server can compute the occurrence probability of the original data by inversely shifting it by l bits. The server then selects the result with the higher occurrence probability as an estimate of the user's set-valued data. Fig. 1 provides a schematic representation of the attack process with u_1's data as an example when the protocol applies a 3-bit left shift.

Step 1: **Estimating the frequency distribution of individual items.** The attacker collects the perturbed set-value data from all users into S^1 and S^2, and estimates the frequency of individual items based on the perturbation algorithm and has:

$$f_i^* = \hat{f}_i \cdot p + \hat{f}_{i+l} \cdot q, \tag{4}$$

where $\hat{f}_i$ and f_i^* are the calibrated frequency and perturbed frequency of item i, respectively. Similar to Huang et al.'s protocol [16], the recursive equation in Eq. (4) can be solved using matrix division, as described in Algorithm 1. The attacker first generates the calibration matrix M based on the coefficients before the calibration frequency vector. Subsequently, the calibrated frequency vector $\boldsymbol{f}^*$ is obtained by performing a left division of the perturbed frequency vector $\hat{\boldsymbol{f}}$ by the matrix M.

Step 2: **Calculate the joint probability.** Under the assumption of item independence, the aggregator can derive the occurrence probability of the user's original set-valued data using the calibrated frequency vector $\hat{\boldsymbol{f}}$. Specifically, if the user's perturbed data is not subjected to a shift operation, the original data $\boldsymbol{v}_i$ can be directly inferred as the perturbed data $\boldsymbol{s}_i$. The occurrence probability of user u_i's original data $\boldsymbol{v}_i$ is expressed as $\Pr(\hat{\boldsymbol{v}}_i = \boldsymbol{s}_i) = \prod_j (\hat{f}_j)^{\boldsymbol{s}_i[j]}$, where $\boldsymbol{s}_i[j]$ denotes the j-th bit of user i's data $\boldsymbol{s}_i$, indicating whether item j is included in user u_i's set-valued data. Conversely, if the user's perturbed data undergoes a shift operation, the probability is calculated as $\Pr(\hat{\boldsymbol{v}}_i = \text{circshift}(\boldsymbol{s}_i, l)) = \prod_j (\hat{f}_j)^{\text{circshift}(\boldsymbol{s}_i, l)[j]}$.

Step 3: **Select the most probable original data of the user.** Based on the two joint probabilities calculated in Step 2, the adversary can select the data with the higher probability as the inferred result $\hat{\boldsymbol{v}}_i$:

$$\hat{\boldsymbol{v}}_i = \begin{cases} \boldsymbol{s}_i, \text{if } \Pr(\hat{\boldsymbol{v}}_i = \boldsymbol{s}_i) > \Pr(\hat{\boldsymbol{v}}_i = \text{circshift}(\boldsymbol{s}_i, l)), \\ \text{circshift}(\boldsymbol{s}_i, l), \text{otherwise.} \end{cases}$$

Algorithm 1 Frequency Calibration Mechanism

Input: Perturbed frequency vector f^*, Privacy budget ϵ
Output: Calibrated frequency vector $\hat{f}$
Data: $M = [0]_{c\times c}$; p and q as per Eq. (2)
for $i = 1$ **to** c **do**
  $M(i,i) = p$;
  $k = (i + l) \mod c$;
  $M(i,k) = q$;
$\hat{f} = M \backslash f^*$;
return $\hat{f}$

Step 4: **Output inference results based on the confidence level.** The attacker calculates the confidence level of each inference result to quantify its reliability as follows.

$$\text{conf}(\hat{\boldsymbol{v}}_i) = \frac{\max(\Pr(\hat{\boldsymbol{v}}_i = \boldsymbol{s}_i), \Pr(\hat{\boldsymbol{v}}_i = \text{circshift}(\boldsymbol{s}_i, l)))}{\Pr(\hat{\boldsymbol{v}}_i = \boldsymbol{s}_i) + \Pr(\hat{\boldsymbol{v}}_i = \text{circshift}(\boldsymbol{s}_i, l))}.$$

A confidence value $\text{conf}(\hat{\boldsymbol{v}}_i)$ close to 1 reflects high reliability of the inference result. The attacker can compare $\text{conf}(\hat{\boldsymbol{v}}_i)$ to a predefined threshold τ to determine the final inference result, i.e., if $\text{conf}(\hat{\boldsymbol{v}}_i) \geq \tau$, the attacker outputs $\hat{\boldsymbol{v}}_i$; otherwise, "null" is returned. The attack is successful if $\hat{\boldsymbol{v}}_i = \boldsymbol{v}_i$.

Scope of application and Limitations. Our Joint Shift Inference Attack is specifically designed for perturbation mechanisms based on cyclic shifts. It leverages the deterministic shift correlation between the original and perturbed data. Therefore, it cannot be directly applied to other types of local differential privacy mechanisms, e.g., random response, whose perturbations are fully random and lack similar structural correlations.

C. Evaluation

We evaluate the practical effectiveness of our joint shift inference attack by conducting experiments on real-world datasets. The source code is available on Github.[1]

1) Datasets: We only conduct experiments on the following real-world datasets, as synthetic datasets generated based on predefined distributions lack individual user preferences. The frequency of each item after aggregation exhibits no significant differences, making it challenging to determine whether users have performed shifting operations on the data.

- **Online Retail (Retail):** [30] This dataset encompasses all transactions recorded by a UK-based online retailer from 2009 to 2011. The product names are treated as the items owned by each user, resulting in a total of 4,459 items and 4,383 users.
- **Takeaway Food Orders (Food):** [31] This dataset comprises order records from two Indian takeaway restaurants located in London. Food names are treated as the items associated with each user, yielding two datasets: Food-1 contains 248 items and 13,397 users, while Food-2 includes 337 items and 19,658 users.

[1] https://github.com/vvYl/SVJDA/tree/master/attack

2) Measurement: To evaluate the accuracy of the attack results, we adopt the methodology outlined in previous work [32]. For a given confidence threshold τ, users with an attack confidence score below τ are designated "null users", i.e., users for whom the adversary refrains from making inferences. The effectiveness of the attack is evaluated using the following two metrics.

- **Null Rate:** The proportion of users classified as null users relative to the total users for a given threshold τ.
- **Precision:** The proportion of accurate inferences among all users who are not classified as null.

3) Baseline: The baseline attack does not perform any inferences and directly treats the perturbed results as the original data. Consequently, the precision of the baseline attack is equal to the perturbation probability p, which is related to the privacy budget ϵ.

4) Results: Fig. 2 illustrates the variation of attack precision and null rate as the privacy budget ϵ changes, with the threshold τ set to 0.3 and 0.7. As the figure shows, our attack consistently achieves the highest attack precision and a lower null rate across all datasets.

The level of privacy protection in LDP is determined by ϵ, i.e., the larger ϵ, the weaker the privacy protection. It is reflected in the baseline method, and similarly, the precision of our attack increases slightly as the privacy budget grows. Specifically, as shown in Fig. 2(a), when the $\tau = 0.3$ and the privacy budget increases from 1 to 2.4, the attack precision improves from 97.31% to 97.66%. Additionally, as the threshold τ increases, the attacker outputs more confident inference results, further enhancing the attack precision. For example, in Fig. 2(b), when $\epsilon = 1$, the attack precision is 96.23% for $\tau = 0.3$ and 98% for $\tau = 0.7$, showing a slight increase as τ grows. Overall, our attack achieves precision exceeding 96% across all three datasets, providing strong evidence that the cyclic shifting-based LDP mechanism, as proposed by Huang et al. [16], fails to protect user privacy effectively.

The baseline scheme directly treats the perturbed results as original data, resulting in a null rate of 0. In contrast, our attack determines inference results based on confidence, leading to some users with null inference results. The null rate across the three datasets decreases as the privacy budget increases. This is because, although Huang et al.'s scheme [16] does not satisfy differential privacy, it still introduces noise into the data. As the privacy budget increases, the amount of noise decreases, enhancing the attacker's confidence in the inference results. Specifically, as shown in Fig. 2(d), when $\tau = 0.3$, the null rate of the attack on the online retail dataset decreases from 17.30% to 7.78% as the privacy budget rises from 1 to 2.4. Furthermore, in the two Food datasets, the null rate of our attack does not exceed 3%. This may be attributed to the larger number of items and the smaller user base in the retail dataset, leading to less accurate frequency estimation for individual items.

D. Summary

As shifting does not ensure that user data is perturbed strictly according to the specified probability and significantly

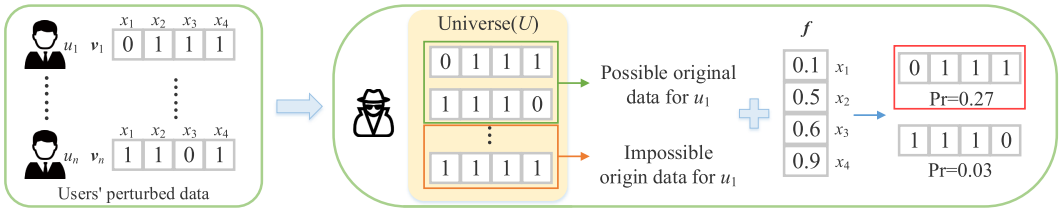
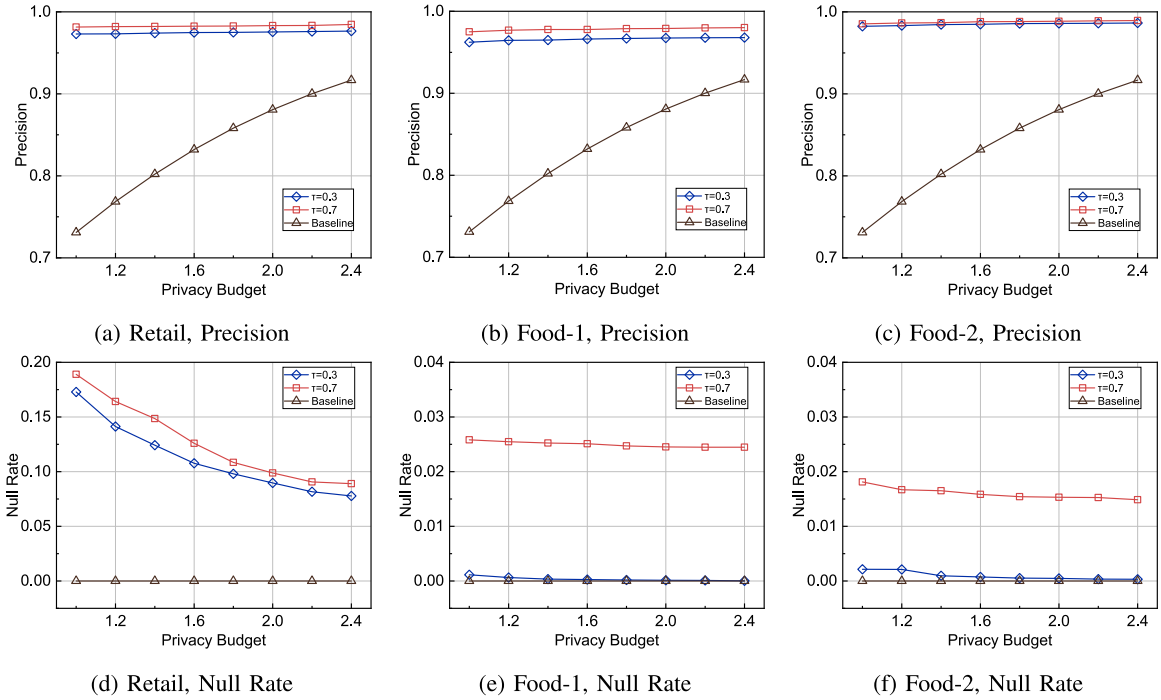


Fig. 1. An attack example on user u_1.



(a) Retail, Precision (b) Food-1, Precision (c) Food-2, Precision

(d) Retail, Null Rate (e) Food-1, Null Rate (f) Food-2, Null Rate

Fig. 2. Demonstration of attack effectiveness. The subfigures in the top and bottom rows show the precision and null rate results, respectively, across a privacy budget range of 1.0 to 2.4.

reduces the output space, it fails to meet the fundamental definition of differential privacy, resulting in the leakage of user privacy. Therefore, shift-based LDP protocols are theoretically and practically ineffective in safeguarding user privacy.

IV. The Proposed Scheme

In this section, we first introduce the basic idea of SVJDA and then describe it in detail.

A. Basic Idea

In joint distribution analysis, the data estimation requirements grow exponentially with dimensionality. For instance, in the joint distribution analysis of d items, the estimation requirements for binary relationships are $O(d^2)$. Such high demand disperses the privacy budget, reducing the accuracy of itemset frequency estimation and degrading the overall analytical precision. To tackle the "curse of dimensionality", it is crucial to limit the number of quantities that must be estimated. SVJDA addresses this by leveraging pre-estimation techniques in SVSM [15], thereby concentrating the privacy budget on critical components such as high-frequency itemsets. This reduces the number of itemsets to be estimated, significantly improving accuracy while lowering complexity. Formally, the objective of SVJDA is to estimate the true joint frequency distribution $\boldsymbol{f}$ from perturbed data $\hat{\boldsymbol{f}}$ by minimizing the mean squared error (MSE): $\min_{\hat{\boldsymbol{f}}} \mathbb{E}[\|\hat{\boldsymbol{f}} - \boldsymbol{f}\|_2^2]$. To achieve this MSE minimization objective, SVJDA proceeds in two steps: (i) the aggregator first calculates frequencies under the assumption of item independence; and (ii) it then recollects user data to refine the frequencies of high-frequency itemsets. This enables a more effective allocation of the privacy budget and enhances overall estimation accuracy.

Stage 1. "Item Frequency Estimation" estimates item frequencies to rank items and obtain an initial joint distribution.

Stage 2. "Frequent Itemset Frequency Estimation" refines the frequency estimates for itemsets within the candidate set.

To optimize the use of the privacy budget, users are randomly divided into two disjoint groups, each corresponding to one stage, with each user making a single query per task. Let n_1 and n_2 represent the number of users in each group, respectively. Since no user participates in both stages,

the overall protocol still satisfies ϵ-LDP without any budget division or composition cost.Such a user-level partitioning strategy has been shown to achieve higher overall utility than repeatedly querying the same users with split privacy budgets, as discussed in previous work [15].

Although SVJDA adopts the architecture of SVSM [15], it focuses on the precise estimation of the joint distribution. Unlike SVSM [15], which primarily identifies the most frequent items or itemsets, SVJDA aims to provide accurate frequency estimates for all itemsets, thus expanding its applicability. Additionally, SVJDA treats user data as sparse vectors and employs the SVME perturbation protocol, which is more suitable for sparse data. Since SVME [22] does not require sampling user data, it can collect more information and has been shown to significantly improve accuracy [22], thus enhancing the reliability of the frequency estimates. However, the SVME protocol [22] requires determining the sparsity parameter L, representing the maximum number of items per user. Under local differential privacy, L is unknown to the aggregator, necessitating its retrieval from users. Additionally, we made adjustments to reduce estimation errors in the joint distribution without compromising security.

B. Detailed Description of SVJDA

We now describe our SVJDA in detail, following its two-stage procedure.

1) Stage 1. Item Frequency Estimation: When the domain value range is large, the aggregator must narrow the focus to a smaller set of candidates. In the first task, SVJDA estimates item frequencies and sorts the items accordingly. To do this, the aggregator interacts with the first group of users and applies the SVME protocol [22]. Stage 1 is divided into 3 steps, with users grouped into three sets, each participating in one step. If the number of users is small, only the first step is executed.

Step 1: **Initial Estimation of Item Frequencies.** In this step, we estimate item frequencies under the assumption that each user contributes at most one item. This assumption does not restrict users to holding only a single item; rather, it constrains their contribution to minimize the sensitivity associated with L, thereby significantly reducing the amount of injected noise. Importantly, obtaining the exact per-user value of L incurs no additional cost. Although using a lower estimate of L may lead to an underestimation of absolute frequencies, frequency rankings remain robust to such bias and can still accurately capture the relative ordering of items even under noise. Furthermore, since the frequency estimates produced in Task 1 serve as the initial values for joint distribution estimation, obtaining reliable item-frequency estimates is an essential prerequisite.

According to [22], the optimal parameters under user-level LDP are set with the number of buckets $b = 1$, the cropping range $\eta = \sqrt{2L\log(4nb/\beta)}$, and the noise parameter $\Delta = 2\eta$, where β represents the probability that the mechanism does not satisfy the error bound. In this step, we set the sparsity parameter $L_{1,1} = 1$ and calculate the corresponding clipping range $\eta_{1,1}$ and noise parameter $\Delta_{1,1}$. Each user then reports their private vector $\boldsymbol{v}$:

$$\Psi_{\mathrm{SVME}(1,\eta_{1,1},\Delta_{1,1},\epsilon)}(\boldsymbol{v}).$$

Subsequently, the aggregator collects all reports and applies the aggregation algorithm to estimate the frequency of item $x \in X$:

$$\Phi_{\mathrm{SVME}(1,\eta_{1,1},\Delta_{1,1},\epsilon)}(v_x),$$

selecting the top $2k$ most frequent items to form the candidate set S. Selecting $2k$ candidates ensures that the true top-k items are included with high probability, as the relative frequency ranking remains robust despite the underestimation bias caused by setting $L = 1$. The candidate set S is then transmitted to the users participating in the next step.

Step 2: **Sparsity Estimation.** The frequency estimates obtained in the first step are generally lower than the true values. To improve the accuracy of frequency estimates, it is necessary to obtain information regarding the sparsity parameter L for each user. To this end, we utilize the candidate set S derived in Step 1 and conduct further frequency estimation on the items within S.

In this step, since each user possesses only one data, we set the sparsity parameter $L_{1,2} = 1$. Simultaneously, based on the user's reports, we calculate the sparsity parameter $L_{1,3}$ for the next step. Specifically, each user first constructs a candidate vector $\boldsymbol{v}_S$ of length $2k$ using the items in the candidate set S, where each element $\boldsymbol{v}_S[i]$ indicates whether the user owns item i. To determine $L_{1,3}$, each user reports the number of non-zero values in their candidate vector $\boldsymbol{v}_S$, i.e., the number of candidate items owned:

$$\Psi_{\mathrm{SVME}(1,\eta_{1,2},\Delta_{1,2},\epsilon)}\left(\sum_{i=1}^{2k}\boldsymbol{v}_S[i]\right),$$

The aggregator aggregates all users' reports and estimates the length distribution:

$$\Phi_{\mathrm{SVME}(1,\eta_{1,2},\Delta_{1,2},\epsilon)}(\ell),$$

where $\ell \in [1, 2, \cdots, 2k]$. Next, the aggregator determines the smallest value of L that satisfies

$$\frac{\sum_{\ell=1}^{L}\Phi_{\mathrm{SVME}(1,\eta_{1,2},\Delta_{1,2},\epsilon)}(\ell)}{\sum_{\ell=1}^{2k}\Phi_{\mathrm{SVME}(1,\eta_{1,2},\Delta_{1,2},\epsilon)}(\ell)} > 0.9.$$

Finally, the aggregator transmits $L_{1,3} = L$ to the users in the final step.

Step 3: **Frequent Item Frequency Estimation.** The final group of users reports their vector $\boldsymbol{v}_S$ using L as a parameter:

$$\Psi_{\mathrm{SVME}(L_{1,3},\eta_{1,3},\Delta_{1,3},\epsilon)}(\boldsymbol{v}_S).$$

The aggregator then estimates the frequency of each frequent item $x \in S$ using the following formula:

$$\Phi_{\mathrm{SVME}(L_{1,3},\eta_{1,3},\Delta_{1,3},\epsilon)}(x).$$

Estimation Update. In Step 3, we set L to the 90th percentile of the input itemset lengths. However, since some users may have an effective sparsity exceeding L, the aggregated results may still exhibit a slight downward bias. Let $\Phi_{\mathrm{SVME}}(\ell)$

denote the estimated length distribution over the candidate set S. Without limiting each user's contribution, the expected total number of item occurrences can be written as

$$total = \sum_{\ell=1}^{2k} \Phi_{\text{SVME}}(\ell)\,\ell.$$

When each user is allowed to contribute at most L items, the expected observed total becomes

$$\begin{aligned}\widetilde{total} &= \sum_{\ell=1}^{2k} \Phi_{\text{SVME}}(\ell)\,\min(\ell, L)\\ &= total - \sum_{\ell=L+1}^{2k} \Phi_{\text{SVME}}(\ell)(\ell - L).\end{aligned}$$

Assuming this effect is approximately uniform across items, we define a correction factor $u(L) = total/\widetilde{total}$, yielding

$$u(L) = \frac{total}{total - \sum_{\ell=L+1}^{2k} \Phi_{\text{SVME}}(\ell)(\ell - L)}. \tag{5}$$

The aggregator then utilizes the frequent item frequencies obtained in step 3 to update the item frequencies from step 1, ultimately yielding the final estimate of item frequencies.

Under the assumption that the items are independent, the aggregator calculates the joint probability between any two items. Specifically, for an item set (x_a, x_b), its estimated frequency $\tilde{f}_{(x_a,x_b)}$ is defined as the product of the estimated frequencies of the two items in the set:

$$\tilde{f}_{(x_a,x_b)} = \Phi(x_a)\Phi(x_b).$$

Here, $\Phi(x_a)$ and $\Phi(x_b)$ denote the output of the aggregation algorithm Φ, corresponding to the estimated frequencies $\hat{f}_{x_a}$ and $\hat{f}_{x_b}$ of items x_a and x_b, respectively.

The top $2k$ itemsets with the highest estimated frequencies are selected to form the candidate set of itemsets IS, constructed as follows:

$$IS = \left\{ x : x \subseteq X, |x| = 2, f_x = \prod \Phi(x) > t \right\}.$$

Here, the threshold t is chosen such that $IS = 2k$. Since our goal is to obtain a high-accuracy joint distribution, the size of the candidate set of itemsets IS can be adjusted based on the number of users. When the number of users is large, a larger IS can be used to estimate and update the frequencies of more itemsets.

2) Stage 2: Frequent Itemset Frequency Estimation: After constructing the candidate set of itemsets IS, the subsequent protocol for estimating itemset frequencies follows a procedure similar to that of Task 1, where parameter determination (i.e., η, Δ, and L) remains consistent with Stage 1. Specifically, each user constructs the candidate vector $\boldsymbol{v}_{IS}$ based on the candidate set of itemsets IS. Each user reports the number of non-zero values in their candidate vector $\boldsymbol{v}_{IS}$:

$$\Psi_{\text{SVME}(1,\eta_{2,1},\Delta_{2,1},\epsilon)}\left(\sum_{i=1}^{2k} \boldsymbol{v}_{IS}[i]\right).$$

The aggregator aggregates all users' reports and finds the 90th percentile $L_{2,2}$, which is sent to the last group of users. The last group of users reports their vector $\boldsymbol{v}_{IS}$ based on parameter $L_{2,2}$:

$$\Psi_{\text{SVME}(L_{2,2},\eta_{2,2},\Delta_{2,2},\epsilon)}(\boldsymbol{v}_{IS}).$$

The aggregator estimates the frequency of each itemset $\boldsymbol{x} \in IS$ as:

$$\Phi_{\text{SVME}(L_{2,2},\eta_{2,2},\Delta_{2,2},\epsilon)}(\boldsymbol{x}) \cdot u'(L),$$

where $u'(L)$ represents the update factor used for correcting bias (same format as Eq. (5)). Subsequently, the aggregator utilizes the aggregated results to update the estimated joint frequencies from Task 1, yielding the final estimate of the joint distribution.

V. Theoretical Analysis

In this section, we first prove that SVJDA satisfies ϵ-LDP, and then analyze its estimation bias and L_∞ error.

A. Privacy Analysis

Theorem 1: The SVJDA protocol satisfies ϵ-LDP.

Proof. In the SVJDA protocol, each user participates in only one step and adopts the SVME protocol to perturb their data. Below, we first prove the privacy guarantee of the SVME protocol.

As stated in SVME [22], with probability $1-O(\beta)$, we have $|B_i| \le \eta$ for all $i \in [n]$. Therefore, given the inputs v and v', the difference between B_i and B_i' is bounded by 2η with probability $1-O(\beta)$. After applying the clipping with threshold η, the difference between B_i and B_i' after clipping remains at most 2η with probability 1. When the noise parameter is set to $\Delta = 2\eta$, we obtain:

$$\begin{aligned}\frac{\Pr[\bar{B}_i + r = x \mid h]}{\Pr[\bar{B}_i' + r = x \mid h]} &= \frac{\exp\left(\frac{\epsilon}{\Delta}|x - \bar{B}_i|\right)}{\exp\left(\frac{\epsilon}{\Delta}|x - \bar{B}_i'|\right)}\\ &= \exp\left(\frac{\epsilon}{2\eta}\left(|x - \bar{B}_i| - |x - \bar{B}_i'|\right)\right)\\ &\le \exp\left(\frac{\epsilon}{2\eta}|\bar{B}_i - \bar{B}_i'|\right) \le \exp(\epsilon).\end{aligned}$$

Thus, the SVME protocol satisfies ϵ-LDP.

Similarly, the SVJDA protocol also satisfies ϵ-LDP. Specifically, since the security of the SVME protocol primarily depends on the parameters η and Δ. By restricting the differences between user data through η and introducing noise related to η via Δ, the protocol ensures ϵ-LDP. Consequently, in SVJDA, setting the sparsity parameter L to 1 or using an estimated value instead of the true value does not compromise the protocol's security. Therefore, every task in our protocol satisfies ϵ-LDP. The aggregation and subsequent estimation performed by the server operate solely on perturbed data, and therefore fall under the post-processing property of differential privacy, which does not incur additional privacy loss. If a user were to participate in multiple tasks or communication rounds, the total privacy loss would be accumulated according to the sequential composition theorem, i.e., the total privacy budget would be the sum of the individual budgets used in each round. Since each sub-procedure in the SVJDA protocol satisfies ϵ-LDP and each user only participates in one task, it follows that the overall SVJDA protocol also satisfies ϵ-LDP.

B. Utility Analysis

Theorem 2: When $\eta \geq \sqrt{2L\log(4n/\beta)}$, SVJDA is unbiased.

Proof. SVJDA utilizes the SVME protocol to estimate the item frequencies and itemset frequencies of users. The SVME protocol has been proven to be unbiased in [22]. By leveraging the unbiased nature of SVME, SVJDA guarantees the unbiasedness of the estimation results.When extending to joint distribution estimation, the unbiasedness result still holds, because each candidate itemset is treated as a single binary attribute and perturbed independently using the SVME mechanism in Stage 2. Therefore, the expected value of the final joint estimator depends only on the SVME perturbation and remains unbiased. The initial estimates obtained in Stage 1 are used solely for candidate pruning and do not affect the unbiasedness of the final estimator.

Theorem 3: With probability at least $1-\beta$, SVJDA achieves the L_∞ error of $O\left(\frac{\epsilon\sqrt{L}+\sqrt{L\log(n/\beta)}}{\epsilon}\sqrt{\frac{\log(d/\beta)}{n}}\right)$.

Proof. The error analysis of SVJDA follows a three-stage decomposition analogous to the SVME framework. For $x \in X$, the aggregator computes the estimated value $\hat{v}[x] = \frac{1}{n}\sum_{i\in[n]} h_i(x)\cdot\tilde{B}_i$. We decompose the total error via:

$$\begin{aligned}|\bar{v}[x]-\hat{v}[x]| \leq & \left|\frac{1}{n}\sum_{i\in[n]} v_i[x]-\frac{1}{n}\sum_{i\in[n]} B_i h_i(x)\right| \\ & + \left|\frac{1}{n}\sum_{i\in[n]} B_i h_i(x)-\frac{1}{n}\sum_{i\in[n]} \bar{B}_i h_i(x)\right| \\ & + \left|\frac{1}{n}\sum_{i\in[n]} \bar{B}_i h_i(x)-\frac{1}{n}\sum_{i\in[n]} \tilde{B}_i h_i(x)\right|.\end{aligned}$$

First, we analyze the binning error $|\frac{1}{n}\sum_{i\in[n]} v_i[x] - \frac{1}{n}\sum_{i\in[n]} B_i h_i(x)|$. According to [22], the binning error is bounded by $O\left(\sqrt{L\cdot\frac{\log(d/\beta)}{n}}\right)$ with a probability of at least $1-\beta/10d$. Next, we analyze the clipping error $\left|\frac{1}{n}\sum_{i\in[n]} B_i h_i(x)-\frac{1}{n}\sum_{i\in[n]} \bar{B}_i h_i(x)\right|$. According to Theorem 2, when $\eta \geq \sqrt{2L\log(4n/\beta)}$, we have $\bar{B}_i = B_i$, so the error in this term is 0. Finally, for the noise term $\left|\frac{1}{n}\sum_{i\in[n]} \bar{B}_i h_i(x)-\frac{1}{n}\sum_{i\in[n]} \tilde{B}_i h_i(x)\right|$, we can similarly use [22] to show that this error is bounded by $O\left(\frac{\Delta}{\epsilon}\sqrt{\frac{\log(d/\beta)}{n}}\right)$ with a probability of at least $1-\beta/10d$. Since Theorem 2 establishes that each itemset is treated as a single binary attribute in Stage 2, the above error decomposition applies directly to joint estimation as well.

Therefore, with probability at least $1-\beta$, the L_∞ error of SVJDA protocol satisfies the bound: $\max_{x\in X}|\bar{v}[x]-\hat{v}[x]| \leq O\left(\frac{\epsilon\sqrt{L}+\sqrt{L\log(n/\beta)}}{\epsilon}\sqrt{\frac{\log(d/\beta)}{n}}\right)$.

C. Communication and Computational Overhead Analysis

The proposed SVJDA protocol leverages a two-stage framework to achieve privacy-preserving joint distribution analysis. Both stages rely on the SVME mechanism, which significantly minimizes overhead compared to traditional LDP protocols. We analyze the computational and communication costs below.

1) Computational Overhead: The total computational cost is the sum of the overhead at the user side and the aggregator side.

- **User-Side:** Users execute Ψ_{SVME} once per stage. Since clipping and noise addition are $O(1)$, the complexity is dominated by hashing and summing the L-sparse input vector v_i, resulting in $O(L)$. The total cost $O(L_{\max})$ is therefore negligible relative to the domain size d.
- **Aggregator-Side:** The aggregator sums n reports with a complexity of $O(n\cdot d)$, where d varies by stage. Combined with the $O(d^2)$ cost for calculating the initial joint distribution and generating the candidate set in Stage 1, the total computational overhead is dominated by $O(n\cdot d+d^2)$. This polynomial complexity ensures the protocol remains efficient and scalable for practical applications.

2) Communication Overhead: The communication overhead is determined by the bits transmitted per user. Leveraging the SVME mechanism, each user transmits a description of the hash function (e.g., a PRF seed of size λ) and the perturbed data bounded by $O(\log L)$ bits. Consequently, the total per-user overhead is $O(\log L+\lambda)$. This cost is significantly more efficient than the linear $O(d)$ complexity of traditional protocols, making it highly suitable for large-domain set-valued data.

VI. Experiments

This section conducts an experimental evaluation of SVJDA and compares its performance with relevant methods. It begins with a description of the experimental setup, followed by an in-depth analysis of the results. The source codes of the experiments are available on Github.[2]

A. Experimental Setup

1) Comparison Schemes: To the best of our knowledge, there is no research on the joint distribution analysis of set-valued data with local differential privacy. We compare our SVJDA with two state-of-the-art related works, SVSM [15] and LDP-RM [28], both of which are based on the SVIM protocol [15]. SVSM [15] is designed for frequent itemsets mining, while LDP-RM [28] is for relationship mining. To facilitate the comparison, we make certain adjustments to the schemes. SVIM [15] consists of three steps: the first step calculates the initial frequency estimation for each item, and the third step provides a precise frequency estimation for frequent items. In our experiments, we refine the estimates of the first step using the results from the third step to obtain more accurate item frequency estimates. Based on this, we calculate the joint probability of any two items by constructing an initial joint distribution matrix under the assumption of item independence. For SVSM [15], we then update the matrix using its frequent itemset estimation results, thereby obtaining the final joint distribution analysis. For LDP-RM [28], we modify the high-confidence relationship mining in Task 3 to high-support relationship mining and use the frequency estimation results to update the initial joint distribution estimation matrix.

[2]https://github.com/vvYl/SVJDA

TABLE II
PARAMETER SETTING IN SVJDA

Symbol	Recommended setting
Bins b	1
Sparsity L	estimated per stage
Clipping range η	$\sqrt{2L\log(4nb/\beta)}$
Laplacian Noise Magnitude Δ	2η

TABLE III
STATISTICS OF DATASETS

Datasets	**#Clients**	**#Items**	**#Records**	**Sparsity** L
IFTTT [33]	300,000	354	300,000	45
Food-2 [31]	19,658	337	119,183	24

2) Datasets: Two datasets are used for experimental evaluation, namely the IFTTT dataset [33] and the Takeaway Food Orders-2 dataset [31]. The IFTTT dataset contains 354 unique items extracted from the top 200 applets in IFTTT 2023 by 300,000 users, while the Takeaway Food Orders-2 dataset contains 337 items and 19,658 users. Table III gives more information about the datasets.

3) Evaluation Metrics: The following four metrics are adopted to evaluate the accuracy of the estimation. Let $\bar{v} = \frac{1}{n}\sum_{i\in[n]} v_i$ represent the true mean vector, and $\hat{v}$ denote the estimated vector. Define $\boldsymbol{x}_i$ as the i-th most frequent itemset, and let $S_t = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$ represent the ground truth for the top-k itemsets. Additionally, let S_e denote the top-k itemsets identified by the protocol, and $S_t \cap S_e$ represent the set of true itemsets identified correctly.

- L_∞ **Error:** Measures the maximum absolute deviation between the estimated and true values:

$$L_\infty \text{error} = \max_{i\in[d]} |\hat{v}[i] - \bar{v}[i]|.$$

- **Mean Square Error (MSE):** Measures the overall average deviation between the estimated and true values:

$$\text{MSE} = \frac{1}{d}\sum_{i\in[d]} (\hat{v}[i] - \bar{v}[i])^2 .$$

- **Normalized Cumulative Rank (NCR):** Evaluates the priority of high-ranking itemsets identified by the protocol:

$$\text{NCR} = \frac{\sum_{\boldsymbol{x}\in S_e\cap S_t} q(\boldsymbol{x})}{\sum_{\boldsymbol{x}\in S_t} q(\boldsymbol{x})},$$

  here $q(\boldsymbol{x})$ is the quality function used to evaluate the importance of an itemset $\boldsymbol{x}$, and $q(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \notin S_t$. For the k correct itemsets in S_t, the quality function $q(\boldsymbol{x}_i)$ is defined as $q(\boldsymbol{x}_i) = k - i + 1$, where i is the rank of the itemset.
- **Variance (VAR):** Measures the estimation accuracy of the frequencies of the identified itemsets:

$$\text{VAR} = \frac{1}{|S_t|}\sum_{\boldsymbol{x}\in S_t} \left(\hat{f}_{\boldsymbol{x}} - f_{\boldsymbol{x}}\right)^2 ,$$

  where $\hat{f}_{\boldsymbol{x}} = 0$ if $\boldsymbol{x} \notin S_e$.

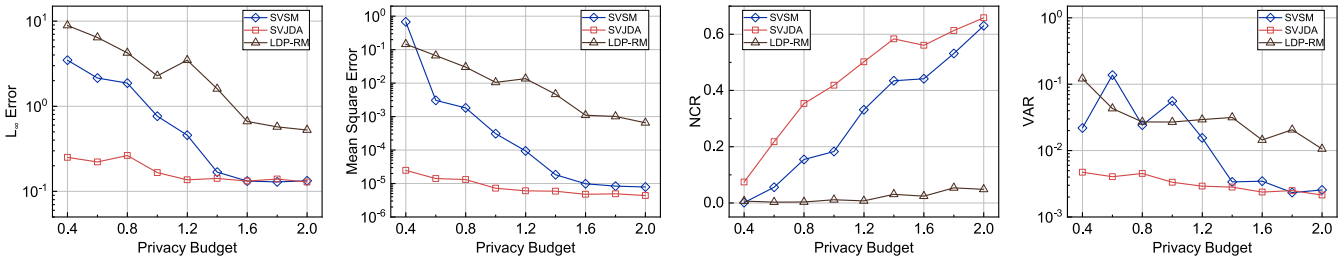
4) Environment: All experiments are developed using Python 3.9 and executed on a computer with an Intel Core i7-12700F 2.10 GHz processor, 16GB memory, and the Windows 10 operating system. The results of each experiment are taken by averaging the outcomes from 10 runs.

B. Results

1) Impact of Privacy Budget: Fig. 3 and Fig. 4 show the performance comparison of joint distribution analysis with $k = 64$ and varying ϵ on the Takeaway Food Order-2 and the IFTTT dataset, respectively. The results indicate that SVJDA has a significant advantage in joint distribution estimation accuracy and frequent itemset identification accuracy compared to SVSM [15] and LDP-RM [28]. In general, the joint distribution estimation accuracy improves as ϵ increases, as evidenced by the decrease in both L_∞ error and MSE with larger ϵ. Similarly, the frequent itemsets identification accuracy also improves with increasing ϵ, as reflected in the increasing NCR score and the decreasing estimation error VAR of the frequent itemsets.

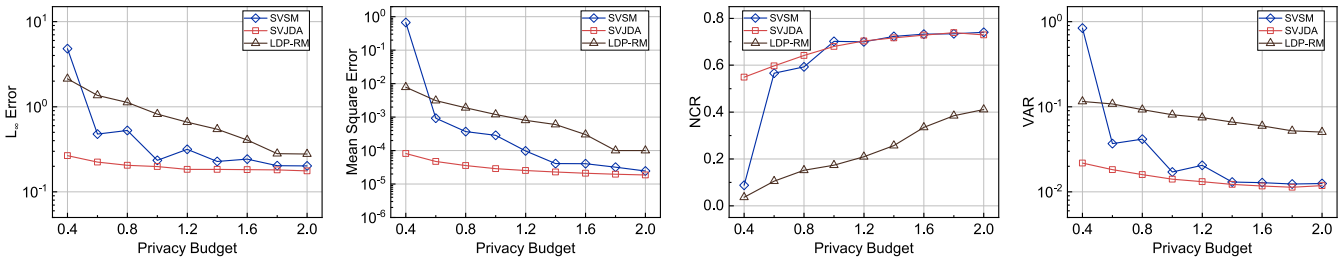
Figs. 3(a) and 3(b) show the comparison of the L_∞ error and MSE for joint distribution estimation among the three schemes on the Food-2 dataset, with the privacy budget ϵ ranging from 0.4 to 2.0. As ϵ increases, both L_∞ error and MSE decrease across all three schemes. However, SVJDA consistently outperforms SVSM [15] and LDP-RM [28], exhibiting notably lower error values, especially under low privacy budget conditions. Specifically, when ϵ is in the range of $0.4 - 1.0$, the L_∞ error of SVJDA is $7.208\% - 21.725\%$ of that of SVSM [15] and $2.821\% - 7.279\%$ of that of LDP-RM [28], while its MSE is just $0.00364\% - 2.338\%$ of SVSM [15] and $0.017\% - 0.068\%$ of LDP-RM [28]. For instance, when $\epsilon = 0.6$, the L_∞ error of SVJDA is 0.22123, which is only 10.299% of SVSM (2.1481) and 3.449% of LDP-RM (6.41387). Additionally, SVJDA's MSE is 0.0000141, which is 0.462% of SVSM (0.00305) and 0.06639% of LDP-RM (0.021). This is mainly because SVJDA adopted the SVME protocol [22]. By leveraging a symbolic hash function, SVME [22] compresses the variance, enabling the incorporation of lower noise. According to the theoretical bound in Theorem 3, the L_∞ error can be decomposed into three components: the binning error, the clipping error, and the noise introduced by the Laplace mechanism. When the privacy budget ϵ is small, the overall error is dominated by the noise term. Since user data are mapped into a single bucket and clipped, the required noise magnitude can be relatively small. As ϵ increases, the impact of the noise term gradually weakens, and the total error becomes primarily constrained by the binning error. This observation is consistent with the experimental results in Figs. 3(a) and 3(b), where both the L_∞ error and MSE decrease steadily with larger ϵ and eventually approach the theoretical lower bound determined by the inherent binning variance.

Figs. 3(c) and 3(d) illustrate the accuracy of frequent itemsets identification among the three schemes on the Food-2 dataset from the perspectives of NCR score and VAR. As shown in Fig. 3(c), when ϵ increases from 0.4 to 2.0, the



(a) L_∞ Error (b) MSE (c) NCR (d) VAR

Fig. 3. Performance comparison of joint distribution analysis with $k = 64$ and varying ϵ on the Food-2 dataset.



(a) L_∞ Error (b) MSE (c) NCR (d) VAR

Fig. 4. Performance comparison of joint distribution analysis with $k = 64$ and varying ϵ on the IFTTT dataset.

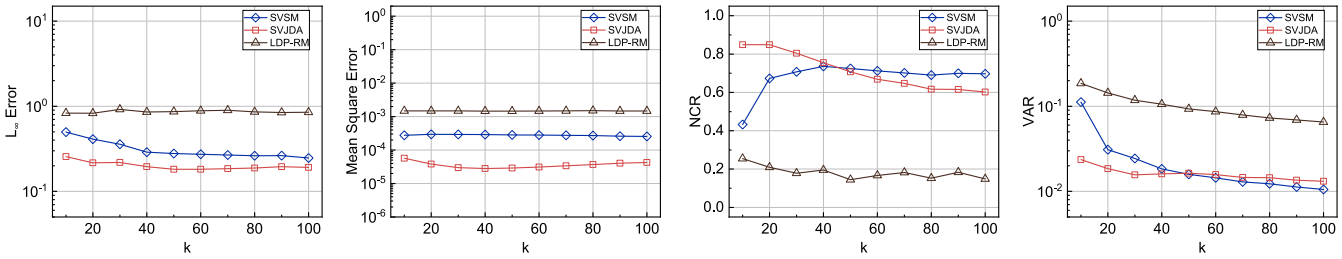
NCR scores of SVJDA consistently exceed those of SVSM [15] and significantly outperform LDP-RM [28]. Specifically, the NCR score of SVJDA is approximately 1.04–57.06 times that of SVSM [15] and 11.4–108.14 times that of LDP-RM [28]. For instance, when $\epsilon = 0.4$, the NCR scores of SVSM [15] and LDP-RM [28] are nearly 0, indicating that they fail to identify frequent itemsets correctly. In contrast, SVJDA achieves an NCR score of 0.074, enabling the identification of approximately 5 frequent itemsets when $k = 64$. When $\epsilon = 1$, SVJDA's NCR score increases to 0.419, whereas SVSM [15] and LDP-RM [28] reach only 0.183 and 0.011, respectively. This means that at $k = 64$, SVJDA can identify 27 frequent itemsets, while SVSM [15] and LDP-RM can identify only 12 and 1 frequent itemsets, respectively. Regarding VAR, SVJDA consistently exhibits lower VAR values than both SVSM [15] and LDP-RM [28]. The VAR of SVSM [15] fluctuates significantly, primarily due to the noise and inaccuracies in padding length estimation, leading to unstable results. LDP-RM [28], leveraging singular value decomposition and low-rank approximation, reduces dimensionality and achieves more accurate item frequency estimation over larger domains. However, its performance deteriorates due to its higher sensitivity to the number of users and privacy budget requirements.

The overall trend in Fig. 4 is similar to that observed in Fig. 3. As shown in Fig. 4, LDP-RM [28] exhibits a slight improvement in frequent itemsets identification accuracy on the larger-scale dataset IFTTT due to the increased number of users (Fig. 4(c)). However, its joint distribution estimation accuracy and frequent itemsets identification accuracy remain inferior to SVJDA and SVSM [15]. SVJDA continues to demonstrate significant advantages across all evaluation metrics, particularly in L_∞ error and MSE. At $\epsilon = 0.4$, SVJDA achieves an NCR score of 0.55, providing a notable advantage in frequent itemsets identification. When $\epsilon > 1$, SVSM [15] approaches SVJDA in terms of NCR, but its overall performance remains slightly inferior to SVJDA.

2) Impact of the Number of Frequent Itemsets: Since the trends observed in the IFTTT and Food-2 datasets are similar, we only show the performance of the three schemes with $\epsilon = 1$ and varying k on the IFTTT dataset, as shown in Fig. 5. The results indicate that SVJDA significantly outperforms both SVSM [15] and LDP-RM [28] in the two most critical metrics, L_∞ error and MSE, which reflect the accuracy of joint distribution analysis. Meanwhile, in terms of NCR and VAR, which measure the accuracy of frequent itemsets frequency estimation, SVJDA performs comparably to SVSM [15] but remains significantly superior to LDP-RM [28].

As shown in Figs. 5(a) and 5(b), when $\epsilon = 1$, the mean L_∞ error of SVJDA is approximately 65.79% and 23.32% of that of SVSM [15] and LDP-RM [28], respectively, while its mean MSE is only 13.3% and 2.48% of that of SVSM [15] and LDP-RM [28], respectively. For example, when $k = 20$, SVJDA has an L_∞ error of 0.2168, which is only 52.87% of SVSM (0.41007) and 26.2% of LDP-RM (0.82758). Furthermore, as k increases, the L_∞ error gradually decreases. This is because a larger k improves the overall estimation accuracy but expands the estimation domain, reducing precision for individual items. The MSE follows a similar trend, reinforcing that SVJDA achieves the highest accuracy in joint distribution estimation.

From Figs. 5(c) and 5(d), it is evident that SVJDA's NCR score and VAR significantly outperform those of LDP-RM [28], with NCR score being approximately 3.96 times that of LDP-RM and VAR being only 16.79% of LDP-RM [28]. In comparison to SVSM [15], SVJDA's NCR score and VAR are close to those of SVSM [15]. When $k > 50$, SVJDA's



(a) L_∞ Error (b) MSE (c) NCR (d) VAR

Fig. 5. Performance comparison of joint distribution analysis with $\epsilon = 1$ and varying k on the IFTTT dataset.

NCR score and VAR are slightly inferior to those of SVSM. However, when $k < 50$, SVJDA achieves higher NCR scores and lower VAR values than SVSM [15], demonstrating a clear advantage in scenarios with smaller k.

3) Impact of Dataset Characteristics: Table III summarizes the key characteristics of the datasets used in our experiments. Although both datasets have comparable item domain sizes, they differ significantly in user scale and sparsity level. The average sparsity L of IFTTT is 45, nearly twice that of Food-2 ($L = 24$), indicating that user data in IFTTT is denser and contains more item co-occurrences. This difference in sparsity influences the performance of joint distribution estimation.

Specifically, as shown in Fig. 4, SVJDA exhibits smoother performance on the denser IFTTT dataset because the increased number of items per user improves the statistical stability of the SVME-based perturbation and aggregation processes. On the sparser Food-2 dataset, the performance gap between SVJDA and SVSM becomes more pronounced under low privacy budgets (e.g., $\epsilon \leq 1.0$), since SVJDA effectively leverages the sparsity of the data to achieve accurate joint estimation. Furthermore, LDP-RM suffers from reduced accuracy on both datasets due to its reliance on low-rank approximation, which is sensitive to sparsity and user scale. These results indicate that the performance of SVJDA is consistent under different dataset characteristics, suggesting that it can effectively handle relatively sparse set-valued data.

VII. Conclusion

This paper proposes a joint shift inference attack targeting the cyclic shift-based local differential privacy scheme introduced by Huang et al. [16]. Theoretical analysis and experimental results demonstrate that this attack can infer the user's true data with approximately 96% probability, and its accuracy is almost independent of the privacy budget, indicating that cyclic shift-based differential privacy fails to effectively protect user privacy. Next, we introduce SVJDA, a privacy-preserving joint distribution analysis protocol for set-valued data via local differential privacy. Theoretical analysis proves that SVJDA strictly satisfies LDP, achieving higher accuracy in joint distribution analysis while maintaining a high identification rate for frequent itemsets. Experimental comparisons on real-world datasets further demonstrate that SVJDA outperforms state-of-the-art methods in both joint distribution estimation and frequent itemsets identification. Despite its advantages, SVJDA currently focuses on estimating the joint distribution between two items. In future work, we plan to extend the framework to handle multi-item joint distributions and larger item domains, further improving its applicability and scalability in real-world scenarios.

References

[1] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[2] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 425–438.

[3] H. Kang, Y. Ji, and S. Zhang, "Enhanced privacy preserving for social networks relational data based on personalized differential privacy," *Chin. J. Electron.*, vol. 31, no. 4, pp. 741–751, Jul. 2022.

[4] X. Zhu, V. Y. F. Tan, and X. Xiao, "Blink: Link local differential privacy in graph neural networks via Bayesian estimation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2023, pp. 2651–2664.

[5] S. Xu, S. Su, L. Xiong, X. Cheng, and K. Xiao, "Differentially private frequent subgraph mining," in *Proc. IEEE 32nd Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 229–240.

[6] X. Li, H. Yan, Z. Cheng, W. Sun, and H. Li, "Protecting regression models with personalized local differential privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 960–974, Mar. 2023.

[7] X. Liu, W. Gan, L. Yu, and Y. Liu, "DP-PartFIM: Frequent itemset mining using differential privacy and partition," *IEEE Trans. Emerg. Topics Comput.*, vol. 13, no. 3, pp. 567–577, Jul. 2025.

[8] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. - CCS*, 2013, pp. 901–914.

[9] H. Wang, H. Hong, L. Xiong, Z. Qin, and Y. Hong, "L-SRR: Local differential privacy for location-based services with staircase randomized response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2022, pp. 2809–2823.

[10] B. Jiang, J. Li, H. Wang, and H. Song, "Privacy-preserving federated learning for industrial edge computing via hybrid differential privacy and adaptive compression," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1136–1144, Feb. 2023.

[11] X. Xu, Z. Fan, M. Trovati, and F. Palmieri, "MLPKV: A local differential multi-layer private key-value data collection scheme for edge computing environments," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1825–1838, 2023.

[12] Y. Zhou, Z. Wang, and Y. Liu, "A local differential privacy hybrid data clustering iterative algorithm for edge computing," *Chin. J. Electron.*, vol. 33, no. 6, pp. 1421–1434, Nov. 2024.

[13] Z. Xu et al., "Learning to generate image embeddings with user-level differential privacy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7969–7980.

[14] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 192–203.

[15] T. Wang, N. Li, and S. Jha, "Locally differentially private frequent itemset mining," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 127–143.

[16] Y. Huang, K. Xue, B. Zhu, D. S. L. Wei, Q. Sun, and J. Lu, "Joint distribution analysis for set-valued data with local differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 7106–7117, 2024.

[17] X. Ma, H. Liu, and S. Guan, "Improving the effect of frequent itemset mining with Hadamard response under local differential privacy," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021, pp. 436–443.

[18] L. Wang, Q. Ye, H. Hu, and X. Meng, "EPS^2: Privacy preserving set-valued data analysis in the shuffle model," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 6084–6098, Nov. 2024.

[19] Y. Zhu, Y. Cao, Q. Xue, Q. Wu, and Y. Zhang, "Heavy hitter identification over large-domain set-valued data with local differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 414–426, 2024.

[20] S. Wang, L. Huang, Y. Nie, P. Wang, H. Xu, and W. Yang, "PrivSet: Set-valued data analyses with locale differential privacy," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 1088–1096.

[21] S. Wang et al., "Locally private set-valued data analyses: Distribution and heavy hitters estimation," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8050–8065, Aug. 2024.

[22] M. Zhou, T. Wang, T-H. H. Chan, G. Fanti, and E. Shi, "Locally differentially private sparse vector aggregation," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 422–439.

[23] Q. Ye, H. Hu, X. Meng, and H. Zheng, "PrivKV: Key-value data collection with local differential privacy," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 317–331.

[24] X. Gu, M. Li, Y. Cheng, L. Xiong, and Y. Cao, "PCKV: Locally differentially private correlated key-value data collection with optimized utility," in *Proc. 29th USENIX Secur. Symp.*, 2019, pp. 967–984.

[25] Q. Ye et al., "PrivKVM: Revisiting key-value statistics estimation with local differential privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 17–35, Jan. 2023.

[26] 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.

[27] M. Maruseac and G. Ghinita, "Precision-enhanced differentially-private mining of high-confidence association rules," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 6, pp. 1297–1309, Nov. 2020.

[28] K. Dong et al., "Relation mining under local differential privacy," in *Proc. 33rd USENIX Secur. Symp.*, 2024, pp. 955–972.

[29] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th Annu. Symp. Found. Comput. Sci.*, Jordan, Oct. 2013, pp. 429–438.

[30] (2024). *Online Retail II Data Set*. Accessed: Oct. 2024. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Online

[31] (2024). *Takeaway Food Orders*. Accessed: Oct. 2024. [Online]. Available: https://www.kaggle.com/datasets/henslersoftware/19560-indian-takeaway-orders

[32] A. Gadotti, F. Houssiau, M. S. M. S. Annamalai, and Y.-A. de Montjoye, "Pool inference attacks on local differential privacy: Quantifying the privacy guarantees of Apple's count mean sketch in practice," in *Proc. 31st USENIX Secur. Symp. (USENIX Security)*, 2022, pp. 501–518.

[33] (2023). *IFTTT. Top Applets of 2023*. Accessed: Oct. 2024. [Online]. Available: https://ifttt.com/explore/top-applets-2023