Privacy-Preserving Federated Heavy Hitter Analytics for Non-IID Data

Jiaqi Shao [1] **Shanshan Han** [2] **Chaoyang He** [3] **Bing Luo** [1]

Abstract

Federated heavy hitter analytics involves the identification of the most frequent items within distributed data. Existing methods for this task often encounter challenges such as compromising privacy or sacrificing utility. To address these issues, we introduce a novel privacy-preserving algorithm that exploits the hierarchical structure to discover *local and global heavy hitters in non-IID data* by utilizing perturbation and similarity techniques. We conduct extensive evaluations on both synthetic and real datasets to validate the effectiveness of our approach. We also present FedCampus, a demonstration application to showcase the capabilities of our algorithm in analyzing population statistics.

1. Introduction

Identifying heavy hitters (frequently occurring items) is crucial in data mining. However, this task becomes challenging with distributed and sensitive data due to privacy and scalability concerns (Jia & Gong, 2018). For example, analyzing user behaviors across multiple devices (smartphones, smartwatches, IoT) requires finding frequent items while maintaining privacy and efficiency.

The advent of Federated Analytics (FA) has facilitated the examination of data from disparate entities without the requirement of data centralization (Ramage, 2020; Elkordy et al., 2023). It follows federated learning (FL) (Li et al., 2020), where a central server interacts with clients and aggregates their responses to gain global insights. Some algorithms rely on a trusted server and use central differential privacy (CDP) (Dwork, 2008) to protect data privacy, such as TrieHH (Zhu et al., 2020) and TrieHH++ (Cormode & Bharadwaj, 2022). These algorithms have shown promising results in finding heavy hitters while achieving a good trade-off between accuracy and efficiency. Nonetheless, the necessity of a trusted server might not be feasible in several scenarios. Other algorithms employ LDP (Kasiviswanathan et al., 2011) to protect individual privacy without a trusted server by perturbing individual's local data before sending, such as PEM (Wang et al., 2019), RAPPOR (Erlingsson et al., 2014), PrivTrie (Wang et al., 2018), TreeHist and Bitstogram (Bassily et al., 2017). These approaches normally need to construct a tree via the data's prefixes (also known as "trie"), which, however, may suffer from domain limitation. In other words, the next level of construction completely depends on the construction of previous levels, and thus, this can affect the accuracy and efficiency of identifying heavy hitters.

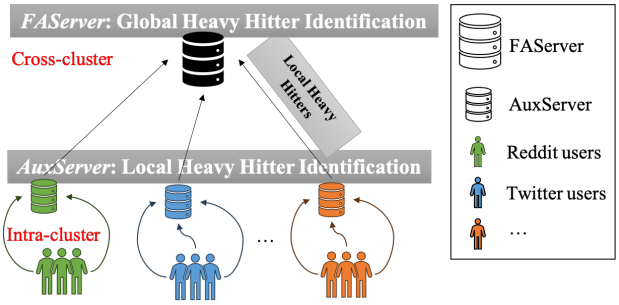

Figure 1. A hierarchical design for identifying heavy hitters across non-IID clusters. The *intra-cluster* level finds *local* heavy hitters within each cluster by *AuxServer*, and the *cross-cluster* level finds *global* heavy hitters across clusters by the *FAServer*.

Additionally, non-Independent and Identically Distributed (non-IID) data present unique obstacles to distributed heavy hitter identification in practical applications. For instance, variations in user tweets or Reddit comments regarding vocabulary and word frequency, arising from factors such as topics, communities, and personal preferences, form *clusters* that are more homogenous internally than when compared with different clusters, as illustrated in Figure 1. Consequently, these data form **clusters** characterized by higher similarity within a cluster compared to different clusters. However, existing heavy hitters identification algorithms (Bodon, 2005; Bhaskar et al., 2010; Dwork et al., 2006a;b; 2010; Bonomi & Xiong, 2013; Bassily et al., 2017;

[1]Department of Data Science Research Center, Duke Kunshan University, Jiangsu, China; [2]Department of Computer Science, University of California, Irvine, Irvine, USA; [3]FedML Inc., Sunnyvale, USA;. Correspondence to: Bing Luo <bing.luo@dukekunshan.edu.cn>.

Workshop of Federated Learning and Analytics in Practice, colocated with 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA.

Cormode et al., 2018; Acharya et al., 2020; Erlingsson et al., 2014) do not consider such non-IID scenarios, potentially reducing the effectiveness of the algorithm. For instance, when data is clustered by topic, current algorithms tend to focus on identifying the most common words across all topics, rather than the most distinctive heavy hitters for each individual topic. This, coupled with the domain limitation issue in LDP, further decrease informativeness in non-IID settings.

In response to the above concerns, this paper introduces a *hierarchical* FA design to recognize heavy hitters within non-IID clusters, as depicted in Figure 1. We summarize the key contributions as follows: (1) We develop an intra-cluster identification algorithm with a novel LDP-based intra-cluster algorithm to *avoid domain limitation* when identifying *local heavy hitters*; (2) We propose a cross-cluster identification algorithm to filter out noisy local heavy hitters from non-IID data clusters; (3) We evaluate our algorithm on synthetic and real datasets and demonstrate its superior performance in non-IID settings. Moreover, we also deploy the algorithm in our demo application *FedCampus* for a campus-scale population statistics analysis.

2. System Design

This section describes our design of a hierarchical FA algorithm to identify top-k heavy hitters from non-IID data clusters of clients, which consists of two phases: intra-cluster (IC) identification (§2.1) and cross-cluster (CC) identification (§2.2), as shown in Figure 1.

2.1. Intra-Cluster (IC) Identification

In the IC identification, we use LDP-based perturbation to discover local heavy hitters within each cluster while preserving privacy. An auxiliary server (*i.e.*, AuxServer in Figure 1) interacts with clients in each cluster to construct a trie based on their perturbed data. The trie efficiently stores and retrieves heavy hitters, facilitating their identification. However, transmitting individual data to the AuxServer poses privacy risks, and the identification process may have domain limitations by excluding data beyond the predefined domain during trie construction.

GRRX. To tackle the aforementioned challenges, we propose a novel algorithm called Intra-Cluster (IC) identification algorithm (Algorithm 1). Our approach leverages the GRRX mechanism to perturb clients' data, ensuring individual data privacy while mitigating the domain limitation issue. GRRX extends the Generalized Random Response (GRR) technique (Wang et al., 2017) that provides ε-LDP (ε being the privacy parameter). However, GRR suffers from a domain limitation problem, potentially omitting data items falling outside the predefined domain Φ.

Algorithm 1 Intra-cluster Identification (AuxServer)

Input : n_κ (No. of clients in cluster κ); b (Required bits); g (Trie's maximum depth); L (Maximum bit-length); Φ (Prefix domain); ε (Privacy parameter)
Output : A_g (Heavy hitters of cluster κ)

1 Partition clients into g disjoint groups $G_1, \dots G_g$
2 Initializes $A_0 = \emptyset$
3 **for** $i = 1, 2, \dots g$ **do**
4 $\Phi_i \leftarrow A_i$ Broadcast Φ_i to G_i, require b_i bits prefixes
5 $A_i \leftarrow$ AGGREGATION(G_i) //Aggregate prefixes
6 **return** A_g //Intra-cluster heavy hitters
7 **Client Side:** $y \leftarrow \text{GRRX}(prefix_b(v), \Phi)$ //Random response
8 **FUNCTION** $\text{GRR}_\text{X}(prefix, \Phi)$
9 **if** $prefix \in \Phi$: $\Phi^\star \leftarrow \Phi + \{x\}$ //x is randomly generated
10 **else** $\Phi^\star \leftarrow \Phi + \{prefix\}$
11 **return** *prefix with p, or* $y \in \Phi^\star \setminus$ *prefix with q*

To overcome domain limitation, we introduce GRRX (corresponding to Line 8-11 in Algorithm 1) which enables each client to add an arbitrary item X to Φ, *masking* out-of-domain data and expanding the domain to $\Phi^\star$. Specifically, it perturbs a data item v to another item y, where the perturbation probability of GRRX depends on the size of the extended domain $\Phi^\star$, the privacy parameter ε, and is defined as $\Pr_{\Phi^\star}[v = y] = \frac{e^\varepsilon}{e^\varepsilon + d}$ for v and y in $\Phi^\star$, and $\Pr_{\Phi^\star}[v \neq y] = \frac{1}{e^\varepsilon + d}$ for v and y not in $\Phi^\star$, where d represents the size of the original domain Φ. To determine the random item X added to the domain, each client ℓ utilizes its own data v_ℓ and its prefix. If the prefix is not in Φ, X is set to v_ℓ. Otherwise, X is randomly selected from a binary prefix range $[0, 2^b]$, where b denotes the prefix length.

By incorporating GRRX into our algorithm, we can effectively handle any data item without compromising privacy or accuracy while effectively addressing the domain limitation inherent in the GRR mechanism.

Incremental Group-size Strategy. To enhance the effectiveness of the IC algorithm, we introduce an incremental group-size strategy. Traditional approaches such as (Zhu et al., 2020; Wang et al., 2019) employ a uniform group-size strategy, where an equal number of clients are assigned to each group. However, this uniform approach often leads to a loss in the identification accuracy of heavy hitters.

Our incremental group-size strategy is motivated by the insight that *later groups can benefit from the information obtained by earlier groups*, allowing for improved refinement of prefixes and more precise identification of heavy hitters. Consequently, we allocate a larger number of clients to the later groups using a linearly incremental group-size strategy. In this strategy, the number of clients in each group G_i is calculated as $\frac{n}{2g} + (i-1)\frac{n}{g(g-1)}$, where n denotes the total number of clients for trie construction, and g represents the total number of groups. By implementing this incremental group-size strategy, we enhance both the efficiency and accuracy of heavy hitter identification within each cluster.

2.2. Cross-Cluster (CC) Identification

In the previous IC identification subsection, we adopted LDP-based perturbation to protect the privacy of local heavy hitters within each cluster. Although the perturbation mechanism achieves LDP via adding noise to the local data, it brings redundancy issue among the local heavy hitters from different clusters. Moreover, the data across clusters are non-IID, which makes it more challenging to find global heavy hitters that are consistent and representative across clusters. Therefore, we propose a Cross-Cluster (CC) identification algorithm (Algorithm 2) that aggregates the local heavy hitters from different clusters to identify global heavy hitters. The CC Identification algorithm consists of two main steps: *importance calculation* and *similarity filtering*.

Importance Calculation The process of *importance calculation* (refer to Line 1 to 4 in Algorithm 2) involves assigning a score to each local heavy hitter based on its relative frequency within its cluster. This score serves as an indicator of the *representativeness* exhibited by the local heavy hitter for its respective cluster, while simultaneously preserving those with higher relative frequencies in their clusters.

Similarity Filtering The process of *similarity filtering* aims to eliminate redundant or noisy local heavy hitters by evaluating their hamming distance (Norouzi et al., 2012) against a predetermined threshold, denoted as δ. The selection of this threshold, along with its underlying rationale, is elaborated upon in the proof provided in Appendix A.1. By leveraging the hamming distance, we can quantify the *distinctiveness* exhibited by each local heavy hitter relative to the other local heavy hitters. This design corresponds to the algorithm outlined in Line 5 to 16 in Algorithm 2.

By combining these two techniques, our algorithm can find global heavy hitters across non-IID data by selecting and filtering consistent and distinctive local heavy hitters from different clusters. These informative global heavy hitters reflect the diversity and similarity of the data across clusters, providing insights into unique and common data patterns.

3. Experiments and Implementation

This section first describes the experiment setting in § 3.1 and then presents the results of our algorithm evaluation in § 3.2. Finally, we illustrate our algorithm deployment in our demo application, called FedCampus in § 3.3.

3.1. Experiment Setting

Datasets. In order to emulate real-world situations characterized by diverse linguistic communities, we employ a simulation approach to generate synthetic non-IID data. These clusters, along with their non-IID properties, such as the number of clients and unique words per cluster, are

Algorithm 2 Cross-cluster Identification (FAServer)

Input : $HH^{(\kappa)}_{local}$: local heavy hitters from clusters.
Output : HH_{top}: identified heavy hitters among clusters.
1 **Initialization:** $HH_{local} \leftarrow \emptyset$; $HH_{top} \leftarrow \emptyset$ **for** $\kappa = 1, 2, \ldots$ **do**
2 **for** c *in* $HH^{(\kappa)}_{local}$ **do**
3 $HH_{local}[c] += \alpha$ // Update importance for c
4 $HH_{local} \leftarrow$ SORT(HH_{local}) // Descending order by importance
5 **return** $HH_{top} \leftarrow$ SELECT(HH_{local}, k, δ)
6 **FUNCTION** SELECT(HH_{local}, k, δ)
7 $HH_{top} \leftarrow \emptyset$
8 **for** $i = 1, 2, \ldots, len(HH_{local})$ **do**
9 $c \leftarrow HH_{local}[i]$
10 **if** c is a noisy result: **continue**
11 **for** $j = i+1, i+2, \ldots, len(HH_{local})$ **do**
12 $c' \leftarrow HH_{local}[j]$
13 **if** HAMMDISTANCE$(c, c') < \delta$: Mark c' as a *noisy* result
14 Add c to HH_{top}
15 **if** HH_{top} has k items: **break**
16 **return** HH_{top}

Table 1. Non-IID Data: Total clients and unique words per dataset.

	NO. TOTAL CLIENT	NO. UNIQUE WORDS
CLUSTER 1	2000	726
CLUSTER 2	3500	1052
CLUSTER 3	5000	1256
CLUSTER 4	6500	1498
CLUSTER 5	8000	1643
CLUSTER 6	9500	1778

summarized in Table 1 (each client is associated with a single word). Additionally, we adhere to the well-established *Zipf's* distribution as described by (Wang et al., 2019) to determine the frequency distribution of unique words within each cluster, and these unique words are not shared across the other clusters.

Metrics. For evaluating the algorithm, we employ the metrics of *recall* and *F1 score*. Furthermore, we explore the impact of different privacy parameters ε from 0.5 to 9.5 to examine the effects on the algorithm's performance.

3.2. Evaluation

In this section, we conduct experiments to evaluate our proposed algorithms. In § 3.2.1, we assess the intra-cluster algorithm to identify local heavy hitters within clusters. In § 3.2.2, we examine the cross-cluster algorithm to identify global heavy hitters across non-IID data clusters. In addition, we assess the impact of the expected number (k) of heavy hitters on performance in § 3.2.3.

3.2.1. EVALUATIONS OF THE INTRA-CLUSTER (IC)

This section evaluates the effectiveness of the IC algorithm for finding *local* heavy hitters within each cluster. We compare our approach with state-of-the-art FA heavy hitter identification algorithm, TrieHH (Zhu et al., 2020) and PEM with GRR (Wang et al., 2019).

Table 2. Ablation methods for intra-cluster algorithm with different mechanisms and group size.

MODEL	GRR/GRRX	GROUP-SIZE STRATEGY
PEM (GTU)	GRR	UNIFORM
GTF	GRR	INCREMENTAL
XTU	GRRX	UNIFORM
OURS (XTF)	GRRX	INCREMENTAL

Ablation study of GRR/GRRX and group-size strategy. We conduct an ablation study to measure the impact of perturbation mechanism and group-size strategy on top-k local heavy hitter identification. We compare the perturbation algorithm with three variants (Table 2) that differ in the noise mechanism (GRR or GRRX) and the group-size strategy (uniform or incremental). We also include TrieHH, a CDP algorithm, as a baseline for comparison.

Component Analysis. We compare our method with others on six synthetic clusters, varying privacy levels (ε). Figure 2 presents *recall* and *F1 score* for cluster sizes of $2,000$ and $9,500$, with similar results for other cluster sizes (see Appendix A.2). Our algorithm consistently outperforms other methods across clusters and privacy levels. It excels in handling non-IID data with GRRX, overcoming domain limitations, and leveraging incremental group-size for increased information utilization. Moreover, our IC algorithm effectively handles *small cluster sizes*, ideal for scenarios with fewer available clients. While similar to XTU, our method surpasses it due to incremental group-size, enhancing informativeness. PEM and GTF exhibit poorer performance due to domain limitations. TrieHH performs well only for the $9,500$ client cluster, indicating *sensitivity* to cluster size.

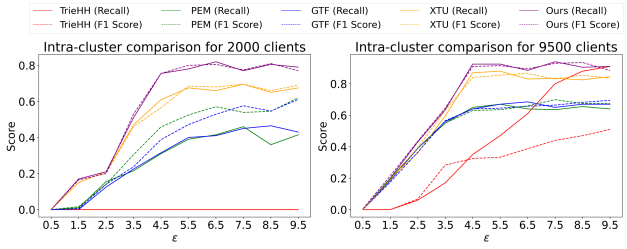

Figure 2. Comparison for finding top-5 heavy hitters in clusters with $2,000$ and $9,500$ clients.

3.2.2. EVALUATIONS OF THE CROSS-CLUSTER (CC)

In this section, we conduct evaluations of our CC algorithm for the identification of *global* heavy hitters across non-IID clusters. We begin by presenting the experimental assessment of our algorithm on synthetic data and subsequently evaluate its performance on real datasets.

Global heavy hitters identification. We compare the performance of our CC algorithm with TrieHH, a baseline method that uses CDP, for identifying the *global heavy hitters* across *non-IID clusters*. We measure the *recall* and *F1 scores* of the methods under different values of the privacy parameter ε, ranging from 0.5 to 9.5. The global heavy hitters are the union of the local heavy hitters in each cluster, and the higher the relative frequency of a local heavy hitter in its cluster, the more likely it is to be a global result.

Performance comparison for Synthetic Data. We aggregate the same six synthetic clusters (Table 2) used in the intra-cluster experiment into a single dataset comprising $34,500$ clients. Figure 3 illustrates the *recall* and *F1 scores* of our algorithm and TrieHH at different privacy levels (ε). Our algorithm consistently outperforms TrieHH across all privacy levels, underscoring its ability to effectively filter out noisy local heavy hitters and handle non-IID data.

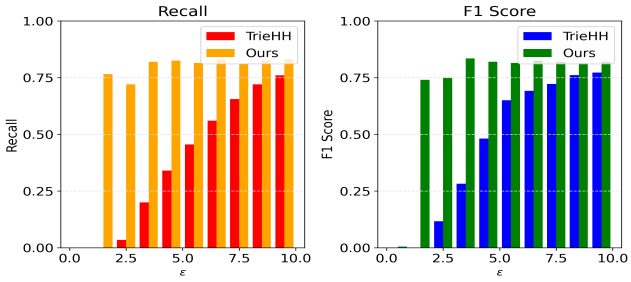

Figure 3. Comparison of Ours and TrieHH for cross-cluster non-IID heavy hitters identification at various ε levels.

Performance comparison for Real Data Next, we use two real datasets, *Sentiment140* (Go et al., 2009) and *Reddit* (Ofer, 2018), to simulate two non-IID clusters, each representing a cluster with non-IID data. To alleviate the computational and communication burdens and address the issue of client availability, we employ weighted sampling to carefully select a total of $20,000$ words from each cluster while preserving the frequency distribution that is inherent to the original dataset. This cost-effective design, which draws inspiration from prior research on Federated Learning (FL) (Luo et al., 2021; 2022), enables us to make optimal use of limited resources while ensuring the data remains representative. Table 3 shows the number of clients and unique words before and after sampling. We then apply our algorithm to identify the top-k heavy hitters across the clusters. The results also indicate that our algorithm consistently outperforms TrieHH in most cases, demonstrating its superior accuracy and efficiency in handling real non-IID data. Additional details on the performance evaluation can be found in Appendix A.3. However, we observed that our algorithm's performance deteriorates when ε is too small. This can be attributed to the fact that a smaller ε corresponds to a higher level of privacy protection, which introduces more noise in the data perturbation and aggregation process under LDP.

Table 3. Statistics of the Real Datasets Before and After Sampling

DATASET	NO. CLIENTS	NO. UNIQUE WORDS
SENTIMENT140	695,524	120,164
REDDIT	256,521	24,164
SAMPLED SENTIMENT140	14,600	6,082
SAMPLED REDDIT	5,400	1,987

3.2.3. IMPACT OF K

We evaluate the impact of k, the number of expected heavy hitters, on the real datasets. The results of varying k are shown in Figure 4. We achieve high recall and F1 scores across different values of k compared to other methods. Our algorithm has a stable performance when varying k from 3 to 5, and the recall and F1 scores do not change much as k increases, indicating that it can handle different levels of granularity and diversity in the data.

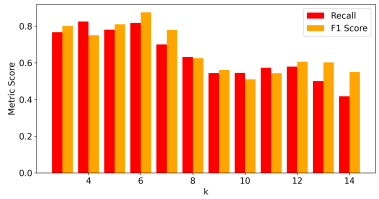

Figure 4. Varying k from $3 \sim 14$

3.3. FedCampus Application: Step Count Analysis

This section illustrates the application of our algorithm within FedCampus, a platform that facilitates privacy-preserving federated analytics on a campus-wide scale. One specific application within FedCampus involves analyzing *step counts* among distinct participant clusters. Step counts inherently involve sensitive personal data, necessitating privacy safeguards. Therefore, we utilize our algorithm to identify the most prevalent step counts, known as "heavy hitters", across various clusters.

Table 4. Collection of Step Count and Device Validity

CLUSTER	TOTAL VALID STEPS	NO. VALID DEVICES
1	195	30
2	213	31
3	204	31
4	87	20
5	164	25

Data Collection and Preprocessing. Our study involved participants with varying backgrounds and degrees of physical activity. Each participant was equipped with a wearable device configured to register their daily step count automatically. To capture the non-IID nature of the step count data, we meticulously arranged participant clusters to ensure a broad representation of walking routines, lifestyles, and activity levels (Table 4 presents the statistics of the data amassed from FedCampus, including the total number of daily steps for each participant over one week). Each cluster exhibited a distinct distribution of step counts, thereby encapsulating real-world disparities and dependencies. Moreover, all collected data were anonymized through the removal of identifiable user information. Moreover, we conducted rigorous manual scrutiny to eliminate invalid data, such as missing values or outliers.

To augment privacy safeguards and mitigate data sensitivity, we applied quantization methodologies at various precision levels. These strategies partitioned the step count data into discrete intervals and depicted them with fewer bits. By approximating interval modes instead of precise values, we further enhance privacy protection while preserving utility.

Evaluation of Results. We utilized the proposed algorithm to compute *mode intervals* for step counts, representing ranges of frequently occurring values. The chosen precision level for quantization impacts the balance between utility and privacy. Figure 5 delineates the mode interval ranges across disparate quantization levels. Elevated precision levels result in narrower intervals, thus providing enhanced utility but compromising privacy. Conversely, reduced precision levels produce broader intervals, forfeiting some utility to amplify privacy. Additional details regarding FedCampus can be found in Appendix A.4.

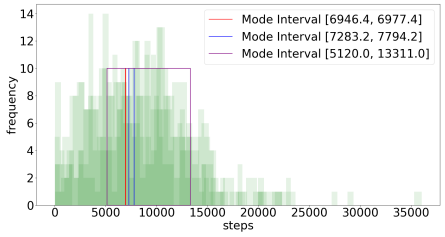

Figure 5. Mode intervals with different levels of quantization.

4. Conclusion and Future Works

In conclusion, we have introduced a federated heavy hitters identification algorithm for non-IID scenarios. Our hierarchical design achieves good performance at identifying local and global heavy hitters on both synthetic and real datasets, while effectively managing privacy risks and utility loss. We also demonstrated FedCampus, our privacy-preserving campus-scale statistics analysis platform. Our empirical study reveals successful heavy hitter identification via our LDP-based perturbation method with uniform privacy parameters. For future work, we aim to incorporate individual privacy preferences, conduct a formal privacy analysis. These efforts will enhance the customization of privacy protection and contribute to a more rigorous understanding of the privacy guarantees provided by our approach.

Acknowledgements

We would like to acknowledge the support received from the Kunshan Government Research (KGR) Funding 23KKSGR024 for the research conducted by Jiaqi Shao and Bing Luo.

References

Acharya, J., Bonawitz, K. A., Kairouz, P., Ramage, D., and Sun, Z. Context-aware local differential privacy. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Bassily, R., Nissim, K., Stemmer, U., and Thakurta, A. Practical locally private heavy hitters, 2017. URL https://arxiv.org/abs/1707.04982.

Bhaskar, R., Laxman, S., Smith, A., and Thakurta, A. Discovering frequent patterns in sensitive data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 503–512, 2010.

Bodon, F. A trie-based apriori implementation for mining frequent item sequences. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pp. 56–65, 2005.

Bonomi, L. and Xiong, L. Mining frequent patterns with differential privacy. *Proceedings of the VLDB Endowment*, 6(12):1422–1427, 2013.

Cormode, G. and Bharadwaj, A. Sample-and-threshold differential privacy: Histograms and applications. In *International Conference on Artificial Intelligence and Statistics*, pp. 1420–1431. PMLR, 2022.

Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., and Wang, T. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pp. 1655–1658, 2018.

Dwork, C. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pp. 1–19. Springer, 2008.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pp. 486–503. Springer, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006b.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 715–724, 2010.

Elkordy, A. R., Ezzeldin, Y. H., Han, S., Sharma, S., He, C., Mehrotra, S., and Avestimehr, S. Federated analytics: A survey, 2023.

Erlingsson, Ú., Pihur, V., and Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

Go, A., Bhayani, R., and Huang, L. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Jia, J. and Gong, N. Z. Calibrate: Frequency estimation and heavy hitter identification with local differential privacy via incorporating prior knowledge, 2018.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Luo, B., Li, X., Wang, S., Huang, J., and Tassiulas, L. Cost-effective federated learning design. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2021.

Luo, B., Xiao, W., Wang, S., Huang, J., and Tassiulas, L. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pp. 1739–1748. IEEE, 2022.

Norouzi, M., Fleet, D. J., and Salakhutdinov, R. R. Hamming distance metric learning. *Advances in neural information processing systems*, 25, 2012.

Ofer, D. Sarcasm on reddit, May 2018. URL https://www.kaggle.com/datasets/danofer/sarcasm?select=train-balanced-sarcasm.csv.

Ramage, D. Federated analytics: Collaborative data science without data collection, May 2020. URL https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html.

Wang, N., Xiao, X., Yang, Y., Hoang, T. D., Shin, H., Shin, J., and Yu, G. Privtrie: Effective frequent term discovery under local differential privacy. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 821–832. IEEE, 2018.

Wang, T., Blocki, J., Li, N., and Jha, S. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pp. 729–745, 2017.

Wang, T., Li, N., and Jha, S. Locally differentially private heavy hitter identification. *IEEE Transactions on Dependable and Secure Computing*, 18(2):982–993, 2019.

Zhu, W., Kairouz, P., McMahan, B., Sun, H., and Li, W. Federated heavy hitters discovery with differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pp. 3837–3847. PMLR, 2020.

A. Appendix

A.1. Proof of Theorem

Theorem A.1. *Given a dataset consisting of binary strings, a similarity measure based on the Hamming distance can be established such that, for any pair (c, c'), the similarity threshold $\delta = 0.5 * len(c)$, where $len(c)$ denotes the bit length of c, guarantees avoidance of the degenerate case wherein all items are construed as identical.*

Proof. Suppose there are n bit-strings $\Gamma = \{c_1, c_2, \ldots, c_n\}$, and the maximum bit length of the strings is represented as $L = max\{len(c_i), \ldots, len(c_n)\}$.

For each pair of c_i and c_j, if their Hamming distance satisfies $d(c_i, c_j) < \delta$, c_i and c_j can be deemed similar. We establish the similarity threshold as $\delta = \alpha * len(c_i)$, where $\alpha \geq \frac{d(c_i,c_j)}{len(c_i)}$ for similar entities.

To begin, the total Hamming distance of all pairs of (c_i, c_j) is $\sum_{i=1}^{n}\sum_{j=1}^{n} d(c_i, c_j) = \sum_{i=1}^{L} m_i(n - m_i)$, where m_i represents the count of 1's in all $c_i \in \Gamma$ for the i^{th} bit position, and $n - m_i$ represents the count of 0's in all $c_i \in \Gamma$ for the i^{th} bit position.

To calculate the expected value of the total Hamming distance, we assume the value (1 or 0) of the i^{th} bit position for all $c_i \in \Gamma$ follows a binomial distribution.

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} d(c_i, c_j)\right] &= \mathbb{E}\left[\sum_{i=1}^{L} m_i(n - m_i)\right] \\
&= \sum_{i=1}^{L} \mathbb{E}\left[m_i(n - m_i)\right] \\
&= \sum_{i=1}^{L} \mathbb{E}\left[m_i\right]\mathbb{E}\left[n - m_i\right] \\
&= L(\frac{n}{2})^2
\end{aligned}$$

If all entities in Γ is similar, then we calculate the expectation of α:

$$\begin{aligned}
\mathbb{E}[\alpha] &\geq \mathbb{E}[\frac{d(c_i, c_j)}{max\{len(c_i, c_j)\}}] \\
&\geq \frac{1}{L}\mathbb{E}[d(c_i, c_j)] \\
&= \frac{1}{(n(n-1)/2) * L}\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} d(c_i, c_j)\right] \\
&> 0.5
\end{aligned}$$

Thus, to discern the *discrepancies* across all entities in Γ, we can set $\alpha = 0.5$ as the similarity threshold, *i.e.*, $\delta = 0.5 * len(c_i)$.

□

A.2. Experimental Results for Intra-cluster Local Heavy Hitters Identification

In this section, we present the experimental results for our intra-cluster local heavy hitters identification algorithm, which aims to identify high-frequency words within each cluster while preserving the privacy of the participants. We evaluate our algorithm using six synthetic datasets with varying numbers of clients and unique words per cluster, as shown in Table 1. The frequency of the unique words in each cluster follows Zipf's distribution.

We compare our algorithm with three variants that differ in the perturbation mechanism (GRR or GRRX) and the group-size strategy (uniform or incremental), as well as TrieHH, a baseline method that uses CDP. We measure the recall and F1 scores of the methods under different values of the privacy parameter ε, ranging from 0.5 to 9.5. Figure 6 shows that our algorithm consistently outperforms the other methods across all clusters and privacy levels, demonstrating its effectiveness for non-IID data with GRRX, which overcomes the domain limitation, and incremental group-size, which leverages more information from later groups. We also find that our algorithm can handle small cluster sizes, which means it can work with clusters with fewer clients and benefit for scenarios with fewer clients. Our method is similar to XTU, but it outperforms XTU because it uses incremental group-size, increasing informativeness. PEM and GTF are also similar methods, but they perform worse than ours because they suffer from domain limitations. Furthermore, we observed that TrieHHwith poorly on most clusters, except for the one with $9,500$ clients, where it achieves high recall and F1 score at high ε values. This suggests that TrieHH is sensitive to cluster size.

Overall, our algorithm demonstrates superior performance in identifying local heavy hitters within each cluster while preserving the participants' privacy.

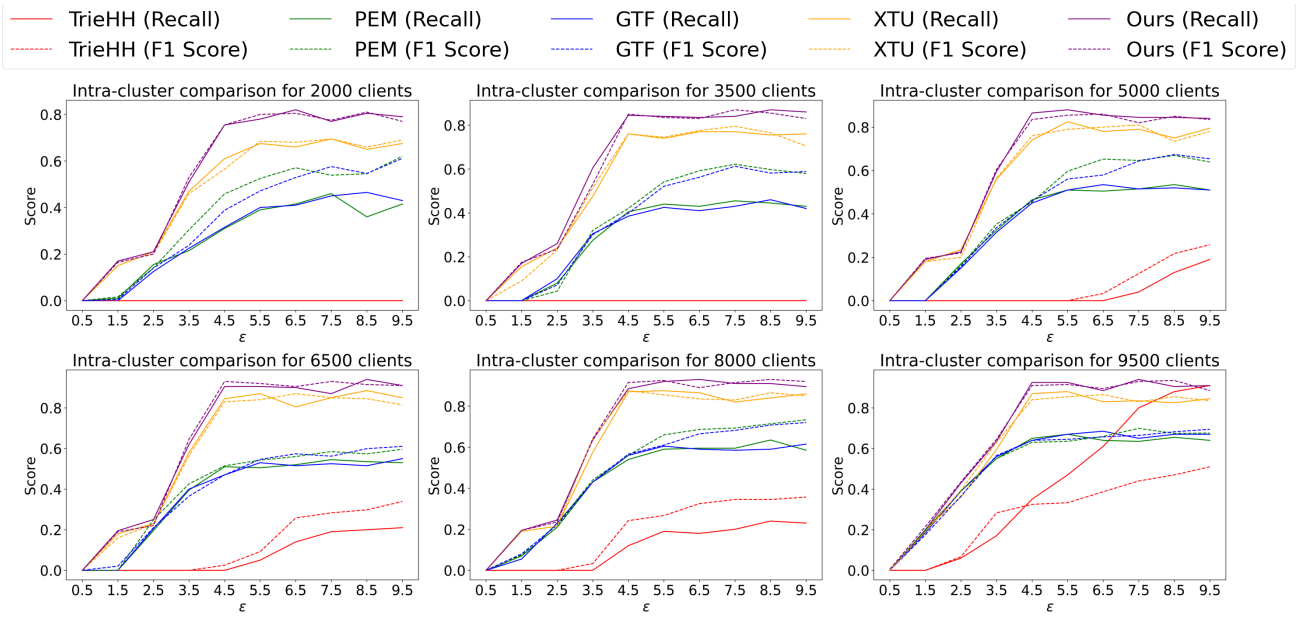

Figure 6. Comparison for Intra-cluster identification within six different clusters

A.3. Comparative Analysis of Real Data Performance

In this section, we focus on the evaluation of our algorithm with two real-world datasets, specifically *Sentiment140* and *Reddit*. These datasets facilitate the simulation of two non-IID clusters, with each signifying a unique cluster encompassing non-IID data.

To mitigate computational and communication burdens and accommodate for client availability, we utilize weighted sampling to select a total of $20,000$ words from each cluster while upholding the frequency distribution intrinsic to the original data. Table 3 delineates the number of clients and unique words pre and post-sampling. Subsequent to this, we implement our algorithm for the identification of the top-k heavy hitters across these clusters. The resultant findings, as illustrated in Figure 7, indicate that our algorithm consistently outperforms TrieHH in the majority of scenarios, thus demonstrating its enhanced accuracy and efficiency when dealing with non-IID data. Nevertheless, we noted a degradation in our algorithm's performance when ε is excessively diminutive. This can be ascribed to the fact that a smaller ε corresponds to heightened privacy safeguards, which inadvertently introduces amplified noise and uncertainty within the data perturbation and aggregation phases under the mechanism of LDP.

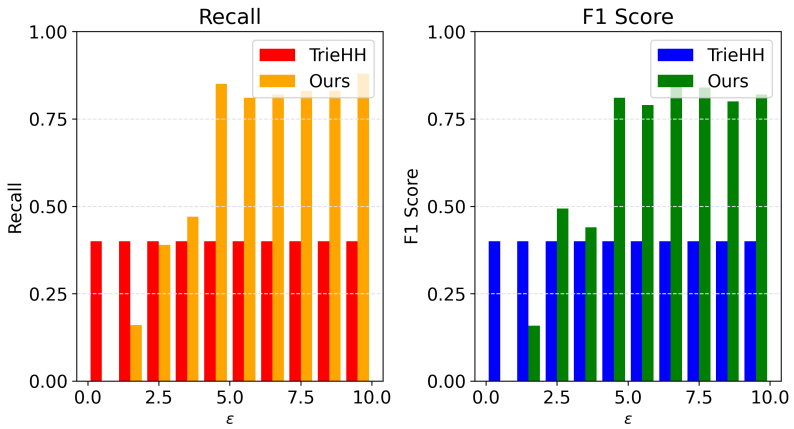

Figure 7. Performance comparison of our algorithm and TrieHH on real data, measured in terms of *recall* and *F1 scores* across different values of the privacy parameter ε.

A.4. FedCampus Demo

FedCampus is a platform that facilitates federated analytics (FA) on data collected from various edge devices, such as smartphones and smartwatches, within a campus-scale environment. One of the features of this platform is its ability to maintain the privacy and security of the participants throughout the analytics process. By leveraging advanced privacy-preserving techniques, FedCampus provides a powerful tool for conducting secure and privacy-preserving analytics on edge devices. The application's interface is shown in Figure 8.

Peering into the future, FedCampus aspires to extend its support for federated learning, along with an array of diverse computational and analytical paradigms, ushering in a new era of privacy-preserving data analysis and machine learning on the edge. This broad spectrum of capabilities envisages transforming FedCampus into a comprehensive tool for diverse research and practical applications in distributed, privacy-aware learning and analytics.

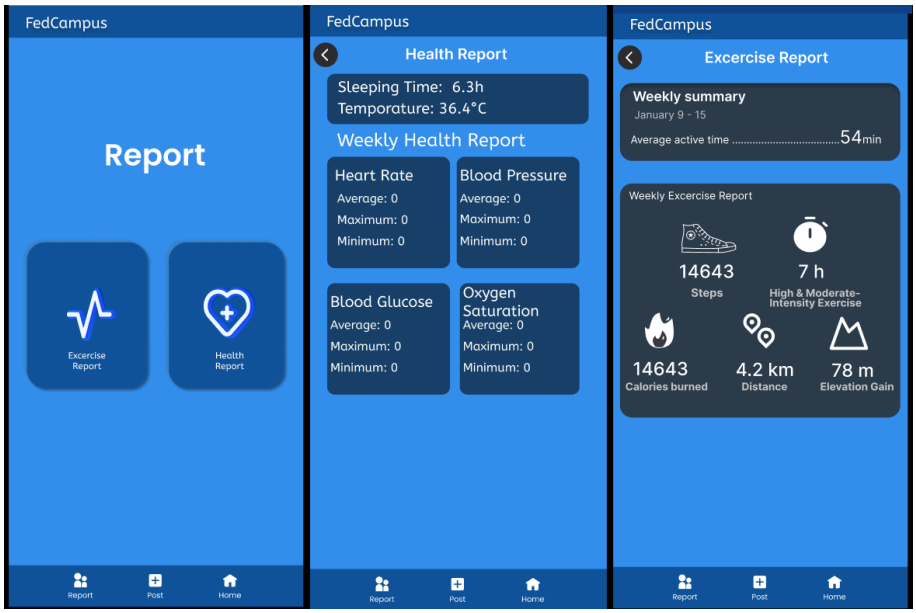

Figure 8. FedCampus Demo