Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 221 (2023) 664–672

Procedia Computer Science

www.elsevier.com/locate/procedia

Tenth International Conference on Information Technology and Quantitative Management (ITQM 2023)

The Utility Impact of Differential Privacy on Credit Card Data in Machine Learning Algorithms

Xiaopeng LUO[1,a,b], Siyuan WANG[1,c], Haolong CHEN[1,c], Zongwei LUO*[,d,e]

[a]*Hong Kong Baptist University, Kowloon Tong, Hong Kong, China*
[b]*IRADS, BNU-HKBU United International College,Zhuhai, China*
[c]*Faculty of Science and Technology, BNU-HKBU United International College,Zhuhai, China*
[d]*BNU-UIC Institute of AI and Future Networks, Beijing Normal University, Zhuhai,China, 519000*
[e]*Artificial Intelligence and Data Science Research Hub, BNU-HKBU United International College,Zhuhai, China*

Abstract

With the development of networks and financial technologies, credit card data is increasingly being used in various fields of data analysis such as user behavior, financial transactions, and market analysis. These fields often use multiple machine learning algorithms for data mining on credit card dataset. It is worth noting that credit card data contains more diverse and comprehensive information compared to traditional data, and the dataset may contain multiple data types. At the same time, credit card data may also expose users' privacy information. Differential privacy algorithms can add random noise to the data set, protecting sensitive information while ensuring certain data utility. However, there has been little research on the use of differential privacy algorithms on credit card data in multiple machine learning algorithms, and there has been insufficient exploration of the utility impact of differential privacy on complex credit card data. These research gaps exist in both the financial technology and privacy protection industries. Therefore, this paper applies differential privacy to credit card data in multiple classic machine learning algorithms, discusses the utility impact of various differential privacy algorithms on credit card data, and compares the performance of credit card data sets protected by differential privacy in different algorithms.

Keywords: credit card default; differential privacy; machine learning algorithm

1

1. Introduction

The importance of privacy in credit card transactions cannot be overstated. With the proliferation of digital transactions, the risk of privacy breaches has increased significantly. As highlighted by Martin, Borup, and Porse

[1]These authors contributed equally to this work and should be regarded as co-first authors

*Corresponding author :Zongwei LUO.
E-mail address: lzwqhk@outlook.com.

10.1016/j.procs.2023.08.036

(2017), the privacy of credit card transactions is a critical aspect of consumer trust and confidence in financial institutions. They argue that the lack of adequate privacy measures can lead to significant financial losses and damage to the reputation of financial institutions.

In recent years, local differential privacy (LDP) has emerged as a promising approach to protect private information. It has been successfully applied in various domains to ensure data privacy. For instance, Li et al. (2019) demonstrated the effectiveness of LDP in protecting patient data in a federated learning setup for brain tumour segmentation. Similarly, Wang, Tong, and Shi (2020) proposed a LDP-based framework for federated learning of Latent Dirichlet Allocation (LDA) models, which provided theoretical guarantees on both data privacy and model accuracy.

However, the application of LDP in the credit card industry remains limited. A study by de Montjoye, Radaelli, Singh, and Pentland (2015) revealed that four spatiotemporal points are enough to uniquely reidentify 90% of individuals in credit card metadata, indicating a significant privacy risk. While some efforts have been made to apply federated learning for credit card fraud detection (Yang et al., 2019), the use of LDP in this domain is still in its infancy.

This gap in the literature motivates our work. In this paper, we propose a novel approach to apply LDP in the credit card industry to enhance privacy protection. We believe that our work will contribute to the ongoing efforts to improve privacy in credit card transactions and inspire further research in this area.

2. Data description

To enhance the reliability of the experiment and the convincing nature of the results, we implemented the algorithm discussed in the paper on two distinct numerical datasets related to credit card transactions.

2.1. Numeric dataset 1

We use credit card fraud detection dataset from Kaggle website which comprises transactions conducted via credit cards in September 2013 by European cardholders. The dataset spans a period of two days, during which a total of 284,807 transactions were recorded. The dataset contains 492 instances of fraudulent transactions, thereby making the dataset significantly imbalanced. The proportion of positive class instances, which denote frauds, is a mere 0.172% of all transactions. The dataset exclusively contains numerical input variables, which have been derived as a result of a Principal Component Analysis (PCA) transformation.

2.2. Numeric dataset 2

The dataset under consideration in this study was sourced from the UCI Machine Learning Repository, titled ”Default of Credit Card Clients” (Yeh, I-Cheng, 2016). The dataset was created with the objective of understanding the default payments of customers in Taiwan, with a specific focus on comparing the predictive accuracy of the probability of default across various data mining methods. The primary intent behind this research was to move beyond the binary classification of clients as ’credible’ or ’non-credible’, and instead provide a more nuanced understanding of the actual probability of default. Comprising 30,000 instances and 24 attributes, the dataset presents a mix of both integer and real data types. However, for the purpose of this paper, the study was confined to the examination of the numeric attributes. These include:

X1: Amount of the given credit (NT dollar) - This numerical attribute captures the amount of credit given, inclusive of both individual consumer credit and supplementary credit for their family.

X5: Age (in years) - This attribute records the age of the clients.

X6 to X11: History of past payment - These attributes provide an historical record of past payments from April to September 2005. The repayment status is encoded numerically, with -1 signifying ’pay duly’, and increasing positive values indicating payment delay for corresponding number of months.

X12 to X17: Amount of bill statement (NT dollar) - These attributes document the amount of bill statement from April to September 2005.

X18 to X23: Amount of previous payment (NT dollar) - These attributes detail the amount paid in the previous months from April to September 2005.

3. Comparison of classical machine learning algorithm on credit card dataset

Following our comprehensive exploration of the credit card fraud detection issue, we proceeded to apply an array of traditional machine learning algorithms as part of our initial analytical approach. These algorithms were chosen due to their proven efficacy and widespread use in similar data-rich scenarios. By employing these classical methodologies, we aimed to establish a baseline performance, providing us with a fundamental point of reference for each algorithm. This preliminary measurement will serve a crucial role as we endeavor to assess the impact of noise introduction to our models, facilitating a robust comparison of the algorithmic performance before and after the integration of such perturbations. Our subsequent sections will elaborate on the results of these explorations, offering insight into the intricate dynamics of machine learning algorithms in the presence of noise.

3.1. CART tree

CART use Gini index as metric to calculate the impurity of the current node:

$$Gini = 1 - \sum_{i=1}^{c} p_i^2$$

The pseudo-code for the CART algorithm is given by:

Algorithm 1 CART Algorithm

Require: Training dataset D, maximum depth $d_{\max}$
Ensure: Decision tree T

```
Initialize tree T with root node containing all training instances
for i = 1 to d_max do
    if node n is not a leaf then
        Calculate the best split point based on the Gini index or mean squared error
        Split the node into two child nodes based on the best split point
        Assign the training instances to the child nodes based on the split condition
    end if
end for
return T
```

The algorithm works by initializing a tree with a root node that contains all training instances. It then iteratively splits each internal node into two child nodes based on the best split point, which is calculated based on the Gini index for classification or mean squared error for regression. The training instances are then assigned to the child nodes based on the split condition. The algorithm continues until the maximum depth of the tree is reached.

3.2. Bayesian classifier

A Bayesian algorithm begins by specifying a probability distribution over the space of possible hypotheses, called the prior distribution. This prior distribution represents our initial beliefs about the hypotheses before any data has been observed. As data becomes available, the algorithm uses Bayes' theorem to update the prior distribution and obtain a posterior distribution, which represents our updated beliefs about the hypotheses after taking the data into account. This posterior distribution can then be used to make predictions about future data.

The formula for Bayes' theorem is given by:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where $P(A|B)$ is the posterior probability of hypothesis A given the evidence B, $P(B|A)$ is the likelihood of the evidence B given hypothesis A, $P(A)$ is the prior probability of hypothesis A, and $P(B)$ is the marginal probability of the evidence B.

3.3. SVM

Support vector machines (SVM) [1] can be used for classification or regression tasks.The goal of an SVM is to find the decision boundary that maximally separates the data points of different classes, or in the case of regression, to find the line or hyperplane that best fits the data. The formula for the SVM algorithm is given by:

$$\min_{w,b} \frac{1}{2}|w|^2 + C\sum_{i=1}^{N} \xi_i$$

Subject to the constraints:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i = 1, 2, \ldots, N$$

Where w is the weight vector, b is the bias term, C is a hyperparameter that controls the trade-off between the margin and the number of misclassifications, x_i is a data point, y_i is the true class of the data point x_i, and ξ_i is the slack variable for the ith data point. The objective function is a trade-off between the margin, which is controlled by the first term $\frac{1}{2}|w|^2$, and the number of misclassifications, which is controlled by the second term $C\sum_{i=1}^{N} \xi_i$. The constraints ensure that each data point is correctly classified or lies within the margin.

3.4. KNN

KNN is based on the idea that data points that are close to each other in feature space (i.e., have similar feature vectors) are likely to belong to the same class. We would follow to achieve that:

Given a training set of data points $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ and a new data point (x, y)

Identify the k number of points in the training set that are closest (in a Euclidean sense) to (x, y)

Classify the new data point (x, y) based on the majority class among those k points.

3.5. K-means

K-means works by first randomly selecting k data points as the initial cluster centers, and then iteratively assigning each data point to the cluster with the nearest center and updating the cluster centers to the mean of the assigned data points. This process is repeated until the cluster centers stop changing or a maximum number of iterations is reached.

The formula for the within-cluster sum of squares for a given clustering is given by:

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2$$

Where k is the number of clusters, C_i is the set of data points assigned to cluster i, μ_i is the cluster center of cluster i, and x is a data point. The within-cluster sum of squares is a measure of how well the data points are clustered around their respective cluster centers. A lower within-cluster sum of squares indicates a better clustering.

3.6. Comparison and Result analysis

The machine learning algorithms considered in this study are KNN, Bayesian, SVM with RBF kernel, SVM with polynomial kernel, CART tree, and Kmeans. The accuracy of these algorithms is compared on two datasets, a credit card fraud detection dataset (Dataset 1) and a credit card default payment dataset (Dataset 2). Both datasets are numeric and were sourced from Kaggle and the UCI Machine Learning Repository, respectively.

The following are each algorithms' accuracy in dataset1 and dataset2:

	KNN	Bayesian	SVM_RBF	SVM_poly	CART tree	Kmeans
Accuracy	0.999567	0.978465	0.999391	0.999438	0.999099	0.921595

The observed performance of the machine learning algorithms across the two datasets might be rooted in their individual characteristics. K-Nearest Neighbors (KNN) exhibits robust performance on both datasets, potentially due to its adept handling of complex decision boundaries, though it may falter with increased dimensionality

	KNN	Bayesian	SVM_RBF	SVM_poly	CART tree	Kmeans
Accuracy	0.777778	0.352667	0.784889	0.784111	0.695111	0.321633

or noisy data. The performance drop of the Bayesian classifier on Dataset 2 might be attributed to its inherent assumption of feature independence, which may not apply given the complex interrelationships in Dataset 2. Superior accuracies achieved by Support Vector Machines (SVMs) with Radial Basis Function (RBF) and Polynomial (poly) kernels could be ascribed to their ability to model intricate, non-linear decision boundaries. Although the Classification and Regression Trees (CART) algorithm demonstrates reasonable performance on both datasets, its superior performance on Dataset 1 may suggest a tendency to overfit, thereby reducing its generalization performance. Lastly, the marked decline in Kmeans' performance when applied to Dataset 2 might be due to its limitations in handling high-dimensional data or non-spherical clusters with varying sizes and densities. This differential performance underscores the influence of algorithm-specific properties on their effectiveness, emphasizing the importance of algorithm selection commensurate with dataset characteristics.

4. Comparison on local differential privacy algorithm

4.1. Laplace mechanism

Given a function $f : D \to R^d$ that operates on a database D, the Laplace mechanism adds noise that is scaled to the sensitivity of f defined as $\Delta f = max_{D,D':\|D-D'\|_1=1}\|f(D) - f(D')\|_1$.The mechanism is defined as:

$$M(D) = f(D) + (Y_1, ..., Y_d)$$

where each Y_i is drawn from the Laplace distribution $Lap(\Delta f/\epsilon)$.The Laplace distribution with location 0 and scale b is defined as:

$$Lap(x|0, \lambda) = \frac{1}{2\lambda} exp(-\frac{|x|}{\lambda})$$

This mechanism ensures ϵ-differential privacy.

4.2. Duchi et al.'s mechanism

The Duchi mechanism is a method designed for handling multidimensional numeric data under the Local Differential Privacy (LDP) model. It is particularly useful for tasks such as estimating the mean of numerical attributes and the frequency of categorical values. The mechanism operates by perturbing a multidimensional tuple,$t_i \in [-1, 1]^d$, from each user.The perturbed tuole, t_i^*, has non-zero value on k attributes, where k is determined by:

$$k = \max\left(1, \min\left(d, \left\lceil \frac{\epsilon}{2.5} \right\rceil\right)\right)$$

Here, d, is the dimension of the tuple and ϵ is teh privacy budget. Each of the k attributes is selected uniformly at random(without replacement)from all d attributes of the tuple.The pretubed value for each selected attribute, $t_i^*[A_j]$, is set to $\frac{d}{k} \cdot x$, where x is generated by the mechanism given the original attribute value $t_i[A_j]$ and $\frac{\epsilon}{k}$ as input.

The Duchi mechanism also incorporates the composition property of differential privacy. If a user participates in m iterations, and the i^{th} gradient returned by the user satisfies ϵ_i differential privacy, then the total privacy budget ϵ should satisfy $\sum_{i=1}^{m} \epsilon_i \leq \epsilon$. If we set $\epsilon_i = \frac{\epsilon}{m}$, the amount of noise in each gradient becomes $O\left(\frac{\sqrt{m \cdot d \cdot \log d}}{\epsilon}\right)$, and the group size becomes $|G| = \Omega\left(\frac{m^2 \cdot d \cdot \log d}{\epsilon^2}\right)$, which is m^2 times larger compared to the case where each user only participates in at most one iteration. The following is the pesudo code of Duchi mechanism.

Algorithm 2 Duchi et al.'s Solution for Multidimensional Numeric Data

Require: Tuple $t_i \in [-1, 1]^d$, privacy parameter ϵ
Ensure: Perturbed tuple t_i^* in $\{-B, B\}^d$
1: Generate a random tuple $v \in \{-1, 1\}^d$ by sampling each $v[A_j]$ independently from the following distribution:
2: $\Pr[v[A_j] = x] = \begin{cases} \frac{1}{2} + \frac{1}{2}t_i[A_j], & \text{if } x = 1 \\ \frac{1}{2} - \frac{1}{2}t_i[A_j], & \text{if } x = -1 \end{cases}$
3: Let T^+ (resp. T^-) be the set of all tuples $t^* \in \{-B, B\}^d$ such that $t^* \cdot v \geq 0$ (resp. $t^* \cdot v \leq 0$);
4: Sample a Bernoulli variable u that equals 1 with $\frac{e^\epsilon}{e^\epsilon+1}$ probability;
5: **if** $u = 1$ **then**
6: **return** a tuple uniformly at random from T^+;
7: **else**
8: **return** a tuple uniformly at random from T^-;
9: **end if**

4.3. Piecewise Mechanism

The Piecewise Mechanism (PM) is a method designed for empirical risk minimization under local differential privacy, particularly suited for unidimensional numeric data. The PM operates by returning a perturbed value, denoted as t_i^* for a given input t_i.

The perturbed value t_i^* is determined as follows:

$$t_i^* = \begin{cases} e^\epsilon + \frac{1}{e^\epsilon - 1} & \text{or} \\ -e^\epsilon + \frac{1}{e^\epsilon - 1} \end{cases}$$

This holds even when the input tuple $t_i = 0$. Consequently, the noisy value t_i^* always has an absolute valye greater than 1, leading to a variance larger than 1 when $t_i = 0$, irresoective of the size of the privacy budget ϵ.

The worst-case variance of the noisy values returned by the PM is given by:

$$\mathrm{Var}(t_i^*) = \left(e^\epsilon + \frac{1}{e^\epsilon - 1}\right)^2$$

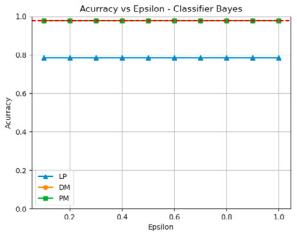
This occurs when $t_i = 0$. Upon receiving the perturbed tuples output by the PM, the aggregator computes the average value of the attribute over all users to obtain an estimated mean.

Algorithm 3 Piecewise Mechanism for Multidimensional Numeric Data

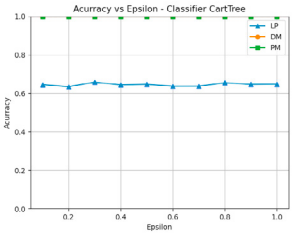
Require: A tuple $t_i \in [-1, 1]^d$ and privacy parameter ϵ
Ensure: A perturbed tuple $t_i^* \in [-C, C]^d$
1: **for** each dimension j of t_i **do**
2: Sample x uniformly at random from $[0, 1]$
3: **if** $x < \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$ **then**
4: Sample $t_i^*[j]$ uniformly at random from $[\ell(t_i[j]), r(t_i[j])]$
5: **else**
6: Sample $t_i^*[j]$ uniformly at random from $[-C, \ell(t_i[j])) \cup (r(t_i[j]), C]$
7: **end if**
8: **end for**
9: **return** t_i^*

4.4. Comparison and Result analysis

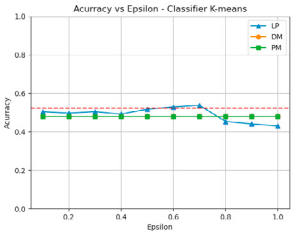
Figure 1 is the corresponding result of dataset1, each subfigure shows the accuracy trend under different privacy budget ϵ.



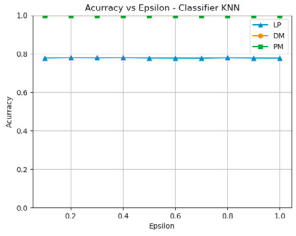
(a) numeric dataset1 Bayesian



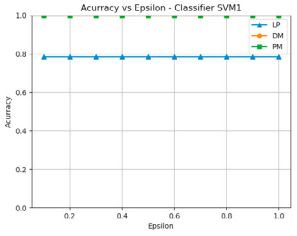
(b) numeric dataset1 CART tree



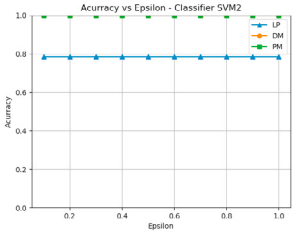
(c) numeric dataset1 Kmeans



(d) numeric dataset1 KNN

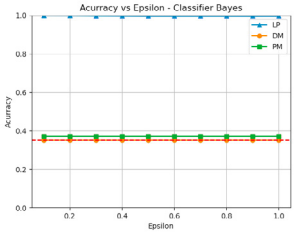


(e) numeric dataset1 SVM RBF kernel

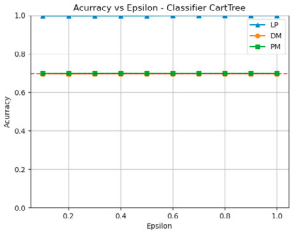


(f) numeric dataset1 SVM polynomial kernel

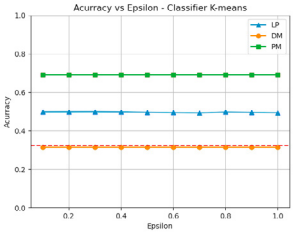
Fig. 1. Numeric dataset1 results



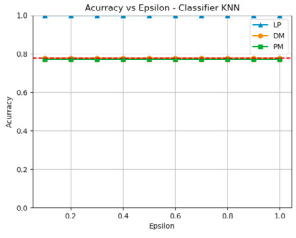
(a) numeric dataset2 Bayesian



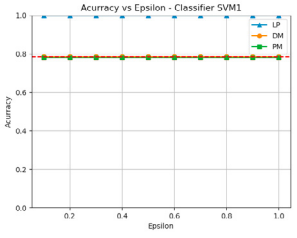
(b) numeric dataset2 CART tree



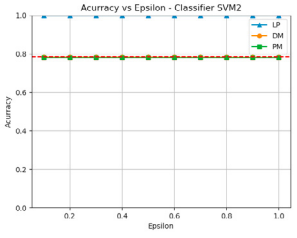
(c) numeric dataset2 Kmeans



(d) numeric dataset2 KNN



(e) numeric dataset2 SVM RBF kernel



(f) numeric dataset2 SVM polynomial kernel

Fig. 2. Numeric dataset2 results

Figure 2 is the corresponding result of dataset2, each subfigure shows the accuracy trend under different privacy budget ϵ.

Analyzing the performance of machine learning algorithms after the implementation of various Local Differential Privacy (LDP) mechanisms such as Laplace, Duchi, and Piecewise provides valuable insights into their efficacy under enhanced privacy conditions.

In relation to Dataset 1, Duchi and Piecewise mechanisms generally surpass the performance of the Laplace mechanism for all algorithms with the exception of Kmeans. This relative superiority may be attributed to the capability of Duchi and Piecewise mechanisms to better handle the specific characteristics of the data, therefore preserving data utility more effectively. In the case of Kmeans, the Laplace mechanism demonstrates marginally superior performance, which could be due to the mechanism's resilience to data noise that does not significantly impact the calculation of cluster center points, derived as average values.

Conversely, when considering Dataset 2, the Laplace mechanism consistently outperforms the Duchi and Piecewise mechanisms. This implies that the Laplace mechanism might be better equipped to manage the complex and diverse numeric attributes encompassed in this dataset, thus retaining more data utility under increased privacy guarantees.

Analyzing from the perspective of the LDP mechanisms, it is evident that the performance of machine learning algorithms does not drastically diminish with the introduction of privacy noise, particularly within a privacy budget range of [0,1]. This observation underscores the robustness of these algorithms in maintaining their accuracy even under strict privacy constraints.

In a broader perspective, all three LDP mechanisms display a commendable ability to ensure that the utility of the data remains consistent with that of the original dataset. The overall trend confirms the potential of LDP mechanisms to provide a viable trade-off between data privacy and utility, enabling the application of machine learning algorithms on private data without substantial degradation in performance. The findings from this experiment emphasize the significance of selecting the appropriate LDP mechanism, tailored to the characteristics of the dataset and the specificities of the machine learning task at hand.

5. Conclusion

As the ubiquity of credit card data surges in various domains of data analysis, including user behavior, financial transactions, and market analysis, the need for effective privacy protection mechanisms has grown exponentially. Despite the potential of differential privacy (DP) to preserve data utility while protecting sensitive information, there is a significant gap in its application to credit card data across diverse machine learning algorithms.

In the present study, we have aimed to bridge this research gap by conducting a comprehensive exploration of the utility impact of different DP algorithms on credit card datasets. This work serves as a pivotal guide in assessing the effects of combining DP and credit card data analysis.

Our findings indicate that DP mechanisms can adeptly maintain the utility of credit card data, while ensuring stringent privacy protection. We have demonstrated that the application of DP mechanisms such as Laplace, Duchi, and Piecewise does not severely degrade the performance of multiple machine learning algorithms on credit card data. We also found that the selection of an appropriate DP mechanism significantly depends on the characteristics of the dataset and the specifics of the machine learning task.

In conclusion, this research underscores the potential of differential privacy as a viable solution for ensuring user privacy in future data analysis tasks involving credit card data. By providing a comprehensive analysis of the utility impact of different DP algorithms on credit card data, this study makes a significant contribution to both the financial technology and privacy protection fields. This encourages further exploration and adoption of DP mechanisms in dealing with sensitive credit card data in data mining and analysis tasks. We envision that the insights from this study will inform future work in this critical area, paving the way towards more secure and private data analysis practices.

6. Acknowledgements

This research is partially funded by Guangdong Universities Special Key Project (Project No. 2021ZDZX3021), and in part by Guangdong Higher Education Upgrading Plan (2021-2025) of "Rushing to the Top, Making Up Shortcomings and Strengthening Special Features" with UIC research grant UICR0400052-21CTL and Guangdong AI Institute of Higher Education Project(Project No. UICR0400005-23).

References

[1] K. Martin, J. Borup, M. Porse, The privacy implications of social media surveillance for public health, Public Health Genomics 20 (4) (2017) 197–206.
[2] W. Li, F. Milletarì, D. Xu, N. Rieke, J. Hancox, W. Zhu, M. Baust, Y. Cheng, S. Ourselin, M. Cardoso, A. Feng, Privacy-preserving federated brain tumour segmentation, in: Lecture Notes in Computer Science, Springer, 2019, pp. 133–141.
[3] Y. Wang, Y. Tong, D. Shi, Federated latent dirichlet allocation: A local differential privacy based framework, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 6096–6103.
[4] Y. de Montjoye, L. Radaelli, V. Singh, A. Pentland, Unique in the shopping mall: On the reidentifiability of credit card metadata, Science 347 (6221) (2015) 536–539.
[5] W. Yang, Y. Zhang, K. Ye, L. Li, C. Xu, Ffd: A federated learning based method for credit card fraud detection, in: Lecture Notes in Computer Science, Springer, 2019, pp. 14–26.
[6] J. C. Duchi, M. I. Jordan, M. J. Wainwright, Minimax optimal procedures for locally private estimation, Journal of the American Statistical Association 113 (521) (2018) 182–201.
[7] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: TCC, 2006, pp. 265–284.
[8] Collecting and analyzing multidimensional data with local differential privacy (2019).
URL https://doi.org/10.48550/arXiv.1907.00782
[9] I.-C. Yeh, Default of credit card clients, UCI Machine Learning Repository (2016). doi:10.24432/C55S3H.