UMEÅ UNIVERSITY

Exploring Privacy-preserving Models in Model Space

Ayush Kumar Varshney

Licentiate Thesis, March 2024
Department of Computing Science
Umeå University
Sweden

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

ayushkv@cs.umu.se

ISBN (print) 978-91-8070-275-1
ISBN (pdf) 978-91-8070-276-8
ISSN 0348-0542
UMINF 24.03

Printed by City Print i NorrAB, Umeå, 2024.
Title image generated by DALL-E

Abstract

Privacy-preserving techniques have become increasingly essential in the rapidly advancing era of artificial intelligence (AI), particularly in areas such as deep learning (DL). A key architecture in DL is the Multilayer Perceptron (MLP) network, a type of feedforward neural network. MLPs consist of at least three layers of nodes: an input layer, hidden layers, and an output layer. Each node, except for input nodes, is a neuron with a nonlinear activation function. MLPs are capable of learning complex models due to their deep structure and non-linear processing layers. However, the extensive data requirements of MLPs, often including sensitive information, make privacy a crucial concern. Several types of privacy attacks are specifically designed to target Deep Learning learning (DL) models like MLPs, potentially leading to information leakage. Therefore, implementing privacy-preserving approaches is crucial to prevent such leaks. Most privacy-preserving methods focus either on protecting privacy at the database level or during inference (output) from the model. Both approaches have practical limitations. In this thesis, we explore a novel privacy-preserving approach for DL models which focuses on choosing anonymous models, i.e., models that can be generated by a set of different datasets. This privacy approach is called Integral Privacy (IP). IP provide sound defense against Membership Inference Attacks (MIA), which aims to determine whether a sample was part of the training set.

Considering the vast number of parameters in DL models, searching the model space for recurring models can be computationally intensive and time-consuming. To address this challenge, we present a relaxed variation of IP called Δ-Integral Privacy (Δ-IP), where two models are considered equivalent if their difference is within some Δ threshold. We also highlight the challenge of comparing two DNNs, particularly when similar layers in different networks may contain neurons that are permutations or combinations of one another. This adds complexity to the concept of IP, as identifying equivalencies between such models is not straightforward. In addition, we present a methodology, along with its theoretical analysis, for generating a set of integrally private DL models.

In practice, data often arrives rapidly and in large volumes, and its statistical properties can change over time. Detecting and adapting to such drifts is crucial for maintaining model's reliable prediction over time. Many approaches for detecting drift rely on acquiring true labels, which is often infeasible. Si-

multaneously, this exposes the model to privacy risks, necessitating that drift detection be conducted using privacy-preserving models. We present a methodology that detects drifts based on uncertainty in predictions from an ensemble of integrally private MLPs. This approach can detect drifts even without access to true labels, although it assumes they are available upon request.

Furthermore, the thesis also addresses the membership inference concern in federated learning for computer vision models. Federated Learning (FL) was introduced as privacy-preserving paradigm in which users collaborate to train a joint model without sharing their data. However, recent studies have indicated that the shared weights in FL models encode the data they are trained on, leading to potential privacy breaches. As a solution to this problem, we present a novel integrally private aggregation methodology for federated learning along with its convergence analysis.

Sammanfattning

Integritetsskyddande tekniker har blivit allt viktigare i den snabba utvecklingen av artificiell intelligens (AI), och särskilt inom områden som djupinlärning (DL). En central arkitektur i DL är det så kallade MLP-nätverket (Multilayer Perceptron), vilket är en typ av neuronnät. MLP består av minst tre lager av noder: ett inmatningslager, dolda lager och ett utmatningslager. Varje nod, med undantag för ingångar, är en neuron med en icke-linjär aktiveringsfunktion. MLP kan lära sig komplexa modeller tack vare sin djupa struktur och sina icke-linjära bearbetningslager. MLP:ernas omfattande databehov – och att de dessutom ofta innehåller känslig information – gör dock datasekretess till en avgörande fråga. Flera typer av integritetsattacker är specifikt utformade för att rikta sig just mot Deep Learning learning-modeller som MLP, vilket potentiellt kan leda till informationsläckor. Därför är det avgörande att implementera integritetsbevarande metoder för att förhindra sådana läckor. De flesta integritetsbevarande metoder fokuserar antingen på att skydda integriteten på databasnivå eller under modellens inferens, (slutsats/utdata). Båda metoderna har praktiska begränsningar. I den här avhandlingen utforskar vi en ny integritetsbevarande metod för DL-modeller som fokuserar på att välja anonyma modeller, dvs. modeller som kan genereras av en uppsättning olika dataset. Denna integritetsstrategi kallas Integral Privacy (IP). IP ger ett gediget försvar mot medlemsinferensattacker (Membership Inference Attacks, MIA), som syftar till att avgöra om ett prov var en del av träningsdata. Med tanke på det stora antalet parametrar i DL-modeller kan det vara beräkningsintensivt och tidskrävande att söka i modellutrymmet efter återkommande modeller. För att möta denna utmaning presenterar vi en avslappnad variant av IP som kallas Δ-Integral Privacy (Δ-IP), där två modeller anses vara likvärdiga om deras skillnad ligger inom en viss Δ-tröskel. Vi lyfter också fram utmaningen med att jämföra två DNN:er, särskilt när liknande lager i olika nätverk kan innehålla neuroner som är permutationer eller kombinationer av varandra. Detta gör IP-konceptet mer komplext, eftersom det inte är helt enkelt att identifiera ekvivalenter mellan sådana modeller. Dessutom presenterar vi en metod, tillsammans med dess teoretiska analys, för att generera en uppsättning integrerat privata DL-modeller. I praktiken kommer data ofta in snabbt och i stora volymer, och dess statistiska egenskaper kan förändras över tiden. Att upptäcka och anpassa sig till sådana avvikelser är avgörande för att upprätthålla modellens tillförlitliga förutsägelser över tid. Många metoder

som används för att upptäcka avvikelser bygger på att man skaffar äkta etiketter, vilket ofta är omöjligt. Samtidigt utsätter detta modellen för integritetsrisker, vilket kräver att driftdetektering utförs med hjälp av integritetsskyddande modeller. Vi presenterar en metod som upptäcker avvikelser baserat på osäkerhet i förutsägelser från en ensemble av integrerat privata MLP:er. Detta tillvägagångssätt kan upptäcka avvikelser även utan tillgång till äkta etiketter, även om den förutsätter att de är tillgängliga på begäran. Dessutom tar avhandlingen också upp problemet med medlemsinferens i federerat lärande för datorseendemodeller. Federerat lärande (Federated Learning, FL) introducerades som ett integritetsskyddande paradigm inom DL, där användare samarbetar för att träna en gemensam modell utan att dela sina data. Nya studier har dock visat att de delade vikterna i FL-modeller kodar de data de tränas på, vilket leder till potentiella integritetsintrång. Som en lösning på detta problem presenterar vi en ny metod för integralt privat aggregering för federerat lärande tillsammans med dess konvergensanalys.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Vicenç Torra, for his unwavering support and guidance throughout my studies. His insightful comments and engaging discussions have been invaluable over the past two years. I also appreciate his patience when I was struggling with writing the initial drafts. I am thankful to my co-supervisor, Lili Jiang, and my reference person, Monowar Bhuyan, for their supportive and helpful roles in navigating the PhD responsibilities over the years. I am also thankful to my former supervisor, Pranab K. Muhuri, for his consistent support and genuine encouragement. Additionally, I am thankful to WASP and Umeå University for funding my PhD studies. I also extend my gratitude to the HR, Administration, and IT staff at the university, who have always been readily available to assist with any queries.

Umeå has been a place of great learning, where I have been fortunate enough to meet some wonderful people and create lasting memories. I am grateful to my friends and colleagues with whom I have shared lunches, fikas, and much laughter. My heartfelt gratitude goes to the members of my research group, NAUSICA, a great bunch who have always provided support, advice, or have cheered me up. I am thankful to the badminton and squash groups at IKSU. I am also grateful to other friends in Umeå and Börlänge, for their humorous companionship, support, and inspiration throughout. And I am forever indebted to *meri pyari pinku* for her love, and unwavering belief in me, even in moments when I doubted myself.

I am thankful to my parents and brothers for their unconditional support and encouragement throughout my studies. Last but not the least, I am thankful to god *Shiva* for giving me the strength to move forward each day!

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

List of papers

This thesis is based on the following papers.

Paper I Ayush K. Varshney, Vicenç Torra. Integrally Private Model Selection for Deep Neural Networks. In *Proceedings of the 35th Database and Expert Systems Applications Conference*, LNCS Springer, 2023.

Paper II Ayush K. Varshney, Vicenç Torra. Recurrence Analysis of Integrally Private Support Vector Machine. Submitted, 2024.

Paper III Ayush K. Varshney, Vicenç Torra. Concept Drift Detection using Ensemble of Integrally Private Models. In *Proceedings of the 5th Workshop on Machine Learning for Cybersecurity co-located with ECML PAKDD*, Springer, 2023.

Paper IV Ayush K. Varshney, Vicenç Torra. k-IPfedAvg: k-Anonymous Integrally Private Federated Averaging with Convergence Guarantee. Submitted, 2024.

The author contributed to the following research papers during their PhD, but these are not included in the thesis.

Paper V Ayush K. Varshney, Vicenç Torra. Literature Review of the Recent Trends and Applications in Various Fuzzy Rule-Based Systems. In *International Journal of Fuzzy Systems*, Springer, 2022.

Paper VI Ayush K. Varshney, Vicenç Torra. Efficient Privacy-preserving Federated Unlearning under Plausible Deniability. Submitted, 2024.

Paper VII Ayush K. Varshney, Vicenç Torra. Designing distributed Chi-fuzzy rule based classification system. In *IEEE International Conference on Fuzzy Systems*, IEEE, 2022.

Paper VIII Ayush K. Varshney, Pranab K. Muhuri, Q.M. Danish Lohani. PIFHC: The Probabilistic Intuitionistic Fuzzy Hierarchical Clustering Algorithm. In *Applied Soft Computing*, Elsevier, 2022.

Paper IX Ayush K. Varshney, Pranab K. Muhuri, Q.M. Danish Lohani. Density-based IFCM along with its interval valued and probabilistic extensions, and a review of intuitionistic fuzzy clustering methods. In *Artificial Intelligence Review*, Springer, 2023.

Paper X Priyanka Mehra, Ayush K. Varshney[1]. HFedRF: Horizontal Federated Random Forest. In *International Congress and Workshop on Industrial AI and eMaintenance*, Springer, 2023.

Paper XI Saloni Kwatra, Ayush K. Varshney[1], Vicenç Torra. Integrally private model selection for Support Vector Machine. In *Proceedings of 18th International Workshop on Data Privacy Management*, Springer, 2023.

Paper XII Debanjan Chakraborty, Ayush K. Varshney, Pranab K. Muhuri, Q.M. Danish Lohani. Modified Probabilistic Intuitionistic Fuzzy c-Means Clustering Algorithm: MPIFCM. In *IEEE International Conference on Fuzzy Systems*, IEEE, 2022.

Paper XIII Debanjan Chakraborty, Ayush K. Varshney, Pranab K. Muhuri, Q.M. Danish Lohani. P-IT2IFCM: Probabilistic Interval Type-2 Intuitionistic Fuzzy c-Means Clustering Algorithm. In *IEEE International Conference on Fuzzy Systems*, IEEE, 2022.

[1] Corresponding author

Contributions

In all papers presented in this thesis, the work of drafting the initial research outline, conducting background research, producing the core parts of the theoretical and empirical results, implementing engineering artifacts, and writing the final paper were mainly conducted by the first author. The co-authoring supervisor was continuously involved in all aspects of the work, providing feedback, corrections, and detailed training. The supervisor also offered guidance on methods and strategies at various stages of the research.

Contents

Chapter 1
Introduction

In today's world, the presence of Artificial Intelligence (AI) in our daily lives is constantly growing, driven by the availability of vast amount of data, enhanced computing power, and increased community interest. Such growth in AI has contributed to a multitude of disciplines ranging from Healthcare [1], Finance [2], and many more. However, the rise in data usage by AI systems has heightened privacy concerns. Regulations like the General Data Protection Regulation (GDPR) [3] and the California Consumer Privacy Act (CCPA) [4] have been implemented globally to govern data usage. These regulations enforce strict guidelines on data collection and processing, ensuring that analysis made should be privacy-preserving.

In the literature, several privacy models have been proposed such as k-Anonymity [5], differential privacy [6] and integral privacy [7], etc. to protect individual/organization(s) from adversaries who aim to gain sensitive information. k-Anonymity and its variants offers data masking i.e. while storing data in a database, the server aggregates the data so that for each record there are $k-1$ other indistinguishable records. This is usually implemented with clustering, where k similar records are replaced with their mean or a generalized representation. k-Anonymity, while useful, has many drawbacks, such as homogeneity attacks [8], background knowledge attacks [8], etc. Consider an example, if all the records in k-anonymous group share the same sensitive information, the k-anonymity can lead to privacy breach. On the other hand, the widely accepted differential privacy model and its variants perturb the data or the model to generate privacy-preserving output(s). Differential privacy (DP) is achieved when the probability of a query producing a similar output on neighboring datasets (datasets that differ by only one element) is almost the same. The similarity in outputs helps DP protect individual data points from being inferred. It ensures that the presence or absence of a single individual's data in the dataset does not significantly affect the outcome of the query. In DP, the parameter ϵ quantifies the degree of privacy protection. It sets a bound on how much the probability of any output can differ between two neighboring datasets. A smaller value of ϵ provides stronger privacy guarantees, indicating a greater similarity in the outputs for neighboring datasets. Theoretically, DP offers sound privacy guarantees but it has its own practical limitations. For instance, when the value of ϵ is small (indicating high privacy), the level of

perturbation needed can be substantial. This means that queries with high sensitivity require a significant amount of noise to maintain privacy. Additionally, in scenarios involving multiple queries, the limited privacy budget may necessitate adding substantial noise to each query. However, this high level of noise can lead to a decrease in the utility or accuracy of machine learning models.

In this thesis, we explore integral privacy (IP) as an alternative privacy model to k-anonymity and DP for Deep Neural Networks (DNNs) which does not cost much utility while generating privacy-preserving models. Integrally private models recur from multiple disjoint datasets i.e. a model can be mapped to a set of disjoint datasets. This creates ambiguity to the intruder who is looking to infer if a record or a set of records were part of the training or not i.e. perform membership inference attack and model comparison attack [9]. The authors in [9] approximate the model space to find the recurring models for small datasets. But in case of DNNs, the models are usually very large i.e. the number of parameters in Deep Learning (DL) models can vary from thousands to billions and hence approximating the model space can be computationally challenging. It has been shown in [10] that under higher batch size and similar training environment multiple mini batches can results in similar parameter updates with probability close to one. This analysis suggests that, under specific training environment models can probably recur without generating the complete (or approximate) model space.

Most privacy approaches are designed for static environments; however, real-world data often exists in a streaming format, meaning it arrives continuously. Such streaming data can experience concept drift over time, which refer to changes in its statistical properties. It is crucial for models to detect and adapt to these changes to ensure reliable predictions. The preferred methodologies for addressing concept drift include Adaptive Windowing (ADWIN) [11] or Kolmogorov-Smirnov Windowing (KSWIN) [12]. They detect drifts based on either false positive/negative rates which requires true labels or with the fluctuations in the output probabilities over time. Having access to true labels in real-time is unrealistic in most real-world assumptions.

In the context of Deep Neural Networks (DNNs), training requires a substantial amount of data, and acquiring ground truth for drift detection can be costly. A recent uncertainty drift detection scheme [13] identifies drift during inference without needing true labels. It calculates prediction uncertainty using dropout in DNNs and employs the entropy of these uncertainty values to detect drifts. Alternatively, prediction uncertainty can be obtained through an ensemble of DNN models. Different DNNs yield varying probabilities during predictions, and the collective uncertainty of these predictions can be used for drift detection. The Streaming Ensemble Algorithm [14] was one of the first to use an ensemble of models for this purpose. However, almost none of the approaches in the literature of concept drift focuses on the privacy aspect of drift detection. Acquiring true labels to detect drifts can leak private information in real-time. The output probabilities can further be used for membership

inference attacks [15], which necessitate the need for detecting drifts privately.

Furthermore, this thesis also explore the increasing concerns about privacy in data collection, where Federated Learning (FL) has emerged as a promising approach. FL enables the training of a shared model across multiple users without necessitating the sharing of their raw data. McMahan et al. [16] introduced federated averaging (fedAvg) which is the first and perhaps the most widely used algorithm to aggregate the models trained on user data. FedAvg performs several communication rounds among users with heterogeneous data, and in each communication round, it aggregates the model weight collected from each user. However, FL has its own privacy challenges as the weights shared by the user encodes their private information. Several attacks such as model inversion attacks [17], membership inference attacks [18], and many others, can lead to costly privacy leakage. Hence, ensuring user privacy is critical to enhance the impact and applicability of FL in everyday life.

To further explore the field of privacy-preserving machine learning (ML), this thesis seeks to explore the following research questions:

1. Given the large number of weights and biases in a deep learning model, do we have multiple generators for such models to satisfy integral privacy? (Papers I)

2. What is the probability of obtaining integrally private models for ML and DL? (Paper II, Paper III)

3. Can a set of integrally private models detect drifts in Online Learning? (Paper III)

4. How can we generate integrally private models in heterogeneous setting under collaborative learning, and do they converge? (Paper IV)

This thesis sets out on generating privacy-preserving, computationally efficient and high utility deep learning models and their applications. In Paper-I, we introduce a relaxed notion of Integral privacy which we now call Δ-Integral Privacy (Δ-IP) as a defense against model comparison attack [19]. This allows DNNs which are at most Δ distant apart to be considered for integrally private models, thereby reducing the extensive need to explore the model space. To generate Δ-IP models, we train DNNs on multiple training sets and we find that under a similar training environment, a very high number of models recurs under the definition of Δ-IP. Based on the analysis of [10], we first explore with what probability a typical machine learning model like Support Vector Machine (SVM) can recur after training from a set of disjoint datasets in Paper-II. We further prove that with high probability the DNN models recur in Paper-III. The algorithm for Δ-IP returns an ensemble of integrally private models which we use to detect the concept drift in Paper-III. Paper-III presents a methodology, where we first compute the prediction uncertainty on the streaming data from the models in integrally private ensemble. The prediction uncertainty is

further used by drift detection approaches like ADWIN, to detect drifts. Furthermore, we propose an integrally private federated aggregation mechanism in Paper-IV to avoid inference attacks. Paper-I, Paper-II and Paper-III propose methodologies to generate integrally private solutions in homogeneous settings, while Paper-IV proposes methodologies to generate integrally private models in heterogeneous setting along with its convergence guarantee.

Chapter 2
Preliminaries

In this chapter, we lay the groundwork for the thesis by providing essential background information. We first introduce the major privacy models in the literature and then the attacks for which our privacy model has been proposed. This is accompanied by a brief description of the support vector machine. We then describe the concept drifts and their types and how uncertainty in the DNNs can be used to predict the drifts. The chapter concludes with a concise overview of the federated learning framework.

2.1 Privacy Models

Privacy models are crucial in ensuring the confidentiality of personal sensitive information such as health records, finances, sexual orientation, etc., during the training and inference phases of a machine learning model.

2.1.1 k-Anonymity

k-Anonymity [5] is a well-known privacy model that seeks to protect individual identity. It ensures that each individual's information is indistinguishable from $k-1$ other individuals in the same dataset. Indistinguishability is with respect to quasi identifiers and not necessarily all attributes. Quasi identifiers are those attributes which in combination can uniquely identify an individual.

Definition 1 (k-Anonymity)
With respect to the set of quasi identifiers $\mathcal{Q}$, A database $\mathcal{D}$ satisfies k-Anonymity if the projection of $\mathcal{D}$ on $\mathcal{Q}$ results into partition of $\mathcal{D}$ in sets of atleast k indistinguishable individuals.

k-Anonymity offers sound defence against identity disclosure but may struggle against attribute disclosure. Consider an example of a diabetes database having four records [(Umeå, 30, Diabetic-1), (Umeå, 30, Diabetic-1), (Borlänge, 43, Diabetic-2), (Borlänge, 43, Diabetic-2)]. This dataset is 2-Anonymous, but allows intruders to infer that an individual in Umeå of age 30 is Diabetic type-1. Additionally, if an intruder is aware that an individual's data is present in this dataset, they can infer that the individual is diabetic. Several variants

such as l-diversity[8], p-sensitive k-Anonymity [20] were introduced in order to overcome these drawbacks.

2.1.2 Differential Privacy

Since the inception of differential privacy (DP) [6], it has been widely accepted as the go to privacy model in the industry as well as in the research community. A function f_r is differentially private if the presence or absence of a record does not influence the outcome of f_r on a query r significantly. That is, for neighbouring datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ which differ by at most one record, $f_r(\mathcal{D}_1)$ and $f_r(\mathcal{D}_2)$ are similar.

Definition 2 (ϵ-differential privacy)
For two neighbouring datasets $\mathcal{D}_1, \mathcal{D}_2$, the function f_r for a query r is considered ϵ-differentially private if and only if

$$Pr[f_r(\mathcal{D}_1) \in S] \leq e^{\epsilon} Pr[f_r(\mathcal{D}_2) \in S] \tag{2.1}$$

where $S \subseteq Range(f_r)$ and ϵ is the privacy budget.

Here, it is clear that that lower ϵ means higher privacy. Moreover, when $\epsilon = 0$, it implies an ideal scenario where there is absolutely no privacy leakage. This means that the output of the data analysis or query would be the same, irrespective of whether any individual's data is included in or excluded from the dataset. However, achieving $\epsilon = 0$ in practical applications is virtually impossible and impractical. The primary reason for this is the trade-off between privacy and utility. DP with $\epsilon = 0$ would add so much noise to the data or query output that it would render the resulting information useless for any meaningful analysis. A relaxed version of DP, (ϵ, δ)-differential privacy [21] introduces δ in the eq. (2.1) i.e. $Pr[f_r(\mathcal{D}_1) \in S] \leq e^{\epsilon} Pr[f_r(\mathcal{D}_2) \in S] + \delta$. Another popular variant called local differential privacy [22] protects the data at the record level, and is particularly useful in distributed settings.

2.1.3 Integral Privacy

Integral Privacy [7] is a privacy model which focuses on avoiding inference by choosing a model which can recur from multiple different databases. The key component in integral privacy is the concept of generators of a model. In the terminology of functions, let f be a function and y be a possible outcome of f, then the generators of y consistent with some background information S^*, would be the set of databases which leads to y i.e., the set of $f^{-1}(y)$. Higher the cardinality of the set of generators, higher the privacy. Formally,

Definition 3 (Integral privacy)
For a population P, and model $G \in \mathcal{G}$ generated by algorithm A. Let $Gen^*(G, S^*)$ be the set of generators consistent with the background knowledge S^*. Then, the model G is k-anonymous integrally private if $Gen^*(G, S^*)$

has at least k elements and

$$\bigcap_{S \in Gen^*(G,S^*)} S = \emptyset$$

Here, the null intersection is required to avoid any inter-sectional analysis.

2.2 Privacy Attacks

Privacy attacks in the context of machine learning refer to methods by which an intruder attempts to compromise the privacy of individuals and organizations. Privacy attacks often aim to leak sensitive, confidential, and personal information. There exists several types of these attacks such as model inversion attack, re-identification attacks and many more. The focus of this thesis is on the integral privacy model, which specifically addresses the following two types of attacks.

1. **Membership Inference Attack:** Membership inference attack aims to determine whether a particular sample was part of the training set used to train a machine learning model. With membership inference attack, an intruder can expose sensitive information such as an individual's medical conditions in healthcare-related ML applications, reveal financial status, and enable targeted advertising, among other privacy concerns.

2. **Model Comparison Attack:** In a model comparison attack, it is assumed that the intruder has access to the model and some background information (say S^*) about the dataset. As the name suggests, the intruder compares the model generated by the random subsamples drawn from S^* with the available model. Through this comparison, the intruder finds the generators of the model. Consequently, the intruder may gain access to the underlying training data or could conduct intersectional analysis between subsamples to facilitate a membership inference attack.

2.3 Machine Learning

Machine learning (ML) is a subfield of artificial intelligence (AI) that has garnered significant attention and revolutionized various domains in recent years. It encompasses a diverse set of techniques and algorithms that enable computer systems to learn from data, identify patterns, and make predictions or decisions without being explicitly programmed. ML algorithms tend to improve their performance with the availability of the data. Broadly, ML can be categorized into supervised and unsupervised learning. In supervised ML, the training entity has access to the labeled dataset while in unsupervised learning the dataset is unlabelled. We focus on supervised leaning in this thesis, where the data set $D_{n \times d}$ has n points x_i each of dimension d, and the output

$y_i \in \{1, ..., c\}$ is the label of the input x_i. The usual training in supervised learning is done to tune parameters w of the ML model M.

2.3.1 Support Vector Machine

Within the realm of supervised learning, Support Vector Machine or SVM, stands out as a powerful and widely-used approach. SVM is particularly effective in binary classification tasks, where it aims to find the optimal hyperplane that best separates data points belonging to different classes. For a given dataset $D_{n\times d}$, the hyperplane for optimal support vector machine is defined by $J(w, b)$: $w^T x + b = 0$, where $w \in \mathcal{R}^n$ represents the normal vector of this hyperplane, and $b \in \mathcal{R}$ denotes the bias term. For binary classification, $y_i \in \{-1, 1\}$ represents the negative or positive class label of the i^{th} sample. The optimal hyperplane is computed using the following optimization problem.

$$J(w, b) : \min_{w\in\mathbb{R}^n} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \max(0, 1 - y_i(w^T x_i + b))$$

subject to,

$$y_i(w_T x_i + b) \geq 1 - \max(0, 1 - y_i(w^T x_i + b)), \quad \text{for } i = 1, 2, ..., d \tag{2.2}$$

The objective of minimizing the L_2 norm regularization in the first term is to maximize the margin, ensuring broader separation between datapoints of different classes. While the second term penalizes the objective function for samples that are incorrectly classified. The parameter C regulates the trade-off between the maximizing the margin and minimizing the number of incorrectly classified samples. The initial set of constraints guarantees that the projections of data points are separated by at least one unit. In cases where this condition cannot be met, the optimization process adjusts to form a soft-margin hyperplane by minimizing the error variable. This soft-margin approach allows for the classification of data points that fall closer to the decision boundary, accommodating some level of misclassification.

2.4 Deep Neural Networks

Deep neural networks (DNNs) are a class of machine learning models designed to emulate human brain learning processes. The structure of these networks consists of perceptrons, commonly referred to as neurons, which intake input arrays and transform them into output signals. DNNs learns from data through a list of layered structures, with each layer learning a specific relationships or functionalities within the input data. Each layer is essentially a group of neurons tasked with discerning patterns within the input.

When there is only a single hidden layer, the network is called artificial neural networks (ANNs). DNNs extend the concept of ANNs with two or

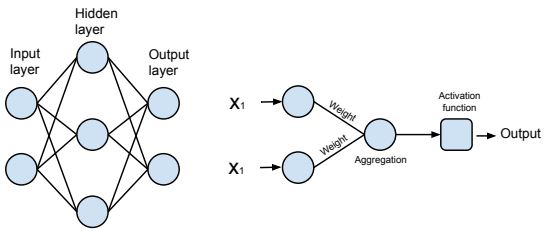


Figure 2.1: A typical ANN and a typical neuron

more hidden layers. A neuron in each layer computes a weighted sum of the input with a bias term which is transformed using an activation function. The transformed output is then communicated to the next layer of the DNN (as shown in fig. 2.1).

The last layer of DNNs is the output layer, which, in classification tasks, often employs a softmax function. This function returns a probability distribution across various classes, aiding in determining the most likely class for a given input. Often times, these class probabilities are misinterpreted as models confidence. In many cases, a model can given high class probabilities for unseen data even when it is uncertain in its predictions.

Measuring the uncertainty of DNNs in predicting outputs is crucial for detecting concept drift in streaming data, where data arrives continuously over time. Concept drifts are the changes in the statistical propertied of data with time. They must be detected early in order to minimize the utility loss. Many of the approaches [23] required true labels to detect drifts. But in real-world applications, the data may come with high volume and velocity, which may require manual labeling, expertise to label samples accurately and constant monitoring. Hence, in such cases acquiring true labels can be time consuming and costly. To overcome this drawback, in [13], proposed an uncertainty based drift detection scheme using DNNs. The authors [13] quantifies the uncertainty in prediction from an ensemble of DNNs and the uncertainty obtained is used to detect concept drift. However, with a family of DNNs, the inference attacks can be easily done with their range of output probabilities. Hence, privacy remains a big concern.

2.5 Federated Learning

Federated Learning has established itself as a significant paradigm in the field of distributed machine learning. In Federated Learning (FL), multiple users collaborate with a central server to train a global model iteratively without sharing their raw data. The central server hosts the global model, which it then distributes to users. The global model is communicated to users, users in turn train the model on their local data for few epochs and communicate the

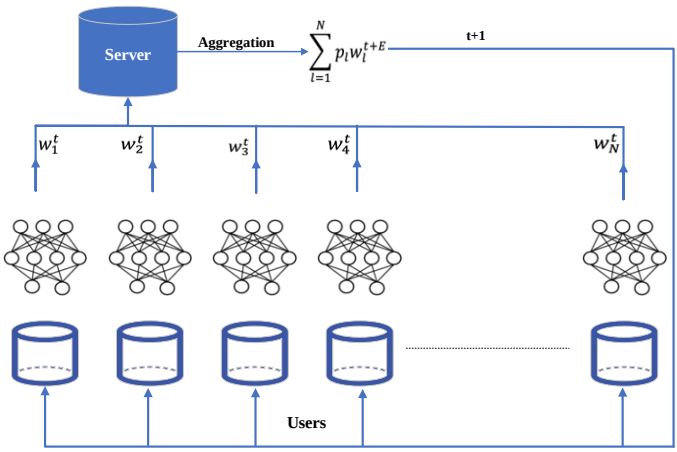


Figure 2.2: Federated averaging framework.

updated model to the server. The server aggregates the weights from the user. This aggregation process, known as federated averaging (fedAvg), continues until the model converges or reaches the desired number of communication rounds. Figure 2.2 shows the framework when all the devices participates. In partial participation, few users are randomly selected in each communication round. In FL, users can follow iid (independent and identically distribution) as well as non-iid. distribution. In literature, fedAvg has been shown to converge in both settings with $\mathcal{O}(\frac{1}{T})$ rate, where T is the total number of training rounds.

2.6 Mean samplers

In machine learning, the objective is to train a model M with parameters w, b (weights and biases) using a subset X of the dataset $\mathcal{D}$. The goal is for the model to accurately classify instances into one of c classes. Training involves iteratively processing n independently and identically distributed batches of data. Each batch contains pairs (x_i, y_i) , where x_i is a data instance and y_i its corresponding class label. During each training epoch, the model's parameters w are adjusted to minimize the loss function $\mathcal{L}$, which measures classification error. This is achieved by applying an update rule g, resulting in the parameter update $w_{t+1} \leftarrow g(w_t, X)$. A key approach in this process involves mean samplers, which divide the training data X into n distinct batches $\hat{x}_i$, each containing b data instances. The update rule in this context is represented as $w_{t+1} \leftarrow g(w_t, \hat{x}_i) = \frac{1}{b} \sum_{i=1}^{b} g(w_t, (x_i, y_i))$.

An example of a mean sampler is the minibatch stochastic gradient descent (SGD), which updates parameters by minimizing the average loss across a

minibatch. The update rule in minibatch SGD is expressed as:

$$w_{t+1} = g(w_t, \hat{x}_k) = w_t - \eta \frac{1}{b} \sum_{i=1}^{b} \nabla_w \mathcal{L}(M_w(\hat{x}_i, y_i))|_{w_t} \tag{2.3}$$

Here, η is the learning rate, and $\frac{1}{b} \sum_{i=1}^{b} \nabla_w$ represents the average gradient with respect to w_t for the minibatch $\hat{x}_i$. A set of models is considered similar if they learn comparable average gradients from their respective minibatches. This similarity is used to estimate the likelihood of model recurrence.

Chapter 3
Contribution of the Thesis

This chapter presents an overview of the contributions made in this thesis, summarizing the included research papers, and probabilistic analysis for the recurrence of SVM model. In the following sections, each paper's summary is provided, followed by its contributions and limitations, which are listed in bullet points.

3.1 Paper I

The key intuition of the thesis is based on the assumption that the machine learning models recur i.e. there exists several models in the complete model space which can be generated by various different datasets [9]. In paper I, the focus is on finding such recurrent DNN models which costs minimal utility loss. DNNs generally are applicable for big data, have huge number of parameters, and hence huge model space. In order to avoid the computational cost of exploring the model space, we introduce the relaxed variant of integral privacy called ϵ-integral privacy (which we now onwards call Δ-integral privacy) in paper I. Δ-IP considers two DNNs recurring if they are at most Δ distant apart from each other. This enables us to identify the recurrent models without exploring the entire model space. Additionally, we require a set of integrally private models to enable us to choose a model that incurs minimal utility loss. Therefore, we propose a methodology that returns a set of integrally private models from a given database.

Contributions

The paper presents the following contributions:

- A formalization of computationally efficient relaxed variant of integral privacy called Δ-integral privacy.
- A methodology to return a set of Δ-integrally private models with their statistics.
- An experimental analysis of four different DNN architectures on datasets with varied sizes and data types to validate the performance of our

methodology to achieve Δ-IP.

Limitations

The following limitations of the paper's contributions are worth highlighting:

- Generation of Δ-IP models is probabilistic in nature, the paper does not provide any formal proof that the proposed approach works.
- In the presence of outliers, generation of Δ-IP models can be difficult as they disturb the original distribution.
- The approach may struggle to generate a set of Δ-IP models for small datasets.

3.2 Paper II

It has been established in the literature that a minibatch used in the optimization algorithms employing mean samplers (see section 2.6) can be forged with probability close to one [10, 24]. The Δ-IP requires that the models should recur after complete training. Establishing the recurrence of models with high probability will offer insights into the privacy gurantees of Δ-IP. This paper presents the recurrence analysis of a machine learning model (SVM in our case) which uses mean sampling optimizers (e.g., SGD, Adam) recurs after complete training.

Contributions

The paper presents the following contributions:

- Establishes that a machine learning model like SVM recurs with high probability.
- Highlights that the probability of recurrence of two models can be bounded by a distance measure, number of training sets and the number of datapoints.

Limitations

The following limitations of the paper's contributions are worth highlighting:

- Many algorithms, such as k-nearest neighbours and decision trees, do not use mean samplers. Hence, the probabilistic analysis does not hold for such models.

3.3 Paper III

In this paper, we focus on generating integrally private models in the streaming context. In such settings, data arrives continuously. The models must be

capable of detecting concept drifts while maintaining privacy. Additionally, drift detection schemes that require true labels for incoming data may be impractical. Therefore, it is necessary to detect drifts in real time and maintain privacy, even when labels are not available. Similar to [13], our method detects drifts from the uncertainty in prediction from an ensemble of integrally private models. Once the drift is detected, true labels are requested and the models are retrained. Paper-III also provides theoretical analysis on the recurrence of DNNs that for a given set of disjoint datasets.

Contributions

The paper presents the following contributions:

- A concept drift detection methodology called 'Integrally Private Drift Detection' which is one of first approaches in literature to detect drift privately.
- Privacy-preserving methodology that does not require the true labels to detect drifts, but assumes they are available or can be requested once the drift has been detected.
- Theoretical analysis on the generation of Δ-integrally private DNNs, removing one of the drawbacks of paper-I.
- An experimental analysis of the methodology that shows benchmark comparable utility (tested on multiple scores) and benchmark comparable drift detection in the absence of true labels.

Limitations

The following limitations of the paper's contributions are worth highlighting:

- The generation of Δ-IP models require training on large datasets, with computationaly expensive comparison between models. Hence, we have running time as the cost of privacy instead of utility.

3.4 Paper IV

This paper extends the methodology for the generation of integrally private models in federated learning environments for Convolutional Neural Networks (CNNs) in order to protect the identity disclosure of the clients participating. In each communication round of fedAvg, the central server clusters the models which are at most Δ distant apart. The server aggregates randomly chosen model updates from each cluster to generate the global model for the next communication round. This protects the identity of the participating clients. The paper also presents the convergence analysis of the proposed methodology, we find that just like fedAvg [25] our methodology also has the convergence rate of $\mathcal{O}(\frac{1}{T})$, T is the total number of training epochs.

Contributions

The paper presents the following contributions:

- Novel privacy preserving federated averaging algorithm, called 'k-Anonymous Integrally Private Federated Averaging' (k-IPfedAvg), to protect the identity disclosure of participating clients.
- Theoretical analysis prove that under similar training environment, k-IPfedAvg also has the convergence rate of $\mathcal{O}(\frac{1}{T})$.
- An experimental analysis using three distance measures and various privacy parameters to demonstrate the minimal impact of privacy parameter k in k-IPfedAvg on utility. The analysis also shows that k-IPfedAvg outperforms its differentially private counterpart across multiple levels of privacy.

Limitations

The following limitations of the paper's contributions are worth highlighting:

- The randomly chosen models in each round may lead to some accuracy drops in k-IPfedAvg.
- The degree of non-iidness can affect the convergence and the privacy i.e., if all the user's model updates are very different then higher Δ for Δ-IP must be chosen which may lead to poor privacy and convergence. Further experimentation is required.

Chapter 4
Future Work

The papers presented in this thesis focus on generating models which can be generated by a set of different datasets. This approach creates ambiguity for potential intruders, preventing them from mapping the machine learning model to a specific dataset, even when they have background knowledge. Paper-I lays the foundation by introducing a relaxed variation of integral privacy (Δ-integral privacy), paper-II provides the probabilistic analysis for the recurrence of Δ-IP models along with a methodology for concept drift detection, and paper-III extends the idea of Δ-IP for federated averaging along with its convergence analysis.

Mitigating challenges. Although generating Δ-IP models is computationally less expensive compared to integrally private models, it is still significantly costly. Computationally efficient methods for model comparison can lead to reduced cost. The Δ-IP models have been generated for relatively small architectures e.g. architecture with 3-7 layers, further experiments on complex network architectures such ResNet-50 can give interesting results. At the same time, the models generated are probabilistic in nature, hence reproducibility posses a big question. Furthermore, the quality of data can significantly affect the performance of the generated model, exploring its impact on privacy presents an interesting future direction. On the similar lines, for federated learning the impact of non-iidness for Δ-IP models presents an interesting future direction.

Machine unlearning. Machine unlearning aims to remove the influence of data point(s) required by the regulations such as GDPR. A naive approach to machine unlearning is retraining from scratch, and as expected will be very computationally expensive. Machine unlearning becomes even more challenging for distributed ML, such as federated learning, where the global model is distributed across the users in each communication round. Integral privacy provides an efficient and plausibly deniable solution for federated unlearning, as discussed in [26] on unlearning. Future research could explore its application in generative AI for large models.

References

[1] A. Panesar, *Machine learning and AI for healthcare*. Springer, 2019.

[2] L. Cao, "AI in finance: challenges, techniques, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–38, 2022.

[3] P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[4] E. Goldman, "An introduction to the california consumer privacy act (CCPA)," *Santa Clara Univ. Legal Studies Research Paper*, 2020.

[5] P. Samarati, "Protecting respondents identities in microdata release," *IEEE transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[6] C. Dwork, "Differential privacy," in *Automata, Languages and Programming* (M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds.), (Berlin, Heidelberg), pp. 1–12, Springer Berlin Heidelberg, 2006.

[7] V. Torra, G. Navarro-Arribas, and E. Galván, "Explaining recurrent machine learning models: integral privacy revisited," in *International Conference on Privacy in Statistical Databases*, pp. 62–73, Springer, 2020.

[8] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, pp. 3–es, 2007.

[9] N. Senavirathne and V. Torra, "Integrally private model selection for decision trees," *computers & security*, vol. 83, pp. 167–181, 2019.

[10] A. Thudi, H. Jia, I. Shumailov, and N. Papernot, "On the necessity of auditable algorithmic definitions for machine unlearning," in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 4007–4022, 2022.

[11] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM international conference on data mining*, pp. 443–448, SIAM, 2007.

[12] C. Raab, M. Heusinger, and F.-M. Schleif, "Reactive soft prototype computing for concept drift streams," *Neurocomputing*, vol. 416, pp. 340–351, 2020.

[13] L. Baier, T. Schlör, J. Schöffer, and N. Kühl, "Detecting concept drift with neural network model uncertainty," *arXiv preprint arXiv:2107.01873*, 2021.

[14] W. N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 377–382, 2001.

[15] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang, "When machine unlearning jeopardizes privacy," in *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*, pp. 896–911, 2021.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[17] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7232–7241, 2021.

[18] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753, IEEE, 2019.

[19] A. K. Varshney and V. Torra, "Integrally private model selection for deep neural networks," in *Database and Expert Systems Applications* (C. Strauss, T. Amagasa, G. Kotsis, A. M. Tjoa, and I. Khalil, eds.), (Cham), pp. 408–422, Springer Nature Switzerland, 2023.

[20] T. M. Truta and B. Vinay, "Privacy protection: p-sensitive k-anonymity property," in *22nd International Conference on Data Engineering Workshops (ICDEW'06)*, pp. 94–94, IEEE, 2006.

[21] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "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, 2006.

[22] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 127–135, 2015.

[23] R. S. M. de Barros and S. G. T. de Carvalho Santos, "An overview and comprehensive comparison of ensembles for concept drift," *Information Fusion*, vol. 52, pp. 213–244, 2019.

[24] Z. Kong, A. Roy Chowdhury, and K. Chaudhuri, "Forgeability and membership inference attacks," in *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*, pp. 25–31, 2022.

[25] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[26] A. K. Varshney and V. Torra, "Efficient privacy-preserving federated unlearning under plausible deniability." unpublished, 2024.

Paper I

Integrally Private Model Selection for Deep Neural Networks

Ayush K. Varshney( ) and Vicenç Torra

Department of Computing Sciences, Umeå University, 90740 Umeå, Sweden
{ayushkv,vtorra}@cs.umu.se

Abstract. Deep neural networks (DNNs) are one of the most widely used machine learning algorithms. In the literature, most of the privacy related work to DNNs focus on adding perturbations to avoid attacks in the output which can lead to significant utility loss. Large number of weights and biases in DNNs can result in a unique model for each set of training data. In this case, an adversary can perform model comparison attacks which lead to the disclosure of the training data. In our work, we first introduce the model comparison attack for DNNs which accounts for the permutation of nodes in a layer. To overcome this, we introduce a relaxed notion of integral privacy called ϵ-integral privacy. We further provide a methodology for recommending ϵ-Integrally private models. We use a data-centric approach to generate subsamples which have the same class-distribution as the original data. We have experimented with 6 datasets of varied sizes (10k to 7 million instances) and our experimental results show that our recommended private models achieve benchmark comparable utility. We also achieve benchmark comparable test accuracy for 4 different DNN architectures. The results from our methodology show superiority under comparison with three different levels of differential privacy.

Keywords: Data privacy · Integral privacy · Deep neural networks · Privacy-preserving ML

1 Introduction

In today's world, Artificial Intelligence (AI) plays a crucial role in our day-to-day life. AI techniques are widely used in object recognition, speech recognition, medical imaging, robotics and many other fields. AI approaches and Machine Learning (ML) in particular are very data hungry [1]. They tend to improve with the quality and quantity of data. The data often include sensitive and personal information which must be guarded to ensure security/privacy of each individual or organization. Several guidelines exists such as Europe's General Data Protection Regulation (GDPR), to regulate the use of data in ML. GDPR requires

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

C. Strauss et al. (Eds.): DEXA 2023, LNCS 14147, pp. 408–422, 2023.
https://doi.org/10.1007/978-3-031-39821-6_33

that the analysis to be made should use the minimum amount of data and must be privacy-preserving. There are several data masking and privacy-preserving models such as k-anonymity [2], differential privacy [3], integral privacy [4], etc. which try to protect privacy of individuals and organizations from any adversaries. Adversaries aim to gain sensitive information about individuals or a group of individuals making inferences from ML models.

Data masking is used to modify sensitive information so that a record can not be uniquely identified. K-anonymity is one of the most used data masking methods. A database satisfies k-anonymity if for each record there are k-1 other indistinguishable records. This can be implemented using clustering (replacing k similar records with their mean or with their generalization). In the recent years, much attention has been given to differential privacy (DP) and its variants (see [5] for more details). Differential privacy is satisfied if the outputs of a query on neighbouring datasets are similar i.e. addition or removal of one record should not affect the outcome of the query. Differential privacy depends on a parameter ϵ that establishes the level of this similarity. Theoretically, DP offers sound privacy-preserving models but it has practical limitations such as the amount of noise for small ϵ (high privacy) can be very high. Therefore, high sensitivity queries require high amount of noise. However, in case of multiple queries as the privacy budget is limited, high amount of noise is also required. High noise leads to a loss of utility for ML models. In our approach, we have considered Integral Privacy as an alternative to DP to achieve high utility privacy-preserving machine learning.

Integral Privacy models [4] are the data-driven models that appear recurrently with different training data sets. This makes inferences on sensitive information harder for an intruder. Formally, the set of integrally private models are the set of recurrent models, i.e. generated by different datasets for the same problem. This approach has practical limitations, as in general, we rarely have a huge number of different datasets. The first practical approach for Integral private model selection was given for decision trees [6], where instead of having an available set of datasets, the authors have used sampling approaches to build the model space and eventually suggesting models which are integrally private. The authors expanded the idea with integral privacy guarantees for linear regression. This is given in [7]. In [8], authors have shown how maximal c-consensus meets (see [9] for further details) can be used in the context of integral privacy to find datasets which can produce the same models. The work presented in [6] generates or approximates the model space for a given dataset. A stratified subsampling approach is used to approximate the model space for small datasets ($\approx$ 200 instances). The authors approximate the model space using 100k, 150k and 300k subsamples from each datasets. This can be time consuming and 100–300k subsamples may not be enough to approximate the model space for real-world big datasets. Overall, the approach is computationally expensive.

Deep Neural Networks is one of the most successful machine learning paradigms for several computer vision tasks such as image classification [10], object detection [11], video classification [12], and many other areas. However, DNNs are known to be highly dependent on the input data. In the last few years,

interest in adversarial DNN examples has grown [13]. DNNs are assumed to work well with large datasets. They have large number of weights and biases which can result in very few generators (unique in many of the cases) for each model. In other words, generation or discovery of recurrent models in DNNs is difficult.

Considering these challenges in mind, we introduce a relaxed variant of integral privacy called 'ϵ-Integral Privacy' where models in the ϵ range are considered perturbated version of each other and, thus, they are considered ϵ-integrally private. We also propose a model selection strategy for choosing ϵ-integrally private models for Deep Neural Networks (DNNs). Our algorithm recommends the mean of the top recurrent models as the private model. We distribute the data in disjoint subsamples having same class-distribution as the original dataset. We find that large enough disjoint subsets having same class-distribution as the original dataset leads to the generation of the models which are utmost ϵ-different, with utility comparable to the benchmark model. This way we do not need to generate 100–300k sub samples. Our approach also supports the data-centric approach [14]. We are able to generate benchmark comparable models with samples sizes 1/100th of the original dataset. There hasn't been much work in the literature which discusses about using smaller datasets for training DNNs. The work in [15] improves the quality of data by eliminating the invalid instances, our approach is focused on maintaining the class-distribution of the data.

In this paper, we have also extended the potential model comparison attack [6] for DNNs. In this type of attack, an intruder gets access to the training data by comparing the models learned by the intruder obtained from original data and the model obtained from a modified dataset. In case of DNNs, the attack becomes tricky as any permutation of the similar set of nodes at any given layer l results in the same learning. We incorporate this to extend the model comparison attack on DNNs.

We have arbitrarily chosen a 3-hidden layered DNN for 6 datasets with varied sizes. Our experimental results show that large enough disjoint sets lead to the generation of ϵ-integral private models with benchmark comparable utility and loss. We get benchmark metrics by training and testing on our chosen DNN on 70-30 split for each data. We have also compared ϵ-integral private models with high DP (differential privacy) model, moderate DP model and low DP model; we found integrally private models have better utility in many cases and have significant improvement in terms of loss for most of the datasets.

This paper is organized as follows. In Sect. 2 we introduce the model comparison attack for DNNs; In Sect. 3 we introduce the notion of ϵ-integral privacy and present the algorithm for private model selection procedure for DNNs; In Sect. 4 we present the experimental analysis to support our claim and in Sect. 5 we present our conclusion and directions for future work.

2 Model Comparison Attack for DNNs

In this section, we describe our model comparison attack for deep neural networks. Deep neural networks are machine learning models which were created to learn like the human brain. The underlying architecture of DNNs consists of the

perceptron (or commonly known as neuron) which receives an array of inputs and transform them into output signal(s). DNNs learns from data by putting together a list of layers. Each layer is responsible for learning some relationship or functionality in the input. Each layer is a collection of neurons that learns to detect patterns in the input. Each neuron in the DNNs can be considered as a logistic regression. DNNs are the extension of artificial neural networks with two or more hidden layers. In each neuron, the weighted sum of the input with a bias term is computed which is then transformed using an activation function, which is then passed on to the next layer of the DNNs. Nodes at layer l receive input from the nodes at layer $l-1$, which means each neuron has $|l-1|+1$ ($+1$ for bias) number of parameters to be tuned in training. Final weights and biases of each neuron highly depends on their initialization.

2.1 Framework

In this section, we propose our framework. Let X be the training set from the original dataset D, $\mathcal{G}$ be the model generated on X. In our work, we have considered DNNs as learning algorithm. Let us denote an initial architecture and weight by $Arch$ and let A be the algorithm.

We assume the intruder has some background knowledge $S^* \subseteq D$. They are the records that are known to be used to train the model. The intruder also has access to the model. That to $\mathcal{G}$ which was learned from the training set X on the initial architecture $Arch$. That is, $\mathcal{G} = DNN(Arch, X)$. With this information, the intruder aims to gain knowledge on the training set and do membership inference attacks

The intruder essentially can perform the model comparison attack once they can generate the model space associated to S^*. The intruder can perform comparison with the models in model space and his knowledge of G. After comparison, if there is a single generator for the model, the intruder gets complete access to the training set and their inferences. If there are more than one generator for the model, an intruder can do membership inference attack for dominant records by finding the intersection between the generators.

2.2 Intruders Approach

The intruder has some background information S^*. Then, they can draw a block of subsamples $S = \{S_1, S_2, ..., S_n\}$ where $S_i \subseteq S^*$ to generate the (approximated) model space. Each subsample is a set of instances from S^* which are used to generate a DNN (see Fig. 1). Generation of the complete model space can be computationally expensive but can be approximated using sampling approaches.

Comparison of two DNNs for model comparison attack is a difficult task because we need to deal with a combinatorial problem. We need to align neurons in each layer. Observe that layers in both DNNs must contain the same neurons i.e. for two DNNs to be the same they must have equal layers; and for two layers to be equal, neurons in one layer must be some permutation of the neurons in the other layer. Given r neurons, we will have r! possible permutations.

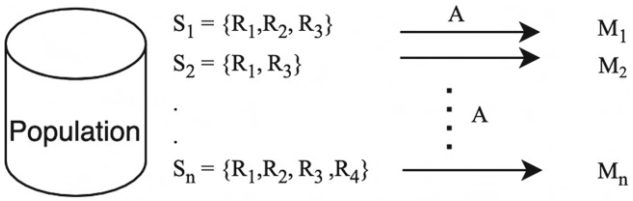


Fig. 1. Demonstration of model generation using algorithm A for subsamples $S_1, S_2, ..., S_n$.

Each model in the generated model space can be compared with the original model G. In case of DNNs, each model has one or very few generators due to the high number of parameters of the model. Therefore, after the comparison attack, the intruder may be able to uniquely identify the training set used to generate the model. When there are more than one generator for a model G, an intruder can check for membership inference by finding the dominant records from the intersection of the generators for the model.

2.3 Integral Privacy

This privacy model [4] aims to protect the disclosure of training data and inferences from a model comparison attack. Let A be an algorithm to compute model G from a given population of samples P. The model G is integrally private if it can be generated by enough number of samples from the population. Let S^* be the background information available to the intruder, then $Gen^*(G, S^*) = \{S' \setminus S^* \mid S^* \subseteq S' \subseteq P, A(S') = G\}$ is the possible set of generators for the model G. K-anonymous integral privacy holds when there are at least k disjoint generators in the set $Gen^*(G, S^*)$. Disjoint generators are required to avoid membership inference attacks. Formal definition for Integral privacy is as follows**.**

Integral Privacy. Let P be the set of samples or a dataset. For model $G \in \mathcal{G}$ generated by algorithm A on samples $S \subseteq P$, let $Gen^*(G, S^*)$ represent the set of all generators of G which are consistent with the background knowledge S^*. Then, the model G is said to be k-anonymous integrally private if $Gen^*(G, S^*)$ contains at least k sets of generators and

$$\bigcap_{S \in Gen^*(G,S^*)} S = \emptyset \tag{1}$$

3 ϵ-Integrally Private Model Selection for DNNs

To construct the complete model space is computationally intractable for large sets. Consider an example of a dataset with 5000 instances. Considering all possible datasets to produce all possible models of the model space (say M_c)

corresponds to producing 2^{5000} generators and the corresponding models. The alternative to M_c is to construct an approximation of the model space (M_e) using sampling. This approach was used in previous works [6,7]. Nevertheless, even in this case the number of generators and their corresponding models can be high and computationally expensive. In case of bigger datasets say with 5 million instances, the process of building an approximation of a model space will be very costly. In our approach, we have focused on reducing the huge computational requirement to recommend relaxed integrally private deep neural network models.

Let us consider the problem of finding the set of different models of the model space. First, let us recall that each neuron at layer l in DNNs receive inputs from all the neurons in layer $l-1$, which in turn require weights and bias for the neuron. The weights and biases in DNNs can take any value between -1 and $+1$. Even for a small DNN there can be a unique generator for each model or only very few models will have more than one generator. Our initial studies on DNNs confirms this even when we round-off weights to 3 digits. It is worth mentioning here that initialization of DNNs also affects the number of generators. More concretely, we may not get the same generators on differently initialized models. This makes achieving integral privacy difficult.

Because of this in our approach, we have adopted the relaxed version of integral privacy which we call 'ϵ-Integral privacy' in which models utmost ϵ different from each other are considered. In case of DNNs, two models are utmost ϵ different if and only if the difference between weights for the same connections between neurons is always less than ϵ I.e. if $G1$, $G2$ represent the weights for two DNNs then $||G_1 - G_2|| \leq \epsilon$, where $||G_1 - G_2||$ represent the difference between every same connection between neurons for both DNNs. Now, let $Gen^*(G, S^*, \epsilon)$ denote the set of possible pairwise disjoint generators for the models which are utmost ϵ different than G (generators that are consistent with the background knowledge S^*), then k-anonymous ϵ-Integral privacy holds if $Gen^*(G, S^*, \epsilon)$ has at least k elements and their intersection is empty. A more formal definition follows.

ϵ-Integral Privacy: Let P be the set of samples or datasets. For a model $G \in \mathcal{G}$ generated by algorithm A on samples $S \subseteq P$, let $Gen^*(G, S^*, \epsilon)$ represent the set of all generators of G which are consistent with the background knowledge S^* and are utmost ϵ different. Then, the model G is said to be k-anonymous ϵ-Integrally private if $Gen^*(G, S^*, \epsilon)$ contains at least k elements and

$$\bigcap_{S \in Gen^*(G, S^*, \epsilon)} S = \emptyset \tag{2}$$

Now, we will focus on the private model selection procedure for DNNs. Our approach to generate subsampling is data centric. We choose subsamples of size N with same class-distribution as the original dataset D. We denote these subsamples by $S_1, S_2, ..., S_n$ (here $n = \lfloor |D|/N \rfloor$). Here, we also satisfy there is no intersection between subsamples i.e. $S_1 \cap S_2 \cap ... \cap S_n = \emptyset$. This condition is

Algorithm 1. Integrally private model selection procedure for Deep Neural Networks for a given perturbed dataset D'. The algorithm returns top 5 integrally private models with their accuracies

Inputs: D - Perturbed Dataset
N - Size of subsamples
ϵ - Privacy parameter
A - Algorithm to generate DNNs
Output: returns a list of integrally private models with their accuracies
Algorithm:

```
S = Generate_subsample(D, N)          ▷ Generate n subsamples of size N
ModelList = [[]]
for S_i in S do
    M_i ← A(S_i)
    present = False
    for each m_j ∈ ModelList do
        if compare_model(m_j, M_i) ≤ ε then
            ModelList[j].append(M_i)
            present = True
            break
        end if
    end for

    if present == False then
        ModelList.append(list(M_i))
    end if
end for
chosen_models = choseXModels(ModelList)
                                      ▷ Chose top X recurring models
meanModels = A(mean(chosen_models))   ▷ Compute mean models
statistics = computeMetrics(meanModels)   ▷ Statistics of mean models
return meanModels, statistics
```

important to avoid membership inference attack from the intersection analysis between generators.

Now, we propose our algorithm for choosing integrally private models for DNNs. Its flowchart is given in Fig. 2. The algorithm is as follows for a given dataset D. First, we generate n subsamples each of size N having the same class-distribution as the original. Second, we compute models and cluster them so that each cluster has models that are utmost ϵ different from each other. Finally, we can choose a cluster of models which are recurring in nature and has high utility. In our methodology, we chose the mean of all the models in the cluster as our recommended model. I.e. we generate a new model whose weights are the mean of the weights of all the ϵ-integrally private models.

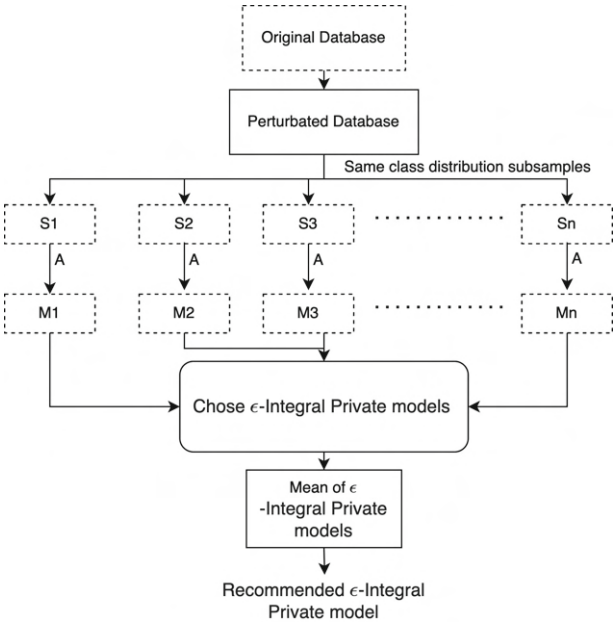


Fig. 2. Flowchart of the proposed methodology to recommend an ϵ-integral private model.

Algorithm 1 formalizes this approach. In the algorithm we have a dataset D, Algorithm A, privacy parameter ϵ and size of each subsample N as inputs. We initialize an empty list of lists and append models which are utmost ϵ distant apart from the first one. For our results we can either chose the top recurring model or X most frequent models (for more ambiguity) which is done in function choseXModels(). Our recommended model is the mean of the models in the cluster. For X $\epsilon-$ranged models, we recommend X mean models and their statistics as the output of our proposed algorithm.

4 Experimental Results

In this section, we present our experimental results for our proposed methodology. Our approach is valid for both numerical/categorical data and for classification problems with an arbitrary number of classes. Table 1 shows the details of the datasets we have considered for our experiments namely Adult, Susy, ai4i and HepMass from UCI repository [16]; and Churn_Modelling, Diabetes [17]. Of these datasets, Churn Modelling and Adult have categorical data and Diabetes is a multi-class problem. We have considered small datasets (≈10–50K instances), medium dataset (≈250K instances) and large datasets (≈5–7 million instances) for our experimental study. Table 1 also shows the size of the subsamples. The size is chosen so that there are enough subsamples to find integrally private models.

Table 1. Details of the used datasets

Dataset	# instances	# attribute	Data type	# classes	subsample size
Adult	48842	14	Categorical Integer	2	1000
Susy	5000000	18	Real	2	10000
ai4i	10000	14	Real	2	500
HepMass	7000000	28	Real	2	10000
Churn Modelling	10000	21	Categorical Real	2	500
Diabetes	254000	21	Real	3	5000

To compare the performance of our approach and 2 benchmark, we have used an architecture of 5-layered DNN with 3-hidden layers with 5-10-5 neurons. As we explain later, we have considered other architectures as well. Then, we have taken $\epsilon = 0.05$ for all the datasets, other values could be used depending on the application requirements.

The results of our methodology have been compared with results with a differential private solution [18] and the benchmark results. Benchmark results are obtained by training the model with 70-30 train-test split of original dataset. Now, let us look at the number of generated models from randomly chosen subsamples of the size given in Table 1. In case of the adult dataset, the total possible models which can be considered for integral privacy are 47, similarly for ai4i dataset we have 19, for susy dataset we have 498, for hepmass dataset we have 698, for churn modelling dataset we have 18, and for diabetes dataset we have 49 models to be considered for integral privacy.

Figure 3 shows the training f1 score of top 5 (for ai4i and Churn Modelling datasets there are 2 and 3 generators only) recurring models along with the training score of the benchmark model in black solid line and three level of differential privacy(DP): high privacy ($\epsilon \approx 0.1$, represented by ◆), moderate privacy ($\epsilon \approx 0.5$, represented by $\cdot-$) and low privacy ($\epsilon \approx 1.0$, represented by ●). In general, higher DP privacy (low ϵ, ◆) leads to lower training score and higher training loss. In the plots, the f1 scores of all the models are in the light shade, and the dark solid line represents the mean of the ϵ ranged integral private models. Observe from Fig. 3a and 3b, we achieve better training score than the benchmark training scores while from Fig. 3c, 3d, 3e and 3f we can observe benchmark comparable results. It can be seen from Fig. 3a, 3c and 3d, integrally private models have better training score than all three variants of differentially private models on the other hand Fig. 3b, 3e and 3f, the training utility of integrally private model is comparable with the differentially private models. We get similar results for the training loss as shown in Fig. 4. We have denoted the loss of each model in the lighter shade solid line, their mean loss in dark solid line, the benchmark model loss with solid black line and three level of differential privacy: high privacy with ◆, moderate privacy with $\cdot-$ and

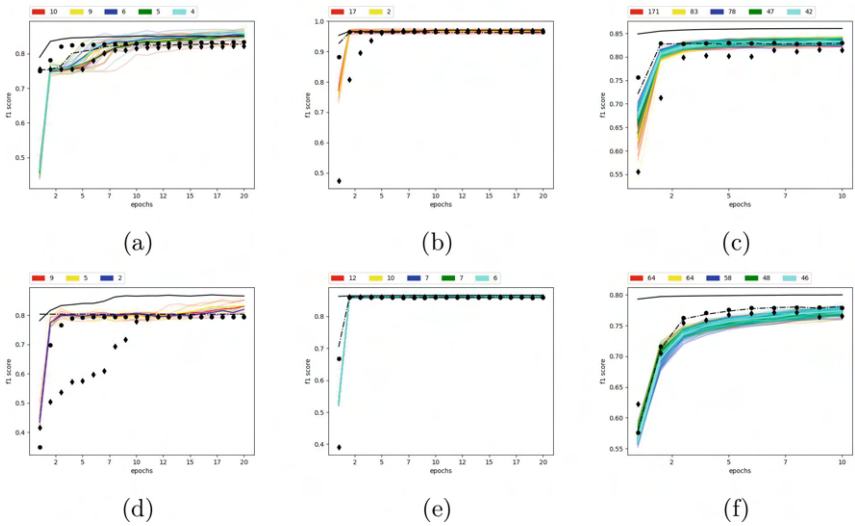


Fig. 3. f1 score of top 5 ϵ-recurring models over training data for (a) Adult (b) ai4i (c) HepMass (d) Churn Modelling (e) Diabetes (f) Susy Datasets

low privacy with •. It can be seen that the loss for integrally private models is comparable with the benchmark model loss. We can observe from Fig. 4b, 4c and 4d, integrally private models have significant improvement in terms of training loss from DP variants while Fig 4a, 4e shows some improvement from DP variants in contrast to Fig. 4f where low, moderate DP privacy has improvement in training loss from integrally private models.

The concept of data-centric AI simply suggests that good quality of data can lead to good models. In our approach, we have only used 0.15% to 2% of the original data, but with the same class-distribution, to train our model (see Table 1 for subsample size). We got surprising result when we compared their performance on test data i.e. 30% of the original data. Figure 5 shows the result on the test data, lighter shade circles represent the test result for each model while dark solid colored circle represents their mean value. From Fig. 5, we can say that our ϵ-integrally private models achieve benchmark comparable f1 score on much bigger test datasets (15 to 200 times).

Our recommended model is the mean of all the models in the ϵ-integral private range. The result in Fig 5 motivated us to compare performance of the aggregated ϵ-integrally private models with the original training and testing datasets. Figure 6 shows the comparison of f1 score on training data (in solid color circles) and test data (in hollow circles) with benchmark training score (in solid line) and benchmark test score (in dashed line). Our recommended models have benchmark comparable f1 score on all the datasets.

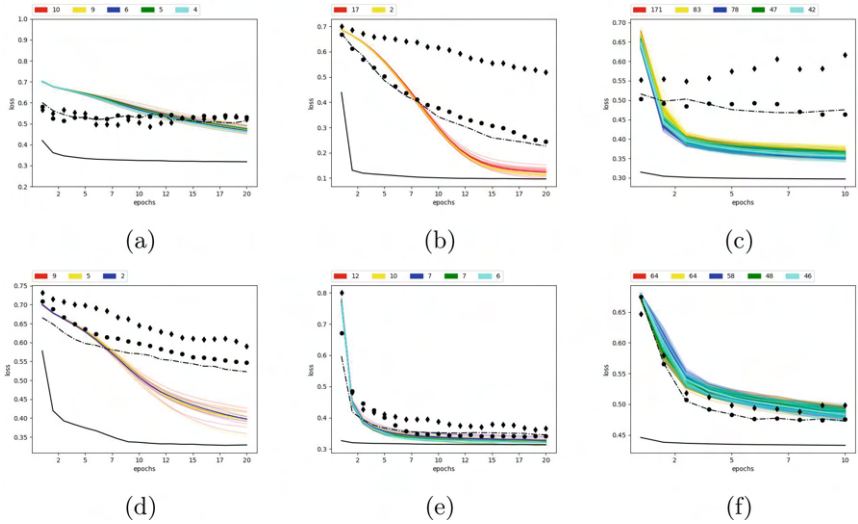


Fig. 4. Training loss of top 5 ϵ-recurring models for (a) Adult (b) ai4i (c) HepMass (d) Churn Modelling (e) Diabetes (f) Susy Datasets

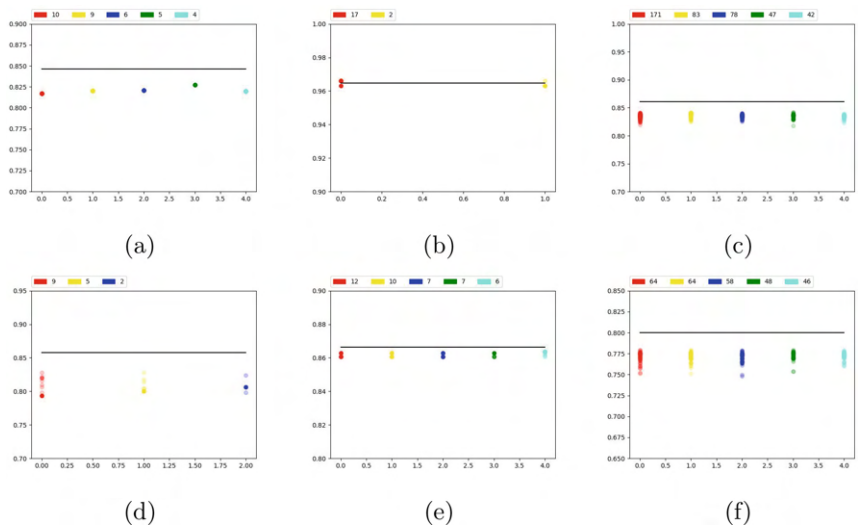


Fig. 5. f1 score of top 5 ϵ-recurring models on bigger test data for (a) Adult (b) ai4i (c) HepMass (d) Churn Modelling (e) Diabetes (f) Susy

Table 2 shows the recurrence of the recommended model with the test accuracy on much bigger test sets. We have considered 4 different architectures: DNN-1 has 3-hidden layers (with 5-10-5 neurons respectively) architecture; DNN-2 has 1- hidden layer (with 1024 neurons) architecture; DNN-3 has 3-hidden layers (with 10-20-10 neurons respective) architecture; and DNN-4 has 5-hidden

Table 2. Different architectures and their f1 score on 30% test dataset.

Dataset	DNN-1		DNN-2		DNN-3		DNN-4	
	recurrence	test_acc	recurrence	test_acc	recurrence	test_acc	recurrence	test_acc
Adult	10	0.8387	89	0.7797	16	0.8286	36	0.8284
Susy	64	0.7758	366	0.7917	8	0.7636	6	0.7882
ai4i	17	0.9647	19	0.9723	12	0.9683	10	0.9747
HepMass	171	0.8325	562	0.8344	68	0.8325	51	0.8336
Churn Modelling	9	0.8145	13	0.8520	10	0.7927	10	0.7870
Diabetes	12	0.8627	21	0.8596	13	0.8634	5	0.8596

layers (with 5-10-20-10-5 neurons respectively) architecture. Table 2 shows that the proposed methodology produces benchmark comparable results for different DNN architectures as well.

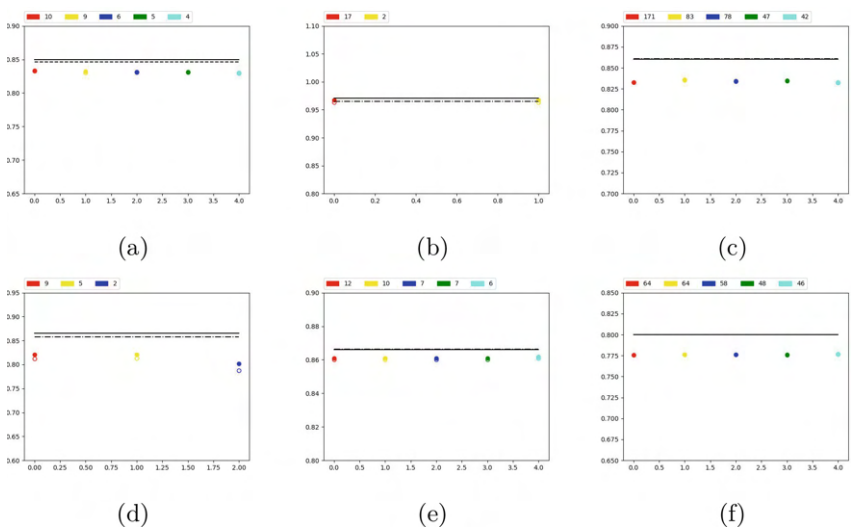


Fig. 6. f1 score on train and test data for mean of the ϵ-recurring models for (a) Adult (b) ai4i (c) HepMass (d) Churn Modelling (e) Diabetes (f) Susy

4.1 Discussion

In summary, our results with varied sized, multi-class and categorical datasets suggest that we can achieve ϵ-integral privacy with good utility (comparable to benchmark utility) from the list of the recommended models depending on the value of k (number of models in ϵ range) with no additional computational cost.

The good results of our approach can essentially be linked to the data centric AI approach where we train our model for smaller datasets with the same class-distribution as the original dataset and get good results. We further explored the impact of subsample size and compared their performance on separate 70-30

training data and testing data on moderately sized adult and diabetes datasets. Our results from Fig. 7 shows that the f1 score for both training and testing data is non-decreasing but it is neither increasing significantly with respect to the increase in subsample size. Our results are in line with [19] which highlights that one can generate arbitrarily similar model of finite floating point weights from two (or more) non-overlapping dataset. The paper [19] also suggest that we can get good results on smaller datasets as well, which aligns with the results in Fig. 7.

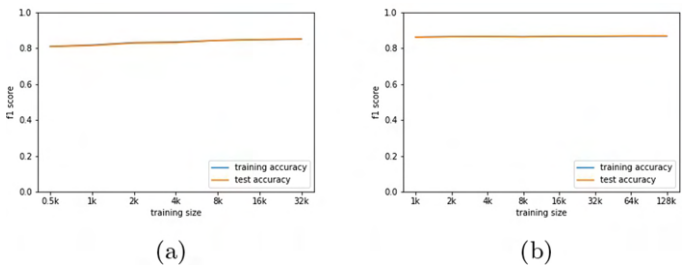


Fig. 7. f1 score of various subsample sizes on (a) Adult (b) Diabetes datasets

For our proposed methodology, we must chose subsamples size (N) very carefully. The choice for N must be large enough to generate the model with good utility at the same time it should be able to generate sufficient number of disjoint subsamples. Probably approximately correct (PAC) [20] can suggest an estimate for the choice of the parameter N. A model G is said to be PAC learnable with respect to loss l if and only if the difference between the loss for the learned model G and true (best possible) model $\bar{G}$ is at most ϵ with probability at least $1 - \delta$ i.e. $P[G_l - \bar{G}_l \leq \epsilon] \geq 1 - \delta$. With this the minimum number of samples required for a PAC learnable model is bounded by $O([VC(G) + ln(1/\delta)]/\epsilon^2)$ [21] where $VC(G)$ is the Vapnik-Chervonenkis dimension of the model G. Quantifying the VC-dimension for complex models like deep neural network is still an open problem [22]. Therefore, in the literature scientists follow the rule-of-thumbs: (1) The VC dimension of DNNs is considered equal to the number of weights in DNNs [23] and then (2) the minimum number of samples required to learn the DNN is established as 10 times the VC dimension [24]. Considering this, i.e., a sample size of 10-times the VC-dimension (number of weights) should provide a PAC learnable model. For datasets ai4i, and Churn_Modeling the number of weights are 172 and 197, respectively, and hence the minimum subsample size is estimated as 1720 and 1970 for PAC learnability. This results in very few disjoint subsamples (5 for both datasets) which may not be enough to find integrally private models. This suggests a trade-off between model complexity (number of weights) and its learning ability for integral privacy. Further study in this area is required to investigate the impact of this trade-off for integral privacy.

4.2 Limitations

Based on a critical analysis of our approach and the results obtained, we can underline the following limitations of our approach:

1. Our methodolgy may not be suitable in the presence of outliers as the outliers disturbs the distribution of the dataset.
2. Selection of private models on very small datasets with our proposed methodology is not feasible.
3. High model complexity may result in less number of models in ϵ-range.

5 Conclusion and Future Work

In this paper, we have first extended the model comparison attack to deep neural networks. We have also introduced the concept of ϵ-integral privacy which is then used to recommend integrally private models for deep neural networks. Our results show that we are able to achieve ϵ-integrally private models without any significant utility loss (improvement of utility in some cases). Our results also highlights that small data of good quality can result in a well trained model.

For our proposed methodology, we have arbitrarily chosen the size of the subsamples; the privacy parameter ϵ and the DNNs architecture. Tuning of these areas may yield interesting results. Another interesting direction is to use a data-enhancement approach to remove outliers as done in [15]. Federated Learning takes advantage of data distributed across multiple users, where learning takes place locally. Our methodology can be seen as independent and identically distributed (IID) ϵ-integral private model selection in federated learning for a single pass. Our work can further be extended into non-IID settings of federated learning.

References

1. Obermeyer, Z., Emanuel, E.J.: Predicting the future-big data, machine learning, and clinical medicine. New Engl. J. Med. **375**(13), 1216 (2016)
2. Samarati, P.: Protecting respondents identities in microdata release. IEEE Trans. Knowl. Data Eng. **13**(6), 1010–1027 (2001)
3. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
4. Torra, V., Navarro-Arribas, G.: Integral privacy. In: Foresti, S., Persiano, G. (eds.) CANS 2016. LNCS, vol. 10052, pp. 661–669. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48965-0_44
5. Ji, Z., Lipton, Z.C., Elkan, C.: Differential privacy and machine learning: a survey and review. arXiv preprint arXiv:1412.7584 (2014)
6. Senavirathne, N., Torra, V.: Integrally private model selection for decision trees. Comput. Secur. **83**, 167–181 (2019)

7. Senavirathne, N., Torra, V.: Approximating robust linear regression with an integral privacy guarantee. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST), pp. 1–10. IEEE (2018)
8. Torra, V., Navarro-Arribas, G., Galván, E.: Explaining recurrent machine learning models: integral privacy revisited. In: Domingo-Ferrer, J., Muralidhar, K. (eds.) PSD 2020. LNCS, vol. 12276, pp. 62–73. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57521-2_5
9. Torra, V., Senavirathne, N.: Maximal C consensus meets. Inf. Fusion **51**, 58–66 (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778 (2016)
11. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
12. Jiang, Y.-G., Zuxuan, W., Wang, J., Xue, X., Chang, S.-F.: Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **40**(2), 352–364 (2017)
13. Oh, C., Xompero, A., Cavallaro, A.: Visual adversarial attacks and defenses. In: Advanced Methods and Deep Learning in Computer Vision, pp. 511–543. Elsevier (2022)
14. Ng, A.: MLOps: from model-centric to data-centric AI (2021). https://www.deeplearning.ai/wp-content/uploads/2021/06/MLOps-From-Model-centric-to-Data-centricAI.pdf. Accessed 09 Sept 2021
15. Motamedi, M., Sakharnykh, N., Kaldewey, T.: A data-centric approach for training deep neural networks with less data. arXiv preprint arXiv:2110.03613 (2021)
16. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
17. Centers for Disease Control, Prevention, et al.: National diabetes statistics report, 2017. Centers for disease control and prevention, Atlanta, GA (2015, 2017)
18. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
19. Thudi, A., Jia, H., Shumailov, I., Papernot, N.: On the necessity of auditable algorithmic definitions for machine unlearning. In: 31st USENIX Security Symposium (USENIX Security 2022), pp. 4007–4022 (2022)
20. Vapnik, V.N., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. In: Vovk, V., Papadopoulos, H., Gammerman, A. (eds.) Measures of Complexity, pp. 11–30. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21852-6_3
21. Vershynin, R.: High-Dimensional Probability: An Introduction with Applications in Data Science, vol. 47. Cambridge University Press, Cambridge (2018)
22. Anthony, M., Bartlett, P.: Neural Network Learning: Theoretical Foundations. Cambridge University Press, Cambridge (1999)
23. Abu-Mostafa, Y.S.: Hints. Neural Comput. **7**(4), 639–671 (1995)
24. Baum, E., Haussler, D.: What size net gives valid generalization? In: Advances in Neural Information Processing Systems, vol. 1 (1988)

Paper II

Recurrence Analysis of Integrally Private Support Vector Machine

Ayush K. Varshney and Vicenç Torra

Department of Computing Science
Umeå University 90740, Sweden
{ayushkv, vtorra}@cs.umu.se

Abstract. In the era of big data, Machine Learning (ML) applications thrive on extensive datasets, but this exposes them to privacy risks. Regulatory frameworks like GDPR and CCPA aim to mitigate these risks. Integral privacy, an alternative to k-Anonymity and differential privacy, focuses on creating ambiguity for intruders by considering models generated from diverse datasets as privacy-preserving. Integral privacy calls such models as recurring models. While prior research has primarily explored recurrence in deep learning models which have large parameter space, this paper addresses the understudied recurrence analysis of a typical machine learning model with relatively small parameter space like Support Vector Machine (SVM). Models having small parameter space can have significant impact due to the presence and absence of a datapoint. Due to this reason, their probability to recur maybe low. We challenge this hypothesis with the recurrence analysis of SVM models trained on mean samplers like stochastic gradient descent. We find that under constrained environment SVM models recurs with high probability. This research enhances our understanding of privacy-preserving models in the context of SVMs, providing valuable insights into their privacy guarantees.

Keywords: Machine learning · Support Vector Machine · Recurrence Analysis · Integral privacy.

1 Introduction

In the era of big data, the Machine Learning (ML) applications are becoming increasingly evident in our daily lives. These ML models tend to improve with the vast quantity of data available which is readily available with big data. However, this surge in data usage also escalates the potential for the leakage of private and sensitive information. To mitigate these risks, regulatory frameworks like the General Data Protection Regulation (GDPR) [1] and the California Consumer Privacy Act (CCPA) [2] have been implemented. These regulations are designed to govern data usage and minimize privacy risks in ML models, ensuring responsible and secure handling of data.

Several data masking and privacy-preserving methods, such as k-Anonymity [3], [4] and differential privacy [5], have gained prominence in recent times. k-Anonymity safeguards identities in a database by making each record indistinguishable from at least $k-1$ others. This method is effective in maintaining anonymity within a group, particularly when datasets are shared or published. Achieving k-Anonymity often involves clustering data points so that each cluster contains at least k elements, and then replacing each data point with the centroid of its cluster. However, this approach has vulnerabilities; it can be susceptible to various attacks, including those arising from homogeneity within a cluster [6] or from attackers leveraging background knowledge [6]. While differential privacy is satisfied when the results of a query on neighbouring datasets are similar, meaning the addition or removal of a single record does not significantly affect the query's result. In differential privacy, the parameter ϵ determines the privacy budget, with lower values indicating stronger privacy guarantees but potentially reduced data utility. It quantifies the trade-off between the privacy of individual data points and the utility or accuracy of the data analysis. This allows DP to offer theoretically sound privacy guarantees but it has practical limitations. For instance, in privacy critical applications such as healthcare, finance, the amount of noise required to guarantee privacy is very high and may cost significant amount of utility. Most of the privacy approaches in the literature either focuses on storing data privately or preserving privacy from model's inference, not much focus been given to selecting models which are privacy preserving.

Integral privacy [7] was introduced as an alternative to both k-Anonymity and Differential Privacy (DP). It defines a model as privacy-preserving if it can be generated by multiple disjoint training datasets, introducing ambiguity for intruders attempting membership inference and model comparison attacks. Specifically, the set of integrally private models consists of recurring models that can be generated with multiple datasets. However, obtaining diverse datasets for a given application can be challenging.

In the context of machine learning models, Senavirathne et al. [8] proposed a methodology for identifying integrally private decision trees. Their approach involves generating an approximated model space and determining the frequency of recurring models. While this method is feasible for small datasets, it becomes computationally intensive for larger datasets due to the substantial number of subsamples required to approximate the model space. This challenge becomes even more pronounced in the case of deep learning models, which have a vast number of parameters.

A relaxed variant of integral privacy (called Δ-Integral Privacy, Δ-IP) and methodology to generate integrally privacy deep learning models was given in Varshney et al. [9]. They found that under similar training conditions, subsamples with similar distributions result in similar deep learning models. This finding aligns with the observations made by Thudi et al. [10], who noted that, with high probability, a minibatch can be replicated. Varshney et al. [11] provided a probabilistic guarantee for the recurrence of complete deep learning models. It's worth

noting that most recurrence and forging analyses in the literature have focused on deep learning models, which have a large parameter space.

However, the impact of a single minibatch on a weight in a deep learning model may not be as significant as its impact on a weight in a machine learning model with less number of weights (in comparison with DL models), such as Support Vector Machine (SVM) [12]. Similar to [9], Kwatra et. al [13] gives the methodology to generate recurring SVM models where the authors first randomly selects disjoint subsamples from the training data. Then, trains the SVM models on the subsamples. The trained SVM models are compared to find a set of recurring models. The requirement of disjoint subsamples is required in order to avoid any membership inference analysis. The methodology provides a set of recurring models and their statistics but lacks theoretical guarantees on the recurrence. Hence, in this paper, we present the recurrence analysis for SVM models and show that with high probability, a typical machine learning model like SVM also recurs.

2 Background

2.1 Support Vector Machine

Algorithm 1 Stochastic gradient descent for SVM

Input: S - Training set, N - Size of subsamples, T - Number of epochs
Output: returns weights for
Algorithm:
Initialize $w^0 = 0 \in \mathcal{R}^n$
for epochs i *in* $[1, .., T]$ **do**
  Randomly sample N datapoints from the training set, S ($N \leq |S|$)
  Repeat for each instance (x_i, y_i) of N data points:
  $w^t \leftarrow w^{t-1} - \eta \nabla J^t(w^{t-1})$
end for
return final w

SVM, or Support Vector Machine, is a widely recognized binary classifier. Given a dataset $D_{n \times d}$, SVM identifies the most effective hyperplane from a set of hyperplanes to differentiate between data samples belonging to distinct classes. The optimal SVM hyperplane satisfies $J(w, b)$: $w^T x + b = 0$, where $w \in \mathcal{R}^n$ is the normal vector of $J(w, b)$, $b \in \mathcal{R}$ represents the bias term. The optimal hyperplane is solved using the following optimization problem.

$$J(w, b) : \min_{w \in \mathbb{R}^n} \frac{1}{2} w^T w + C \sum \max(0, 1 - y_i(w^T x_i + b))$$

for N training examples,

$$J(w,b) : \min_{w \in \mathbb{R}^n} \frac{1}{2} w^T w + C.N. \max(0, 1 - y_i(w^T x_i + b)) \tag{1}$$

subject to,

$$y_i(w_T x_i + b) \geq 1 - \max(0, 1 - y_i(w^T x_i + b)), \text{ for } i = 1, 2, ..., d$$

Minimizing the L_2 norm regularization is equivalent to maximizing the margin in the first term. The second term penalizes $J(w, b)$ for the incorrectly classified samples. Here, $y_i \in \{-1, 1\}$ denotes the label of the i^{th} sample and the parameter C regulates the trade-off between the maximizing the margin and minimizing number of incorrectly classified samples. The initial set of constraints ensures that the projections of data points are separated by at least one unit. If this condition can not be satisfied, minimizing the error variable leads to the formation of soft-margin hyperplane. A typical algorithm for learning the weights of SVM is given in Algorithm 1.

2.2 Mean Samplers

SVM are the supervised machine learning models which train on the labeled dataset $\mathcal{D}$. Usually, we want to learn weights and biases (say w) of a model M using training data X ($\subseteq \mathcal{D}$) so that it returns the class label ($\{1, 2, .., c\}$) for any given instance. The training of the model M is done iteratively from the i.i.d. sampled n batches. Each batch consists of a set of records (x_i, y_i). In each epoch, for parameters w in model M the loss $\mathcal{L} : M_w \times y \to [0, \infty)$ computes the classification error for the batch. Iteratively, we aim to minimize the $\mathcal{L}$ and update the parameters using some update rule g i.e. $w_{t+1} \leftarrow g(w_t, X)$). Mean samplers are an important class of update rules, where the training dataset X is divided into n batches i.e. $X = \bigcup_{i=1}^{n} \hat{x}_i$, $\hat{x}_i \cap \hat{x}_j = \phi$ and each batch consists of b data instances $\hat{x}_i = \{x_1, x_2, ..., x_b\}$. Here, the update rule looks like: $w_{t+1} \leftarrow g(w_t, \hat{x}_i) = \frac{1}{b} \sum_{i=1}^{b} g(w_t, (x_i, y_i))$. Mean samplers like minibatch stochastic gradient descent (SGD) samples a minibatch and update the parameters to minimize the average loss for the minibatch. This variant of SGD is known for its applicability in machine learning algorithms. Here, a typical update rule looks like:

$$w_{t+1} = g(w_t, \hat{x}_k) = w_t - \eta \frac{1}{b} \sum_{i=1}^{b} \nabla_w \mathcal{L}(M_w(\hat{x}_i, y_i))|_{w_t} \tag{2}$$

Where η is the learning rate and $\frac{1}{b} \sum_{i=1}^{b} \nabla_w$ is the average gradient with respect to w_t for $\hat{x}_i$ minibatch. A set of models is said to similar if for each they learn the same average gradients from their minibatch. We use this analogy to find the probability of recurrence of models.

3 Recurrence Analysis SVM

In this section, we present the probabilistic analysis for the recurrence of SVM models. As we can see from Algorithm 1, the update rule for the weights in the SVM is:

$$w^t \leftarrow w^{t-1} - \eta \nabla J^t(w^{t-1})$$

from eq. (1)

$$\nabla J^t(w^{t-1}) = \begin{cases} w^{t-1} - C.N.y_i x_i, & \text{if } 1 - y_i(w^T x_i + b)) \geq 0 \\ w^{t-1}, & \text{otherwise} \end{cases}$$

$$\text{if } y_i w^T x_i \leq 1 \text{ than}$$

$$w^t \leftarrow (1-\eta)w^{t-1} - \eta C N y_i x_i \tag{3}$$

$$\text{Otherwise,} \quad w^t \leftarrow (1-\eta)w^{t-1} \tag{4}$$

Now, let us consider a set of data samples, $D_1, D_2, ..., D_m$, i.i.d. (independent and identically distributed) sampled from a given dataset $\mathcal{D}$ with some distribution and $M_1, M_2, ..., M_m$ be the SVM models we want to train. Since, we know from Algorithm 1 that all the models are initialized with $0(\in \mathcal{R}^n)$. From eq. (3) and (4), we can say that for two different models having same learning rate (η), C, and N, the weights in 1st iteration will depend on the product of $y_i x_i$. Similarly, if two models have same weights at some iteration $t-1$ then at iteration t their weights will depend on the product of $y_i x_i$ sampled at iteration t.

Since, the data samples are chosen iid from $\mathcal{D}$, samples from these data samples will also follow the similar distribution. Let the mean of the product $y_i x_i$ for each of the N samples at any iteration be μ and the trace of the covariance matrix be σ^2. Note here that mean value of $y_i x_i$ would still be $\mu(= \frac{1}{N}\sum_{i=1}^{N} \mu)$ but mean sampling of trace of the covariance matrix will be $\frac{1}{N}\sigma^2$. Then by Markov's inequality we can say that,

$$\begin{aligned} P(|\frac{1}{N}\sum y_i x_i - \mu|_2 \geq \Delta) &= P(|\frac{1}{N}\sum y_i x_i - \mu|_2^2 \geq \Delta^2) \\ &\leq \frac{E(|\frac{1}{N}\sum y_i x_i - \mu|_2^2)}{\Delta^2} \end{aligned} \tag{5}$$

Here the first equality is true by the property of monotonicity of squares and second is Markov's inequality. Note that $E(|\frac{1}{N}\sum y_i x_i - \mu|_2^2)$ is just the trace of the covariance matrix ($\frac{1}{N}\sigma^2$), then we can write:

$$P(|\frac{1}{N}\sum y_i x_i - \mu|_2^2 \geq \Delta^2) \leq \frac{\sigma^2}{N\Delta^2} \tag{6}$$

Furthermore, for two models M_j, and M_k, from eq. (6) we can say

$$|\frac{1}{N}\sum y_i^j x_i^j - \mu|_2^2 \geq \Delta^2 \text{ and } \frac{1}{N}\sum y_i^k x_i^k - \mu|_2^2 \geq \Delta^2 \text{at } i^{th} \text{ iteration} \quad (7)$$

Using triangle inequality ($|a-b| \leq |a|+|b|$) we can say $|\frac{1}{N}\sum y_i^j x_i^j - \frac{1}{N}\sum y_i^k x_i^k| \leq 2\Delta^2$ (product of x_i and y_i for both models to be in Δ^2 ball of μ) with probability $\geq (1-\frac{\sigma^2}{N\Delta^2})^2$. Then, the probability that there exists two models which are in the Δ^2 ball of μ would be:

$$P(|y_i x_i - y_l x_l|_2^2 \leq \Delta^2) \geq \sum_{r=2}^{m} \binom{m}{r} \left(1-\frac{\sigma^2}{b\Delta^2}\right)^r \left(\frac{\sigma^2}{b\Delta^2}\right)^{m-r} \quad (8)$$

And, the probability that m models are in the Δ ball of μ would be:

$$P(|y_i x_i - y_l x_l|_2^2 \leq \Delta^2) \geq (1-\frac{\sigma^2}{N\Delta^2})^m \quad (9)$$

Equation (8) and (9) represents the probability of weights updating in the Δ^2 ball of μ for a single iteration. For T iterations, the probability that there exists a model having weight updates in the Δ^2 ball of μ is $\geq (\sum_{r=2}^{m} \binom{m}{r} \left(1-\frac{\sigma^2}{b\Delta^2}\right)^r \left(\frac{\sigma^2}{b\Delta^2}\right)^{m-r})^T$. After T iterations, the probability of m models to be in the Δ^2 ball of μ will be $\geq ((1-\frac{\sigma^2}{N\Delta^2})^m)^T$.

So, for data samples sampled iid from some dataset, we can conclude that the lower bound for the probability that there exists recurrent models within ϵ^2 ball is $\geq (m(1-\frac{\sigma^2}{N\Delta^2})^2)^T$. And all the m models are in the Δ^2 ball of μ is $\geq ((1-\frac{\sigma^2}{m\Delta^2})^m)^T$.

Remark on the choice of Δ, N, m: In order to generate higher recurring models, we can say that increasing the number of i.i.d samples (m), N (Number of data points trained in each epochs) and Δ (the error value) increases the probability of getting recurrent models.

The experimental results verifying the recurrence of SVM for tabular data is given in [13].

4 Discussion and Conclusion

In this paper, we present the recurrence analysis for integrally private Support Vector Machines (SVMs). The integral privacy model selects models which recurs from multiple datasets, introducing uncertainty for potential intruders. Unlike existing privacy models in the literature, such as k-Anonymity and ϵ-Differential Privacy (ϵ-DP), which primarily focus on preserving the privacy of stored data or the privacy during model inferences, integral privacy centers around the selection of private models during training. Unlike k-Anonymity and ϵ-DP, integrally

private models does not cost much utility. The privacy budget in integral privacy depends on the Δ value (see eq. 9). A higher value of Δ means higher difference in the models and reduced privacy.

The analysis presented in this paper emphasizes that when mean samplers like SGD and Adam optimizers are employed during training, models tend to exhibit recurrence with a high probability, regardless of the data type or model size. However, certain machine learning models, such as decision trees and k-nearest neighbors, do not utilize mean samplers in their training process. Further recurrence analyses are needed to confirm whether machine learning models that do not employ mean samplers also exhibit recurrence.

Acknowledgement: This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[1] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[2] E. Goldman, "An introduction to the california consumer privacy act (ccpa)," *Santa Clara Univ. Legal Studies Research Paper*, 2020.

[3] P. Samarati, "Protecting respondents identities in microdata release," *IEEE transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[4] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression," 1998.

[5] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[6] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 3–es, 2007.

[7] V. Torra, G. Navarro-Arribas, and E. Galván, "Explaining recurrent machine learning models: Integral privacy revisited," in *International Conference on Privacy in Statistical Databases*, Springer, 2020, pp. 62–73.

[8] N. Senavirathne and V. Torra, "Integrally private model selection for decision trees," *computers & security*, vol. 83, pp. 167–181, 2019.

[9] A. Varshney and V. Torra, "Integrally private model selection for deep neural networks," *Database and Expert Systems Applications. DEXA 2023*, vol. 14147, 2023.

[10] A. Thudi, H. Jia, I. Shumailov, and N. Papernot, "On the necessity of auditable algorithmic definitions for machine unlearning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 4007–4022.

[11] A. Varshney and V. Torra, "Concept drift detection using ensemble of integrally private models," *European Conference on Machine Learning. ECML 2023*, 2023.
[12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, 1995.
[13] S. Kwarta, A. Varshney, and V. Torra, "Integrally private model selection for support vector machine," *Data Privacy Management, co-located with ESORICS 2023*, 2023.

Paper III

Concept Drift Detection using Ensemble of Integrally Private Models

Ayush K. Varshney[1[0000-0002-8073-6784]] and Vicenç Torra[1[0000-0002-0368-8037]]

Department of Computing Sciences
Umeå University 90740, Sweden
{ayushkv,vtorra}@cs.umu.se

Abstract. Deep neural networks (DNNs) are one of the most widely used machine learning algorithm. DNNs requires the training data to be available beforehand with true labels. This is not feasible for many real-world problems where data arrives in the streaming form and acquisition of true labels are scarce and expensive. In the literature, not much focus has been given to the privacy prospect of the streaming data, where data may change its distribution frequently. These concept drifts must be detected privately in order to avoid any disclosure risk from DNNs. Existing privacy models use concept drift detection schemes such ADWIN, KSWIN to detect the drifts. In this paper, we focus on the notion of integrally private DNNs to detect concept drifts. Integrally private DNNs are the models which recur frequently from different datasets. Based on this, we introduce an ensemble methodology which we call 'Integrally Private Drift Detection' (IPDD) method to detect concept drift from private models. Our IPDD method does not require labels to detect drift but assumes true labels are available once the drift has been detected. We have experimented with binary and multi-class synthetic and real-world data. Our experimental results show that our methodology can privately detect concept drift, has comparable utility (even better in some cases) with ADWIN and outperforms utility from different levels of differentially private models.

Keywords: Data privacy · Integral privacy · Concept Drift · Private drift · Deep neural networks · Streaming data.

1 Introduction

In recent years, the interest in deep learning models has witnessed a steady increase, despite encountering various challenges such as explainability, privacy, and data dependency. To address these issues, significant advancements have been made, including approaches to enhance explainability [1], privacy-preserving techniques [2], adopting a data-centric perspective to facilitate model training with high-quality data, and more. However, limited attention has been given in the context of streaming data, which refers to the continuous arrival of data in real-world scenarios, often accompanied by the problem of concept drift. Concept drift implies that the statistical properties of the data may change over

time, necessitating the model to adapt to these changes to ensure reliable predictions. Noisy data at one point of time may become useful data over time. These changes in data distributions can be due to various hidden factors. Handling of such drifts is a must and has been employed in many applications such as spam detection [3], demand prediction [4]. Learned models must have the ability to detect concept drifts and incorporate them by retraining on the new data. Three types of concept drifts have been shown in Fig. 1. Abrupt drifts are sudden changes in the data distribution. E.g. complete lockdown in many countries due to COVID-19 pandemic. Gradual drifts are the drifts which changes the distribution over time. E.g. in fraud detection system, fraudsters adapt according to the improving security policies in place. Incremental concept drift are the drifts where old concepts vanishes completely with time. E.g. after lifting COVID-19 lockdown, people may be hesitant to return to their normal behaviour.

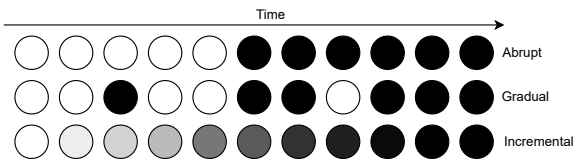


Fig. 1: Types of drifts in the data

In the literature of concept drift detection, there has been several algorithms which can detect concept drifts such as Adaptive windowing (ADWIN) [5] and its variant, and Kolmogorov-Smirnov Windowing (KSWIN). These are the two prominent drift detection methods used in the streaming settings. To detect drifts, these techniques were originally proposed assuming the availability of true labels which is unrealistic in most real-world assumptions. ADWIN employs two windows, one fixed size and one variable size, which slide over the incoming data stream. The fixed size window keeps the most recent points and the variable size window keeps the earlier points. If the statistics of the two windows differs significantly then ADWIN indicates that the drift has been detected.

In case of DNNs, training requires huge amount of data and acquisition of ground truth to detect drift can be very costly. A recently proposed uncertainty drift detection scheme [6] detects drift during inference without the availability of true labels. It computes values for prediction uncertainty using dropout in the DNNs and uses entropy of these uncertainty values to detect drifts. Another approach to get prediction uncertainty is through the ensemble of DNN models. Different DNNs produce different probabilities during predictions and the overall uncertainty in their predictions can be used to detect the drift. In the literature, almost none of the approaches focus on the privacy perspective of drift detection.

Privacy is a crucial factor to take into account in concept drift detection as data is often sensitive. There exists many privacy models such as k-anonymity [7], differential privacy (DP) [8], integral privacy [9] and others for static environment

but their counter-parts for online learning are rather limited. For online learning, k-anonymity [10] tries to protect against identity disclosure by guaranteeing k-anonymity for addition, deletion, and updating the records but may fail to protect attribute disclosure; differential privacy (DP) perturbs the data or the model in order to generate privacy-preserving outputs against the disclosure of sensitive information. Even though DP provides theoretically sound privacy-preserving models, it has a number of practical drawbacks. For instance, when aiming for high privacy (small ϵ), the amount of noise added can become very high. Moreover, there is a finite privacy budget for multiple searches, and high sensitivity queries demand a bigger amount of noise. DP may struggle with the privacy budget when the data distribution changes frequently. You may end up loosing utility or privacy or both in the long run. Also, the addition of a lot of noise to the output can make machine learning models less useful. Most of the privacy approaches in the online learning literature focuses on either storing the data or predicting the output privately. None or very few approaches in literature focuses on detecting drifts privately.

In our approach, we have considered Integral Privacy as an alternative to DP to generate high utility, privacy-preserving machine learning models. Integrally-private models provide sound defence against membership inference attacks and model comparison attacks. A membership inference attack is about getting access to the records used in the training process. On similar lines, a model comparison attack gives intruder access to the complete training set or to a huge portion of the training set through intersectional analysis. A machine learning model is integrally private [9] if it can be generated by multiple disjoint datasets. For an intruder whose aim is to do membership inference attacks or model comparison attacks, integrally-private models create ambiguity as the models are generated by multiple disjoint datasets. It has been proven in [11] that under some conditions it is possible to obtain, with probability close to one, the same parameter updates for a model with multiple minimatchs. They also find that a small fraction of a dataset can also lead to good results. One of the first works which shows the framework for model comparison attack and the defence by integral privacy for decision trees was given in [12]. The authors generate the complete model space and return the integrally private decision tree models which have approximately same model parameters. Generating complete (or approximately complete) model space can be a very computationally intensive task for a dataset with only few thousand instances.

For DNNs, generating model space and comparing models to find integrally private models can be tricky. This is due to the fact that for a given layer of two different models, equivalent neurons can be placed in different positions. Also, due to huge number of learning parameters in DNNs, there can be very few recurring models. In order to overcome these challenges for DNNs, a relaxed variant of integral privacy, Δ-Integral privacy, was proposed in [13]. Δ-Integral privacy (Δ-IP) considers models which are at most Δ distance apart, and then recommends the mean of these models (in the Δ range) as the integrally private model. The Δ-IP algorithm can recommend up to X number of integrally

private models which can be used as an ensemble of private models to detect concept drifts in streaming data. In this paper, we propose a methodology for drift detection through an ensemble of Δ-integrally private models. We compute an ensemble of Δ-IP models and use them to compute a measure of prediction uncertainty. This prediction uncertainty of Δ-IP models on the incoming datastream is used to detect concept drift. Our methodology only requires true labels to recompute the Δ-IP models once a drift has been detected. We also present the probabilistic analysis for the recurring models. Our theoretical analysis is inspired from the work in [11] which focuses on forging a minibatch. In our case, the analysis focus on learning similar parameters after complete training.

Our experimental setup shows results for ANN (one hidden layer with 10 neurons) and DNN of 3 hidden layers (10-20-10 hidden layer architecture). We evaluate our proposed methodology for 3 real-world dataset and 4 synthetic dataset. We have also compared our results with different levels of privacy in DP models. We show that our approach outperforms the DP alternatives. We find that ensemble of integrally private models can successfully detect concept drifts while maintaining the utility of non-private models.

The rest of the paper is organized as follows. Section 2 describes the background for the proposed drift detection methodology. Section 3 describes our proposed work. Section 4 gives the experimental analysis. The paper finishes with some conclusions and future work.

2 Background

In this section we describe the major concepts that are needed in this work.

2.1 Uncertainty in Neural Networks

Understanding the uncertainty of a model is essential to understand the model's confidence. In DNNs, class probabilities can not be the proxy for model's confidence. For unseen data, DNNs may give high probability even when the predictions are wrong. This can be the case in concept drifts i.e, the prediction may be uncertain but the system can give high class probability. Ensemble methods find the uncertainty using predictions from the family of DNNs. Here you train multiple DNNs with different initializations. In this way you generate a set of confidence parameters from multiple DNNs, and the variance of the output can be interpreted as the model uncertainty. In our work, we estimate the model uncertainty using an ensemble of private models. With drift in estimated uncertainty as an indicator for concept drift, we can employ drift detection schemes such as ADWIN, KSWIN to detect concept drift.

2.2 Model Comparison Attack and Δ-Integral privacy

Integral privacy [9] is a privacy model which provides defense against model comparison attacks and membership inference attacks. In a model Comparison Attack [12], [13], an intruder aims to get access to the sensitive information or do membership inference analysis by comparing the model parameters. A model

comparison attack assumes that the intruder has access to the global model M trained using algorithm A on the training set X a subset of the population $\mathcal{D}$, and some background information $S^*(\subseteq \mathcal{D})$. The intruder wants to get the maximum (or total) number of records used in the training process. I.e. the intruder wants to maximize access to X. The intruder draws a number of samples $S_1, S_2, ..., S_n$ from $\mathcal{D}$ and compares the model generated by each S_i with the global model M. Then, the intruder selects the S_i corresponding to the most similar model to M and hence guesses the records used in the training. In case that there are multiple models similar to M, the intruder can do intersectional analysis for membership inference. That is, find common data records which lead to the model M. In case of DNNs, model comparison can be tricky as highlighted before in [13]. The comparison between models is done by comparing each layer and neurons in respective layers.

In order to defend against such attacks, integral privacy requires you to chose a model which recurs from different disjoint datasets. Disjoint datasets are needed to avoid intersectional analysis. In this way an intruder cannot identify the training set because multiple training sets lead to the same model. As explained in Section 1, due to the huge number of parameters in DNNs there are very few recurring models. Δ-IP relaxes the equivalence relation between neurons. It allows the two models to be considered as equal if neurons in each layer of the respective model are at most Δ distance apart. Formally, Δ-IP can be defined as follows.

Δ-Integral Privacy Let $\mathcal{D}$ be the population, $S^* \subset \mathcal{D}$ be the background knowledge, and $M \subset \mathcal{M}$ be the model generated by an algorithm A on an unknown dataset $X \subset D$. Then, let $Gen^*(M, S, \Delta)$ represent the set of all generators consistent with background knowledge S^* and model M or models at most Δ different. Then, k-anonymity Δ-IP holds when $Gen^*(M, S, \Delta)$ has atleast k-elements and

$$\bigcap_{S \epsilon Gen^*(G, S^*, \Delta)} S = \varnothing \tag{1}$$

3 Proposed Methodology

In this section, we provide the details of our proposed methodology which we call *Integrally private drift detection* (IPDD) scheme. Our proposed IPDD methodology detects drifts with unlabeled data but assumes that true labels are available on request. Our approach is based on the detection of concept drifts from the measure of uncertainty in prediction by ensemble of private models. Previous works [14] [6] show that prediction uncertainty from DNN is correlated with prediction error. We argue on similar lines and use drift in prediction uncertainty as a proxy for detecting concept drift. We use Shanon entropy to evaluate the uncertainty over different c class labels.

Then any change detection algorithm such as ADWIN can be employed to detect drifts using this uncertainty measure. We chose ADWIN as it works well with real-valued inputs. The flowchart of the methodology is shown in Fig. 2.

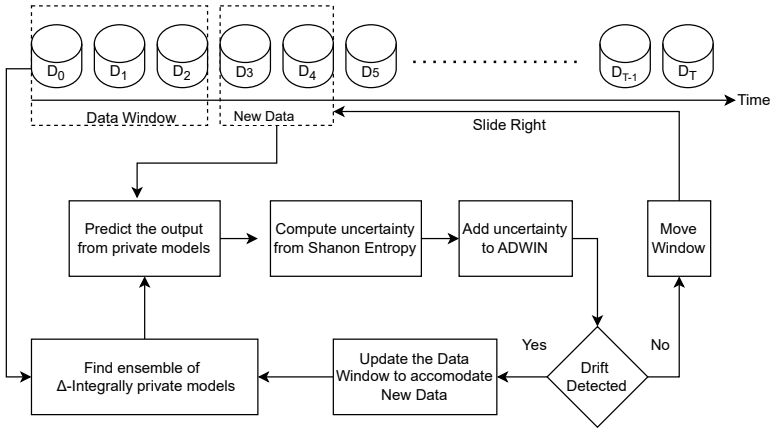


Fig. 2: Flowchart drift detection using ensemble of Δ-Integrally Private Models

Algorithm 1 Algorithm to generate k Δ-Integrally private DNNs for training data D. The algorithm return an ensemble of k private models

Inputs: D - Training data
Output: returns k integrally private models
Algorithm:
N - Size of subsamples
Δ - Privacy parameter
S = Generate_subsample(D, N)
▷ Generate n disjoint subsamples of size N
ModelList = [[]]
while $S_i \leftarrow S$ **do** ▷ For all samples in S
  $M_i \leftarrow$ Train DNN on S_i
  present ← False
  if M_i is utmost Δ distance apart from models in ModelList **then**
    Put M_i in the same bucket
    present ← True
  end if
  if present is False **then**
    Append M_i in ModelList ▷ Create a new bucket with M_i
  end if
end while
Returns mean of top k recurring models from ModelList

First k integrally private models are computed from the initial available training data. With incoming data instances, we predict the output and input the prediction uncertainty from each of the k private models to ADWIN. If drift has been detected, true labels are requested and the training data must be updated with new instances. Here, our methodology does not require true labels to detect

Algorithm 2 Drift detection using Δ-Integrally private models

Inputs: $\mathcal{D}$ - Dataset
Algorithm:
training_data = Initial_data($\mathcal{D}$)
▷ Initial Data to train private DNNs
Private_Models = Algorithm_1(training_data)
while $\mathcal{D}$ has elements **do** ▷ While stream has incoming data
  Receive incoming data x_t
  pred, uncertainty ← Private_Models.predict(x_t)
  Add uncertainty to ADWIN
  if ADWIN detects drift **then**
    Request true labels y_t for x_t
    Update training_data with x_t, y_t
    Private_Models = Algorithm_1(training_data)
  end if
end while

concept drift. New private models are computed from the updated training data. If the training data exceeds the threshold, records are removed on a first-come, first-served basis. In case there is no drift, then the prediction for new instances continues.

Algorithm 1 describes the computation for k private models. First, samples (i.e., sets of records) are generated from the training data in such a way that pairs of samples have empty intersection (i.e., they do not share any record). Models for these samples are computed using the same initialization. Models within Δ distance apart are kept in the same bucket. Buckets are sorted in descending order according to the number of models in each bucket. The algorithm then returns the mean of the top k recurring set of models as an ensemble of k integrally private models.

Algorithm 2 describes how to detect drifts privately. First, it uses initial available data as training data and computes private models using Algorithm 1 on the training data. Predictions on new incoming instances are used to compute the uncertainty measure and to see if the drift is detected using ADWIN. If the drift is detected, true labels must be requested, then training data must be updated and the private models are recomputed on a new training data. This process continues as long as the new data is available for prediction.

3.1 Theoretical Analysis

In this section, we present the probabilistic analysis for the recurrence of DNNs. DNNs are trained using mean samplers such as SGD, Adam etc. This analysis is inspired by the forgeability analysis done in [11]. The analysis in [11] computes the probability of forging a single minibatch while we focus on probabilistic analysis of learning the same model parameters after learning from different training data. Let us consider a set of disjoint datasamples, $D_1, D_2, ..., D_m$, i.i.d. (indepen-

dent and identically distributed) sampled from a given N-dimensional dataset $\mathcal{D}$ with some distribution. Here, each of the D_i is composed of b minibatches $\hat{x} = x_1, x_2, ..., x_b$. $M_1, M_2, ..., M_m$ be the DNN models we want to train which have the same initialization. The update rule looks like $g(w, \hat{x}) = \frac{1}{b}\sum_{i=1}^{b} g(w, x_i)$. The update rule $g(w, x)$ can be seen as a random variable with mean μ and σ^2 ($= \sum_{i=1}^{N} \sigma_i^2$, where σ_i^2 is the covariance of the i^{th} component of a random variable x sampled with distribution $\mathcal{D}$) as the trace of the covariance matrix. The mean sampler for the batch $\hat{x}$, $g(w, \hat{x})$ is still μ ($\frac{1}{b}\sum_{i=1}^{b} g(w, x_i) = \frac{1}{b} * b\mu$) but individual variance will get the $\frac{1}{b}$ i.e. now the trace of the covariance matrix is $\frac{1}{b}\sigma^2$.

Since the data samples are i.i.d sampled from $\mathcal{D}$ and each x_i is i.i.d sampled from data samples, then x_i follows the same distribution of $\mathcal{D}$. Then by Markov's inequality we can say that,

$$P(|g(w, \hat{x}) - \mu|_2 \geq \Delta) = P(|g(w, \hat{x}) - \mu|_2^2 \geq \Delta^2) \leq \frac{E(|g(w, \hat{x}) - \mu|_2^2)}{\Delta^2} \tag{2}$$

Here the first equality is true by the property of monotonicity of squares and the second is Markov's inequality. Note that $E(|g(w, \hat{x}) - \mu|_2^2)$ is just the trace of the covariance matrix ($\frac{1}{b}\sigma^2$). Then, we can write:

$$P(|g(w, \hat{x}) - \mu|_2^2 \geq \Delta^2) \leq \frac{\sigma^2}{b\Delta^2}$$
$$\Rightarrow P(|g(w, \hat{x}) - \mu|_2^2 \leq \Delta^2) \geq 1 - \frac{\sigma^2}{b\Delta^2} \tag{3}$$

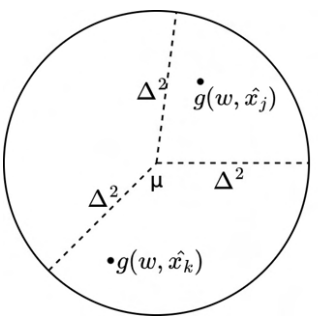


Fig. 3: Two models M_j, M_k at most Δ^2 distance apart from μ with probability defined in Eq. (5)

Let us consider two models M_j and M_k, training on data samples D_j and D_k with $\hat{x_j}, \hat{x_k}$. From Eq. (5), we can say $P(|g(w, \hat{x}_j) - \mu|_2^2 \leq \Delta^2) \geq (1 - \frac{\sigma^2}{b\Delta^2})$ and $P(g(w, \hat{x}_k) - \mu|_2^2 \leq \Delta^2) \geq (1 - \frac{\sigma^2}{b\Delta^2})$ at i^{th} epoch

As demonstrated in Fig. 3, if $g(w, \hat{x}_j), g(w, \hat{x}_k)$ are in the Δ^2 ball of μ with probability defined in Eq. (5) then with probability $\geq (1 - \frac{\sigma^2}{b\Delta^2})^2$ we can say both models are utmost $2\Delta^2$ distant. I.e. $P(|g(w, \hat{x}_j) - g(w, \hat{x}_k)| \leq 2\Delta^2) \geq (1 - \frac{\sigma^2}{b\Delta^2})^2$. Then, the probability that out of m models there exists two models which are in the Δ^2 ball of μ would be:

$$P(|g(w, \hat{x_j}) - g(w, \hat{x_k})|_2^2 \leq 2\Delta^2) \geq \sum_{r=2}^{m} \binom{m}{r} \left(1 - \frac{\sigma^2}{b\Delta^2}\right)^r \left(\frac{\sigma^2}{b\Delta^2}\right)^{m-r} \tag{4}$$

This is equivalent to having at least 2 models out of m in the $2\Delta^2$ ball of μ. The probability that m models are in the $2\Delta^2$ ball of μ would be:

$$P(|g(w,\hat{x_j}) - g(w,\hat{x_k})|_2^2 \leq 2\Delta^2) \geq \left(1 - \frac{\sigma^2}{b\Delta^2}\right)^m \tag{5}$$

Equation (6) and (7) represents the probability of weights updating in the $2\Delta^2$ ball of μ for a single epoch. For T iterations, the probability that there exists a model having weight updates in the $2\Delta^2$ ball of μ is at least $(\sum_{r=2}^{m}\binom{m}{r}(\frac{\sigma^2}{b\Delta^2})^{m-r}(1-\frac{\sigma^2}{b\Delta^2})^r)^T$. After T epochs, the probability of m models to be in the $2\Delta^2$ ball of μ will be at least $((1-\frac{\sigma^2}{b\Delta^2})^m)^T$.

So, for samples sampled i.i.d. from some dataset, we can conclude that the lower bound for the probability that there exists recurrent models within $2\Delta^2$ ball is at least $(\sum_{r=2}^{m}\binom{m}{r}(\frac{\sigma^2}{b\Delta^2})^{m-r}(1-\frac{\sigma^2}{b\Delta^2})^r)^T$. In addition, the probability that all the m models are in the $2\Delta^2$ ball of μ is at least $((1-\frac{\sigma^2}{b\Delta^2})^m)^T$. From this discussion, we have the following theorems.

Theorem 1. *If $D_1, D_2, ..., D_m$ are i.i.d samples from the dataset $\mathcal{D}$ with some distribution and b is the number of minibatches used for training in each of T epochs. Then under similar training environment i.e. same initialization, learning rate, etc. with probability greater than $(\sum_{r=2}^{m}\binom{m}{r}(\frac{\sigma^2}{b\Delta^2})^{m-r}(1-\frac{\sigma^2}{b\Delta^2})^r)^T$, the model will recur.*

Theorem 2. *With the above mentioned properties, a model satisfies k-anonymous integral privacy with probability atleast $(\sum_{r=k}^{m}\binom{m}{r}(\frac{\sigma^2}{b\Delta^2})^{m-r}(1-\frac{\sigma^2}{b\Delta^2})^r)^T$*

Proof: See Eq. (6) for proof. k-Anonymity integral privacy is equivalent to having at least k models out of m in the Δ ball of μ.

Remark on the choice of Δ, m: In order to generate higher k-Anonymity integrally private models, from theorem 2 we can say that increasing the number of i.i.d samples (m), b (Number of batches used in each epochs) and Δ (the distance value) increases the probability of getting recurrent models.

Role of initialization: The probabilistic analysis presented here gives you the lower bound that the model will recur from the samples having similar distribution. The probability can further improve when models are initialized with the same weight as the learning from similar dataset would result in the similar learning for the models.

4 Experiments

In this section, we present the experimental results for our proposed methodology. We will show that our methodology performs well with Categorical, Real, and Integer data with arbitrary number of classes. We perform our experiments on 3 real-world datasets namely Cover type (CovType), Electricity, and Susy dataset [15]. We also run our experiments on artificially generated Sine data and Insects data with abrupt, gradual and incremental drifts [16]. Table 1 shows the number of instances and other details of these datasets.

Dataset	# instances	# attribute	Data type	# classes
CovType	581012	54	Categorical Integer	7
Electricity	45312	8	Real Integer	2
Susy	5000000	18	Real	2
Sine	200000	4	Real	2
Insects_ab	52848	33	Real	6
Insects_grad	24150	33	Real	6
Insects_incre	57018	33	Real	6

Table 1: Details of the used Datasets

For our experiments, we have randomly considered a NN with a single hidden layer (10 neurons) architecture (We will call this architecture ANN) and a three hidden layer NN architecture with 10-20-10 number of neurons (we will call it DNN). For our experimental purpose we have chosen Δ = 0.01 and ADWIN parameter, δ = 0.001. For all the datasets, we have initially trained ANN and DNN over 10% of the dataset, and then stream is evaluated with 2% of the dataset at each time instance.

We compare our results (Integrally private drift detection, IPDD) with No retraining (No_retrain), ADWIN with unlimited label availability (ADWIN_unlim), and ADWIN with limited labels (ADWIN_lim). We have used three levels of differentially private models: high privacy (ϵ = 0.1) under limited label availability (DP_01), moderate privacy (ϵ = 0.5) under limited label availability (DP_05) and low privacy (ϵ = 1.0) under limited label availability (DP_10). All the results have been computed for ANN as well as DNN. The No_retraining model approach does not check for drifts, it trains the model with initial data once and only does the prediction for the rest of the data stream. For ADWIN_unlim we assume it has access to all the true labels of the incoming data stream and it detects drifts using the true labels only. The ADWIN_lim can have true labels upon request but detect the drifts using uncertainty through the ADWIN model. Similar settings were assumed for DP_01, DP_05, DP_10 and IPDD.

We can observe that our methodology IPDD has better or comparable accuracy score for both ANN and DNN. Table 2 provides the accuracy of the learned models. IPDD performs better than its counterparts for CovType and Electricity datasets, it has comparable accuracy score for Insects_ab and comparable results with ADWIN_unlim method. Table 3 provides the results for Mathews correlation coefficient (mcc) in the range $[-1, 1]$ (higher the better). MCC is a reliable statistical rate which assigns high value to a classifier if it performs good in all four confusion matrix categories. In comparison with differentially private models, IPDD performs much better than all three levels of differential privacy for all datasets except Insects_grad and Insects_incre. For ANN, IPDD performs performs better for CovType dataset and has comparable mcc rate with the rest. In case of DNNs, IPDD performs better than its counterparts for

Electricity, Susy and Insects_ab datasets; and performs comparable results for the rest of the datasets. Table 4 shows the score for the area under the curve (auc score). Auc score is the probability that a model ranks a random positive instance higher than a random negative instance. Table 4 highlights that auc score for IPDD's ANN and DNN performs better than its counterparts in case of all the datasets except Insect_grad and Insect_incre dataset.

We observed that with the addition of noise DP models may struggle to detect drifts. Table 5 shows the number of drifts detected by each method. It highlights that DP models at times may detect very few drifts because of the noise. On the other hand, IPDD detects comparable drifts to ADWIN_unlim and ADWIN_lim for both ANN and DNN. Table 5 also highlights that proposed IPDD does not detect unnecessary drifts i.e. IPDD does not necessary detect any drift when there is none (counterparts of IPDD does not detect any drift). As expected when the noise for DP models decreases, more drifts were detected by DP models as shown in Fig. 4.

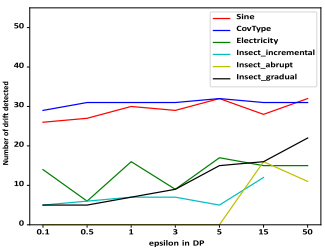


Fig. 4: Drift detected by different ϵ-differentially private models

In most of the cases, differentially private models does not perform as good as IPDD models even when the ϵ is very high (very low privacy). This is shown in Fig. 5. It compares the accuracy score between DNN model of DP and IPDD for all the datasets. Fig. 5a, 5b, 5c, 5d, 5e, and 5f highlights that even though the accuracy score improves for DP models, IPDD still performs better than DP. Only in case of Fig. 5g the DP model has slightly better accuracy score than IPDD even in case of high privacy.

Section 3.1 shows the probabilistic analysis on the lower bound of the recurrence of Integrally private models. As discussed in the remarks of Section 3.1, the higher the value of Δ the higher the number k-anonymity in integrally private models. For models with same initialization trained on 100 i.i.d samples $(D_1, D_2, ..., D_{100})$ from Sine dataset, the k in k-anonymity Integral privacy has been plotted against increasing Δ in Fig. 6a. As can be seen increasing the Δ value leads to the higher value of k in k-anonymity Integral privacy. Similarly, we can see in Fig. 6b that for a fixed $\Delta = 0.01$, increasing the number of i.i.d samples leads to a higher value of k in k-Anonymity integral privacy. It is important to highlight the distinction between k-anonymity and ensemble of k models chosen with k-anonymity Integral privacy. Simply, in k-Anonymity integral privacy, there exists a bucket which has at least k models while we require an ensemble of k such buckets.

We observe in Fig. 7a that for a fixed $\Delta = 0.01$, increasing the number of i.i.d samples also leads to the higher k in k-Anonymity integral privacy. In cases where all the models are clustered to only one IP model, generating an ensemble of such models can be tricky. An easier way to avoid this problem is to generate

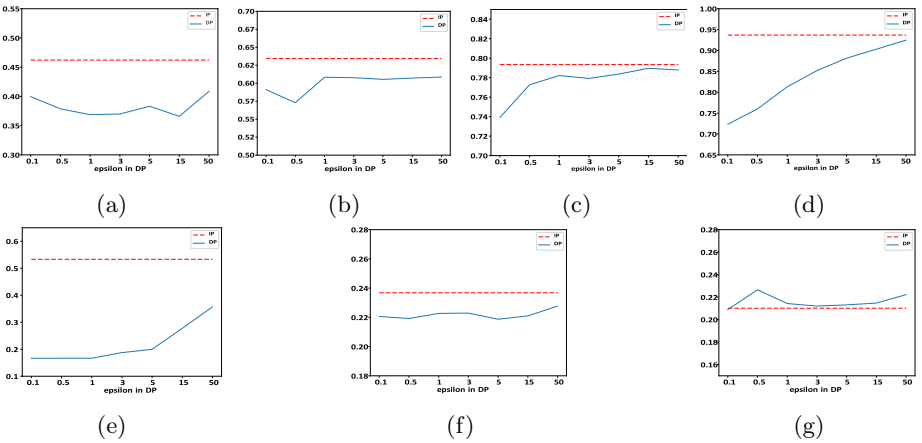


Fig. 5: Comparison of the accuracy score between differential privacy and integral privacy: (a) CovType (b) Electricity (c) Susy (d) Sine (e) Insect_ab (f) Insect_grad (g) Insect_incr.

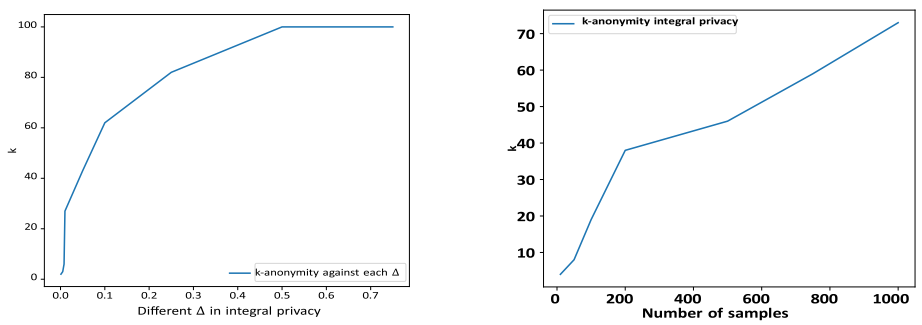


Fig. 6: K-anonymity integral privacy against (a) increasing Δ (b) increasing the number of i.i.d samples

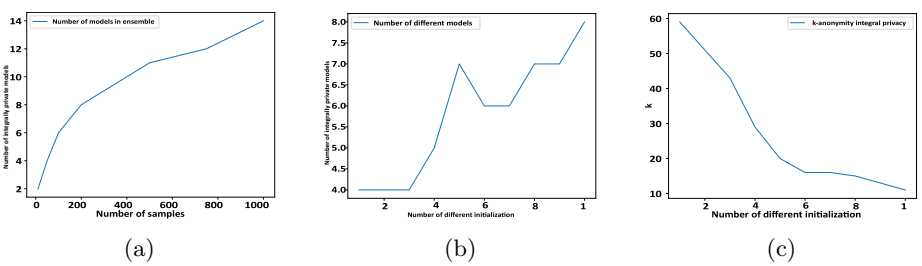


Fig. 7: Number of integral private models in an ensemble against (a) increasing the number of i.i.d samples (b) increasing the number of different initialization. (c) k-anonymity against different initialization

Table 2: Accuracy Score.

Dataset	No_retrain		Adwin_unlim		Adwin_lim		DP_01		DP_05		DP_10		IPDD	
	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN
CovType	0.4182	0.3552	0.4437	0.4582	0.4246	0.4012	0.4789	0.3996	0.4434	0.3787	0.4563	3688	**0.5836**	**0.4621**
Electricity	0.5754	0.5790	0.5928	0.6154	0.5953	0.6096	0.5875	0.5911	0.5994	0.5729	0.5978	0.6083	**0.6281**	**0.6343**
Susy	**0.7985**	**0.7985**	**0.7985**	**0.7985**	**0.7985**	**0.7985**	0.6844	0.7391	0.7327	0.7727	0.7539	0.7822	0.7729	**0.7934**
Sine	**0.9505**	**0.9536**	0.93425	0.9677	0.9421	0.9577	0.6204	0.6134	0.7600	0.6447	0.8129	0.9005	**0.9365**	**0.9369**
Insects_ab	0.4416	0.4434	0.5252	0.5237	0.2903	0.5348	0.1739	0.1666	0.2437	0.1667	0.2126	0.1667	**0.5241**	**0.5331**
Insects_grad	**0.2810**	**0.2841**	0.2293	0.2412	0.2221	0.232	0.2104	0.2206	0.2238	0.2192	0.2213	0.2224	0.2227	0.2368
Insects_incr	0.2181	**0.2475**	0.2282	**0.2475**	0.2335	0.2437	0.2140	0.2089	0.1914	0.2255	0.2143	0.2143	0.2022	0.2101

Table 3: Mathews Correlation Coefficient.

Dataset	No_retrain		Adwin_unlim		Adwin_lim		DP_01		DP_05		DP_10		IPDD	
	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN
CovType	0.0816	0.1940	0.1618	0.2941	0.1421	0.2140	0.1012	0.0599	0.0183	0.0261	-0.0158	**-0.0014**	**0.3575**	**0.2871**
Electricity	0.2652	0.0	0.1459	0.2101	0.1559	0.1898	0.0906	0.1120	0.1615	0.0033	0.1576	0.1817	**0.2593**	**0.2837**
Susy	**0.5936**	**0.5936**	**0.5936**	**0.5936**	**0.5936**	**0.5936**	0.3608	0.4782	0.4627	0.5411	0.5050	0.5623	0.5591	**0.5902**
Sine	0.8995	0.9059	0.8667	**0.9345**	0.8824	0.9142	0.1879	0.1974	0.5105	0.3335	0.6670	0.8024	**0.8726**	**0.8722**
Insects_ab	0.3841	0.3849	**0.4412**	**0.4398**	0.1274	0.4528	0.0129	0.0	0.1164	0.0014	0.0588	0.0	**0.4397**	**0.4505**
Insects_grad	0.1356	0.1181	0.0967	**0.1586**	0.0858	0.1291	0.0675	0.1176	0.1100	0.1333	0.1248	0.1258	0.1112	0.1181
Insects_incr	0.0628	**0.0978**	0.0746	**0.0978**	0.0809	0.0962	0.0834	0.0755	0.0356	0.0924	0.0902	0.0954	0.0448	0.0553

Table 4: Auc Score.

Dataset	No_retrain		Adwin_unlim		Adwin_lim		DP_01		DP_05		DP_10		IPDD	
	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN
CovType	0.7349	0.8801	0.7985	**0.9058**	0.7982	0.8596	0.5188	0.5197	0.5679	0.5422	0.5910	0.5293	**0.8881**	**0.9087**
Electricity	0.6195	0.5	0.5209	0.5508	0.5239	0.5433	0.5261	0.5430	0.5299	0.5004	0.5277	0.5424	**0.6311**	**0.6431**
Susy	**0.7923**	**0.7923**	**0.7923**	**0.7923**	**0.7923**	**0.7923**	0.6751	0.7275	0.7217	0.7674	0.7451	0.7741	**0.7582**	**0.7828**
Sine	0.9492	0.9531	0.9311	**0.9677**	0.9403	0.9565	0.5769	0.5776	0.7247	0.5979	0.8156	0.8912	**0.9381**	**0.9339**
Insects_ab	0.7989	0.80	0.8469	0.8587	0.6310	0.8469	0.5829	0.5137	0.5966	0.5749	0.5528	0.5516	**0.8474**	**0.8581**
Insects_grad	0.6283	0.6206	0.6190	0.6068	0.6024	0.6118	0.5377	**0.6220**	0.5800	0.6067	0.5779	0.6219	0.5508	**0.5956**
Insects_incr	0.5439	0.5795	0.5470	0.5799	0.5577	0.5730	0.6026	0.5841	0.6035	**0.6245**	0.6044	0.6161	0.5249	0.5362

Table 5: Number of Drifts detected by each algorithm.

Dataset	No_retrain		Adwin_unlim		Adwin_lim		DP_01		DP_05		DP_10		IPDD	
	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN	ANN	DNN
CovType	0	0	35	35	31	32	24	29	24	31	18	31	**35**	**33**
Electricity	0	0	37	37	42	39	13	14	8	6	15	16	**23**	**24**
Susy	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sine	0	0	28	28	33	28	27	31	27	21	28	28	17	14
Insects_ab	0	0	24	24	23	17	**0**	**0**	**5**	**0**	**12**	**0**	16	14
Insects_grad	0	0	18	18	16	17	**5**	**5**	**5**	**5**	**5**	**7**	14	15
Insects_incr	0	0	0	0	16	14	**7**	**5**	**8**	**6**	**6**	**7**	**15**	**19**

an ensemble of k-anonymity models using different initializations. The reason for this could be attributed to the comparable learning process when using similar training data. For 100 i.i.d samples of Sine data, in Fig. 7b, x-axis shows the number of different initialization and y-axis shows the number of different IP models in an ensemble. It is important to note here that for 100 samples if the number of IP models increases in an ensemble, k-anonymity of each IP model will decrease as depicted in Fig. 7c.

4.1 Limitations of our approach:

The analysis of our method as well as our experiment permits us to state the following.

1. Generating k-anonymous integrally private models requires training on large number of samples which is a time consuming process. The proposed IPDD methodology has running time as the cost of privacy.
2. As shown in Section 3.1, the generation of integrally private models is a probabilistic approach and depends on the samples selected. That is, different runs can provide different results.

5 Conclusion and Future work

In this paper we have presented a private drift detection methodology called 'Integral Privacy Drift Detection' (IPDD). Our methodology detects drifts using an ensemble of k-anonymity integrally private models. Simply, we generate an ensemble of k models which are recurring from multiple disjoint datasets. Our methodology does not require the ground truth to detect concept drift but assumes they are available for retraining. We find that our methodology can successfully detect concept drifts while maintaining the utility of non-private models. It is useful in generating models which have comparable (better in some cases) accuracy score, mcc score and auc score against ADWIN with unlimited label availability and limited label availability. In comparison with its differentially private counterpart, IPDD perfoms significantly better in most of the cases.

As shown above different parameters can lead to different levels of privacy. It can also affect the number of drifts detected and the utilty of the model. Fine-tuning of these parameters for each application is an interesting direction for future work. Furthermore, extension of our work for non-i.i.d. samples would be an interesting future direction.

Acknowledgement: This work was partially supported by the Wallenberg Al, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by the supercomputing resource Berzelius provided by National Supercomputer Centre at Linköping University and the Knut and Alice Wallenberg foundation.

References

[1] P. Schwab and W. Karlen, "Cxplain: Causal explanations for model interpretation under uncertainty," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[2] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

[3] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17*, Springer, 2004, pp. 286–295.

[4] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," *Big data analysis: new algorithms for a new society*, pp. 91–114, 2016.

[5] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM international conference on data mining*, SIAM, 2007, pp. 443–448.

[6] L. Baier, T. Schlör, J. Schöffer, and N. Kühl, "Detecting concept drift with neural network model uncertainty," *arXiv preprint arXiv:2107.01873*, 2021.

[7] P. Samarati, "Protecting respondents identities in microdata release," *IEEE transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[8] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[9] V. Torra, G. Navarro-Arribas, and E. Galván, "Explaining recurrent machine learning models: Integral privacy revisited," in *International Conference on Privacy in Statistical Databases*, Springer, 2020, pp. 62–73.

[10] J. Salas and V. Torra, "A general algorithm for k-anonymity on dynamic databases," in *Data privacy management, cryptocurrencies and blockchain technology*, Springer, 2018, pp. 407–414.

[11] A. Thudi, H. Jia, I. Shumailov, and N. Papernot, "On the necessity of auditable algorithmic definitions for machine unlearning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 4007–4022.

[12] N. Senavirathne and V. Torra, "Integrally private model selection for decision trees," *computers & security*, vol. 83, pp. 167–181, 2019.

[13] A. K. Varshney and V. Torra, "Integrally private model selection for deep neural networks," *DEXA 2023, DOI: https://doi.org/10.21203/rs.3.rs-2944008/v1*, 2023.

[14] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.

[15] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: `http://archive.ics.uci.edu/ml`.

[16] V. M. Souza, D. M. dos Reis, A. G. Maletzke, and G. E. Batista, "Challenges in benchmarking stream learning algorithms with real-world data," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1805–1858, 2020.

Paper IV

k-IPfedAvg: k-Anonymous Integrally Private Federated Averaging with Convergence Guarantee

Ayush K. Varshney, *Student Member, IEEE,* Vicenç Torra, *Fellow, IEEE*

***Abstract*—Federated Learning (FL) has established itself as a widely accepted distributed paradigm. Without sharing data, it may seem like a privacy-preserving paradigm, but recent studies have revealed vulnerabilities in weight sharing which results in information disclosure. Hence, privacy-preserving approaches must be incorporated during aggregation to avoid disclosures.**

In the literature of FL, not much focus has been given on generating generalized models which can be generated by multiple sets of datasets thus avoiding identity disclosure. Integrally private models are the models which recur frequently from different datasets. So, in this paper we focus on generating the integrally private global models proposing k-Anonymous Integrally Private Federated Averaging (k-IPfedAvg), a novel aggregation algorithm which clusters similar user weights to compute a global model which can be generated by multiple sets of users. Convergence analysis of k-IPfedAvg reveals a rate of $\mathcal{O}(\frac{1}{T})$ over training epochs. Furthermore, the experimental analysis shows that k-IPfedAvg maintains a consistent level of utility across various privacy parameters in contrast to existing noise based privacy-preserving mechanisms. We have compared k-IPfedAvg with classical fedAvg and its differentially private counterpart. Our results shows that k-IPfedAvg has comparable accuracy score with baseline fedAvg and outperforms DP-fedAvg on iid and non-iid distributions of MNIST, FashionMNIST and CIFAR10 datasets.

***Index Terms*—Data privacy, federated learning, integral privacy, generalized models.**

I. Introduction

In recent years, artificial intelligence (AI) and machine learning (ML) have undeniably revolutionized a multitude of disciplines, ranging from healthcare [1] and finance [2] to arts and communication [3] and many others. These technologies can uncover patterns from large information, perform predictive studies, and even simulate human decision-making processes due to their ability to learn from data and powerful computing resources. The widespread use of AI and ML applications across a range of industries not only highlights the tools' disruptive potential but also necessitates a thorough analysis of their techniques, ramifications, and ethical implications. The crucial issue of data privacy is integral to these factors. This issue becomes even more pronounced when one considers the vast quantities of data available to these models. Traditional machine learning models were trained in an environment where data is aggregated on a single server or cluster which poses significant data privacy risks.

Federated learning (FL) [4] framework has emerged as a feasible paradigm which allows multiple users to train a shared model without requiring them to share their raw sensitive data. This makes FL, the most sought distributed machine learning framework. McMahan et al. [4] introduced federated averaging (fedAvg) which is the first and perhaps the most widely used algorithm to aggregate the models trained on user data. Fedavg has been shown to be communication efficient and converges on iid as well as non-iid[1] data. FL framework involves a central server and can have any number of participating users. Central server first distributes a global model to the users. Users then optimize in parallel the global model using stochastic gradient descent (SGD) or its variants on their local data. The central server aggregates the model parameters of the users to construct a new global model. This process is repeated until a well performing model is obtained. Unlike the traditional ML, only the model parameters are transmitted between server and users.

Since, the FL server does not have access to the users local data, the problem of heterogeneity (non-iid) is a big challenge. With data heterogeneity, the training data across multiple devices does not belong to a single distribution. User data can come from various distributions. Some users may have data that follow a common distribution, while other users may have data following other different distributions. In such cases, the global model can diverge considerably with the one we would infer from user data, and due to this divergence, the construction of the model may exhibit slow convergence. Apart from this, FL has its own privacy challenges, the weights exchanged between the server and the user encodes the private information of the users local data which bears the risk of privacy leakage using attacks such as model inversion attacks [5], membership inference attacks [6], data poisoning attacks [7] and many more. Such leakage is very costly and thus ensuring user's privacy is critical to increase the impact of federated learning in day-to-day life.

Recent works have started employing privacy-preserving mechanisms in FL framework to overcome the privacy challenges. Privacy models such as k-anonymity [8], differential privacy (DP) [9] and their variants have dominated most of the research in the literature. In a FL environment, k-Anonymity can safeguard the local data on users' devices. Specifically, before training the global model, users can protect their data by aggregating each data instance with the closest $k-1$ instances.

Manuscript submitted December 21, 2023. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

A. K. Varshney and V. Torra are with the Department of Computing Sciences, Umeå University, 90740 Umeå, Sweden (e-mail: ayushkv@cs.umu.se; vtorra@cs.umu.se).

correspondence: ayushkv@cs.umu.se

[1]Throughout the paper, "non-iid" means data is not identically distributed.

While this method is beneficial for publishing data without disclosing individual data instances, there currently exists no approach in the literature that protects user identities during the aggregation of user weights in federated averaging. The differentially private approaches clip the model updates and add artificial noise in order to prevent information leakage. The differentially private solutions [10] offer theoretically sound privacy but have practical limitations. Due to heterogeneity of the data, clipping the model weights across all the layers may imply a significant information loss. If the noise is added in each training step, the privacy budget will accumulate causing the explosion of the total privacy budget.

Considering the drawbacks of the existing privacy mechanism, in this paper we explore Integral privacy as an alternative to k-Anonymity and DP to generate high utility, privacy-preserving models. A machine learning model is considered to be integrally private if it recurs from multiple disjoint datasets. Previous work shows [11] a methodology for generating an integrally private solution and provides [12] a probabilistic analysis for the recurrence of the models. The private models generated have comparable utility with non-private models however the recurrence of the models is probabilistic. It has been shown [13] [14] that, given a similar training environment and a large batch size, models trained on data sampled from certain distributions will likely have their gradient updates separated by only a small distance (say Δ), with a probability close to 1. This gives no guarantee on the number of models that can be in the range of Δ i.e. the precise number of times a model recurs is not known. In this work, we remove this drawback by considering k-nearest distant models as recurring models (since $\Delta << 1$). This allows us to define a privacy-preserving aggregation mechanism for FL, which we call k-anonymous integrally private federated averaging (k-IPfedAvg). In k-IPfedAvg, the server identifies the clusters of the nearest k models. The server randomly picks a model from each cluster and aggregates them in order to generate the global model for the next round. Since the models in the cluster are at most Δ distant apart, the global model k-IPfedAvg is also at most Δ distance apart. With the assumption of Δ being small, we prove that the global model from k-IPfedAvg converges on iid as well as non-iid data. The main contributions of this paper are summarized as follows:

- We propose a novel k-anonymous integrally private federated average algorithm which protects the identity disclosure of the clients participating in the training.
- We provide theoretical analysis for the convergence of k-IPfedAvg algorithm. We find that k-IPfedAvg has $\mathcal{O}(\frac{1}{T})$ convergence rate, where T represents the total number of training epochs.
- We compare our results on 3 well known datasets: MNIST, CIFAR10, FashionMNIST. Our results find that k-IPfedAvg outperforms its existing counterparts.
- We find that k-IPfedAvg has a marginal effect on utility.

The rest of the paper is organized as follows. Section II describes the background for the proposed aggregation methodology. Section III describes our proposed work. Section IV gives the experimental analysis. The paper finishes with some conclusions and future work in Section V.

II. Background and Related works

A. Related works

Federated learning proposed by McMahan et al. [4] is a type of collaborative learning without collecting users data. In the literature of FL, most of the work is focused on efficient communication and data privacy. Our work focuses on the data privacy issue of FL.

Privacy attacks in FL. Although the FL framework does not require the local data, recent studies have found several attacks that can lead to data breaches. Typical FL attacks are: Data reconstruction attack (reconstruct training records based on the model weights) [15], membership inference attack (infer whether a data record participated in the training) [6] and data poisoning attacks (several malicious participants manipulate the model weight in order to get desired inference) [7].

Privacy models in FL. k-Anonymity, by Samarati et al. [8], is employed to safeguard a dataset prior to its release by ensuring that each data record in the dataset is indistinguishable from at least k-1 other records. k-Anonymity and its variants (such as l-diversity [16] and t-closeness [17]) provides defense against membership inference attacks and reconstruction attacks. For example, in [17], discriminative attributes were identified and anonymized on the local devices before syntactic learning at the server. In [18] based on the degree of privacy, each client decides the amount of data to be shared with the server, in [19] a hierarchical structure was given where the server communicated with clients who in turn have sub-clients based on the distance measure and many others. On the other hand, differential privacy [9] and its variants (such as local-DP [20], client-level DP [10]) offer privacy guarantee by adding noise. In FL, in each iteration DP is often guaranteed by adding noise from a Laplacian or Gaussian distribution during training on local devices with an assumption of a trusted server which aggregates their results and this continues for a fixed number of rounds. The issue of non-trusted servers can be resolved by taking a decentralized approach in the DP-FL literature. For example, in [21], a peer-to-peer structure for FL is given, where the aggregation is handled by each node participating in the communication round, in [22] for each training iteration a master node is selected at random which aggregates the global model and send it to all the devices. Further updates in the DP-FL literature can be found in [23]. However, the privacy budget for DP-FL steadily increases with each round of communication between server and users which may lead to budget explosion for DP-FL. Apart from this, DP-FL approaches in literature provably can not achieve the data anonymization and de-identification required in the regulations such as GDPR and HIPAA [24].

Convergence of fedAvg. Convergence of fedAvg has been proved in several works, e.g. in [25] convergence of distributed SGD was proved under the assumption that the data are iid and all the users participate in a single round of communication, in [26], the convergence assuming the all devices participate in a single round of communication was given, convergence

in case of non-iid data and partial participation was proved in [27]. Our convergence analysis is inspired from the work in [25] and [27].

B. Federated Learning

This section provides an overview of a well recognized federated learning (FL) framework [4]. FL is a type of distributed learning that accounts for non-identical and independently distributed (non-iid) data. The typical FL optimization at the server looks like:

$$\min_{w} \left\{ F(w) \triangleq \sum_{l=1}^{N} p_l F_l(w) \right\} \tag{1}$$

where N is the total number of user devices communicating model weights, p_l ($p_l \geq 0$ and $\sum_{l=1}^{N} p_l = 1$) represents the weight of the l^{th} user and the function $F_l(w)$ is the local objective function. The local objective function $F_l(w)$ with the model weights w aims to minimize the loss (represented by the loss function, $l()$) on the local data. Assuming the l^{th} user has n_l training instances $((x_1, y_1), (x_2, y_2)...., (x_{n_l}, y_{n_l}))$. Then, $F_l(w)$ is defined as

$$F_l(w) \triangleq \frac{1}{n_l} \sum_{i=1}^{n_l} l(w; x_i, y_i) \tag{2}$$

Algorithm 1 Federated averaging (fedAvg)

Server side
Initialize global model w_0
for $t = 1, 2, ..., \lfloor \frac{T}{E} \rfloor$ **do** ▷ communication rounds
  Disseminate w_t to the user devices
  for each user $l = 1, 2, .., N$ **do**
    $w_{t+1}^{l} =$ **UserUpdate** (w_t)
  end for
  $w_{t+1} = \sum_{l=1}^{N} p_l w_{t+1}^{l}$ ▷ Weighted Aggregation
end for
UserUpdate(w_t)
Consider $w = w_t$ as initial weight
for local epochs $e = 1, 2, .., E$ **do**
  $w \leftarrow w - \eta_t \nabla F_l(w, \xi_l^{t+e})$
end for
return w

In a typical federated learning setup, a central server initializes the global model. During each round of communication, this global model is sent to the active clients. The clients then train the model with their own data for some epochs and send their updated models back to the server. The central server then combines these updated models from the clients. This cycle is repeated over a specified number of communication rounds. When all clients in the network are involved in training the global model in each round, it is known as full-device participation. On the other hand in partial-device participation, only a set of random users participate to train the global model. The federated averaging (fedAvg) algorithm for full-device participation, where all N clients participate, is outlined in Algorithm 1. For partial-device participation, the model is communicated to selected few clients and only their updates are considered for aggregation. In Algorithm 1, T represents the total number of SGD steps, N is the number of users, E is the number of local epochs, η_t is the learning rate during t^{th} communication round, and ξ_l^{t+e} is a sample randomly chosen from l^{th} client's data.

C. Integral Privacy

Integral privacy [11][28] is a privacy model focused on addressing model comparison attacks and membership inference attacks. Integrally private models are the models which recur from multiple sets of disjoint datasets. For deep learning models, the number of weights are huge and generating precisely the exact same weights with disjoint datasets is challenging. The authors in [11] introduced a flexible Δ-Integral privacy (Δ-IP) for DNNs which considers two models similar even when they are Δ distance apart. A model is k-anonymous Δ-integrally private if there exists at least $k-1$ other similar models. Formally Δ-IP is defined below.

Δ-Integral Privacy Let $\mathcal{D}$ represent the population, $S^* \subset \mathcal{D}$ be the background knowledge, and $M \subset \mathcal{M}$ be the model generated using algorithm A on an unknown dataset $X \subset D$. Then, let $Gen^*(M, S, \Delta)$ represent the set of all generators consistent with background knowledge but not including S^* and model M or models at most Δ distant. Then, k-anonymity Δ-IP holds when $Gen^*(M, S, \Delta)$ has at least k-elements and

$$\bigcap_{S \in Gen^*(G, S^*, \Delta)} S = \emptyset. \tag{3}$$

III. Proposed Work

In this section we present our k-anonymous integrally private federated averaging along with its convergence analysis for strongly convex and smooth functions.

k-IPfedAvg. In k-anonymous integrally private fedAvg, we cluster the weights according to some distance measure. The server randomly selects one participant from each cluster as their representative and aggregate them to get global model parameters. In our case the optimization problem looks like:

$$\min_{w} \left\{ F(w) := \sum_{c=1}^{|C|} p_c F_c(w) \right\} \tag{4}$$

where $|C| = \lfloor \frac{N}{k} \rfloor$ is the number of clusters, k is the privacy parameter, $p_c = \sum_{i \in C_c} p_i$ and $F_c(w)$ is the local objective function of the randomly selected participant in cluster C_c.

Fig. 1 shows the framework for k-IPfedAvg. In a typical round of communication (say t-th) in k-IPfedAvg, a server broadcasts the latest model, w_t to all the user devices. The devices then train the w_t for E epochs on their local data. All the devices have similar training environment i.e., they have similar learning rate for each round (η_t), fixed E, similar optimizer (SGD in our case) and so on. After local training, the server clusters the received local models into $|C|$ clusters based on some distance measure (say $dist(w_{t+1}^{i}, w_{t+1}^{j})$), each cluster has between $[k, 2k]$ number of local models. In the end, the server randomly choses a representative of the cluster and

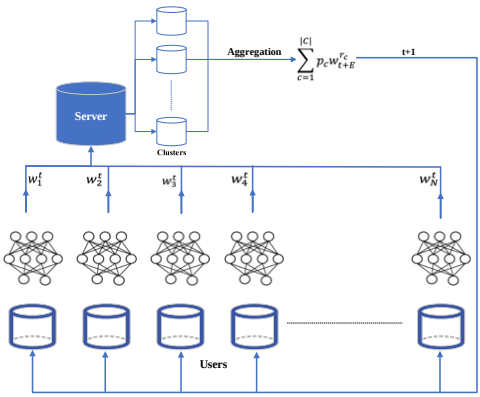


Fig. 1: Generic k-Anonymous Integrally Private Federated Averaging framework

Algorithm 2 k-Anonymous Integrally Private Federated Averaging (k-IPfedAvg)

Server side
Initialize global model w_0
for $t = 1, 2, ..., \lfloor \frac{T}{E} \rfloor$ **do** ▷ communication rounds
  Broadcast w_t to the clients
  for each client $l = 1, 2, .., N$ **do**
    $w_{t+1}^l =$ **ClientUpdate** (w_t)
  end for
  Compute clusters $C = C_1, C_2, ...C_{\lfloor \frac{N}{k} \rfloor}$
  $w_{t+1} = \sum_{c=1}^{|C|} p_c w_{t+1}^{r_c}$ ▷ Aggregate randomly chosen models
end for
ClientUpdate(w_t)
Consider $w = w_t$ as initial weight
for local epochs $e = 1, 2, .., E$ **do**
  $w \leftarrow w - \eta_t \nabla F_l(w, \xi_l^{t+e})$
end for
return w

aggregates the representative to produce a new global model w_{t+E} i.e.,

$$w_{t+E} := \sum_{c=1}^{|C|} p_c w_{t+E}^{r_c} \tag{5}$$

where $w_{t+E}^{r_c}$ is a randomly selected parameter from each cluster.

When the models are clustered such that the distance between them is small then according the definition of Δ-Integral privacy (see section II-C), we can call such models as integrally private i.e. in a given cluster C, if $dist(w_t^i, w_t^j) \leq \Delta \ \ \forall i, j \in C$. Algorithm 2 provides the formal algorithm for k-Anonymous Integrally Private Federated Averaging where the parameter k is a privacy parameter which determines the number of clusters and the number of weights in each cluster. The server has a predefined value of k, then in each round of communication it broadcasts the global model to all the clients, clients in turn train the received model on their local data and communicate the updated model back to the server. The server clusters the model weights, then it randomly chooses a model from each cluster as its representative and aggregates them. This process continues for a given number of communication rounds or until the convergence is obtained.

A. Theoretical Analysis

In this section, we focus on the convergence analysis of proposed k-IPfedAvg. We will prove that just like fedAvg, k-IPfedAvg also has convergence rate of $\mathcal{O}(\frac{1}{T})$. We cluster the weights from each clients based on some distance measure and aggregate representatives from each cluster to generate global weights. Our work is similar to the one of Li et al. [27]. In our work, however, in each round of communication, users who are chosen as a cluster representative participate to generate a global model.

Let N be the number of user devices participating in each round for federated averaging by a trusted server. Let T be the total number of iterations for SGDs on all the user devices, E be the number of local iterations of SGDs on each user device. $F_1, F_2, ..., F_N$ be the local objective functions on each device. Let F^*, F_l^* be the minima for the global and local objective functions. Let $\Gamma = F^* - \sum_{l=1}^{N} p_l F_l^*$ represent the degree of non-iid [27]. We consider the following assumptions in our work:

Assumption 1. $F_1, F_2, ..., F_N$ are all L-smooth i.e., $\forall\ x, y : F_l(x) \leq F_l(y) + (x-y)^T \nabla F_l(y) + \frac{L}{2}\|x-y\|_2^2$

Assumption 2. $F_1, F_2, ..., F_N$ are all μ- strongly convex i.e., $\forall\ x, y : F_l(x) \geq F_l(y) + (x-y)^T \nabla F_l(y) + \frac{\mu}{2}\|x-y\|_2^2$

Assumption 3. Let ξ be uniformly sampled at random from the l-th device's local data. Then, for each device l, the variance of SGD is bounded i.e. there exists σ_l such that $\mathbb{E}\|\nabla F_l^{\xi}(w_t^l) - \nabla F_l(w_t^l)\|^2 \leq \sigma_l^2$.

Assumption 4. In all the communication rounds, for each device, the expected squared norm of SGD is bounded i.e., $\mathbb{E}\|\nabla F_l^{\xi}(w_t^l)\|^2 \leq G^2$.

Assumption 5. For a given batch size b and a large N there exists at least k samples (ξ) in each of the non-iid distributions. Then, as a consequence we assume that for each cluster $c \in C$,

$$\mathbb{E}|\nabla F_r^{\xi}(w_t^{r_c})| = \sum_{l \in C} \frac{p_l}{p_c} \nabla F_l(w_t^l).$$

Assumptions 1 and 2 are typical assumptions in machine learning literature for convergence analysis. While Assumptions 3 and 4 were considered especially for convergence analysis of federated averaging in [25] and [27]. For Assumption 5, we consider a set of M non-iid distributions $(\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_M)$ and each device draws all samples from any (but only one) of these distributions. When we have a large number of user devices then we have enough samples following a similar distribution. This assumption is needed in a proof later. In

particular, Assumption 5 implies

$$\begin{aligned}\mathbb{E}|\sum_{c=1}^{|C|} p_c \nabla F_r^{\xi}(w_t^{r_c})| &= \sum_{c=1}^{|C|} p_c \mathbb{E}|\nabla F_r^{\xi}(w_t^{r_c})| \\ &= \sum_{c=1}^{|C|} p_c \sum_{l\in C} \frac{p_l}{p_c} \nabla F_l(w_t^l) \\ &= \sum_{l=1}^{N} p_l \nabla F_l(w_t^l) \end{aligned} \tag{6}$$

Intra-cluster weight distance: We define intra-cluster δ_c $(c = 1, 2, .., |C|)$ as the maximum distance between the group average and user weights for a given cluster (say C_c) in a single round of communication in k-IPfedAvg. Mathematically,

$$\delta_c^t = \max_{l' \in C_c} \left\| \frac{\sum_{l\in C_c} p_l}{p_c} \nabla F_l^{\xi}(w_t^l) - \nabla F_{l'}^{\xi}(w_t^{l'}) \right\|_2 \tag{7}$$

Hence, for any member (say r) of the cluster C_c, $p_c \nabla F_r^{\xi}(w_t^r) \geq \sum_{l\in C_c} p_l \nabla F_l^{\xi}(w_t^l) - p_c \delta_c^t$, or $p_c \nabla F_r^{\xi}(w_t^r) \leq \sum_{l\in C_c} p_l \nabla F_l^{\xi}(w_t^l) + p_c \delta_c^t$ holds. The inter-cluster distance between the distance is defined as.

$$\delta^t = \sum_{c=1}^{|C|} p_c \delta_c^t \tag{8}$$

Let $\mathcal{I}_E$ be the set synchronization step for the global model i.e. server only performs the aggregation when $t \in \mathcal{I}_E$. Let v_{t+1}^l represent the immediate result for one step of SGD. i.e. $v_{t+1}^l = w_t^l - \eta_t \nabla F_l^{\xi}(w_t^l)$ ($\nabla F_l^{\xi}(w_t^l)$ is the gradient on sample ξ for the user l) and w_{t+1}^l be the weight for l^{th} user after its communication with the server i.e.,

$$w_{t+1}^l = \begin{cases} v_{t+1}^l, & \text{if } t+1 \notin \mathcal{I}_E \\ \sum_{c=1}^{|C|} p_c v_{t+1}^{r_c} & \text{otherwise} \end{cases}$$

where $v_{t+1}^{r_c}$ is the intermediate result for the client randomly selected from each cluster. We also define two virtual sequences such as in [25], [27], $\overline{v}_t = \sum_{c=1}^{|C|} p_c v_t^{r_c}$ and $\overline{w}_t = \sum_{c=1}^{|C|} p_c w_t^{r_c}$. For the sake of convenience we also define, $\overline{g}_t = \sum_{l=1}^{N} p_l \nabla F_l(w_t^l)$ and $g_t = \sum_{c=1}^{|C|} p_c \nabla F_r^{\xi}(w_t^{r_c})$. Then, $\overline{v}_{t+1} = \overline{w}_t - \eta_t g_t$ and under Assumption 5, in a cluster the user's gradients are similar i.e. $\mathbb{E}|g_t| = \overline{g}_t$ (Equation 6).

Lemma 1. Under Assumption 3, the following holds:

$$\mathbb{E}\|\overline{g}_t - g_t\|^2 \leq \sum_l p_l^2 \sigma_l^2$$

Proof. Using the definition of $\overline{g}_t$ and g_t in the left hand side of the inequality, we get:

$$\begin{aligned}\mathbb{E}\|\overline{g}_t - g_t\|^2 &= \mathbb{E}\|\sum_{l=1}^{N} p_l \nabla F_l(w_t^l) - \sum_{c=1}^{|C|} p_c \nabla F_{r_c}^{\xi}(w_t^{r_c})\|^2 \\ &= \mathbb{E}\|\sum_{c=1}^{|C|} p_c \sum_{l\in C} \frac{p_l}{p_c} \nabla F_l(w_t^l) - \sum_{c=1}^{|C|} p_c \nabla F_{r_c}^{\xi}(w_t^{r_c})\|^2 \\ &= \mathbb{E}\|\sum_{c=1}^{|C|} p_c \left(\sum_{l\in C} \frac{p_l}{p_c} \nabla F_l(w_t^l) - \nabla F_{r_c}^{\xi}(w_t^{r_c}) \right) \|^2 \\ &= \mathbb{E}\|\sum_{c=1}^{|C|} p_c \left(\sum_{l\in C} \frac{p_l}{p_c} \nabla F_l(w_t^l) - \sum_{l\in C} \frac{p_l}{p_c} \nabla F_l^{\xi}(w_t^l) \right) \|^2 \\ &= \mathbb{E}\|\sum_{c=1}^{|C|} p_c \sum_{l\in C} \frac{p_l}{p_c} \left(\nabla F_l(w_t^l) - \nabla F_l^{\xi}(w_t^l) \right) \|^2 \\ &= \mathbb{E}\|\sum_{c=1}^{|C|} \sum_{l\in C} p_l \left(\nabla F_l(w_t^l) - \nabla F_l^{\xi}(w_t^l) \right) \|^2 \\ &= \sum_{c=1}^{|C|} \sum_{l\in C} p_l^2 \mathbb{E}\|\nabla F_l(w_t^l) - \nabla F_l^{\xi}(w_t^l)\|^2 \\ &= \sum_{c=1}^{|C|} \sum_{l\in C} p_l^2 \sigma_l^2 = \sum_{l=1}^{N} p_l^2 \sigma_l^2 \end{aligned} \tag{9}$$

Lemma 2. Under Assumption 4 and non-increasing η_t such that $\eta_t \leq 2\eta_{t+E}$ $\forall\, t \geq 0$, we find:

$$\mathbb{E}\left[\sum_{l=1}^{N} p_l \|\overline{w}_t - w_t^l\|^2\right] \leq 4\eta_t^2 (E-1)^2 G^2$$

Proof. The proof follows the proof of Lemma 3 from [27]. In our algorithm we are clustering the models and then selecting at random a model r_c from each cluster c in C. Nevertheless, similar to [27], we are also communicating at each E steps. Therefore, as in [27], for any $t \leq 0$, there exists a $t_0 \leq t$ such that $t - t_0 \leq E - 1$ and $w_{t_0}^l = \overline{w}_{t_0}$. Also, we have η_t non-decreasing and $\eta_{t_0} \leq 2\eta_t$. Then,

$$\mathbb{E}\sum_{l=1}^{N} p_l \|\overline{w}_t - w_t^l\|^2 = \mathbb{E}\sum_{l=1}^{N} p_l \|(w_t^l - \overline{w}_{t_0}) - (\overline{w}_t - \overline{w}_{t_0})\|^2$$

Then, using $\mathbb{E}\|X - EX\|^2 \leq \mathbb{E}\|X\|^2$ (observe that $\overline{w}_t$ is the mean of w_t^l using the r_c for $c \in C$),

$$\mathbb{E}\sum_{l=1}^{N} p_l \|\overline{w}_t - w_t^l\|^2 \leq \mathbb{E}\sum_{l=1}^{N} p_l \|(w_t^l - \overline{w}_{t_0})\|^2$$

Using Cauchy-Schwarz Inequality for sequences i.e. $||\sum_{i=1}^{E} a_i||^2 \leq E\sum_{i=1}^{E} ||a_i||^2$ and $w_t^l =$

$\overline{w}_{t_0} + \sum_{t=t_0}^{t-1} \nabla F_l^{\xi}(w_t^l)$.

$$\mathbb{E}\sum_{l=1}^{N} p_l \|\overline{w}_t - w_t^l\|^2 \leq \sum_{l=1}^{N} p_l \sum_{t=t_0}^{t-1} (t-t_0)\eta_t^2 \mathbb{E}\|\nabla F_l^{\xi}(w_t^l)\|^2 \leq \sum_{l=1}^{N} p_l \sum_{t=t_0}^{t-1} (E-1)\eta_t^2 \mathbb{E}\|\nabla F_l^{\xi}(w_t^l)\|^2 \leq \sum_{l=1}^{N} p_l \eta_{t_0}^2 (E-1)^2 G^2 \leq 4\eta_t^2 (E-1)^2 G^2 \quad \square \tag{10}$$

Lemma 3. Under Assumption 1 and 2, If $\eta_t \leq \frac{1}{4L}$ then,

$$\mathbb{E}\|\overline{v}_{t+1} - w^*\|^2 \leq (1-\mu\eta_t)\mathbb{E}\|\overline{w}_t - w^*\|^2 + \eta_t^2 \mathbb{E}\|g_t - \overline{g}_t\|^2 - \frac{1}{2}\eta_t \mathbb{E}(F(\overline{w}_t) - F^*) + 2\eta_t L \sum_{l=1}^{N} p_l \mathbb{E}\|\overline{w}_t - w_t^l\|^2$$

Proof. Since $\overline{v}_{t+1} = \overline{w}_t - \eta_t g_t$. We have,

$$\mathbb{E}\|\overline{v}_{t+1} - w^*\|^2 = \mathbb{E}\|\overline{w}_t - \eta_t g_t - w^*\|^2 \tag{11}$$

Subtracting and adding $\eta_t \overline{g}_t$ in Eq. (11)

$$\begin{aligned}\mathbb{E}\|\overline{v}_{t+1} - w^*\|^2 &= \mathbb{E}\|\overline{w}_t - \eta_t g_t - w^* - \eta_t \overline{g}_t + \eta_t \overline{g}_t\|^2 \\ &= \mathbb{E}\|\overline{w}_t - w^* - \eta_t \overline{g}_t\|^2 + \eta_t^2 \mathbb{E}\|g_t - \overline{g}_t\|^2 \\ &\quad + 2\eta_t \mathbb{E}|\langle \overline{w}_t - w^* - \eta_t \overline{g}_t, \overline{g}_t - g_t\rangle|\end{aligned} \tag{12}$$

Since $\mathbb{E}|g_t| = \overline{g}_t$ by our Assumption 5 (see Equation 6), the last term in Eq. (12) equates to 0 i.e. $2\eta_t \mathbb{E}|\langle \overline{w}_t - w^* - \eta_t \overline{g}_t, \overline{g}_t - g_t\rangle| = 0$. We only need to focus on bounding the first two terms of Eq. (12). Then, for the first term we have:

$$\|\overline{w}_t - w^* - \eta_t \overline{g}_t\|^2 = \|\overline{w}_t - w^*\|^2 + \eta_t^2 \|\overline{g}_t\|^2 - 2\eta_t \langle \overline{w}_t - w^*, \overline{g}_t\rangle \tag{13}$$

where, $\eta_t^2 \|\overline{g}_t\|^2 = \eta_t^2 \sum_{l=1}^{N} p_l \|\nabla F_l(w_t^l)\|^2$. Due to L-smoothness of F_l,

$$\|\nabla F_l(w_t^l)\|^2 \leq 2L\left(F_l(w_t^l) - F_l^*\right)$$

Then,

$$\eta_t^2 \|\overline{g}_t\|^2 \leq 2L\eta_t^2 \sum_{l=1}^{N} p_l \left(F_l(w_t^l) - F_l^*\right) \tag{14}$$

Now, let us consider $-2\eta_t \langle \overline{w}_t - w^*, \overline{g}_t\rangle$, we know $\overline{g}_t = \sum_{l=1}^{N} p_l \nabla F_l(w_t^l)$ then we have,

$$\begin{aligned}-2\eta_t \langle \overline{w}_t - w^*, \overline{g}_t\rangle &= -2\eta_t \sum_{l=1}^{N} p_l \langle \overline{w}_t - w^*, \nabla F_l(w_t^l)\rangle \\ &= -2\eta_t \sum_{l=1}^{N} p_l \langle \overline{w}_t + w_t^l - w_t^l - w^*, \nabla F_l(w_t^l)\rangle \\ &= -2\eta_t \sum_{l=1}^{N} p_l \langle w_t^l - w^*, \nabla F_l(w_t^l)\rangle \\ &\quad -2\eta_t \sum_{l=1}^{N} p_l \langle \overline{w}_t - w_t^l, \nabla F_l(w_t^l)\rangle.\end{aligned} \tag{15}$$

Since $F_k(.)$ follows μ-strong convexity, then the first term in Eq. (15) can be written as:

$$\begin{aligned}&-\langle w_t^l - w^*, \nabla F_l(w_t^l)\rangle \\ &\leq -\left(F_l(w_t^l) - F(w^*)\right) - \frac{\mu}{2}\|w_t^l - w^*\|^2.\end{aligned} \tag{16}$$

Similar to [25], using $2\langle a, b\rangle \leq \eta_t \|a\|^2 + \eta_t^{-1}\|b\|^2$, for $\eta_t > 0$ for the second term of the inequality in Eq. (15). We get:

$$\begin{aligned}&-2\langle \overline{w}_t - w_t^l, \nabla F_l(w_t^l)\rangle \\ &\leq \frac{1}{\eta_t}\|\overline{w}_t - w_t^l\|^2 + \eta_t \|\nabla F_l(w_t^l)\|^2.\end{aligned} \tag{17}$$

Then, by smoothness of F and similar to Equation (14),

$$-2\langle \overline{w}_t - w_t^l, \nabla F_l(w_t^l)\rangle \leq \frac{1}{\eta_t}\|\overline{w}_t - w_t^l\|^2 + 2L\eta_t (F_l(w_t^l) - F_l^*). \tag{18}$$

Applying these expressions from Equations (14), (15), (16), (17) back in Equation (13), we get,

$$\begin{aligned}\|\overline{w}_t - w^* - \eta_t \overline{g}_t\|^2 &\leq \|\overline{w}_t - w^*\|^2 + 2L\eta_t^2 \sum_{l=1}^{N} p_l (F_l(w_t^l) - F_l^*) \\ &\quad -2\eta_t \sum_{l=1}^{N} p_l \left[(F_l(w_t^l) - F^*) + \frac{\mu}{2}\|w_t^l - w^*\|^2\right] + \\ &\quad \sum_{l=1}^{N} p_l \left[\|\overline{w}_t - w_t^l\|^2 + 2L\eta_t^2 (F_l(w_t^l) - F_l^*)\right].\end{aligned} \tag{19}$$

Applying the Jensen inequality and after rearranging the terms, we get:

$$\begin{aligned}\|\overline{w}_t - w^* - \eta_t \overline{g}_t\|^2 &\leq (1-\mu\eta_t)\|\overline{w}_t - w^*\|^2 + \sum_{l=1}^{N} p_l \|\overline{w}_t - w_t^l\|^2 \\ &\quad +4L\eta_t^2 \sum_{l=1}^{N} p_l (F_l(w_l^k) - F^*) - 2\eta_t \sum_{l=1}^{N} p_l (F_l(w_t^l) - F_l^*).\end{aligned} \tag{20}$$

From [27], we find that the term $4L\eta_t^2 \sum_{l=1}^{N} p_l (F_l(w^*) - F_l^*) - 2\eta_t \sum_{l=1}^{N} p_l (F_l(w_t^l) - F_l^*)$ is bounded by $6L\eta_t^2 \Gamma + \sum_{l=1}^{N} p_l \|w_t^l - \overline{w}_t\|^2$ with $\Gamma = F^* - \sum_{l=1}^{N} p_l F_l^*$ under the Assumptions 1-4.

Now we can put all the variables in Eq. (12). We get:

$$\mathbb{E}\|\overline{v}_{t+1} - w^*\|^2 \leq (1-\mu\eta_t)\mathbb{E}\|\overline{w}_t - w^*\|^2 + \eta_t^2 \mathbb{E}\|g_t - \overline{g}_t\|^2 + 6L\eta_t^2 \Gamma + 2\sum_{l=1}^{N} p_l \mathbb{E}\|\overline{w}_t - w_t^l\|^2 \quad \square$$

Theorem 1. Let L, μ, σ_l, T, G be defined as above. Then, if the Assumptions 1-5 hold, for $\kappa = \frac{L}{\mu}, \gamma = \max(8\kappa, E)$, the IPfedAvg with N devices satisfies the following:

$$\mathbb{E}[F(w_T) - F^*] \leq \frac{\kappa}{(\gamma + T)}\left(\frac{2Z}{\mu} + \frac{\mu(\gamma+1)}{2}\mathbb{E}\|w_1 - w^*\|^2\right)$$

where $Z = \sum_{l=1}^{N} p_l^2 \sigma_l^2 + 6L\Gamma + 8(E-1)^2 G^2$.

Proof. Irrespective of the number of iterations, we always find $\overline{w}_{t+1} = \overline{v}_{t+1}$. Let $\Delta_t = \mathbb{E}\|\overline{w}_t - w^*\|$ then from Lemma 3 we get,

$$\Delta_{t+1} = (1-\eta_t\mu)\Delta_t + \eta_t^2\mathbb{E}\|g_t - \overline{g}_t\|^2 + 6L\eta_t^2\Gamma + 2\sum_{l=1}^{N} p_l\mathbb{E}\|\overline{w}_t - w_t^l\|^2 \quad (21)$$

Using Lemma 1, we have $\mathbb{E}\|\overline{g}_t - g_t\|^2 \leq Z_0$ with $Z_0 = \sum_{l=1}^N p_l^2\sigma_l^2$. Also, from Lemma 2, we have $\mathbb{E}\left[\sum_{l=1}^N p_l\|\overline{w}_t - w_t^l\|^2\right] \leq 4\eta_t^2(E-1)^2G^2$. Putting these values in Eq. (21), we get

$$\Delta_{t+1} \leq (1-\eta_t\mu)\Delta_t + \eta_t^2 Z_0 + 6L\eta_t^2\Gamma + 8\eta_t^2(E-1)^2G^2$$

which implies $\Delta_{t+1} \leq (1-\eta_t\mu)\Delta_t + \eta_t^2 Z$ where $Z = Z_0 + 6L\Gamma + 8(E-1)^2G^2$.

For step size $\eta_t = \frac{\alpha}{t+\gamma}$, for some $\alpha > \frac{1}{\mu}, \gamma > 0$ so that $\eta_1 \leq \min\{\frac{1}{\mu}, \frac{1}{4L}\} = \frac{1}{4L}$ and $\eta_t \leq 2\eta_{t+E}$. Similar to the convergence proof in [27], we will also prove $\Delta_t \leq \frac{b}{\gamma+t}$ where $b = \max\{\frac{\alpha^2 Z}{\alpha\mu-1}, (\gamma+1)\Delta_1\}$

We will prove this using induction over t. For $t=1$, it is easy to see that $\Delta_1 \leq \frac{b}{\gamma+1}$ to be true. We assume it is true for some t as well then for some $t+1$, we find:

$$\begin{aligned}
\Delta_{t+1} &\leq (1-\eta_t\mu)\Delta_t + \eta_t^2 Z \\
&\leq (1-\frac{\alpha\mu}{t+\gamma})\frac{b}{t+\gamma} + \frac{\alpha^2 Z}{(t+\gamma)^2} \\
&= \frac{t+\gamma-1}{(t+\gamma)^2}b + \left[\frac{\alpha^2 Z}{(t+\gamma)^2} - \frac{\alpha\mu-1}{(t+\gamma)^2}b\right] \\
&\leq \frac{t+\gamma-1}{(t+\gamma)^2}b
\end{aligned}$$

Since $\frac{a-1}{a^2} = \frac{a-1}{a^2-1+1} = \frac{a-1}{(a-1)(a+1)+1} \leq \frac{a-1}{(a-1)(a+1)} = \frac{1}{a+1}$. Then,

$$\Delta_{t+1} \leq \frac{b}{(t+\gamma+1)}$$

Hence, using induction we have proved $\Delta_t \leq \frac{b}{\gamma+t}$. Now, using L-smoothness of F and using $\Delta_t \leq \frac{b}{\gamma+t}$,

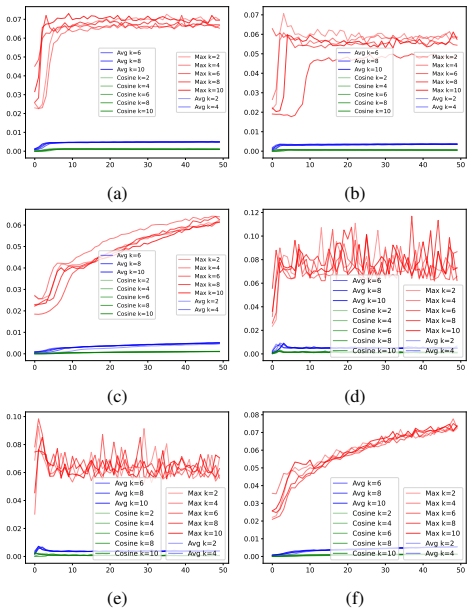
$$\mathbb{E}[F(\overline{w}_T) - F^*] \leq \frac{L}{2}\Delta_T \leq \frac{Lb}{2(\gamma+T)} \quad (22)$$

Now, for $\alpha = 2/\mu, \gamma = \max\{\frac{8L}{\mu}, E\} - 1, \kappa = L/\mu$. Then

$$\begin{aligned}
v &= \max\{\frac{\alpha^2 Z}{\alpha\mu-1}, (\gamma+1)\Delta_1\} \leq \frac{\alpha^2 Z}{\alpha\mu-1} + (\gamma+1)\Delta_1 \\
&\leq \frac{4Z}{\mu^2} + (\gamma+1)\Delta_1.
\end{aligned}$$

Putting these values in Eq. (22),

$$\mathbb{E}[F(\overline{w}_T) - F^*] \leq \frac{\kappa}{(\gamma+T)}\left(\frac{2Z}{\mu} + \frac{\mu(\gamma+1)}{2}\Delta_1\right). \quad \square$$



(a) (b) (c) (d) (e) (f)

Fig. 2: The average different distance measures (max, avg, cosine) between model weights in each round of communication for k-IPfedAvg under sevaral ks (2,4,6,8,10): (a) MNIST-iid (b) FashionMNIST-iid (c) CIFAR10-iid (d) MNIST-noniid (e) FashionMNIST-noniid (f) CIFAR10-noniid.

IV. Experimental Section

In this section, we present the experimental setup and analysis for k-IPfedAvg. In this work, we have simulated the FL environment on a local machine. We have randomly chosen 50 users and 50 communication rounds. In a given round of communication, each user trains the global model for 3 epochs on their local data and then communicate its model updates back to the server. The global model consists of two convolution layers (each with 10 filters and $(3,3)$ as kernel size) and a dense layer (32 neurons) as hidden layers. The input and output layers of the global model depends on the number of channels and output classes in each dataset. Table I shows the details of our experimental setup. We have compared the performance of k-IPfedAvg with baseline fedAvg [4] and DP-fedAvg [10]. To show the effectiveness of k-IPfedAvg, we have considered various ks $(2,4,6,8,10)$ against several noise multipliers $(0.2, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1)$ in DP-fedAvg (lower the noise multiplier, lower the noise addition and poorer the privacy).

We have shown our results on three benchmark datasets MNIST [29] (60k training and validation images, 10k test images), FashionMNIST [30] (60k training and validation images, 10k test images), and CIFAR10 [31] (50k training and validation images, 10k test images) to validate the performance of k-IPfedAvg. All three datasets have ten output classes. They

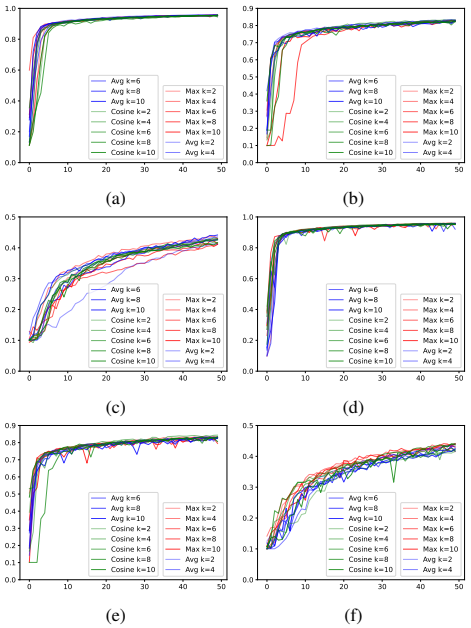


Fig. 3: The test accuracy in each round of communication for k-IPfedAvg under sevaral ks (2,4,6,8,10) using various distance measures (max, avg, cosine): (a) MNIST-iid (b) FashionMNIST-iid (c) CIFAR10-iid (d) MNIST-noniid (e) FashionMNIST-noniid (f) CIFAR10-noniid.

have been considered in the iid as well as non-iid manner to validate the performance in heterogeneous FL setting.

In k-IPfedAvg, as soon as the server receives the weight from the users, it clusters the users based on some distance measures. We considered three distance measures, namely Cosine, Avg and Max, to compare model weight. Consider two models M^i, M^j trained on i^{th}, j^{th} user's local data with L number of layers with $N_1, N_2, ..N_L$ number of neurons in them. Then, the three distance measures are:

Parameters	Values	Deescription
Users	50	Number of users in each round of communication
Global Server	1	Server aggregate the local models
Algorithms compared	3	fedAvg, k-IPfedAvg, DP-fedAvg
k in k-IPfedAvg	2,4,6,8,10	Determines the number of users in each cluster
Noise multiplier	0.2,0.4,0.5,0.6 0.7,0.8,0.9,1.0,1.1	Determines the amount of noise needed while training
Datasets	MNIST, CIFAR10, FashionMNIST	iid & non-iid distribution of these datasets
Local Epochs	3	Number of local training iterations in each round
Global rounds	50	Number of communications between server and uses.
Distance Measures	Cosine, Maximum, Average	Distance measure to compare two models.

TABLE I: Experimental setup.

1) Cosine $= \frac{M^i \dot{M}^j}{||M^i||||M^j||}$
2) Avg $= \frac{1}{L}\sum_{l=1}^{L}\frac{1}{N_l}\sum_{n=1}^{N_l}||M_{ln}^i - M_{ln}^j||_2^2$
3) Max $= \max_{l=1,..,L;n=1,..,N_l}||M_{ln}^i - M_{ln}^j||_2^2)$

The average distances in each round of communication for various values of k ($k = 2, 4, 6, 8, 10$) for the iid and non-iid distributions is given in Fig. 2. As expected, the Max distance measure which computes the maximum Euclidean distance between corresponding neurons in the same layer of two different models has the highest average distance between model weights. Fig. 3 presents the test accuracy score using the above mentioned distance measures. Although, all three distance measures have comparable test accuracy on the used datasets, a closer look suggests Max distance measure has the worst performance. Considering this, we chose the Max distance measure for further experiments to show that k-IPfedAvg performs as good as the baseline fedAvg while preserving privacy.

Fig. 4 shows the training accuracy over the number of communication rounds on iid and non-iid distributions of MNIST, FashionMNIST and CIFAR-10 datasets. A closer look at Fig. 4 suggests that higher k has marginally negative impact on the accuracy of the global model i.e. k-IPfedAvg's performance does not degrade much with improvement in the privacy. On the other hand, DP-fedAvg's performance drops significantly with the increase in the noise in the noise multiplier i.e. DP-fedAvg's performance degrades significantly with an increase in the privacy level. In case of DP-fedAvg, the perturbation during training affects its performance, the higher the privacy the poorer the performance as can be clearly seen in the CIFAR10 case (see Fig. 4c and Fig. 4f).

Fig. 5 shows the training loss for k-IPfedAvg, fedAvg and DP-fedAvg. Here as well, even with various values of k, k-IPfedAvg's training loss overlaps with the training loss of fedAvg and outperforms its DP counterparts with various noise multipliers. DP-fedAvg's training goes haywire in case of CIFAR10 (see Fig. 5c and Fig. 5f). We can observe the similar trend in Fig. 6 which shows the test accuracy of k-IPfedAvg, fedAvg and DP-fedAvg. k-IPfedAvg has baseline comparable test accuracy as well while its counterpart DP-fedAvg has poorer performance with an increase in the privacy level.

From Fig. 4, 5, 6 we can see some small accuracy drops specially for non-iid distribution of datasets (see for FashionMNIST-noniid and CIFAR10-noniid results). This is probably due to poor selection of the model weights during training. Further analysis required to overcome this gap.

V. Conclusion

In this paper, we have presented a novel k-Anonymous integrally private federated average algorithm (k-IPfedAvg) which protects the identity disclosure of the clients participating in the training. In k-IPfedAvg, the server clusters the user weights based on the privacy parameter and randomly selects one weight from each cluster randomly to protect the identity disclosure of the participating user. We have also presented convergence analysis of k-IPfedAvg. Just like fedAvg, k-IPfedAvg also has convergence rate of $\mathcal{O}(\frac{1}{T})$, where T represents the total number training epochs. Through

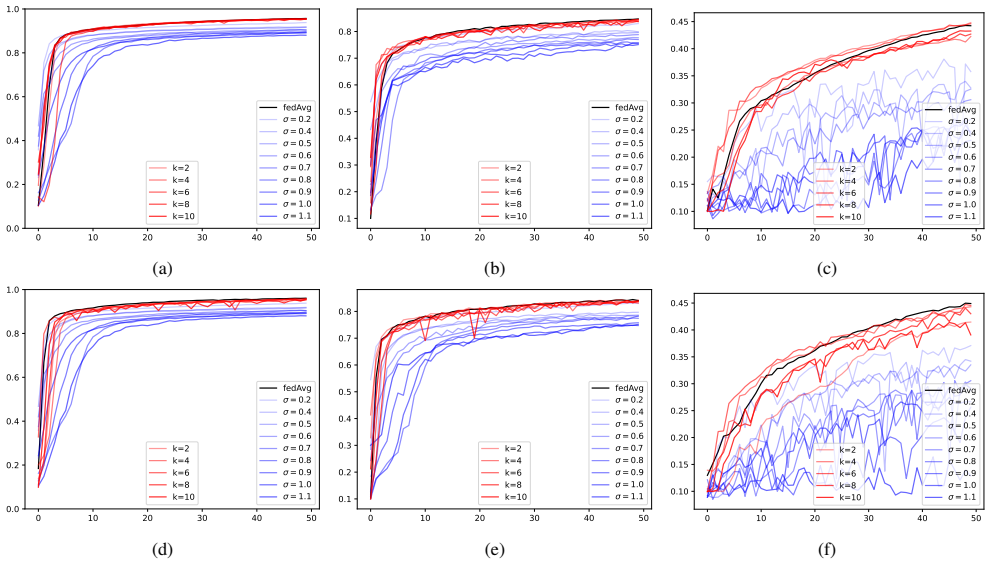


Fig. 4: The training accuracy of k-IPfedAvg under sevaral ks (2,4,6,8,10), fedAvg and DP-fedAvg under several noise multiplier (0.2, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1) for: (a) MNIST-iid (b) FashionMNIST-iid (c) CIFAR10-iid (d) MNIST-noniid (e) FashionMNIST-noniid (f) CIFAR10-noniid.

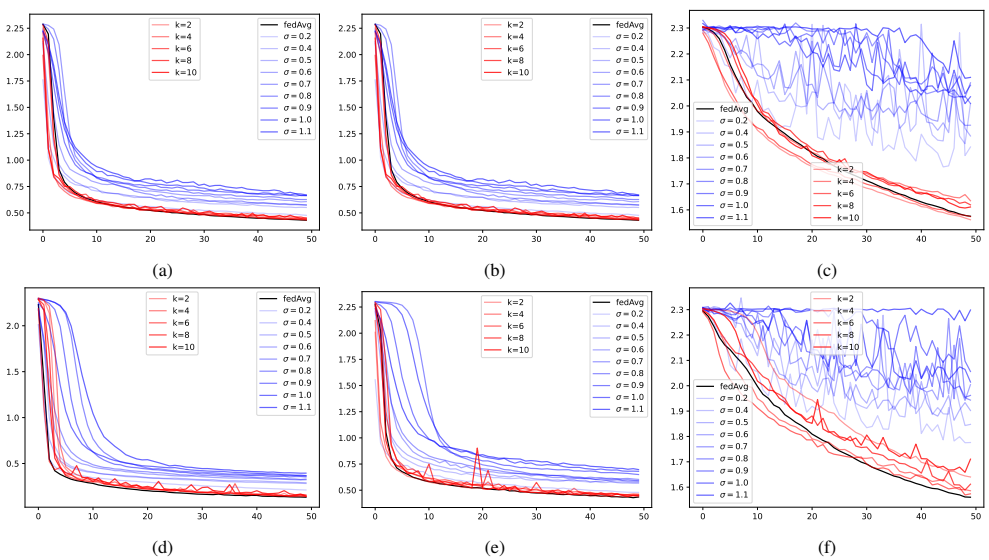


Fig. 5: The training loss of k-IPfedAvg under sevaral ks (2,4,6,8,10), fedAvg and DP-fedAvg under several noise multiplier (0.2, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1) for: (a) MNIST-iid (b) FashionMNIST-iid (c) CIFAR10-iid (d) MNIST-noniid (e) FashionMNIST-noniid (f) CIFAR10-noniid.

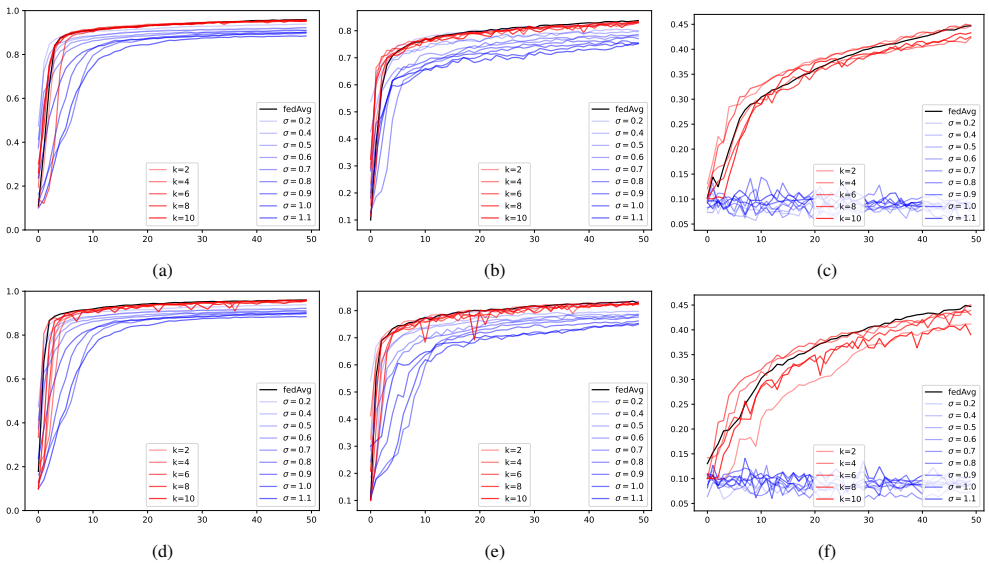


Fig. 6: The test accuracy of k-IPfedAvg under sevaral ks (2,4,6,8,10), fedAvg and DP-fedAvg under several noise multiplier (0.2, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1) for: (a) MNIST-iid (b) FashionMNIST-iid (c) CIFAR10-iid (d) MNIST-noniid (e) FashionMNIST-noniid (f) CIFAR10-noniid.

rigorous experimental analysis, we find that k-IPfedAvg has comparable accuracy score with fedAvg for iid as well as non-iid distributions of MNIST, FashionMNIST and CIFAR10 datasets. On the other hand, it performs significantly better than its DP counterparts with various levels of noise.

Our methodology has marginal effect of privacy parameter on utility but may have small accuracy drops because of the poor randomly chosen model(s). An interesting future direction can be to avoid such drops between communications. The k-IPfedAvg uses fedAvg as baseline, but can be used with other aggregation algorithms for federation such as fedProx [32]. Another interesting direction can be personalization [33] in k-IPfedAvg.

Acknowledgment

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by the supercomputing resource Berzelius provided by National Supercomputer Centre at Linköping University and the Knut and Alice Wallenberg foundation.

References

[1] S. Garg, S. Kumar, and P. K. Muhuri, "A novel approach for covid-19 infection forecasting based on multi-source deep transfer learning," *Computers in Biology and Medicine*, vol. 149, p. 105 915, 2022.

[2] M. L. De Prado, *Advances in financial machine learning*. John Wiley & Sons, 2018.

[3] Ö. YAVUZ, "An optimization focused machine learning approach in analysing arts participative behavior with fine arts education considerations," *International Scientific and Vocational Studies Journal*, vol. 5, no. 2, pp. 241–253, 2022.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[5] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7232–7241, 2021.

[6] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, IEEE, 2019, pp. 739–753.

[7] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*, PMLR, 2020, pp. 2938–2948.

[8] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression," 1998.

[9] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[10] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[11] A. Varshney and V. Torra, "Integrally private model selection for deep neural networks," *Database and Expert Systems Applications. DEXA 2023*, vol. 14147, 2023.

[12] A. Varshney and V. Torra, "Concept drift detection using ensemble of integrally private models," *European Conference on Machine Learning. ECML 2023*, 2023.

[13] A. Thudi, H. Jia, I. Shumailov, and N. Papernot, "On the necessity of auditable algorithmic definitions for machine unlearning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 4007–4022.

[14] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[15] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE conference on computer communications*, IEEE, 2019, pp. 2512–2520.

[16] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 3–es, 2007.

[17] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd international conference on data engineering*, IEEE, 2006, pp. 106–115.

[18] M. Fisichella, G. Lax, and A. Russo, "Partially-federated learning: A new approach to achieving privacy and effectiveness," *Information Sciences*, vol. 614, pp. 534–547, 2022.

[19] M. Asad, M. Aslam, S. F. Jilani, S. Shaukat, and M. Tsukada, "Shfl: K-anonymity-based secure hierarchical federated learning framework for smart healthcare systems," *Future Internet*, vol. 14, no. 11, p. 338, 2022.

[20] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[21] E. Cyffers and A. Bellet, "Privacy amplification by decentralization," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2022, pp. 5334–5353.

[22] A.-T. Tran, T.-D. Luong, J. Karnjana, and V.-N. Huynh, "An efficient approach for privacy preserving decentralized deep learning models based on secure multi-party computation," *Neurocomputing*, vol. 422, pp. 245–262, 2021.

[23] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE access*, vol. 10, pp. 22 359–22 380, 2022.

[24] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, *et al.*, "Anonymizing data for privacy-preserving federated learning," *arXiv preprint arXiv:2002.09096*, 2020.

[25] S. U. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[26] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local sgd on identical and heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 4519–4529.

[27] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[28] V. Torra, G. Navarro-Arribas, and E. Galván, "Explaining recurrent machine learning models: Integral privacy revisited," in *International Conference on Privacy in Statistical Databases*, Springer, 2020, pp. 62–73.

[29] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[30] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[31] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[32] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[33] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Ayush K. Varshney received his B.Sc. (hons) degree from the University of Delhi in 2015; the M.Sc. degree from South Asian University in 2018.

He is currently a PhD student with the Department of Computing Sciences, Umeå University. His main interests include Data Privacy, Machine Learning, and Decision Making

Vicenç Torra is currently a WASP professor on AI at Umeå University (Sweden). He is an IEEE and EurAI Fellow, and ISI elected member. His fields of interests include approximate reasoning (fuzzy sets, fuzzy measures/non-additive measures and integrals), decision making, and data privacy.

He has written seven books including "Modeling decisions" (with Y. Narukawa, Springer, 2007), "Data Privacy" (Springer, 2017), and "Scala: from a functional programming perspective" (Springer, 2017). He is founder and editor of the Transactions on Data Privacy, and started in 2004 the annual conference series Modeling Decisions for Artificial Intelligence (MDAI). His web page is: http://www.mdai.cat/vtorra.