Neurocomputing 624 (2025) 129476

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A defense mechanism against label inference attacks in Vertical Federated Learning

Marco Arazzi [a], Serena Nicolazzo [b,*], Antonino Nocera [a]

[a] Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Via A. Ferrata, 5, Pavia, 27100, PV, Italy
[b] Department of Computer Science, University of Milan, Via G. Celoria, 18, Milan, 20133, MI, Italy

ARTICLE INFO

Communicated by G. Yu

Keywords:
Federated learning
Vertical Federated Learning
VFL
Label inference attack
Knowledge distillation
k-anonymity

ABSTRACT

Vertical Federated Learning (VFL, for short) is a category of Federated Learning that is gaining increasing attention in the context of Artificial Intelligence. According to this paradigm, machine/deep learning models are trained collaboratively among parties with vertically partitioned data. Typically, in a VFL scenario, the labels of the samples are kept private from all parties except the aggregating server, that is, the label owner. However, recent work discovered that by exploiting the gradient information returned by the server to bottom models, with the knowledge of only a small set of auxiliary labels on a very limited subset of training data points, an adversary could infer the private labels. These attacks are known as label inference attacks in VFL. In our work, we propose a novel framework called KD*k* (knowledge distillation with *k*-anonymity) that combines knowledge distillation and *k*-anonymity to provide a defense mechanism against potential label inference attacks in a VFL scenario. Through an exhaustive experimental campaign, we demonstrate that by applying our approach, the performance of the analyzed label inference attacks decreases consistently, even by more than 60%, maintaining the accuracy of the whole VFL almost unaltered.

1. Introduction

Federated Learning (FL, for short) has emerged in the last years as a key technology enabling collaborative model training across different entities (typically, one aggregator server producing the global model and multiple local clients) without the need to gather data in a central location [1,2]. According to the different data partition strategies adopted, different main categories of FL have been formulated [3]. For instance, in Horizontal FL (HFL, for short) all parties hold the same attribute space but different sample space, whereas, Vertical FL (VFL, hereafter) is based on the collaborations among non-competing entities with vertically partitioned data that share overlapping data samples but differ in the feature space (i.e., a mobile phone company and a TV streaming service provider).

However, even if raw data are not shared due to the calculation and exchange of features, combining information between features and the possible presence of a compromised participant may raise privacy leakage [4–6]. Possible attacks that represent a significant concern in this context are *Label Inference Attacks*, because of the high sensitivity of the labels that may reveal crucial client information. In these types of attacks, an adversary tries to infer the labels of data points held by other participants based on the information exchanged during the FL process. This attack can compromise the privacy and security of sensitive data also in healthcare applications, raising concern that medical images and electronic health records containing sensitive patient information can be vulnerable [7]. In this scenario, various organizations frequently work together to improve diagnostic models and treatment methods that use VFL to collaboratively train machine learning models without the need to share their raw data [8]. In a VFL setup, different organizations possess distinct features for the same group of patients. For instance, one hospital might have demographic information, another could have medical imaging data, and a third one might have genetic information. A defense mechanism is needed to ensure that no institution can infer sensitive data from the model updates, thus maintaining patient confidentiality and complying with the privacy regulations on health data [9]. Another sector that can be impacted by such kind of threat is finance, where multiple institutions often collaborate to improve risk assessment models and fraud detection systems [10]. In this context, VFL enables the joint training of ML models without exchanging their raw data. For example, one bank might have a transaction history, another might have credit scores, and a third could have investment portfolios.

In our work, we start considering the label inference attacks to VFL described in the recent paper of Fu et al. [11] to provide a defense

* Corresponding author.
E-mail addresses: marco.arazzi01@universitadipavia.it (M. Arazzi), serena.nicolazzo@unimi.it (S. Nicolazzo), antonino.nocera@unipv.it (A. Nocera).

https://doi.org/10.1016/j.neucom.2025.129476
Received 8 September 2024; Received in revised form 3 January 2025; Accepted 16 January 2025
Available online 23 January 2025

against them. Our application scenario is the classical VFL scenario in which two typologies of participants, a server and a set of clients, collaboratively train an ML holding different feature spaces. In the following, we will refer to the server (or active participant) which stores the labels that are kept private from the other clients, and the clients (or passive participants). The adversary controls some passive participants and aims at discovering the private labels. Fu et al. [11] demonstrate that various types of label inference attacks are highly successful in this context, achieving strong accuracy results. They specifically find that the attacker can successfully carry out a label inference attack by leveraging *(i)* the trained local model held by the malicious participant and *(ii)* the received gradients of the loss function, which contain hidden information about the labels. They describe three possible categories of label inference attacks. The first attack is a *passive attack*, in which, with the help of some auxiliary labeled data, the malicious participant can fine-tune his/her trained bottom model to infer the labels in a semi-supervised manner. The second type is an *active attack*, in which the attacker tries to scale up the learning rate of her/his bottom model during the training phase to force the top model to rely more on her/his model thus boosting the label inference accuracy. The third type is a *direct attack* through which the adversary can infer labels by analyzing the sign of gradients from the server. In any case, the attacker exploits the feedback received from the loss function estimated on the top model, necessary to perform the backpropagation on the local model, to extract information on the labels.

Our proposal consists of designing a novel framework called KDk (Knowledge Discovery and k-anonymity) as a defense mechanism relying on an additional component for the server (or active) participant. This includes a *Knowledge Distillation* step and an *obfuscation* algorithm. Specifically, Knowledge Distillation (KD, hereafter) [12] is an ML compression technique that transfers knowledge from a larger teacher model to a smaller student model. The teacher network produces softer probability distributions instead of hard labels that can better capture essential features and relationships in the data.

Therefore, in the active participant, we include a teacher network whose outputs are soft labels. These are then processed by an algorithm based on the concept of k-anonymity [13] to add a further level of uncertainty. This step groups the k labels with the higher probabilities making it hard for the attacker to infer the most probable one. Then the top model of the server can be fed with these new soft and partly anonymized labels and the VFL tasks can be executed collaboratively.

Our defense strategy is based on the main intuition that, an attacker, located on one of the clients, can infer labels from server-returned gradients only if there is a strong correlation between the gradients and the labels. Some of the existing approaches, such as the *active attack* proposed in [11], exploit more this relationship by leveraging the learning rate to amplify the strength of this correlation, thereby enhancing the signal returned by the server. To contrast this situation, our approach employs an obfuscation technique designed to weaken the correlation between the gradients and the labels. However, to maintain the performance of the final global model, such an obfuscation strategy must be carefully designed and informed by the likelihood of different labels being associated with each other. To achieve this objective, we utilize knowledge distillation, ensuring that the blurring strategy is effective in protecting label information while also maintaining the accuracy of the global model.

In summary, our framework represents an essential advancement for *(i)* safeguarding sensitive information that may be revealed or exploited by performing label inference attacks, *(ii)* building trust in collaborative environments and reassuring participants that their data contributions remain confidential and secure, fostering cooperation and *(iii)* enhancing model security by protecting against label inference attacks, which can also stepping stone for more severe attacks.

Our experimental campaign demonstrates that using our approach the accuracy of the three types of label inference attacks decreases significantly. The source code of our defense along with the setting to replicate our experiments are publicly available at https://anonymous.4open.science/r/KDK_Anonymous-CADB.

In summary, the main contributions of this paper are:

- We design a countermeasure for the different types of label inference attacks proposed by [11].
- We conduct an experimental campaign to demonstrate that the accuracy of all the analyzed types of label inference attacks consistently decreases if our complete approach is applied.
- We provide a comparison with existing defense strategies and show the higher effectiveness of our solution.

The organization of this paper is outlined as follows. Section 2 describes the main works related to our approach. Section 3 delves into the details about FL, k-anonymity, and Knowledge Distillation that are essential to the understanding of our solution. Section 4 presents the types of label inference attacks against which we provide a defense. Section 6 discusses the experimental campaign, including the setup and results of our defense mechanisms. Ultimately, Section 7 concludes the work and presents possible future directions.

2. Related work

Recent works have shown that FL is vulnerable to multiple types of inference attacks, such as membership inference, property inference, and feature inference [14–16]. The objective of a membership inference is to discriminate whether a specific record is in a party's training dataset or not. Nevertheless, this type of attack has no reason to exist in VFL as every participant knows all the training sample IDs. Property inference aims to extract some properties about a party's training dataset, which are uncorrelated to the training task. In a feature inference attack, instead, a party tries to recover the samples used in another party's training dataset. For example, Luo et al. propose a feature inference attack for VFL [17], in which the active party tries to infer the features owned by the passive party. However, the authors strongly assume that the active party knows the model parameters of the passive party, which is difficult to achieve in real-world scenarios. Unlike the works cited above, our proposal deals with a different type of inference attack in VFL, known as a label inference attack, conducted by the passive party and aimed at leaking the labels owned by the active participant. Since labels often contain highly sensitive information, this type of attack deserves more and more attention. Although finding possible defense strategies against these attacks is crucial, they are still an open challenge and only a few efforts have been made.

For example, the articles [18,19] study label inference attacks in VFL, but they specifically focus on split learning scenarios. In particular, [18] formalizes a threat model for label leakage in two-party split learning in the context of binary classification and proposes a countermeasure based on random perturbation techniques that minimize the amount of label leakage of a worst-case adversary. However, the proposal in [19] presents a passive clustering label inference attack for split learning, in which the adversary (which can be any client or the server) retrieves the private labels by collecting the exchanged gradients and smashed data both during and after the training phase. [20] design the inversion and replacement attacks to reveal private labels from batch-level messages in a VFL whose communication is protected by a Homomorphic Encryption mechanism and a confusional autoencoder (CoAE) method as a possible countermeasure. The proposal in [21] deals with the design of a label leakage attack from the forward embedding in two-party split learning and a corresponding defense that reduces the distance correlation between the cut layer embedding and private labels. Kholod et al. [22] propose a parallelization method to decrease data transmission, and, consequently, both the learning cost and privacy leakage risk. A framework called LabelGuard has been designed in [23] to defend against label inference attacks via a cascade VFL algorithm through a minimization of the VFL task training loss.

Table 1
Summary of the acronyms used in the paper.

Symbol	Description
DL	Deep Learning
FCNN	Fully Connected Neural Network
FL	Federated Learning
FTL	Federated Transfer Learning
GC	Gradient Compression
HFL	Horizontal Federated Learning
KD	Knowledge Distillation
ML	Machine Learning
NG	Noise Gradient
OA	Original Architecture
PPDL	Privacy-Preserving Deep Learning
VFL	Vertical Federated Learning

In this work, we start from the proposals [11]. The authors of [11] describe three kinds of label inference attack, i.e., passive label inference attack, active label inference attack, and direct label inference attack, for VFL. Adversaries could infer the labels of the active party from both the received plaintext gradients and obtained plaintext final model weights. Although these attacks are very effective, they make a strong assumption on auxiliary labels that have to be held for the adversary.

3. Background

In this section, we describe some concepts useful to understand our approach. In particular, we examine the key aspects and different categories of Federated Learning, we recall the concept of k-anonymity and we delve into the analysis of the main features of Knowledge Distillation.

Table 1 summarizes the acronyms used in this paper.

3.1. Federated learning

As stated in the Introduction, FL is a machine-learning method designed to train a model in a distributed manner across different devices holding private local data samples. The actors of this protocol are C devices ("clients", hereafter), running local training and holding private data; and a central server called "aggregator", that coordinates the whole FL process aggregating the local updates. Specifically, FL aims to train a global model $\mathbf{w}$ by uploading the weights of local models $\{\mathbf{w}^i | i \in C\}$ to a parametric server optimizing a loss function:

$$\min_{\mathbf{w}} l(\mathbf{w}) = \sum_{i=1}^{n} \frac{s_i}{C} L_i(\mathbf{w}^i) \tag{1}$$

where $L_i(\mathbf{w^i}) = \frac{1}{s_i} \sum_{j \in I_i} l_j(\mathbf{w}^i, x_i)$ is the loss function, s_i is the local data size of the i-th client, and I_i identifies the set of data indices with $|I_i| = s_i$, and x_j is a data point. The basic FL workflow [24] is shown in Fig. 1.

FL can be classified into different scenarios according to the data partition strategies adopted [3].

In this paper, we target **Vertical FL** (VFL, for short) or feature-based FL, explicitly. According to this FL scheme, the involved entities own local datasets with overlapping data samples but different feature spaces. VFL can be with or without model splitting. In the presence of model splitting, every client runs a bottom (or local) model without sharing the entire model with other participants and relying on features locally available at each party. The final top (or global) model is reconstructed by a server that combines the locally trained model portions to compute a final output.

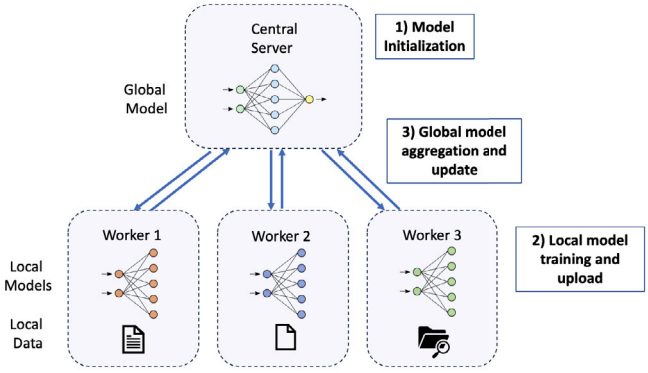

Fig. 1. The Federated Learning workflow.

3.2. k-anonymity

The concept of k-anonymity, first described in [13], represents one foundational principle in database theory for privacy-preserving data publishing. It aims to safeguard the anonymity of the individuals' data by ensuring that each record in the dataset is indistinguishable from at least $k-1$ other records. Several procedures can be applied to attributes to obtain k-anonymity, such as:

- Suppression, which implies removing or cleansing certain information.
- Generalization replaces distinctive values with more general ones (e.g., substituting exact ages with age ranges).

3.3. Knowledge Distillation

Knowledge Distillation (KD, for short) is an ML model compression technique, in which the knowledge from a complex model, or "teacher" model, is transferred to a smaller and more efficient model, known as the "student" model without a significant drop in accuracy [12]. The general idea was first presented by Bucilua et al. in 2006 [25] and modeled in its current known form in 2014 by Hinton et al. [26] who found it easier to train a classifier using the outputs of another classifier as target values than using actual ground-truth labels. The teacher network outputs are represented by the so-called soft probabilities that contain more information about a data point than just the class label (or hard predictions) and are the input of the student network.

In practice, given an input x the teacher network produces a vector of scores $s^t_x = [s^t_1, s^t_2, \ldots, s^t_K]$ that are converted into probabilities:

$$p^t_k(x) = \frac{e^{s^t_k}}{\sum_j e^{s^t_j}} \tag{2}$$

Hinton et al. [26] proposed to modify these probabilities in *soft probabilities* as following:

$$p^t_k(x) = \frac{e^{s^t_k}/\tau}{\sum_j e^{s^t_j}/\tau} \tag{3}$$

where τ is a hyperparameter. A student network will produce a softened class probability distribution, $\tilde{\mathbf{p}}^s(x)$. The loss for the student network is a linear combination of the cross entropy loss, namely $\mathcal{L}_{cl}$ and a knowledge distillation loss $\mathcal{L}_{KD}$:

$$\mathcal{L} = \alpha \mathcal{L}_{cl} - (1-\alpha)\mathcal{L}_{KD} \tag{4}$$

where $\mathcal{L}_{KD} = -\tau^2 \sum_k \tilde{p}^t(x) \log \tilde{p}^s(x)$ and α and τ are hyperparameters.

Fig. 2 shows the generic architecture of the KD using the teacher–student model. Thanks to the distillation algorithm the student mimics the teacher network learning the relationship between different classes discovered by the teacher model that contains information beyond the ground truth labels.

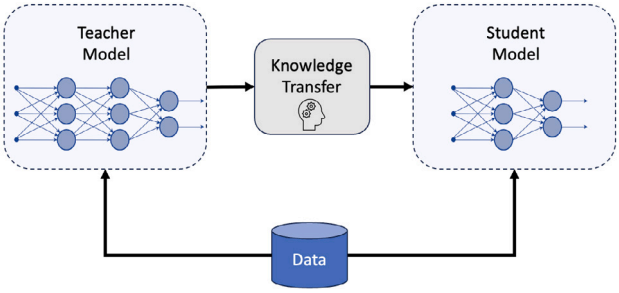

Fig. 2. Generic architecture of knowledge distillation using a teacher–student model.

4. Label inference attacks

In this section, we describe the most common label inference attacks against VFL, which we focus on when designing our defense strategy. As typically done in the related literature, we make explicit reference to the more complex scenario in which VFL is combined with model splitting [27] (see Section 3).

In this setting, as originally proposed by [11], label inference attacks are carried out by adversaries, controlling one or more of the bottom models, which aim to infer the private labels for any samples in the dataset. Recall that, according to the model splitting paradigm, only the active party, i.e., the server, has the classification layer, whose objective is the prediction of the correct label for each datapoint in input. Therefore, labels are available only to this active party of the FL system and, therefore, are considered sensitive information. To carry out a label inference attack, adversaries can mainly exploit two main aspects of VFL that, according to [11], may generate label leakage, namely: *(i)* the trained local model that is under the full control of the malicious participant; or *(ii)* the received gradients of the loss that contain hidden information about labels.

In the following, we describe four main types of label inference attacks, namely: *(i)* Passive Label Inference attack; *(ii)* Active Label Inference attack; and *(iii)* Direct Label Inference Attack [11].

4.1. Passive Label Inference Attack

Adversaries can perform this attack by exploiting their locally owned bottom model. It is referred to as *passive* because the malicious participant does not perform any active action during the training or inference phase, but she/he remains *honest but curious*. This type of attack assumes that the adversarial can rely on a few auxiliary labeled data (in [11] only the 0.08% of the labeled training samples have been used as auxiliary labels in the experimental campaign). If the attacker can get this additional knowledge, she/he can infer the labels by fine-tuning her/his bottom model through a further classification layer in a semi-supervised manner. This step is referred to as *model completion* attack. Once the training is completed, the model can predict a label for every item of the sample of the adversary.

4.2. Active Label Inference Attack

This attack is classified as *active* because the malicious participant performs some actions in the training stage, in particular, she/he tries to scale up the learning rate during the training phase of her/his bottom model. In this way, she/he aims to accelerate the gradient descent on her/his bottom model to submit better features to the server in each iteration. Consequently, she/he can force the top model to rely more on her/his bottom model than the other participants. Since increasing the learning rate does not always result in a more efficient gradient descent, the authors of [11] perform this attack by designing and executing a malicious local optimizer. This component adaptively scales up the gradient of each parameter in the adversary's bottom model to avoid the oscillation phenomenon around the local minimum point, which is typical of the use of an overly large learning rate for gradient descent.

Using the malicious local optimizer, the attacker can get a trained bottom model with more hidden information about labels. In addition, she/he can perform the model completion step of the passive attack (see Section 4.1) to fine-tune the bottom model with an additional classification layer and obtain the final label inference model.

4.3. Direct label inference attack

In this attack, the target architecture is slightly modified by removing the top model and directly using the summed outputs of the bottom models, sized to match the number of classes in the given task, in the loss calculation to obtain the desired gradients on the partial embeddings. To carry out this attack the adversary directly exploits the gradients she/he receives from the active party to infer the labels of the training examples. This is based on the analysis of the signs of the gradients of the losses. The authors of [11] demonstrate through mathematical proof that this method works for label inference in VFL without model splitting (see Section 3.1 for details about model splitting).

Since no gradients are available at the inference time, with this attack, the malicious participant can only infer the labels of training examples. Nevertheless, these discovered labels can be used as the auxiliary data necessary to perform a passive label inference attack. In this way, the attacker can infer the label of an arbitrary sample.

Still in this context, another attack strategy has been also described in [28]. Both attacks utilize a similar configuration, omitting the top model and relying solely on the output of the bottom model, which corresponds to the number of classes. Differently from [11], the attack described in [28] seeks to match the gradients provided by the active party with synthetic gradients generated by the attacker. This process aims to replicate the training process and leak label information. Considering that both methods operate within the same architectural scenario and exploit the gradients returned from the active party, the approach in [11] achieves exceptional performance reaching 100% accuracy across all benchmark datasets, therefore we target it as the representative attack for gradient-based analysis in this paper.

5. Approach description

Our approach aims to provide a countermeasure for all the types of label inference attacks described in Section 4.

To better present our defense strategy, as done once again in [11], we will focus on a basic VFL attack scenario shown in Fig. 3. Here, two participants holding the same set of samples but with features from different spaces want to train a model collaboratively through VFL.

The first participant, that is the server, runs both the top T_M and the bottom B_A models, hence it is the label owner HL and holds part of the vertically partitioned data X_A. For this reason, it is also referred to as an *active* participant. Its objective is to enhance the model performance by combining its features with the ones of other entities coming from different business domains.

The second party in our example is the adversarial, that is a *passive* participant or client who aims at inferring the labels from the training process and has access only to its bottom model B_P and its part vertically partitioned data X_P. At each training round, the bottom model outputs $H = \{H_A, H_P\}$ are sent to the server running the top model T_M, which, hence, returns the correspondent partial gradients ∇H_A and ∇H_P of the loss l. These are used to update the clients' bottom model parameters W_A (W_P). The local models updates ∇W_A and ∇W_P are calculated as follows (CE = cross-entropy, SM = *softmax*):

$$H = CONCAT(H_A, H_P), preds = T_M(H) \quad (5)$$
$$l = CE(SM(preds),\ SM(HL))$$

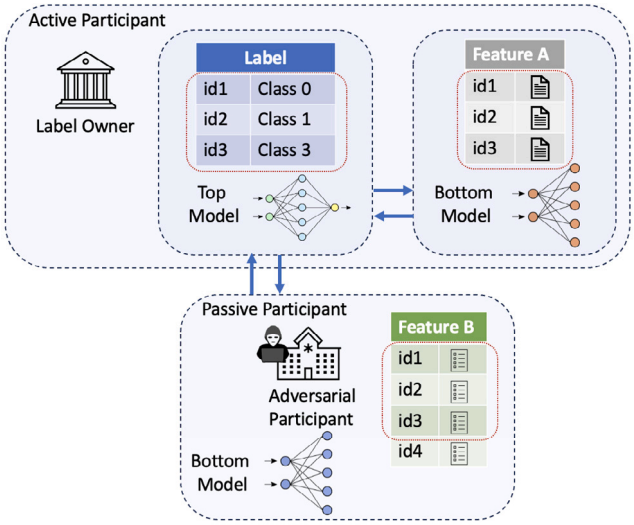

Fig. 3. Label inference attack scenario against VFL.

$$\nabla W_A = \sum \frac{\partial l}{\partial H_A} \cdot \frac{\partial H_A}{\partial B_A}. \tag{6}$$

$$\nabla W_P = \sum \frac{\partial l}{\partial H_P} \cdot \frac{\partial H_P}{\partial B_P}. \tag{7}$$

Although the private labels HL never leave the first participant's storage, the adversary can exploit the received partial gradients and the trained bottom model to conduct a label inference attack.

In particular, to perform the first attack, or Passive Label Inference Attack, the adversary relies on a small set of auxiliary labels. If she/he manages to obtain this set she/he can fine-tune her/his bottom model through a further classification layer in a semi-supervised manner to infer the training labels. To conduct the other attacks (i.e., the Active and the Direct Label Inference Attacks), instead, the malicious participant exploits the fact that, even though she/he does not have direct access to the label, her/his bottom model implicitly holds information about them, because of the training step. With these last strategies, the adversary cannot obtain all the private labels, but he/she can, then, run a subsequent passive attack to improve the attack performance.

At this point, we are ready to present our defense mechanism against the above-cited types of label inference attacks. In particular, we include in the active participant architecture an additional component comprised of a fine-tuned teacher network that performs a Knowledge Distillation KD step to output soft labels SL instead of hard ones HL. The output vector of a given data point contains the probabilities that it belongs to each class represented by the private labels. The output from this layer is then processed by an algorithm based on the concept of k-anonymity (see Section 3.2 for detail) to add a second level of uncertainty. Through this further step, instead of selecting a single label for each sample, we select a set of k labels in SL with the highest probability. Hence, as shown in Algorithm 1, if the label is the one associated with the highest confidence it is scaled by ϵ (where ϵ is a *smoothing* parameter), otherwise, if the label belongs to the $k-1$ highest probability labels (excluding the maximum) a $\epsilon/(k-1)$ factor is applied to scale up their final probability value. At this point, since the correct label for each item is obfuscated in a group of k labels, as we will demonstrate in the experiments, the attacker can no longer easily infer the most probable one performing any of the above-cited attacks. The VFL process changes as follows:

$$SL \leftarrow KDk(HL, k, \epsilon) \tag{8}$$

$$KDk(HL, k, \epsilon) = \begin{cases} 1-\epsilon & \text{if } L_i \in \max(KD(HL)) \\ \frac{\epsilon}{k-1} & \text{if } L_i \in top_k(KD(HL)) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$l_{kdk} = CE(SM(preds), SM(SL)) \tag{10}$$

$$\nabla W_P = \sum \frac{\partial l_{kdk}}{\partial H_P} \cdot \frac{\partial H_P}{\partial B_P}. \tag{11}$$

Here $KD(HL)$ contains the soft labels (a probability vector) returned by the knowledge distillation model for each datapoint in the original training set. The function $\max(KD(HL))$ returns the labels with the highest probability for each data point. Whereas, the function $top_k(KD(HL))$ returns the set of the $k-1$ labels having the highest values following the maximum (i.e., once again, the highest probabilities of being the correct label of the target datapoint as estimated by the KD model) for each data point.

Algorithm 1 Soft Label Algorithm.

Require:
HL: set of Hard Labels
K: set of top-k labels with higher confidence
k: $|K|$ cardinality of K
ϵ: smoothing parameter
n: number of classes
$TopKIndexes, MaxValue \leftarrow getTopKIndexes(HL,K)$
$SL \leftarrow zeros(n)$
for i in $TopKIndexes$ **do**
  if $HL[i] == MaxValue$ **then**
    $SL[i] \leftarrow 1-\epsilon$
  else
    $SL[i] \leftarrow \epsilon/(k-1)$
  end if
end for

5.1. Defense strategy analysis

This section focuses on explaining the core rationale behind the proposed defense KDk. Specifically, using soft labels instead of their one-hot encoded versions, the active client introduces a level of uncertainty into the base models of the passive participants. Although this strategy obfuscates the label information from potential attackers, it may also impact the overall effectiveness of the vertical federated model in the primary task. To mitigate this issue, Knowledge Distillation is employed to identify the most related classes, using a teacher model trained with data from the same feature space. In the field of Knowledge Distillation [26], this technique has been demonstrated to maintain or even enhance the performance of the student network. This dual-component approach helps incorporate noise into the attacker's base model while minimizing disruption to the federated model's ultimate accuracy.

Consider now the passive and active attacks mentioned above. In both cases, the attack strategy employs a semi-supervised approach to refine the bottom model by leveraging a limited number of auxiliary known labels and the pseudo-labels produced by the model itself. Our defense significantly influences the model's confidence in delivering accurate predictions since the bottom model is now trained only on our anonymized labels, thus leading to the following situation:

$$MC = B_P^{kdk} + FC, \; PL = MC(UD), \tag{12}$$

$$l = CE([KD, UD], [KL, PL]) \tag{13}$$

$$MC \leftarrow \sum \frac{\partial l}{\partial MC}, \; MC(UD) \rightarrow IfredL \tag{14}$$

Where UD and PL are respectively the unknown data and their associated pseudo-labels, KD, and KL are the known auxiliary data with labels and MC is the bottom model. MC is composed of a feature extraction part B_P^{kdk} trained with our anonymized labels and, then, an FC classification layer is added and trained. The attack strategy relies mainly on the confidence of the obtained bottom model, which is trained on our controlled and anonymized labels. As a matter of fact, for the unknown data, the pseudo-labels derived by the bottom model

are used to fine-tune the MC model in a semi-supervised manner. In particular, the availability of a small set of known data, for which the corresponding labels are available, is used to evaluate the cross-entropy between the labels of the known data and the ones generated by the bottom model (Eq. (13)). Now, the produced l is used to fine-tune the classification layer of the bottom model to make it capable of predicting the labels for the unknown data. However, the cross-entropy estimated in Eq. (13) is now compromised by the fact that PL is anonymized by KDk.

The direct attack, instead, relies on the sign of the gradients on the partial representation H_p as follows:

$$g_i^{\text{adv}} = \frac{\partial \text{loss}(x, c^{KDk})}{\partial y_i^{\text{adv}}} = -\frac{\partial \log e^{y_c^{\text{adv}}} - \partial \log \sum_j e^{y_j^{\text{adv}}}}{\partial y_i^{\text{adv}}} \tag{15}$$

$$= \begin{cases} -1 + \frac{e^{y_i^{\text{adv}}}}{\sum_j e^{y_j^{\text{adv}}}}, & \text{if } i = c, \\ \frac{e^{y_i^{\text{adv}}}}{\sum_j e^{y_j^{\text{adv}}}}, & \text{if } i \neq c. \end{cases} \tag{16}$$

Where g_i^{adv} is the logit corresponding to the ith label, while c^{KDk} is the soft label mapped with the ith data point by our defense. The attack looks at the sign of the logits and if $g_i^{\text{adv}} < 0$ the ith label is considered as the ground-truth target. However, since c^{KDk} is now controlled by KDk and is computed by using a soft label strategy, the fundamental premise of the attack is undermined. The significance of the primary labels is now spread among the secondary ones, thus leading to a change in the sign of their logit.

This analysis highlights that the main strength point for our defense relies on its capability of obfuscating and, hence, anonymizing the original labels. Therefore, no matter the attack variant, the defense will always be effective if the attack strategy tries to exploit the indirect knowledge acquired by the bottom model and reverse-engineer it to extract the original labels. As a matter of fact, by design, bottom models will not have access to usable information on the correct mapping between the classes and the data points, as this information is blurred in k-anonymous sets of equally probable soft labels.

5.2. Time complexity analysis

The time complexity of our defense is analyzed in this section. Our defense does not include any variation on the activities of passive clients; instead, the protection strategy is entirely executed on the active participant. As explained in the previous section, the main idea behind our approach is to replace the original labels for each data point of the training set, with a group of k labels. The selected k labels preserve the property of being highly plausible, each with a high probability of representing the specific data point. To meet this condition, a Knowledge Distillation (KD) step is carried out to transform the labels from hard to soft ones.

As visible in Algorithm 1, the loop is bounded by the value of k, which is constant and typically less than the total number of available classes; therefore, the main computational cost of our solution concerns only the KD step. This step typically requires the exploitation of a teacher network trained to output label probabilities for each data input. The training complexity of this network strictly depends on its architecture, which, on the other hand, is defined according to the type of target training data (e.g., images, textual data, audio, and so forth). However, this training step is not part of the federated learning task, as it can be carried out in advance and offline. Actually, in our solution, even a pre-trained teacher network can be successfully employed, since its only objective is to map each label to the probability that it belongs to the input data. To reach this goal, the teacher network has to be able to build rich latent representations (embeddings) for the input data points, which can be obtained with sufficiently large general-purpose pre-trained networks (for instance, a pre-trained ResNet-50 would be adequate to work with image data input, whereas a pre-trained BERT model can be successfully employed for textual data). Once the teacher network is available, our defense can be deployed and executed during an FL task. The teacher network is only exploited during the first epoch to obtain the soft labels for each data point. Observe that, importantly, this step can even be carried out as a pre-processing before the FL task is even started. This would reduce to practically almost zero the impact of our defense on the time complexity of the FL task. Estimating the inference cost of a teacher network can be complex, as it heavily depends on its specific design, which is typically tailored to the characteristics of the input data. To provide the reader with a sense of scale, we describe here a concrete example, allowing for an understanding of the order of magnitude involved. This gives a more tangible reference point, even though the actual cost can vary depending on the network's architecture and data. In particular, suppose the input data is composed of images. A possible teacher network can be obtained by relying on a pre-trained ResNet-50 network [29]. This network is based on convolution layers that represent the most relevant computations. The complexity for a single convolution layer in Big $\mathcal{O}$ notation is approximately:

$$\mathcal{O}(C_{in} \cdot C_{out} \cdot K \cdot H \cdot W)$$

where C_{in} and C_{out} are the input and output channels, respectively, K is the kernel size, H and W are the dimensions of the input. The kernel size is a constant, while the number of channels (both input and output) is typically much smaller than the input dimensions. Therefore, we might simplify the overall complexity of a ResNet-50 (i.e, with 50 layers) as bounded by $\mathcal{O}(50 \cdot H \cdot W)$, which for a squared image ($s \times s$) becomes $\mathcal{O}(s^2)$. This cost is then repeated for each training input data point, say N times. Hence, the overall cost is bounded by $\mathcal{O}(N \cdot s^2)$. In many real-life application contexts, the size of images can be considered negligible compared to the size of the overall input dataset. For instance, the tiny Imagenet dataset [30] contains 100,000 images, each of size 64×64. In this case, it is evident how the size of an image is almost negligible compared to the overall size of the dataset ($64 \times 64 = 4096 \ll 100,000$). Therefore, in this case, the overall cost can be considered bounded by a linear cost against the input dataset $\mathcal{O}(N)$. As a final remark, we observe that, since to obtain the soft labels, this cost is only required once at the beginning or right before the FL task, the cost introduced by our defense can be assumed negligible in most concrete application scenarios.

6. Experimental results

In this section, we illustrate the experiments carried out to assess the performance of our defense mechanism. Specifically, in Section 6.1, we describe the dataset, the evaluation metrics, and the environment used for our experiments. The remaining sections are devoted to analyzing the results and the performance of our defense approach against the different types of analyzed label inference attacks and the comparison with other defense mechanisms.

6.1. Testbeds description

To evaluate the robustness of our approach against label inference attacks we adopt some of the datasets used by [11], namely:

- CIFAR-10 dataset [31] consisting of 60,000 32×32 color images divided into 10 classes with 6000 images per class. There are 50,000 in training images and 10,000 in test images.
- CIFAR-100 [31] dataset that is similar to CIFAR-10, but it has 100 classes containing 600 images each with 500 training images and 100 testing images per class.
- CINIC-10 [32], which is a large dataset and an extended alternative for CIFAR-10 with 270,000 images, (i.e., 4.5 times more that of CIFAR-10).

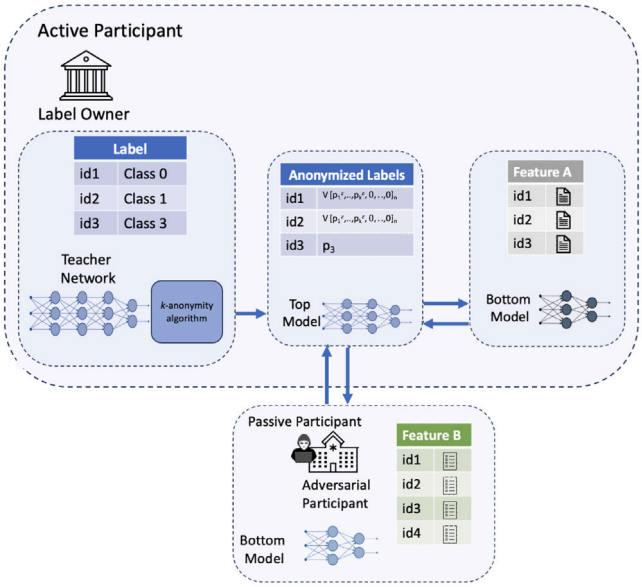

Fig. 4. KDk main components.

- Yahoo! Answers topic classification dataset [33] is formed by 10 main categories and each class contains 140,000 training samples and 6000 testing samples.
- Criteo [34] is a real-world dataset related to commerce for predicting ad click-through rates. In this dataset, composed of only 2 classes, categorical and continuous features are employed.

To assess the effectiveness of our defense approach, we adopt the following evaluation metrics:

- **Top-1 Accuracy**, is the conventional accuracy or ratio of correctly predicted samples to the total number of samples in the dataset. It measures how often the network has predicted the correct label with the highest probability.
- **Top-5 Accuracy**, is a metric that indicates how many times the correct label appears in the network's top five predicted classes. It is useful for large-scale datasets with numerous classes and for cases in which a degree of flexibility is acceptable and the exact class cannot be predicted with high confidence [35].
- **Top-1 Attack Success Rate** (Top-1 ASR) is the percentage of labels correctly extracted by attacks.
- **Top-5 Attack Success Rate** (Top-5 ASR) measures how often the label correctly extracted by attacks appears in the network's top five predicted classes.

For our experimental campaign, we refer to an *Original Architecture* (OA, hereafter) that represents a VFL scenario without any defense mechanism as presented by [11]. This architecture, shown in Fig. 3, employs different types of networks for each of the above-described datasets. In particular, as visible in Table 2, the top model of the VFL is implemented through a pre-trained ResNet-18 (i.e., an 18-layer convolutional neural network pre-trained on general data and fine-tuned on the active participant data) for the CIFAR-10, CIFAR-100, and CINIC-10 datasets; a fine-tuned BERT model [36] for Yahoo! Answer (that includes textual data); and a 3-layer Fully Connected Neural Network (FCNN-3) to process samples in the Criteo dataset.

Moreover, we implemented our KDk solution whose components are illustrated in Fig. 4. Compared to the Original Architecture, KDk includes a preliminary processing step executed only by the active participant to anonymize the labels. This step is realized through *(i)* a teacher network that shares the same or a scaled-up version of the

Table 2
Original model architectures.

Dataset	Top model architecture	Bottom model architecture
CIFAR-10	FCNN-4	ResNet-18
CIFAR-100	FCNN-4	ResNet-18
CINIC-10	FCNN-4	ResNet-18
Yahoo! Answers	FCNN-4	Bert
Criteo	FCNN-3	FCNN-3

Table 3
Teacher network architectures for KDk.

Dataset	Teacher network architecture
CIFAR-10	ResNet-50 + FCNN-1
CIFAR-100	ResNet-50 + FCNN-1
CINIC-10	ResNet-50 + FCNN-1
Yahoo! Answers	Bert + FCNN-1
Criteo	FCNN-4

OA architecture (bottom+top model) trained just on their partial data implementing the Knowledge Distillation and *(ii)* an algorithm that obfuscates the k labels with higher confidence based on k-anonymity. As we will see in the following sections, the k value can be used to solve a trade-off between the performance of the main model and the effectiveness of our defense against attacks, this makes it a tunable parameter used to tune the strength of the proposed defense. The teacher network architectures that have been implemented for each dataset are visible in Table 3.

All experiments have been performed on a workstation equipped with a AMD(R) Ryzen(R) 7 CPU 5800x @ 3.80 GHz, 32 GB RAM, and an NVIDIA RTX 3070Ti GPU card.

6.2. Label inference attacks performance comparison

In this section, we report the performance results of our defense mechanism against the four types of Label Inference attacks described in Section 4.

For each analyzed attack, we chose the appropriate configuration of the two anonymization parameters ϵ and k, where ϵ is the smoothing parameter and k is the number of the highest labels considered for each data point to build our soft label solution.

6.2.1. Passive Label Inference Attack

We carried out the Passive Label Inference Attack with a 0.08% of auxiliary labeled data (as proposed in [11]) and we tested it against our KDk framework with the two anonymization parameters ϵ and k set for each dataset as are reported in Table 4. This table reports also the accuracy results of the attack against the original model proposed by [11] and against our defense mechanism. These values show that the performance of the attack against KDk is drastically reduced and, in most cases, halved compared to the performance against the original model of [11]. It is worth observing that, in the results obtained on the Yahoo!Answer dataset we can see a smaller reduction in the attack performance, which is caused by the fact that the bottom model is an already pre-trained Bert model. The knowledge included in the Bert model is already enough to obtain a basic classification of the text (i.e., information on the labels), even if the attacker does not infer additional information from the top model. Therefore the performance of the attack does not decrease as much as in the other cases.

6.2.2. Active Label Inference Attack

To perform the Active Label Inference Attack, we executed the malicious local optimizer in the training stage of our KDk model and then we performed the completion step of the passive inference attack to get the final label inference model as done in [11]. The configurations of the two anonymization parameters ϵ and k chosen for each dataset are reported in Table 5. Similarly to the previous experiment, when we

Table 4
Passive Label Inference Attack performance against OA and KDk.

Attack success rate (ASR)							
Dataset	Type of ASR	ϵ	k	OA		KDk	
				Training set	Test set	Training set	Test set
CIFAR-10	Top-1	0.45	3	80.6%	61.7%	43.9%	35.8%
CIFAR-100	Top-1	0.50	3	31.3%	18.0%	15.9%	10.5%
CIFAR-100	Top-5	0.50	3	62.2%	41.0%	40.3%	29.7%
CINIC-10	Top-1	0.45	3	65.2%	49.0%	32.0%	26.0%
Yahoo! Answers	Top-1	0.35	3	63.3%	63.7%	47.5%	47.4%
Criteo	Top-1	0.40	2	71.2%	71.9%	50.6%	50.3%

Table 5
Active Label Inference Attack performance against OA and KDk.

Attack success rate (ASR)							
Dataset	Type of ASR	ϵ	k	OA		KDk	
				Training set	Test set	Training set	Test set
CIFAR-10	Top-1	0.50	3	84.8%	63.4%	40.5%	35.1%
CIFAR-100	Top-1	0.60	3	39.3%	21.4%	17.6%	12.2%
CIFAR-100	Top-5	0.60	3	72%	47.4%	43.3%	32.7%
CINIC-10	Top-1	0.50	3	73.5%	50.5%	34.5%	29.3%
Yahoo! Answers	Top-1	0.40	3	64.2%	64.1%	52.2%	52.1%
Criteo	Top-1	0.40	2	71.2%	71.9%	50.0%	50.0%

Table 6
Active Label Inference Attack performance with different levels of strength against KDk.

Attack success rate (ASR)							
Datasets	Malicious lr	ϵ	k	OA		KDk	
				Training set	Test set	Training set	Test set
CIFAR10	0.05	0.50	3	84.8%	63.4%	40.5%	35.1%
	0.01	0.60	3	92.4%	69.6%	40.67%	35.1%
	0.005	0.60	3	94.3%	70.9%	41.3%	36.8%
CIFAR100 (Top-5)	0.05	0.60	3	39.3% (72%)	21.4% (47.4%)	17.6% (43.3%)	12.2% (32.7%)
	0.01	0.70	5	68.1% (92.9%)	28.4% (55.4%)	31.5% (67.5%)	17.3% (41.7%)
	0.005	0.70	5	72.8% (95.5%)	28.4% (56.7%)	35.3% (71.9%)	20.0% (47.1%)
CINIC10	0.05	0.50	3	65.2%	49.0%	32.0%	26.0%
	0.01	0.60	3	81.5%	56.6%	42.8%	37.4%
	0.005	0.60	3	82.9%	58.3%	44.4%	37.9%

performed an active label inference attack against our KDk framework the ASR consistently decreased compared to the ASR of the attacked OA. Observe that, the attack strategy of increasing the learning rate on the controlled client to promote more informative feedback from the server does not result in an advantage, because thanks to our defense the received signal is heavily obfuscated. Also in this case, the results from the Yahoo! Answer dataset present a smaller reduction in the attack performance, which is, once again, caused by the fact that the bottom model is an already pre-trained Bert model.

To further assess the capability of our approach against an active attacker we performed additional experiments that test our defense against different levels of "aggressiveness" compared to the baseline presented in [11]. In particular, we try to make the entire system rely gradually more on the attacker's model changing its local learning rate.

As we can see from Table 6, the attack without any countermeasure performs significantly better but still, against our approach, its effect is mitigated and the attack accuracy is almost reduced to half even in the most complex scenarios.

6.2.3. Direct Label Inference Attack

We carried out the Direct Label Inference Attack described in [11] and we tested it against our KDk framework with the two anonymization parameters ϵ and k set for each dataset as are reported in Table 7. In this table, we report the ASR results of the Direct Label Inference Attack against OA and our KDk framework. Since no gradients are available at the inference time, this attack can be conducted only at the training step hence we report the ASR values referred to the training set. As we can observe, in general, our defense mechanism can reduce the ASR of the attack by more than 60% except for the Criteo dataset because the number of classes is equal to 2 and therefore a random choice of the target label would lead to an ASR result higher than 0.5.

Table 7
Direct Label Inference Attack performance against OA and KDk.

Attack success rate (ASR)					
Dataset	Type of ASR	ϵ	k	OA	KDk
				Training set	
CIFAR-10	Top-1	0.45	3	100%	38.5%
CIFAR-100[a]	Top-1	0.5	3	100%	32.6%
CINIC-10	Top-1	0.45	3	100%	38.3%
Yahoo! Answers	Top-1	0.35	3	100%	39.6%
Criteo	Top-1	0.4	2	100%	80%

[a] In this case, we do not consider the Top-5 accuracy because the Top-1 is already 100%.

6.3. Models performance comparison

The main idea behind our approach, as presented in Section 5, is to obfuscate the information of the real label to add uncertainty in

Table 8
Performance on KDk compared to OA for the used datasets.

Model accuracy			
Dataset	Accuracy	OA	KDk
CIFAR-10	Top-1	81%	79%
CIFAR-100	Top-1	49.1%	49.4%
CIFAR-100	Top-5	78.5%	79.9%
CINIC-10	Top-1	66.7%	64.3%
Yahoo! Answers	Top-1	71.1%	67.5%
Criteo	Top-1	71.3%	69.9%

the bottom model of the attacker to inhibit the effectiveness of the attacks presented in Section 4. Inevitably, this approach will affect the performance of the model on the original task, though we try to minimize it by obfuscating the real information in a set of highly probable alternatives (and, therefore, possibly avoiding heavily impacting the performance of the top model). The results in the previous section have been obtained by setting the anonymization parameters to guarantee the preservation of the original global model performance. The accuracy performance of our KDk model compared to OA (the original architecture proposed in [11]), is shown in Table 8 for each analyzed dataset. As we can see the performance of the original model is mostly preserved with small drops of 4% at maximum. Observe that for the CIFAR-100 dataset, the accuracy result is even higher, because of the effect of KD. Indeed, for large datasets, our model benefits from the generalization capabilities of the teacher network [37,38]. As said our approach can impact the model accuracy according to the strength level of the parameters k and ϵ. In the following Section 6.4, we present a detailed analysis combining parameters with different levels of strength and recording the model accuracy in model and the attack success rate.

6.4. Performance with different values of the anonymization parameters

In this experiment, we analyze how both the ASR and the performance of the model (in terms of accuracy) change in relation to higher values of the anonymization parameters ϵ and k. For this study, we considered only three datasets, namely CIFAR-10, CIFAR-100, and CINIC-10 because they have at least 10. Criteo has not been considered since it is a dataset with binary labels making it impossible to test our solution with k higher than 2. Yahoo instead is omitted since it relies on a pre-trained Bert model and, as we already stated in Section 6.2.1, its accuracy is intrinsically guaranteed by the performance of such underlying model, hence it cannot decrease lower than the values presented in Table 4 with any parameter combination.

That said, we studied the performance for different values of k (i.e., $k = 3$, $k = 5$, and $k = 10$). As typically done in the literature [11, 20], for CIFAR-100 we consider only the Top-5 accuracy that provides a more nuanced evaluation because of the large number of classes.

As visible in Fig. 5, using higher ϵ values results in balancing the probabilities of the classes, and this affects the overall performance of both the attack and the KDk model. Instead, using different k values does not affect our defense mechanism. Interestingly, setting $k = 10$ and using a dataset composed of 10 classes make the defense ineffective. This result confirms our intuition behind the logic that makes our proposal work. Our approach is effective because it relies on the uncertainty instilled in the bottom models obtained by anonymizing the real label between k additional and related ones (as indicated by our knowledge distillation component). In the case of $k = 10$, we set all the secondary labels to the same probability. This breaks the main logic behind our approach. Setting all the secondary labels to the same value produces similar (with the addition of an offset) loss values compared to the scenario using hard labels, directly. In this case, the offset added to the cross-entropy loss is not sufficient to properly obfuscate the labels, thus resulting in a small decrease in the accuracy of the attack

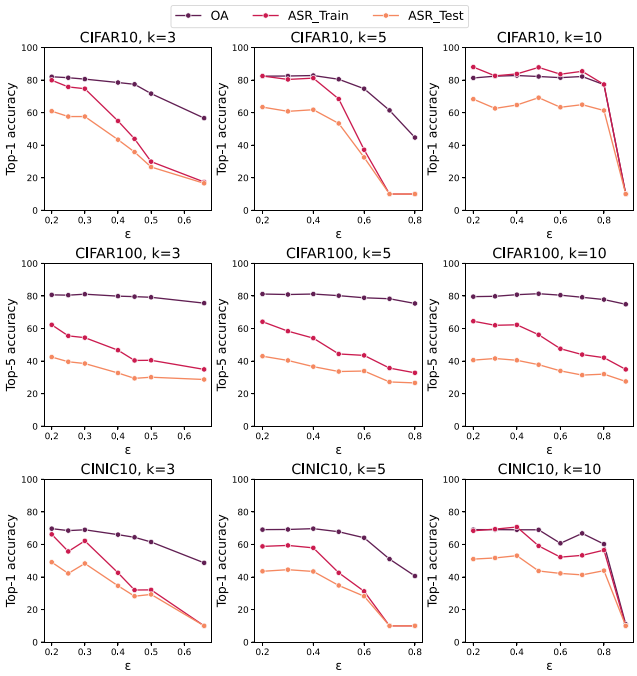

Fig. 5. Analysis of the performance of the attack and the performance of KDk for different ϵ and k values.

in the case of the CINI10 dataset or can even be ineffective in the case of CIFAR-10. To be effective with $k = 10$, our approach must push the ϵ value to extreme values. This setting is effective against the attack but also prevents the model from training using the probability distribution balanced across all the labels, thus resulting in an accuracy close to random guessing.

6.5. Comparison with other defense mechanisms

In the proposal of [11] several defensive strategies are applied to the gradients to prevent information leakage from the server and try to mitigate the different label inference attacks. In this section we compare our defense mechanisms with the following approaches in [11]:

- **Noisy Gradients (NG)**. To perform this defense in VFL, the server adds a Laplacian noise to gradients before sending them to passive participants. The metric we analyze to compare this approach with our KDk is the *noise scale*, which represents several scales of the used Laplacian noise.
- **Gradient Compression (GC)**. This strategy used for communication efficiency and privacy protection consists of sharing fewer gradients with the largest absolute values. The metric we consider to compare this approach with our KDk is the *compression rate*, which is the ratio between the uncompressed size and compressed size of the gradient values.
- **Privacy-Preserving Deep Learning (PPDL)**. In each iteration, the server *(i)* randomly selects one gradient value and adds noise to this gradient; *(ii)* sets to zero the gradient values smaller than a threshold value τ; *(iii)* repeats the first two steps until Θ_u fraction of gradient values are collected. Both τ and Θ_u are hyperparameters to balance the trade-off between model performance and defense performance. We evaluate the performance of this type of defense by analyzing different settings of the hyperparameter Θ_u.
- **DiscreteSGD** a customized version of signSGD [39] thought for VFL. The defense mechanism proceeds as follows. *(i)* In the

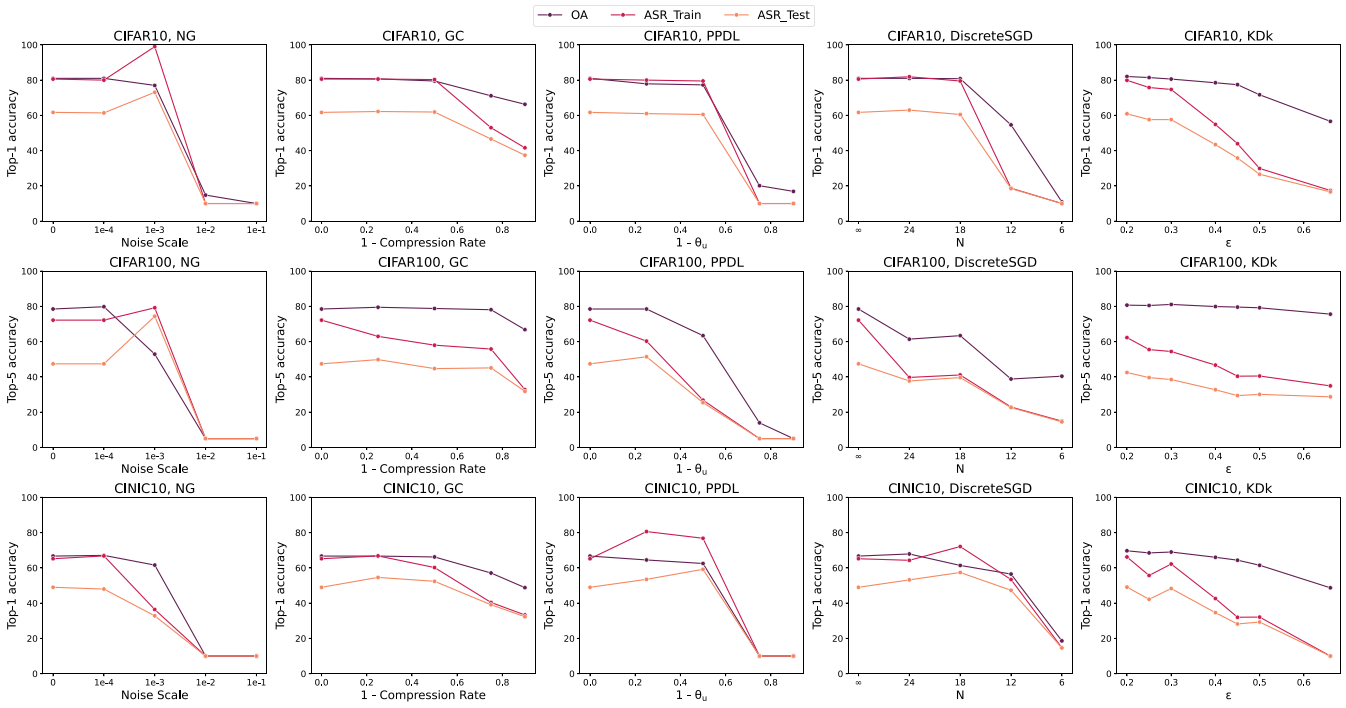

Fig. 6. Comparison with other Defense Mechanisms against passive and active label inference attacks.

Table 9
Comparison with other Defense Mechanisms against direct label inference attacks using CIFAR datasets.

			CIFAR-10		CIFAR-100	
Noisy gradients	Noise scale		Model accuracy	Attack accuracy	Model accuracy	Attack accuracy
	1e−4		81.4%	80.6%	82.4%	12.2%
	1e−3		81.1%	49.1%	83.1%	2.0%
	1e−2		71.9%	24.5%	5.1%	2.0%
	1e−1		10.0%	12.7%	5.0%	0.6%
Gradient compression	Compression rate		Model accuracy	Attack accuracy	Model accuracy	Attack accuracy
	75%		80.4%	99.9%	82.4%	100%
	50%		80.5%	99.3%	83.1%	100%
	25%		80.4%	92.4%	82.4%	99.9%
	10%		10.0%	0.1%	73.8%	99.9%
Privacy-preserving DL	Θ_u		Model accuracy	Attack accuracy	Model accuracy	Attack accuracy
	0.75		79.8%	39.0%	81.9%	4.6%
	0.50		80.5%	38.9%	81.7%	4.5%
	0.25		19.9%	0.1%	5.2%	1.1%
	0.10		10.0%	0.1%	5.0%	0.9%
Discrete SGD	N		Model accuracy	Attack accuracy	Model accuracy	Attack accuracy
	24		81.0%	96.7%	11.2%	99.9%
	18		80.8%	94.3%	8.8%	99.9%
	12		78.7%	94.7%	7.1%	99.9%
	6		74.3%	91.5%	7.3%	99.7%
KDk	k	ϵ	Model accuracy	Attack accuracy	Model accuracy	Attack accuracy
	3	0.45	79.0%	38.5%	80.6%	32.6%
	5	0.70	71.1%	23.0%	80.5%	19.2%
	5	0.75	66.7%	21.7%	79.7%	19.1%
	5	0.85	37.5%	14.2%	76.7%	18.8%

first epoch, the server observes the distribution of the shared gradients. Following the three-sigma rule [40], the server sets an interval as $[\mu - 2\sigma, \mu + 2\sigma]$ (where μ is the mean and σ is the standard deviation). The gradients outside of the interval are regarded as outliers and not considered. *(ii)* The server slices the interval into N sub-intervals. *(iii)* Before transmitting the gradients to all the participants, the server first rounds each gradient value to the nearest endpoint of the sub-intervals. The hyperparameter N controls how much magnitude information of the shared gradients is preserved.

We evaluate the four defense approaches introduced above and we compare them with our KDk approach.

For this experiment, we performed both passive and active label inference attacks on three datasets: CIFAR-10, CIFAR-100, and CINIC-10. Observe that, once again, as typically done in the related literature, for CIFAR-100 we considered only Top-5 accuracy to cope with the large number of classes. As for the setting of the different defense mechanisms, we considered the following parameters: Laplacian noise level $\in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, gradient compression percentage $\in$

Table 10
Comparison with other Defense Mechanisms against direct label inference attacks using CINIC-10 dataset.

			CINIC-10	
Noisy gradients	Noise scale		Model accuracy	Attack accuracy
	1e−4		70.5%	84.3%
	1e−3		69.9%	49.7%
	1e−2		55.5%	24.3%
	1e−1		10.3%	12.6%
Gradient compression	Compression rate		Model accuracy	Attack accuracy
	75%		70.9%	99.8%
	50%		69.1%	99.3%
	25%		54.7%	92.5%
	10%		10.0%	0.01%
Privacy-preserving DL	Θ_u		Model accuracy	Attack accuracy
	0.75		69.4%	38.9%
	0.50		68.4%	38.6%
	0.25		20.8%	0.10%
	0.10		12.9%	0.04%
Discrete SGD	N		Model accuracy	Attack accuracy
	24		63.1%	97.9%
	18		59.6%	95.6%
	12		45.8%	94.3%
	6		43.6%	90.3%
KDk	k	ϵ	Model accuracy	Attack accuracy
	3	0.45	67.7%	38.3%
	5	0.70	62.2%	24.1%
	5	0.75	56.7%	22.2%
	5	0.85	34.5%	15.3%

$\{75\%, 50\%, 25\%, 10\%\}$, PPDL Θ_u fraction $\in \{10\%, 25\%, 50\%, 75\%\}$, DiscreteSGD number of intervals $N \in \{6, 12, 18, 24\}$. The parameters of our approach instead are set as follows: $k = 3$ and ϵ varying between the values $\in \{0.25, 0.3, 0.45, 0.5, 0.66\}$.

6.5.1. Passive and active attacks

The results of the defenses against the model completion attack are reported in Fig. 6. As visible in Figs. 6, for noisy gradients (NG), we experimented using several scales of Laplacian noise to evaluate its defense performance against model completion inference attack. Obviously the greater the value of the noise scale the more successful are all the mitigation techniques. To be effective this defense must apply to the gradients an extremely high level of noise that disrupts the performance of the model on the original task. With lower levels of noise, it is interesting to see how this approach can even help obtain higher attack performance.

In the second column of sub-figures in Fig. 6, we evaluate Gradient Compression (GC) techniques for different compression rates. Also from these figures, we can notice that for greater compression rates both the model and the attack performance decrease. We can see how between the selected defenses compared to ours, gradient compression is the best preserving the original accuracy of the model but only slightly affecting the performance of the attack, especially with lower values of compression. As for the PPDL mechanism, from the sub-figures in the third column in Fig. 6 we can notice that for all three analyzed datasets, the defense can mitigate label inference attacks with the hyperparameter Θ_u set to 0.25 or lower (i.e., the accuracy result is lower than 40% for $1 - \Theta_u = 0.75$). Even in this case, we can notice how the defense is effective only with a high level of manipulation of the gradient resulting in a heavy loss in terms of performance on the original task for both CIFAR-10 and CINIC-10. As for CIFAR-100, instead, PPDL represents the best-performing defense we are comparing with.

Finally, similarly to the previous defenses, we can notice how DiscreteSGD is not capable of affecting the attack preserving the functionality of the original model. This defense can achieve slightly high performance only for CIFAR-10.

Looking at our solution compared to the others we can see how we can prevent the attack from decreasing its success rate to almost the same as a random guess value on CIFAR-10 and CINIC-10 using extreme values for ϵ while preserving most of the accuracy of the main model. It is also interesting to see how, even with lower defense intensity values, our approach affects the attack more than the other solutions.

6.5.2. Direct Label Inference Attack

In Tables 9 and 10, we analyze the performance of the three analyzed defense mechanisms and KDk against the direct label inference attack. The employed datasets are, once again, CIFAR-10, CIFAR-100 (see Table 9), and CINIC-10 (see Table 10). As we can see, the behavior of the defenses we are comparing is similar to the one witnessed for the passive and active attacks. Indeed, especially for CIFAR-10 and CINIC-10, the defenses are effective only when the alteration is such that the impact on the main task accuracy is not negligible. The only countermeasure capable of matching our solution in terms of preservation of the original model accuracy and detriment of the attack performance is the Noisy Gradients defense. Looking at CIFAR-100, instead, we can see how also the PPDL defense is capable of achieving good results. Compared to the others, our defense is equally effective on the three considered datasets. In this case, though, a more powerful setting is required to counter the more powerful Direct Attack.

In summary, from the above experiments, we can conclude that our defense strategy is the only one obtaining good performance against all the different attacks and for all the analyzed datasets. Generally, most of the other defenses failed to protect against label inference attacks. Only the PPDL and Noisy Gradients succeeded in *some* of the considered attack scenarios but, as visible in our results, they cannot be used as a general defense because they do not provide adequate protection against all the possible attack settings.

6.5.3. Comparison between defenses using defense score

In the previous sections, we showed how our defense is capable of mitigating the attacks across *all* the different settings, even when the existing related defenses drastically fail. To further prove the advantage introduced by our defense, we define here a *Defense Score* (DS, for short) metric that gives global information about the quality of our defense when compared to existing defenses in the different scenarios considered above. The metric is defined as follows:

$$DS = \frac{(1 - (BTA - TAD)) + (BAA - AAD)}{2} \tag{17}$$

The metric is composed of two parts. The first checks the defense's capability of preserving the accuracy of the model on the main task by subtracting to 1 the difference between the baseline task accuracy (BTA) and the task accuracy with the defense (TAD). The second contribution, instead, is obtained by computing the difference between the baseline attack accuracy (BAA) and the accuracy of the attack with the defense in place (AAD). The two contributions are then averaged to obtain a score between 0 and 1 for each experiment. In Table 11, we display all the obtained scores for different datasets and attack types. Bold values indicate the highest score, while underlined values represent the second highest. Our solution consistently outperforms other defenses across all scenarios, with competitors only achieving second-best values in certain cases but failing in others. Finally, Fig. 7 presents the aggregate scores for all reviewed defenses, further reinforcing that our proposed approach delivers the best overall performance in comparison to others.

7. Conclusion

Federated Learning (FL) is a novel paradigm aiming to train ML models in a privacy-preserving and collaborative way. Differently from Horizontal FL, in Vertical FL (VFL) participants share the same sample space, but their local private data differ in the feature space. Moreover, in standard VFL, the labels of the samples contain sensitive information

Table 11
Comparison of defenses in different attack scenarios using the Defense Score metric (Bold values=best performance, Underlined values=second best performance).

	CIFAR-10		CIFAR-100		CINIC-10	
	Active/passive attack	Direct attack	Active/passive attack	Direct attack	Active/passive attack	Direct attack
Noisy gradients	0.47	0.69	0.38	0.80	0.50	0.71
Gradient compression	0.52	0.54	0.50	0.51	0.51	0.56
Privacy-preserving DL	0.49	0.73	0.45	0.81	0.45	0.78
Discrete SGD	0.52	0.53	0.47	0.15	0.48	0.46
KD*k*	**0.58**	**0.79**	**0.55**	**0.89**	**0.57**	**0.82**

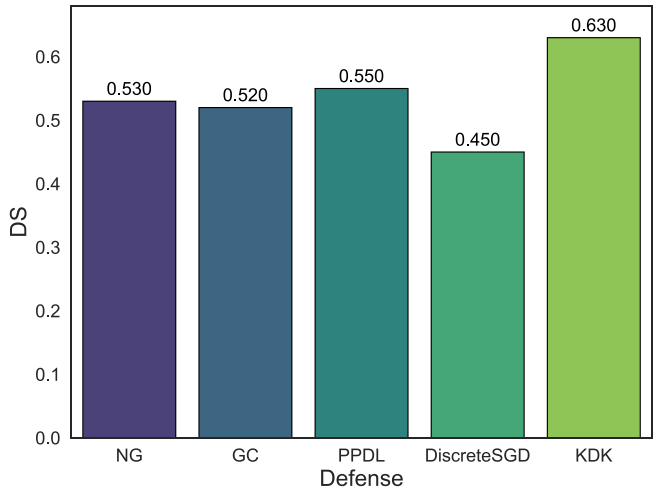

Fig. 7. Average performance comparison of defense strategies using the Defense Score metric.

and should be protected from honest-but-curious parties. Hence, only the aggregating server (or active actor) knows them, whereas they are kept secret from all the other parties (passive actors). Nevertheless, recent works have started to describe label leakage issues in this context proposing strategies for label inference attacks, namely passive, active, and direct attacks. In this paper, we analyzed such existing attacks and proposed a novel defense mechanism, called KD*k*, able to protect VFL from all the known types of label inference attacks with very high performance. Our approach modifies the active participant model, integrating both a Knowledge Distillation teacher network and a *k*-anonymity processing step to obtain a group of *k* most probable soft labels for each item instead of a single hard label. This adds a level of uncertainty that prevents the attacker from performing label inference successfully. We tested the performance of our solution with a thorough experimental campaign, whose objective was to demonstrate that our approach can effectively inhibit the attacker from being able to perform label inference (attack success rate reduced, in some cases, even more than 60% compared to its performance in the absence of our defense), still maintaining an almost unaltered accuracy of the federated global model (less than 2% performance decrease on average). Finally, we demonstrated the superiority of our proposal compared to the most recent and state-of-the-art existing defenses, which proved to be either ineffective against the attack or, in some cases, effective against only some attack variants, often at the cost of an extremely high, and hence not acceptable, impact on the federated global model performance.

The proposal and results described in this paper must not be seen as the conclusion of this research. In fact, in the future, we plan to develop our proposed KD*k* defense method further to provide enhanced protection for other kinds of FL and attacks, designing a complete protection framework. For instance, we intend to focus also on Horizontal FL. Due to the peculiarities of this variant, a thorough examination must be conducted to comprehend how our defense mechanism can be adjusted accordingly.

CRediT authorship contribution statement

Marco Arazzi: Writing – original draft, Validation, Software, Resources, Investigation, Data curation, Conceptualization. **Serena Nicolazzo:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Conceptualization. **Antonino Nocera:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Finanziato dall'Unione europea
NextGenerationEU

This work was supported in part by the following projects:
(i) The PRIN 2022 Project "HOMEY: a Human-centric IoE-based Framework for Supporting the Transition Towards Industry 5.0" (code: 2022NX7WKE, CUP: F53D23004340006) funded by the European Union - Next Generation EU.
(ii) The project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor the Italian MUR can be held responsible for them.

Data availability

Data is publicly accessible.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Aguera y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, Ft. Lauderdale, FL, USA, 2017, pp. 1273–1282.
[2] Marco Arazzi, Serena Nicolazzo, Antonino Nocera, A fully privacy-preserving solution for anomaly detection in iot using federated learning and homomorphic encryption, Inf. Syst. Front. (2023) 1–24.
[3] Qiang Yang, Yang Liu, Tianjian Chen, Yongxin Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) 1–19.
[4] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Sha Wei, Fan Wu, Guihai Chen, Thilina Ranbaduge, Vertical federated learning: Challenges, methodologies and experiments, 2022, arXiv:2202.04309[cs.LG].
[5] Sean Vucinich, Qiang Zhu, The current state and challenges of fairness in federated learning, IEEE Access 11 (2023) 80903–80914.
[6] Marco Arazzi, Mauro Conti, Antonino Nocera, Stjepan Picek, Turning privacy-preserving mechanisms against federated learning, in: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, ACM, Copenhagen, Denmark, 2023, pp. 1482–1495.
[7] Maoqiang Wu, Xinyue Zhang, Jiahao Ding, Hien Nguyen, Rong Yu, Miao Pan, Stephen T. Wong, Evaluation of inference attack models for deep learning on medical data, 2020, URL https://arxiv.org/abs/2011.00177, arXiv:2011.00177[cs.LG].
[8] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, Björn Eskofier, Federated learning for healthcare: Systematic review and architecture proposal, ACM Trans. Intell. Syst. Technol. 13 (4) (2022) 1–23.

[9] Mark Phillips, International data-sharing norms: from the OECD to the general data protection regulation (GDPR), Hum. Genet. 137 (2018) 575–582.
[10] Aydin Abadi, Bradley Doyle, Francesco Gini, Kieron Guinamard, Sasi Kumar Murakonda, Jack Liddell, Paul Mellor, Steven J. Murdoch, Mohammad Naseri, Hector Page, George Theodorakopoulos, Suzanne Weller, Starlit: Privacy-preserving federated learning to enhance financial fraud detection, 2024, URL https://arxiv.org/abs/2401.10765arXiv:2401.10765[cs.LG].
[11] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, Ting Wang, Label inference attacks against vertical federated learning, in: 31st USENIX Security Symposium (USENIX Security 22), USENIX Association, Boston, MA, USA, 2022, pp. 1397–1414.
[12] Jianping Gou, Baosheng Yu, Stephen J Maybank, Dacheng Tao, Knowledge distillation: A survey, Int. J. Comput. Vis. 129 (2021) 1789–1819.
[13] Pierangela Samarati, Protecting respondents identities in microdata release, IEEE transactions on Knowledge and Data Engineering 13 (6) (2001) 1010–1027.
[14] Luca Melis, Congzheng Song, Emiliano De Cristofaro, Vitaly Shmatikov, Exploiting unintended feature leakage in collaborative learning, in: 2019 IEEE Symposium on Security and Privacy, SP, IEEE, San Francisco, CA, USA, 2019, pp. 691–706.
[15] Ligeng Zhu, Zhijian Liu, Song Han, Deep leakage from gradients, Adv. Neural Inf. Process. Syst. 32 (2019) 14747–14756.
[16] Milad Nasr, Reza Shokri, Amir Houmansadr, Comprehensive privacy analysis of deep learning, in: Proceedings of the 2019 IEEE Symposium on Security and Privacy, SP, IEEE, San Francisco, CA, USA, 2018, pp. 1–15.
[17] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, Beng Chin Ooi, Feature inference attack on model predictions in vertical federated learning, in: 2021 IEEE 37th International Conference on Data Engineering, ICDE, IEEE, Chania, Greece, 2021, pp. 181–192.
[18] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, Chong Wang, Label leakage and protection in two-party split learning, 2022, arXiv:2102.08504[cs.LG].
[19] Junlin Liu, Xinchen Lyu, Clustering label inference attack against practical split learning, 2022, arXiv:2203.05222[cs.LG].
[20] Yang Liu, Tianyuan Zou, Yan Kang, Wenhan Liu, Yuanqin He, Zhihao Yi, Qiang Yang, Batch label inference and replacement attacks in black-boxed vertical federated learning, 2022, arXiv:2112.05409[cs.LG].
[21] Jiankai Sun, Xin Yang, Yuanshun Yao, Chong Wang, Label leakage and protection from forward embedding in vertical federated learning, 2022, arXiv:2203.01451[cs.LG].
[22] Ivan Kholod, Andrey Rukavitsyn, Alexey Paznikov, Sergei Gorlatch, Parallelization of the self-organized maps algorithm for federated learning on distributed sources, J. Supercomput. 77 (2021) 6197–6213.
[23] Wensheng Xia, Ying Li, Lan Zhang, Zhonghai Wu, Xiaoyong Yuan, Cascade vertical federated learning towards straggler mitigation and label privacy over distributed labels, IEEE Trans. Big Data 1 (2023) 1–14.
[24] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, Yuan Gao, A survey on federated learning, Knowl.-Based Syst. 216 (2021) 106775.
[25] Cristian Buciluǎ, Rich Caruana, Alexandru Niculescu-Mizil, Model compression, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery (ACM), Beijing, China, 2006, pp. 535–541.
[26] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, Distilling the knowledge in a neural network, 2015, arXiv:1503.02531[stat.ML].
[27] Kai Fan, Jingtao Hong, Wenjie Li, Xingwen Zhao, Hui Li, Yintang Yang, FLSG: A novel defense strategy against inference attacks in vertical federated learning, IEEE Internet Things J. 11 (2023) 1816–1826.
[28] Tianyuan Zou, Yang Liu, Yan Kang, Wenhan Liu, Yuanqin He, Zhihao Yi, Qiang Yang, Ya-Qin Zhang, Defending batch-level label inference and replacement attacks in vertical federated learning, IEEE Trans. Big Data 10 (6) (2022) 1016–1027.
[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Las Vegas, NV, USA, 2016, pp. 770–778.
[30] Jiayu Wu, Qixiang Zhang, Guoxi Xu, Tiny imagenet challenge, Technical Report, Stanford University, 2017.
[31] Alex Krizhevsky, Geoffrey Hinton, et al., Learning multiple layers of features from tiny images, University of Toronto, Toronto, ON, Canada, 2009.
[32] Luke N. Darlow, Elliot J. Crowley, Antreas Antoniou, Amos J. Storkey, CINIC-10 is not ImageNet or CIFAR-10, 2018, arXiv:1810.03505[cs.CV].
[33] Xiang Zhang, Junbo Zhao, Yann LeCun, Character-level convolutional networks for text classification, Adv. Neural Inf. Process. Syst. 28 (2015) 649—-657.
[34] Criteo, Criteo AI lab, 2024, https://ailab.criteo.com/ressources.
[35] Felix Petersen, Hilde Kuehne, Christian Borgelt, Oliver Deussen, Differentiable top-k classification learning, in: International Conference on Machine Learning, PMLR, Baltimore, MD, 2022, pp. 17656–17668.
[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.
[37] Jang Hyun Cho, Bharath Hariharan, On the efficacy of knowledge distillation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, Seoul, South Korea, 2019, pp. 4794–4802.
[38] Chenglin Yang, Lingxi Xie, Siyuan Qiao, Alan Yuille, Knowledge distillation in generations: More tolerant teachers educate better students, 2018, arXiv:1805.05551[cs.CV].
[39] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, Animashree Anandkumar, Signsgd: Compressed optimisation for non-convex problems, in: International Conference on Machine Learning, PMLR, Vienna, Austria, 2018, pp. 560–569.
[40] Friedrich Pukelsheim, The three sigma rule, Amer. Statist. 48 (2) (1994) 88–91.

Marco Arazzi is currently a postdoc at the University of Pavia. He got his Ph.D. Student in Computer Engineering at the same University. From March 2023 to the end of July 2023 he worked as Visiting Researcher in the Cyber Security group of the Delft University of Technology (TU Delft). His research interests include Data Science, Machine Learning, Social Network Analysis, Internet of Things, Privacy and Security. He is an author of about 20 scientific papers in these research fields.

Serena Nicolazzo is currently a Research Fellow (RTDA) at the University of Milan. She got a Ph.D. in Information Engineering at the University Mediterranea of Reggio Calabria in 2017. Her research interests include IoT, Security, Privacy, and Social Network Analysis. She is an Editorial Board Member of Online Social Networks and Media (OSNEM) and she is involved in several TPCs of prestigious International Conferences in the context of Data Science and Cybersecurity. She is the author of about 50 scientific papers. She was a Visiting Researcher at the Middlesex University of London.

Antonino Nocera is an Associate Professor at the University of Pavia. His research interests span over Artificial Intelligence, Cybersecurity, and Data Science. The results of his work in these domains are collected in about 100 research papers published in prestigious International journals and conferences. He is a member of the DCALab laboratory of the University of Pavia in which he leads a research group, characterized by several international collaborations, focusing on Artificial Intelligence solutions applied to the Cybersecurity domain. He is Associate Editor of Information Sciences (Elsevier) and of the IEEE Transaction on Information Forensics and Security (T-IFS). Moreover, he is involved in the TPC of many renowned International Conferences focusing both on cybersecurity and artificial intelligence. Furthermore, he is the director of the local node of the University of Pavia for the CINI “Data Science” National Lab and a member of the local node of the University of Pavia for the CINI “Cybersecurity” National Lab.