Federated and Privacy-Preserving Learning of Accounting Data in Financial Statement Audits

Marco Schreyer  
University of St.Gallen (HSG)  
St. Gallen, Switzerland  
marco.schreyer@unisg.ch

Timur Sattarov  
Deutsche Bundesbank  
Frankfurt am Main, Germany  
timur.sattarov@bundesbank.de

Damian Borth  
University of St.Gallen (HSG)  
St. Gallen, Switzerland  
damian.borth@unisg.ch

ABSTRACT
The ongoing ‘digital transformation’ fundamentally changes audit evidence’s nature, recording, and volume. Nowadays, the International Standards on Auditing (ISA) requires auditors to examine vast volumes of a financial statement’s underlying digital accounting records. As a result, audit firms also ‘digitize’ their analytical capabilities and invest in Deep Learning (DL), a successful sub-discipline of Machine Learning. The application of DL offers the ability to learn specialized audit models from data of multiple clients, e.g., organizations operating in the same industry or jurisdiction. In general, regulations require auditors to adhere to strict data confidentiality measures. At the same time, recent intriguing discoveries showed that large-scale DL models are vulnerable to leaking sensitive training data information. Today, it often remains unclear how audit firms can apply DL models while complying with data protection regulations. In this work, we propose a Federated Learning framework to train DL models on auditing relevant accounting data of multiple clients. The framework encompasses Differential Privacy and Split Learning capabilities to mitigate data confidentiality risks at model inference. Our results provide empirical evidence that auditors can benefit from DL models that accumulate knowledge from multiple sources of proprietary client data.

CCS CONCEPTS
• Computing methodologies → Machine learning; Multi-task learning; Dimensionality reduction and manifold learning; • Information systems → Enterprise resource planning.

KEYWORDS
federated learning, differential privacy, financial auditing, anomaly detection, computer-assisted audit techniques, accounting information systems, enterprise resource planning systems

ACM Reference Format:
Marco Schreyer, Timur Sattarov, and Damian Borth. 2022. Federated and Privacy-Preserving Learning of Accounting Data in Financial Statement Audits. In 3rd ACM International Conference on AI in Finance (ICAIF ’22), November 2–4, 2022, New York, NY, USA. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3533271.3561674

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

ICAIF ’22, November 2–4, 2022, New York, NY, USA  © 2022 Copyright held by the owner/author(s).  
ACM ISBN 978-1-4503-9376-8/22/11  
https://doi.org/10.1145/3533271.3561674

1 Interpretations of the Statements of Auditing Standard (SAS) No. 56 indicate that ‘in circumstances where the auditor specializes in a specific industry, the auditor may use client data’ to develop reasonable expectations [21].
advantage of the FL setup is the idea of local model training and central model aggregation [38]. Thereby, the training data remains decentralized, enabling the clients to learn a common model without sharing proprietary data [36]. However, as the benefits of such a learning scheme become apparent for audit firms, there is a gradual awakening to the fact that associated risks may also arise [41]. Recently, it has been demonstrated that ML models are vulnerable to data leakage attacks, e.g., designed to extract sensitive or personally identifiable information [10, 61].

At the same time, auditors are obliged to protect the confidentiality of their client’s data [1]. Currently, audit firms train ML models using proprietary client data, e.g., the journal entries posted in a client’s ERP system by an employee exhibiting sensitive information. Furthermore, (inter-) national legislation requires auditors to implement measures that ensure the protection of data privacy. Such regulations pose additional requirements for the cross-client learning and application of ML-based audit models. In 2006, Dwork et al. [16] proposed the concept of Differential Privacy (DP) to address the challenge of inferring confidential training data information through ML models. One of the DP learning setup’s central ideas is introducing random noise perturbations to ensure privacy through plausible deniability. Today, it often remains unclear how audit firms can use such learning setups and preserve the confidentiality of sensitive client data.

Inspired by these developments, we examine whether FL enhanced by DP can be utilized to (i) learn industry-specific audit models that (ii) comply with data confidentiality obligations. In summary, we present the following primary contributions:

- We propose a Federated Learning framework that enables auditors to learn industry-specific models from data of multiple clients without the need to access it centrally.
- We demonstrate how such a framework can be enhanced by Differential Privacy and Split Learning to learn confidentiality-preserving models of accounting data.
- We conduct an extensive evaluation using real-world datasets and illustrate the benefits of the learning setup based on the general audit procedure of detecting anomalies.

Driven by the progressive application of DL in financial audits, new and effective ways to securely learn ‘meta’ ML models from client data become of high relevance [64]. We see this work as a promising step toward the future adoption of Artificial Intelligence (AI) augmented auditing by audit firms. The remainder of this work is structured as follows: In Section 2, we provide an overview of related work. Section 3 follows with a description of the proposed framework to learn federated and privacy-preserving models of accounting data. In Section 4 and Section 5, we present an extensive evaluation using real-world datasets and illustrate the benefits of the learning setup based on the general audit procedure of detecting anomalies.

2 RELATED WORK

The application of ML in financial audits triggered a sizable body of research by academia [4, 56] and practitioners [14, 57]. In this section, we present our literature study focusing on the federated and differential-private learning of accounting data representations.

Federated Learning enables multiple decentral entities to collaboratively train ML models under the orchestration of a central trusted entity without compromising data protection [29]. Model aggregation is one of the most common federated learning mechanisms, which trains a global model by summarizing the model parameters from all decentralized parties. McMahan et al. [40] proposed FedAvg a deep network federation learning framework based on an iterative model averaging the local model in each round of updates. Papernot et al. [44] introduced PATE, which aggregates...
knowledge transferred from a teacher model that is trained on separated data of student models. Yurochkin et al. [65] developed a Bayesian framework for federated learning, establishing a global model by matching neurons of local models. The combination of federated learning and multitasking [55] allows multiple users to train models of different tasks locally, which is also a typical method of model aggregation. In [34], federated learning and blockchain are combined to exchange and update the model data of each piece of equipment based on the blockchain. Recently FL setups have been deployed in a variety of application scenarios, e.g., recommendation systems [22] and finance [31].

Privacy-Preserving Learning emerged to protect data processed by ML algorithms and prevent information leakage to non-trusted third parties. Nowadays, prominent techniques encompass secure multi-party computation [60], homomorphic encryption [46], and differential privacy [15]. Differential Privacy addresses the issue that attackers might infer sensitive training data information through trained models [16]. Dwork et al. [17] initially proposed the Gaussian Mechanism (GM) to solve the challenge of real-valued database queries. The idea of GM is the addition of noise to each dimension of the query result. Several approaches have been proposed in the context of ML to inject noise perturbations into the learning process. Fukuchi et al. [18], and Kang et al. [30] applied such perturbations to the input training data. In contrast, Zhang et al. [66] introduced gradient descent learning with model output perturbations. In the context of training deep learning models, a significant body of research investigates the idea of designing deliberate gradient perturbations [2], as well as perturbations of the learning objective [33]. In addition, Papernot et al. [44] proposed an approach of adding perturbation-based noisy labelling. Nowadays, DP learning mechanisms are applied in sensitive areas, such as healthcare [13] and financial risk modelling [67].

Representation Learning denotes techniques that allow systems to discover relevant data features to solve a specific task [7]. Nowadays, the majority of ML methods used in financial audits depend on ‘human’ engineered data features [12]. Such techniques encompass, for example, Naive Bayes classification [6], network analysis [39], univariate and multivariate attribute analysis [5], cluster analysis [58], transaction log mining [32], or business process mining [28]. With the advent of DL, novel representation learning techniques have emerged in financial audits [42]. Such methods rely on accounting data representations that are learned ‘end-to-end’ without human intervention [56]. Nowadays, applied representation learning techniques encompass, autoencoder neural networks [53], adversarial autoencoders [51], or variational autoencoders [68]. Lately, self-supervised learning techniques have been proposed to learn rich representations for multiple downstream audit tasks [48].

Concluding the literature survey and, to the best of our knowledge, this work presents the first step towards learning accounting data representations in a confidentiality-preserving manner.

3 METHODOLOGY

In this section, we introduce a framework for learning accounting data from multiple audit clients for accounting anomaly detection. The framework comprises four interacting learning setups, namely (i) representation, (ii) federated, (iii) split, and (iv) differential-private learning, described in the following.

Accounting Journal Entries: Formally, let \( D = \{x_1, x_2, ..., x_N\} \) define a population of \( i = 1, 2, 3, ..., N \) journal entries. Each individual entry, denoted by \( x_i = \{x_i^{M}, x_i^{1}, x_i^{2}, ..., x_i^{K}\} \), consists of \( j = 1, 2, 3, ..., M \) categorical accounting attributes and \( l = 1, 2, 3, ..., K \) numerical accounting attributes. The individual attributes encompass the journal entry’s details, such as an entry’s posting type, posting date, amount, or general ledger. When examining real-world journal entry data populations, two characteristics can be observed: First, the journal entry attributes exhibit a high variety of distinct attribute values, e.g., due to the high number of vendors or distinct posting amounts. Second, journal entries exhibit a plethora of distinct attribute value correlations, e.g., a document type usually posted in combination with a specific general ledger and vendor.

Accounting Anomalies: Derived from this observation and inspired by Breunig et al. [9], we distinguish two types of ‘anomalous’ entries [49], namely global and local accounting anomalies:

- **Global Accounting Anomalies** correspond to journal entries that exhibit unusual or rare individual attribute values, e.g., rarely used ledgers, or unusual posting times. Such anomalies often correspond to unintentional mistakes, are comparably simple to detect, and possess a high error risk.
- **Local Accounting Anomalies** correspond to journal entries that exhibit unusual attribute value correlations, e.g., rare co-occurrences of ledgers and posting types. Such anomalies might correspond to intentional deviations, are comparably difficult to detect, and possess a high fraud risk.

Ultimately, auditors aim to detect both types of accounting anomalies in real-world audits using unsupervised learning [43, 53, 68].

Representation Learning: To establish the anomaly detection setup we utilize bottleneck Autoencoder Networks (AEN’s) [24]. Formally, AEN’s are comprised of two nonlinear functions referred to as encoder and decoder networks that are jointly trained to reconstruct a given input. The encoder network \( q_\psi(x_i) \) maps an input \( x_i \in \mathbb{R}^K \) to a code vector \( z_i \in \mathbb{R}^m \) referred to as latent representation, where usually \( k > m \). Subsequently, the representation is mapped back by the decoder network \( p_\phi(z_i) \) to a reconstruction \( \tilde{x}_i \in \mathbb{R}^K \) of the original input. The AEN parameters are denoted as \( \theta = (\psi \cup \phi) \). In an attempt to achieve \( x \approx \tilde{x} \) both AEN networks are trained in parallel to minimize the dissimilarity of a given journal entry \( x_i \) and its reconstruction \( \tilde{x}_i = p_\phi(q_\psi(x_i)) \) as faithfully as possible. Consequently, the AEN learns optimal model parameters \( \theta^* = (\psi^* \cup \phi^*) \) by optimizing a training objective, formally defined as:

\[
\arg\min_{\phi, \psi} \| x_i - p_\phi(q_\psi(x_i)) \|.
\]

Upon successful AEN training, we examine the reconstruction loss \( L_{\text{rec}} \) of each journal entry [23]. Entries that correspond to a high reconstruction error will be selected for detailed audit procedures [47, 53]. Figure 1 illustrates a schematic view of the AEN based accounting anomaly detection approach.

Federated Learning: To preserve the confidentiality of client data, we extend the AENs anomaly detection capability by Federated Learning (FL) [40]. We assume that the entire training data is available as \( K \) sub-datasets \( D = (D_i)_{i=1}^K \). To establish the FL setup,
let $\xi$ denote a central AEN model with parameters $\theta^\xi = \{\psi^\xi \cup \phi^\xi\}$ maintained by a trusted audit firm. Let furthermore $\{\omega\}^K_{i=1}$ denote a set of decentral AEN models hosted by the audit clients. Formally, FL then solves an optimization task of the form:

$$\min_{x \in \mathcal{D}} \mathcal{L}(x) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \mathcal{L}_i(x),$$

where $\mathcal{L}_i(x) = \mathbb{E}_{x \sim \mathcal{D}_i} \left[ \left( \omega_i(x, \hat{x}) \right) \right]$, denotes the reconstruction loss of the $i$-th client, and $\mathcal{D}_i$ is the JE data subset of the $i$-th client. Similar to real-world audits, each subset $\mathcal{D}_i$ is only privately accessible by a single audit client. For $i \neq j$, $\mathcal{D}_i$ and $\mathcal{D}_j$ may be non-iid. An overview of the proposed training setup is illustrated in Fig. 2 (a). At training time, a synchronous update scheme is established that proceeds in rounds of communication [40]. At each round $r \in \mathbb{R}$, a set of $\lambda$ of available audit clients $\omega = \{\omega\}^\lambda_{i=1}$ and $|\omega_i| = \lambda$ is randomly selected. To initialize the FL, the auditor broadcasts its current central model $\psi^\omega$ to the selected clients. Subsequently, the clients $\omega_i$ conduct a round of decentral training iterations using their private sub-datasets $\mathcal{D}_i$. At each client, the training results in an updated decentral model $\omega_i$ with parameters $\theta^\omega_i = \{\psi^\omega_i \cup \phi^\omega_i\}$. Upon successful training, the clients send their decentral model parameters $\theta^\omega_i$ to the audit firm. Finally, the firm updates the central model parameters $\theta^\omega$ by aggregating the received parameters. This process repeats until the final communication round $R$ is reached.

To aggregate the updates, we adapt the Federated Averaging (FedAvg) [40] algorithm as presented in Algo 1. FedAvg computes a weighted average over the decentral updates, as defined by:

$$\theta^\omega_{r+1} = \frac{N_i}{N} \sum_{i=1}^{\lambda} \theta^\omega_{i, r},$$

where $N$ denotes the total number of journal entries, and $N_i$ is the number of entries privately accessible by a particular client. The actual central model is then utilized at inference time to audit a client’s proprietary data. An overview of the proposed inference scheme is illustrated in Fig. 2 (b).

**Split Learning:** When investigating the representations learned by AENs [52] it becomes apparent that representations learned by early (later) layers of the encoder (decoder) function encode detailed attribute characteristics, e.g., a client’s journal entry attribute co-occurrence pattern. In contrast, the later layers of the encoder (early decoder) function learn representations that encode general or ‘meta’-post patterns, e.g., a client’s prevalent financial accounting principles [47, 48]. Derived from this observation and to increase the level of privacy, we apply principles of Split Learning (SL) [20]. Given an AEN architecture, we interpret the non-linear encoder (and decoder) function as a sequential application of layers, formally defined by $q_\ell(\cdot) \leftarrow r_{\ell 1}^q(p_{\ell - 1}^q(t_{\ell 1}^q(\phi_{\ell - 1}(t_{\ell 1}^q(\cdot)))))$ (and $p_\ell(\cdot) \leftarrow r_{\ell 1}^p(t_{\ell 1}^p(\psi_{\ell - 1}(t_{\ell 1}^p(\cdot))))$). As illustrated in Fig. 3, we divide the encoder (decoder) network into two sub-networks: a private $q_\ell^p$ and a public sub-network $q_\ell^w$ ($p_\ell^p$ and $p_\ell^w$). Thereby, the encoder (decoder) cut-layer $\ell_1^p$ defines the network separation. In each communication round, a client only transfers its public parameters $\theta^\omega = \{\psi^\omega; \phi^\omega\}$ to the auditor. The private parameters $\theta^\omega_i = \{\psi^\omega_i; \phi^\omega_i\}$ are not shared and kept private.

**Differential Private Learning:** To further reduce the risk of data leakage by the shared public model parameters $\theta^\omega$ we apply Differential Privacy (DP) [16] techniques. DP builds on the idea of introducing randomized noise into the learning procedure. A learning algorithm $A$ is understood to satisfy a $\epsilon$-DP mechanism if and only if for any pair of datasets $D$ and $D'$ that differ in only one element, the following statement holds true:

$$P\left[A(D) = t \right] \leq e^\epsilon P\left[A(D') = t \right], \forall t,$$

where $P\left[A(D) = t \right]$ denotes the probability that algorithm $A$ outputs $t$ and $\epsilon > 0$. Thereby, the quantity $\ln P\left[A(D) = t \right]$ denotes the privacy loss. To establish Differential Private-Stochastic Gradient Descent (DP-SGD) learning [2], we apply two modifications to the original SGD algorithm. First, the sensitivity of each per sample gradient is bounded by clipping it in the $l_2$-norm. We denote the parameter $\nu_{max}$ as the maximum norm threshold of the per-sample gradients. Second, before updating the model parameters, perturbation noise is added to the clipped gradients using the Gaussian Mechanism (GM) [17]. Assuming a target function $f : D \rightarrow \mathbb{R}$ over the data $D$, the GM formally denotes a randomization algorithm $f(D) = f(D) + z$. Thereby, $z$ denotes a perturbation random variable drawn from a

**Algorithm 1:** Federated Learning (FL) setup inspired by McMahan et al. [40], $\lambda$ denotes the number of audit clients, $\rho$ the batch-size, $T$ the number of decentral iterations, and $\eta$ the learning rate.

**Audit Firm executes:**

1. randomly initialize $\theta^\omega$
2. for each round $r = 1, 2, \ldots, R$ do
3. $\theta^\omega_i \leftarrow$ select random set of $\lambda$ clients
4. for each client $\omega_i \in \lambda$ in parallel do
5. $\theta^\omega_{i, r+1} \leftarrow$ ClientUpdate($i, \theta^\omega_i, r$)
6. end for
7. $\theta_\mathcal{F}^{\omega+1} \leftarrow \frac{N_i}{\lambda} \sum_{i=1}^{\lambda} \theta^\omega_{i, r+1}$
8. end for

**Audit Client executes:**

1. ClientUpdate($i, \theta^\omega_i, r$):
2. $\theta^\omega_i \leftarrow$ initialize client model with $\theta^\omega_i$
3. $\mathcal{B} \leftarrow$ split private data $\mathcal{D}_i$ into batches of size $\rho$ each
4. for each iteration $r = 1, 2, \ldots, T$ do
5. $b_i \leftarrow$ random sample batch $b \sim \mathcal{B}$
6. $\theta^\omega_{i, r+1} \leftarrow \theta^\omega_i - \eta \nabla \mathcal{L}(\theta^\omega_i, b)$
7. end for
8. return $\theta^\omega_i$ to Audit Firm
Gaussian distribution $z \sim N(\mu, \sigma)$. The amount of added noise is determined by the Gaussian’s standard deviation $\sigma$ derived from $\sigma = \sqrt{\text{max} \times \kappa}$. To ensure $e^\text{DP}$, increasing the per sample gradient threshold $\sqrt{\text{max}}$ will result in additional noise, where the parameter $\kappa$ denotes a factor of noise multiplication.

4 EXPERIMENTAL SETUP

In this section, we describe the experimental details of the privacy-preserving federated learning setup to audit large-scale JE data.\(^4\)

Datasets and Data Preparation: To evaluate the learning capabilities of the learning framework, we use three publicly available datasets of real-world financial payment data. The datasets exhibit high similarity to ERP accounting data, e.g., typical manual payments (SAP T-Code: F-53) or payment runs (SAP T-Code: F-110).

- The City of Philadelphia payments\(^5\) denoted as $D^A$ encompass a total of 238,894 payments generated by 58 distinct city departments in 2017. Each payment exhibits 10 categorical and 1 numerical attribute(s).
- The City of Chicago payments\(^6\) denoted as $D^B$ encompass a total of 108,478 payments generated by 54 distinct city departments in 2021. Each payment exhibits 6 categorical and 1 numerical attribute(s).
- The City of York payments\(^7\) denoted as $D^C$ encompass a total of 102,026 payments generated by 49 distinct city departments in 2020/21 and 2021/22. Each payment exhibits 11 categorical and 2 numerical attribute(s).

For all datasets, let $\{A_i\}^N_{i=1}$ denote the set of payments generating city departments. For each department $A_i$, the payment’s attribute values are pre-processed to derive $\hat{x} \in \hat{X}$. Thereby, the categorical attribute values of $\hat{x}_i$ are converted into one-hot numerical tuples of bits $\hat{x}_i^j \in \{0, 1\}^\varepsilon$, where $\varepsilon$ denotes the number of unique attribute values in $x_i$. The numerical attribute values of $\hat{x}_i$ are scaled, according to $\hat{x}_i = (x_i^j - \min(x_i^j))/\max(x_i^j) - \min(x_i^j))$, where the min and max are obtained over all attribute values in $x_i$.

Anomaly Detection Setup: To establish the anomaly detection setup, we inject a small amount of $60 (30, 20)$ randomly sampled global and 140 (70, 30) local anomalies into dataset $D^A$ ($D^B$, $D^C$).\(^8\) We use a symmetrical encoder $q_\theta$ and decoder $p_\phi$, architecture, as shown in Tab. 1, in all our experiments. In both networks, we apply Leaky-ReLU non-linear activations with scaling factor $\alpha = 0.4$ except in the encoder’s bottleneck and decoder’s final layer, which comprise Tanh activations. The models are trained for $R = 100$ $(200, 200)$ communication rounds and $T = 200$ $(400, 400)$ update iterations per round. We use a batch size of $\rho = 64$ $(32, 32)$ journal entries, apply Adam optimization with $\beta_1 = 0.9, \beta_2 = 0.999$, and early stopping once the loss converges. Furthermore, optimize the AEN models using a combined loss-function [52] that computes the reconstruction error of an encoded journal entry $\hat{x}_i$, according to:

\[\ell(\hat{x}_i) = \sum_{j=1}^{M} \log \left(1 - \hat{y}_i^{j} \right) + \sum_{j=1}^{K} \ell(\hat{x}_i^{j}) \]

where $\ell(\hat{x}_i^{j})$ denotes the $j$-th journal entry reconstruction error for the $i$-th anomaly, $M$ the number of categorical attributes, and $K$ the number of numerical attributes. Thus, $\ell(\hat{x}_i)$ denotes the normalized binary-cross-entropy error of a categorical attribute $\hat{x}_i$.

\[\ell(\hat{x}_i) = \ell(\hat{x}_i) + \frac{1}{T} \sum_{t=1}^{T} \log \left(1 - \hat{x}_i^t \right) + \log \left(1 - \hat{x}_i^t \right) \]

To account for the higher number of categorical attributes in each dataset, we set $\theta = \frac{1}{T}$ to balance both loss terms.

Federated Learning (FL) Setup: We establish the FL setup using (i) non-iid and (ii) iid data partitions for each payment dataset $D \in \{D^A, D^B, D^C\}$. In the non-iid setup, the payments of a dataset $D_i$ are partitioned according to the dataset’s generating city departments $\{A_i\}^N_{i=1}$. Subsequently, the derived partitions are distributed across the federated clients to yield an approximately equal number of payments per federated client $\lambda$. In the iid setup, the payments are randomly distributed across the federated clients disregarding the generating city department information. In both setups, the payments $x_i$ of each client are pre-processed to derive $\hat{x}_i$, where $\hat{x}_i = (\hat{x}_1^1, \hat{x}_2^2, ..., \hat{x}_M^1, \hat{x}_1^2, ..., \hat{x}_M^2)$. Finally, global and local anomalies are injected into the payments residing at the first client to serve as the basis for our evaluation. In all our experiments, we partition a dataset into $Y = 8$ sub-datasets. We use $\lambda \in \{1, 4, 8\}$ federated clients and the corresponding number of data partitions. The number of payments per client is shown in Tab. 2.

Split Learning (SL) Setup: To facilitate the SL setup we set the cut-layer to $\ell^L = 1$ and $\ell^E = 1$. Thereby, for each decentralized client model $\theta^D$, the parameters of the first encoder (last decoder) layer are kept private and are not transferred for a central model aggregation. We implement the FL/SL setup by adapting the Flower library [8].

---

\(^4\)Due to the general confidentiality of JE data, we propose the evaluated methodology made publically available real-world datasets to allow for result reproducibility.

\(^5\)https://www.phila.gov/2019-03-29-philadelphia-initial-release-of-city-payments-data

\(^6\)https://data.cityofchicago.org/Administration-Finance/Payments/s4vu-giwb/

\(^7\)https://data.yorkopedata.org/dataset/all-payments-to-suppliers/

\(^8\)To create both classes of anomalies, we use the ‘Faker’ project: https://github.com/jokerclick/faker.
Differential Private Learning (DPL) Setup: To evaluate the DPL setup we run several experiments using maximum norm thresholds \( \nu_{\text{max}} \in [0.001, 0.01, 0.1, 1.0, 10.0] \). We applied ‘flat’ gradient clipping that calculates the norm of the gradient over all parameters. Furthermore, we evaluate a set of noise multiplier configurations \( \kappa \in [0.0, 0.001, 0.005, 0.01, 0.015, 0.1, 0.15, 0.2] \). To implement the DP setup we use the Opacus library [63].

Evaluation Metric: To quantitatively assess the anomaly detection capability of the fine-tuned models we derive the average precision \( AP \) over the sorted payments reconstruction errors \( \mathcal{L}_{\text{REC}} \). The \( AP \) summarizes the precision-recall curve, formally defined by:

\[
AP(\mathcal{L}_{\text{REC}}) = \sum_{i=1}^{N} (R_i - R_{i-1}) P_i,
\]

where \( P_i(\mathcal{L}_{\text{REC}}) = TP_i / (TP_i + FP_i) \) denotes the detection precision, and \( R_i(\mathcal{L}_{\text{REC}}) = TP_i / (TP_i + FN_i) \) denotes the detection recall of the \( i \)-th reconstruction error threshold.

5 EXPERIMENTAL RESULTS

In this section, we present and assess the results of three Research Questions (RQs) when evaluating the different learning setups that constitute the proposed framework.

RQ 1: Will the accounting anomaly detection performance benefit from applying different FL setups?

We evaluate the FL setup by comparing the learned model’s anomaly detection capabilities to baseline models \( AEN_{\text{base}} \). For the dataset, \( \mathcal{D}^A (\mathcal{D}^B, \mathcal{D}^C) \) the baseline is trained for 100 (200, 400) epochs and 200 (400, 400) iterations per epoch on the entire dataset as proposed in [49]. The distinct FL models \( AEN_{FL} \) are trained with \( \nu_{\text{max}} = 1.0, \kappa = 0.1, \) and \( \psi^i = \mathbb{I} \) (disregarding DPL and SL variations) in the non-iid (\( \mathcal{D}_{\text{niid}} \)) and iid (\( \mathcal{D}_{\text{id}} \)) setting to optimize the learning objective defined in Eq. 5.

Results. Table 3 shows the average precision obtained for different client configurations \( \lambda \), both anomaly classes \( AP_{\text{all}} \), the global anomalies \( AP_{\text{global}} \), and the local anomalies \( AP_{\text{local}} \). For all datasets, increasing the number of clients \( \lambda > 1 \) yields an improved detection performance of the FL models. This observation holds for both anomaly classes. When comparing the FL models \( AEN_{FL} \) trained on \( \lambda \geq 4 \) clients to the baseline models \( AEN_{\text{base}} \), the FL models outperform the baselines in detecting global anomalies. This performance gap can be attributed to two effects of the different learning setups. First, the FL setup yields a high global anomaly detection performance in the non-iid and iid settings since such anomalies violate general posting patterns observable at multiple clients. Second, the baseline setup yields a low global anomaly detection performance since such anomalies tend to be learned due to the comparably higher model capacity (resulting in a low reconstruction error \( \mathcal{L}_{\text{REC}} \)). In summary, the results provide initial evidence of the benefits of learning specialized accounting ‘meta’-models from multiple clients, encouraging the learning of industry, sector, or jurisdiction-specific audit models.

**Table 3: Average precision scores obtained for all journal entry anomalies \( AP_{\text{all}} \), global anomalies \( AP_{\text{global}} \), and local anomalies \( AP_{\text{local}} \) with different number of clients \( \lambda \in [1, 4, 8] \) and non-iid (\( \mathcal{D}_{\text{niid}} \)) and iid (\( \mathcal{D}_{\text{id}} \)) data configurations.**

| Model |
|-------|
| Data  |
| \( \nu_{\text{max}} \) |
| \( \lambda \) |
| \( AP_{\text{all}} \) |
| \( AP_{\text{global}} \) |
| \( AP_{\text{local}} \) |

**Figure 4:** Average precision scores obtained in the iid data setting for the global \( AP_{\text{global}} \) and local \( AP_{\text{local}} \) anomalies with \( \lambda = 8 \) federated clients when evaluating the grid of distinct DPL configurations over all payment datasets. The scores of a particular \( \nu_{\text{max}} \) and \( \kappa \) combination denotes the mean over five distinct public \( \theta^p \) and private \( \theta^p \) model parameter initializations.
We evaluate the SL setup by comparing the learned FL model’s anomaly detection capabilities over distinct cut-layer \( c \in \{1, 2, \ldots, 7\} \) configurations. The AENFL are trained with \( \nu_{\text{max}} = 100, \kappa = 0.0, \) and \( \lambda \in [4, 8] \) federated clients (disregarding DPL setup variations).

**Results.** Figure 5 shows the average precision scores obtained for global anomalies \( \text{AP}_{\text{global}} \) and local anomalies \( \text{AP}_{\text{local}} \). Given the cut-layer configurations \( \ell \). The obtained results demonstrate that reducing the number of shared public model parameters \( \theta = (\theta^g \cup \theta^f) \) decreases the anomaly detection performance. This observation holds in particular for the global anomalies and might not come as a surprise since such anomalies often violate general accounting principles observable at multiple clients. Therefore, the detection of global anomalies unusual for a specific industry, sector, or jurisdiction constitutes the idea of knowledge sharing. In summary, it becomes evident that audit firms can adjust the degree of parameter sharing based on their client’s consent to detect industry, sector, or jurisdiction-specific global anomalies.

### 6 SUMMARY

In this work, we proposed a federated learning framework to learn audit models in a privacy-preserving manner from real-world accounting data. Based on three deliberately designed experimental setups, we demonstrated the framework’s ability to detect anomalies using knowledge learned from multiple clients. The experimental results provide initial evidence that such a learning setup offers a promising avenue for the learning ‘meta’ audit models to provide enhanced assurance services. Especially, when auditing a particular industry, sector, or jurisdiction. We believe that the future analysis of such models and learned representations will provide novel insights for audit firms that benefit their clients without sharing confidential data.

### ACKNOWLEDGMENTS

We thank the anonymous members of Deutsche Bundesbank for their valuable review and remarks. The research of Marco Schreyer was funded by the Mobility grant for doctoral students of the University of St.Gallen (Mobi.Doc) under project no. 1031606. Opinions expressed in this work are solely those of the authors and do not necessarily reflect the view of the Deutsche Bundesbank.

### REFERENCES

[1] 2014. AICPA Code of Professional Conduct. American Institute of Certified Public Accountants (AICPA).

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In ACM SIGSAC Conference on Computer and Communications Security. 308–318.

[3] Michael G Alles. 2015. Drivers of the Use and Facilitators and Obstacles of the Evolution of Big Data by the Audit Profession. Accounting Horizons 29, 2 (2015), 439–449.

[4] Deniz Appelbaum. 2016. Securing Big Data Provocateur for Auditors: The Big Data Provocateur Black Box as Reliable Evidence. Journal of Emerging Technologies in Accounting 13, 1 (2016), 17–36.

[5] Argyris Argyrou. 2012. Auditing Journal Entries Using Self-Organizing Map. In Proceedings of the Eighteenth Americas Conference on Information Systems.

[6] Stephen Bay, Krishna Kumaraswamy, Markus G. Andrelle, Rohit Kumar, David M Steier, Almaden Blvd, and San Jose. 2006. Large Scale Detection of Irregularities in Accounting Data. In Sixth International Conference on Data Mining. IEEE, 75–86.

[7] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35, 8 (2013), 1798–1828.
[58] Sutapat Thiprungsri and Miklos A Vasarhelyi. 2011. Cluster Analysis for Anomaly Detection in Accounting Data: An Audit Approach. *International Journal of Digital Accounting Research* 11 (2011).

[59] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. 1096–1103.

[60] Andrew C Yao. 1982. Protocols for Secure Computations. In *23rd Annual Symposium on Foundations of Computer Science*. IEEE, 160–164.

[61] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. 2021. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[62] Kyunghee Yoon, Lucas Hoogduin, and Li Zhang. 2015. Big Data as Complementary Audit Evidence. *Accounting Horizons* 29, 2 (2015), 431–438.

[63] Ashkan Yousefpour, Igor Shalov, Alexandre Sahlayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. 2021. Opacus: User-Friendly Differential Privacy Library in PyTorch. *arXiv preprint arXiv:2109.12298* (2021).

[64] Manal M Yunis, Raed El-Khalil, and Miray Ghanem. 2021. Towards a Conceptual Framework on the Importance of Privacy and Security Concerns in Audit Data Analytics. (2021).

[65] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*. PMLR, 7252–7261.

[66] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. 2017. Efficient private ERM for smooth objectives. *arXiv preprint arXiv:1703.09947* (2017).

[67] Yali Zheng, Zhenyu Wu, Ye Yuan, Tianlong Chen, and Zhangyang Wang. 2020. PCAL: A privacy-preserving intelligent credit risk modeling framework based on adversarial learning. *arXiv preprint arXiv:2010.02529* (2020).

[68] Mario Zupan, Verica Budimir, and Svjetlana Letinic. 2020. Journal entry anomaly detection model. *Intelligent Systems in Accounting, Finance and Management* 27, 4 (2020), 197–209.