When FinTech Meets Privacy: Securing Financial LLMs with Differential Private Fine-Tuning

Sichen Zhu*, Hoyeung Leung*, Xiaoyi Wang†, Jia Wei‡, Honghui Xu‡§
*Georgia Institute of Technology, Atlanta, GA, USA
†Sichuan University of Media and Communications, Chengdu, Sichuan, China
‡Kennesaw State University, Marietta, GA, USA
§Corresponding author: Email: hxu10@kennesaw.edu

***Abstract*—The integration of Large Language Models (LLMs) into financial technology (FinTech) has revolutionized the analysis and processing of complex financial data, driving advancements in real-time decision-making and analytics. With the growing trend of deploying AI models on edge devices for financial applications, ensuring the privacy of sensitive financial data has become a significant challenge. To address this, we propose DPFinLLM, a privacy-enhanced, lightweight LLM specifically designed for on-device financial applications. DPFinLLM combines a robust differential privacy mechanism with a streamlined architecture inspired by state-of-the-art models, enabling secure and efficient processing of financial data. This proposed DPFinLLM can not only safeguard user data from privacy breaches but also ensure high performance across diverse financial tasks. Extensive experiments on multiple financial sentiment datasets validate the effectiveness of DPFinLLM, demonstrating its ability to achieve performance comparable to fully fine-tuned models, even under strict privacy constraints.**

***Index Terms*—FinTech, Differential Privacy, Financial LLM**

I. Introduction

The proliferation of LLMs has revolutionized natural language understanding and generation, driving significant advancements in the FinTech sector. These models excel in processing complex financial data, enabling applications such as sentiment analysis [1], risk management [2], fraud detection [3], and credit scoring [4]. By providing actionable insights and facilitating real-time decision-making, LLMs have become indispensable tools in modern financial services, empowering FinTech solutions to deliver smarter, faster, and more secure financial operations [5].

Recently, there has been a growing trend toward deploying AI models on edge devices for financial applications [6]. In light of this, on-device financial LLMs will offer several advantages, including real-time data processing, reduced dependency on cloud services, and enhanced user data privacy—critical components in the evolving FinTech landscape. However, this trend also introduces significant challenges. On-device models [7], [8], [9], [10], [11] must achieve a delicate balance between computational efficiency, privacy protection, and task performance, making their design and training particularly complex. Addressing these challenges is essential to advancing the integration of LLMs into secure, efficient, and innovative FinTech ecosystems.

The sensitive nature of financial data amplifies the challenges associated with deploying on-device financial LLMs. Membership inference attacks [12] and model inversion attacks [13] on these AI models pose significant risks, enabling unauthorized access to private financial information. While existing differential privacy techniques have been explored across three key phases, input dataset preparation [14], model training [15], and model output generation [12], their application to on-device financial LLMs remains under-researched. Bridging this gap is crucial to ensuring the secure and effective deployment of financial LLMs on edge devices, safeguarding sensitive data while maintaining financial LLMs' performance.

To address these challenges, we introduce DPFinLLM, a novel on-device financial large language model that integrates a lightweight architectural design with a robust differential privacy mechanism. DPFinLLM employs a privacy-enhanced training pipeline to safeguard sensitive financial data while maintaining high performance across various financial tasks. Drawing inspiration from state-of-the-art models like Llama2 and ChatGLM2, DPFinLLM features a streamlined architecture optimized for edge devices. By incorporating Low-Rank Adaptation (LoRA) for fine-tuning, the model achieves computational efficiency and supports task-specific optimization with minimal resource requirements. The key contributions of this paper are summarized as follows:

- We propose DPFinLLM, a privacy-enhanced on-device financial LLM that integrates differential privacy techniques with a lightweight architectural design tailored for edge devices.
- A robust differential privacy mechanism is incorporated into the fine-tuning process to protect sensitive financial data from privacy breaches.
- Comprehensive experiments on multiple financial sentiment datasets validate the effectiveness of DPFinLLM, demonstrating performance comparable to baseline models even under strict privacy constraints.

The remainder of this paper is structured as follows: Section II reviews existing research on financial LLMs and privacy-preserving techniques. Section III details the architectural design and privacy-preserving training framework of DPFinLLM. Section IV presents the experimental setup and results, and Section V concludes with insights and directions for future research.

II. Related Works

This section will conclude the related work of financial language models and review the current mainstream privacy-preserving learning approaches.

A. Financial Large Language Models

Recently, financial language models have demonstrated remarkable capabilities in handling complex tasks in the financial sector, including sentiment analysis [16], risk management [2], financial fraud detection [3], and credit scoring [4]. By leveraging their predictive power, users in the economic domain can make more informed decisions, enabling better planning and strategic execution. Modern fine-tuned LLMs, such as FinGPT [17] and FinBERT [18], have been specifically developed to handle natural language processing tasks in financial datasets, showcasing exceptional domain-specific adaptability. These models excel at processing both structured and unstructured data [19] from various sources, including APIs, web scraping tools, and direct database access. Their ability to integrate real-time data streams into model training provides outputs that reflect the most current market conditions or issue statuses, offering users actionable insights in a dynamic financial landscape [20]. As financial language models continue to evolve, a notable trend is emerging toward the development of on-device financial LLMs. However, designing on-device financial LLMs presents considerable challenges. Unlike cloud-based models, which can leverage extensive computational resources, on-device models must operate within the constraints of limited processing power, memory, and energy. Overcoming these hurdles will be crucial to unlocking the full potential of on-device financial LLMs and transforming the way financial insights are generated and utilized.

B. Privacy-Preserving Learning Mechanisms

Privacy-preserving approaches address critical AI cybersecurity challenges by implementing mechanisms across three key phases: input dataset preparation, model training, and model output generation. (1) In the input dataset preparation phase, privacy is safeguarded by injecting noise (e.g., Gaussian or Laplace) into dataset features [21], [14], replacing sensitive text with anonymous tokens or artificial labels, and encrypting data to minimize leakage risks [22]. (2) During the model training process, some techniques ensure privacy by clipping gradients to limit individual data point influence and adding Gaussian noise to gradients for differential privacy [15], [23]. (3) In the model output generation phase, methods introduce noise to prompts or outputs to protect sensitive information, followed by coherence refinement to maintain utility. Additionally, output filtering can replace sensitive terms with anonymized equivalents to prevent disclosure [12]. While these strategies effectively address privacy concerns in traditional large language models, there is a notable lack of research investigating data privacy mechanisms specifically for on-device financial LLMs. Developing privacy-preserving solutions tailored for on-device models is critical, as such systems face unique challenges, including constrained computational resources and heightened sensitivity to data privacy breaches.

In this paper, we propose an on-device differential privacy-enhanced financial LLM (called DPFinLLM), tackling two critical challenges: designing a lightweight architecture suitable for resource-constrained edge devices and incorporating a differential privacy mechanism during model training to protect users' sensitive data effectively.

III. DPFinLLM

The proposed model, DPFinLLM, is an on-device differential privacy-enhanced financial large language model designed to address two primary challenges: creating a lightweight architecture for deployment on resource-constrained edge devices and integrating a robust differential privacy mechanism to safeguard sensitive financial data. DPFinLLM employs a transformer-based architecture with modifications inspired by Llama2. For fine-tuning, the model leverages LoRA, a parameter-efficient technique that reparameterizes weight updates during training, significantly reducing memory and computational costs. To ensure privacy preservation, the model incorporates an (ϵ, δ)-differential privacy framework during training. The framework limits the influence of individual samples on parameter updates using gradient clipping and adds Gaussian noise to batch gradients. By fine-tuning DPFinLLM with a well-structured loss function and optimizing hyperparameters such as privacy leakage bounds (ϵ, δ), gradient norm limits, and batch sizes, the model ensures robust privacy protection while maintaining high performance for financial-specific tasks. This architecture and training approach make DPFinLLM suitable for secure and efficient deployment in sensitive financial environments.

A. Lightweight Financial LLM Design and Fine-Tuning

We denote the basic LLM learning process as $\pi_\theta \in \Pi$, parameterized by $\theta \in \Theta$ through a neural network. The input prompt $\mathbf{x}$ is tokenized into tokens from a predetermined vocabulary $\mathcal{V}$, represented as $\mathbf{x} = [x_1, \dots, x_n], x_i \in \mathcal{V}$. The output sequence is denoted as $\mathbf{y}$. The generated outputs $\mathbf{y}$, given input $\mathbf{x}$, are expressed as:

$$\pi_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} \pi\left(y_t | x_1, x_2, \dots, x_n, , , y_1, y_2, \dots, y_{t-1}\right), \quad (1)$$

where y_i is sampled from the conditional probability distribution $\pi_\theta(\cdot|\mathbf{x})$, and the next token y_t is generated based on all previously generated tokens $(y_1, y_2, \dots, y_{t-1})$.

We utilize a transformer-based LLM, where the core component is the attention function, mapping a query and a set of key-value pairs to an output:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2)$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the size of the hidden dimensions. The factor $1/\sqrt{d_k}$ serves as a scaling term. To capture semantic and contextual information from input tokens with varying emphases at different positions, we employ a multi-head attention mechanism [24]:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \quad (3)$$

where $\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$, with parameter matrices W_i^Q, W_i^K, W_i^V, and W^O.

To pursue a lightweight architecture for the LLM, we adopt a modified attention mechanism inspired by Llama2 [25], replacing traditional multi-head attention to improve performance and reduce memory costs as context window and batch sizes increase in larger models. Grouped-query attention [26] enables the sharing of key and value projection matrices (W_i^K, W_i^V) across multiple heads without compromising model performance. Furthermore, we incorporate RMSNorm [27] for layer normalization, with the activation function SwiGLU [28], to mitigate the internal covariate shift issue often encountered in vanilla neural networks.

Though trained on generic datasets, a lightweight architecture for financial LLMs can effectively adapt to specific domains by being fine-tuned on internet-scale financial data from diverse tasks. This adaptation is enabled by leveraging the basic LLM architecture. For positional embeddings, we employ rotary positional embeddings [29], an interpretable method that integrates relative positional information into the rotation of context representations, enhancing the model's ability to capture sequential dependencies.

For the fine-tuning of our lightweight LLM, we utilize the Low-Rank Adaptation (LoRA) [30] method alongside instruction tuning. LoRA is a parameter-efficient approach for fine-tuning pre-trained LLMs to specific tasks. It operates under the assumption that updates to model weights during fine-tuning have a low "intrinsic rank." For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, the weight update during fine-tuning is reparameterized as:

$$W_0 + \Delta W = W_0 + BA, \quad (4)$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$. The rank r of ΔW controls the number of trainable parameters and is typically small, $r \ll \min(d, k)$. During fine-tuning, only A and B are updated, while the pre-trained weight W_0 remains frozen. For an input vector x and an output vector h, the forward pass is computed as:

$$h = (W_0 + \Delta W)x = W_0x + \Delta Wx = W_0x + BAx, \quad (5)$$

where ΔWx is further scaled by α/r, allowing the rank of ΔW to remain intact while scaling the impact of ΔW without increasing the number of training parameters. LoRA significantly reduces memory and computational costs by avoiding expensive matrix multiplications, thereby enabling the fine-tuning of LLMs in a computation- and parameter-efficient manner.

B. DP Mechanism for Secure Financial Model Training

To protect sensitive financial data, we propose training the financial model using an (ϵ, δ)-differential privacy (DP) mechanism [31]. Specifically, in this (ϵ, δ)-DP training framework, the influence of individual samples on parameter updates is limited during training. For training samples $x_1, \ldots, x_N$, the loss function with respect to θ, minimized during neural network training, is expressed as:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i). \quad (6)$$

For a random batch of training samples $\mathcal{B}$, the gradient for each individual sample $x_i \in \mathcal{B}$ at training step t is computed as $\mathbf{g}t(x_i) = \nabla\theta_t\mathcal{L}(\theta_t, x_i)$. To constrain the influence of any single sample, the gradient is clipped as follows:

$$\overline{\mathbf{g}}_t(x_i) = \frac{\mathbf{g}_t(x_i)}{\max\left(1, , \frac{||\mathbf{g}_t(x_i)||_2}{C}\right)}, \quad (7)$$

where C is the gradient norm bound, preventing the model from being overly influenced by any single training sample. To further protect against memorization of training data, Gaussian noise is added to the average gradient over the batch $\mathcal{B}$:

$$\tilde{\mathbf{g}}_t = \frac{1}{L}\left(\sum_{i=1}^{L}\overline{\mathbf{g}}_t(x_i) + \mathcal{N}\left(0, \sigma^2C^2\mathbf{I}\right)\right), \quad (8)$$

where σ is the noise scale, and L is the lot size—the number of samples whose gradients are computed and averaged. To limit memory consumption, the batch size $\mathcal{B}$ is typically much smaller than the lot size L, as gradient clipping and noising are performed per sample. Finally, a standard gradient descent step updates θ as:

$$\theta_{t+1} = \theta_t - \eta_t\tilde{\mathbf{g}}_t, \quad (9)$$

where η_t is the learning rate.

From an engineering perspective, a larger batch size can improve convergence during training. However, increasing the batch size also raises the privacy cost, as reflected in Theorem 1, derived from Definition 1.

Definition 1: A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$, with domain $\mathcal{D}$ and range $\mathcal{R}$, satisfies (ϵ, δ)-differential privacy [32] if, for any two adjacent inputs $d, d' \in \mathcal{D}$ and any subset of outputs $S \subseteq \mathcal{R}$, it holds that:

$$\Pr[\mathcal{M}(d)] \leq e^\epsilon, \Pr[\mathcal{M}(d')] + \delta, \quad (10)$$

where $\mathcal{M}(d), \mathcal{M}(d') \in S$.

Theorem 1: There exist constants c_1 and c_2 such that, given the lot size L, sample size N, sampling probability $q = L/N$, and number of steps T, for any $\epsilon < c_1q^2T$, the DPSGD algorithm satisfies (ϵ, δ)-DP for any $\delta > 0$, provided

$$\sigma \geq c_2\frac{q\sqrt{T\log(1/\delta)}}{\epsilon}. \quad (11)$$

This theorem establishes that the algorithm is (ϵ, δ)-DP, meaning the outputs are perturbed by a factor governed by e^ϵ, ensuring indistinguishability of datasets differing by a single data point. The additive term δ slightly relaxes the constraint and is typically chosen to be smaller than $1/|d|$. The parameters ϵ and δ quantify *privacy leakage* [33], mathematically defining the privacy-preserving goal. Smaller values of ϵ and δ guarantee that differences in the input dataset minimally affect the algorithm's random output, reducing the risk of private data exposure to inference attacks.

In conclusion, we will apply the proposed differential privacy learning algorithm to fine-tune our financial LLM, as detailed in Section III-A, using the loss function in Eq. (6) and exploring various hyperparameter settings, including ϵ, δ, gradient norm bounds, and batch sizes.

IV. Experiments

This section outlines the experimental setup and presents a thorough analysis of the results, highlighting the effectiveness of our proposed DPFinLLM. The open-source codes of the experiments can be found in https://github.com/SichenZhu/DP_FinLLM.

A. Experimental Settings

The datasets, training setup, hyperparameter settings, baselines, and performance metrics are described below.

1) Datasets: We evaluate the effectiveness of our proposed DPFinLLM model using four sentiment analysis datasets from the financial domain. (1) The first dataset, FPB, contains 4,846 news entries that cover a diverse range of small and large companies, various industries, and multiple news sources. The sentiment labels in this dataset are assigned from an investor's perspective [34]. (2) The second dataset, FIQA, includes 1,213 entries sourced from financial news headlines and microblogs, with annotations for target entities, sentiment scores, and aspects [35]. (3) The third dataset, TFNS, comprises 11,931 finance-related tweets collected via the Twitter API, capturing social media sentiment [36]. (4) Lastly, NWGI is composed of 20,231 entries generated using GPT-based instructions, providing a wide range of sentiment-rich content [37]. The datasets are organized in the following way for fine-tuning of LLMs:

Instruction: "What is the sentiment of this news/tweet? Please choose an answer from negative/neutral/positive"

Input: [input] **Answer**: [output]

2) Baseline: We use two models as baselines for performance comparison. (1) Llama2 [25], is an open-source large language model (LLM) developed by Meta. Llama2 is a highly capable chat model that surpassed existing open-source chat models on most benchmarks at the time of its release. The state-of-the-art performance of FinGPT in sentiment analysis was achieved using the Llama2 model family. (2) ChatGLM2 [38], is an open-source bilingual general language model. ChatGLM2 served as the base model for Financial GPT, which also achieved SOTA performance in sentiment analysis. We compare the performance of our proposed DPFinLLM against the original Llama2-7B and ChatGLM2-6B models, both trained on a financial multi-task dataset that includes all four sentiment analysis datasets.

3) Performance Metrics: The performance evaluation metric is accuracy and three different calculations for F1-score "micro", "macro" and "weighted"). The micro F1 score counts total true positives, false negatives and false positives. The macro F1 score averages the F1 score calculated for each label. The weighted F1 score takes the weighted average F1 score from each label to take the label imbalanceness into consideration.

4) Training Setup: We adopt the same network architecture design as Llama2 and ChatGLM2 while incorporating the proposed DPFinLLM framework during the training process of the LLM, as outlined in Section III. Specifically, the base model is fine-tuned using the LoRA method described in Section III-A, with the differential privacy (DP) mechanism detailed in Section III-B integrated into the fine-tuning process. This approach is tailored to address the sentiment analysis task, ensuring both performance optimization and robust privacy protection.

5) Hyperparameter Settings: We experiment with various hyperparameter settings during training and testing to optimize model performance. Notably, for certain experiments involving the differential privacy mechanism, the parameter δ is set to $1/|d|$, where d represents the size of the training dataset.

TABLE I: Comparison Performance of Sentiment Analysis on FPB Dataset (Baselines v.s. Our DPFinLLM)

Metric	Accuracy	F1 macro	F1 micro	F1 weighted
Llama2-7B (Llama2-based)	0.46947	0.52394	0.46947	0.40989
FinGPT-llama2-mt (Llama2-based)	0.79785	0.75539	0.79785	0.77654
FinGPT v3.2 (Llama2-based)	0.86634	0.85190	0.86634	0.86371
Llama2-based DPFinLLM with DP $\epsilon = 8.0$	**0.79785**	**0.77880**	**0.79785**	**0.79524**
ChatGLM2-6B (ChatGLM2-based)	0.45462	0.47733	0.45462	0.36403
FinGPT v3.1 (ChatGLM2-based)	0.85231	0.83502	0.85231	0.85100
ChatGLM2-based DPFinLLM with DP $\epsilon = 8.0$	**0.47030**	**0.49355**	**0.47030**	**0.39147**

TABLE II: Comparison Performance of Sentiment Analysis on FIQA Dataset (Baselines v.s. Our DPFinLLM)

Metric	Accuracy	F1 macro	F1 micro	F1 weighted
Llama2-7B (Llama2-based)	0.78909	0.59376	0.78909	0.77448
FinGPT-llama2-mt (Llama2-based)	0.43636	0.45265	0.43636	0.52937
FinGPT v3.2 (Llama2-based)	0.75273	0.67581	0.75273	0.80064
Llama2-based DPFinLLM with DP $\epsilon = 8.0$	**0.80727**	**0.61251**	**0.80727**	**0.78646**
ChatGLM2-6B (ChatGLM2-based)	0.83636	0.57016	0.83636	0.80312
FinGPT v3.1 (ChatGLM2-based)	0.82909	0.73848	0.82909	0.84298
ChatGLM2-based DPFinLLM with DP $\epsilon = 2.0$	**0.83636**	**0.57164**	**0.83636**	**0.80454**

TABLE III: Comparison Performance of Sentiment Analysis on TFNS Dataset (Baselines v.s. Our DPFinLLM)

Metric	Accuracy	F1 macro	F1 micro	F1 weighted
Llama2-7B (Llama2-based)	0.38023	0.40368	0.38023	0.29781
FinGPT-llama2-mt (Llama2-based)	0.78182	0.67878	0.78182	0.75981
FinGPT v3.2	0.88986	0.86065	0.88987	0.88859
Llama2-based DPFinLLM with DP $\epsilon = 4.0$	**0.73199**	**0.60420**	**0.73199**	**0.71027**
ChatGLM2-6B (ChatGLM2-based)	0.33124	0.33976	0.33124	0.18787
FinGPT v3.1 (ChatGLM2-based)	0.88275	0.84917	0.88275	0.88214
ChatGLM2-based DPFinLLM with DP $\epsilon = 6.0$	**0.72069**	**0.55256**	**0.72069**	**0.68417**

TABLE IV: Comparison Performance of Sentiment Analysis on NWGI Dataset (Baselines v.s. Our DPFinLLM)

Metric	Accuracy	F1 macro	F1 micro	F1 weighted
Llama2-7B (Llama2-based)	0.56659	0.52173	0.56659	0.48580
FinGPT-llama2-mt (Llama2-based)	0.58636	0.59366	0.58636	0.58039
FinGPT v3.2 (Llama2-based)	0.62762	0.63837	0.62762	0.62728
Llama2-based DPFinLLM with DP $\epsilon = 2.0$	**0.57129**	**0.52893**	**0.57129**	**0.49360**
ChatGLM2-6B (ChatGLM2-based)	0.56041	0.48992	0.56041	0.44952
FinGPT v3.1 (ChatGLM2-based)	0.64072	0.64878	0.64072	0.64068
ChatGLM2-based DPFinLLM with DP $\epsilon = 8.0$	**0.56042**	**0.48894**	**0.56042**	**0.44888**

TABLE V: Zero-shot Performance of Llama2-based DPFinLLM Fine-tuned on Various Datasets

Fine-tuned on:	FPB	FIQA	TFNS	NWGI	Llama2-7B base model
FPB	-	0.40696	0.56203	0.40917	0.40989
FIQA	0.49795	-	0.37072	0.78235	0.77448
TFNS	0.65201	0.31464	-	0.29436	0.29781
NWGI	0.62803	0.47925	0.46718	-	0.48580

B. Evaluation Results on Our DPFinLLM

Trained on each dataset, our fine-tuned models—both the Llama2-based DPFinLLM and the ChatGLM2-based DPFinLLM—demonstrate superior performance on test data compared to their respective base models across all evaluation metrics, as shown in Tables I through IV. By comparing these results in these tables, we draw one conclusion that even under stringent privacy constraints, such as a small ϵ, both the Llama2-based and ChatGLM2-based DPFinLLM models exhibit substantial increases in accuracy and F1 scores on the four datasets. This demonstrates that, with appropriately configured hyperparameters, DPFinLLM can achieve performance comparable to fully fine-tuned models while ensuring robust privacy protection for the training data. Additionally, for the FPB dataset (Table I) and the TFNS dataset (Table III), the Llama2-7B-based DPFinLLM model achieves nearly a twofold improvement in each evaluation metric compared to its base model, Llama2-7B. Besides, for the FIQA dataset (Table II), the configuration "Llama2-based DPFinLLM with $\epsilon = 8.0$" significantly outperforms "FinGPT-llama2-mt," which was fully fine-tuned on all four datasets.

TABLE VI: Zero-shot Performance of ChatGLM2-based DPFinLLM Fine-tuned on Various Datasets

Fine-tuned on:	FPB	FIQA	TFNS	NWGI	ChatGLM2-6B base model
FPB	-	0.36003	0.61345	0.34909	0.36403
FIQA	0.80120	-	0.34792	0.80343	0.80312
TFNS	0.18960	0.18718	-	0.18662	0.18787
NWGI	0.46563	0.44901	0.46840	-	0.44952

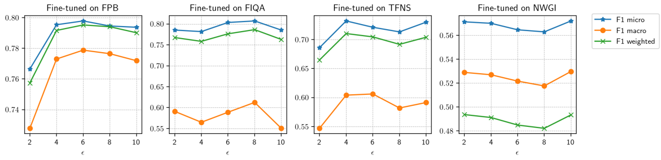

Fig. 1: DPFinLLM's Performance with Different Values of ϵ

Moreover, we conduct comprehensive experiments to evaluate the impact of different ϵ values in the privacy-preserving fine-tuning of our proposed DPFinLLM. In these experiments, ϵ is the only variable, while all other hyperparameters remain fixed. The results are summarized in Fig. 1. As shown in Fig. 1, the FPB, TFNS, and FIQA datasets exhibit a similar trend as ϵ increases: the F1 scores initially rise to a peak before declining, which indicates that our proposed DPFinLLM can protect data privacy while maintaining the performance of sentiment analysis on financial data. However, this observation highlights an important insight for hyperparameter tuning in differential privacy—relaxing the privacy constraint (i.e., increasing ϵ) does not always lead to better performance in terms of F1 scores or prediction accuracy. Interestingly, the NWGI dataset demonstrates a slightly different pattern with larger ϵ values. A potential explanation for this deviation could be the nature of the dataset: NWGI instructions are generated by ChatGPT, whereas the other three datasets are manually labeled and curated, potentially leading to differences in data characteristics and model behavior.

C. Zero-shot Performance of Our Proposed DPFinLLM

After fine-tuning the model on one dataset, we evaluate its zero-shot performance on the remaining three datasets. The weighted F1 scores are presented in Tables V and VI, where each row indicates the dataset the model was fine-tuned on, and each column corresponds to the test dataset used to assess model performance. (1) In Table V, with Llama2-7B as the base model, fine-tuning on TFNS leads to a significant increase in F1 score on the FPB test dataset, and vice versa—fine-tuning on FPB improves performance on the TFNS test dataset. Additionally, fine-tuning on FPB enhances the model's performance on both the TFNS and NWGI test datasets. For the remaining zero-shot experiments, fine-tuning on one dataset generally does not significantly compromise the model's generalization ability, except in the case where the model was fine-tuned on TFNS and evaluated on FIQA, which resulted in a notable drop in performance. (2) Similarly, in Table VI, using ChatGLM2-6B as the base model, we observe that fine-tuning on one dataset does not significantly degrade performance on the other datasets, with the exception of fine-tuning on TFNS and testing on FIQA. While the exception case highlights the ongoing challenge of balancing generalization and specialization in zero-shot settings, fine-tuning LLMs with DP still demonstrates great potential in maintaining the model's zero-shot performance on unseen datasets after achieving strong performance on the fine-tuned dataset.

V. Conclusion

In this paper, we introduce DPFinLLM, a novel privacy-preserving financial large language model designed to address the growing concern of sensitive data leakage in fine-tuning processes. By integrating differential private training mechanism into the fine-tuning pipeline, DPFinLLM ensures robust protection of sensitive financial data while maintaining competitive performance across sentiment analysis tasks. The model's lightweight architecture, inspired by state-of-the-art small LLM designs, enables efficient deployment on resource-constrained edge devices. Extensive experiments conducted on four financial sentiment datasets validate the efficacy of DPFinLLM, demonstrating significant improvements over baseline models even under strict privacy constraints, and the experimental results also highlight DPFinLLM's ability to balance generalization and specialization, achieving superior zero-shot performance across unseen datasets. This capability underscores its potential as a versatile tool for on-device financial applications, where privacy and accuracy are paramount. To sum up, by combining the differential privacy training idea with an efficient architectural design, the proposed DPFinLLM marks a significant breakthrough in securing financial LLMs, paving the way for the broader adoption of privacy-preserving technologies in on-device financial applications.

References

[1] B. Zhang, H. Yang, T. Zhou, M. Ali Babar, and X.-Y. Liu, "Enhancing financial sentiment analysis via retrieval augmented large language models," in *Proceedings of the fourth ACM international conference on AI in finance*, 2023, pp. 349–356.

[2] C. Yang, C. Xu, and Y. Qi, "Financial knowledge large language model," *arXiv preprint arXiv:2407.00365*, 2024.

[3] P. Boulieris, J. Pavlopoulos, A. Xenos, and V. Vassalos, "Fraud detection with natural language processing," *Machine Learning*, vol. 113, no. 8, pp. 5087–5108, 2024.

[4] M. Sanz-Guerrero and J. Arroyo, "Credit risk meets large language models: Building a risk indicator from loan descriptions in p2p lending," *arXiv preprint arXiv:2401.16458*, 2024.

[5] Z. Xue, L. Li, S. Tian, X. Chen, P. Li, L. Chen, T. Jiang, and M. Zhang, "Domain knowledge is all you need: A field deployment of llm-powered test case generation in fintech domain," in *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, 2024, pp. 314–315.

[6] W. Hassan and H. Mohamed, "Applications of federated learning in ai, iot, healthcare, finance, banking, and cross-domain learning," in *Artificial Intelligence Using Federated Learning*. CRC Press, 2024, pp. 175–195.

[7] J. Xu, Z. Li, W. Chen, Q. Wang, X. Gao, Q. Cai, and Z. Ling, "On-device language models: A comprehensive review," *arXiv preprint arXiv:2409.00088*, 2024.

[8] W. Chen, Z. Li, Z. Guo, and Y. Shen, "Octo-planner: On-device language model for planner-action agents," *arXiv preprint arXiv:2406.18082*, 2024.

[9] W. Chen and Z. Li, "Octopus v2: On-device language model for super agent," *arXiv preprint arXiv:2404.01744*, 2024.

[10] ——, "Octopus v3: Technical report for on-device sub-billion multimodal ai agent," *arXiv preprint arXiv:2404.11459*, 2024.

[11] ——, "Octopus v4: Graph of language models," *arXiv preprint arXiv:2404.19296*, 2024.

[12] Z. Zhang, C. Gong, Y. Cai, Y. Yuan, B. Liu, D. Li, Y. Guo, and X. Chen, "No privacy left outside: On the (in-) security of tee-shielded dnn partition for on-device ml," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 3327–3345.

[13] T. Nayan, Q. Guo, M. Al Duniawi, M. Botacin, S. Uluagac, and R. Sun, "{SoK}: All you need to know about {On-Device}{ML} model extraction-the gap between research and practice," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 5233–5250.

[14] K. D. Martin and J. Zimmermann, "Artificial intelligence and its implications for data privacy," *Current Opinion in Psychology*, p. 101829, 2024.

[15] Y. Liu, L. Xiong, Y. Liu, Y. Gu, R. Liu, and H. Chen, "Dpdr: Gradient decomposition and reconstruction for differentially private deep learning," *arXiv preprint arXiv:2406.02744*, 2024.

[16] J. Delgadillo, J. Kinyua, and C. Mutigwe, "Finsosent: Advancing financial market sentiment analysis through pretrained large language models," *Big Data and Cognitive Computing*, vol. 8, no. 8, p. 87, 2024.

[17] H. Yang, X.-Y. Liu, and C. D. Wang, "Fingpt: Open-source financial large language models," *arXiv preprint arXiv:2306.06031*, 2023.

[18] A. H. Huang, H. Wang, and Y. Yang, "Finbert: A large language model for extracting information from financial text," *Contemporary Accounting Research*, vol. 40, no. 2, pp. 806–841, 2023.

[19] H. Li, H. Gao, C. Wu, and M. A. Vasarhelyi, "Extracting financial data from unstructured sources: Leveraging large language models," *Journal of Information Systems*, pp. 1–22, 2023.

[20] H. Zhao, Z. Liu, Z. Wu, Y. Li, T. Yang, P. Shu, S. Xu, H. Dai, L. Zhao, G. Mai *et al.*, "Revolutionizing finance with llms: An overview of applications and insights," *arXiv preprint arXiv:2401.11641*, 2024.

[21] A. Majeed and S. O. Hwang, "When ai meets information privacy: The adversarial role of ai in data sharing scenario," *IEEE Access*, 2023.

[22] L. Yang, M. Tian, D. Xin, Q. Cheng, and J. Zheng, "Ai-driven anonymization: Protecting personal data privacy while leveraging machine learning," *arXiv preprint arXiv:2402.17191*, 2024.

[23] J. Fu, Z. Chen, and X. Ling, "Sa-dpsgd: Differentially private stochastic gradient descent based on simulated annealing," *arXiv preprint arXiv:2211.07218*, 2022.

[24] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[25] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[26] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, "Gqa: Training generalized multi-query transformer models from multi-head checkpoints," *arXiv preprint arXiv:2305.13245*, 2023.

[27] B. Zhang and R. Sennrich, "Root mean square layer normalization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[28] N. Shazeer, "Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[29] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[30] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

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

[32] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[33] X. Li, F. Tramer, P. Liang, and T. Hashimoto, "Large language models can be strong differentially private learners," *arXiv preprint arXiv:2110.05679*, 2021.

[34] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, pp. 782–796, 2014.

[35] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, and A. Balahur, "Www'18 open challenge: financial opinion mining and question answering," in *Companion proceedings of the the web conference 2018*, 2018, pp. 1941–1942.

[36] N. Magic, "Twitter financial news sentiment." https://huggingface.co/datasets/zeroshot/twitter-financialnews-sentiment, 2022.

[37] H. Yang, "Data-centric fingpt. open-source for open finance," .https://github.com/AI4Finance-Foundation/FinGPT, 2023.

[38] T. GLM, "Chatglm: A family of large language models from glm-130b to glm-4 all tools," 2024.