Federated Learning with Efficient Local Adaptation for Realized Volatility Prediction

Anonymous authors
Paper under double-blind review

Abstract

Financial markets are inherently complex, with private trading activities distributed across various exchanges and platforms, leading to isolated datasets and fragmented data sources. Learning from limited local data leads to inaccurate realized volatility prediction due to incomplete representations of market dynamics. Federated learning (FL) can foster collaborative insights while ensuring privacy and regulatory compliance across diverse trading platforms. However, heterogeneity in datasets and dynamic participation making FL for financial markets slow and inaccurate. To address this issue, we propose Federated Learning with Efficient Local Adaptation (FLELA). The key idea is to enhance the local model with probabilistic techniques, including local linearization of the global model and a crucial optimization step to fine-tune parameters, so each participants can apply enhanced local model to achieve higher accuracy. Through extensive experimental evaluations, FLELA consistently outperforms existing federated learning algorithms, demonstrating superior predictive accuracy and efficiency in realizing volatility prediction. Even in the face of significant data fragmentation across massive trading venues, the proposed FLELA can achieve mean loss of 7.358×10^{-5}, VaR95% of 2.284×10^{-4}, and CVaR95% of 3.978×10^{-4} in merely five rounds of FL, which is one order better than the state-of-the-art FL approaches, underscoring its efficacy and superiority.

1 Introduction

Predicting realized volatility within the realm of deep hedging is a critical task in financial forecasting, crucial for effective risk management and strategic investment decisions. Financial markets inherently produce fragmented, asynchronous data across multiple trading venues. These platforms do not release their data to any third party, due to privacy concerns, regulatory restrictions, and technical challenges Kairouz et al. (2021). The fragmentation of trading data across various exchanges or venues significantly impacts the accuracy and reliability of realized volatility predictions. When data is scattered across different platforms, the collective understanding of the market state is incomplete. Volatility prediction, which often relies on comprehensive market data to assess price movements and liquidity, can suffer from inaccuracies or biases. Each exchange may display varying liquidity levels and pricing for the same asset, leading to potential misestimations of volatility if one exchange's data does not reflect broader market conditions Otero (2002)Madhavan (2000).

Implementing a federated learning (FL) system allows each trading platform to retain control over its data while contributing to a collaborative model that predicts realized volatility. This approach addresses privacy and regulatory concerns by avoiding direct data sharing Yang et al. (2019). However, there are unique challenges in financial markets faced by FL. Trading platforms may not always be available to participate in training due to operational constraints or schedules. The timing of data updates and the frequency of trades vary across platforms, challenging the federated model training in predicting volatility Hasbrouck (2007). Varying availability, communication limitations, and regulatory restrictions lead to dynamic federated training environment, which slows down the training speed. Furthermore, the communication intervals between updates can significantly exceed the time required for local computations, reflecting the on-and-off nature of trading activity McMahan et al. (2017). It is crucial to mitigate the risk of training stagnation which often occurs when the system overly relies on trading platforms that are slow to respond.

When applying FL across multiple trading platforms, the inherent heterogeneity of local datasets and the dynamic nature of participation highlight the importance of tailoring the global model to the specific characteristics of each trading platform's data. To deliver robust performance with heterogeneous trading platform datasets and the varying participation rates of different trading platforms, we propose a novel approach that leverages the strengths of FL alongside local adaptation, i.e., Federated Learning with Efficient Local Adaptation (FLELA). Our method involves linearizing the trained global model suitable for fast local optimization. By leveraging enhanced linearized local training and adaptive strategies, FLELA demonstrates its effectiveness in local realized volatility prediction. Furthermore, we integrate inductive biases into a probabilistic framework, utilizing the Jacobian matrix of the deep neural network model as the kernel. This enables interpretable posterior inference in function space, providing precise estimates of uncertainty and predictive distributions for local realized volatility prediction Jacot et al. (2018). This allows each trading platform to quickly and effectively fine-tune the global model to better capture the unique characteristics of its local market data, thereby improving the accuracy of volatility predictions. The experimental results show that across diverse trading platform distributions and varying participation rates, FLELA consistently achieves lower mean loss, Value at Risk (VaR95%), and Conditional Value at Risk (CVaR95%) values, underscoring its versatility and robustness.

In the following sections, we summarize related work in Section 2 and describe the fragmented financial markets and our problem formulation in Section 3. We explain our proposed method for efficient local adaptation using probabilistic frameworks in Section 4. We then present empirical evaluations of our approach in Section 5, discuss the implications of our findings, and explore avenues for future research in Section 6.

2 Related Work

In the context of financial markets, particularly when addressing the challenge of predicting realized volatility using order book data, the concept of FL can be particularly relevant due to the decentralized nature of data acquisition Banabilah et al. (2022). Order books in financial markets, which record buy and sell orders for securities, represent a dynamic and fragmented data environment, i.e., data islands, that can benefit from FL approaches Hasbrouck (2007).

The existing FL methods face specific limitations when applied to realized volatility prediction in financial markets. The high heterogeneity, rapid data changes, and need for timely updates in financial environments necessitate more adaptive and efficient FL methods. By limiting the impact of local updates that deviate significantly from the global model, FedProx Li et al. (2020) introduces a proximal term to the local objective function to stabilizes the optimization process. Financial markets exhibit significant heterogeneity in trading activities and market conditions across different exchanges and platforms Arthur et al. (2018)Cantillon & Yin (2011). Although FedProx mitigates some effects of heterogeneity, its proximal term may not fully capture the complex, dynamic nature of financial data. The proximal term can slow down convergence, which is critical in high-dynamic trading scenarios where rapid model updates are essential. While SCAFFOLD Karimireddy et al. (2019) introduces control variates to correct the drift in local updates, ensuring better alignment with the global model, the rapidly changing nature of financial data can still lead to significant misalignments between local and global models, affecting prediction accuracy Boukherouaa et al. (2021). Although FedPer Arivazhagan et al. (2019) allows each client to have a personalized model by decoupling the shared global parameters from the client-specific local parameters, the high variability and unpredictability in financial markets may require frequent adjustments to the personalized models, complicating the learning process. Managing and updating personalized models for a large number of participants can be resource-intensive, potentially limiting the scalability of FedPer in large-scale financial networks.

Recent advancements have expanded our understanding of DNN behavior, revealing that infinitely wide DNNs behave similarly to their associated Taylor expansions around initialization Chizat et al. (2019). This analysis was extended to finite-width DNNs, demonstrating similarities to linear models during training Seleznova & Kutyniok (2022). Further research investigates the inductive biases of linearized neural networks, finding that they can effectively summarize full network functions Maddox et al. (2021). These insights inspire our research, addressing the challenge in federated learning (FL) where the global model may not capture the unique characteristics of each trading platform's local data, resulting in suboptimal local volatility predictions.

The proposed FLELA aims to refine the global model through adaptive local training, enhancing accuracy for local trading platforms.

3 Fragmented Financial Markets

3.1 Background

In financial markets, private trading occurs across numerous exchanges and platforms, leading to the creation of isolated datasets. Each platform maintains its own transaction and order book data, reflecting buy and sell orders and their execution. This fragmentation offers a partial view of market activity for any given asset, with significant variations in prices and order depth across platforms Hasbrouck (2007).

The order book is a vital tool for traders, providing insights into short-term trading decisions by displaying order imbalances and potential support and resistance levels for a stock. Realized volatilities tend to increase when directional movements become more frequent, reflecting heightened market activity and uncertainty. Trading data represents executed transactions in the market, offering insights into market dynamics such as price movements and trading volumes. Predicting short-term realized volatility is essential for risk management and trading strategies. By analyzing order book and trade data within fixed time intervals, we aim to forecast future volatility levels, enabling better decision-making and risk mitigation.

Extracting insights from order book data is crucial for understanding market dynamics and assessing stock value. Metrics such as the bid-ask spread, weighted average price, and volume-related metrics provide valuable information about market liquidity and potential volatility. FL is desirable for all trading platforms because it enables them to leverage a global model that incorporates diverse data sources, leading to more robust and accurate local predictions while preserving data privacy and confidentiality.

3.2 Problem Formulation

Consider a distributed dataset consisting of n data sample pairs $\{\mathbf{x}_i, y_i\}_{i=1}^n$ across $|E|$ trading platforms. Each data sample pair represents features extracted from order book and trading data, with $\mathbf{x}_i$ denoting the feature vector and y_i representing the corresponding label, which is the volatility. There are 363 features for each sample generated from order book and trading data, capturing essential market dynamics such as bid-ask spreads, price movements, and trading volumes.

We denote the local dataset of the c-th trading platform as P_c, which contains n_c training samples. The union of all local datasets from each trading platform, $P_1 \cup P_2 \cup \cdots \cup P_{|E|}$, encompasses the entire dataset, ensuring that each sample belongs to exactly one trading platform's dataset. For trading platform c, the labels $\{y_i\}_{i \in P_c}$ represent the volatility levels observed in the corresponding platform's trading data. These volatility labels are used as the ground truth for training the predictive model.

We aim to develop a predictive model, represented by a deep neural network function f, which maps an input feature vector $\mathbf{x}$ to an output volatility prediction y. The model is trained using the distributed dataset across multiple trading platforms, leveraging the features extracted from order book and trading data to predict future volatility levels accurately. The local objective function for trading platform c is defined as

$$\underset{\boldsymbol{w}}{\textbf{minimize}} \quad \boldsymbol{L_c(w)} = \frac{1}{2} \sum_{i \in P_c} \left(f(\boldsymbol{x_i}, \boldsymbol{w}) - y_i\right)^2 \quad \text{for } c = 1, \cdots, |E|. \tag{1}$$

Meanwhile, the global objective function, aggregating the local objectives across all trading platforms, is given by

$$\underset{\boldsymbol{w}}{\textbf{minimize}} \quad \boldsymbol{L(w)} = \frac{1}{|E|} \sum_{c=1}^{|E|} \boldsymbol{L_c(w)}. \tag{2}$$

This formulation underscores the decentralized nature of the data and the collaborative effort involved in training a neural network model across multiple trading platforms.

4 Federated Learning with Efficient Local Adaptation (FLELA)

In this section, we propose Federated Learning with Efficient Local Adaptation (FLELA) to address the challenges of heterogeneous local datasets and dynamic participation in financial markets. While the global objective function in (2) aims to capture general patterns across all trading platforms, it may not fully encapsulate the unique characteristics of each local dataset, leading to suboptimal performance for individual platforms in (1). FLELA enables each trading platform to adapt the globally trained model to its specific local data. Initially, trading platforms collaboratively train a global model, ensuring data privacy and leveraging collective knowledge. Following this, each platform fine-tunes the global model through adaptive local training, adjusting the parameters to better fit local market data. This approach enhances the accuracy of volatility predictions by combining the strengths of FL with tailored local adaptations, effectively addressing the heterogeneity of local datasets and the dynamic nature of the financial markets.

4.1 Federated Training Procedure

The initial values of the weights play a crucial role in the efficiency of the training process. Arbitrary initialization methods can impede training progress, potentially leading to slow convergence or even stagnation Xie et al. (2017). To ensure stable and effective training, it's essential to maintain consistent variance in the activation distributions as the network deepens. The initial weights are drawn from a Gaussian distribution with a mean of zero and a standard deviation that is inversely proportional to the square root of the number of input units, which is expressed as $\boldsymbol{w^0} \sim \mathcal{N}\left(0, 1/\sqrt{n_{\text{in}}}\right)$, where $\boldsymbol{w^0}$ represents the initial weight vector, and n_{in} denotes the number of input units feeding into the layer. Drawing weights from a distribution tailored to the network architecture helps maintain a balanced variance in the activation distributions, which helps prevent issues such as vanishing or exploding gradients, promoting smoother gradient flow and more stable training dynamics.

Each training round unfolds in a dynamically evolving environment where the participation of trading platforms fluctuates unpredictably. The subset of trading platforms engaging in FL training, denoted as $S^t \subseteq |E|$, remains uncertain and is subject to variation. To mirror the dynamic participation characteristic of real-world scenarios, we simulate the participation set S^t by sampling it from a predefined distribution, where we explore Exponential, Geometric, Gamma, and Chi-square in this work.

Upon determining the active participants for round t, the current global model $\boldsymbol{w^t}$ is distributed to the selected trading platforms in S^t. These platforms then initialize their local models for the training round by

$$\{\boldsymbol{w^t_{c,0}} = \boldsymbol{w^t}\}_{c \in S^t}, \tag{3}$$

where $\boldsymbol{w^t_{c,0}}$ represents the initial local model weights for trading platform c at the onset of round t. This approach ensures that all participating trading platforms commence the round with identical copies of the global model, fostering collaboration within the dynamic participation environment. For the local training on trading platform c, corresponding to financial market data, the k-th step of updating the model is formulated as

$$\boldsymbol{w^t_{c,k+1}} = \boldsymbol{w^t_{c,k}} - \alpha_l \boldsymbol{\nabla L_c}(\boldsymbol{w^t_{c,k}}), \tag{4}$$

where α_l represents the local learning rate, tailored to the specific dynamics of each platform. The local training process extends over K iterations, resulting in the final local model

$$\boldsymbol{w^t_{c,K}} = \boldsymbol{w^t} - \sum_{k=1}^{K} \alpha_l \boldsymbol{\nabla L_c}(\boldsymbol{w^t_{c,k}}), \tag{5}$$

which integrates weighted gradient descents across all local steps. The discrepancy between the local and global models after K iterations is quantified by

$$\triangle \boldsymbol{w^t_c} = \boldsymbol{w^t_{c,K}} - \boldsymbol{w^t}, \tag{6}$$

illustrating the divergence of each trading platform's model from the initial global parameters in the context of financial market data. The aggregation of these local updates to form the next iteration of the global

model is governed by

$$\boldsymbol{w^{t+1}} \leftarrow \boldsymbol{w^t} + \frac{\alpha_g^t}{|S^t|} \sum_{c \in S^t} \triangle \boldsymbol{w_c^t}, \tag{7}$$

where the contribution of each local model is normalized by the number of participating trading platforms and adjusted by the global learning rate α_g^t in the round t, ensuring an equitable update based on collective learning within the financial market context. This step marks the conclusion of the t-th round, preparing for subsequent rounds of federated learning in the dynamic environment of financial markets.

4.2 Local Adaptation Procedure

The global model $\boldsymbol{w}^*$, obtained after FL training, may not be fully optimized or may exhibit poor local performance due to the diverse nature of local datasets and the dynamic participation. Nonetheless, it serves as the baseline for adaptive local training. To derive the local adaptive training strategy, we consider a given neural network model function f. We can approximate f around the trained model parameters $\boldsymbol{w}^*$ using a Taylor expansion

$$f(x; \boldsymbol{w}) \approx f(x; \boldsymbol{w}^*) + J_{\boldsymbol{w}^*}(\boldsymbol{x})^T (\boldsymbol{w} - \boldsymbol{w}^*), \tag{8}$$

where $J_{\boldsymbol{w}^*}(\boldsymbol{x})$ denotes the Jacobian matrix of partial derivatives of f with respect to the model parameters at $\boldsymbol{w}^*$, with dimensions $p \times |P_c|$. This Jacobian represents the sensitivity of the output with respect to changes in the model parameters near $\boldsymbol{w}^*$.

We formulate the probabilistic model governing the output y, given input features x extracted from order book and trading data, and model parameters $\boldsymbol{w}$ as

$$p(y \,|\, x, \boldsymbol{w}) = \mathcal{N}\left(f(\boldsymbol{x}; \boldsymbol{w}), \sigma_c^2\right) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(y - f(\boldsymbol{x}; \boldsymbol{w}))^2}{2\sigma_c^2}}, \tag{9}$$

where σ_c^2 represents the variance associated with the Gaussian noise, capturing the inherent uncertainty and noise in the model predictions of volatility. This distribution's mean is specified by the linear approximation obtained from the Taylor expansion of f, with a variance σ_c^2.

For volatility prediction in financial markets using federated learning, deviations from the baseline global model $\boldsymbol{w}^*$ influence the mean prediction through the Jacobian adjustment, while the Gaussian term $\mathcal{N}(0, \sigma_c^2)$ accounts for the stochastic nature of the predictions. This framework establishes a robust basis for trading platforms to adapt and retrain the global model locally, ensuring performance optimization tailored to the unique characteristics of individual datasets.

For each trading platform c with its local dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|P_c|}$, the likelihood function quantifies the probability of observing the given data. It incorporates both the individual variances from the Gaussian noise and the deviations of the model predictions from actual data points. This integration is captured by the model's output and its linear approximation around $\boldsymbol{w}^*$ which is formulated as

$$P_c(\boldsymbol{w}) = \frac{1}{(2\pi\sigma_c^2)^{\frac{|P_c|}{2}}} \exp\left(-\frac{1}{2\sigma_c^2} \sum_{i=1}^{|P_c|} (y_i - (f(\boldsymbol{x}_i; \boldsymbol{w}^*) + J_{\boldsymbol{w}^*}(\boldsymbol{x}_i)^T (\boldsymbol{w} - \boldsymbol{w}^*)))^2 \right). \tag{10}$$

This formulation enables trading platforms to effectively assess the fit between their local data and the global model, guiding them in refining the model parameters to better capture the underlying patterns in volatility dynamics.

For rapid local adaptation within our financial market volatility prediction, we transform the likelihood function into its logarithmic form as

$$\log(P_c(\boldsymbol{w})) = -\frac{|P_c|}{2} \log(2\pi\sigma_c^2) - \frac{1}{2\sigma_c^2} \sum_{i=1}^{|P_c|} (y_i - (f(\boldsymbol{x}_i; \boldsymbol{w}^*) + J_{\boldsymbol{w}^*}(\boldsymbol{x}_i)^T (\boldsymbol{w} - \boldsymbol{w}^*)))^2, \tag{11}$$

which simplifies the expression by converting the product of probabilities into a sum of logarithms, linearizing the effects of the parameters and enhancing the tractability of the optimization problem. Notably, $-\frac{1}{2\sigma_c^2}\sum_{i=1}^{|P_c|}(y_i - f(\boldsymbol{x}_i;\boldsymbol{w}))^2$ represents the sum of squared residuals, adjusted by the inverse of the noise variance σ_c^2.

Therefore, the local adaptation process can be formulated as minimizing the following loss function

$$\hat{L}_c(\boldsymbol{w}) = \frac{1}{2\sigma_c^2}\sum_{i=1}^{|P_c|}(y_i - (f(\boldsymbol{x}_i;\boldsymbol{w}^*) + J_{\boldsymbol{w}^*}(\boldsymbol{x}_i)^T(\boldsymbol{w}-\boldsymbol{w}^*)))^2 + \frac{|P_c|}{2}\log(2\pi\sigma_c^2), \tag{12}$$

which comprises a term that evaluates the sum of squared deviations between the predicted volatility and the actual volatility, scaled by the noise variance σ_c^2, and a constant term that standardizes the loss based on the dataset size and noise level in the context of local financial market data.

We define $\boldsymbol{J_{w^*}} = \{\boldsymbol{J_{w^*}(x_i)}\}_{i=1}^{P_c}$ as the collection of Jacobian matrices of the model's predictions with respect to the features generated from order book and trading data, evaluated at $\boldsymbol{w}^*$. The sum of the outer products of these Jacobian matrices across all data points forms a symmetric matrix, which can be expressed as

$$\sum_{i=1}^{|P_c|}\boldsymbol{J_{w^*}(x_i)J_{w^*}(x_i)^T} = \boldsymbol{J_{w^*}J_{w^*}^T}, \tag{13}$$

which reflects the covariance structure of the gradients, capturing the sensitivity of the model's predictions to the features derived from the trading platforms' data. To facilitate a clearer understanding and to simplify computations in practice, this loss function can be reformulated as

$$\begin{aligned}\hat{L}_c(\boldsymbol{w}) = (\boldsymbol{w}-\boldsymbol{w}^*)^T\frac{1}{2\sigma_c^2}\boldsymbol{J_{w^*}J_{w^*}^T}(\boldsymbol{w}-\boldsymbol{w}^*) - (\boldsymbol{w}-\boldsymbol{w}^*)^T\frac{1}{\sigma_c^2}\boldsymbol{J_{w^*}(y_c - f_c)} \\ + \frac{1}{2\sigma_c^2}\boldsymbol{(y_c - f_c)^T(y_c - f_c)} + \frac{|P_c|}{2}\log(2\pi\sigma_c^2),\end{aligned} \tag{14}$$

where $\boldsymbol{f_c} = \{f(\boldsymbol{x}_i;\boldsymbol{w}^*)\}_{i=1}^{P_c}$ and $\boldsymbol{y_c} = \{y_i\}_{i=1}^{P_c}$. It quantifies the balance between the model's internal predictions and the observed deviations from the actual volatility outcomes, scaled by the noise variance, σ_c^2. This local loss function is critical for adapting the global model to better fit the specific characteristics of the local trading platform's data. The local model adaptation is achieved by setting the gradient of the designed local loss function, $\boldsymbol{\nabla\hat{L}_c(w)}$, to zero as

$$\boldsymbol{\nabla\hat{L}_c(w)} = \frac{1}{\sigma_c^2}\boldsymbol{J_{w^*}J_{w^*}^T}(\boldsymbol{w}-\boldsymbol{w}^*) - \frac{1}{\sigma_c^2}\boldsymbol{J_{w^*}(y_c - f_c)} = \boldsymbol{0}. \tag{15}$$

By solving this equation, we identify the stationary point, which is typically a minimum for a well-defined convex function

$$\boldsymbol{w} = \boldsymbol{(J_{w^*}J_{w^*}^T)^{-1}J_{w^*}(y_c - f_c)} + \boldsymbol{w}^*, \tag{16}$$

which suggests that the local model adaptation is proportional to the pseudo-inverse of the aggregated Jacobian product, adjusted by the residuals between the observed volatility and the model's predicted volatility. Importantly, the term $\boldsymbol{(J_{w^*}J_{w^*}^T)^{-1}J_{w^*}}$ only needs to be computed once, providing significant computational efficiency.

When predicting for a new data sample, $\boldsymbol{x}_i$ derived from order book and trading data, the model leverages both the learned parameters and the inherent variability in observations for making predictions by following formulation

$$\hat{y}_i = f(\boldsymbol{x}_i;\boldsymbol{w}^*) + J_{\boldsymbol{w}^*}(\boldsymbol{x}_i)^T\boldsymbol{(J_{w^*}J_{w^*}^T)^{-1}J_{w^*}(y_c - f_c)} + \mathcal{N}(0;\sigma_c^2). \tag{17}$$

This formula represents the linearized update to the model's prediction, adjusted by the newly optimized parameters, and includes a Gaussian noise term, which accounts for the inherent uncertainty in the prediction. It plays a crucial role in ensuring a realistic forecast of local volatility.

By incorporating the baseline prediction using the global model parameters $f(\boldsymbol{x}_i; \boldsymbol{w}^*)$, the adjustment to the prediction based on the local training data $J_{\boldsymbol{w}^*}(\boldsymbol{x}_i)^T(\boldsymbol{J}_{\boldsymbol{w}^*}\boldsymbol{J}_{\boldsymbol{w}^*}^{\boldsymbol{T}})^{-1}\boldsymbol{J}_{\boldsymbol{w}^*}(\boldsymbol{y_c} - \boldsymbol{f_c})$ and the inherent variability in the predictions, we provide an adaptive approach to predicting volatility, tailored to the unique characteristics of each trading platform's data. This approach ensures that the predictions remain both accurate and robust, even in the face of dynamic and heterogeneous market conditions. The convergence analysis is shown in Appendix A.

5 Experiments

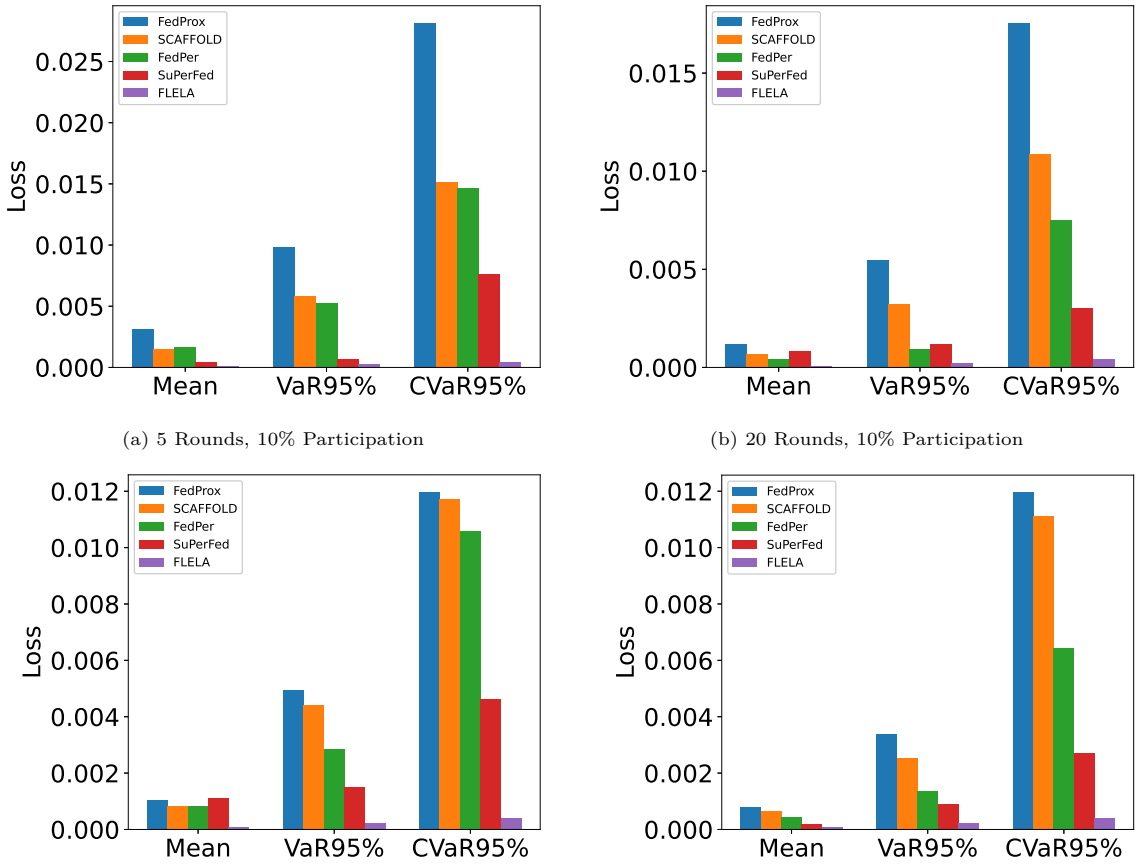
5.1 Local Training and Test Datasets Design

We aim to forecast short-term volatility for 112 stocks spanning multiple sectors Andrew Meyer (2021). The dataset comprises both order book and trade data for these stocks, aggregated into multiple time buckets. The values in the order book represent the latest snapshots of market activity, taken at one-second intervals. Each time bucket comprises order book data spanning the 600 seconds. Our experiments involve predicting the volatility for each time bucket of the stocks. There are $428,932$ samples in the entire dataset, where 107 of the stocks have data for 3830 time buckets, while 3 stocks have data for 3829 time buckets, 1 stock has data for 3820 time buckets, and another stock has data for 3815 time buckets. The entire dataset is divided into $10,000$ trading platforms based on a Dirichlet distribution-based non-IID setting Hsu et al. (2019). The Dirichlet distribution's concentration parameter, α, determines the stock distribution for each trading platform which is set to 0.5 in our experiments. Each trading platform randomly splits its data into a training set and a test set, with 20% allocated for testing. This setup allows us to estimate the performance of each FL algorithm on each trading platform's test set using its personalized model.

We compare the performance of the proposed FLELA to other state-of-the-art FL methods: FedProx Li et al. (2020), SCAFFOLD Karimireddy et al. (2019), and FedPer Arivazhagan et al. (2019). The neural network architecture comprises 363 elements in the input layer and 1 in the output layer. It includes 2 hidden layers, each with 40 neurons using the Tanh activation function. The experimental results provide compelling evidence supporting the contributions of FLELA in addressing the challenges of federated learning. We employ stochastic gradient descent (SGD) optimization with a learning rate of 0.01 for both local and global updating. The batch size for local update in each trading platform is 500. Our experimental platform features an 8-core CPU, a 14-core GPU, and 16GB of RAM.

5.2 Performance Comparison

In Fig. 1, we present a comparative analysis of the proposed FLELA against baseline approaches across varying levels of trading platform participation. Across various participation rates and rounds, FLELA consistently outperforms other methods, including Individual Train, FedProx, SCAFFOLD, and FedPer, in terms of mean loss, Value at Risk (VaR), and Conditional Value at Risk (CVaR). The Individual Train only conducts local training with the same number of parameter updating as the other FL methods, i.e., 50 epochs in Fig. 1(a)(c) and 200 epochs in Fig. 1(b)(d). As shown in Fig. 1(a), FLELA demonstrates rapid convergence to low loss values even with only 10% trading platform participation over 5 rounds, with each trading platform undergoing 10 local epochs, where FLELA achieves significantly lower mean loss (7.726e-05) compared to Individual Train (0.0132), FedProx (0.0031), SCAFFOLD (0.0015), and FedPer (0.0017). Moreover, with a trading platform participation rate of 30% over 20 rounds, FLELA continues to exhibit superior performance with substantially lower mean loss, VaR95%, and CVaR95% values compared to other algorithms, as depicted in Figures 1(c) and 1(d). These results underscore FLELA's ability to adapt to heterogeneous trading platform datasets and varying participation rates, effectively mitigating the risk of training stagnation and ensuring robust and efficient federated model training. By leveraging enhanced linearized local training, FLELA not only improves predictive accuracy but also addresses the challenge of insufficient training by maximizing the utility of available data. Thus, the experimental findings provide strong empirical support for FLELA's advancement in federated learning, particularly in the context of realizing volatility prediction tasks in dynamic financial environments with heterogeneous local datasets among numerous trading platforms.



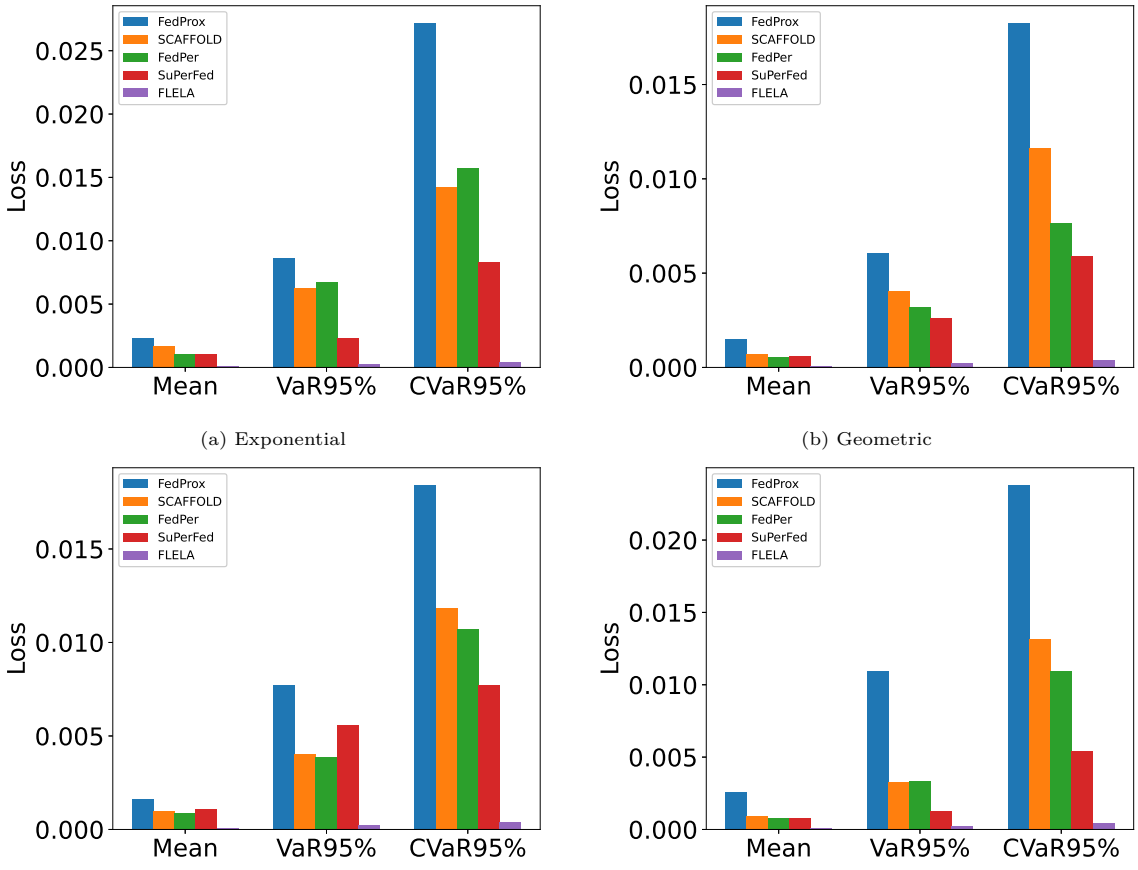
(a) 5 Rounds, 10% Participation

(b) 20 Rounds, 10% Participation

(c) 5 Rounds, 30% Participation

(d) 20 Rounds, 30% Participation

Figure 1



(a) Exponential

(b) Geometric

(c) Gamma

(d) Chisquare

Figure 2

Table 1: The Computation Cost Comparison for One Round

Participation Rate	Fedprox (s)	SCAFFOLD (s)	FedPer (s)	SuPerFed (s)	FLELA (s)
30%	74.39	105.9	46.18	99.63	48.16
60%	144.83	208.51	90.42	186.67	94.43

As shown in Fig. 2, we provide the comparative analysis across various participation distributions to evaluate the efficacy of FLELA in addressing the inherent challenges of FL with dynamic participation. The experiments were conducted with a 20% participation rate, 10 federated rounds, and 10 local epochs in each round. As shown in Fig. 2(a) where trading platforms are sampled from an exponential distribution, with a scale parameter of 1.0, FLELA demonstrates a remarkable ability to achieve a mean loss of 7.358×10^{-5}, VaR95% of 2.284×10^{-4}, and CVaR95% of 3.978×10^{-4}, outperforming FedProx, SCAFFOLD, and FedPer by an order. As shown in Fig. 2(b), where trading platforms are sampled from a geometric distribution with a probability of success of an individual trial set at 0.35, FLELA once again emerges as the top-performing algorithm. Fig. 2(c) explores the performance of algorithms when trading platforms are sampled from a Gamma distribution, with a shape parameter of 2.0 and a scale parameter of 1.0. In Fig. 2(d), where trading platforms are sampled from a chi-square distribution with the number of degrees of freedom set at 2.0, FLELA continues to outshine the baseline algorithms.

By consistently delivering superior performance metrics, FLELA showcases its adaptability in scenarios characterized by varying levels of data availability and participation. By consistently achieving lower mean loss, VaR95%, and CVaR95% values, FLELA underscores its resilience and adaptability in optimizing federated model training across a spectrum of trading platform distributions. These results indicate FLELA's adeptness at navigating volatile market conditions and optimizing federated model training even in the face of erratic trading platform participation patterns.

5.3 Computation Cost Comparison

FLELA demonstrates efficient computation times across different participation rates, indicating its adaptability to varying levels of trading platform involvement. This adaptability mitigates the challenge of irregular participation by ensuring timely model updates and preventing stagnation in the federated learning process. By achieving competitive computation times without compromising privacy or regulatory compliance, FLELA enables collaborative model training across fragmented financial datasets. This privacy-preserving collaboration addresses concerns related to data sharing restrictions and enhances the feasibility of federated learning in decentralized financial environments.

Despite data fragmentation and irregular participation, FLELA maintains stable performance in predicting realized volatility. Its robustness ensures reliable volatility estimates, crucial for risk management and investment decision-making in dynamic financial markets. As shown in Table 1, at a participation rate of 60%, FLELA exhibits relative efficient computation time among the evaluated FL methods, with a processing time of 94.43 seconds. This performance surpasses that of both Fedprox and SCAFFOLD, which require 144.83 seconds and 208.51 seconds, respectively. FedPer demonstrates comparable efficiency to FLELA, with a computation time of 90.42 seconds. Considering Fedprox, SCAFFOLD, and FedPer require more rounds to converge, the overall computation costs of FLELA is much lower. This numerical comparison underscores the superior computational efficiency of FLELA.

6 Conclusions and Discussions

In conclusion, we introduce a novel approach FLELA tailored to address the challenges of FL in dynamic financial environments. By leveraging probabilistic frameworks and local adaptation, FLELA demonstrates remarkable adaptability and robustness in handling heterogeneous trading platform datasets and varying participation rates. Through comprehensive experimental evaluations, FLELA consistently outperforms existing FL algorithms, including FedProx, SCAFFOLD, and FedPer, in terms of predictive accuracy and

efficiency. The experimental results underscore FLELA's potential in realizing volatility prediction tasks, which are crucial for risk management and strategic decision-making in financial markets. In addition, the proposed light-weight local adaptation will be applied for many other FL applications with heterogeneous datasets and dynamic participants. In our future research, we aim to incorporate unobserved market factors and refine our approximation methods to enhance predictive accuracy, particularly during periods of high volatility or low liquidity.

References

CameronOptiver IXAGPOPU Jiashen Liu Matteo Pietrobon (Optiver) OptiverMerle Sohier Dane Stefan Vallentine Andrew Meyer, BerniceOptiver. Optiver realized volatility prediction, 2021. URL `https://kaggle.com/competitions/optiver-realized-volatility-prediction`.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

W Brian Arthur, John H Holland, Blake LeBaron, Richard Palmer, and Paul Tayler. Asset pricing under endogenous expectations in an artificial stock market. In *The economy as an evolving complex system II*, pp. 15–44. CRC Press, 2018.

Syreen Banabilah, Moayad Aloqaily, Eitaa Alsayed, Nida Malik, and Yaser Jararweh. Federated learning review: Fundamentals, enabling technologies, and future applications. *Information processing & management*, 59 (6):103061, 2022.

El Bachir Boukherouaa, Mr Ghiath Shabsigh, Khaled AlAjmi, Jose Deodoro, Aquiles Farias, Ebru S Iskender, Mr Alin T Mirestean, and Rangachary Ravikumar. *Powering the digital economy: opportunities and risks of artificial intelligence in finance*. International Monetary Fund, 2021.

Estelle Cantillon and Pai-Ling Yin. Competition between exchanges: A research agenda. *International journal of industrial organization*, 29(3):329–336, 2011.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Joel Hasbrouck. *Empirical market microstructure: The institutions, economics, and econometrics of securities trading*. Oxford University Press, 2007.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*, 31:8571–8580, 2018.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2(6), 2019.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Wesley Maddox, Shuai Tang, Pablo Moreno, Andrew Gordon Wilson, and Andreas Damianou. Fast adaptation with linearized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 2737–2745. PMLR, 2021.

Ananth Madhavan. Market microstructure: A survey. *Journal of Financial Markets*, 3(3):205–258, 2000.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Jorge Otero. High-frequency data, frequency domain inference, and volatility forecasting. *Review of Economics and Statistics*, 84(4):669–681, 2002.

Mariia Seleznova and Gitta Kutyniok. Neural tangent kernel beyond the infinite-width limit: Effects of depth and initialization. In *International Conference on Machine Learning*, pp. 19522–19560. PMLR, 2022.

Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6176–6185, 2017.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. In *ACM Transactions on Intelligent Systems and Technology (TIST)*, volume 10, pp. 1–19. ACM, 2019.

A Convergence Analysis

In each training round t, we dynamically select a subset of trading platforms $S^t \subseteq E$, where $|S^t| = S$ denotes the number of participating platforms in that round. The current global model $\boldsymbol{w^{t-1}}$ is distributed to all selected platforms. Each participating platform i initializes its local model with the received global model, i.e., $\boldsymbol{w^t_{i,0}} = \boldsymbol{w^{t-1}}$. The local models are then updated through K iterations of stochastic gradient descent (SGD) based on their local data. The update rule for the local parameters at iteration k is given by

$$\boldsymbol{w^t_{i,k}} = \boldsymbol{w^t_{i,k-1}} - \alpha_l \boldsymbol{\nabla L_i}(\boldsymbol{w^t_{i,k-1}}), \tag{18}$$

where α_l is the local learning rate, and $\boldsymbol{\nabla L_i(w)}$ represents the stochastic gradient of the local loss function L_i at platform i. After K iterations, the final local model for platform i is

$$\boldsymbol{w^t_{i,K}} = \boldsymbol{w^{t-1}} - \sum_{k=0}^{K-1} \alpha_l \boldsymbol{\nabla L_i}(\boldsymbol{w^t_{i,k}}). \tag{19}$$

We assume that $\boldsymbol{\nabla L_i(w)}$ is an unbiased stochastic gradient with variance bounded by σ^2. The global model is updated by aggregating the updates from all selected local models. The update rule for the global model with global step size α_g is

$$\boldsymbol{w^t} = \boldsymbol{w^{t-1}} + \frac{\alpha_g}{S} \sum_{i \in S^t} (\boldsymbol{w^t_{i,K}} - \boldsymbol{w^{t-1}}) = \boldsymbol{w^{t-1}} - \frac{\alpha_g}{S} \sum_{i \in S^t} \sum_{k=0}^{K-1} \alpha_l \boldsymbol{\nabla L_i}(\boldsymbol{w^t_{i,k}}). \tag{20}$$

To facilitate the analysis, we define the effective step size as $\tilde{\alpha} = K\alpha_l\alpha_g$. The update applied to the server model in round t can be expressed as

$$\boldsymbol{\delta^{t-1}} = -\frac{\tilde{\alpha}}{KS} \sum_{i \in S^t} \sum_{k=0}^{K-1} \boldsymbol{\nabla L_i}(\boldsymbol{w^t_{i,k}}). \tag{21}$$

The expectation of the server update, considering the participation of all platforms E, is

$$\mathbb{E}[\boldsymbol{\delta^{t-1}}] = -\frac{\tilde{\alpha}}{K|E|} \sum_{i \in E} \sum_{k=0}^{K-1} \boldsymbol{\nabla L_i}(\boldsymbol{w^t_{i,k}}). \tag{22}$$

The reduction can be shown by examining the distance from the minimizer $\boldsymbol{w}^*$

$$\begin{aligned}\|\boldsymbol{w}^{\boldsymbol{t}} - \boldsymbol{w}^*\|^2 &= \|\boldsymbol{w}^{\boldsymbol{t-1}} + \boldsymbol{\delta}^{\boldsymbol{t-1}} - \boldsymbol{w}^*\|^2 \\ &= \|\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*\|^2 + 2\left(\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*\right)^T \boldsymbol{\delta}^{\boldsymbol{t-1}} + \|\boldsymbol{\delta}^{\boldsymbol{t-1}}\|^2.\end{aligned} \tag{23}$$

We use $\mathbb{E}_{t-1}[\cdot]$ to denote the expectation conditioned on all the randomness generated prior to round t. Thus, we have

$$\mathbb{E}_{t-1}\left[\left(\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*\right)^T \boldsymbol{\delta}^{\boldsymbol{t-1}}\right] = -\frac{\tilde{\alpha}}{K|E|}\sum_{i\in E}\sum_{k=0}^{K-1}\mathbb{E}\left[\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k}}^{\boldsymbol{t}})^T\left(\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*\right)\right]. \tag{24}$$

We assume the eigenvalues of the Hessian of all $\{L_i(\boldsymbol{w})\}_{i\in E}$ are bounded within (μ, β), and the quadratic upper bound and quadratic lower bound for local objective function $L_i(\boldsymbol{w}^{\boldsymbol{t-1}})$ can be obtained as

$$L_i(\boldsymbol{w}^{\boldsymbol{t-1}}) \le L_i(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}) + \boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})^{\boldsymbol{T}}(\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}) + \frac{\beta}{2}\|\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}\|^{\boldsymbol{2}}, \tag{25}$$

and

$$L_i(\boldsymbol{w}^*) \ge L_i(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}) + \boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})^{\boldsymbol{T}}(\boldsymbol{w}^* - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}) + \frac{\mu}{2}\|\boldsymbol{w}^* - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}\|^{\boldsymbol{2}}. \tag{26}$$

Then, we can get

$$\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})^{\boldsymbol{T}}(\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*) \ge L_i(\boldsymbol{w}^{\boldsymbol{t-1}}) - L_i(\boldsymbol{w}^*) + \frac{\mu}{2}\|\boldsymbol{w}^* - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}\|^{\boldsymbol{2}} - \frac{\beta}{2}\|\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}\|^{\boldsymbol{2}}. \tag{27}$$

By Triangle inequality, we have

$$\|\boldsymbol{w}^* - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}\|^{\boldsymbol{2}} \ge \frac{1}{2}\|\boldsymbol{w}^* - \boldsymbol{w}^{\boldsymbol{t-1}}\|^{\boldsymbol{2}} - \|\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}\|^{\boldsymbol{2}}. \tag{28}$$

Combining with $\beta \ge \mu$, we can obtain

$$\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})^{\boldsymbol{T}}(\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*) \ge L_i(\boldsymbol{w}^{\boldsymbol{t-1}}) - L_i(\boldsymbol{w}^*) + \frac{\mu}{4}\|\boldsymbol{w}^* - \boldsymbol{w}^{\boldsymbol{t-1}}\|^{\boldsymbol{2}} - \beta\|\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}\|^{\boldsymbol{2}}. \tag{29}$$

Therefore, we have

$$\begin{aligned}&\mathbb{E}_{t-1}\left[(\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*)^T\boldsymbol{\delta}^{\boldsymbol{t-1}}\right] \\ &\le -\frac{\tilde{\alpha}}{K|E|}\sum_{i\in E}\sum_{k=0}^{K-1}\left(L_i(\boldsymbol{w}^{\boldsymbol{t-1}}) - L_i(\boldsymbol{w}^*) + \frac{\mu}{4}\|\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*\|^2 - \beta\|\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}} - \boldsymbol{w}^{\boldsymbol{t-1}}\|^2\right).\end{aligned} \tag{30}$$

The drift of the local model from the global model is formulated as

$$\varepsilon = \frac{1}{K|E|}\sum_{i\in E}\sum_{k=0}^{K-1}\|\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}} - \boldsymbol{w}^{\boldsymbol{t-1}}\|^2, \tag{31}$$

then we obtain

$$\mathbb{E}_{t-1}\left[(\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*)^T\boldsymbol{\delta}^{\boldsymbol{t-1}}\right] \le -\tilde{\alpha}\left(L(\boldsymbol{w}^{\boldsymbol{t-1}}) - L(\boldsymbol{w}^*) + \frac{\mu}{4}\|\boldsymbol{w}^{\boldsymbol{t-1}} - \boldsymbol{w}^*\|^2\right) + \tilde{\alpha}\beta\varepsilon. \tag{32}$$

For the sequence of local gradients $\{\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})\}$ during the training procedure, the variance is defined by

$$\begin{aligned}&\mathbb{E}[\|\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}}) - \mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})]\|^{\boldsymbol{2}}] \\ &= \mathbb{E}[\|\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})\|^{\boldsymbol{2}}] - 2\|\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})]\|^{\boldsymbol{2}} + \|\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})]\|^{\boldsymbol{2}} \\ &= \mathbb{E}[\|\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})\|^{\boldsymbol{2}}] - \|\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w}_{\boldsymbol{i,k-1}}^{\boldsymbol{t}})]\|^{\boldsymbol{2}}.\end{aligned} \tag{33}$$

Similarly, we can get that

$$\begin{aligned}&\mathbb{E}[\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}(\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})])\|^2]\\&=\mathbb{E}[\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})\|^2]-\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})]\|^2.\end{aligned}\tag{34}$$

We assume the variance of local gradients is upper bounded by

$$\mathbb{E}[\|\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})]\|^{\boldsymbol{2}}]\leq\gamma^2,\tag{35}$$

and by Jensen's inequality, we have that

$$\begin{aligned}&\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}(\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})])\|^2\\&\leq\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\|\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})]\|^2.\end{aligned}\tag{36}$$

Using the linearity of the expectation we have

$$\begin{aligned}&\mathbb{E}[\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}(\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})])\|^2]\\&\leq\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\mathbb{E}[\|\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\mathbb{E}[\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})]\|^2].\end{aligned}\tag{37}$$

Then, we have the upper bound of $\mathbb{E}_{t-1}\left[\|\boldsymbol{\delta^{t-1}}\|^2\right]$ as

$$\mathbb{E}_{t-1}\left[\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})\|^2\right]\leq\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})\|^2+\frac{\tilde{\alpha}^2\gamma^2}{KS}.\tag{38}$$

By the triangle inequality, we have

$$\begin{aligned}&\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\left(\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\boldsymbol{\nabla L_i}(\boldsymbol{w^{t-1}})+\boldsymbol{\nabla L_i}(\boldsymbol{w^{t-1}})\right)\|^2\\&\leq 2\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\left(\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\boldsymbol{\nabla L_i}(\boldsymbol{w^{t-1}})\right)\|^2+2\|\frac{\tilde{\alpha}}{S}\sum_{i=1}^{S}\boldsymbol{\nabla L_i}(\boldsymbol{w^{t-1}})\|^2.\end{aligned}\tag{39}$$

By Jensen's inequality and the β-smoothness property, we have

$$\begin{aligned}&\|\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\left(\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\boldsymbol{\nabla L_i}(\boldsymbol{w^{t-1}})\right)\|^2\\&\leq\frac{\tilde{\alpha}}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\|\boldsymbol{\nabla L_i}(\boldsymbol{w_{i,k-1}^t})-\boldsymbol{\nabla L_i}(\boldsymbol{w^{t-1}})\|^2\\&\leq\frac{\tilde{\alpha}\beta^2}{KS}\sum_{i=1}^{S}\sum_{k=0}^{K-1}\|\boldsymbol{w_{i,k-1}^t}-\boldsymbol{w^{t-1}}\|^2.\end{aligned}\tag{40}$$

We can also obtain

$$\begin{aligned}\|\frac{\tilde{\alpha}}{S}\sum_{i=1}^{S}\boldsymbol{\nabla L_i(w^{t-1})}\|^2 &= \|\frac{\tilde{\alpha}}{S}\sum_{i=1}^{S}\left(\boldsymbol{\nabla L_i(w^{t-1})}-\boldsymbol{\nabla L(w^{t-1})}\right)+\boldsymbol{\nabla L(w^{t-1})}\|^2\\ &\leq 2\|\frac{\tilde{\alpha}}{S}\sum_{i=1}^{S}\left(\boldsymbol{\nabla L_i(w^{t-1})}-\boldsymbol{\nabla L(w^{t-1})}\right)\|^2+2\|\tilde{\alpha}\boldsymbol{\nabla L(w^{t-1})}\|^2\\ &\leq 2\tilde{\alpha}^2B+4\beta\tilde{\alpha}^2\left(L(\boldsymbol{w^{t-1}})-L(\boldsymbol{w^*})\right),\end{aligned} \tag{41}$$

by the triangle inequality, where we define the gradient dissimilarity is upper bounded by

$$\|\frac{1}{S}\sum_{i=1}^{S}\left(\boldsymbol{\nabla L_i(w^{t-1})}-\boldsymbol{\nabla L(w^{t-1})}\right)\|^2\leq B. \tag{42}$$

We can conclude that

$$\mathbb{E}_{t-1}[\|\boldsymbol{\delta^{t-1}}\|^2]\leq 2\tilde{\alpha}\beta^2\varepsilon+4\tilde{\alpha}^2B+8\beta\tilde{\alpha}^2\left(L(\boldsymbol{w^{t-1}})-L(\boldsymbol{w^*})\right)+\frac{\tilde{\alpha}^2\gamma^2}{KS}, \tag{43}$$

and the improvement in one round is

$$\begin{aligned}\mathbb{E}_{t-1}[\|\boldsymbol{w^t-w^*}\|^2] &= \|\boldsymbol{w^{t-1}-w^*}\|^2+2\mathbb{E}_{t-1}[(\boldsymbol{w^{t-1}-w^*})^T\boldsymbol{\delta^{t-1}}]+\mathbb{E}_{t-1}[\|\boldsymbol{\delta^{t-1}}\|^2]\\ &\leq \|\boldsymbol{w^{t-1}-w^*}\|^2-2\tilde{\alpha}\left(L(\boldsymbol{w^{t-1}})-L(\boldsymbol{w^*})+\frac{\mu}{2}\|\boldsymbol{w^{t-1}-w^*}\|^2\right)\\ &\quad+2\tilde{\alpha}\beta\varepsilon+2\tilde{\alpha}\beta^2\varepsilon+4\tilde{\alpha}^2B+8\beta\tilde{\alpha}^2\left(L(\boldsymbol{w^{t-1}})-L(\boldsymbol{w^*})\right)+\frac{\tilde{\alpha}^2\gamma^2}{KS}\\ &= (1-\tilde{\alpha}\mu)\|\boldsymbol{w^{t-1}-w^*}\|^2+(8\beta\tilde{\alpha}^2-2\tilde{\alpha})\left(L(\boldsymbol{w^{t-1}})-L(\boldsymbol{w^*})\right)\\ &\quad+2\tilde{\alpha}\beta(\beta+1)\varepsilon+4\tilde{\alpha}^2B+\frac{\tilde{\alpha}^2\gamma^2}{KS}.\end{aligned} \tag{44}$$

Since the local updating is stochastic, and we have defined the variance of the sampled gradient from the full local gradient as σ^2

$$\mathbb{E}\|\boldsymbol{g_i(w)}-\nabla L_i(\boldsymbol{w})\|^2=\sigma^2=\mathbb{E}\|\boldsymbol{g_i(w)}\|^2-\|\nabla L_i(\boldsymbol{w})\|^2. \tag{45}$$

If we define $a=\frac{1}{K-1}$, then we can obtain the upper bound of the expectation

$$\begin{aligned}\mathbb{E}\|\boldsymbol{w^t_{i,k}-w^{t-1}}\|^2 &\leq \left(1+\frac{1}{K-1}\right)\mathbb{E}\|\boldsymbol{w^t_{i,k-1}-w^{t-1}}\|^2+K\alpha_l^2\mathbb{E}\|\boldsymbol{g_i(w^t_{i,k-1})}\|^2\\ &= \left(1+\frac{1}{K-1}\right)\mathbb{E}\|\boldsymbol{w^t_{i,k-1}-w^{t-1}}\|^2+K\alpha_l^2\|\nabla L_i(\boldsymbol{w^t_{i,k-1}})\|^2+K\alpha_l^2\sigma^2.\end{aligned} \tag{46}$$

Then, we want to eliminate the gradients with the local updating model $\boldsymbol{\nabla L_i(w^t_{i,k-1})}$ by applying the inequality

$$\begin{aligned}\|\nabla L_i(\boldsymbol{w^t_{i,k-1}})\|^2 &= \|\nabla L_i(\boldsymbol{w^t_{i,k-1}})-\nabla L_i(\boldsymbol{w^{t-1}})+\nabla L_i(\boldsymbol{w^{t-1}})\|^2\\ &\leq 2\|\nabla L_i(\boldsymbol{w^t_{i,k-1}})-\nabla L_i(\boldsymbol{w^{t-1}})\|^2+2\|\nabla L_i(\boldsymbol{w^{t-1}})\|^2.\end{aligned} \tag{47}$$

Based on the Lipschitz continuous gradient, we have

$$\|\nabla L_i(\boldsymbol{w^t_{i,k-1}})-\nabla L_i(\boldsymbol{w^{t-1}})\|^2\leq\beta^2\|\boldsymbol{w^t_{i,k-1}-w^{t-1}}\|^2, \tag{48}$$

and we can obtain

$$\begin{aligned}\mathbb{E}\|\boldsymbol{w^t_{i,k}-w^{t-1}}\|^2 &\leq \left(1+\frac{1}{K-1}+2K\alpha_l^2\beta^2\right)\mathbb{E}\|\boldsymbol{w^t_{i,k-1}-w^{t-1}}\|^2\\ &\quad+2K\alpha_l^2\|\nabla L_i(\boldsymbol{w^{t-1}})\|^2+K\alpha_l^2\sigma^2.\end{aligned} \tag{49}$$

To upper bound the drift over K local updates, we can unroll the recursion from $\boldsymbol{w_{i,0}^t}$ to $\boldsymbol{w_{i,K-1}^t}$. Since $\boldsymbol{w_{i,0}^t = w^{t-1}}$, we can obtain

$$\mathbb{E}\|\boldsymbol{w_{i,K}^t - w^{t-1}}\|^2 \leq \sum_{k=0}^{K-1}\left(1+\frac{1}{K-1}+2K\alpha_l^2\beta^2\right)^k (2K\alpha_l^2\|\nabla L_i(\boldsymbol{w^{t-1}})\|^2 + K\alpha_l^2\sigma^2). \tag{50}$$

This upper bound is a geometric series where $2K\alpha_l^2\|\nabla L_i(\boldsymbol{w^{t-1}})\|^2 + K\alpha_l^2\sigma^2$ is the coefficient, and $1+\frac{1}{K-1}+2K\alpha_l^2\beta^2$ is the common ratio between adjacent terms. This upper bound can also be written as

$$\sum_{k=0}^{K-1}(1+\frac{1}{K-1}+2K\alpha_l^2\beta^2)^k(2K\alpha_l^2\|\nabla L_i(\boldsymbol{w^{t-1}})\|^2 + K\alpha_l^2\sigma^2) = q(2K\alpha_l^2\|\nabla L_i(\boldsymbol{w^{t-1}})\|^2 + K\alpha_l^2\sigma^2). \tag{51}$$

where q is a constant with a fixed local learning rate α_l and local updating iterations K defined as

$$q = \frac{1-\left(1+\frac{1}{K-1}+2K\alpha_l^2\beta^2\right)^K}{1-\left(1+\frac{1}{K-1}+2K\alpha_l^2\beta^2\right)}. \tag{52}$$

Then, we come to analysis of the dynamic trading platform participation. According to the quadratic upper bound and the linear lower bound of the local objective function, we can obtain the inequality as

$$\begin{aligned} L_i(\boldsymbol{w^*}) - L_i(\boldsymbol{w}) &= L_i(\boldsymbol{w^*}) - L_i(\boldsymbol{z}) + L_i(\boldsymbol{z}) - L_i(\boldsymbol{w}) \\ &\leq \boldsymbol{\nabla L_i(w^*)^T(w^*-z)} + \boldsymbol{\nabla L_i(w)^T(z-w)} + \frac{\beta}{2}\boldsymbol{\|z-w\|^2} \\ &= \boldsymbol{\nabla L_i(w^*)^T(w^*-w)} + \boldsymbol{(\nabla L_i(w^*)-\nabla L_i(w))^T(w-z)} + \frac{\beta}{2}\boldsymbol{\|z-w\|^2}. \end{aligned} \tag{53}$$

We define

$$\boldsymbol{z = w - \frac{1}{\beta}(\nabla L_i(w) - \nabla L_i(w^*))}, \tag{54}$$

and then, we have

$$\begin{aligned} &\boldsymbol{(\nabla L_i(w^*)-\nabla L_i(w))^T(w-z)} = -\frac{1}{\beta}\boldsymbol{\|\nabla L_i(w^*)-\nabla L_i(w)\|^2}, \\ &\frac{\beta}{2}\boldsymbol{\|z-w\|^2} = \frac{1}{2\beta}\boldsymbol{\|\nabla L_i(w^*)-\nabla L_i(w)\|^2}, \end{aligned} \tag{55}$$

hence,

$$L_i(\boldsymbol{w^*}) - L_i(\boldsymbol{w}) \leq \boldsymbol{\nabla L_i(w^*)^T(w^*-w)} - \frac{1}{2\beta}\boldsymbol{\|\nabla L_i(w^*)-\nabla L_i(w)\|^2}, \tag{56}$$

which leads to

$$L_i(\boldsymbol{w}) - L_i(\boldsymbol{w^*}) - \boldsymbol{\nabla L_i(w^*)^T(w-w^*)} \geq \frac{1}{2\beta}\boldsymbol{\|\nabla L_i(w^*)-\nabla L_i(w)\|^2}. \tag{57}$$

Since

$$\frac{1}{|E|}\sum_{i\in E}(L_i(\boldsymbol{w}) - L_i(\boldsymbol{w^*})) = L(\boldsymbol{w}) - L^*, \tag{58}$$

then, we have

$$2\beta(L(\boldsymbol{w}) - L^*) \geq \frac{1}{|E|}\sum_{i\in E}\boldsymbol{\|\nabla L_i(w)-\nabla L_i(w^*)\|^2}. \tag{59}$$

The bound on the local gradient can be found as

$$\begin{aligned}\frac{1}{|E|}\sum_{i\in E}\|\boldsymbol{\nabla L_i(w)}\|^{\mathbf{2}} &= \frac{1}{|E|}\sum_{i\in E}\|\boldsymbol{\nabla L_i(w)-\nabla L_i(w^*)+\nabla L_i(w^*)}\|^{\mathbf{2}}\\ &\le \frac{2}{|E|}\sum_{i\in E}\|\boldsymbol{\nabla L_i(w)-\nabla L_i(w^*)}\|^{\mathbf{2}}+\frac{2}{|E|}\sum_{i\in E}\|\boldsymbol{\nabla L_i(w^*)}\|^{\mathbf{2}}\\ &\le 4\beta(L(\boldsymbol{w})-L^*)+\frac{2}{|E|}\sum_{i\in E}\|\boldsymbol{\nabla L_i(w^*)}\|^{\mathbf{2}}.\end{aligned} \tag{60}$$

And the upper bound of the local training drift is

$$\begin{aligned}\varepsilon &\le \frac{1}{|E|}\sum_{i\in E} q\left(2K\alpha_l^2\|\boldsymbol{\nabla L_i(w^{t-1})}\|^{\mathbf{2}}+K\alpha_l^2\sigma^2\right)\\ &\le 8qK\alpha_l^2\beta(L(\boldsymbol{w})-L^*)+\frac{4qK\alpha_l^2}{|E|}\sum_{i\in E}\|\boldsymbol{\nabla L_i(w^*)}\|^{\mathbf{2}}+qK\alpha_l^2\sigma^2.\end{aligned} \tag{61}$$

The improvement in one round can be rewritten as

$$\begin{aligned}\mathbb{E}_{t-1}[\|\boldsymbol{w^t-w^*}\|^{\mathbf{2}}] &\le (1-\tilde{\alpha}\mu)\|\boldsymbol{w^{t-1}-w^*}\|^{\mathbf{2}}+(8\beta\tilde{\alpha}^2-2\tilde{\alpha})(L(\boldsymbol{w^{t-1}})-L(\boldsymbol{w^*}))\\ &\quad+2\tilde{\alpha}\beta(\beta+1)\varepsilon+4\tilde{\alpha}^2B+\frac{\tilde{\alpha}^2\gamma^2}{K|E|}\\ &\le (1-\tilde{\alpha}\mu)\|\boldsymbol{w^{t-1}-w^*}\|^{\mathbf{2}}+c_3(L(\boldsymbol{w^{t-1}})-L(\boldsymbol{w^*}))\\ &\quad+2\tilde{\alpha}\beta(\beta+1)c_1+c_2,\end{aligned} \tag{62}$$

where we define

$$\begin{aligned}c_1 &= \frac{4qK\alpha_l^2}{|E|}\sum_{i\in E}\|\boldsymbol{\nabla L_i(w^*)}\|^{\mathbf{2}}+qK\alpha_l^2\sigma^2,\\ c_2 &= 4\tilde{\alpha}^2B+\frac{\tilde{\alpha}^2\gamma^2}{K|E|},\\ c_3 &= 16\beta^2(\beta+1)qK\tilde{\alpha}\alpha_l^2+8\beta\tilde{\alpha}^2-2\tilde{\alpha}.\end{aligned} \tag{63}$$

Then, we can obtain the following upper bound

$$\begin{aligned}\mathbb{E}_{t-1}[L(\boldsymbol{w^{t-1}})-L(\boldsymbol{w^*})] \le \mathbb{E}_{t-1}&\left[\frac{1}{c_3}(1-\tilde{\alpha}\mu)\|\boldsymbol{w^{t-1}-w^*}\|^{\mathbf{2}}-\frac{1}{c_3}\|\boldsymbol{w^t-w^*}\|^{\mathbf{2}}\right]\\ &+\frac{2}{c_3}\tilde{\alpha}\beta(\beta+1)c_1+\frac{c_2}{c_3}.\end{aligned} \tag{64}$$

We assume the eigenvalues of the Hessian of $\hat{L}_i(\boldsymbol{w})$ are bounded within (μ_i,β_i), i.e.,

$$\mu_i \le \|\frac{1}{\sigma_i^2}\boldsymbol{J_{w^*}J_{w^*}^T}\| \le \beta_i. \tag{65}$$

We assume the local gradient w.r.t $\boldsymbol{w^*}$ is bounded by ϵ_i, i.e., $\|\boldsymbol{\nabla\hat{L}_i(w^*)}\|\le\epsilon_i$, and $\boldsymbol{w_i^*}$ is the optimal model for trading platform i. The improvement of local adaptation in the model space can be bounded by

$$\begin{aligned}\|\boldsymbol{w-w_i^*}\| &= \|\boldsymbol{w^*-(\nabla^2\hat{L}_i(w^*))^{-1}\nabla\hat{L}_i(w^*)-w_i^*}\|\\ &= \|\boldsymbol{(\nabla^2\hat{L}_i(w^*))^{-1}[\nabla\hat{L}_i(w^*)+\nabla^2\hat{L}_i(w^*)(w_i^*-w^*)]}\|.\end{aligned} \tag{66}$$

Since

$$\boldsymbol{\nabla\hat{L}_i(w_i^*)=\nabla\hat{L}_i(w^*)+\nabla^2\hat{L}_i(w^*)(w_i^*-w^*)}, \tag{67}$$

we obtain that

$$\|\boldsymbol{w}-\boldsymbol{w}_{\boldsymbol{i}}^{*}\|=\|(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}[\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}_{\boldsymbol{i}}^{*})-\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})]+(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})\|. \tag{68}$$

We assume the local gradient w.r.t $\boldsymbol{w}^{*}$ is bounded by ϵ_i. Then, we have

$$\|(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})\|\leq\|(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\|\|\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})\|\leq\frac{\epsilon_i}{\mu_i}. \tag{69}$$

Furthermore, we have

$$\begin{aligned}\|(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}[\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}_{\boldsymbol{i}}^{*})-\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})]\|&\leq\|(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\|\|\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}_{\boldsymbol{i}}^{*})-\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})\|\\&\leq\frac{\beta_i}{\mu_i}\|\boldsymbol{w}_{\boldsymbol{i}}^{*}-\boldsymbol{w}^{*}\|.\end{aligned} \tag{70}$$

Therefore, we have

$$\|\boldsymbol{w}-\boldsymbol{w}_{\boldsymbol{i}}^{*}\|\leq\frac{\beta_i}{\mu_i}\|\boldsymbol{w}_{\boldsymbol{i}}^{*}-\boldsymbol{w}^{*}\|+\frac{\epsilon_i}{\mu_i}. \tag{71}$$

By approximating the global model with a local linearization w.r.t each trading platform's local dataset, the model updates are tailored to the local data distribution. This leads to more accurate predictions for each trading platform's data, reducing overall prediction error. To obtain the decrement of objective function $\hat{L}_i(\boldsymbol{w})$, we first derive the second-order Taylor expansion of $\hat{L}_i(\boldsymbol{w}_{\boldsymbol{i}}^{*})$ as

$$\begin{aligned}\hat{L}_i(\boldsymbol{w})&=\hat{L}_i(\boldsymbol{w}^{*})-\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})^{\boldsymbol{T}}(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})\\&\quad+\frac{1}{2}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})^{\boldsymbol{T}}(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})\\&=\hat{L}_i(\boldsymbol{w}^{*})-\frac{1}{2}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})^{\boldsymbol{T}}(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}).\end{aligned} \tag{72}$$

Local adaptation can significantly reduce the objective function $\hat{L}_i(\boldsymbol{w})$, thereby decreasing the need for additional rounds of FL. The second-order Taylor expansion shows that the loss reduction is proportional to the squared norm of the residual $\boldsymbol{y_i}-\boldsymbol{f_i}$, bounded by curvature information from the Hessian matrix as

$$\begin{aligned}\hat{L}_i(\boldsymbol{w}^{*})-\hat{L}_i(\boldsymbol{w})&=\frac{1}{2}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})^{\boldsymbol{T}}(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})\\&\leq\|(\boldsymbol{\nabla^2}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*}))^{-1}\|\cdot\|\boldsymbol{\nabla}\hat{\boldsymbol{L}}_{\boldsymbol{i}}(\boldsymbol{w}^{*})\|^2\\&=\sigma_i^2\|(\boldsymbol{J}_{\boldsymbol{w}^{*}}\boldsymbol{J}_{\boldsymbol{w}^{*}}^{\boldsymbol{T}})^{-1}\|\cdot\|\frac{1}{\sigma_i^2}\boldsymbol{J}_{\boldsymbol{w}^{*}}(\boldsymbol{y_i}-\boldsymbol{f_i})\|^2\\&\leq\frac{\beta_i}{\sigma_i^2\mu_i}\cdot\|(\boldsymbol{y_i}-\boldsymbol{f_i})\|^2.\end{aligned} \tag{73}$$

By effectively reducing the local loss, each trading platform contributes more accurate volatility prediction. Consequently, fewer communication rounds are needed, making the FL process more efficient and scalable. This reduces the need for extensive FL rounds, ultimately leading to better performance in realized volatility prediction.