arXiv:2309.02072v4 [econ.EM] 29 Apr 2024

Data Scaling Effect of Deep Learning in Financial Time Series Forecasting

Chen Liu* Minh-Ngoc Tran* Chao Wang* Richard Gerlach *

Robert Kohn†

Abstract

For many years, researchers have been exploring the use of deep learning in the forecasting of financial time series. However, they have continued to rely on the conventional econometric approach for model optimization, optimizing the deep learning models on individual assets. In this paper, we use the stock volatility forecast as an example to illustrate global training - optimizes the deep learning model across a wide range of stocks - is both necessary and beneficial for any academic or industry practitioners who is interested in employing deep learning to forecast financial time series. Furthermore, a pre-trained foundation model for volatility forecast is introduced, capable of making accurate zero-shot forecasts for any stocks.

Keywords— volatility modeling, deep learning, transfer learning

**Discipline of Business Analytics, The University of Sydney Business School*

†*School of Economics, UNSW Business School*

1 Introduction

Researchers have long been utilizing deep learning models in various financial time series forecasting tasks such as return, volatility forecasting, and portfolio optimization, as outlined in a comprehensive review [10]. Despite extensive research, there remains no consensus in the econometric community on whether deep learning models outperform their straightforward econometrics counterparts. This discrepancy may stem from the fact that most studies evaluate deep learning models on very small datasets. While deep learning models may perform well on a specific dataset, the generality of those conclusions is questionable.

Moreover, almost all studies train individual deep learning models for each individual time series, which mimics the practice in training econometrics models. However, deep learning models are computationally intensive and require a more careful hyperparameter tuning process. Thus, in practice, training and tuning tens of thousands of deep learning models for each individual asset is not feasible. Most importantly, deep learning models, with their complex parameterization, require substantial data to prevent overfitting and capture generalized patterns. They will perform poorly when trained with only thousands of time steps. Furthermore, research in natural language processing (NLP) and computer vision (CV) suggests that with adequate data, deep learning models can make accurate forecasts for previously unseen data, known as zero-shot forecasting, which will avoid the intensive training and managing cost of numerous deep learning models for each individual financial asset.

This study addresses the issues in the training approach for deep learning models in financial time series tasks, using stock volatility as a case study. Extensive empirical experiments were conducted on over 110,000 simulated data series and 12,000 stock series worldwide.

The study first compared local deep learning models with simple local econometric models on a huge amount of time series, revealing that local deep learning models did not outperform their econometric counterparts. Secondly, the study demonstrated that global training negatively impacted econometric models' performance but on the contrary was both beneficial and essential for deep learning models. The study also explored how deep learning model performance varied with the dataset size used for training, known as the data scaling law in computer science terminology. Furthermore, the study introduced an innovative pre-trained foundational model for volatility forecasting, showcasing its ability to generate accurate zero-shot forecasts for any stocks. Lastly, the application of transfer learning in foreign exchange rate forecasting was examined.

Remark 1. *The definition of global training and transfer learning both bear similarities and differences, making it challenging to decide which term best suits our study. In this paper, we primarily adopt the term global training; however, a detailed discussion on the connection to transfer learning is presented in Section. 4.3.*

1.1 Related Works

In the past decade, deep learning models have emerged as a popular tools for financial time series forecasting. Many studies have claim that DL models outperform their econometric counterparts. Despite this, most models are typically trained on a per-series basis. An exception is found in [5], where the authors trained their deep learning models globally. However, they actually overlooked the advantages of global training. By comparing globally trained OLS and deep learning models for predicting stock returns, they erroneously concluded that deep learning models outperform OLS models on their own rather than attribute to global training. As we will demonstrate later, simple econometric models are not suitable for global training and their performance deteriorates as they are trained with to more series.

In the broader context of general time series forecasting, there have been efforts towards global training. For instance, [7] introduced a cross-sectional regression model that considers related sets of time series observed simultaneously to handle missing values in individual time series. Similarly, [13] utilized a pooled regression model by combining related time series to produce reliable forecasts, particularly when historical sales data is lacking. [1] expanded on this concept by incorporating recurrent neural networks (RNNs) and employing clustering techniques to group similar series for joint modeling.

In the computer science community, two very recent works, TimeGPT-1 [4] and TimeFM [2], aim to develop a foundational model for general time series data using zero-shot learning. To the best of our knowledge, these are the only two parallel works related to the concept of global training.

In comparison to those works, our studies diverge in two main aspects: Firstly, those studies focus on general time series characterized by clear trends, seasonality, and cyclic patterns, while financial time series lack of such patterns, making them a different forecasting task from general time series. Secondly, those studies concentrate on introducing a specific model, while our research is centered on examining the global training approach itself and the impact of data scaling on deep learning for financial time series forecasting.

2 Model Description

2.1 Training schemes

Local training: Let $\boldsymbol{y} = \{y_t, t = 1, \ldots, T\}$ be the daily return series of a time series. In volatility modeling, the key quantity of interest is the conditional return variance on the past information, $\sigma_t^2 = \operatorname{var}(y_t \mid \mathscr{F}_{t-1})$; where $\mathscr{F}_{t-1}$ denotes the information up to time $t-1$, in our case, it is the past asset returns. Training a local model assuming Gaussian errors can be expressed as:

$$y_t = \sigma_t \varepsilon_t, \quad \varepsilon_t \overset{\text{i.i.d}}{\sim} \mathscr{N}(0,1), \tag{1a}$$

$$\sigma_t = f(y_{1:t-1}), \quad t = 1, 2, \ldots, T, \tag{1b}$$

$$\ell(\boldsymbol{y}|\boldsymbol{\theta}) = -\frac{1}{T_{in}} \sum_{t=1}^{T_{in}} \log(p(y_t \mid \sigma_t)), \tag{1c}$$

where $f(\cdot)$ denotes the model that can be either an econometric or deep learning model, $\ell(\boldsymbol{y}|\boldsymbol{\theta})$ denotes the minimization objective, the averaged negative log-likelihood, $\boldsymbol{\theta}$ denotes the model parameters; $p(\cdot|\sigma_t)$ denotes the Gaussian density function with mean zero, and $T, , T_{in}, T_{out}$ denotes total, in-sample, out-of-sample steps, respectively.

In this paper, we refer this training approach as local training and models trained this way as local models or assets specific models as their parameters are optimized for a individual time series. When dealing with multiple time series, $\boldsymbol{Y} = \{\boldsymbol{y}^n, n = 1, \ldots, N\}$, there will N local models $f^1(\cdot), ..., f^N(\cdot)$ trained individually.

Global training: On the other hand, training a global model on multiple time series can be expressed as:

$$y_t^n = \sigma_t^n \varepsilon_t^n, \quad \varepsilon_t^n \overset{\text{i.i.d}}{\sim} \mathscr{N}(0,1), \quad t = 1, 2, \ldots, T_{in}, \quad n = 1, 2, \ldots, N \tag{2a}$$

$$\sigma_t^n = f^*(y_{1:t-1}^n) \tag{2b}$$

$$\ell(\boldsymbol{Y}|\boldsymbol{\theta}^*) = -\frac{1}{N} \frac{1}{T_{in}} \sum_{n=1}^{N} \sum_{t=1}^{T_{in}} \log(p(y_t^n|\sigma_t^n)). \tag{2c}$$

In this scenario, there is only one global or universal model for all time series, denoted as $f^*(\cdot)$. As stated in the introduction, the local training approach is effective for econometrics models because of their straightforward structure. However, when applied to deep learning models, it faces two challenges. Firstly, the vast number of financial assets in the real world makes it computationally expensive and often impossible to train, tune, and manage. More importantly, unlike econometrics models, deep learning models are heavily parameterized, and the limited data available for local training prevents them from capturing generalized patterns and achieving satisfactory performance. Conversely, global training addresses the issue of asset-specific nature by constructing a single universal model for all available time series. As the number of time series $\boldsymbol{Y}$ expands, the global model becomes more resilient to overfitting and exhibits improved generalization. Eventually, it can even generate accurate forecasts for previously unseen time series.

2.2 Deep learning models

In this section, a concise overview will be presented regarding the deep learning models employed in this study, namely RNN, GRU, LSTM, and Transformer. Readers uninterested in these can opt to bypass this section. These models essentially function as autoregressive mapping functions, represented as $f : (y_{1:t}) \to \sigma_{t+1}$. They operate similarly to GARCH but possess distinct internal structures.

RNN Recurrent Neural Networks (RNNs) are a class of neural networks designed for processing sequential data. They are particularly well-suited for time series analysis, making them useful in financial volatility forecasting. An RNN processes inputs sequentially, maintaining a hidden state vector that captures information from previously seen inputs. The basic update equation for a standard RNN is given by:

$$h_t = \psi\left(W_{hy}y_t + W_{hh}h_{t-1} + b_h\right) \tag{3a}$$

$$\sigma_t = \text{softplus}\left(W_{\sigma h}h_t + b_\sigma\right) + \varepsilon \tag{3b}$$

where y_t is the input at time step t, h_t is the hidden state at time step t, σ_t is the output at time step t, W_{hy}, W_{hh}, and $W_{\sigma h}$ are the weight matrices, b_h and b_σ are the bias vectors, ψ represents the sigmoid function. To ensure positive constraint for σ_t, a softplus$(\cdot)$ function and $\varepsilon = 1e^{-8}$ is added.

GRU The Gated Recurrent Unit (GRU) is an advanced variant of RNN that aims to solve the vanishing gradient problem associated with standard RNNs. It introduces two gates: the reset gate r_t and the update gate z_t, which help the model to capture dependencies over various time scales efficiently. The GRU's update equations are:

$$r_t = \psi\left(W_{ry}y_t + W_{rh}h_{t-1} + b_r\right) \tag{4a}$$

$$z_t = \psi\left(W_{zy}y_t + W_{zh}h_{t-1} + b_z\right) \tag{4b}$$

$$\tilde{h}_t = \tanh\left(W_{hy}y_t + W_{hh}\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{4c}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{4d}$$

$$\sigma_t = \text{softplus}\left(W_{\sigma h}h_t + b_\sigma\right) + \varepsilon \tag{4e}$$

where $\tilde{h}_t$ is the candidate activation.

LSTM LSTMs are specifically designed to avoid the long-term dependency problem, enabling them to remember information for longer periods. They use three gates: the input gate (i_t), the forget gate (f_t), and the output gate (o_t), in conjunction with a cell state (C_t), to regulate the flow of information. The LSTM update equations are:

$$f_t = \psi\left(W_{fy}y_t + W_{fh}h_{t-1} + b_f\right) \tag{5a}$$

$$i_t = \psi\left(W_{iy}y_t + W_{ih}h_{t-1} + b_i\right) \tag{5b}$$

$$o_t = \psi\left(W_{oy}y_t + W_{oh}h_{t-1} + b_o\right) \tag{5c}$$

$$\tilde{C}_t = \tanh\left(W_{Cy}y_t + W_{Ch}h_{t-1} + b_C\right) \tag{5d}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{5e}$$

$$h_t = o_t \odot \tanh\left(C_t\right) \tag{5f}$$

$$\sigma_t = \text{softplus}\left(W_{\sigma h}h_t + b_\sigma\right) + \varepsilon \tag{5g}$$

Decoder-only Transformer The Decoder-only Transformer represents a significant advancement in sequence modeling by leveraging self-attention mechanisms, $\text{Attention}(Q,K,V,M) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V$, to forecast future values based on past observations. It is particularly effective and scalable for tasks requiring an understanding of long-range dependencies in sequential data and it has fundamentally transformed the NLP field. While we offer a brief overview of the Transformer here, it is challenging to delve into all the components in detail within this context. Interested readers are encouraged to refer to the original transformer paper for a more comprehensive understanding [14].

1. Input Embedding and Position Encoding

   $X_0 = \text{Embedding}\,(y_t) + \text{ PositionEncoding }(t)$

2. Multi-Head Self-Attention

   $Q, K, V = X_0$

   $\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V, M\right)$

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

3. Layer Normalization and Residual Connection

   $\text{LN}_{\text{att}}\,(X_0) = \text{LayerNorm}(X_0 + \text{MultiHead}(Q, K, V, M))$

4. Position-wise Feedforward Networks

   $\text{FFN}(x) = \max\,(0, xW_1 + b_1)\, W_2 + b_2$

   $X_L = \text{LN}_{\text{ff}}\,(\text{LN}_{\text{att}}(X_0) + \text{FFN}\,(\text{LN}_{\text{att}}(X_0)))$

5. Output Layer with Softplus Function

   $\sigma_t = \text{Softplus}\,(X_L W_\sigma + b_\sigma) + \varepsilon$

3 Simulation Study

This section compares local training and global training on simulated GARCH processes to highlight three main points: 1. Econometrics models should not be trained globally and their performance decreases as the number of training series increase. 2. Local deep learning models underperform their straightforward econometrics counterparts. 3. deep learning models should be trained globally and their performance improves as the number of training time series increase. 4. With a sufficiently large dataset, deep learning models can acquire the capability to forecast unseen time series accurately, even those not included in the training data. To this end, we simulated 110,000 distinct GARCH (1,1) series where each series consisted of 4,000 observations. Subsequently, each time series was divided into training, validation, and testing period

	ω	α	β
10	0.017	0.145	0.851
100	0.039	0.188	0.789
1000	0.041	0.187	0.787
10000	0.041	0.191	0.781
100000	0.041	0.193	0.779

Table 1: The estimated parameters of GARCH models trained with different number of series.

in proportions of 60%, 20%, and 20%, respectively. Out of the 110,000 simulated GARCH series, 100,000 of them were used for training the global model referred as training series. The remaining 10,000 were used to assess the zero-shot forecast capability of the global models, referred as unseen series

3.1 Global GARCH

First, we trained five global GARCH models with different numbers of series ranging from $1e^1$ to $1e^5$. And then evaluate their forecast accuracy on the same ten series, which were part of the training data for all five models. Notably, the $1e^1$ model was effectively trained on these ten series.

Figure 1(a) plots how average testing NLL and MSE varies against the number of series used for training, in comparison to the local GARCH benchmarks for the same set of ten series. It is clear that global training significantly worsen the forecast accuracy even when it is trained with only ten series. And the forecast accuracy continue to decrease as the number training series increase. This result is expected, as there are only three parameters for the GARCH model thus there is no way it can fit more than two series simultaneous, when it is trained globally, it will simply try to find a GARCH process that best represent the characteristics of all training series. Table 1 list the estimated parameters for all five global models, as shown the parameters for $1e^3, 1e^4, 1e^5$ models are nearly identical and their NLL and MSE are also nearly identical in Figure 1(a). At this point, the global training simply find a GARCH process that best represent the characteristics of population.

3.2 Global Deep Learning

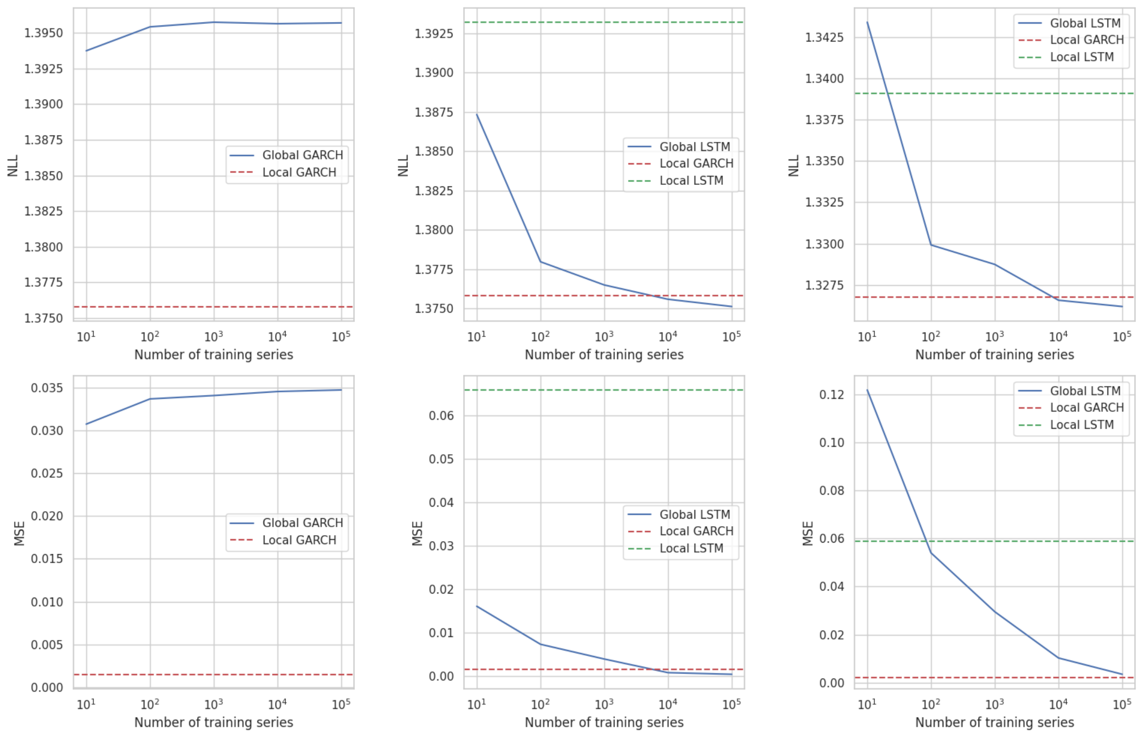
Similar to the global GARCH, we globally trained five neural network (NN) models with different data size range from $1e^1$ to $1e^5$. To ensure that the model size wouldn't be a limiting factor for larger training set, we opted for a two layer LSTM model with 128 hidden states and use early stopping to prevent overfitting on smaller datasets. To benchmark the global models, we also trained local NN models for individual series. For local NN models we opted a smaller architecture - a single layer LSTM with 32 hidden states. Both gloabl and local models are trained with same constant learning rate, $1e^{-4}$.

We reports the scaling effect of NN models on two subset of simulated GARCH series: 1. 10 training series that included in all five models training set. 2. 10000 new series that not included in any training set. The former is to show that even only a small set of time series is of the interest, training DL models with larger dataset is beneficial while the latter is used to evaluate the model's zero-shot capacity. Figure 1(b) plots the scaling effect on ten training series together with local NN and local GARCH benchmarks. From the results, we can see locally trained DL models perform significantly worse than local GARCH (which is reasonable since we are fitting GARCH models on simulated GARCH series). While in contrast to the GARCH models, the predictive accuracy of DL models showed significant enhancement as the number of training series increased. Notably, the model is trained with more than 10,000 series, the global DL models even outperformed the local GARCH models. Figure 1(c) plots the scaling effect on 10,000 unseen series. Clearly, zero-shots forecast is a more difficult tasks for global models. the $1e^1$ and $1e^2$ global models perform worse than local DL benchmarks. While as the training size increase, the accuracy of zero-shot forecasts increase significantly when the model is trained with more than 100,000 the zero-shot forecasts is close to local GARCH models.

Figure 2 and plots the volatility forecasts produced global DL models with varying training series size. The plots focus on the last 50 time steps for better visualization. It is evident that for both training series and unseen series, as the number of training series grows, the accuracy of the forecasts improves. In particular, the global model trained with $1e^5$ series almost perfectly aligns with the ground truth.

4 Empirical Study

The aim of this section is to investigate the data scaling effect and the reliability of a universal volatility model when applied to real-world financial data. In this section, we will use GARCH, GJR, and EGARCH as three representative baseline models to demonstrate the prediction accuracy of global DL models. While there are a myriad of popular econometrics volatility models by now, our focus is to highlight the data



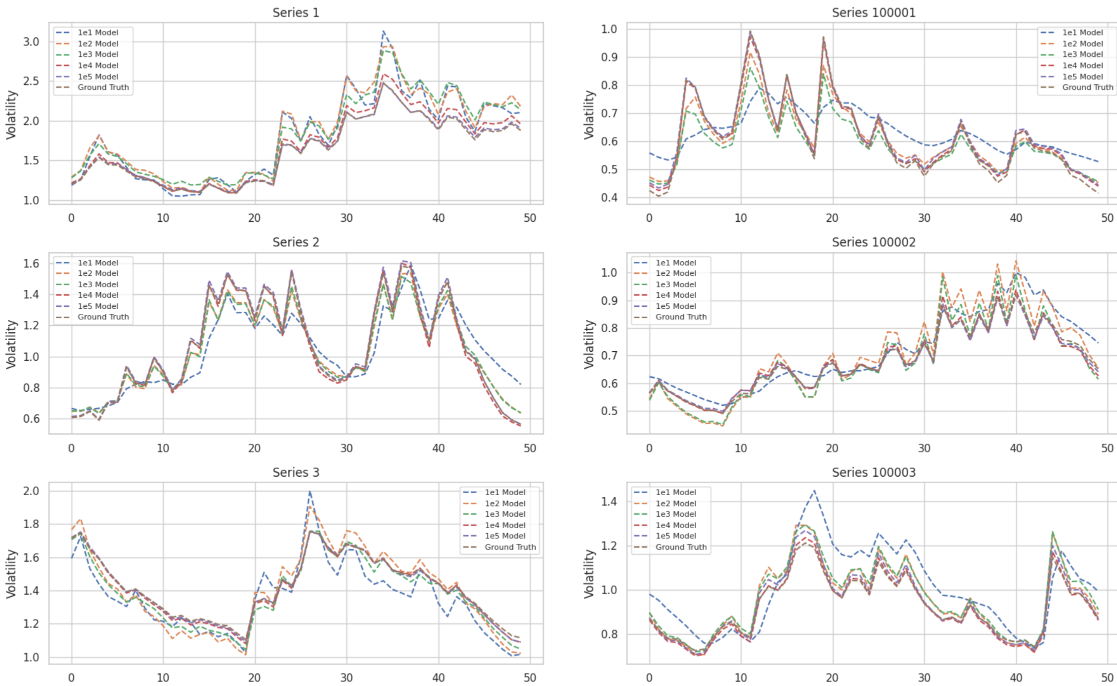
(a) GARCH on 10 training series. (b) LSTM on 10 training series. (c) LSTM on 10,000 Unseen series.

Figure 1: Scaling effect on simulation data.

scaling effect for deep learning models rather than conduct a model horse racing or claim superiority over any specific econometrics model.

We download daily data for over 12,000 stocks across ten exchanges, covering a decade from 01/01/2014 to 01/01/2024. For simplicity, we split the data using the 60%/20%/20% convention. The training period spans 2014 to 2020, the validation period spans 2020 to 2022, and the testing period spans 2022 to 2024. To ensure the local models can be effectively trained, we excluded stocks with fewer than 1,260 (five years) observations during the training period, resulting in 11,765 stocks remaining. Please refer to Table **??** for a breakdown of the data sources. Similar to the simulation study, we allocated 10,000 stocks for training the global models and reserved the remaining 1,765 stocks for zero-shot forecast evaluation. Given the in-sample closing prices $\{P_t, t = 0, ..., T_{in}\}$, we computed the demeaned close-to-close return process for each stock as:

$$y_t = 100\left(\log\frac{P_t}{P_{t-1}} - \frac{1}{T_{in}}\sum_{i=1}^{T_{in}}\log\frac{P_i}{P_{i-1}}\right), \quad t = 1, 2, \ldots, T_{in}. \tag{6}$$



(a) Training series. (b) Unseen series.

Figure 2: Volatility forecast of deep learning models with different number of training series. We zoom in to the last 50 steps for better visualization.

4.1 Scaling effect

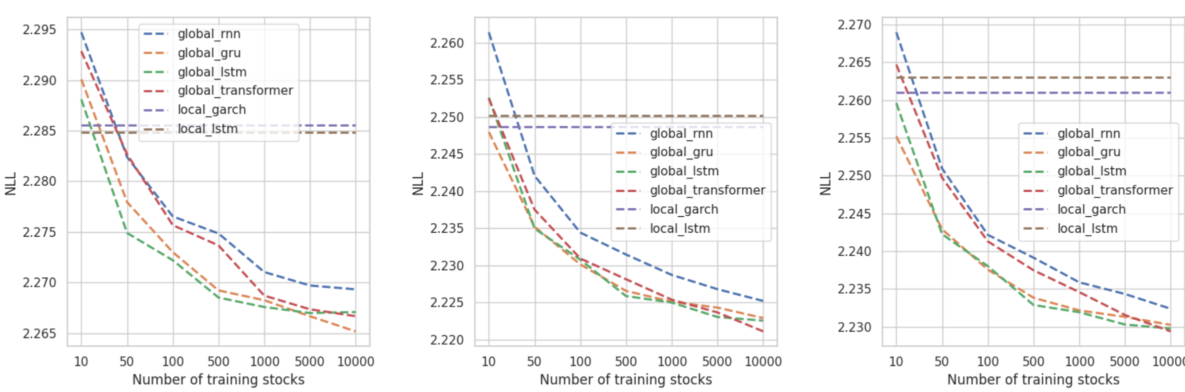
In a similar manner to simulation studies, we trained multiple global DL models using different data sizes and assessed their performance on a common set of stocks. To explore how the network architecture reacts to scaling effects, we repeated the experiments four times each for RNN, GRU, LSTM, and Transformer.

Considering the smaller data size compared to the simulation studies, we employed a smaller architecture for global models. RNN, GRU, and LSTM were set up with a single layer and 128 hidden states, while the Transformer was configured as follows: $d_{model} = 128$, $d_{ff} = 128$, $n_{heads} = 2$, and $n_{layers} = 2$. Due to the computational intensity of training 12,000 local DL models, we opted to use only local LSTM models as the baseline, each featuring a single layer and 32 hidden states. To ensure simplicity and reproducibility, we minimize the need for hyperparameter tuning. All models underwent training with a consistent learning rate set at $1e^{-}4$ and a fixed random seed of 0, devoid of any dropout layers. The impact of data scaling is demonstrated across three scenarios: on 10 training stocks, on 1,756 unseen stocks, and on all 11,756 stocks. The results are presented in Figure 3 and key findings can be summarized as follows:

- DL models trained locally outperform GARCH models slightly on 10 training stocks. Nevertheless, their performance tends to worse than GARCH when tested on a larger scale of dataset.
- Consistent with our findings from simulation studies, the data scaling remains evident for empirical data, where deep learning models exhibit improved performance with increasing training size.
- Although different model architectures result in different forecast accuracy, the scaling effect remains consistent. In other words, the scaling effect is not influenced by the model architecture.

Additionally, there are some discrepancies in results compared to simulation studies:

- Both local and global deep learning models exhibit better relative performance to GARCH compared to simulation studies. This discrepancy arises because in simulation studies, GARCH models are fitted on a GARCH process, whereas real stocks do not necessary follow a GARCH process.
- As illustrated in Figure 4.1, deep learning models develop zero-shot forecasting abilities more rapidly in real-world scenarios than in simulation studies. Even with just 50 stocks in the training set, the global model is capable of producing fairly accurate forecasts. This phenomenon may be attributed to the similarity among stock series caused by underlying market movement. In contrast to the markedly different simulated series, stock data exhibit stronger correlations, making them more predicable for global models.



(a) 10 training stocks (b) 1500 unseen stocks (c) All stocks

Figure 3: Scaling effect on empirical data

4.2 Universal volatility model

We now turn attention to assess the validity of a universal volatility model referred as DeepVol here. Through our experiments on the scaling effect, we observed that the LSTM architecture outperformed the other three candidates marginally. Consequently, we opted to build DeepVol with an LSTM architecture featuring the same single-layer configuration with 128 hidden states as in the scaling studies. Beside NLL, our evaluation of DeepVol will center on its application in risk management. In addition to examining its overall performance, it is crucial to assess how DeepVol operates at the individual stock level. Due to the impracticality of scrutinizing each stock individually, we will evaluate their individual performances using the Model Confidence Set (MCS) introduced by [6]. This section will start with a brief overview of risk metrics and the MCS. Furthermore, we include two additional econometric models as baselines, GJR and EGARCH.

Risk metrics: One of the most important application of volatility forecast is risk management, particularly in the prediction of Value at Risk (VaR) and Expected Shortfall (ES). The accuracy of the VaR forecast can be assessed using the standard quantile loss function [8]:

$$\mathrm{Qloss}_{\alpha} := \sum_{y_t \in D_{test}} (\alpha - I(y_t < Q_t^{\alpha}))(y_t - Q_t^{\alpha}), \tag{7}$$

where Q_t^{α} is the forecast of α-level VaR at time t. The quantile loss function is strictly consistent [3], i.e., the expected loss is lowest in the true quantile series. Therefore, the most accurate VaR forecasting model should minimize the quantile loss function. There is no strictly consistent loss function for ES; however, [3] found that ES and VaR are jointly elicitable, i.e., there is a class of strictly consistent loss functions for evaluating VaR and ES forecasts jointly. [11] shows that the negative logarithm of the likelihood function built from the Asymmetric Laplace (AL) distribution is strictly consistent for VaR and ES considered jointly. This AL based joint loss function is

$$\mathrm{JointLoss}_{\alpha} := \frac{1}{T_{\text{test}}} \sum_{D_{\text{test}}} \left(-\log\left(\frac{\alpha - 1}{\mathrm{ES}_t^{\alpha}}\right) - \frac{(y_t - Q_t^{\alpha})(\alpha - I(y_t \leq Q_t^{\alpha}))}{\alpha \mathrm{ES}_t^{\alpha}} \right) \tag{8}$$

with ES_t^{α} the forecast of α-level ES of y_t. Following ecnometrics literature convention, we reports both quantile loss and joint loss at 1% and 2.5% quantile.

Model confidence set: Let $\mathscr{M}$ be a set of competing models. A set of superior models (SSM) is established under the MCS procedure, which consists of a series of equal predictive accuracy tests given a specific confidence level. Let $L_{i,t}$ be a performance loss, such as the MSE or quantile loss, incurred by model $i \in \mathscr{M}$ at time t. Define $d_{i,j,t} = L_{i,t} - L_{j,t}$ to be the relative loss of model i compared to model j at time t. The MCS test assumes that $d_{i,j,t}$ is a stationary time series for all i, j in $\mathscr{M}$, i.e., $\mu_{i,j} = \mathbb{E}(d_{i,j,t})$ for all t. By testing the equality of the expected loss difference $\mu_{i,j}$, MCS determines if all models have the same level of predictive accuracy. The null hypothesis is

$$H_0 : \mu_{i,j} = 0, \quad \text{for all } i, j \in \mathscr{M}. \tag{9}$$

A model is eliminated when the null hypothesis H_0 of equal forecasting ability is rejected. The collection of models that do not reject the null hypothesis H_0 is then defined as the SSM. For each model $i \in \mathscr{M}$, the MCS produces a p-value p_i. The lower the p-value of a model, the less likely that it will be included in the SSM. See [6] for more details.

Results The results are presented in Table 2 and Table 3 and can be summarized as follows:

- On average, local LSTM models underperform all three econometrics baselines. However, when examining the MCS, we can see that local LSTM is included in the SSM more frequently than GARCH and GJR models. This aligns with our research findings which indicate that the performance of local DL models is heavily dependent on the evaluation dataset. While these models may excel with certain stocks, they may perform poorly with others. In contrast, econometrics models demonstrate more consistent performance. Therefore, although local LSTM models tend to perform worse on average, they have a higher likelihood of being included in the SSM when they do perform well. This could explain why many studies suggest that DL models are superior to econometrics models. In conclusion, our study did not find evidence that local DL models consistently outperform their econometrics counterparts.

- On average, DeepVol demonstrates a clear improvement over local LSTM across all metrics and exhibits competitive accuracy compared to econometrics baselines. At the individual stock level, it also demonstrates reliable performance as it is included in the SSM for most of the stocks.

	Local GARCH	Local GJR	Local EGARCH	Local LSTM	Global LSTM
NLL	2.261	2.257	2.256	2.266	2.230
QLoss 1%	0.425	0.425	0.431	0.427	0.418
QLoss 2.5%	0.160	0.160	0.163	0.164	0.156
JointLoss 1%	2.489	2.487	2.486	2.496	2.460
JointLoss 2.5%	2.840	2.835	2.834	2.862	2.787

Table 2: Risk metrics of local GARCH, GJR, EGARCH, LSTM and global LSTM averaged over all 11765 stocks.

	Local GARCH	Local GJR	Local EGARCH	Local LSTM	Global LSTM
NLL	2021	1867	3106	2775	8791
QLoss 1%	2492	2088	3491	2622	8538
QLoss 2.5%	2399	1988	3371	2843	8440
JointLoss 1%	2535	2189	3439	2417	8633
JointLoss 2.5%	2393	2045	3309	2544	8884

Table 3: MCS: the number of times each model is included in the SSM.

	Local GARCH	Local GJR	Local EGARCH	Local LSTM	Global LSTM
NLL	0.105	0.096	0.186	0.171	0.684
QLoss 1%	0.135	0.108	0.211	0.154	0.659
QLoss 2.5%	0.132	0.104	0.208	0.170	0.653
JointLoss 1%	0.141	0.112	0.209	0.138	0.668
JointLoss 2.5%	0.130	0.103	0.195	0.146	0.692

Table 4: MCS: the averaged p value for each model.

4.3 Connection to transfer learning

Transfer learning [12, 9] is a machine learning terminology where a model trained on one task is repurposed or adapted to a related task. Instead of starting the learning process from scratch, transfer learning leverages knowledge gained from solving one problem and applies it to a different, but related, problem. Deep neural networks often face challenges of overfitting and limited generalization when there are a large number of parameters but inadequate training data. Transfer learning greatly mitigate this problem.

Whether label our studies as transfer learning or global training depends on if the econometrics community view forecasting various stocks as different tasks. From our experience while most literature effectively treat them as different tasks by training models locally, the community won't consider a universal stock volatility model as an instance of transfer learning.

A typical transfer learning formalizes a two-phase learning framework: a pre-training phase to capture knowledge from one or more source tasks, and a fine-tuning stage to transfer the captured knowledge to target tasks. In the spirit of this, we conducted an experiment to fine-tune DeepVol individually for each stock, with the aim of enhancing its performance. Due to the relatively large size of DeepVol compared to the limited sample size of a single stock, we try to minimize task-specific parameterization. Thus, we used a layer freezing fine-tuning method, whereby the parameters of the LSTM layers were frozen, and only the output layer was fine-tuned. We still used a constant learning rate of $1e^{-4}$. Results are presented in Figure 4.3 and can be summarized as follows:

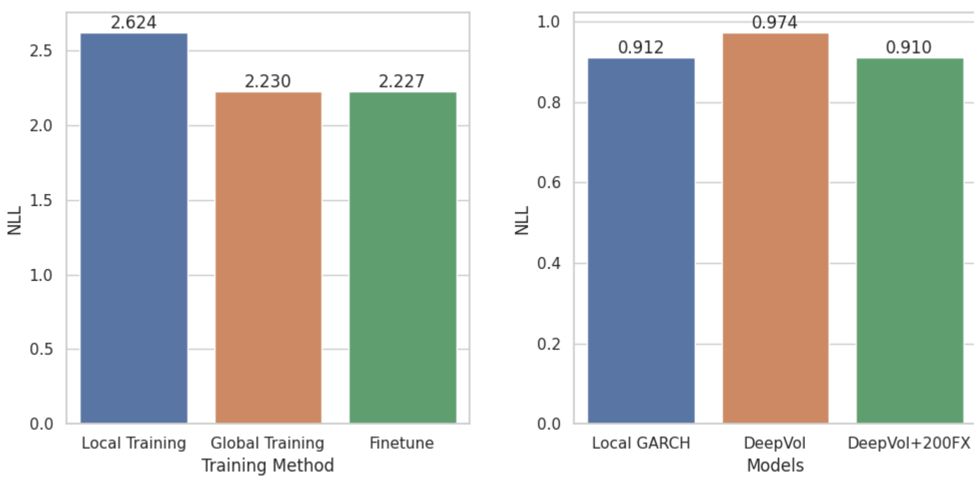
- Fine-tuning DeepVol on individual stocks enhances performance, however marginally, with no statistical significance.
- For most stocks, fine-tuning leads to a decrease in training loss while validation and testing losses show an immediate increase, indicating overfitting. The fine-tuning process for several stocks are plotted in Figure 5.
- The findings suggest that global training is more effective in mitigating overfitting and identifying optimal optimization minima compared to local models.

On the other hand, a universal model capable of forecasting various asset classes can be seen as an implementation of transfer learning. Given the limitations of this study, apart from stock data, we only had access to foreign exchange (FX) data. Therefore, we download daily data for 200 FX rates, also spanning a decade from 2014 to 2024, and applied the same training/validation/testing split with the stock data. For the FX volatility predictions, we evaluate three models: GARCH baseline, DeepVol and DeepVol 2.0 trained with both 10,000 stock series and the 200 exchange rate series. Results are presented in Figure 4.3 and can be summarized as follows:

- DeepVol can forecast the volatility of FX rates with reasonable accuracy; however, it still underperform local GARCH models, especially compared to their relative performance for stock data. This indicates that no matter how large the training size is, training DL models on only one type of asset is not enough to build a universal model for all assets.

- Forecast accuracy increases when the FX series are incorporated into the DeepVol's training set, approaching the GARCH baseline. Nevertheless, its relative performance with local GARCH models remains not as good as for stock data. This is primarily due to the small proportion of FX series in the training set compared to stock data.

Based on the outcomes of fine-tuning and predictions of FX volatility, we believe that the term "global training" is more appropriate for the scope of this research. However, the FX forecast experiment indicates a promising direction towards developing a truly universal model that can be applied to financial time series come form any asset class when more diverse training dataset is available.



(a) Fine-tuning on stocks.

(b) Forecasts of FX rate.

Figure 4: Transfer learning experiments.

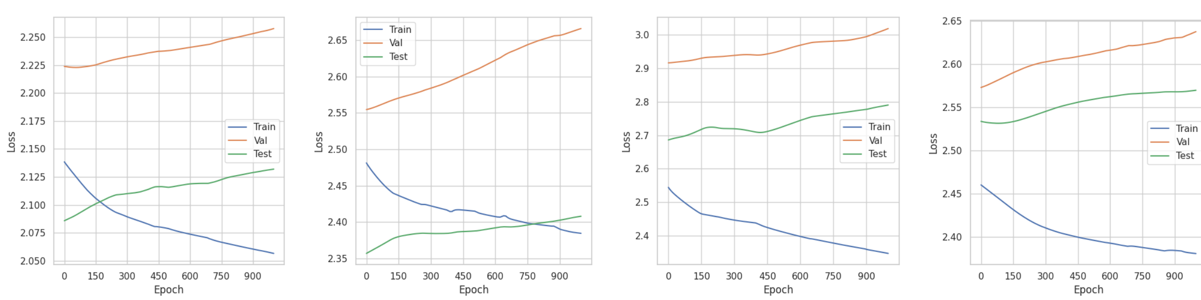


Figure 5: Fine-tuning process of four stocks.

5 Conclusion

This paper demonstrates that, in contrast to econometric models, the predictive accuracy of deep learning models notably enhances as the size of training data increases. Although locally trained DL models do not consistently outperform econometric models, when the training data is sufficiently large, a universal global DL model can produce comparable accurate forecasts even for data that was not seen before. We thus recommend adopting the global training approach as the new standard for training DL models in financial time series forecasting applications.

References

[1] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach". en. In: *Expert Systems with Applications* 140 (Feb. 2020), p. 112896.

[2] Abhimanyu Das et al. *A decoder-only foundation model for time-series forecasting*. en. Oct. 2023.

[3] Tobias Fissler and Johanna F. Ziegel. "Higher order elicitability and Osband's principle". In: *The Annals of Statistics* 44.4 (Aug. 2016), pp. 1680–1707.

[4] Azul Garza and Max Mergenthaler-Canseco. *TimeGPT-1*. Oct. 2023.

[5] Shihao Gu, Bryan Kelly, and Dacheng Xiu. "Empirical Asset Pricing via Machine Learning". In: *The Review of Financial Studies* 33.5 (May 2020), pp. 2223–2273.

[6] Peter R. Hansen, Asger Lunde, and James M. Nason. "The Model Confidence Set". en. In: *Econometrica* 79.2 (2011), pp. 453–497.

[7] Claudio Hartmann et al. "Exploiting big data in time series forecasting: A cross-sectional approach". In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Oct. 2015, pp. 1–10.

[8] Roger Koenker and Gilbert Bassett. "Regression Quantiles". In: *Econometrica* 46.1 (1978), pp. 33–50.

[9] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (Oct. 2010), pp. 1345–1359.

[10] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. "Financial time series forecasting with deep learning : A systematic literature review: 2005–2019". In: *Applied Soft Computing* 90 (May 2020), p. 106181.

[11] James W. Taylor. “Forecasting Value at Risk and Expected Shortfall Using a Semiparametric Approach Based on the Asymmetric Laplace Distribution”. In: *Journal of Business & Economic Statistics* 37.1 (Jan. 2019), pp. 121–133.

[12] Sebastian Thrun and Lorien Pratt. “Learning to Learn: Introduction and Overview”. en. In: *Learning to Learn*. Ed. by Sebastian Thrun and Lorien Pratt. Boston, MA: Springer US, 1998, pp. 3–17. ISBN: 978-1-4615-5529-2.

[13] Juan R Trapero, Nikolaos Kourentzes, and Robert Fildes. “On the identification of sales forecasting models in the presence of promotions”. In: *Journal of the Operational Research Society* 66.2 (Feb. 2015), pp. 299–307.

[14] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.