Quantitative Finance and Economics

QFE, 10(4): 728–752.
DOI: 10.3934/QFE.2026028
Received: 17 January 2025
Revised: 22 April 2026
Accepted: 02 June 2026
Published: 10 November 2026

https://www.aimspress.com/journal/QFE

Research article

Do pretrained models improve financial time series forecasting?

Linghan Huang[1], Yunhan Li[2] and Xuan Feng[3,*]

[1] ISCTE Business School, Lisbon, 1649-026, Portugal
[2] University of Edinburgh Business School, Edinburgh, EH89YL, UK
[3] Independent Researcher, Nice, 06200, France

* **Correspondence:** Email: xuan.feng@edhec.com.

Appendix: Detailed formula for statistic and machine learning models

A.1 ARIMA statistic model

ARIMA is a classic time series prediction model that builds an autoregressive and moving average model after differential and stationary time series data.

For each asset–region pair (i, j), the ARIMA(p, d, q) model is specified as

$$\phi(B)(1-B)^d r_t^{(i,j)} = \theta(B)\varepsilon_t^{(i,j)}, \tag{A1}$$

where B denotes the backshift operator, $\phi(B) = 1 - \sum_{k=1}^{p} \phi_k B^k$, and $\theta(B) = 1 + \sum_{k=1}^{q} \theta_k B^k$. The innovation sequence is assumed to satisfy

$$\varepsilon_t^{(i,j)} \sim \mathcal{WN}(0, \sigma^2). \tag{A2}$$

Model orders (p, d, q) are selected using information criteria such as AIC and BIC.

The One-step forecast equals:

$$\hat{r}_{t+1|t} = \sum_{k=1}^{p} \phi_k \, r_{t+1-k} \tag{A3}$$

Forecast error:

$$e_{t+1} = r_{t+1} - \hat{r}_{t+1|t} \tag{A4}$$

Mean squared forecast error:

$$\mathrm{MSFE} = \mathbb{E}[e_{t+1}^2] \tag{A5}$$

Gaussian predictive density:

$$r_{t+1} \mid \mathcal{F}_t \sim \mathcal{N}(\hat{r}_{t+1|t}, \sigma^2) \tag{A6}$$

A.2 XGBoost machine learning model

XGBoost is a machine learning model based on the gradient improvement framework, with high efficiency and strong scalability, and can process large-scale data and perform well in a variety of prediction tasks.

Let $x_t^{(i,j)} \in \mathbb{R}^d$ denote the feature vector constructed from lagged returns and auxiliary variables. XGBoost approximates the forecasting function by an additive ensemble of regression trees:

$$\hat{r}_t^{(i,j)} = \sum_{k=1}^{K} f_k\,(x_t^{(i,j)}), f_k \in \mathcal{F}, \tag{A7}$$

where $\mathcal{F}$ denotes the space of decision trees. The model parameters are estimated by minimizing the regularized objective

$$\mathcal{L} = \sum_{t=1}^{T} \ell(r_t^{(i,j)}, \hat{r}_t^{(i,j)}) + \sum_{k=1}^{K} \Omega(f_k), \tag{A8}$$

with regularization

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \parallel w \parallel^2. \tag{A9}$$

The Empirical risk:

$$\hat{\mathcal{R}}(f) = \frac{1}{T}\sum_{t=1}^{T} \ell(r_t, f(x_t)) \tag{A10}$$

Population risk:

$$\mathcal{R}(f) = \mathbb{E}[\ell(r_t, f(x_t))] \tag{A11}$$

Excess risk:

$$\mathcal{E}(f) = \mathcal{R}(f) - \inf_{g \in \mathcal{F}} \mathcal{R}(g) \tag{A12}$$

Bias–variance decomposition:

$$\mathbb{E}[(r_t - \hat{r}_t)^2] = \mathrm{Bias}^2 + \mathrm{Var} + \sigma^2 \tag{A13}$$

Regularized minimizer:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{\mathcal{R}}(f) + \lambda \parallel f \parallel \tag{A14}$$

A.3 LSTM deep learning model

LSTM is a recurrent neural network specializing in processing sequence data with long-term dependencies and is widely used in fields such as time series prediction.

Let x_t denote the input vector at time t. The LSTM dynamics are defined by

Forget gate:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{A15}$$

Input gate:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{A16}$$

Cell candidate:

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{A17}$$

Cell state update:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{A18}$$

Output gate:

$$h_t = o_t \odot tanh(c_t) \tag{A19}$$

The forecast $\hat{r}_{t+1}$is obtained via a linear projection of h_t.

Loss function:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} (r_t - \hat{r}_t)^2 \tag{A20}$$

Gradient update:

$$\theta_{k+1} = \theta_k - \eta \nabla_\theta \mathcal{L} \tag{A21}$$

Vanishing gradient bound:

$$\parallel \frac{\partial h_t}{\partial h_{t-k}} \parallel \leq \prod_{j=1}^{k} \parallel f_{t-j} \parallel \tag{A22}$$

Stability condition:

$$\parallel f_t \parallel < 1 \tag{A23}$$

A.4 Transformer-based large models

Transformer-based models map an input sequence $X = (x_1, \ldots, x_T)$ into latent representations using self-attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V. \tag{A24}$$

Multi-head attention is defined as

$$\text{MHA}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)W^{O}. \tag{A25}$$

Temporal embeddings encode seasonality and frequency information:

$$z_t = x_t + e_t^{(pos)} + e_t^{(freq)}. \tag{A26}$$

Attention weight:

$$\alpha_{ij} = \frac{\exp(q_i^{\top} k_j)}{\sum_{l=1}^{T} \exp(q_i^{\top} k_l)} \tag{A27}$$

Context vector:

$$c_i = \sum_{j=1}^{T} \alpha_{ij}\, v_j \tag{A28}$$

Effective memory length:

$$\mathcal{M} = \max_{i,j} | \alpha_{ij} | \tag{A29}$$

Time complexity:

$$\mathcal{O}(T^2 d) \tag{A30}$$

Low-rank approximation (TimesFM motivation):

$$QK^{\top} \approx UV^{\top} \tag{A31}$$

AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)