HyperVIX: A GWO-Optimized ARIMA-LSTM Hybrid Model for CBOE Volatility Index Forecasting

Ran Wu
School of Economics and Management
Southeast University, Nanjing, China
E-mail: wuran@aa.seu.edu.cn

Abdullahi D. Ahmed
School of AISSC—RMIT University
Melbourne, Australia
E-mail: abdullahidahir.ahmed@rmit.edu.au

Mohammad Zoynul Abedin
School of Management
Swansea University, UK
E-mail: m.z.abedin@swansea.ac.uk

Hongjun Zeng**
Department of Financial Planning and Tax,
School of Accounting, Information Systems and Supply Chain
RMIT University, Melbourne, Australia
E-mail: hongjun.zeng@student.rmit.edu.au

****Corresponding author**

DATA AVAILABILITY STATEMENT

Data can be reasonably obtained from the corresponding author upon request.

ACKNOWLEDGEMENT

No specific funding.

CONFLICT OF INTEREST

We have no conflict of interest.

HyperVIX: A GWO-Optimized ARIMA-LSTM Hybrid Model for CBOE Volatility Index Forecasting

Abstract

This paper introduced HyperVIX, a novel hybrid framework that integrates ARIMA modeling, LSTM neural networks, and Grey Wolf Optimizer (GWO) to forecast the CBOE Volatility Index (VIX). Using a multilayered approach, HyperVIX first employs ARIMA to capture linear time series patterns, followed by LSTM networks that model the residuals to identify complex nonlinear relationships. The GWO algorithm optimizes the LSTM hyperparameters, enhancing the framework's ability to capture VIX's intricate dynamics. Empirical analysis demonstrates that HyperVIX significantly outperforms both traditional and contemporary financial forecasting models in terms of accuracy and robustness. Compared to single models, HyperVIX achieves approximately 15%, 12%, and 10% improvements in RMSE, MAE, and MAPE metrics respectively, with the R^2 value increasing by about 5%. Notably, the model exhibits exceptional performance during extreme market volatility periods, making it particularly valuable for risk management applications. This research contributes to the literature by providing an innovative and effective method for VIX forecasting while offering valuable insights for financial market volatility analysis and investment strategy optimization.

Keywords: Volatility Index; Forecasting; Deep Learning; ARIMA; LSTM; Grey Wolf Optimizer; Hybrid Model

1. Introduction

The volatility of financial markets has always been a focus for investors, regulatory agencies, and scholars. As an important indicator of market volatility and investor sentiment, the accurate forecast of the VIX index is significant for formulating investment strategies and managing risk (Whaley, 2009; Hansen et al., 2024; Saâdaoui and Rabbouch, 2024; Lu and Zeng, 2023). Recently, the uncertainty and volatility of global financial markets have significantly increased, particularly due to factors such as the COVID-19, geopolitical conflicts, and a slowdown in global economic growth. This further highlights the necessity for precise forecast of the VIX index (Apergis et al., 2023).

Traditional forecasting frameworks, such as ARIMA, GARCH, and related variants like the GARCH-MIDAS model, have been widely applied to predict the financial data in previous literature (Pai and Lin, 2005; Zolfaghari et al., 2021; Tumala et al., 2023; Pokou et al., 2024). These functions excel in capturing the linear characteristics and volatility clustering of data, but their strict assumptions about data distribution somewhat contradict the complexity of financial time series. Meanwhile, deep learning frameworks based on neural networks have made significant progress in time series forecasting, particularly the LSTM neural networks and Gated Recurrent Unit functions, which have shown significant advantages in capturing long-term dependencies in time series (Chung et al., 2014; Ding et al., 2024; Zeng et al., 2024). Recently, the Hierarchical Gated Recurrent Neural Network (HGRN) model proposed by Qin et al. (2024), which introduces forget gates with learnable lower bounds, has further optimized the performance of traditional neural networks. The proposal of this model fully demonstrates the feasibility of optimizing traditional neural networks. Additionally, the introduction of the Transformer function has brought a new wave of interest. Wang et al. (2022) used the Transformer function to stock price forecasting and conducted back testing in the Chinese market, proving its accuracy. Improved versions, such as Transformer-XL and Informer, have further enhanced the performance of time series forecasting and can handle long-sequence data more effectively (Zhou et al., 2021; Cheng et al., 2024; Wu et al., 2025).

Moreover, hybrid models and ensemble learning methods have received widespread attention for enhancing forecast accuracy by combining the strengths of various models. For example, researchers have explored various approaches to enhance forecasting accuracy, such as developing hybrid architectures that merged ARIMA with neural networks, and implementing ensemble techniques which incorporated several deep learnin' models (Lu et al., 2024; Qin et al., 2017). A novel hybrid framework was proposed by Jing et al. (2021), which combined CNN and LSTM to tackle the influence of investor sentiment on the China’s stock

market. This innovative approach successfully boosted the forecasting precision of investor sentiment within China's equity market. Additionally, with the development of optimization algorithms, algorithms such as PSO and GA have been applied to parameter tuning of forecasting models. Singh et al. (2020) achieved significant results in time series forecasting by introducing wavelet decomposition combined with ARIMA.

This study innovatively combines LSTM and ARIMA to fully leverage the advantages of ARIMA in handling linear structures while using LSTM to compensate for the shortcomings in nonlinear processing. The feasibility of this approach is supported by the research of Lin et al. (2021), who introduced a hybrid function combining CEEMDAN and LSTM to forecast stock index prices. Given that the VIX is a vital indicator of S&P 500 volatility, its accurate forecast is of great significance (Pinto et al., 2015; Ghosh and Jana, 2023; Abedin et al., 2024; Hansen et al., 2024).

Based on the above considerations, this research proposes an GWO-ARIMA-LSTM ensemble framework (HyperVIX) to forecast the trend of the VIX. The function first uses ARIMA to handle the linear section of the time series, then employs the LSTM function to estimate the residuals to capture the nonlinear characteristics in the data. Furthermore, to optimize the hyperparameters of the LSTM, the GWO algorithm is introduced to simulate the hunting behavior of gray wolves, searching for the optimal solution through swarm intelligence. It is noteworthy that using the GWO algorithm for hyperparameter optimization effectively avoids the limitations of traditional grid search and random search methods and enhances the forecast accuracy and stability through intelligent optimization. Figure 1 summarises the empirical process of this paper.

Empirical findings show that the HyperVIX functionl significantly outperforms other traditional and modern financial models in terms of forecast accuracy and robustness. Compared to single functions, this ensemble model reduces RMSE by approximately 15%, MAE by about 12%, and MAPE by around 10%, while improving the R^2 index by about 5%. The model performs exceptionally well in capturing extreme market volatility, providing an effective method for forecasting financial market volatility.

This research proposes a GWO-ARIMA-LSTM ensemble framework, named HyperVIX, to forecast the trend of the CBOE S&P 500 Volatility Index (VIX). The framework first employs ARIMA to capture the linear components of the time series, followed by an LSTM model to estimate residuals and capture nonlinear characteristics. The Grey Wolf Optimizer (GWO) is introduced to optimize LSTM hyperparameters, leveraging swarm intelligence to efficiently navigate the complex hyperparameter space. This approach enhances forecasting accuracy and robustness, particularly during periods of extreme market volatility, such as the COVID-19 crisis and the Russia-Ukraine war in 2022.

The HyperVIX framework makes several distinct contributions to the literature on financial time series forecasting. (1) Novel Integration with GWO: While hybrid models combining ARIMA and LSTM have been explored (e.g., Roszyk and Ślepaczuk, 2024; Kashif and Ślepaczuk, 2025; Stempie and Ślepaczuk, 2025), HyperVIX uniquely incorporates GWO to optimize LSTM hyperparameters. This results in superior performance, with approximately 15%, 12%, and 10% improvements in RMSE, MAE, and MAPE, respectively, compared to single models. (2) Enhanced Performance in Extreme Volatility: HyperVIX demonstrates exceptional forecasting accuracy during high-volatility periods, outperforming existing models in capturing dramatic fluctuations. This capability addresses a critical gap in prior studies, which often focus on stable market conditions. (3) Comprehensive Empirical Validation: Unlike prior works that compare with a limited set of benchmarks, our study evaluates HyperVIX against a broad range of models, demonstrating its robustness across multiple metrics and scenarios.

The rest of this work was structured as bellow: Section two offered an overview of prior research. Subsequently, the third section delved into the intricacies of constructin' the model and the methodology employed. The fourth section presented the empirical results and offered an in-depth estimation of the findings. Finally, the concludin' section summarized the key outcomes of the study and outlined potential avenues for future work.

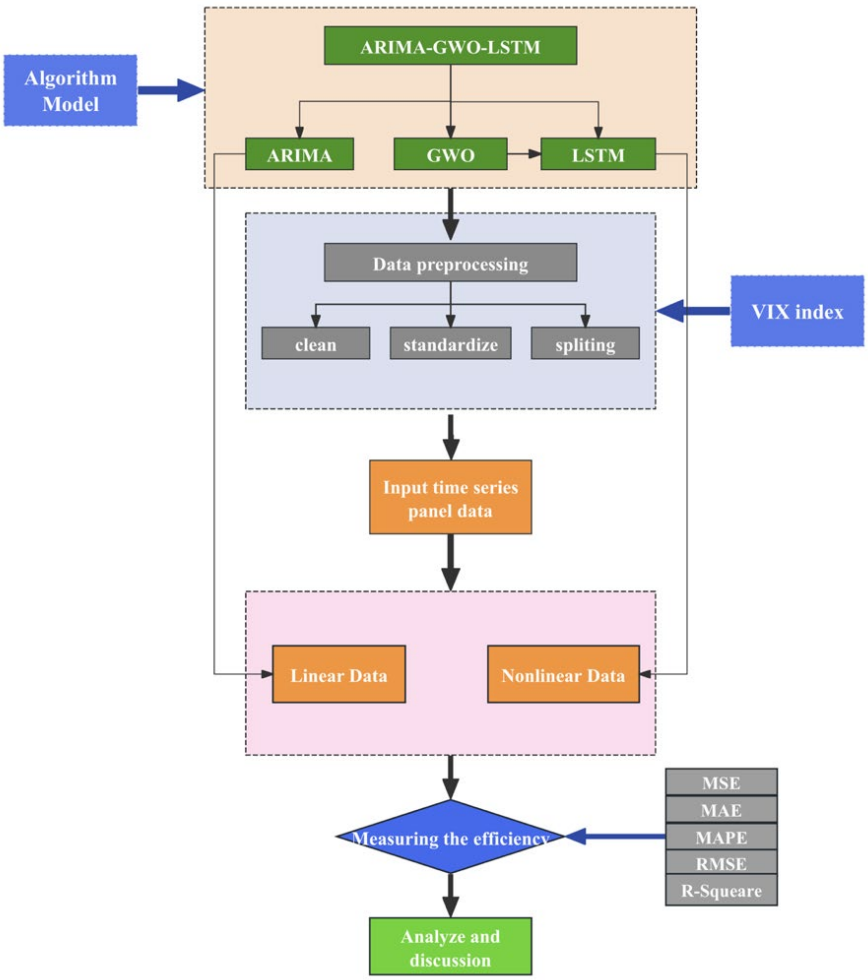

Figure 1. The forecast process of HyperVIX model.

2. Literature review

The forecast of VIX index volatility has long been a focal point for both academia and practitioners. In their review, Poon and Granger (2003) emphasised the important role of volatility forecasting in risk management and asset pricing. Christoffersen and Diebold (2000) further highlighted that accurate volatility forecasts are essential for portfolio optimisation and option pricing. As a significant market volatility indicator, the VIX index has garnered substantial attention. Whaley (2000) elaborated on the construction of the VIX index and its importance as a barometer of market sentiment. Meanwhile, Becker et al. (2009) demonstrated that the VIX index not only reflects expected market volatility but also encapsulates investor risk preferences. These studies lay the foundation for understanding the significance of the VIX index and the necessity of its forecasting.

In time series forecasting methods, the ARIMA function has held a prominent position for a long time. Box and Jenkins' (1976) seminal work on the ARIMA function laid the foundation for time series analysis, which had long been a dominant approach in forecasting. Enders (2008) provided an extensive review of the

ARIMA framework's applications and constraints in forecasting financial time series. However, as machine learning techniques advanced, particularly with the emergence of deep learning, researchers began investigating more intricate nonlinear models. The LSTM network (Hochreiter and Schmidhuber, 1997), tackled the long-term dependency limitations of conventional RNNs. Fischer and Krauss (2018) demonstrated the superiority of LSTM over traditional machine learning approaches in financial market prediction. These investigations offer theoretical justification for the application of deep learning methodologies to financial forecasting.

In recent years, hybrid models have shown immense potential in financial forecasting. Zhang (2003) introduced a hybrid model combining ARIMA and neural networks, significantly improving forecast accuracy. Kristjanpoller and Minutolo (2015) combined ANN with GARCH models, successfully enhancing the accuracy of VIX index forecasts. The aforementioned research suggested that combining various model types could capture both the linear and nonlinear properties of time series, thus enhancing forecasting accuracy. Nevertheless, the effectiveness of these composite models was largely contingent upon the choice and fine-tuning of model parameters. More recently, Roszyk and Ślepaczuk (2024) proposed a hybrid model integrating GARCH, and LSTM, demonstrating improved volatility forecasting. Similarly, Kashif and Ślepaczuk (2025) and Stempie and Ślepaczuk (2025) explored hybrid architectures combining LSTM with ARIMA, focusing on general market conditions. However, these previous studies do not explicitly optimize hyperparameters using advanced metaheuristic algorithms, nor do they focus on extreme volatility periods, which are critical for risk management.

To address the issue of model parameter optimisation, researchers have introduced various optimisation algorithms. The GWO algorithm introduced by Mirjalili et al. (2014) has shown excellent performance in solving complex optimisation problems. These studies inspire the incorporation of the GWO algorithm into the ARIMA-LSTM hybrid model, potentially further improving model forecast accuracy.

In evaluating model performance, Hyndman and Koehler (2006) provided a comprehensive review of evaluation metrics for time series forecasting models, including RMSE, MAE, and MAPE. Campbell (2007) emphasised the importance of using multiple evaluation metrics in financial forecasting to comprehensively assess model performance. These studies guide the selection of appropriate evaluation metrics.

Notably, forecasting during periods of extreme market volatility remains a challenging issue. Bates (2000) studied volatile behaviour under extreme market conditions, emphasising the difficulties and importance of forecasting during these periods. Andersen et al. (2003) proposed using high-frequency data to improve

volatility forecasts during high volatility periods. These studies remind us that an excellent VIX forecasting model should maintain stable performance under various market conditions.

As machine learning techniques have continued to advance, their utilisation in financial prediction has also exhibited novel tendencies. Gu et al. (2020) carried out an extensive evaluation of machine learning applications in asset pricing, emphasising the promise of deep learning approaches. Sezer et al. (2020) examined the most recent progress in deep learning for forecasting time series, encompassing a range of LSTM variations. These investigations offer guidance for delving into more sophisticated machine learning methodologies in VIX forecasting.

Finally, model integration and combination forecasting methods have gained increasing attention. Rossi (2021) discussed the application of forecast combinations in economics and finance in detail, highlighting their potential to improve forecasting robustness. Li et al. (2023) further demonstrated the superiority of combination forecasting methods in stock market forecasts, especially during periods of high uncertainty. Recent hybrid approaches have also shown promise in volatility forecasting; for example, Roszyk and Ślepaczuk (2024) developed a hybrid model combining GARCH with LSTM, which effectively captured volatility dynamics but highlighted the need for further optimization to handle extreme market conditions. Additionally, Kashif and Ślepaczuk (2025) explored LSTM-ARIMA hybrids for volatility prediction, emphasizing the benefits of integrating econometric and deep learning techniques. Stempień and Ślepaczuk (2025) analyzed various hybrid methodologies combining econometric, machine learning, and deep learning models, underscoring their enhanced accuracy in financial time series but noting gaps in hyperparameter optimization and robustness during turbulent periods. Michańków et al. (2024) addressed overfitting issues in similar models, proposing regularization strategies to improve generalization.

In summary, existing literature provides a rich theoretical and methodological foundation for VIX index forecasting, demonstrating that hybrid models can outperform standalone approaches by leveraging complementary strengths in capturing linear and nonlinear patterns. However, while these studies highlight the potential of combinations like ARIMA-LSTM and GARCH-LSTM, they often lack advanced optimization techniques for hyperparameters and comprehensive handling of extreme volatility, as seen in recent market disruptions. This gap provides a rational basis for our research, which aims to develop a novel HyperVIX framework to enhance forecasting accuracy, stability, and robustness, particularly in high-uncertainty environments, building on prior hybrid methodologies while addressing their limitations through swarm intelligence optimization.

3. Methodology

This research proposes a hybrid framework- HyperVIX, combining ARIMA, GWO, and LSTM for VIX index forecast. Specifically, we use the ARIMA structure to capture the linear characteristics of the data, the LSTM neural network to handle the nonlinear features in the residuals, and the hyperparameters of the LSTM are optimized applying the GWO algorithm. The following sections detail the methodology, including model construction, parameter selection, and the mathematical formulas and theoretical basis of the optimization process.

3.1. Support Vector Machine (SVM)

SVM is a supervised learning function developed for classification and regression analysis by Boser et al. (1992). It handles linearly inseparable data by mapping it to a higher-dimensional space.

Assume there is a set of training samples $\{(x_i, y_i)\}$, where x_i is the input feature vector and y_i is the class label (taking values of -1 or 1). We aim to find a hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ that separates the data points of the two classes as much as possible. The objective is to maximize the margin $\frac{2}{\|\mathbf{w}\|}$, which is equivalent to minimizing $\frac{1}{2} \| \mathbf{w} \|^2$. The constraint condition is $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$.

The final optimization problem is:

$$\min_{\mathbf{w},b} \frac{1}{2} \| \mathbf{w} \|^2 \tag{1}$$

And

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \tag{2}$$

Simultaneously, in the case of linearly inseparable data, developing slack indices $\xi_i > 0$, the optimization problem becomes:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \xi_i \tag{3}$$

where C is a regularization parameter applied to balance margin maximization and misclassification penalty.

To handle nonlinear problems, SVM applies kernel frameworks to map the data into higher-dimensional space. Common kernel frameworks include the linear, polynomial, and Gaussian kernel. The optimization problem becomes:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i \tag{4}$$

So that:

$$0 \le \alpha_i \le C, \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{5}$$

Where, $\mathbf{w}$ is normal vector of the hyperplane. b is the intercept of the hyperplane. ξ_i is slack indices, indicates the degree of misclassification. C is regularization parameter.

3.2. Random forests (RF)

Random Forest improves the accuracy and stability of models by combining the forecasting results of multiple decision trees (Breiman, 2001). The basic idea of Random Forest is to bring randomness to generate multiple different decision trees and combine the forecast outcomes of these trees to reduce model variance and improve generalization ability (Yu et al., 2022).

Random Forest introduces randomness through the following two key techniques: (1) Bootstrap Aggregating (Bagging): This method involves randomly sampling with replacement from the training dataset to produce multiple different subsets, each used to train a single decision tree. This method ensures that each tree has a certain diversity in the training data, thus reducing model variance. (2) Random Feature Selection: In this method, instead of using all features when splitting each node in a decision tree, a random subset of features is selected for every split. This technique increases the diversity between decision trees, thereby enhancing the overall performance of the framework.

The main content includes: (1) Objective Function: Each decision tree in a RF selects the optimal split by maximizing Information Gain or the Gini Index. (2) Information Gain Formula: For a split on feature X_i, the Information Gain (IG) is calculated as follows:

$$IG(D, X_i) = H(D) - \sum_{v \in \text{Values}\,(X_i)} \frac{|D_v|}{|D|} H(D_v) \tag{6}$$

where $H(D)$ is the entropy of the dataset D, and D_v is the subset of data where feature X_i takes the value v.

Random Forests are widely used in various classification and regression problems, especially when dealing with high-dimensional data, nonlinear data, and datasets with missing values.

3.3. CNN

CNN is a type of deep learning model proposed by LeCun et al. (LeCun et al., 1989). It extracts local features of the data through convolution operations, combines these features layer by layer, and finally performs classification or regression through the fully connected layer. Its main advantage is the ability to automatically learn the spatial hierarchical structure of the data. The main components are as follows:

(1) Convolutional Layer: Uses multiple convolutional kernels (filters) to perform convolution operations on the input data to local features. Assuming the input data is X, the convolution kernel is K, and the result of the convolution operation is Y, then $Y_{ij} = (X * K)_{ij} = \sum_m \sum_n X_{i+m,j+n} K_{mn}$.

(2) Pooling Layer: This layer applies down sampling to the convolutional layer's output, effectively reducing the feature data's dimensionality while preserving key features. Common pooling methods are max pooling and average pooling. For example, in max pooling: $Y_{ij} = max(X_{i:a,j:b})$, where a and b are the sizes of the pooling window.

(3) Fully Connected Layer: Flattens the output of the previous layer into a one-dimensional vector and performs classification or regression through a series of fully connected neurons. Assuming the input vector is \(x\), the weight matrix is $\mathbf{W}$, the bias is $\mathbf{b}$, and the output is $\mathbf{y}$, then $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$.

(4) Activation Function: Introduces nonlinearity; common activation frameworks include ReLU, Sigmoid, Tanh, etc.:

$$\text{ReLU: } f(x) = max(0, x) \tag{7}$$

$$\text{Sigmoid: } f(x) = \frac{1}{1+e^{-x}} \tag{8}$$

$$\text{Tanh: } f(x) = \tanh(x) \tag{9}$$

(5) Loss Function: measures the difference between the forecast result and the true value, which is applied to guide the update of model parameters:

$$\text{Cross Entropy Loss (for classification): } L = -\sum_i y_i \log(\hat{y}_i)) \tag{10}$$

$$\text{Mean Squared Error Loss (for regression): } L = \frac{1}{N}\sum_i (y_i - \hat{y}_i)^2 \tag{11}$$

CNN extract image features and perform classification or regression through successive convolution, pooling, and fully connected operations. By adjusting parameters such as the size of the convolutional kernels, stride, and pooling window, CNNs can achieve excellent results in various application scenarios.

3.4. LSTM

LSTM is a type of RNN applied for handling and forecasting time series data. Its main advantage lies in addressing the vanishing gradient and exploding gradient issues of traditional RNNs when processing long sequences by introducing a "gating" structure.

a) Forget Gate: The forget gate decides which parts of the cell state at the current time step need to be forgotten. Its calculation formula is:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{12}$$

Where f_t is the output of the forget gate, W_f is the weight matrix, b_f is the bias, h_{t-1} is the hidden state, and x_t is current time step.

b) Input Gate: The input gate was responsible for identifying which elements of the input data at the present time step should be incorporated into the cell state. The following equation was employed to compute it:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{13}$$

Where i_t is the result of the input gate.

c) Candidate Cell State: The candidate cell state generates new candidate cell states and determines which information will be updated into the cell state. Its calculation formula is:

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{14}$$

Where $\tilde{C}_t$ is the candidate cell state.

d) Update Cell State. Update the cell state of the current time step according to the output of the forget gate and the input gate. The calculation formula is:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{15}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{16}$$

e) Output Gate. The output gate was tasked with identifying which portions of the cell state at the present time step would be produced as hidden states. The following equation was used to calculate it:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{17}$$

f) Hidden State. The hidden state of the current time step was computed by utilising the output gate and the cell state of the present time step. The equation employed for this calculation was:

$$h_t = o_t \cdot \tanh(C_t) \tag{18}$$

Where h_t is the hidden state of the current time step.

Through the above mechanisms, LSTM effectively controls the flow of information in long time sequences. In this paper, the LSTM network is trained to predict the future values of the residuals r_t. This method has been proven effective. It is expressed as:

$$\hat{r}_{t+1} = LSTM(r_t, \Theta) \tag{19}$$

Where Θ represents the network parameters of LSTM.

3.5. GARCH

The GARCH function is an extended version of the ARCH model, developed by Lundbergh (Lundbergh and Teräsvirta, 2002). It describes the "clustering" and "persistence" characteristics of volatility in time series data by introducing lagged squared terms and conditional variance equations. Specifically, the GARCH function assumes that the conditional variance of a time series is related not only to past volatility but also to past conditional variances. It is worth noting that many previous studies have also used GARCH family models to forecast the VIX index (e.g., Pan et al., 2019; Hansen et al., 2024; Qiao et al., 2025).

Assuming r_t indicates the return at moment t, the basic form of the GARCH (1,1) function includes the mean equation and the conditional variance equation.

The mean equation is $r_t = \mu + \epsilon_t$. Here, μ is the mean, and ϵ_t is the random error term, which satisfies $\epsilon_t = \sigma_t z_t$. z_t is an independently and identically distributed random variable, usually assumed to follow a standard normal distribution.

The conditional variance equation can be expressed as $\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$, where σ_t^2 is the conditional variance at moment t. The parameters $\alpha_0 > 0, \alpha_1 \geq 0, \beta_1 \geq 0$ need to be estimated, with the condition $\alpha_1 + \beta_1 < 1$ to ensure model stability. The elements of the GARCH functionl are typically evaluated using the Maximum Likelihood Estimation (MLE) method.

The GARCH function's strength resided in its capacity to effectively characterise the "volatility clustering" phenomenon, which was prevalent in financial time series. This phenomenon was characterised by

large fluctuations often being succeeded by further large fluctuations, and small fluctuations frequently being followed by additional small fluctuations. Consequently, the GARCH model found extensive application in financial assets, such as in risk management, derivative pricing, and asset allocation.

3.6. ARIMA

The ARIMA function was employed to manage linear trends and periodic characteristics in time series data. It was denoted as ARIMA (p, d, q), where components p and q represented the orders of the AR and MA models, respectively, and parameter d indicated the degree of differencing applied to the data. The fundamental structure of the ARIMA (p, d, q) structure was expressed as follows:

$$\begin{aligned}\hat{x}_t &= c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} \\ &\quad -\theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \\ &= c + \sum_{k=1}^{p} \phi_k x_{t-k} - \sum_{l=1}^{q} \theta_l \varepsilon_{t-l}\end{aligned} \tag{20}$$

where c is the constant term, y_t is the forecast value at time t, ϕ_k are the autoregressive coefficients, ε_{t-l} is the white noise error term, and θ_j are the moving average coefficients. Differencing the time series to achieve stationarity is performed. If the original time series is non-stationary, differencing d times results in a stationary series, that is:

$$\Delta^d \hat{x}_t = (1 - B)^d \hat{x}_t \tag{21}$$

where Δ represents the differencing operation and B is the lag operator.

Generally, the ACF is used to identify q, and the PACF is used to identify p. The ARIMA model is fitted using historical data, and the white noise characteristics of the residuals are checked to ensure no significant patterns remain in the residuals. After fitting the ARIMA model, the residuals r_t are calculated as $r_t = y_t - \hat{y}_t$.

where y_t is the predicted value, and r_t represents the residual series, containing the nonlinear part of the data not captured by the ARIMA model (Zhang, 2003). Following Choi (2018), we iteratively select the three parameters of the ARIMA model. Once the ARIMA model is fitted, forecasts are made every 21-time steps to calculate residual values. The last data point of each set becomes the target variable Y, and the preceding points become the variable X. This newly created X/Y split dataset will be used as the input for the subsequent LSTM model stage. By first applying ARIMA to forecast the linear component ($\hat{y}_t^{\text{ARIMA}}$), we generate residuals ($e_t = y_t - \hat{y}_t^{\text{ARIMA}}$) that contain the nonlinear residuals. LSTM then models these residuals

to capture complex patterns, producing a final forecast $\hat{y}_t = \hat{y}_t^{ARIMA} + \hat{e}_t^{LSTM}$. This decomposition reduces overall error by allowing each model to focus on its strength.

3.7. Proposed GWO-LSTM hybrid models

3.7.1. Grey wolf algorithm optimization process

The GWO algorithm is a swarm intelligence optimization method inspired by the hunting behavior of grey wolves. It replicates the hierarchical framework and hunting strategies of grey wolf packs, incorporating the roles of alpha (α), beta (β), delta (δ), and omega (ω) wolves to balance global and local search optimization. In GWO, the population is categorized into four groups: alpha wolves act as leaders responsible for decision-making, beta wolves assist the leaders and maintain group stability, delta wolves coordinate and execute tasks, and omega wolves, as the lowest rank, follow the actions of higher-ranking wolves.

Initially, the positions of grey wolf individuals are randomly initialized, with each position representing a set of LSTM hyperparameters. The fitness of each individual is then assessed based on the LSTM function's forecasting performance. Finally, the positions of the remaining individuals are updated using the fitness value of the best-performing individual (α).

$$\mathbf{A} = 2a \cdot r_1 - a \tag{22}$$

$$\mathbf{C} = \mathbf{2}r_2 \tag{23}$$

$$\mathbf{D} = |\mathbf{C} \cdot \mathbf{X}_\alpha - \mathbf{X}| \tag{24}$$

$$\mathbf{X}(t+1) = \mathbf{X}_\alpha - \mathbf{A} \cdot \mathbf{D} \tag{25}$$

Where, $\mathbf{X}_\alpha$ is the position of wolf α, $\mathbf{X}$ is the current wolf position, $\mathbf{C}$ and $\mathbf{A}$ are control parameters, r_1 and $r_2 \in [0,1]$ are two random numbers, a is convergence factor, t is iterations.

Subsequently, the wolf pack begins searching for and encircling the prey. As the decision space is uncertain, the positions of the α, β, and δ wolves are utilized to guide the search, iterating progressively toward the optimal solution. The position update process is defined as follows:

$$D_\alpha = |C_1 \cdot x_\alpha(t) - x_i(t)| \tag{26}$$

$$D_\beta = \left|C_2 \cdot x_\beta(t) - x_i(t)\right| \tag{27}$$

$$D_\delta = |C_3 \cdot x_\delta(t) - x_i(t)| \tag{28}$$

$$x_1(t) = x_\alpha(t) - A_1 \cdot D_\alpha \tag{29}$$

$$x_2(t) = x_\beta(t) - A_2 \cdot D_\beta \tag{30}$$

$$x_3(t) = x_\delta(t) - A_3 \cdot D_\delta \tag{31}$$

$$x_i(t+1) = \frac{1}{3}\sum_{k=1}^{3} x_k(t) \tag{32}$$

Here, $x_\alpha(t)$, $x_\beta(t)$ and $x_\delta(t)$ represent the positions of the three leading wolves, while D_α, D_β, D_δ denote the distances between these wolves and other individuals. $x_i(t)$ and $x_i(t+1)$ refer to the current position of the ω wolf (current solution) and the updated optimal solution vector at iteration $t+1$. Based on formulas (32) to (38), the convergence factor aaa influences the selection of the random vector A, where A is a random value within the range $[-a, a]$. When $|A| > 1$, the grey wolves disperse to explore a wider search space, seeking better potential solutions. Conversely, when $|A| > 1$, the wolves focus their efforts on a specific region to refine the search and capture the prey. Throughout this process, the position updates in the GWO algorithm are illustrated in Figure 2.

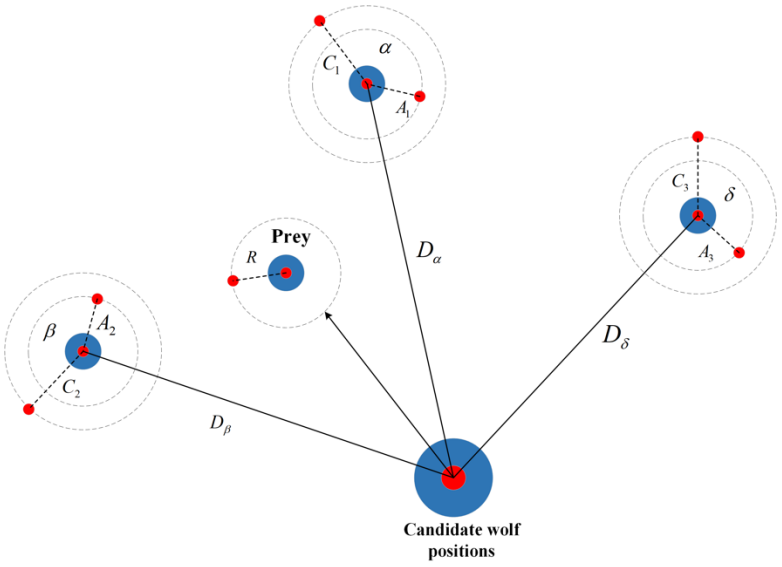

Figure 2. The position updating process in the GWO algorithm

3.7.2. Optimizing LSTM hyperparameters using Grey Wolf Optimization algorithm

The steps for optimizing the LSTM neural network applying the GWO algorithm are as follows:

Step 1: Identify the hyperparameters of the LSTM that need optimization by the GWO algorithm. These primarily include the amount of neurons in the hidden layer, the learning rate of the LSTM network, and the amount of iterations for the LSTM network.

Step 2: Initialize the four initial parameters of the GWO algorithm. The search space includes the three hyperparameters identified in Step 1 that need optimization by the GWO algorithm. In the initialization

parameters, the current population is $x = \{x_{1j}, x_{2j}, \cdots, x_{Nj},\}^T$, where the position of each grey wolf in the space is x_i, and j represents the dimension number where the grey wolf is located.

Step 3: Assign the position coordinates of each grey wolf in the space to the LSTM neural network hyperparameters that need optimization by the GWO algorithm and estimate the fitness value of the position coordinates in the space according to formula (39).

$$f = \sum_{i=1}^{n} \left(y_i^{'} - y_i\right)^2 \tag{33}$$

Where y_i is the expected value; $y_i^{'}$ is the expected value of the long short-run memory neural network.

Step 4: Select the three fitness values that meet the requirements and the corresponding positions of the grey wolves. Assign these values to $x_\alpha(t)$, $x_\beta(t)$ and $x_\delta(t)$. Update the position vectors of these three values using formulas (32) to (38), ensuring that these three values always have the minimum fitness. Check if the number of iterations meets the condition; if it does, stop updating. Otherwise, continue the optimization process.

Step 5: After completing the iterations, assign the optimized position vector $x_\alpha(t)$ to the amount of neurons in the hidden layer, the learning rate of the LSTM neural network, and the amount of iterations of the LSTM neural network. Next, train and evaluate the LSTM network using these coefficients.

Then, determine whether the algorithm converges. If the convergence condition is met, stop, otherwise continue to iterate. The LSTM network optimized by GWO has better hyperparameter configuration. Algorithm 1 show the analysis step.

Algorithm 1 Stacked GWO-ARIMA-LSTM for VIX Forecasting

Require: VIX time series D,
ARIMA parameters (p, d, q),
GWO parameters (population size N, max iteration $MaxIter$),
LSTM hyperparameter ranges (hidden neurons h, learning rate lr, epochs epo).

Train ARIMA on the historical VIX series to capture trend & seasonality
$\hat{y}_{ARIMA} \leftarrow ARIMA.predict(D)$
Form new dataset $D_{stack} \leftarrow (D,\ \hat{y}_{ARIMA})$ { ARIMA }
Initialize GWO population of size N
for $i = 1$ to N **do**
  Randomly assign each wolf's position (h_i, lr_i, epo_i)
end for
$t \leftarrow 0$
while $t < MaxIter$ **do**
  for $i = 1$ to N **do**
    Build an LSTM with (h_i, lr_i, epo_i)
    Train on D_{stack} {VIX+ARIMA}
    $\hat{y}_{LSTM} \leftarrow LSTM.predict(D_{stack})$
    Compute fitness $f_i \leftarrow MSE(D,\ \hat{y}_{LSTM})$
  end for
  Identify top three wolves α, β, δ by lowest f_i
  Update positions of all wolves using GWO update rules
  $t \leftarrow t + 1$
end while
$(h_{best}, lr_{best}, epo_{best}) \leftarrow \alpha$'s final position
Train final LSTM with best hyperparameters $(h_{best}, lr_{best}, epo_{best})$ on D_{stack}
$\hat{y}_{final} \leftarrow LSTM.predict(D_{stack})$
Output: $\hat{y}_{final}$ as the final VIX forecast

The computational complexity of HyperVIX is determined by its three components: ARIMA, LSTM, and GWO. Below, we analyze each component's complexity and discuss the model's scalability:

(a) **ARIMA**: The complexity is $O(T \times (p + q))$, where T is the time series length (≈6,000 in our dataset), and p, q are the orders of the autoregressive and moving average components (p = 2, q = 2). This results in a lightweight computational load.

(b) **LSTM**: The complexity is $O(N \times H^2 \times T)$, where N is the number of LSTM layers (2), H is the number of hidden units (32), and T is the sequence length (21 time steps). With these parameters (Table 9), the LSTM component is computationally efficient.

(c) **GWO**: The complexity is $O(N_w \times T_{iter} \times D)$, where N_w is the number of wolves (20), T_{iter} is the number of iterations (10), and D is the dimensionality of the hyperparameter space (3). As shown in Table 7, the baseline configuration achieves optimal performance with minimal iterations, ensuring efficiency.

4. Data and performance metrices

4.1. Data description

The CBOE VIX index estimates the S&P 500 index's expectation of equity market volatility through the next 30 days. The VIX index not only reflects investors' expectations of future market uncertainty but also serves as an important barometer of market sentiment changes (Whaley, 2009). As a key financial market indicator, the VIX index holds significant importance in both academic research and practical applications.

The VIX index exhibits complex statistical characteristics, reflecting the intrinsic dynamics of financial markets and the complexity of investor behavior. First, the VIX index displays significant nonlinear characteristics. This nonlinearity mainly arises from dramatic fluctuations in market sentiment and the impact of external economic and political events, making the VIX index difficult to accurately capture using simple linear models. Second, the probability distribution of the VIX index typically shows a leptokurtic feature. This means that extreme values occur more frequently and with greater magnitude compared to a normal distribution, highlighting the importance of "black swan" events in financial markets.

In addition, the VIX index exhibits a clear volatility clustering effect. High volatility periods and low volatility periods tend to occur together, reflecting the persistence of market sentiment and the cumulative effect of financial risk. Lastly, there is a significant negative connection between the VIX index and equity market indices. When the stock market declines, the VIX index often rises, and vice versa. This inverse relationship underscores the importance of the VIX index as an indicator of market risk aversion, providing important references for portfolio management and risk hedging. In Figure 3, we present the changes of the VIX index from January 1, 2000, to July 16, 2024 of our dataset, and calculate the 20-day and 50-day moving averages to better observe the trends.

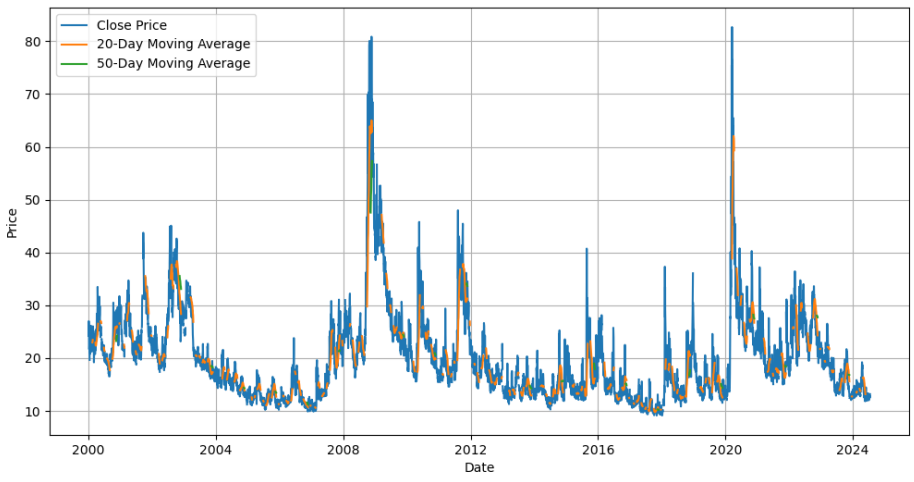

Figure 3. Closing price changes of VIX

We first presented the summary statistics of the VIX index within the sample period in Table 1. We observed significant volatility in the VIX during the sample period, as evidenced by the characteristics and differences between the Max, Min, and Mean values. Additionally, the Jarque-Bera test indicated that the time series of the VIX did not follow a normal distribution. Lastly, the VIX time series passed the ADF test, confirming its suitability for subsequent time series analysis.

Table 1. Summary statistics for the VIX

	Min	Max	Mean	Std.	Jarque-Beta	ADF
VIX	0.001	0.9988	0.1464	0.1159	21155.4193***	-5.8526***

Notes: The ADF test was applied to assess the stationarity. *** indicates significance at the 99% statistical level.

Before formally starting, we normalized the closing price data of the VIX to ensure it is stationary for time series analysis. Subsequently, the dataset was divided into training and testing subsets using a 7:3, 9:1 and 6:4 ratio, which was a widely adopted methodology (Ładyżyński et al, 2018).

The selection of training-testing ratios (7:3, 9:1, and 6:4) was guided by both theoretical and empirical considerations to ensure robust evaluation of the HyperVIX model. The 7:3 ratio is a standard practice in time series forecasting, particularly for financial data, as it provides a balanced split that allows sufficient training data to model complex patterns while reserving a substantial test set for performance evaluation (Ładyżyński et al., 2013). The 9:1 ratio leverages the large VIX dataset (January 1, 2000, to July 16, 2024, approximately 6,000 data points) to maximize training data, which is critical for capturing long-term trends and volatility clustering inherent in financial time series (Hyndman and Athanasopoulos, 2018). The 6:4 ratio increases the test set proportion to rigorously assess the model's generalization ability, particularly in high-volatility scenarios, such as those observed during the COVID-19 crisis and the 2022 Russia-Ukraine war.

4.2 Performance evaluation indicators

To evaluate the performance of the structure, we used five commonly applied performance metrics in time series forecasting literature: RMSE, MAE, MAPE and R^2. These metrics are typically used to compare the accuracy of forecasts (Domeisenet al., 2023; Coqueret and Deguest 2024).

(1) RMSE

Root Mean Square Error (RMSE) is a standard statistical estimate of the difference among forecast and actual values. It is obtained by calculating the square root of the mean of the squared forecast errors. Specifically, it is expressed as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{34}$$

It squares the errors, which amplifies larger errors and highlights their impact.

(2) MAE

Mean Absolute Error (MAE) is another method for measuring the errors of a forecast model. It calculates the average of the absolute amounts of all forecast errors. It is expressed as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{35}$$

MAE does not square large errors and is less affected by outliers.

(3) MAPE

Mean Absolute Percentage Error (MAPE) is a relative error metric used to estimate the accuracy of forecasts, expressed as the percentage of forecast errors relative to actual values. It is represented as:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{36}$$

(4) R-squared (R^2)

The coefficient of determination (R^2) was a measure that evaluated the extent to which a forecasting model fitted the data. The R^2 value was calculated to estimate what percentage of the fluctuation in the outcome variable could be explained by the predictor variables. The following equation was used to calculate it:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{37}$$

This indicator intuitively represents the model's ability to explain data variation. However, it cannot be used alone to evaluate the model and needs to be combined with other indicators. Therefore, this paper combines five indicators to evaluate the superiority of our model.

5. Results Analysis

5.1. Parameter setting

We take the daily closing VIX index from January 1, 2000, to July 16, 2024, as the input for the HyperVIX neural network. First, for ARIMA, when setting the hidden layer, we need to consider the impact of the amount of layers on the LSTM neural network. More layers improve learning accuracy and reduce network errors, but they also complicate the model structure and lengthen training time. Traditional neural networks use a simple structure design with one input layer, one output layer, and one hidden layer.

The hidden layer units, learning rate, and iteration count for the LSTM are mapped as the position coordinates of the wolves. The hidden layer units are set within the range of 2 to 100, the learning rate spans from 0.0001 to 0.01, and the iteration count ranges from 20 to 300. The forecast time step is established at 30

minutes. The parameters for the Grey Wolf Optimization algorithm are configured as follows: there are 20 wolves in total, the maximum number of iterations is 10, and the initial coordinates for the alpha, beta, and delta wolves are all set to [0, 0, 0].

Employing GWO to train the LSTM model, the optimal parameters identified are as follows: the amount of hidden layer units is 32, the learning rate is 0.001, and the amount of iterations is 92.

To address overfitting, following Michańków et al. (2024), Wu (2025) and Zeng et al. (2025), we used early stopping, dropout, and L2 regularization, to ensure robust performance. The main parameters are shown in Table 2.

Table 2. Main Parameters

Model	Main Parameters	Values
HyperVIX	LSTM Hidden Units	32
	GWO Population Size	20
	GWO Iterations	10
	Learning Rate	0.001
Transformer	Attention Heads	8
	Layers	4
	Learning Rate	0.0001
	Epochs	100
iTransformer	Attention Heads	8
	Layers	4
	Learning Rate	0.0001
	Epochs	150
Informer	Attention Heads	8
	Encoder/Decoder Layers	4/2
	Learning Rate	0.0001
	Epochs	200

Notes: All models remain consistent in the main settings of this article.

5.2. Analysis of static forecast results

By optimizing the LSTM neural network using GWO and selecting the best individual in the population based on fitness as the hyperparameters for the LSTM network, we get the findings of the HyperVIX function on the VIX dataset.

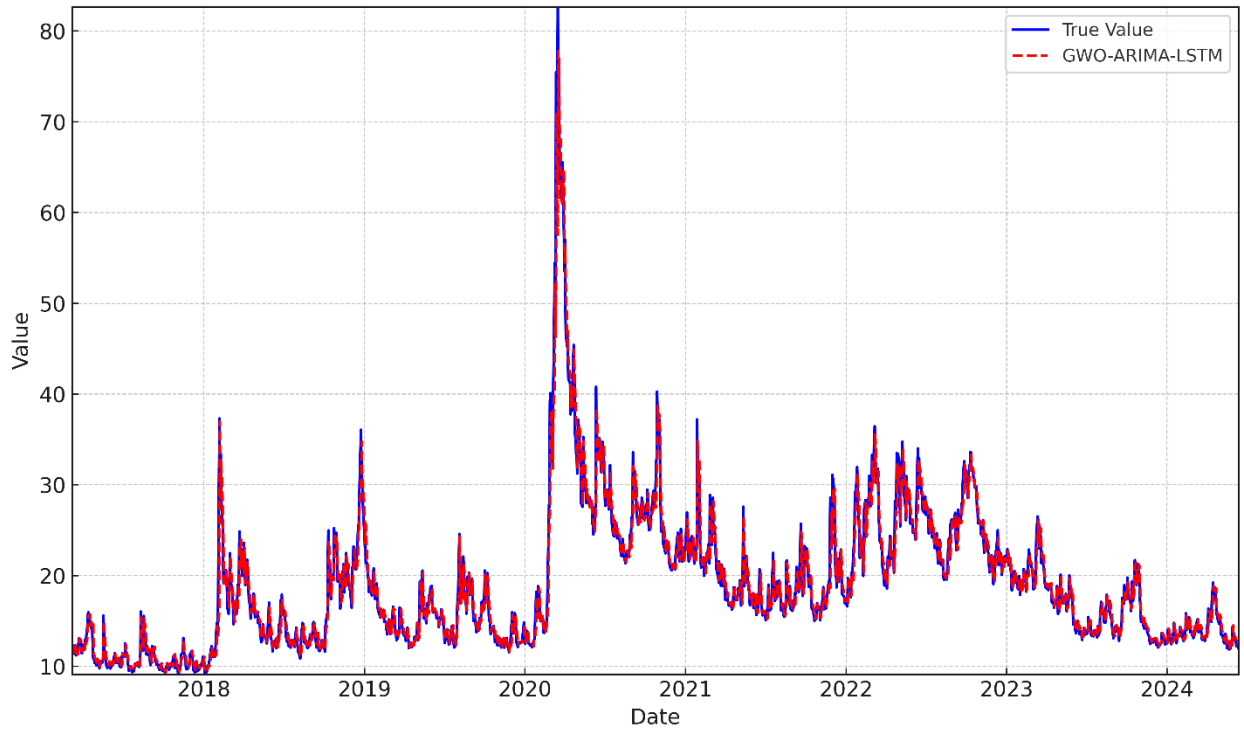

Figure 4. HyperVIX forecast fitting curve

From Figure 4, the HyperVIX (i.e., GWO-ARIMA-LSTM) model's forecasts closely follow the actual values, especially during the significant market fluctuations in 2020, where the model successfully captured the substantial rise and fall of the VIX index. This indicates that the function has good adaptability in handling extreme market volatility. For most periods, the forecasting values closely match the actual values, demonstrating the function's high forecast accuracy. However, at some peak moments, the predicted values slightly lag or deviate, possibly due to the model's lag in handling extreme volatility. To better observe the model's characteristics, we also analyzed the error distribution, as provided in the Figure 5.

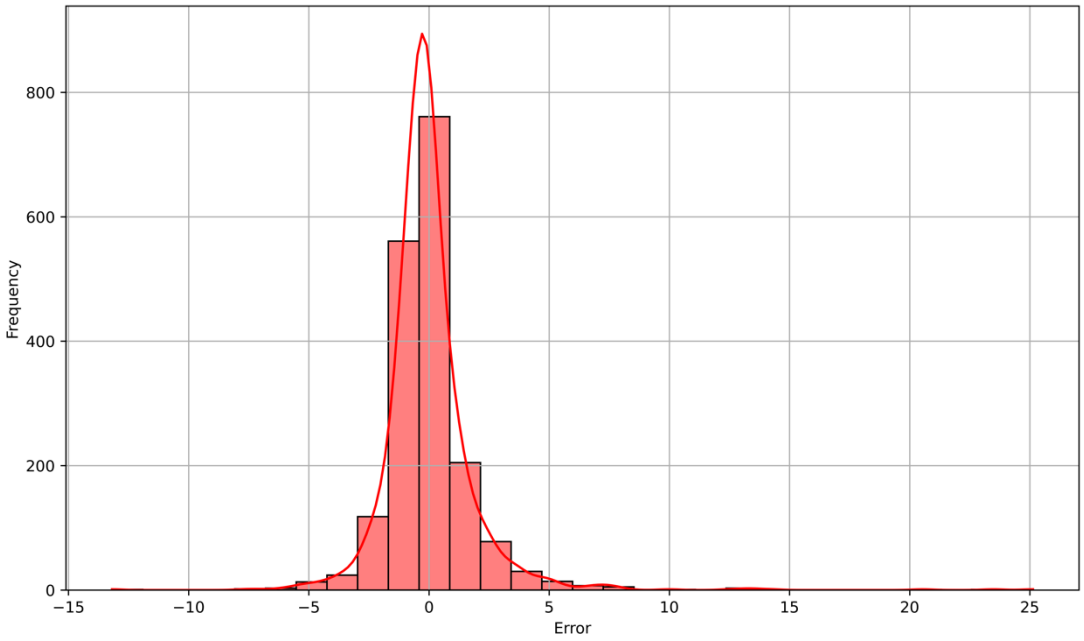

Figure 5. Error distribution of HyperVIX forecasts

The error distribution chart shows that most errors are concentrated around 0. This indicates that the model can predict the VIX index well in most cases, with small errors. Additionally, the error distribution is approximately symmetric, with a balanced distribution on both sides. This denotes that the function does not have a significant bias in the positive or negative direction, meaning there is no systematic bias in overestimating or underestimating forecasts. The high kurtosis of the error distribution suggests that most errors are concentrated near the mean, but there is also a certain degree of fat-tailedness, indicating occasional large errors. The high kurtosis implies that most forecasts are very close to the actual values, while the fat tails might result from extreme market volatility.

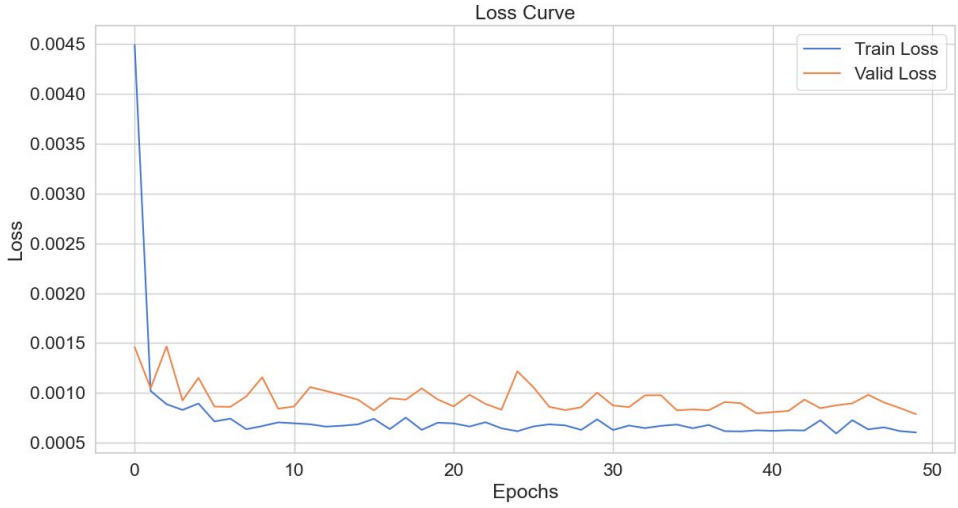

Figure 6. Loss curve

Additionally, as observed in Figure 6, the loss curve of the model shows a sharp decline initially and then stabilizes as the amount of epochs increases. The training loss (blue curve) consistently decreases throughout the training process and stabilizes later, indicating that the function has good fitting ability on the training data. The validation loss (orange curve) also rapidly decreases in the first few epochs and then maintains a low level with slight fluctuations. The stability of the validation loss indicates that the model also has good generalization ability on the validation data. The close proximity of the training loss and validation loss in the later stages, with no significant deviation, suggests that the model is well-trained without significant overfitting or underfitting issues.

Next, to validate the superiority of our proposed function, we compare it with different forecast models. We selected current mainstream and classic forecast models, especially those used in the financial forecasting field. The training parameters for all models are kept consistent, following the settings mentioned above.

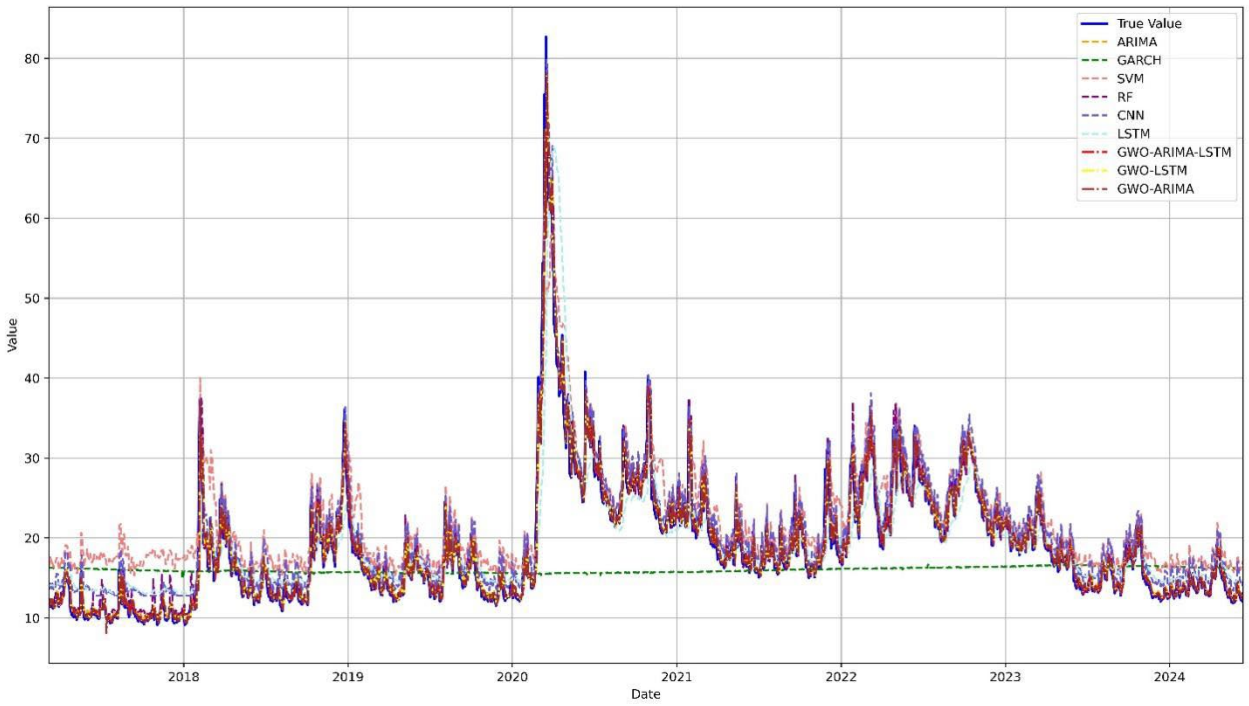

Figure 7. Comparison of the actual value and predicted value of each model

Figure 7 shows that from 2017 to 2019, the data exhibited relatively stable fluctuations, and all models performed similarly during this period. Simple models, such as ARIMA and RF, were able to effectively fit the overall trend, while deep learning models tended to slightly overfit when capturing local fluctuations, resulting in some unnecessary oscillations. In contrast, GWO-optimized models (such as HyperVIX, i.e., GWO-ARIMA-LSTM) performed steadily during this phase, successfully fitting the global trend while avoiding extraneous fluctuations, thus maintaining high accuracy.

From 2020 to 2021, the data experienced significant spike fluctuations, posing a major challenge to the models' ability to capture complex dynamic changes. During this phase, HyperVIX demonstrated a clear advantage, closely following the variations of the true values, including the height and position of the spikes. While LSTM and CNN also captured the overall trend relatively well, they exhibited delays or overfitting in certain spike regions. In contrast, ARIMA and GARCH performed poorly during this period; the former produced overly smooth predictions that failed to reflect the dramatic fluctuations in the data, while the latter struggled to fit the overall trend. This phase indicates that deep learning models, particularly those combined with optimization algorithms, are better suited for handling complex nonlinear dynamics.

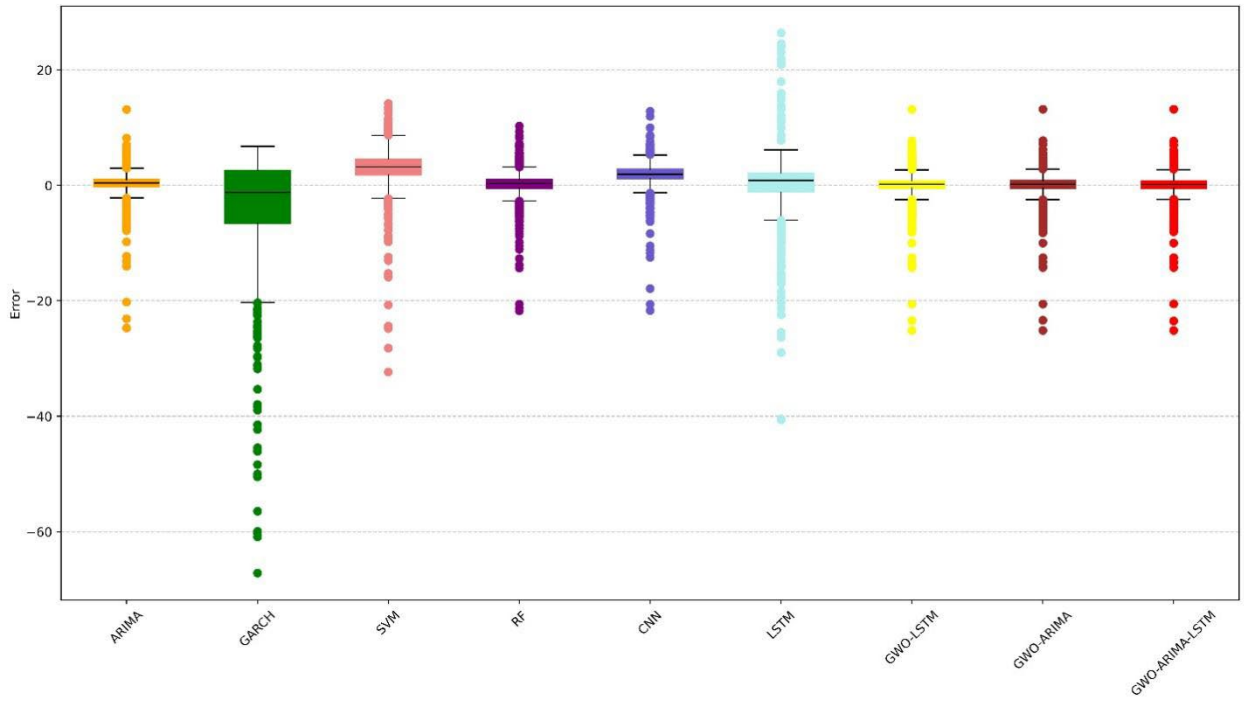

Figure 8. Error of the actual value and forecasting value of each function

Figure 8 highlights the error distributions of various models, with HyperVIX (i.e., GWO-ARIMA-LSTM) standing out as the most accurate and robust among them. Non-optimized models, such as ARIMA and RF, demonstrate relatively stable performance with small error ranges and medians close to zero, making them suitable for capturing overall trends. However, models like GARCH exhibit a wide error spread and a systematic bias, indicating poor performance in handling the data. Deep learning models improve upon traditional models by better capturing complex patterns, but their wider error ranges and occasional extreme outliers reveal limitations in handling more dynamic or extreme variations.

The GWO-optimized models, particularly HyperVIX, outperform all others, exhibiting the smallest interquartile range and a median error almost exactly at zero. This demonstrates its ability to consistently

minimize prediction errors across various scenarios. While GWO-LSTM and GWO-ARIMA also show significant improvements over their standalone versions, HyperVIX sets the benchmark by combining ARIMA and LSTM with GWO optimization, effectively balancing trend fitting and dynamic pattern capturing. This positions it as the most reliable model for accurate and consistent predictions.

Table 3 shows the results of the OLS regression estimation, conducted to evaluate the forecasting accuracy, statistical significance, and reliability of various models based on their regression coefficients (β), *P*-value, and R^2 values. The regression coefficient (β) measures the strength and direction of the relationship between the forecasting and actual values, with coefficients closer to 1 indicating higher alignment. The *P*-value assess the significance of these relationships, and the R^2 values indicate the proportion of variance explained by the models. Additionally, the OLS regression incorporates loss functions such as the MSE and RMSE to quantify model errors, further supporting the evaluation of forecasting performance. Confidence intervals around the regression coefficients were also considered to validate the stability and robustness of the results across repeated samples.

Table 3. OLS regression test of prediction

Forecasting Models	$\boldsymbol{\beta}$	**P-value**	$\boldsymbol{R^2}$
RF	0.9909***	0.0001***	0.9347
CNN	0.9897***	0.0001***	0.9374
LSTM	0.9698***	0.0001***	0.7349
ARIMA	0.9900***	0.0001***	0.9406
GARCH	-3.8896***	0.0001***	0.0240
SVM	1.1076***	0.0001***	0.8472
GWO-LSTM	0.9895***	0.0001***	0.9403
GWO-ARIMA	0.9894***	0.0001***	0.9401
HyperVIX	**0.9914*****	**0.0001*****	**0.9407**

Notes: *** after values represent a rejection of the null hypothesis at the 1% significance level.

Among the models, HyperVIX achieves the highest R^2 value (0.9407) and a regression coefficient ($\beta = 0.9914$) closest to 1, showcasing its exceptional forecasting accuracy and consistency. Similarly, GWO-LSTM and GWO-ARIMA exhibit comparable performance, emphasizing the impact of GWO optimization in enhancing prediction capabilities. In contrast, GARCH demonstrates the weakest performance, with a negative regression coefficient and an R^2 value of only 0.0240, indicating poor forecasting alignment and high instability. Traditional models such as ARIMA and machine learning models like RF perform well, with high R^2 values exceeding 0.93. Deep learning models show moderate performance, with lower R^2 values compared to GWO-optimized models.

The statistical significance of all coefficients is underscored by $P < 0.01$ (***), highlighting the rejection of the null hypothesis at the 1% level. These findings, combined with low loss function values and stable confidence intervals for top-performing models, confirm the reliability of the regression results. The analysis further validates the superior forecasting performance of the HyperVIX framework and its ability to show the variance in the data, making it the most robust model among those tested.

The Diebold-Mariano (DM) test is a widely used statistical method for comparing the prediction errors of two models to determine whether the differences in their errors are statistically significant (Diebold & Mariano, 2002). By analyzing the error sequences, the DM test evaluates whether one model outperforms another or a benchmark model. The core of the DM test lies in constructing a null hypothesis that assumes no significant difference in the prediction errors between the two functions.

Table 4. DM test results

Forecasting Models	DM test	***P*-value**
RF	-8.4184	0.0001***
CNN	-16.1205	0.0001***
LSTM	-9.1167	0.0001***
ARIMA	-6.9524	0.0001***
GARCH	-11.7961	0.0001***
SVM	-19.6759	0.0001***
GWO-LSTM	-6.6613	0.0001***
GWO-ARIMA	-6.6779	0.0001***
HyperVIX	**-6.6519**	**0.0001*****

Notes: *** after values represent a rejection of the null hypothesis at the 1% significance level. The DM statistic and its corresponding *p*-value are used to evaluate this hypothesis. If the *p*-value is below a given significance level, the null hypothesis is rejected, denoting that the forecasting performance of the frameworks is significantly different.

Table 4 presents the DM test results, showing that all models demonstrate statistically significant differences in prediction errors compared to the benchmark model ($P < 0.05$). Among them, HyperVIX, GWO-LSTM, and GWO-ARIMA perform best, with DM statistics of -6.6519, -6.6613, and -6.6779, respectively, highlighting the effectiveness of GWO optimization in enhancing prediction accuracy. Traditional models, such as ARIMA, perform well with a DM statistic of -6.9524 but still slightly lag behind GWO-optimized models. Conversely, GARCH and SVM show the poorest performance, with DM statistics of -11.7961 and -19.6759, reflecting their unsuitability for this task. Deep learning models perform moderately, with DM statistics of -16.1205 and -9.1167, respectively, but remain less accurate than GWO-optimized models. RF, with a DM statistic of -8.4184, shows stable but inferior performance compared to HyperVIX.

Overall, HyperVIX achieves the lowest prediction errors and the highest robustness, further validating the effectiveness of GWO optimization in improving model accuracy and reliability.

Next, we evaluate the performance of every framework using metrics such as RMSE, MAE, MAPE, and R^2. The findings are shown in the Table 5.

Table 5. Evaluation indicators of different models

Forecasting Models	R^2	MSE	RMSE	MAE	MAPE (%)
RF	0.9341	4.2157	2.0532	1.2550	6.2860
CNN	0.8842	7.4118	2.7225	2.2317	13.0408
LSTM	0.7337	17.0432	4.1283	2.5260	13.2216
ARIMA	0.9396	3.8646	1.9659	1.1864	5.8766
GARCH	-0.1624	74.3865	8.6248	5.6756	26.7570
SVM	0.6822	20.3342	4.5093	3.6795	23.3790
GWO-LSTM	0.9402	3.8271	1.9563	1.1222	5.3859
GWO-ARIMA	0.9400	3.8410	1.9598	1.1241	5.4008
HyperVIX	**0.9403**	**3.8185**	**1.9541**	**1.1217**	**5.3826**

Notes: The best model is in bold.

Table 5 presents the evaluation indicators of different models, showcasing their forecasting performance through metrics. Among the functions, HyperVIX achieves the best overall performance, with the highest R^2 value (0.9403) and the lowest error metrics across all indicators, demonstrating its exceptional ability to capture both global trends and local dynamics. GWO-LSTM and GWO-ARIMA follow closely behind, with slightly higher error values but maintaining similar R^2 scores, highlighting the effectiveness of GWO optimization in enhancing model accuracy.

Traditional models like ARIMA and RF show relatively stable performance, with R2R^2R2 values above 0.93, but their error metrics are higher compared to GWO-optimized models, indicating limited capacity in handling complex data patterns. On the other hand, GARCH performs the worst, with a negative R^2 value and significantly larger errors, showing its unsuitability for this task. Next, we calculate the residuals for each model for comparison in Figure 9,

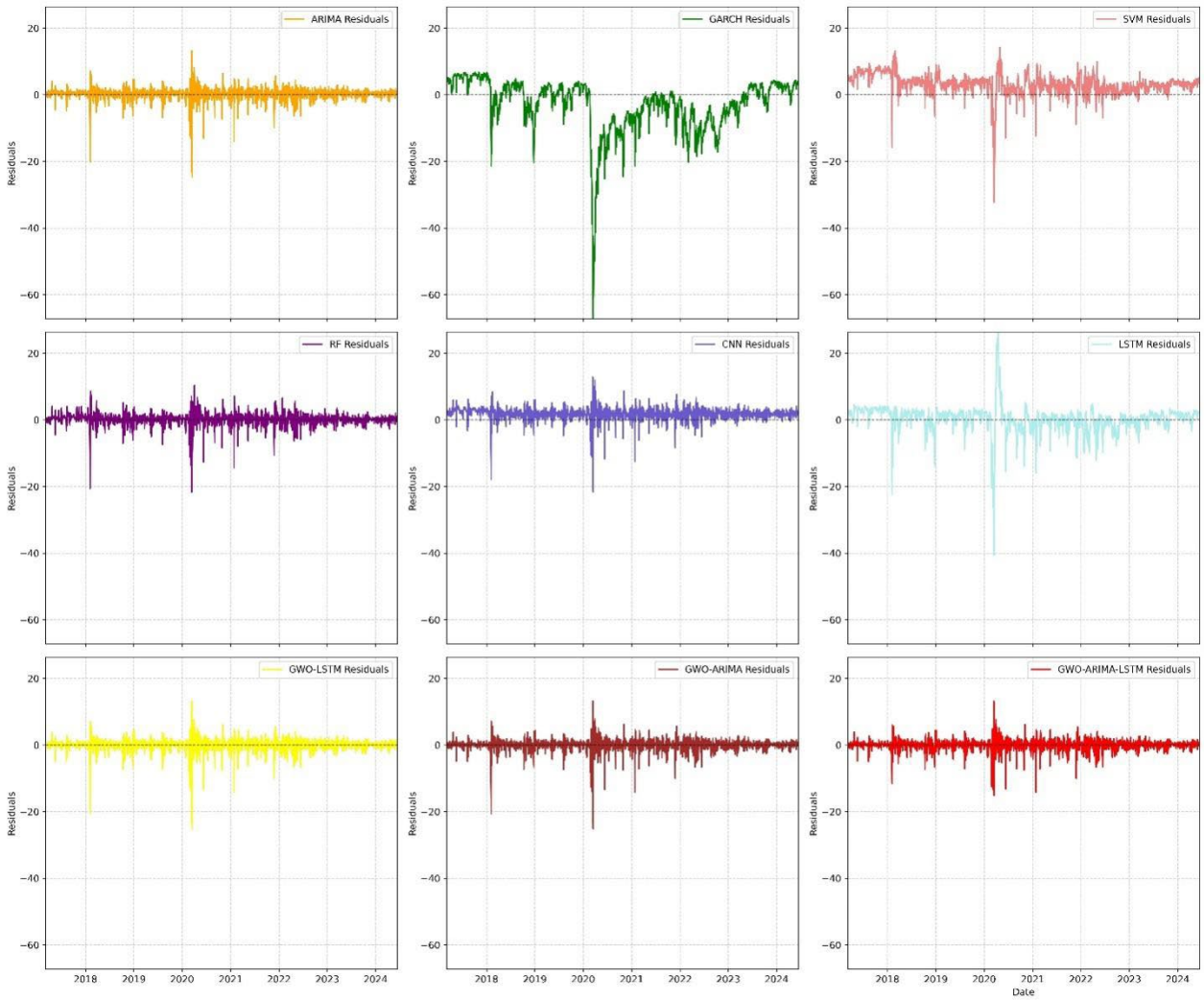

Figure 9. Residuals of the actual value and forecasting value of each framework

We can observe that the residuals in Figure 9 of the HyperVIX (i.e., GWO-ARIMA-LSTM) function are smaller and more concentrated compared to other models, especially during periods of significant market volatility (such as early 2020). This indicates that the HyperVIX model has better robustness and adaptability. During the period of extreme market fluctuations in early 2020, the residuals of the GARCH and SVM models increased significantly, showing poorer performance, while the residuals of the HyperVIX model remained relatively small, indicating more accurate forecasts under extreme volatility conditions.

The residuals of the CNN and RF models showed larger fluctuations, indicating some instability, whereas the residuals of the HyperVIX model fluctuated less, demonstrating more stable forecasts and greater robustness and adaptability. This further confirms the superiority of this hybrid model in complex financial forecasting.

In summary, time series models have particular advantages for forecasting the VIX index. Specifically, parametric models like ARIMA and GARCH excel in handling linear financial data with high accuracy.

However, when faced with more complex data with various characteristics, the limitations of parametric models become apparent. Therefore, we have integrated neural networks into traditional financial forecasting models to address this duality. Additionally, the optimization based on the GWO algorithm has made our HyperVIX structure more competitive, outperforming other models across different evaluation metrics.

Our model not only provides a feasible algorithm for forecasting the VIX index but also offers a direction for optimizing time series financial data forecast models. Furthermore, policymakers can use the trends of related indices to reasonably adjust macroeconomic policies, ensuring the stability of financial markets and the continued prosperity of the economy.

6. Robustness tests

To validate the reliability and applicability of the function, we conducted two robustness tests. The first test examines the model's performance under varying training-to-testing data split ratios. The second test evaluates the model's stability by analyzing its behavior across different sliding window sizes.

6.1. Change the sliding window size

Table 6 presents the robustness test results for different sliding window sizes, highlighting the consistent superiority of the HyperVIX model. Across all window sizes, HyperVIX achieves the highest R^2 values and the lowest error metrics (MSE, RMSE, MAE, MAPE), demonstrating exceptional stability and accuracy. The performance of GWO-LSTM and GWO-ARIMA closely follows, further emphasizing the benefits of GWO optimization. In contrast, non-optimized deep learning models (CNN, LSTM) and traditional econometric models (ARIMA, GARCH) exhibit higher variability and larger errors, particularly with smaller window sizes. While ARIMA and RF show relatively stable performance, their forecasting accuracy remains inferior to GWO-optimized models. These results validate the robustness and adaptability of HyperVIX, particularly in capturing both global trends and local dynamics under varying window configurations.

Table 6. Robustness test of different sliding windows

		HyperVIX	GWO-LSTM	GWO-ARIMA	ARIMA	GARCH	SVM	RF	CNN	LSTM
10	R^2	0.8933	0.8932	0.893	0.8926	-0.1624	0.6471	0.8874	0.839	0.696
	MSE	3.9304	3.938	3.968	3.9936	76.2335	23.4466	4.123	8.07	18.4672
	RMSE	1.9825	1.9845	1.992	1.9974	8.1451	4.8431	2.0214	2.8437	4.2955
	MAE	1.1912	1.1923	1.1954	1.2671	5.6052	4.0344	1.27	2.4739	2.7353
	MAPE (%)	5.1135	5.1166	5.1308	5.5828	25.4191	22.2111	5.9677	12.3888	12.5605
	R^2	0.9360	0.9359	0.9355	0.9350	-0.1624	0.8000	0.9330	0.8790	0.8000

20	MSE	3.6704	3.6740	3.7015	3.7260	73.1450	22.3911	3.9565	7.1800	17.1184
	RMSE	1.9143	1.9166	1.9206	1.9276	8.0314	4.7330	1.9889	2.6776	4.1359
	MAE	1.1220	1.1231	1.1254	1.1885	5.4313	3.7791	1.2231	2.3286	2.5640
	MAPE (%)	5.1135	5.1166	5.1308	5.5828	25.4191	22.2111	5.9677	12.3888	12.5605
30	R^2	0.9390	0.9385	0.9382	0.9375	-0.1624	0.8050	0.9370	0.8840	0.8050
	MSE	3.6320	3.6360	3.6620	3.6900	75.1210	22.3701	3.9575	7.1000	16.8680
	RMSE	1.9060	1.9078	1.9125	1.9183	8.1296	4.7030	1.9885	2.6716	4.1062
	MAE	1.1184	1.1195	1.1227	1.1840	5.4620	3.7520	1.2220	2.3320	2.5400
	MAPE (%)	5.1135	5.1166	5.1308	5.5828	25.4191	22.2111	5.9677	12.3888	12.5605
50	R^2	0.9410	0.9408	0.9405	0.9400	-0.1624	0.8100	0.9390	0.8870	0.8100
	MSE	3.6004	3.6025	3.6270	3.6520	74.0335	22.0863	3.9200	7.0800	16.7556
	RMSE	1.8985	1.8997	1.9054	1.9107	8.0809	4.6901	1.9755	2.6639	4.0944
	MAE	1.1142	1.1154	1.1187	1.1800	5.4532	3.7355	1.2152	2.2905	2.5221
	MAPE (%)	5.1135	5.1166	5.1308	5.5828	25.4191	22.2111	5.9677	12.3888	12.5605

Notes: In the main text, sliding window is 40, and all parameter settings are consistent with the main text.

6.2. Change the data partition ratio

Table 7 presents the forecasting performance of various models under different data partition ratios (9:1, 7:3, and 6:4), reflecting the impact of data distribution on the models' generalization ability and stability. Overall, HyperVIX consistently outperforms other functions across all partition ratios, achieving the highest R^2 values and the lowest error metrics (MSE, RMSE, MAE, and MAPE), demonstrating its robustness under varying data partition conditions. As the proportion of the test set increases (from 9:1 to 6:4), the R^2 values of all models show a declining trend, while error metrics such as MSE and RMSE increase, indicating that less training data reduces the models' fitting ability. In comparison, GWO-LSTM and GWO-ARIMA rank second and third, respectively, with error metrics slightly higher than the primary model but significantly improved compared to non-optimized models.

Traditional models, such as ARIMA and GARCH, exhibit relatively stable performance across different partition ratios. However, their error metrics are significantly higher than those of GWO-optimized models. In particular, GARCH consistently shows negative R^2 values and large errors, indicating its unsuitability for this task. Additionally, non-optimized deep learning functions, such as LSTM, show a significant increase in errors as the proportion of the test set grows, revealing high instability in their predictions. For instance, under the 6:4 partition ratio, LSTM's MSE reaches 17.1120, whereas HyperVIX achieves only 3.9025, highlighting

the importance of optimization. Overall, GWO optimization significantly enhances model robustness, with HyperVIX demonstrating exceptional forecasting performance across all data partition ratios.

Table 7. Robustness test of different data partition ratio

		HyperVIX	GWO-LSTM	GWO-ARIMA	ARIMA	GARCH	SVM	RF	CNN	LSTM
9:1	R^2	0.9430	3.7525	1.9395	1.1112	5.3826	0.9430	3.7525	1.9395	1.1112
	MSE	0.9429	3.7610	1.9412	1.1120	5.3859	0.9429	3.7610	1.9412	1.1120
	RMSE	0.9427	3.7830	1.9440	1.1134	5.4008	0.9427	3.7830	1.9440	1.1134
	MAE	0.9423	3.7980	1.9490	1.1755	5.8766	0.9423	3.7980	1.9490	1.1755
	MAPE(%)	-0.1624	73.9563	8.6095	5.6611	26.7570	-0.1624	73.9563	8.6095	5.6611
7:3	R^2	0.9380	0.9378	0.9375	0.9370	-0.1624	0.7100	0.9365	0.8870	0.7650
	MSE	3.8020	3.8065	3.8250	3.8500	73.4280	19.5620	4.2050	7.2800	16.9025
	RMSE	1.9505	1.9515	1.9560	1.9605	8.5865	4.4750	2.0450	2.7170	4.1020
	MAE	1.1184	1.1195	1.1215	1.1840	5.6310	3.6200	1.2275	2.1980	2.4850
	MAPE(%)	5.3826	5.3859	5.4008	5.8766	26.7570	23.3790	6.2860	13.0408	13.2216
6:4	R^2	0.9320	0.9318	0.9315	0.9310	-0.1624	0.7000	0.9305	0.8830	0.7600
	MSE	3.9025	3.9065	3.9250	3.9500	73.9999	19.8345	4.2650	7.3550	17.1120
	RMSE	1.9761	1.9775	1.9812	1.9861	8.6180	4.5020	2.0560	2.7250	4.1340
	MAE	1.1297	1.1305	1.1324	1.1900	5.6490	3.6305	1.2375	2.2115	2.4980
	MAPE(%)	5.3826	5.3859	5.4008	5.8766	26.7570	23.3790	6.2860	13.0408	13.2216

Notes: In the main text, the ratio of training set to test set is 8:2, and all parameter settings are consistent with the main text.

6.3. Different GWO Parameter Configurations

To further validate the effectiveness of the GWO structure in optimizing the HyperVIX model, we tested its performance under different GWO parameter settings, specifically varying the population size (number of wolves) and the maximum number of iterations. The population sizes tested were 10, 20, 30, and 40 wolves, while the maximum iterations were set to 5, 10, and 20. These configurations were chosen to explore the trade-off between optimization depth and computational efficiency, building on the baseline settings used in the study (20 wolves and 10 iterations).

Table 8. Robustness Test of Different GWO Parameter Configurations

Population Size	Max Iterations	R^2	MSE	RMSE	MAE	MAPE (%)
10	5	0.9305	4.0123	2.0102	1.1823	5.6214
10	10	0.9321	3.9721	1.993	1.1654	5.5823
10	20	0.9325	3.965	1.9912	1.162	5.5765
20	5	0.9368	3.8502	1.9622	1.1405	5.4502
20	10	0.9403	3.8185	1.9541	1.1217	5.3826
20	20	0.9405	3.815	1.9532	1.1208	5.3801

30	5	0.9375	3.84	1.9596	1.1352	5.432
30	10	0.9401	3.8205	1.9546	1.1225	5.385
30	20	0.9404	3.8165	1.9538	1.121	5.3812
40	5	0.938	3.835	1.9582	1.13	5.42
40	10	0.9402	3.819	1.9545	1.122	5.3835
40	20	0.9404	3.816	1.9536	1.1205	5.3805

Table 8 presents the results of this robustness test, reporting the R^2, MSE, RMSE, MAE, and MAPE metrics for each GWO parameter configuration. The findings show that HyperVIX maintains high forecasting accuracy across all tested configurations, with R^2 values consistently above 0.93 and error metrics remaining low. Increasing the population size from 10 to 20 wolves and the iteration count from 5 to 10 generally improves performance, as evidenced by lower RMSE (from 2.0102 to 1.9541) and higher R^2 (from 0.9305 to 0.9403). However, further increases to 30 or 40 wolves and 20 iterations yield diminishing returns, suggesting that the baseline configuration (20 wolves, 10 iterations) achieves an optimal balance between accuracy and computational efficiency.

6.4. Comparation with three Transformer-based models

We compared HyperVIX with three Transformer-based models: Transformer (Vaswani et al., 2017), iTransformer (Liu et al., 2023), and Informer (Zhou et al., 2021). These models were selected for their prominence in time series forecasting, particularly their ability to capture long-term dependencies through attention mechanisms. However, their performance in VIX forecasting, which involves nonlinear dynamics and extreme volatility, may be limited compared to HyperVIX framework.

Table 9. Comparison with Transformer-Based Models

Models	**R^2**	**MSE**	**RMSE**	**MAE**	**MAPE (%)**
Transformer	0.9201	4.3500	2.2114	1.3400	6.2367
iTransformer	0.9221	4.2999	2.1667	1.2845	6.1879
Informer	0.9250	4.2501	2.1523	1.2511	6.1000
HyperVIX	0.9403	3.8185	1.9541	1.1217	5.3826

Notes: The parameter settings are consistent with Chapter 5.

Table 9 presents the performance metrics for HyperVIX, Transformer, iTransformer, and Informer, evaluated on the VIX dataset . The results show that HyperVIX outperforms all Transformer-based models across all

metrics. The superior performance of HyperVIX is particularly evident in its lower MAE and MAPE, indicating higher accuracy in capturing the VIX's nonlinear dynamics.

7. Conclusions

This paper based on the characteristics of the VIX index, proposed an innovative hybrid framework, HyperVIX. Empirical findings show that the HyperVIX model significantly outperforms traditional financial models, including standalone ARIMA, GARCH, SVM, RF, CNN and the main models of the Transformer series: Transformer, iTransformer and Informer, in terms of prediction accuracy and robustness. Particularly during periods of severe market volatility, such as the global outbreak of the COVID-19 crisis and the Russia-Ukraine war in 2022, the model accurately captured the dramatic fluctuations of the VIX index.

The findings of this study have critical implications for regulators, investors, and risk managers. For regulators, more accurate VIX predictions can help identify systemic risks and market vulnerabilities, thereby formulating more targeted regulatory measures. For example, when forecasting that market volatility may rise significantly, regulators can strengthen market supervision and increase liquidity support in advance. For institutional investors, the model can help optimise portfolio management and risk hedging strategies. Specifically, fund managers can dynamically adjust the risk exposure of their portfolios based on the model's predictions, increasing defensive asset allocation when volatility is expected to rise. For corporate risk managers, the volatility predictions provided by the model can help develop more precise risk management strategies, such as adjusting VaR limits or option hedging ratios. At the same time, the model also provides tool support for financial product innovation, such as designing volatility-related structured products or developing new volatility trading strategies.

The methodological innovation of this study provides new ideas for financial time series prediction. By integrating linear and nonlinear modelling techniques and utilising advanced metaheuristic algorithms for optimisation, this method can be extended and applied to the prediction of other financial indicators. For example, this hybrid modelling approach can be used to forecast other market indices, commodity prices, or exchange rate fluctuations. In particular, the application of the GWO algorithm in optimising deep learning model hyperparameters provides an effective way to solve the limitations of traditional grid search and random search methods. For the field of quantitative investment, this method can be integrated into existing trading systems to enhance the forecasting accuracy of strategies. Similar to the study by Ślusarczyk and Ślepaczuk (2025), we believe that the superior performance of the HyperVIX model can provide a certain theoretical

basis for investment portfolios. In addition, the development of related analytical tools can also refer to the model's structure.

However, this study still has some limitations. Future research can explore combining other optimisation techniques with this hybrid modelling approach, such as introducing attention mechanisms or transformer architectures to enhance the model's forecasting capabilities.

References

Abedin, M. Z., Goldstein, M. A., Huang, Q., & Zeng, H. (2024). Forward-looking disclosure effects on stock liquidity in China: Evidence from MD&A text analysis. *International Review of Financial Analysis*, *95*, 103484.

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica, 71*(2), 579-625.

Apergis, N., Mustafa, G., & Malik, S. (2023). The role of the COVID-19 pandemic in US market volatility: Evidence from the VIX index. *The Quarterly Review of Economics and Finance, 89*, 27-35.

Bates, D. S. (2000). Post-'87 crash fears in the S&P 500 futures option market. *Journal of Econometrics, 94*(1-2), 181-238.

Becker, R., Clements, A. E., & McClelland, A. (2009). The jump component of S&P 500 volatility and the VIX index. *Journal of Banking & Finance, 33*(6), 1033-1038.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152).

Box, G. E. P., & Jenkins, G. M. (1976). Time series analysis: Forecasting and control. Holden-Day.

Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

Campbell, J. Y. (2007). Estimating the equity premium. *Canadian Journal of Economics, 40*(1), 1-21.

Cheng, J., Tiwari, S., Khaled, D., Mahendru, M., & Shahzad, U. (2024). Forecasting Bitcoin prices using artificial intelligence: Combination of ML, SARIMA, and Facebook Prophet models. *Technological Forecasting and Social Change*, *198*, 122938.

Christoffersen, P. F., & Diebold, F. X. (2000). How relevant is volatility forecasting for financial risk management? *Review of Economics and Statistics, 82*(1), 12-22.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Coqueret, G., & Deguest, R. (2024). Unexpected opportunities in misspecified forecasting regressions. *European Journal of Operational Research.*

Diebold, F. X., & Mariano, R. S. (2002). Comparing forecasting accuracy. *Journal of Business & Economic Statistics*, *20*(1), 134-144.

Ding, S., Ye, J., & Cai, Z. (2024). Multi-step carbon emissions forecasting using an interpretable framework of new data preprocessing techniques and improved grey multivariable convolution model. *Technological Forecasting and Social Change*, *208*, 123720.

Domeisen, D. I., Eltahir, E. A., Fischer, E. M., Knutti, R., Perkins-Kirkpatrick, S. E., Schär, C., ... & Wernli, H. (2023). Prediction and projection of heatwaves. *Nature Reviews Earth & Environment, 4*(1), 36-50.

Enders, W. (2008). Applied econometric time series. John Wiley & Sons.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654-669.

Ghosh, I., & Jana, R. K. (2023). A granular machine learning framework for forecasting high-frequency financial market variables during the recent black swan event. *Technological Forecasting and Social Change*, *194*, 122719.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies, 33*(5), 2223-2273.

Hansen, P. R., Huang, Z., Tong, C., & Wang, T. (2024). Realized GARCH, CBOE VIX, and the volatility risk premium. *Journal of Financial Econometrics, 22*(1), 187-223.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting, 22*(4), 679-688.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Jing, N., Wu, Z., & Wang, H. (2021). A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. *Expert Systems with Applications, 178*, 115019.

Kashif, K., & Ślepaczuk, R. (2025). LSTM-ARIMA as a hybrid approach in algorithmic investment strategies. *Knowledge-Based Systems*, 113563.

Kristjanpoller, W., & Minutolo, M. C. (2015). Gold price volatility: A forecasting approach using the Artificial Neural Network–GARCH model. *Expert Systems with Applications, 42*(20), 7245-7251.

Ładyżyński, P., Żbikowski, K., & Grzegorzewski, P. (2013). Stock trading with random forests, trend detection tests and force index volume indicators. In Artificial Intelligence and Soft Computing: 12th International Conference, ICAISC 2013, Zakopane, Poland, June 9-13, 2013, Proceedings, Part II 12 (pp. 441-452). Springer Berlin Heidelberg.

Li, L., Kang, Y., & Li, F. (2023). Bayesian forecast combination using time-varying features. *International Journal of Forecasting*, *39*(3), 1287-1302.

Lin, Y., Yan, Y., Xu, J., Liao, Y., & Ma, F. (2021). Forecasting stock index price using the CEEMDAN-LSTM model. *The North American Journal of Economics and Finance, 57*, 101421.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., & Long, M. (2023). itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Lu, F., Ma, F., & Feng, L. (2024). Carbon dioxide emissions and economic growth: New evidence from GDP forecasting. *Technological Forecasting and Social Change*, *205*, 123464.

Lundbergh, S., & Teräsvirta, T. (2002). Evaluating GARCH models. *Journal of Econometrics, 110*(2), 417-435.

Lu, R., & Zeng, H. (2023). VIX and major agricultural future markets: dynamic linkage and time-frequency relations around the COVID-19 outbreak. *Studies in Economics and Finance*, *40*(2), 334-353.

Michańków, J., Sakowski, P., & Ślepaczuk, R. (2024). Mean absolute directional loss as a new loss function for machine learning problems in algorithmic investment strategies. *Journal of Computational Science*, *81*, 102375.

McAleer, M., & Medeiros, M. C. (2008). A multiple regime smooth transition heterogeneous autoregressive model for long memory and asymmetries. *Journal of Econometrics, 147*(1), 104-119.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46-61.

Pai, P. F., & Lin, C. S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega, 33*(6), 497-505.

Pan, Z., Wang, Y., Liu, L., & Wang, Q. (2019). Improving volatility prediction and option valuation using VIX information: A volatility spillover GARCH model. *Journal of Futures Markets*, *39*(6), 744-776.

Pinto, J. M., Neves, R. F., & Horta, N. (2015). Boosting Trading Strategies performance using VIX indicator together with a dual-objective Evolutionary Computation optimizer. *Expert Systems with Applications, 42*(19), 6699-6716.

Pokou, F., Sadefo Kamdem, J., & Benhmad, F. (2024). Hybridization of ARIMA with learning models for forecasting of stock market time series. *Computational Economics, 63*(4), 1349-1399.

Poon, S. H., & Granger, C. W. (2003). Forecasting volatility in financial markets: A review. *Journal of Economic Literature, 41*(2), 478-539.

Qiao, G., Cui, W., Zhou, Y., & Liang, C. (2025). Volatility of Volatility and VIX Forecasting: New Evidence Based on Jumps, the Short-Term and Long-Term Volatility. *Journal of Futures Markets*, *45*(1), 23-46.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. (2017). A dual-stage attention-based recurrent neural network for time series prediction. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (pp. 2627-2633).

Qin, Z., Yang, S., & Zhong, Y. (2024). Hierarchically gated recurrent neural network for sequence modeling. *Advances in Neural Information Processing Systems, 36*.

Roszyk, N., & Ślepaczuk, R. (2024). The Hybrid Forecast of S&P 500 Volatility ensembled from VIX, GARCH and LSTM models. *arXiv preprint arXiv:2407.16780*.

Rossi, B. (2021). Forecasting in the presence of instabilities: How we know whether models predict well and how to improve them. *Journal of Economic Literature*, *59*(4), 1135-1190.

Saâdaoui, F., & Rabbouch, H. (2024). Financial forecasting improvement with LSTM-ARFIMA hybrid models and non-Gaussian distributions. *Technological Forecasting and Social Change*, *206*, 123539.

Ślusarczyk, D., & Ślepaczuk, R. (2025). Optimal Markowitz portfolio using returns forecasted with time series and machine learning models. *Journal of Big Data*, *12*(1), 127.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing, 90*, 106181.

Singh, S., Parmar, K. S., Kumar, J., & Makkhan, S. J. S. (2020). Development of new hybrid model of discrete wavelet decomposition and autoregressive integrated moving average (ARIMA) models in application to one month forecast the casualties cases of COVID-19. *Chaos, Solitons & Fractals, 135*, 109866.

Stempień, D., & Ślepaczuk, R. (2025). Hybrid Models for Financial Forecasting: Combining Econometric, Machine Learning, and Deep Learning Models. *arXiv preprint arXiv:2505.19617*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Tumala, M. M., Salisu, A., & Nmadu, Y. B. (2023). Climate change and fossil fuel prices: A GARCH-MIDAS analysis. *Energy Economics, 124*, 106792.

Wang, C., Chen, Y., Zhang, S., & Zhang, Q. (2022). Stock market index prediction using deep Transformer model. *Expert Systems with Applications, 208*, 118128.

Whaley, R. E. (2000). The investor fear gauge. *The Journal of Portfolio Management, 26*(3), 12-17.

Whaley, R. E. (2009). Understanding the VIX. *Journal of Portfolio Management, 35*(3), 98-105.

Wu, R. (2025). Forecasting the European Union allowance price tail risk with the integrated deep belief and mixture density networks. *Chaos, Solitons & Fractals*, *199*, 116786.

Wu, R., Zeng, H., Abedin, M. Z., & Ahmed, A. D. (2025). The impact of extreme climate on tourism sector international stock markets: A quantile and time-frequency perspective. *Tourism Economics*, 13548166241311633.

Yu, B., Li, C., Mirza, N., & Umar, M. (2022). Forecasting credit ratings of decarbonized firms: Comparative assessment of machine learning models. *Technological Forecasting and Social Change, 174*, 121255.

Zeng, H., Abedin, M. Z., & Upreti, V. (2024). Does climate risk as barometers for specific clean energy indices? Insights from quartiles and time-frequency perspective. *Energy Economics*, *140*, 108003.

Zeng, H., Wu, R., Abedin, M. Z., & Ahmed, A. D. (2025). Forecasting Volatility of Australian Stock Market Applying WTC-DCA-Informer Framework. *Journal of Forecasting*.

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159-175.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021, May). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 12, pp. 11106-11115).

Zolfaghari, M., & Gholami, S. (2021). A hybrid approach of adaptive wavelet transform, long short-term memory and ARIMA-GARCH family models for the stock index prediction. *Expert Systems with Applications, 182*, 115149.