IEEE Access
Multidisciplinary Rapid Review Open Access Journal

Received 13 August 2024, accepted 28 August 2024, date of publication 30 August 2024, date of current version 10 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3452426

RESEARCH ARTICLE

Stock Index Forecasting Using a Novel Integrated Model Based on CEEMDAN and TCN-GRU-CBAM

SIBO LI, GUOQIANG TANG, XUCHANG CHEN, AND TONGZHI LIN
School of Mathematics and Statistics, Guilin University of Technology, Guilin 541006, China

Corresponding author: Guoqiang Tang (tanggq@glut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62166015.

ABSTRACT In the context of the current rapid development of the financial market, how to establish an effective stock price index model to avoid investment risks and enhance investment returns for investors has become a subject of great concern. On this basis, a new method of stock price index prediction employing deep neural networks is investigated. This paper employs the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) technique to break down the time series of the stock index into its constituent Intrinsic Modal Functions (IMFs). According to the similarity of the values of Fuzzy Entropy (FE), the subsequence is reorganized to become a new sequence, which highlights the fluctuation state of the stock index at different frequencies and improves the forecasting efficiency. In terms of a forecasting method, the combination of Temporal Convolutional Network (TCN), Gated Recurrent Unit (GRU), and Convolutional Block Attention Module (CBAM) is used to forecast the reorganized subsequence, and the final forecast results are reconstructed to obtain the final prediction value. To further evaluate the performance of the proposed CEEMDAN-TCN-GRU-CBAM model, this paper selects four representative stock indices in emerging and developed markets while comparing them with the benchmark model. It uses four evaluation metrics to measure the model performance. The study shows that the proposed model outperforms other benchmark models has better robustness and universality, and has higher forecasting accuracy.

INDEX TERMS Stock index forecasting, sequence decomposition, deep learning, combinatorial models.

I. INTRODUCTION

The stock market plays a vital role within the financial market as the main channel for corporate finance and one of the key indicators for investors to make investments. Stock index forecasting can help investors develop smarter investment strategies, including buying stocks, holding stocks, selling stocks, futures, and other financial assets. Investors can better manage risk and capture returns by forecasting market trends and price movements. Stock index forecasting is likewise an important topic of research in finance, economics, and other disciplines, and the study of stock index forecasting models and methods can contribute to the academic community's in-depth understanding of the behavior and laws of financial markets. However, stock index data, as time series, are characterized by high noise, dynamics, and nonlinearity, so it is necessary to find a reasonable stock index feature extraction method and construct a nonlinear forecasting model that can describe the complex stock market, to further reveal the internal operation law of the stock market, better play the function of the stock market, and prevent and resolve the financial risks promptly. Nowadays, stock index series forecasting techniques are classified into three categories: conventional time series approaches, artificial intelligence algorithms, and hybrid forecasting methodologies.

Traditional time series methods have a more complete mathematical and theoretical foundation, and the Autoregressive Integrated Moving Average (ARIMA [1]) model, the Generalized Autoregressive Conditional Heteroskedasticity (GARCH [2]) model, the Grey Markov (GM [3]) model, etc. can rigorously test the constructed models and parameters and assess the model's goodness-of-fit. However, these models usually have strict assumptions on the data, such as having conditions such as stability. Stock price indices have

The associate editor coordinating the review of this manuscript and approving it for publication was Jiajie Fan.

characteristics such as non-stationarity and non-linearity, which make it difficult for traditional time series methods to model them effectively, and therefore it is difficult to achieve high-accuracy forecasts of the stock market.

Artificial intelligence algorithms bypass the limitations of traditional statistical assumptions and do not require pre-processing of data. AI models have significant advantages over conventional time series methods in predicting financial time series data with non-stationary and non-linear characteristics. As science and technology advance, conventional machine learning algorithms, like the Artificial Neural Networks (ANN [4]), Support Vector Machines (SVM [5]), and Extreme Gradient Boosting (XGBOOST [6]) are challenging to precisely characterize the autocorrelation between time series, thereby constraining the precision of time series prediction. In addition, these traditional AI models often lack the mining of deep structure, thus limiting the comprehensiveness of time series data analysis and the precision of prediction. Based on the original AI techniques, researchers have continuously developed new deep learning models, such as the Convolutional Neural Network (CNN [7]), the Long Short-Term Memory Networks (LSTM [8]), the Gated Recurrent Unit (GRU [9]), and the Temporal Convolutional Network (TCN [10]), etc., and used them for financial forecasting, which all have higher forecasting accuracy compared with traditional models, providing a new way to analyze and forecast time series data in the financial field with more effective tools for time series analysis and forecasting.

Hybrid models have been more widely used by combining models based on a single model with enhanced feature extraction capabilities and more precise prediction outcomes.Researchers often combine signal decomposition algorithms and artificial intelligence models, and the choice and implementation of signal decomposition algorithms significantly influence the predictive performance of the hybrid model. Compared to the Wavelet Transform (WT [11]), ordinary Empirical Mode Decomposition (EMD [12]) does not necessitate the establishment of primitive functions as required by the transform, conferring a notable advantage in handling non-smooth and nonlinear complex signals. In recent years, numerous research endeavors have merged EMD or its variations including Ensemble Empirical Mode Decomposition (EEMD [13]), Variational Mode Decomposition (VMD [14]), and Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN [15]) with deep learning techniques for forecasting financial time series. Findings from these studies indicate a substantial enhancement in prediction accuracy when employing hybrid models integrating signal decomposition algorithms compared to individual models.

To enhance the prediction precision of stock indices, this paper introduces a novel stock index prediction model CEEMDAN-TCN-GRU-CBAM. The model integrates the CEEMDAN decomposition method with the fuzzy entropy algorithm to construct a hybrid network of TCN-GRU-CBAM, which can effectively capture both the local features and long-term dependencies present in the stock index series data, thereby enhancing the representation of its feature capabilities, thus enhancing the model's robustness and improving its prediction accuracy. This paper makes the following contributions:

A. IMPROVEMENT OF DECOMPOSITION METHOD
To more effectively capture the features and patterns within the sequences, this paper utilizes the CEEMDAN decomposition method for sequence decomposition, calculates the Fuzzy Entropy (FE), and reorganizes the chaotically similar components to form a new subsequence. It not only reduces the size of the subsequent prediction but also avoids the error superposition caused by multiple predictions.

B. IMPROVEMENT OF PREDICTION METHOD
To effectively improve the prediction precision and robustness, this study adopts the TCN-GRU-CBAM hybrid model, where TCN extracts the features and patterns in the sequence data, the inclusion of GRU addresses long-term dependencies within the time series, while the integration of the attention mechanism via CBAM enhances the focus on crucial information within the sequence, ultimately leading to a notable enhancement in prediction accuracy.

C. COMPARISON OF MODEL UNIVERSALITY
To ascertain the general applicability of the proposed model, emerging and developed markets serve as the contextual backdrop. Four key stock markets, namely, the Chinese market, the Indian market, the American market, and the Japanese market are selected for conducting extensive experiments. The findings consistently demonstrate the paper introduces a model that exhibits superior forecasting precision and reliability compared to alternative models.

The rest of the paper's structure is as follows: Section II provides a literature review of stock forecasting, including statistical learning methods, artificial intelligence methods, and combinatorial modeling forecasting methods. Section III gives the combination framework of the proposed model. Section IV describes in detail the methodology used in this paper. Section V provides the empirical study and analysis, including data description, data processing, time-step comparison, comparison of ablation experiments, comparison of the combination prediction models, and examination of the outcomes. Section VI provides a summary of the whole paper, and provides the follow-up research prospects.

II. RELATED WORKS
With the continuous development of financial markets, the topic of stock index forecasting has received continuous attention. Various forecasting methods, including statistical learning methods, artificial intelligence methods, and combinatorial modeling methods, have been developed to further improve the accuracy of financial time series forecasting, help

investors increase their investment returns, and appropriately avoid risks.

A. STATISTICAL LEARNING METHODS

Researchers have utilized different data and various methods to forecast stock prices. Given the nonlinear characteristics of stock prices, researchers first use the mathematical techniques of curve fitting, parameter estimation, and residual analysis to construct models and predict future trends, which are called statistical learning methods. Wang and Niu [16] used the seasonal ARIMA model to forecast Nasdaq data, and comparing it with the ARMA model, the forecasting effect is improved. Liu and Hung [17] used distributed GARCH and asymmetric GARCH to forecast volatility and consider the more suitable type of GARCH in the case of different data characteristics. Wang et al. [18] used the GM (1, 1) model to make short-term market forecasts and propose solutions.

B. ARTIFICIAL INTELLIGENCE METHODS

The stock market has a certain degree of complexity, and the use of statistical analysis to forecast stock prices requires a strong assumption base, in the era of big data, the applicability of such methods has been reduced to a certain extent. So far, machine learning methods have emerged, which can model non-stationary and non-linear data and have some applicability to financial time series data. Vui et al. [19] investigated various techniques for stock market forecasting using Artificial Neural Networks (ANN), reviewing the application in stock market forecasting. PrasadDas and Padhy [20] used two machine learning techniques, BP and SVM to forecast the futures trading prices in the Indian stock market. Jidong and Ran [21] constructed a dynamically weighted multifactor stock-picking strategy based on the XGBoost model. Manurung et al. [22] proposed an algorithm and a model based on the Long Short Term Memory (LSTM) for forecasting the foreign exchange market in Indonesia and compared it with the ARIMA model. Rahman et al. [23] proposed a model for predicting future prices in the stock market using gated recurrent units (GRUs) neural networks.

C. COMBINATORIAL MODELING METHODS

The effectiveness of hybrid stock index prediction models has been well-established by a plethora of existing studies, hybrid stock index prediction model usually uses the signal decomposition method to preprocess the nonlinear data to carry out the subsequent prediction work. Wen et al. [24] employed principal component analysis to isolate key technical indicators influencing stock prices, conducted data dimensionality reduction, and subsequently utilized LSTM for stock price modeling and prediction; Fang et al. [25] based on GRU, added the VIX information and the minimum absolute contraction and selection operator (Lasso) method, a new prediction model was proposed; Yao et al. [26] used multivariate empirical modal decomposition (MEMD) to put the output components into the TCN model for prediction, and the model predicted better; Han and Yu [27] proposed a Bidirectional Long Short-Term Memory (BiLSTM) model to predict the future prices of the historical prices of stocks; Yan [28] combined the AdaBoost feature selection with a two-layer long and short-term memory based model for predicting stock index futures prices. Although a single neural network model is simple, intuitive, easy to understand, and less computationally expensive, it is prone to overfitting and underfitting. In response to such problems, research in sequence prediction tends to combine individual networks or algorithms while enhancing the interpretability of the model predictions. Sunny et al. [29] combined RNN models, i.e., LSTM models and BiLSTM models, with appropriate hyper-parameter tuning; Mehtab and Sen [30] built regression models including two-layer CNN and three-layer LSTM networks to perform multi-step predictions; Jaiswal and Singh [31] built a hybrid convolutional recurrent modeling architecture based on two different deep learning models, CNN and GRU, for stock price prediction; Janssen et al. [32] improved TCN performance by adding an attention mechanism to TCN as a way to enhance the effectiveness of predicting stock prices; Xiaoyan et al. [33] proposed a TCN-GRU approach for short-term power load forecasting, using TCN to extract local features of time series and GRU for nonlinear fitting.

In summary, using the modal decomposition algorithm to decompose the original stock index series into sub-series with different time scales can capture and predict the volatility of the stock index series more accurately. Still, the complexity of the computation is relatively high. There is not enough differentiation between the sub-series, to address this problem, so this paper introduces fuzzy entropy (FE [34]). It combines it with the CEEMDAN algorithm, which reconstructs the decomposed series and reduces the sub-number of sequences. Based on previous research ideas, this paper utilizes TCN for feature extraction of stock index sequences, GRU to capture the long-term dependencies of sub-sequences, and introduces the CBAM to improve the prediction accuracy based on the TCN-GRU model, which, compared with the traditional attention mechanism, CBAM focuses on the different aspects of the data from the perspectives of both the channel attention and the spatial attention to effectively improve the prediction ability of the model [35].

III. COMBINATORIAL MODELING FRAMEWORK

This paper utilizes the CEEMDAN decomposition algorithm to fully decompose the original series, thereby generating multiple subsequences across various time scales. Simultaneously, the following enhancements are implemented to overcome the limitations of existing stock price prediction models relying on modal decomposition. First, for the problem of large scale of subsequent prediction caused by modal decomposition, the fuzzy entropy algorithm is introduced, which not only reduces the scale of subsequent prediction, but

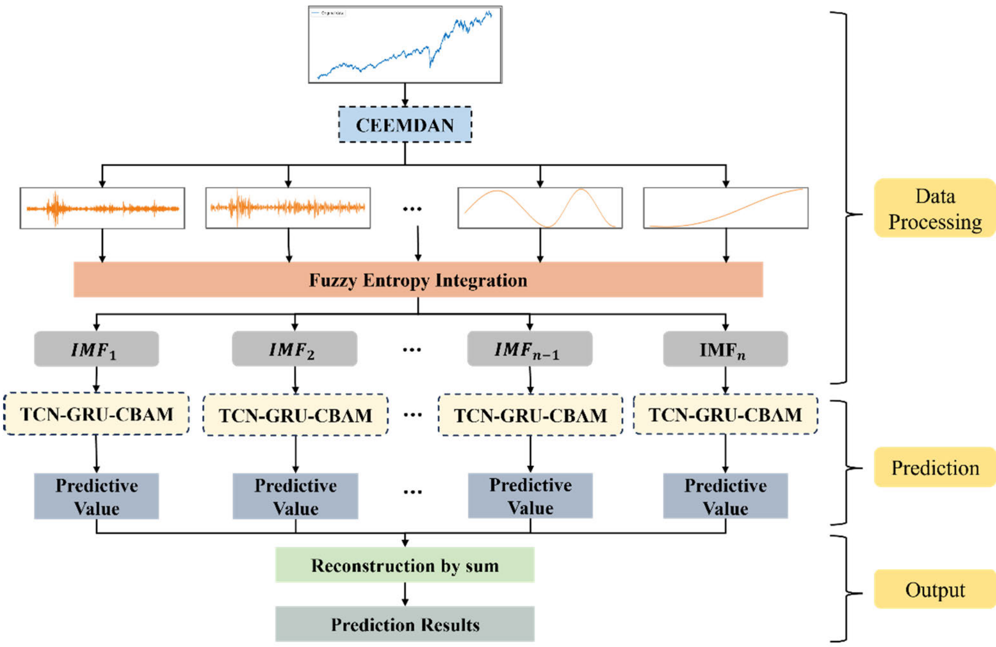

FIGURE 1. Construction framework of the CEEMDAN-TCN-GRU-CBAM model.

also avoids the superposition of errors brought by multiple predictions; second, through the use of a combination of TCN, GRU, and CBAM for learning and prediction of recombined sequences, it realizes the fuller exploitation of effective information of discontinuous features among the data, resulting in more dependable final prediction outcomes. The results are more reliable. The CEEMDAN-TCN-GRU-CBAM model mainly contains three parts, and the model framework is shown in FIGURE 1. The procedure is outlined as follows:

Step 1: During the data processing phase, firstly, the closing price data are processed by the CEEMDAN algorithm, and decomposed into a series of relatively smooth eigenmode functions and one residual component; secondly, the normalized fuzzy entropy value is calculated, and the sequences are reorganized according to the magnitude of the fuzzy entropy value to get a new set of sequences;

Step 2: During the prediction phase, the obtained new sequences are input into the TCN-GRU-CBAM model respectively, and the predicted values of different fluctuation frequency subsequences are obtained;

Step 3: During the result combination phase, the prediction outcomes of all closing price sub-sequences are aggregated and reconstructed to derive the final closing price prediction value.

IV. METHODOLOGY

In order to improve the prediction accuracy of stock indices, this study proposes a new hybrid prediction model, the construction of which consists of two parts: data decomposition and reorganization and combined model prediction. The data decomposition and restructuring technique is utilized to distinguish the frequency domain of sequence fluctuations and reduce the computational complexity, and the proposed new combined model prediction method fully integrates the advantages of the sub-models and improves the accuracy of the model prediction.

A. DATA DISAGGREGATION AND REORGANIZATION

We use decomposition followed by reconstruction for stock index series prediction, which utilizes the CEEMDAN method to separate the different frequency components. The decomposed components contain less noise and allow the model to more accurately capture the patterns in the data and make predictions only for the respective components, which reduces the learning of the noise and reduces the risk of overfitting. This improves the predictive performance of the model compared to the direct prediction of the original data.

This subsection describes the research methods specifically used in the data decomposition and reconstruction part of the model. The CEEMDAN decomposition method decomposes the original data and is used to distinguish the sequence fluctuation frequency domain; the fuzzy entropy algorithm is introduced to reconstruct the decomposed subsequence to reduce the complexity of the time series. The combination of the two methods avoids error superposition in the subsequent prediction.

1) CEEMDAN DECOMPOSITION

To address the issue of modal aliasing present in the decomposed signal of the EMD algorithm, both EEMD and CEEMD algorithms mitigate this problem by introducing pairwise positive and negative Gaussian white noises into the signal during decomposition. Despite this, there is still some remaining white noise within the intrinsic modal components, which could negatively impact the analysis and processing of subsequent signals. In response to these challenges, Torres et al. [36] introduced CEEMDAN based on the original decomposition method. Compared with the traditional empirical modal decomposition (EMD), CEEMDAN can better separate the frequency intervals between different modes; unlike the traditional frequency-domain methods (e.g., Fourier transform) or methods based on fixed basis functions (e.g., wavelet transform), CEEMDAN is an adaptive decomposition method. Illustrated in FIGURE 2, the CEEMDAN decomposition method involves adding IMF components containing auxiliary noise post-EMD decomposition, performing an overall averaging calculation after obtaining the first-order IMF components, and subsequently acquiring the final first-order IMF components. The procedure is repeated for the residual part, effectively resolving the issue of transferring and propagating white noise from high-frequency to low-frequency components.

Define $EMD_n(\cdot)$ as the modal component generated at the nth stage by the EMD algorithm, while the CEEMDAN algorithm generates the nth modal component denoted as IMF_n, and the algorithm is structured as follows:

The signal $f(t)$ earmarked for decomposition is augmented with a Gaussian white noise sequence, featuring a mean value of 0, for n iterations. This process constructs the sequence $f^i(t)$.intended for decomposition across a total of n experiments.

$$f^i(t) = f(t) + \varepsilon_0 \omega_0(t), \quad i = 1, 2, \ldots, n \tag{1}$$

where ε_0 is the signal-to-noise ratio and $\omega_i(t)$ is the ith added white noise sequence.

Utilize the EMD algorithm on each $f^i(t)$ for decomposition to extract the first modal component (IMF) and the initial unique residual component $r_1(t)$:

$$IMF_1(t) = \frac{1}{n}\sum_{i=1}^{n} IMF_1^i(t) = \frac{1}{n} EMD_1(f^i(t)) \tag{2}$$

$$r_1(t) = f(t) - IMF_1(t) \tag{3}$$

The residual component obtained after decomposition is added to the noise to continue applying EMD for decomposition:

$$IMF_k(t) = \frac{1}{n}\sum_{i=1}^{n} EMD_1(r_{k-1}(t) + \varepsilon_{k-1} EMD_{k-1}(\omega^i(t))), \quad k = 2, 3, \ldots, n \tag{4}$$

$$r_k(t) = r_{k-1}(t) - IMF_k(t) \tag{5}$$

Terminate the CEEMDAN algorithm when the residuals do not exceed two extreme points and the decomposition cannot be continued. At this point, the residual trend is obvious and direct, and the original signal sequence is disassembled into n modal components along with a residual term $R(t)$:

$$f(t) = \sum_{k=1}^{n} IMF_k(t) + R(t) \tag{6}$$

2) FUZZY ENTROPY

FE, introduced by Chen et al. [37] in 2007, was initially applied in myoelectric model processing and later adopted in various domains such as fault diagnosis and image processing. It is an enhanced method derived from Approximate Entropy and Sample Entropy, designed to gauge the likelihood of generating a new pattern within a time series as its dimensionality changes. FE enhances Sample Entropy by incorporating an exponential function called the fuzzy affiliation function. The calculation of fuzzy entropy is outlined as follows: For a time series signal $\{u(i) : 1 \leq i \leq N\}$ of length N, given m, form a vector sequence $\left\{X_i^m, i = 1, \ldots, N-m+1\right\}$ as follows:

$$X_i^m = \{u(i), u(i+1), \ldots, u(i+m-1)\} - u_0(i) \tag{7}$$

where $u_0(i)$ denotes the average value of m successive $x(i)$:

$$u_0(i) = \frac{1}{m}\sum_{j=0}^{m-1} u(i+j) \tag{8}$$

Define d_{ij}^m as the maximum difference value between the corresponding elements of the two m-dimensional vectors X_i^m and X_j^m:

$$\begin{aligned} d_{ij}^m &= d\left[X_i^m, X_j^m\right] \\ &= \max_{k\in(0,m-1)}(|(u(i+k) - u_0(i)) - (u(j+k) - u_0(j))|) \end{aligned} \tag{9}$$

where, $i, j = 1, 2, \ldots, N-m, i \neq j$.

Calculate the similarity D_{ij}^m: of X_i^m and X_j^m based on the fuzzy function:

$$D_{ij}^m = \mu(d_{ij}^m, n, r) = \exp(-(\frac{d_{ij}^m}{r})^n) \tag{10}$$

here $\mu(d_{ij}^m, n, r)$ is the exponential function, and n and r are the gradient and width of the exponential function boundary, respectively.

Construct the relation dimension $\Phi^m(n, r)$ in m dimensions:

$$\Phi^m(n, r) = \frac{1}{N-m}\sum_{i=1}^{N-m}(\frac{1}{N-m-1}\sum_{j=1, j\neq i}^{N-m} D_{ij}^m) \tag{11}$$

Similarly, construct m+1 dimensional vectors according to the above description:

$$\Phi^{m+1}(n, r) = \frac{1}{N-m}\sum_{i=1}^{N-m}(\frac{1}{N-m-1}\sum_{j=1, j\neq i}^{N-m} D_{ij}^{m+1}) \tag{12}$$

Define the fuzzy entropy as:

$$FE(m, n, r) = \lim_{N\to\infty}(\ln \Phi^m(n, r) - \ln \Phi^{m+1}(n, r)) \tag{13}$$

The computation of fuzzy entropy entails the selection of four parameters: the data length N, the embedding dimension m, and the boundary-related parameters r and n. Typically, an embedding dimension of m = 2 is employed. The parameter r is the breadth of the fuzzy function boundary,

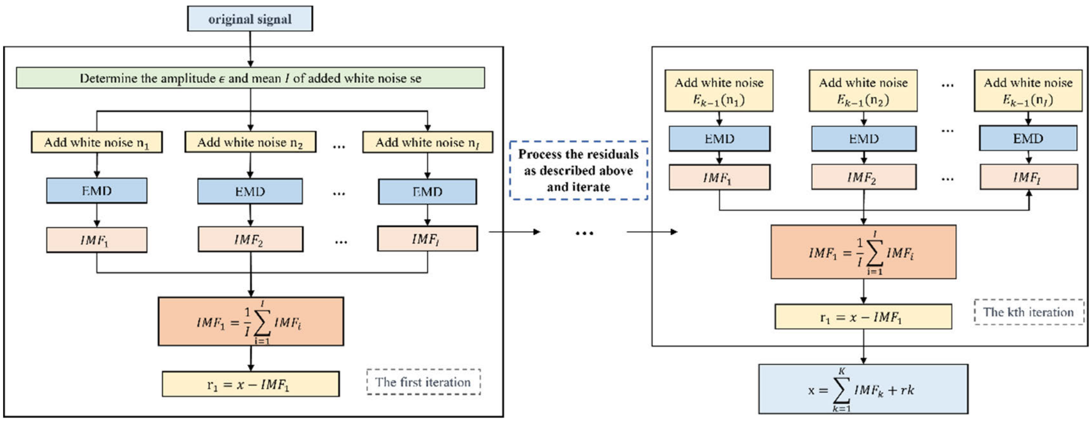

FIGURE 2. Specific steps of CEEMDAN decomposition.

often ranging between 0.1 and 0.25 times the standard deviation of the sequence (SD(x)). Meanwhile, n dictates the slope of the similarity tolerance boundary; larger values of n yield steeper gradients. Additionally, n functions as a weighting factor in determining the similarity among fuzzy entropy vectors. For capturing intricate details, smaller integer values like 2 or 3 are frequently employed during the computational process.

B. FORECASTING METHODOLOGY

This subsection describes the deep learning techniques and attention mechanisms used in combinatorial predictive modeling. Temporal convolutional networks are used to extract features and patterns in the time series, which are specifically introduced in two parts: causal convolution and null convolution; GRU has a simpler model structure and requires fewer parameter settings than LSTM; and the introduction of the CBAM attention module enhances the attention to important information, which is introduced in two aspects: spatial attention and spatial attention, which are introduced in this attention mechanism.

1) TEMPORAL CONVOLUTIONAL NETWORK

In this paper, we utilize TCN to deal with the long-term dependency of time series, the convolution in TCN supports parallel computation and has more efficient computational efficiency compared to RNN and LSTM. It can flexibly adjust the sensory field and can capture complex patterns and features in the data for subsequent modeling prediction.

TCN [38] adopts the idea of CNN and uses convolutional operations to capture local patterns and long-term dependencies in the input sequence, which avoids the problems of gradient vanishing and gradient explosion in RNNs, and has a longer "memory", which makes it easier to compute in parallel. It combines causal convolution, inflationary convolution, and residual network structure to make the receptive field larger and better modeling of time series data.

The causal convolution structure, as depicted in FIGURE 3, operates under strict time constraints. Within this framework, the value of the preceding layer at time 't' is determined solely by the value of the subsequent layer at time 't' and earlier instances. Unlike conventional convolutional neural networks, causal convolution lacks access to future data, thus embodying a unidirectional design. Consequently, it earns its name "causal convolution".

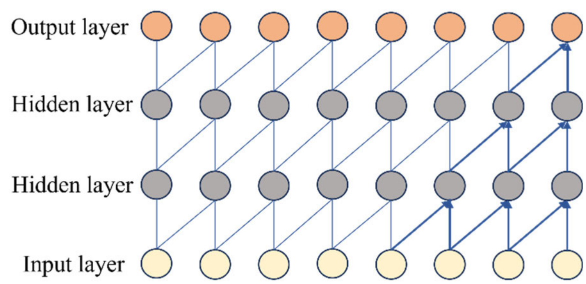

FIGURE 3. Structure of the causal convolution.

The modeling capacity of a single causal convolution is constrained by the size of the convolution kernel, limiting its ability to capture longer dependencies. To address this limitation, researchers have introduced inflated convolution. This is shown in FIGURE 4. Inflated convolution enables spaced sampling of the input, with the sampling rate determined by parameter d in FIGURE 4. A value of d = 1 implies that every point is sampled as input, while d = 2 indicates that every 2 points are sampled from the input. Typically, as the layer increases, the value of d also increases. Consequently, inflated convolution facilitates the exponential expansion of the effective window size as the number of layers increases. Essentially, the convolutional network achieves a broad sensory field using relatively few layers.

The residual module improves network training and gradient propagation by introducing residual connections. Consider x as the input value within the residual module, let $F(\cdot)$ denote the mapping function whose output is added to the input value x of the residual module. The output value

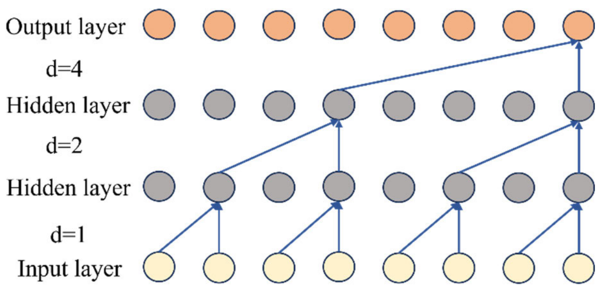

FIGURE 4. **Structure of the dilated convolution.**

generated by the residual module can be represented as:

$$o = Activation(x + F(x)) \tag{14}$$

This enables layers to focus on learning adjustments to the identity mapping rather than the entire transformation, favoring deeper nets. The receptive field of a TCN is contingent upon the network's depth (n), filter size (k), and expansion factor (d). Ensuring stability often requires larger and deeper TCNs. Each layer of the network comprises multiple filters for feature extraction. Hence, rather than conventional convolutional layers, a generic residual module is employed in the TCN model, as illustrated in FIGURE 5.

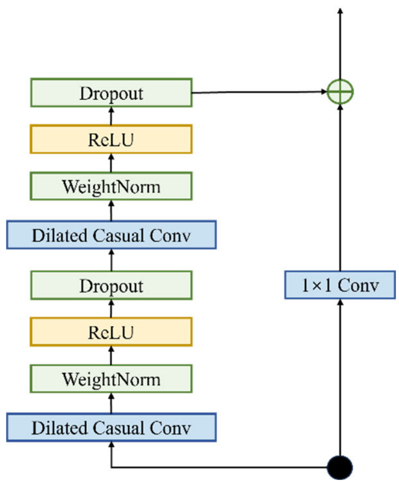

FIGURE 5. **Residual module of the TCN model.**

2) GATED RECURRENT UNIT

GRU originated from Chung et al. [39] in 2014. It addresses issues like gradient vanishing and explosion encountered in standard RNNs, enhancing the model's capability to capture long-term dependencies. Compared to the LSTM model, GRU simplifies the architecture by consolidating the three gates into two: the update gate and the reset gate. This streamlining reduces the model's complexity, trims down the number of trainable parameters, accelerates fitting, and shortens training durations.

The GRU model's structure is depicted in FIGURE 6, where h_{t-1} denotes the previous state, $\tilde{h}_t$ denotes the candidate state, z_t denotes the update gate, which controls how much information is retained by h_t from h_{t-1} as well as the amount of information to be received, r_t represents the reset gate, determining whether information from the previous state should be retained or discarded, and σ is the activation function.

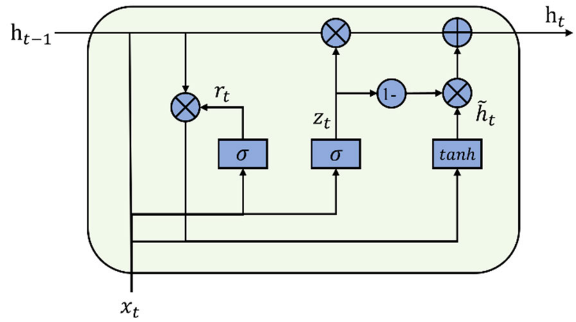

FIGURE 6. **GRU model structure.**

3) CONVOLUTIONAL BLOCK ATTENTION MODULE

CBAM was proposed by Woo et al. [40] to enhance the degree of attention of CNN to different regions in an image to improve the performance of the network, which was later applied in the field of sequence prediction. CBAM consists of two components, the spatial attention module and the channel attention module, which can focus on the key features, ignore the useless features, and focus on the inputs in a global and localized way different aspects of the data. This adaptable and streamlined module is designed to seamlessly integrate into different convolutional neural networks for end-to-end training. Illustrated in FIGURE 7 and FIGURE 8, when provided with a feature layer F, the channel attention module computes a one-dimensional channel attention vector M_c, indicating the relevance of each channel. Similarly, the spatial attention model computes a three-dimensional spatial attention map M_s, highlighting significant locations of various attention. The entire procedure unfolds as follows:

$$F' = M_c(F) \otimes F \tag{15}$$

$$F'' = M_s(F') \otimes F' \tag{16}$$

where $\otimes$ notes element-by-element multiplication, the channel attention module directs its attention to significant aspects within the input data. CBAM comprises two pools: MaxPool and AvgPool. The outputs from these pools are separately processed through the same fully connected layer. The resulting outcomes are then combined and normalized to yield the weight matrix across the channels. FIGURE 7 illustrates the configuration of the channel attention module.

$$M_c(F) = \sigma(MLP(Avgpool(F)) + MLP(Maxpool(F))) \tag{17}$$

The spatial attention module identifies the positional information that holds significance, acting as a complement to the channel attention, as depicted in FIGURE 8. In the input feature layer, both the maximum and average values are computed across the channels for each feature point. Subsequently, these two results are concatenated together.

$$M_s(F) = \sigma(Conv([AvgPool(F); MaxPool(F)])) \tag{18}$$

where $\sigma(\cdot)$ represents the Sigmoid function, $MaxPool(\cdot)$ represents maximum pooling, $AvgPool(\cdot)$ represents average pooling, $MLP(\cdot)$ denotes multilayer perceptron (MLP), and $Conv(\cdot)$ denotes the 3D convolutional layer.

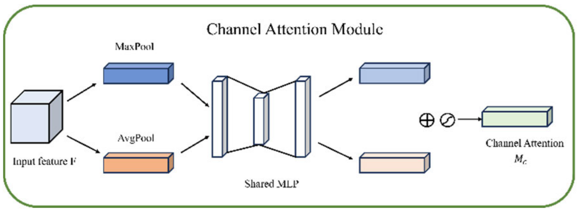

FIGURE 7. Channel attention module.

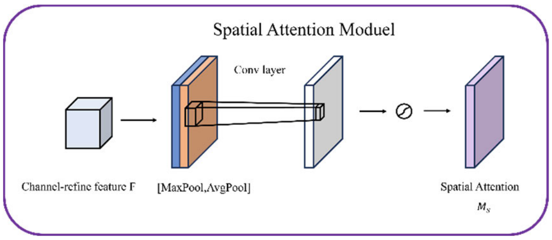

FIGURE 8. Spatial attention module.

4) TCN-GRU-CBAM COMBINED PREDICTION MODELS

Different network architectures possess distinct advantages in capturing data features, and a judicious combination of these structures can enhance the model's prediction accuracy. The integrated prediction model employs TCNs to capture information across various time scales, effectively extracting both local patterns and long-term dependencies within the time series data, with the added benefit of efficient parallelization. By incorporating the GRU network, the model further extracts temporal correlations within the features, mitigating the issue of gradient vanishing inherent in recurrent neural networks. Additionally, the introduction of the CBAM attention mechanism guides the model to focus on crucial information within the sequence, thereby enhancing the model's capability to predict sequence features. Within the CBAM module, a 1-dimensional convolutional layer is included at the module's input, where the input feature layer is first multiplied by the output weight of the channel mechanism, followed by multiplication with the output weight of the spatial mechanism. The combination of these two components is employed to capture important feature information within the sequence. Refer to FIGURE 9 for a detailed depiction of the model structure.

C. MEASUREMENT METRICS

In terms of model evaluation, this paper chooses four indicators [41], namely, the coefficient of determination R^2,

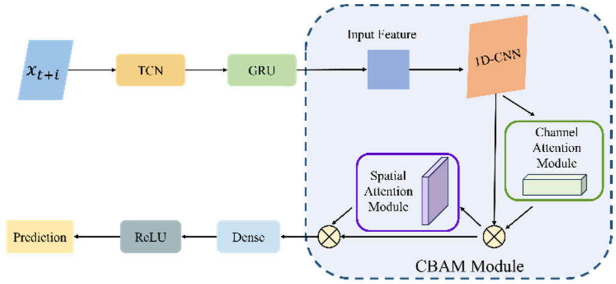

FIGURE 9. Structure of the TCN-GRU-CBAM model.

the root mean square error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE), to test the performance of the model, and the formulas for these indicators are as follows. Where: $\hat{y}_i$ is the predicted value, y_i is the initial actual value, n is the total count of forecasted values, and $\bar{y}$ is the mean of the initial values.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{19}$$

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{20}$$

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - \hat{y}_i \right| \tag{21}$$

$$MAPE(y, \hat{y}) = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{22}$$

R^2 represents the degree of agreement between the model and the data, and as the value approaches 1, the model is more accurately fitted. The RMSE and MAE values signify the discrepancy between predicted and actual values, where smaller values suggest enhanced prediction accuracy within the model. Similarly, a lower MAPE value also indicates better predictive accuracy.

V. EMPIRICAL ANALYSIS

This section outlines the dataset, the model evaluation metrics, the experimental analysis of comparative models, and the outcomes. Prediction studies are conducted using the TensorFlow deep learning framework. To mitigate the influence of random weight initialization on the prediction outcomes, this paper repeats the experiment several times to get the final prediction results.

A. DATA DESCRIPTION

Emerging markets and developed markets have different economic and market characteristics, and stock indices under different markets have different data characteristics; stock indices under emerging markets have high volatility, while stock indices in underdeveloped markets have relatively

TABLE 1. Description of the stock index dataset.

Categorization	Stock index	Country	Data length
Emerging markets	SSEC	China	2500
	BSESN	India	2539
Developed markets	SPX500	America	2588
	N225	Japan	2540

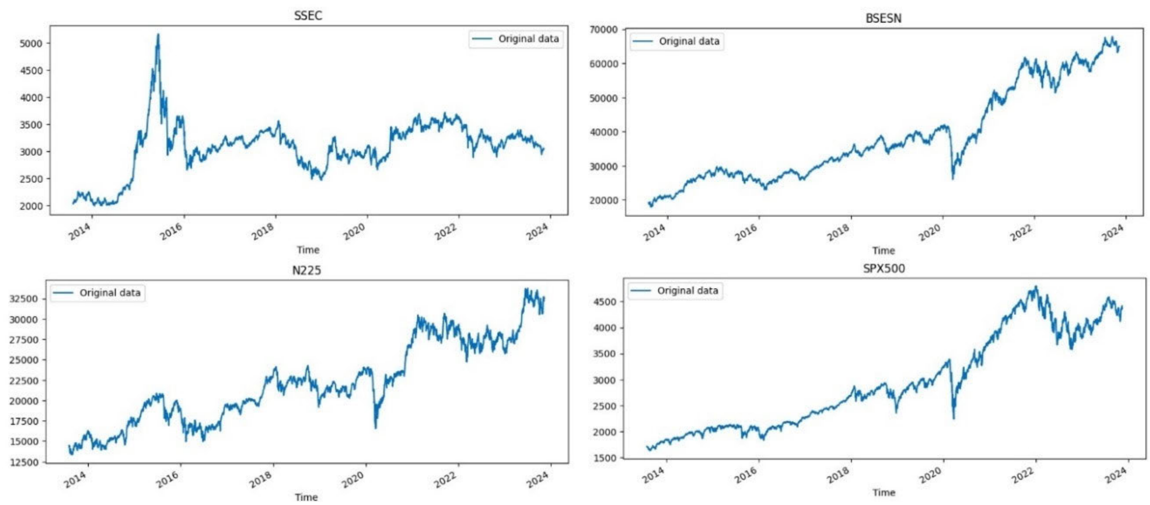

FIGURE 10. Trends in closing price volatility of equity indices.

smooth data. Therefore, the four most representative stock indices in developed and emerging markets that can capture the overarching shifts of their markets are selected as the research objects to test the universality and accuracy of the proposed model in this paper. The dataset includes SSEC (China), BSESN (India), SPX500 (America), and N225 (Japan), which are described in TABLE 1. In this paper, the closing prices of the four stock indices are used as the datasets for the time interval from August 2, 2013, to November 10, 2023, and the data are obtained from the website (http://cn.investing.com). Each stock index dataset is split into two segments: the training set, comprising 80% of the dataset, and the test set, comprising the remaining 20%, which is used for model training and performance testing, respectively. TABLE 1 lists the details of all the datasets. The closing price fluctuation trend of each stock index is shown in FIGURE 10.

After the decomposition algorithm obtains the subsequence, to accelerate the neural network training process and mitigate the impact of sequence dimensions on the results, it is essential to standardize the subsequence. This paper employs the maximum-minimum normalization technique, which scales the data to the range [0, 1]. The normalization formula is expressed as follows:

$$x_t' = \frac{x_t - \min}{\max - \min} \tag{23}$$

TABLE 2 demonstrates the specifics of the descriptive statistics for the four stock indexes. The normality test using the JB (Jarque-Bera) statistic shows that the four stock index data reject the original hypothesis and the series do not follow a normal distribution; the autocorrelation of the series is assessed using the LB (Ljung-Box) statistic. The results indicate that all p-values are below 0.05 at the highest 30th order, rejecting the original hypothesis, and the data have a strong autocorrelation; in this paper, the Augmented Dickey-Fuller (ADF) test is additionally employed to evaluate the data's stationarity, as depicted in TABLE 3, the p-value for each stock index dataset exceeds 0.05, and the test value of ADF is −2.7565, −0.1953, −0.7015, −1.0438, respectively, which surpass the critical test value at various significance levels, indicating that the time series under investigation lack smoothness. To summarize, the stock index series data exhibit non-stationarity and are characterized by significant noise, so the data is decomposed to continue with the subsequent forecasting.

B. DATA DECOMPOSITION AND REORGANIZATION

In this paper, we use Python 3.11.5 and the CEEMDAN function within the EMD-signal 1.0.0 module to decompose and reorganize the stock price indices from two perspectives, emerging markets and developed markets, SSEC and BSESN for emerging markets, and SPX500 and N225 for developed markets, and the decomposition outcomes of the stock indices

TABLE 2. Descriptive statistics.

	SSEC	BSESN	SPX500	N225
Count	2500	2539	2588	2540
Mean	3076.793	38488.46	2933.629	22047.51
Std	484.4979	13470.83	903.7037	4963.893
Max	5166.35	67838.63	4796.56	33753.33
Min	1991.25	17905.91	1630.48	13338.46
Skewness	-0.032	0.629	0.465	0.329
Kurtosis	1.692	-0.908	-1.151	-0.829
J-B	298.556	254.554	236.151	118.659
LB(30)	0.000	0.000	0.000	0.000

TABLE 3. ADF test.

Data	Test Statistic	P-value	Critical Values		
			1%	5%	10%
SSEC	-2.7565	0.0648	-3.433	-2.863	-2.567
BSESN	-0.1953	0.9391			
SPX500	-0.7015	0.8464			
N225	-1.0438	0.7370			

are illustrated in FIGURE 11, from top to bottom, as follows raw data, 8 IMF components and 1 residual component. The horizontal axis represents the quantity of stock index closing price time series, while the vertical axis represents values in dollars. Assuming consecutive closing prices for each trading day and disregarding intervals between trading days, from IMF0-IMF7 to the residual, both the frequency and complexity of the sequence gradually diminish. The trend of change is more visually apparent compared to the initial sequence, making the general price trend clearer.

The sequence obtained after CEEMDAN decomposition is normalized, and then the fuzzy entropy is calculated to assess the intricacy of the subsequence, FIGURE 12 shows the fuzzy entropy of the decomposed elements under different data sets. Grouping time series with comparable fuzzy entropy values can streamline computation, enhance modeling efficiency, and mitigate overfitting concerns. Taking the SSEC dataset as an example, the decomposed sequences are reorganized according to the fuzzy entropy results, and the new sequence is obtained as {imf0, imf1, imf2+imf3, imf4+imf5, imf6+imf7+imf8}, using this recombination sequence for subsequent forecasting. The recombination series for the four stock indices are shown in FIGURE 13, where the frequency of fluctuations in the different sub-sequences is visible.

C. TIME STEP COMPARISON

The time step size immediately impacts the model's capacity to capture changes across various time scales. A time step that is too small may result in longer training times, increased computational expenses, and a heightened risk of overfitting. Conversely, if the time step is excessively large, the model might fail to detect significant temporal patterns, leading to the loss of information and reduced prediction accuracy. Hence, selecting an appropriate time step is crucial. In the experiments, time steps of 5 d, 10 d, 15 d, and 20 d were utilized. FIGURE 14 illustrates the performance evaluation outcomes of the TCN-GRU-CBAM model across varying time steps. Based on the evaluation outcomes for MAE, MAPE, RMSE, and R^2, it's evident that the model achieves superior prediction performance when the time step is configured to 10 d. Therefore, the time input steps of the models in this paper are all set to 10d.

D. TCN-GRU-CBAM MODEL PREDICTION

To illustrate the superior performance of the combined TCN-GRU-CBAM model, the model proposed in this paper is contrasted with other benchmark models. The experiments are conducted using 50 epochs with a batch size of 32, and the training set and test set share are 80% and 20%, respectively. In addition, the parameter settings of both the proposed model and the comparison model are fine-tuned using the grid search algorithm, and the ultimate experimental parameters of the model are presented in TABLE 4, and the parameters of the comparison model are consistent with those of the proposed model.

TABLE 4. Parameters of the TCN-GRU-CBAM model.

Hyperparameters	Parameter settings
Number of TCN filters	64
Kernel size	2 × 2
Dilations	{1, 2, 4}
GRU units	20
Conv1D_filters	128
Conv1D_kenel size	1×1
Conv1D_ Activation	ReLU
CBAM_Activation	Sigmoid
Dropout	0.2
Optimizers	Adam
Activation function	ReLU
Loss function	MSE
Epochs	50
Batch size	32
Regularization	L2
Learning rate	0.001

The results of the comparison experiments based on different datasets are displayed in TABLE 5, TABLE 6, TABLE 7, and TABLE 8, the prediction results of BPNN and RNN in the SSEC dataset are the same, and the prediction results of LSTM and GRU are better than those of BPNN and RNN, the combined model, which integrates a single model with TCN, demonstrates superior prediction results compared to standalone models. Additionally, the performance of the TCN-GRU combined model surpasses that of other hybrid models. The TCN-GRU model is based on the functionality of the CBAM module, and comparing the results of TCN-GRU and TCN-GRU-CBAM, the MAPE value of TCN-GRU

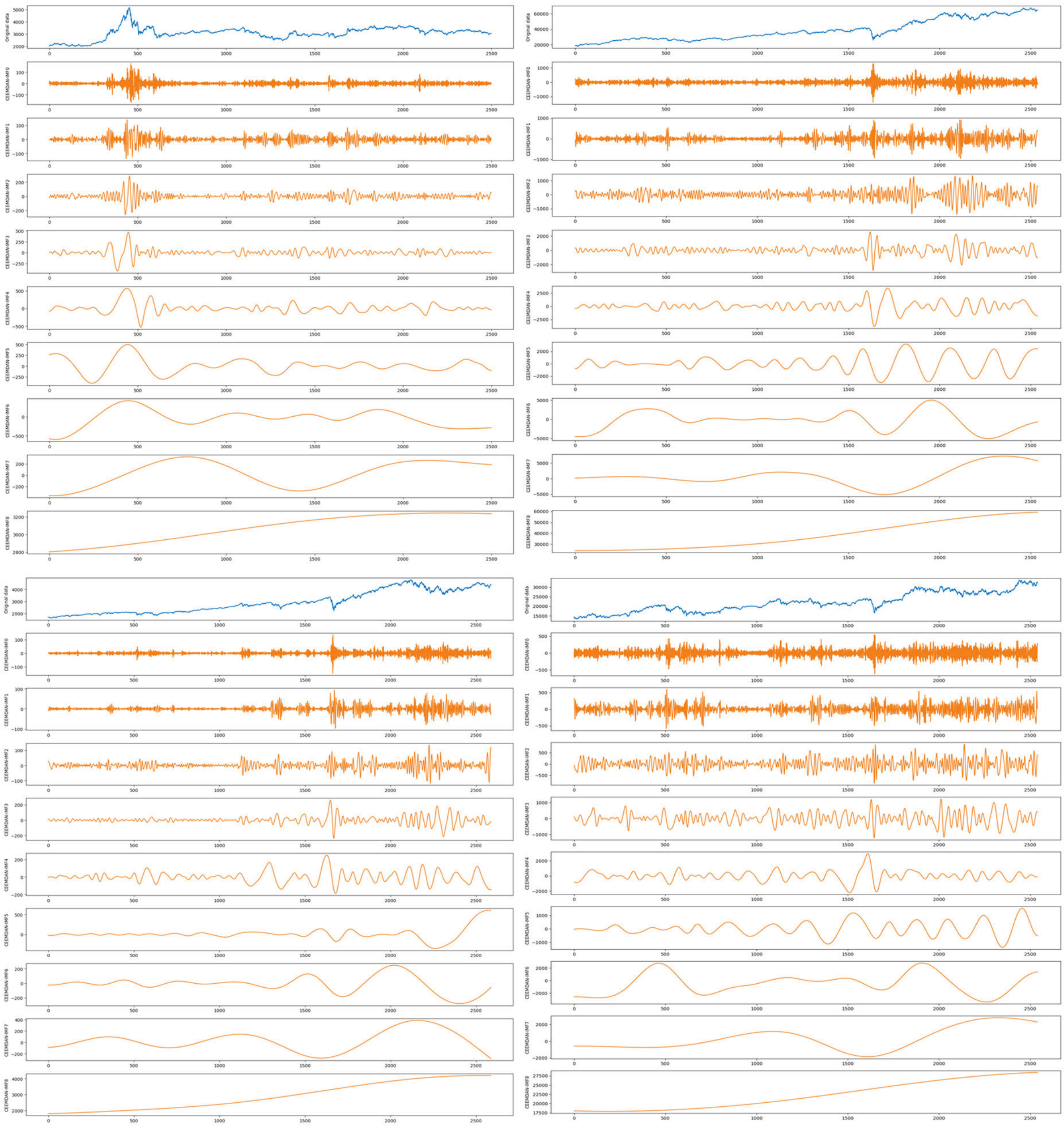

FIGURE 11. Decomposition results for stock index series.

is 1.0679, the RMSE value is 41.6028, the MAE value is 34.4503, and the R^2 value is 0.9276; TCN-GRU-CBAM's MAPE value is 0.9855, the RMSE value is 41.6028, the MAE value is 32.0122, and the R^2 value is 0.9592, which shows that the prediction on the original dataset is better than the TCN-GRU model for TCN-GRU-CBAM. On the SPX500 stock index dataset, the MAPE for TCN-GRU-CBAM value is 0.9746, the RMSE value is 52.1874, the MAE value is 40.4149, and the R^2 value is 0.9658, which shows that this model exhibits the smallest error and the highest level of fit in contrast with alternative models. In summary, for forecasting stock index data with diverse characteristics, the TCN-GRU-CBAM model exhibits the most accurate prediction performance, which indicates that this prediction model has a certain degree of universality and robustness.

Comparing the TCN-GRU-CBAM prediction performance on different datasets in Tables 5-Table 8, namely, the bolded part of the table, the values of RMSE and MAE are also similar due to the similar data scales of SSEC and SPX500, and the similar data scales of BSESN and N225, respectively. The performance of the model is improved compared with the benchmark model. SPX500 shows a long-term stable growth trend, SSEC, N225, and BSESN are all characterized by sharp

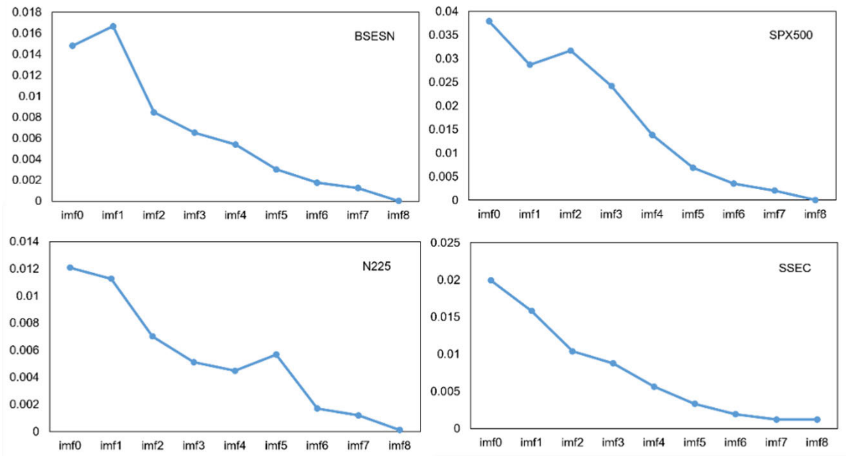

FIGURE 12. Fuzzy entropy corresponding to each component under different data sets.

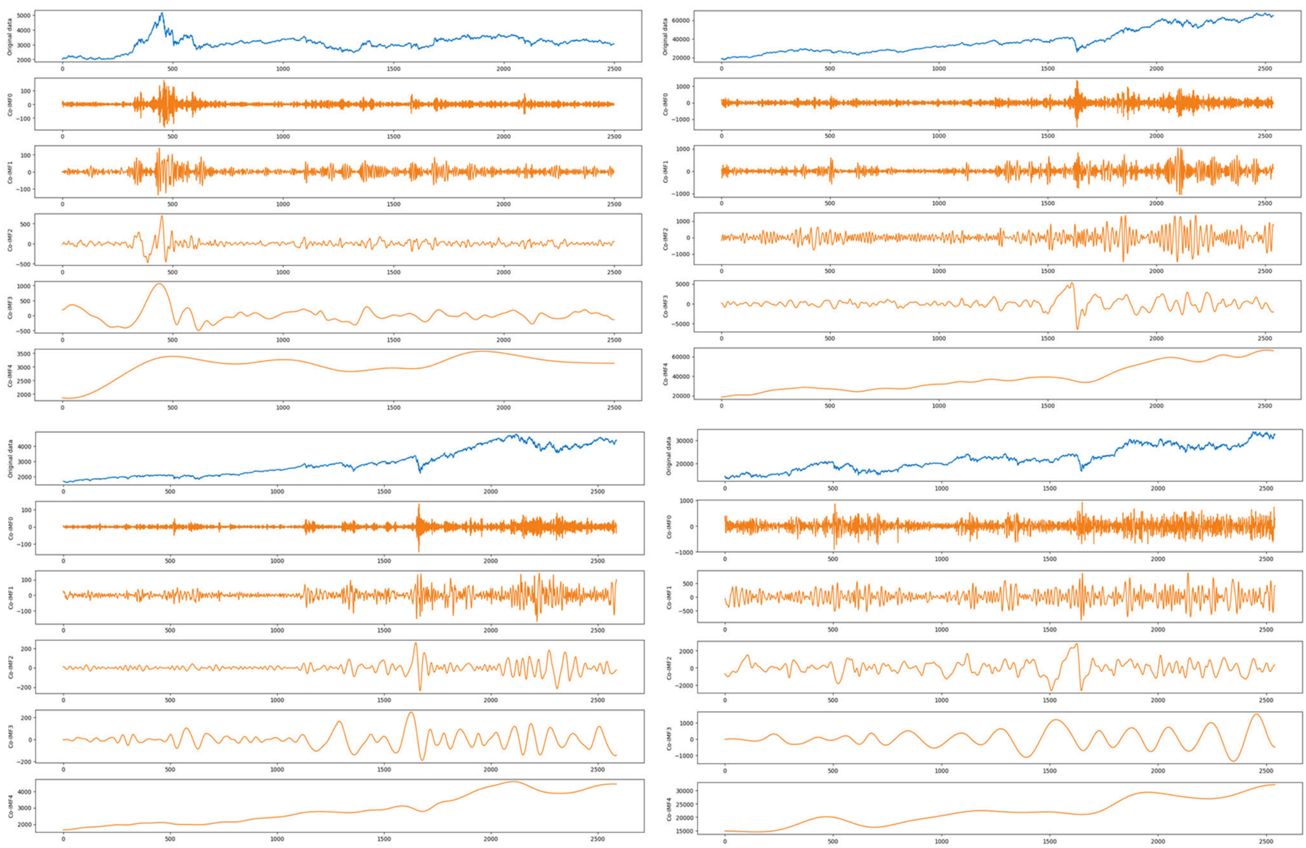

FIGURE 13. Recombination sequences on each dataset.

peaks and thick tails with bias, and the performance of the proposed model in this paper is better than the comparison model under different data characteristics.

The prediction results of all comparison models on different datasets are depicted in FIGURE 15, FIGURE 16, FIGURE 17, and FIGURE 18. The blue curves denote the actual values, while the orange curves represent the predicted values. It is evident that there is a lag in the prediction results of BPNN, RNN, LSTM, and GRU, indicating that the models have a delay in capturing and adapting to new information, and are unable to extract information pertinent to forthcoming trends from the data. By adding TCN to a single model for feature extraction, the lag is mitigated, and the combined model predictions fit the actual values better compared to a single model. Adding the CBAM module to TCN-GRU effectively diminishes the delay in model prediction, and the

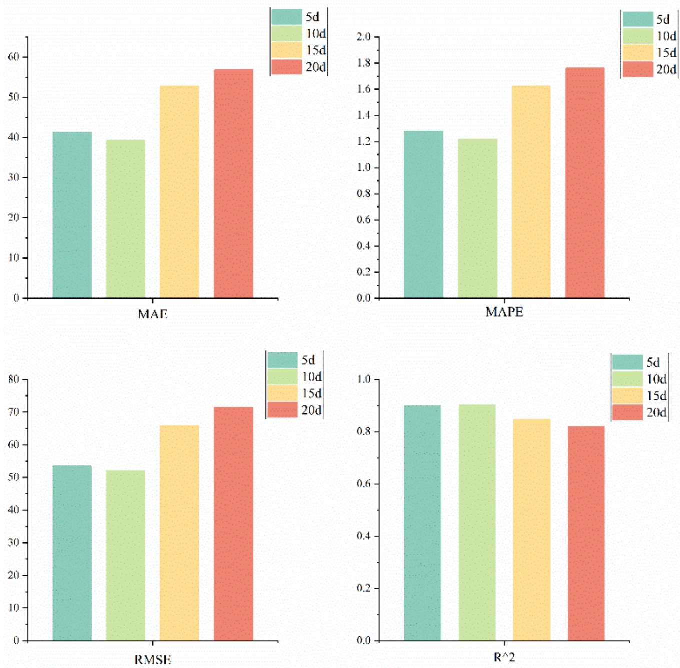

FIGURE 14. Comparison of model performance tests with different time steps.

TABLE 5. Comparative experiments of the TCN-GRU-CBAM model based on the SSEC dataset.

Models	MAPE	RMSE	MAE	R^2
BPNN	1.3462	57.2901	43.3171	0.8847
RNN	1.2812	54.6199	41.2487	0.8952
LSTM	1.2575	53.3384	40.5520	0.9101
GRU	1.2234	51.6039	39.4631	0.9165
TCN-BPNN	1.3266	55.3469	42.8508	0.8924
TCN-RNN	1.2543	52.7966	40.4301	0.9321
TCN-LSTM	1.0850	46.1314	34.9849	0.9353
TCN-GRU	1.0679	45.4134	34.4503	0.9476
TCN-GRU-CBAM	**0.9855**	**41.6028**	**32.0122**	**0.9592**

TABLE 6. Comparative experiments of the TCN-GRU-CBAM model based on the BSESN dataset.

Models	MAPE	RMSE	MAE	R^2
BPNN	1.8801	1411.9766	1113.9674	0.8527
RNN	1.7819	1322.4842	1061.5404	0.8708
LSTM	1.4323	1108.7113	842.1132	0.9092
GRU	1.2849	1015.0460	753.2081	0.9239
TCN-BPNN	1.6296	1244.7770	961.7002	0.8856
TCN-RNN	1.4716	1143.2931	866.9346	0.9035
TCN-LSTM	1.1980	935.2548	703.2612	0.9354
TCN-GRU	0.9909	750.5579	586.2060	0.9584
TCN-GRU-CBAM	**0.7612**	**558.1092**	**455.8884**	**0.9770**

fitting effect is greatly improved. On different stock index datasets, the prediction and fitting effect of the TCN-GRU-CBAM model outperforms other comparative models, but it is not able to achieve accurate prediction in the face of some mutation points, so it is necessary to decompose the data first, and then carry out subsequent prediction, to improve the model prediction effect, and to improve the model's predictive capability.

TABLE 7. Comparative experiments of the TCN-GRU-CBAM model based on the SPX500 dataset.

Models	MAPE	RMSE	MAE	R^2
BPNN	2.1112	111.3351	87.2003	0.8446
RNN	1.9492	104.3749	80.8299	0.8633
LSTM	1.6921	91.1969	69.8944	0.8957
GRU	1.6800	88.4829	69.5332	0.9018
TCN-BPNN	1.8850	99.8592	78.2030	0.8749
TCN-RNN	1.6965	91.0450	70.1417	0.8961
TCN-LSTM	1.4844	78.8302	61.3663	0.9221
TCN-GRU	1.2883	68.8962	53.2521	0.9404
TCN-GRU-CBAM	**0.9746**	**52.1874**	**40.4149**	**0.9658**

TABLE 8. Comparative experiments of the TCN-GRU-CBAM model based on the N225 dataset.

Models	MAPE	RMSE	MAE	R^2
BPNN	2.5926	897.2231	751.1585	0.8355
RNN	2.5427	882.7931	736.6348	0.8407
LSTM	2.0519	735.1952	583.7662	0.8895
GRU	1.9567	711.8986	554.0697	0.8964
TCN-BPNN	1.8354	683.8002	533.2947	0.9044
TCN-RNN	1.8140	656.0516	514.3074	0.9120
TCN-LSTM	1.5964	587.6897	456.6233	0.9294
TCN-GRU	1.5823	581.3506	452.7221	0.9309
TCN-GRU-CBAM	**1.2332**	**492.3012**	**367.0199**	**0.9504**

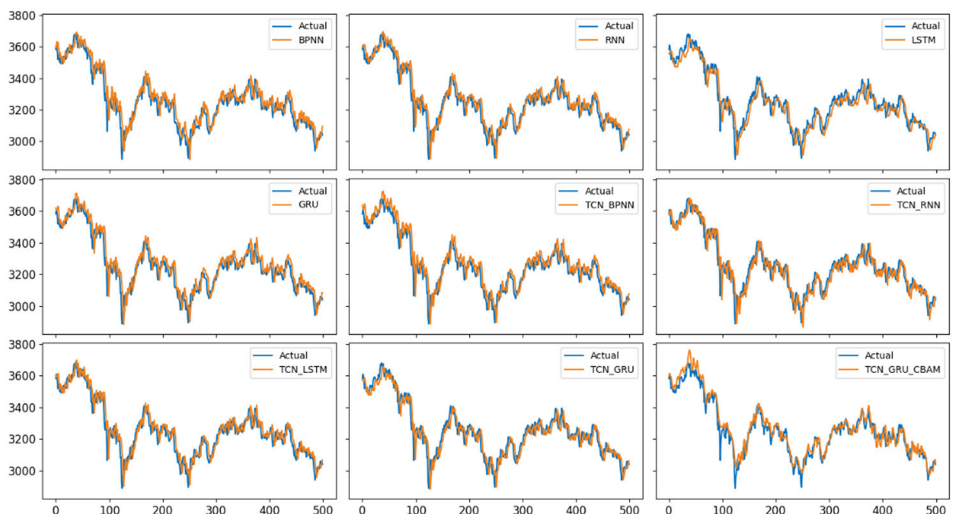

FIGURE 15. Plot comparing predicted and actual values for the SSEC dataset.

E. COMBINED MODEL PREDICTIONS

To confirm the effectiveness of each constituent module in the proposed approach, this paper utilizes the CEEMDAN decomposition to reorganize the subsequence for ablation experiments, taking the SSEC dataset as an example. First, the TCN model is utilized for forecasting alone; second, the GRU model is utilized for forecasting alone; third, the combined model of TCN and GRU is utilized for forecasting the stock index series; fourth, the combination of TCN and CBAM modules is utilized for forecasting; fifth, the combination of GRU and CBAM attention modules is utilized for forecasting; and lastly, it refers to the combined forecasting model proposed in this paper, TCN-GRU-CBAM. Performing the above ablation experiments allows us to verify the extent to which each module in the combined model contributes to the prediction performance. TABLE 9 quantitatively gives the evaluation indicators of the six models on the SSEC dataset, and the model proposed in this paper has a

FIGURE 16. Plot comparing predicted and actual values for the BSESN dataset.

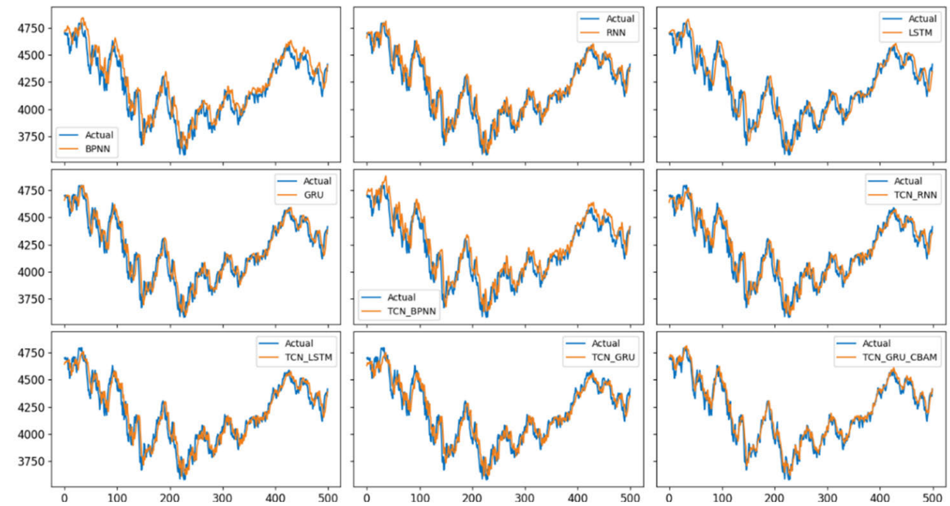

FIGURE 17. Plot comparing predicted and actual values for the SPX500 dataset.

MAPE value of 0.3784, an RMSE value of 15.1367, an MAE value of 12.2692, and an R^2 value of 0.9920, which are all the better than the single and combined models. This ablation experiment demonstrates that the combined TCN-GRU model outperforms individual models in terms of prediction accuracy by utilizing TCN to capture the local patterns and long-term dependencies, and GRU to extract temporal correlations in the features. TCN-CBAM and GRU-CBAM experimental results indicate that the addition of the CBAM attention module improves prediction performance compared to models without it, suggesting that the CBAM attention module can extract crucial information from sequences, enhancing predictive capabilities. The experimental results of TCN-GRU-CBAM demonstrate that the proposed model achieves the best results in all evaluation metrics, effectively combining the advantages of individual models.

FIGURE 19 illustrates the contrast between the predictive outcomes of the six models and the factual values in the ablation trials, where blue signifies the actual values and dark red represents the predicted values of the TCN-GRU-CBAM model. The graph highlights the superior alignment of the proposed predictive model with the actual values, consistently outperforming other models. This underscores the model's comprehensive amalgamation of module advantages within the integrated framework, leading to enhanced predictive outcomes.

It is known that the predictive efficacy of the TCN-GRU-CBAM model surpasses that of both individual and amalgamated models, so the TCN-GRU-CBAM model is used as the model for the prediction part, which is utilized to assess the efficacy of the integrated model in combination with different decomposition methods and to make a prediction by using

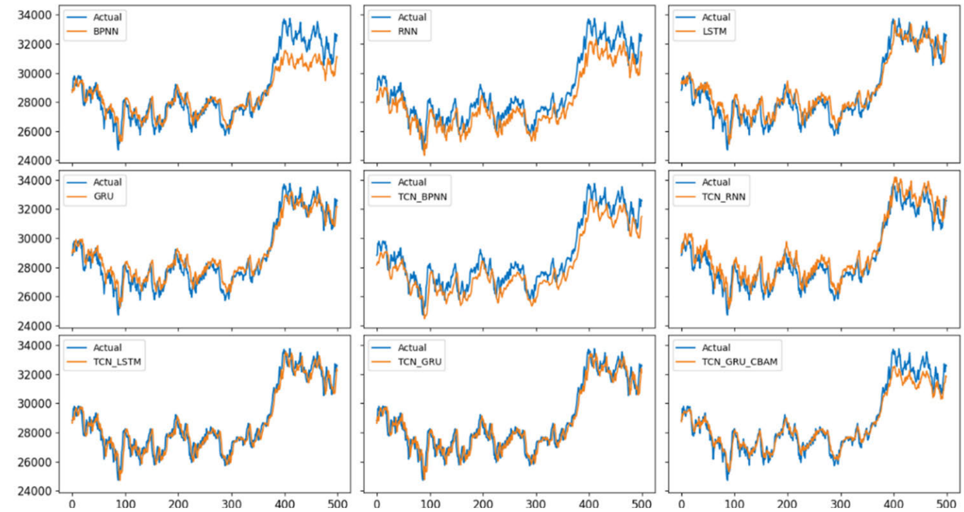

FIGURE 18. Plot comparing predicted and actual values for the N225 dataset.

TABLE 9. Prediction efficacy of the integrated model.

Model	MAPE	RMSE	MAE	R^2
TCN	1.3940	51.0439	45.4870	0.9085
GRU	1.2035	41.2842	39.2241	0.9402
TCN-GRU	0.8505	34.8285	27.4861	0.9574
TCN-CBAM	0.7539	30.1816	24.4418	0.9680
GRU-CBAM	0.6303	23.4014	20.5617	0.9808
TCN-GRU-CBAM	**0.3784**	**15.1367**	**12.2692**	**0.9920**

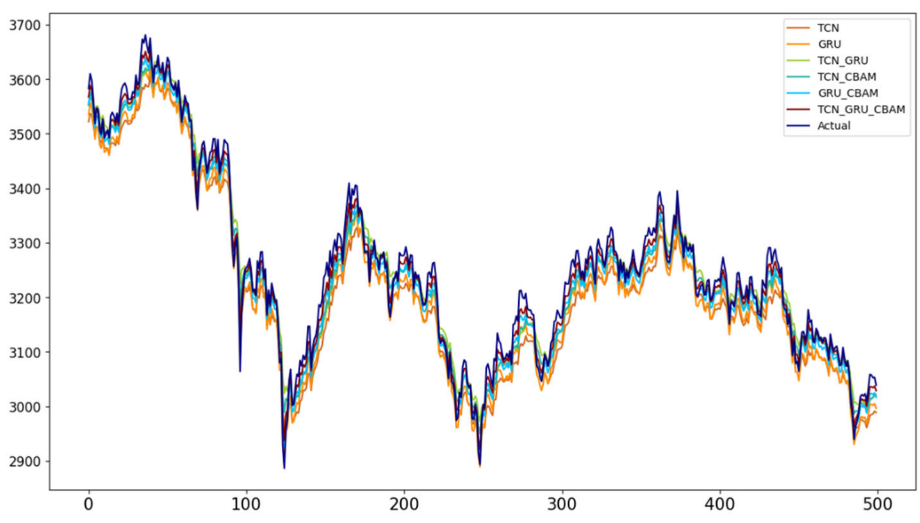

FIGURE 19. Comparison of predicted results from ablation experiments.

the subsequence data, and only the decomposition algorithms are changed in this part, which includes the EMD, EEMD, and VMD, CEEMDAN, the four decomposition algorithms, and CEEMDAN combined with fuzzy entropy. The predictive outcomes of the combined model under varied markets are displayed in FIGURE 20, FIGURE 21, FIGURE 22, and FIGURE 23, and the comparison of their performance tests is displayed in TABLE 10. Taking SSEC in emerging markets as an example, the MAPE value of the model proposed in this paper is 0.3784, the RMSE value is 15.1367, and the MAE value is 12.2692, all of which are lower than that of the hybrid model consisting of other decomposition methods, and with

TABLE 10. Comparison of portfolio model forecasts under emerging markets.

Stock market index		Models	MAPE	RMSE	MAE	R^2
Emerging Markets	SSEC	EMD-TCN-GRU-CBAM	1.5116	60.0839	49.2028	0.8732
		EEMD-TCN-GRU-CBAM	0.8694	37.9591	27.8909	0.9494
		VMD-TCN-GRU-CBAM	0.7290	31.7896	23.4280	0.9645
		CEEMDAN-TCN-GRU-CBAM	0.5732	24.3350	18.5457	0.9792
		Proposed model	**0.3784**	**15.1367**	**12.2692**	**0.9920**
	BSESN	EMD-TCN-GRU-CBAM	1.5310	1114.8572	929.0362	0.8982
		EEMD-TCN-GRU-CBAM	1.2619	876.8005	762.2052	0.9432
		VMD-TCN-GRU-CBAM	1.1600	819.3753	677.1676	0.9504
		CEEMDAN-TCN-GRU-CBAM	0.7612	558.1092	455.8884	0.9769
		Proposed model	**0.3806**	**279.0546**	**227.9442**	**0.9836**
Developed markets	SPX500	EMD-TCN-GRU-CBAM	2.5016	125.068	105.0101	0.8038
		EEMD-TCN-GRU-CBAM	1.8452	89.7019	77.7935	0.8991
		VMD-TCN-GRU-CBAM	1.4568	76.7433	59.7430	0.9261
		CEEMDAN-TCN-GRU-CBAM	1.0174	57.5470	42.5224	0.9585
		Proposed model	**0.9746**	**52.1874**	**40.4149**	**0.9658**
	N225	EMD-TCN-GRU-CBAM	3.0629	915.6387	864.6825	0.8189
		EEMD-TCN-GRU-CBAM	1.7099	559.5632	471.4054	0.9323
		VMD-TCN-GRU-CBAM	1.6648	547.0618	469.9922	0.9353
		CEEMDAN-TCN-GRU-CBAM	1.4096	474.4044	401.0598	0.9514
		Proposed model	**0.9495**	**326.9706**	**263.5874**	**0.9769**

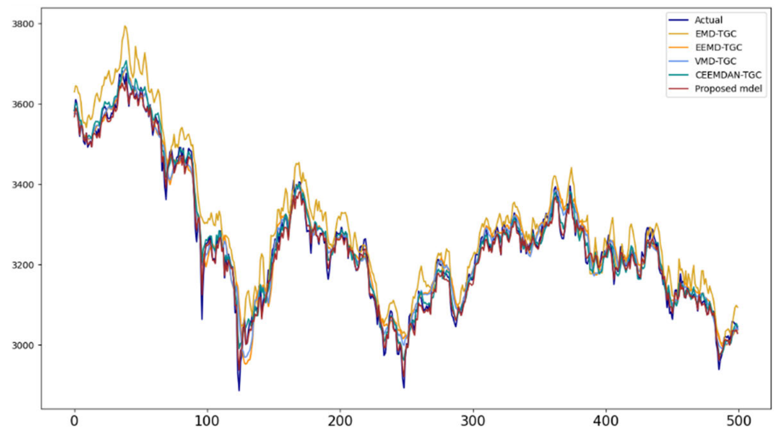

FIGURE 20. Comparison across all models using the SSEC dataset.

an R^2 value of 0.9920, the fitting effect is also better than that of other combination models; taking SPX500 in developed markets as an example, the model proposed in this paper has a MAPE value of 0.9746, an RMSE value of 52.1874, and an MAE value of 40.4149, all of which are better than the other combination models in terms of model performance ability, and an R^2 value of 0.9658, which is a certain degree of improvement in model fitting ability.

The impact of the model performance in terms of the characteristics of different stock indices is analyzed. Figures 20-23 show the prediction effect of four stock index sequences, red represents the predicted value of the proposed model, and blue represents the original value, it can be seen that the original value of the four sequences is very close to the predicted value, which indicates that the proposed model in this paper can get good prediction effect on different stock index data. Table 10 shows the performance of different stock indexes on the model performance, the bolded words represent the optimal effect, the prediction effect of the model proposed in this paper is better than the combination of other decomposition methods and the prediction method proposed in this paper. In the model proposed in this paper, the MAPE

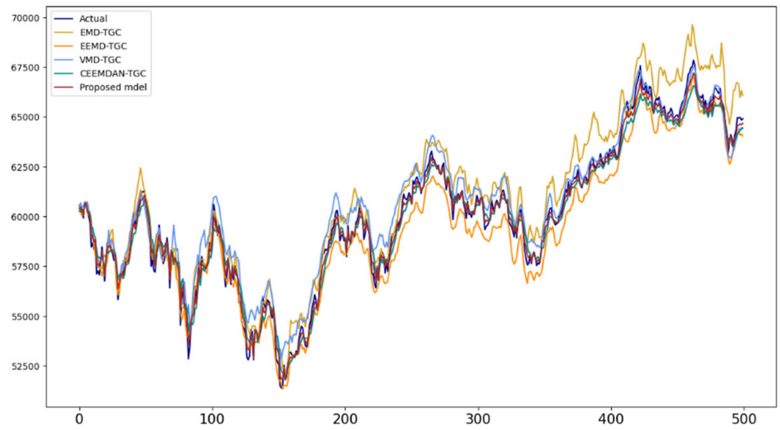

FIGURE 21. Comparison across all models using the BSESN dataset.

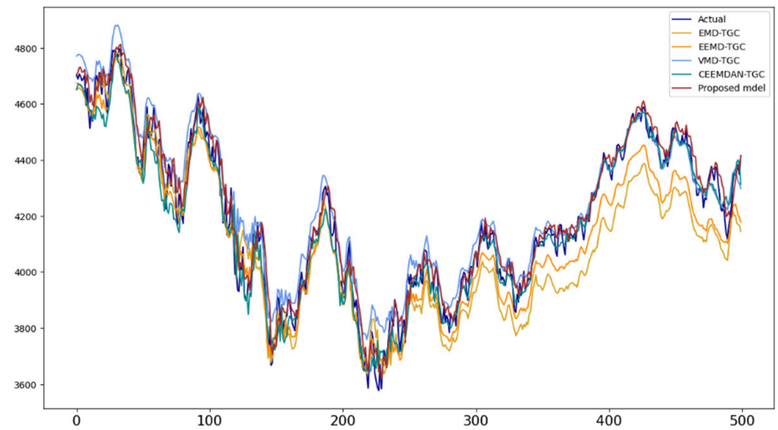

FIGURE 22. Comparison across all models using the SPX500 dataset.

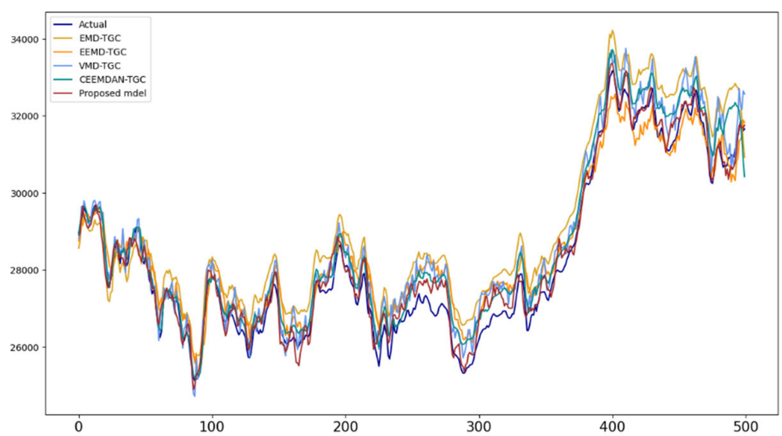

FIGURE 23. Comparison across all models using the N225 dataset.

values of the two stock index series of emerging markets are 0.3784 and 0.3806, which are not much different from each other, while the values of RMSE are 15.1367 and 279.0546, and the values of MAE are 12.2692 and 227.9442, which occurs because of the difference in the data scales of the two stock indices but the errors are all reduced when compared with the comparison model, the errors are reduced. The RMSE values of SPX500 and N225 in developed markets are 52.1874 and 326.9706, and the MAPE values are 40.4149 and 263.5874, respectively. Combined with Fig. 10, it can be seen that the different data scales and volatility have an impact on the performance of the model, but the predictive performance of the model proposed in this paper is improved compared to the other models over different data.

The experimental results indicate that the predictive performance of the decomposition using the CEEMDAN algorithm is better, and the reorganization of the sequences using fuzzy entropy on top of the CEEMDAN algorithm is a practice that further improves the predictive efficiency of the model. The predictive efficiency of this model is improved under different markets, and it has a certain degree of universality and stability. The model maximizes the benefits of the sub-models and the prediction results are improved on different stock index datasets. The combination prediction models based on the decomposition algorithm are all able to forecast the forthcoming stock index trends better, but in comparison to other combination models, the forecasted values from the model introduced in this paper closely align with the actual trend of the closing price, indicating higher accuracy.

VI. CONCLUSION

As artificial intelligence technology continues to expand and evolve, the concept of "decomposition forecasting" has surged in popularity for analyzing time series data, particularly within the domain of stock index prediction. This approach holds great practical significance given the profound impact of stock markets on the broader economy and people's lives.

This paper introduces a novel stock index portfolio prediction model that leverages modal decomposition and deep learning techniques to enhance the accuracy of stock index predictions. In terms of modeling, the prediction accuracy is improved after two stages of improvement; in terms of data selection, four stocks representative of emerging and developed markets are studied, namely, SSEC, BSESN, SPX500, and N225.

In this paper, extensive experimentation validates the efficacy of the proposed hybrid model. Here are the findings from the experimental analysis: firstly, the predictive performance of the proposed hybrid model TCN-GRU-CBAM is verified. Through comparisons with various fundamental models such as BPNN, RNN, LSTM, etc., the comprehensive evaluation demonstrates the robustness of the hybrid model proposed in this paper against noise interference, its efficacy in information extraction, and its superior predictive performance. Secondly, ablation experiments are carried out using the recombined subsequence after CEEMDAN decomposition to the efficacy of individual modules within the proposed methodology outlined in this paper. Finally, four distinct decomposition approaches, EMD, EEMD, VMD, and CEEMDAN, are combined with the model for experiments, and findings indicate that the prediction model utilizing the CEEMDAN decomposition method yields the most effective results. Moreover, integrating the CBAM attention module further enhances the predictive efficiency of the model, and the combined model CEEMDAN-TCN-GRU-CBAM utilizes the advantages of each constituent module to advantages to obtain more accurate prediction results. These results robustly showcase the enhanced prediction accuracy of the model for stock indices and highlight its practical value in stock index series prediction.

Although this paper has a certain degree of novelty and contribution, the model proposed in this paper still has some limitations. For example, the model's performance is more sensitive to selecting hyperparameters, and an empirical and tuning process needs to be utilized to select appropriate hyperparameters. Other decomposition methods can also be tried in data decomposition, and optimization algorithms can be used to optimize the model parameters to further enhance the noise reduction ability of the model. In the prediction model, other attention modules can be selected to improve the prediction ability. In addition, the application of the model can be extended to wind speed prediction, electric charge prediction, and traffic flow prediction in the future.

REFERENCES

[1] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, Dec. 2005.

[2] Z. Lin, "Modelling and forecasting the stock market volatility of SSE composite index using GARCH models," *Future Gener. Comput. Syst.*, vol. 79, pp. 960–972, Feb. 2018.

[3] Q. Wang and X. Song, "Forecasting China's oil consumption: A comparison of novel nonlinear-dynamic grey model (GM), linear GM, nonlinear GM and metabolism GM," *Energy*, vol. 183, pp. 160–171, Sep. 2019.

[4] N. Merh, V. P. Saxena, and K. R. Pardasani, "A comparison between hybrid approaches of ANN and ARIMA for Indian stock trend forecasting," *Bus. Intell. J.*, vol. 3, no. 2, pp. 23–43, 2010.

[5] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.

[6] Y. Yang, Y. Wu, P. Wang, and X. Jiali, "Stock price prediction based on XGBoost and LightGBM," in *Proc. E3S Web Conf.*, vol. 275, 2021, p. 1040.

[7] E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN-based stock market prediction using a diverse set of variables," *Exp. Syst. Appl.*, vol. 129, pp. 273–285, Sep. 2019.

[8] Z. Nourbakhsh and N. Habibi, "Combining LSTM and CNN methods and fundamental analysis for stock price trend prediction," *Multimedia Tools Appl.*, vol. 82, no. 12, pp. 17769–17799, May 2023.

[9] Y. Gao, R. Wang, and E. Zhou, "Stock prediction based on optimized LSTM and GRU models," *Sci. Program.*, vol. 2021, pp. 1–8, Sep. 2021.

[10] C. Guo, X. Kang, J. Xiong, and J. Wu, "A new time series forecasting model based on complete ensemble empirical mode decomposition with adaptive noise and temporal convolutional network," *Neural Process. Lett.*, vol. 55, no. 4, pp. 4397–4417, Aug. 2023.

[11] A. Kulaglic and B. B. Üstündag, "Stock price forecast using wavelet transformations in multiple time windows and neural networks," in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2018, pp. 518–521.

[12] W. Shu and Q. Gao, "Forecasting stock price based on frequency components by EMD and neural networks," *IEEE Access*, vol. 8, pp. 206388–206395, 2020.

[13] Y. Lu, B. Sheng, G. Fu, R. Luo, G. Chen, and Y. Huang, "Prophet-EEMD-LSTM based method for predicting energy consumption in the paint workshop," *Appl. Soft Comput.*, vol. 143, Aug. 2023, Art. no. 110447.

[14] Y. Yujun, Y. Yimei, and Z. Wang, "Research on a hybrid prediction model for stock price based on long short-term memory and variational mode decomposition," *Soft Comput.*, vol. 25, no. 21, pp. 13513–13531, Nov. 2021.

[15] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Phys. A, Stat. Mech. Appl.*, vol. 519, pp. 127–139, Apr. 2019.

[16] W. Wang and Z. Niu, "Time series analysis of NASDAQ composite based on seasonal ARIMA model," in *Proc. Int. Conf. Manage. Service Sci.*, Beijing, China, Sep. 2009, pp. 1–4.

[17] H.-C. Liu and J.-C. Hung, "Forecasting S&P-100 stock index volatility: The role of volatility asymmetry and distributional assumption in GARCH models," *Exp. Syst. Appl.*, vol. 37, no. 7, pp. 4928–4934, Jul. 2010.

[18] S.-J. Wang, W.-L. Wang, C.-T. Huang, and S.-C. Chen, "Improving inventory effectiveness in RFID-enabled global supply chain with grey forecasting model," *J. Strategic Inf. Syst.*, vol. 20, no. 3, pp. 307–322, Sep. 2011.

[19] C. S. Vui, G. K. Soon, C. K. On, R. Alfred, and P. Anthony, "A review of stock market prediction with artificial neural network (ANN)," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng.*, Penang, Malaysia, Nov. 2013, pp. 477–482.

[20] S. PrasadDas and S. Padhy, "Support vector machines for prediction of futures prices in Indian stock market," *Int. J. Comput. Appl.*, vol. 41, no. 3, pp. 22–26, Mar. 2012.

[21] L. Jidong and Z. Ran, "Dynamic weighting multi factor stock selection strategy based on XGboost machine learning algorithm," in *Proc. IEEE Int. Conf. Saf. Produce Informatization (IICSPI)*, Chongqing, China, Dec. 2018, pp. 868–872.

[22] A. H. Manurung, W. Budiharto, and H. Prabowo, "Algorithm and modeling of stock prices forecasting based on long short-term memory (LSTM)," *ICIC Exp. Lett.*, vol. 12, no. 12, pp. 1277–1283, Dec. 2018.

[23] M. O. Rahman, M. S. Hossain, T. S. Junaid, M. S. A. Forhad, and M. K. Hossen, "Predicting prices of stock market using gated recurrent units (GRUs) neural networks," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 1, pp. 213–222, Jan. 2019.

[24] Y. Wen, P. Lin, and X. Nie, "Research of stock price prediction based on PCA-LSTM model," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 790, no. 1, Mar. 2020, Art. no. 012109.

[25] W. Fang, S. Zhang, and C. Xu, "Improving prediction efficiency of Chinese stock index futures intraday price by VIX-Lasso-GRU model," *Exp. Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121968.

[26] Y. Yao, Z.-Y. Zhang, and Y. Zhao, "Stock index forecasting based on multivariate empirical mode decomposition and temporal convolutional networks," *Appl. Soft Comput.*, vol. 142, Jul. 2023, Art. no. 110356.

[27] C. Han and X. Fu, "Challenge and opportunity: Deep learning-based stock price prediction by using bi-directional LSTM model," *Frontiers Bus., Econ. Manage.*, vol. 8, no. 2, pp. 51–54, Apr. 2023.

[28] W.-L. Yan, "Stock index futures price prediction using feature selection and deep learning," *North Amer. J. Econ. Finance*, vol. 64, Jan. 2023, Art. no. 101867.

[29] Md. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi, "Deep learning-based stock price prediction using LSTM and bi-directional LSTM model," in *Proc. 2nd Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Giza, Egypt, Oct. 2020, pp. 87–92.

[30] S. Mehtab and J. Sen, "Stock price prediction using CNN and LSTM-based deep learning models," in *Proc. Int. Conf. Decis. Aid Sci. Appl. (DASA)*, Sakheer, Bahrain, Nov. 2020, pp. 447–453.

[31] R. Jaiswal and B. Singh, "A hybrid convolutional recurrent (CNN-GRU) model for stock price prediction," in *Proc. IEEE 11th Int. Conf. Commun. Syst. Netw. Technol. (CSNT)*, Indore, India, Apr. 2022, pp. 299–304.

[32] P. Janssen, "Attention based temporal convolutional network for stock price prediction," Utrecht Univ., 2022.

[33] H. Xiaoyan, L. Bingjie, S. Jing, L. Hua, and L. Guojing, "A novel forecasting method for short-term load based on TCN-GRU model," in *Proc. IEEE Int. Conf. Energy Internet (ICEI)*, Southampton, U.K., Sep. 2021, pp. 79–83.

[34] H. Gao, H. Jia, and L. Yang, "An improved CEEMDAN-FE-TCN model for highway traffic flow prediction," *J. Adv. Transp.*, vol. 2022, pp. 1–20, May 2022.

[35] Y. Liang, Y. Lin, and Q. Lu, "Forecasting gold price using a novel hybrid model with ICEEMDAN and LSTM-CNN-CBAM," *Exp. Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117847.

[36] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011, pp. 4144–4147.

[37] W. Chen, Z. Wang, H. Xie, and W. Yu, "Characterization of surface EMG signal based on fuzzy entropy," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 2, pp. 266–272, Jun. 2007.

[38] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1003–1012.

[39] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[40] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 11211. Cham, Switzerland: Springer, 2018, pp. 3–19.

[41] H. Xia, J. Zheng, Y. Chen, H. Jia, and C. Gao, "Short-term wind speed combined forecasting model based on multi-decomposition algorithms and frameworks," *Electric Power Syst. Res.*, vol. 227, Feb. 2024, Art. no. 109890.

SIBO LI is currently pursuing the M.A.S. degree with the School of Mathematics and Statistics, Guilin University of Technology, China. Her current research interest includes applied statistics.

GUOQIANG TANG received the M.S. degree in applied mathematics from Kunming University of Science and Technology, China, in 1998, and the Ph.D. degree in probability and mathematical statistics from East China Normal University, China, in 2009. He is currently a Professor with the School of Mathematics and Statistics, Guilin University of Technology, China. His research interest includes applied statistics.

XUCHANG CHEN received the master's degree in statistics from the School of Mathematics and Statistics, Guilin University of Technology, China. His research interest includes financial econometrics.

TONGZHI LIN received the master's degree from the School of Mathematics and Statistics, Guilin University of Technology, China. He is currently pursuing the Ph.D. degree with Guilin University of Technology. His research interest includes economic spatio-temporal big data analysis.