

RESEARCH ARTICLE

LSTM-DGMDH: High-Dimensional Index Tracking Based on LSTM and Adaptive Deep Evolutionary GMDH Neural Network

HE TONG¹, YUSHENG LIU², LIN LIU³, NING LI², SIBAO CHEN⁴, (Member, IEEE),
LIXIANG XU², (Member, IEEE), AND YUANYAN TANG⁵, (Life Fellow, IEEE)

¹Department of Basic, Chinese People's Liberation Army Aviation Institute, Beijing 101123, China

²School of Artificial Intelligence and Big Data, Hefei University, Hefei, Anhui 230601, China

³School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230601, China

⁴School of Computer Science and Technology, Anhui University, Hefei, Anhui 230601, China

⁵Zhuhai UM Science and Technology Research Institute, FST, University of Macau, Macau

Corresponding author: Lixiang Xu (xulixianghf@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62176085 and Grant 62172458; in part by the Industry-University-Research Cooperation Project, Zhuhai, Guangdong, China, under Grant GP/026/2020 and Grant HF-010-2021; in part by the Talent Fund of Hefei University under Grant 20RC25; in part by Intelligent Computing and Information Processing Key Laboratory of the Ministry of Education under Grant 2021A001; in part by the Program for Scientific Research Innovation Team in Colleges and Universities of Anhui Province under Grant 2022AH010095; and in part by the Anhui Natural Science Foundation project under Grant 2108085QA16.


ABSTRACT Stock index is an indicator that describes the changes in the total price level of the stock market, and it is susceptible to many dynamic factors, with such characteristics as high dimension, uncertainty, non-linearity, time delay, complexity, etc., resulting in abnormal and missing values in stock index data, which will lead to instability or unreliability of the stock index tracking model. In order to solve these problems, we take the historical stock index as the input, model the internal dynamic changes of features, and learn the change rule. Firstly, we introduce an attention mechanism, that is, to assign different weights to the implicit state of the long short term memory network (LSTM) through mapping weights and learning parameters. We further propose a stock index data preprocessing model of the LSTM based on the attention mechanism. Secondly, the group method of data handling type neural networks (GMDH-NN) is a self-organizing data mining technology, which is especially suitable for modeling complex systems. So we choose a discrete form of Kolmogorov-Gabor ($K - G$) polynomial of the first-order as the reference function of GMDH-NN to establish the general relationship between input and output variables. We further present a deep evolutionary GMDH polynomial neural network (DGMDH) to perform stock index tracking. Moreover, for a high-dimensional stock index dataset, the traditional external criterion can no longer meet the needs of reality, so we propose a tracking error external criterion (TEEC) for stock indices, which is based on the difference between allocation yield and target yield. The TEEC provides better information for selecting the optimal complex DGMDH model. Our experiments clearly show the effectiveness of our methodology.

INDEX TERMS LSTM, attention mechanism, GMDH neural network, tracking error external criterion, high-dimensional index tracking.

I. INTRODUCTION

A. RELATED WORK

A plethora of complex data will be generated in the financial and securities markets of our globalized economy every day

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

[1]. Faced with such a large amount of data, the traditional statistical methods can no longer meet the application demands [2], [3]. Making sophisticated statistical methods applicable to the existing market data and to provide better data analysis for investor decision-making have become a hot topic. In particular, in the fields of index tracking, portfolio management, and risk hedging, broad application platforms for feature

selection methods arise [4], [5]. Index tracking is a significant investment strategy [2], [3] in fund management that aims to replicate the movements of a specific market index. It involves selecting stocks from the target index to achieve a similar yield as the overall market performance. Compared with active investment, index tracking has the advantages of low risk, low transaction frequency, low transaction costs, and low management costs.

There are two common strategies for index tracking. In the full replication method, all assets in the target index are also purchased [6]. In theory, this method can perfectly track the index, but when there are too many index constituents, the cost of purchasing all assets will be too high. It also entails high management cost, which is generally not feasible in practice. The other strategy is the incomplete replication method, i.e., purchasing some assets in the target index to track the index [2], [3]. Although this method has certain errors, it can greatly reduce the input cost. In practice, the question how to select the assets from the target index to reduce tracking error (TE) arises. However, feature selection analysis is an important approach to solve this problem [7], [8].

According to the level of the feature index, we can divide the discussion of feature selection into three cases [9], [10]: fixed dimension, divergent dimension, and high dimension. The dimensionality directly affects the selection of feature selection methods. Therefore, research on feature selection with different dimensions has important theoretical significance and application value.

In the face of a large quantity of explanatory variables used to describe sample characteristics, data dimensionality reduction can help us build a target feature selection model which is easier to interpret and has better generalization capabilities. Currently, two widely employed techniques for reducing data dimensionality are feature extraction and feature selection [11]. Feature extraction involves mapping the original high-dimensional feature space to a lower-dimensional space. A typical method is principal component analysis. However, principal component analysis does not consider the relationship between independent and dependent variables in the process of data dimensionality reduction. Also, when there are too many explanatory variables, the extracted principal components are usually difficult to interpret [12]. Unlike feature extraction, feature selection directly chooses a subset of features from the original feature space. Notably, representative methods for feature selection encompass approaches from evolutionary computation [13], [14]. Evolutionary computation methods often have many parameters, such as the crossover and mutation rates and the population size. Their optimal configuration is often difficult to determine.

The GMDH-NN, an automated model, excels in determining variables, structure, and parameters [15]. Moreover, the selected features offer excellent interpretability, compensating for principal component analysis limitations. Its success spans diverse fields like economy, engineering, and

others [16], [17]. Notably, the GMDH-NN's potential in high-dimensional index tracking remains underexplored. Originally proposed in 1967 by academician A. G. Ivakhnenko of the Ukrainian Academy of Sciences [18], [19], this heuristic data mining algorithm became a milestone in self-organizing data mining theory. In the 1990s, German scholar Mueller and software expert Frank further developed the theory and algorithm in the Software Knowledge Miner [20]. Recently, the GMDH-NN gained popularity in various applications. For instance, Xiao et al. proposed a GMDH-NN based semi-supervised feature selection for customer classification [16]. Mo et al. developed a GMDH-NN based hybrid model for container throughput forecasting, leveraging selective combination forecasting in nonlinear subseries [15]. Jeddi and Sharifian introduced a GMDH-NN ensemble model for network function virtualization workload forecasting in cloud computing [17], and many others.

B. CONTRIBUTIONS

The CSI 300 index, released jointly by the Shanghai and Shenzhen stock exchange on April 8, 2005 [2], [3], serves as a valuable financial tool. It effectively portrays the CSI 300 index compilation target and operating status, serving as a reliable evaluation criterion for investment performance. Moreover, it establishes the foundation for index-based investment and fosters innovation in index derivative products. In our research, we leverage this index as a fundamental pillar and proudly present the following significant contributions:

(1) Our data is derived from the time series dataset which includes the closing prices of the CSI 300 index. Stock index is susceptible to many dynamic factors, with such characteristics as high dimension, uncertainty, non-linearity, time delay, complexity, etc., resulting in abnormal and missing values in stock index data, which will lead to instability or unreliability of the stock index tracking model. To address these challenges, we present a novel LSTM network-based stock index data preprocessing model incorporating the attention mechanism. This innovative approach optimizes the initial stock index data, transforming it into a more coherent, comprehensive, and sequential dataset, enabling effective learning across time series.

(2) GMDH-NN establishes a general relationship between input and output variables by employing a reference function. To this end, we adopt a discrete form of the first-order Kolmogorov-Gabor ($K - G$) polynomial as the reference function, leading us to develop the concept of the deep evolutionary GMDH polynomial neural network (DGMDH). This innovative approach incorporates an external criterion system, allowing for the selection of diverse external criteria based on specific modeling objectives. Moreover, it enables the creation of novel external criteria as necessary. For high dimensional stock index data, the traditional external criteria can no longer meet the needs of practical applications. Here,

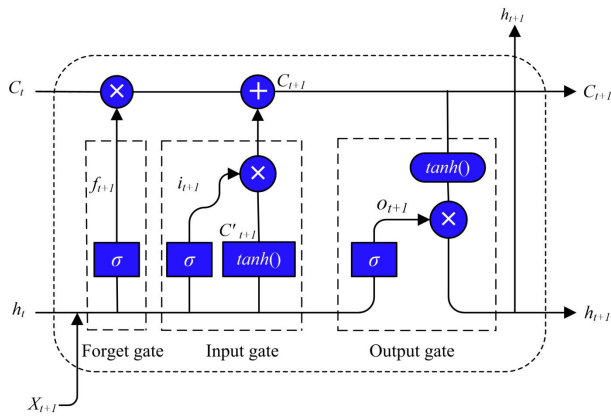


FIGURE 1. LSTM structure diagram.

we propose a TEEC for the stock index, which is based on the difference between allocation yield and target yield.

(3) The DGMDH is a neural network that follows a heuristic self-organizing principle. It employs an evolutionary computing technique to determine input variables and model parameters through a series of operations involving seeding, rearing, crossbreeding, selection, and rejection of seeds. We apply DGMDH to perform feature selection on high-dimensional stock index data and create an evolutionary stock index tracking model. Lastly, we utilize the Copeland scoring sorting technique [21] and random weighted Copeland scoring sorting to analyze and validate the learning capacity of our methodology.

C. PAPER ORGANIZATION

The outline of this paper is as follows. Section II introduces the fundamentals, including LSTM principle and GMDH-NN. Sections III describes a high-dimensional index tracking model, which involves LSTM data preprocessing model, TEEC and high-dimensional index tracking based on DGMDH, respectively. Sections IV gives an example of index tracking based on DGMDH in detail. Section V presents the experimental setup and dataset. Section VI shows the experiment results and analysis in detail. Section VII provides the conclusions.

II. FUNDAMENTALS

A. LSTM PRINCIPLE

Compared to general neural networks, Recurrent Neural Network (RNN) excels in handling sequence-changing data and possesses memory capabilities. However, conventional RNNs suffer from certain limitations such as gradient explosion, gradient disappearance, and inadequate handling of long-term dependencies [18]. These drawbacks can be effectively addressed by long-term and short-term neural networks (LSTM). LSTM consists of several modules, including an input gate, output gate, forget gate, and a memory cell (cell unit). This architecture enables efficient processing of time series information, as depicted in Figure 1.

Regarding the length of memory, what kind of information should be forgotten, and what kind of information should be remembered, the LSTM network can perfectly solve these problems through the structures of forget gate, input gate, and output gate etc..

B. GMDH-TYPE NEURAL NETWORK

GMDH-NN, initially proposed by Ivakhenko [18], [19], is a powerful technology in self-organizing data mining. It excels at autonomously organizing variables, structures, and parameters within the model. Over the years, GMDH-NN has found widespread applications in diverse fields such as engineering, science, and economic research [17], [22]. In this section, we will outline the construction process for the initial model of DGMDH.

To establish the general relationship between input and output variables, GMDH-NN utilizes a reference function. Typically, the discrete form of the $K - G$ polynomial is adopted as this reference function [19], [23]:

$$\begin{aligned}
 Y &= f(X_1, X_2, \dots, X_M) \\
 &= a_0 + \sum_{i=1}^M a_i X_i + \sum_{i=1}^M \sum_{j=1}^M a_{ij} X_i X_j \\
 &\quad + \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M a_{ijk} X_i X_j X_k + \dots, \tag{1}
 \end{aligned}$$

Here, the output is represented as Y , while the input vector is denoted as $X = (X_1, X_2, \dots, X_M)$. The coefficient or weight vector is represented by a . A first-order linear $K - G$ polynomial, encompassing n variables, takes the following form:

$$Y = f(X_1, X_2, \dots, X_n) = a_0 + a_1 X_1 + a_2 X_2 + \dots + a_n X_n. \tag{2}$$

All sub-items are considered as the n initial models in the modeling network structure.

$$v_1 = a_1 x_1, v_2 = a_2 x_2, \dots, v_n = a_n x_n. \tag{3}$$

Equation 3 combines n initial models in pairs to generate $C_n^2 = n(n - 1)/2$ intermediate candidate models for the first layer. The transfer function is as follows:

$$w = f(v_i, v_j); i, j = 1, 2, \dots, n; i \neq j. \tag{4}$$

In Equation 4, w represents the estimated output.

The external criteria values of intermediate candidate models are calculated. Based on their order, the best models are selected and combined to generate new candidates for the next layer. This process is repeated to obtain intermediate candidate models for subsequent layers. The termination rules are determined by the principle of optimal complexity [23]. As the complexity of the model increases, the external criterion value initially decreases and then increases. The model with the minimum external criterion value represents the optimal complexity model.

III. HIGH-DIMENSIONAL INDEX TRACKING

A. LSTM DATA PREPROCESSING MODEL

In practical scenarios, data may be unavailable or missing [24]. If the proportion of missing records is small, complete records can be processed directly while discarding the missing ones. However, in reality, a significant portion of the data is often missing, especially in multivariate data. Deleting cases becomes inefficient as it leads to the loss of valuable information and introduces bias. This bias results in systematic differences between incomplete and complete observation data. Consequently, it is crucial to address the issue of missing data.

In multivariate time series, missing data is very ubiquitous [25]. There are three kinds of methods available to deal with the missing value of time series: The first is the already-mentioned direct deletion method, which may discard some important information in the data. The second filling method is based on statistics, such as mean filling, median filling, and common value filling, but this method ignores the time series information. The third filling method is based on machine learning, like k-nearest neighbors, recurrent neural networks, or expectation-maximization. However, this method again rarely considers the temporal information between two adjacent data. In view of that, we use the LSTM time series data preprocessing model to make up for the missing data [26]. Meanwhile, it can optimize the original stock index data into continuous, complete, and more sequential stock index data, as well as finish the learning task from time series to time series.

The LSTM time series data preprocessing model calculates the average of a certain number of items along the time axis in a progressive manner. It helps mitigate the impact of periodic and irregular changes, enabling the discovery of the underlying development trend. By eliminating fluctuations, this model facilitates the analysis and prediction of the series' long-term trend [27], [28].

This paper proposes an LSTM model with an attention mechanism to preprocess stock index data, aiming to reduce the influence of local minimization and improve the accuracy of index tracking in the GMDH model. The LSTM model with attention mechanism, depicted in Figure 2, consists of an input layer, LSTM layer, Attention layer, and output layer. By utilizing the LSTM layer and Attention layer, the historical stock index data are processed to achieve prediction functionality. The resulting output layer provides stock indices with enhanced continuity, completeness, and time series representation.

In Figure 2, the input sequence is represented by x_1, x_2, \dots, x_n , which consists of stock index data for a period of time. These inputs are transmitted to the LSTM unit to generate the corresponding hidden layer outputs, h_1, h_2, \dots, h_i . The hidden layer incorporates an attention mechanism to compute the attention probability distribution values, $\alpha_1, \alpha_2, \dots, \alpha_i$, for each input. The model's layers are described as follows:

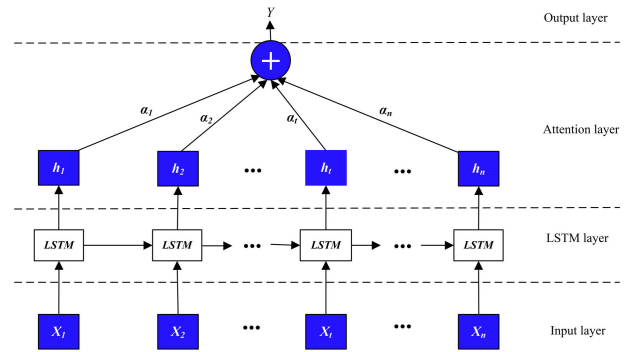


FIGURE 2. LSTM structure diagram with attention mechanism.

(1) Input layer. It treats historical stock index data as the input of the prediction model, represented by $X = [x_1 \cdots x_{t-1}, x_t \cdots x_n]^T$.

(2) LSTM layer. It learns the feature vector of the input layer. The number of historical days is set to n , and the step length m is set to 3, that is, based on the stock index data from the first day to the third day, the LSTM is used to predict the stock index on the fourth day; based on the stock index data from the second day to the fourth day, the LSTM is used to predict the stock index on the fifth day, and so on. Finally, n new stock index data are obtained. We construct the LSTM structure to comprehensively grasp the extracted features and internal change rules. The output of the LSTM layer is referred to as h , and the output at step t is denoted as:

$$y_t = LSTM(H_{C,t-1}, H_{C,t}).$$

(3) Attention layer. The input to the attention mechanism layer is the activated output vector h from the LSTM network layer. The probability of different feature vectors is calculated based on the weight distribution principle, continuously updated and iterated for better weight parameters. The calculation formula for the weight coefficient of the Attention mechanism layer can be expressed as:

$$e_t = \text{utanh} \sum (wh_t + b), \quad (5)$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{j=1}^t (e_j)}, \quad (6)$$

$$s_t = \sum_{i=1}^i (\alpha_i h_i). \quad (7)$$

In the formula, e_t represents the attention probability distribution value determined by the LSTM network layer's output vector h_t at time t . The weight coefficients are represented by u and w , while b denotes the bias coefficient. The output of the attention layer at time t is represented as s_t . This paper utilizes the mean value of the characteristics within the t -th step of the stock index sequence as the input e_t in formula (6).

(4) Output layer. The output layer in this case takes the input from the attention mechanism layer. It is a fully connected layer that calculates the output $Y = [y_1, y_2, \dots, y_n]^T$

using a prediction step of m . The prediction formula for this can be expressed as:

$$y_t = \text{Sigmoid}(w_o, b_o).$$

In the formula, the predicted output value at time t is represented by y_t . The weight matrix is denoted as w_o , while the deviation vector is represented by b_o . Additionally, the activation function chosen for the Dense layer in this study is the Sigmoid function.

B. TRACKING ERROR EXTERNAL CRITERION

When our DGMDH is applied to modeling in different fields, the selection of external criteria plays the key role. The right mathematical descriptions need to be constructed according to the modeling purpose [15], [23]. In the traditional GMDH-NN model, the most commonly used external criterion is the regularization criterion, which focuses on the errors in the established model test set.

In the field of finance, the TE is usually used to measure the difference between allocation yield and target yield [2], [3]. Let \hat{y}_i be the estimated or predicted value of the CSI 300 index y_i , with $i = 1, \dots, T$, and $error_i = \hat{y}_i - y_i$. Then we can define the external criterion TEEC of DGMDH as follows:

$$TE = \sqrt{\frac{\sum (error_i - error)^2}{T - 1}}, \quad (8)$$

here, T represents the largest number of stocks and $error$ is the mean of all the $error_i$ values.

The DGMDH algorithm is used to construct a high-dimensional index tracking model, forming a multilayer network structure. Starting from the initial model input layer, the complexity of the model increases. The stopping rule is determined by the principle of optimal complexity: as the model complexity gradually increases, the external criterion value initially decreases and then increases. The optimal complexity model corresponds to the minimum value of the external criterion. The DGMDH algorithm stops when the external criterion value can no longer be improved, ensuring an optimal balance between data fitting accuracy and prediction ability.

C. DGMDH

The DGMDH is a method of self-organizing data mining for complex nonlinear systems. It is a combined method of data processing based on $K - G$ polynomials to identify non-linear systems through continuous screening and combination. The basic principle of the DGMDH algorithm is that a series of active neurons are generated by cross combination in pairs of each input unit of the system. The best transfer function for each neuron is determined by selecting the internal criteria. Then the generated neurons are screened by selecting the external criteria. The screened neurons are again combined in pairs to generate new neurons. This is repeated until the new neurons are no better than the previous generation. Then

the optimal complex model is produced. By finding the best balance point between the fitting accuracy of the sample and the prediction accuracy of the new data set, the algorithm can reflect the real internal relationship of the system to the greatest extent even when the sample data is small or the data noise is large. The layer by layer selection of the model structure and variables in the modeling process ensures the convergence speed of the calculation, which also significantly reduces the impact of the subjective factors.

Suppose the input data consists of N samples with n attributes v_1, v_2, \dots, v_n and a label Y . The training set, T_r , contains m samples, while the test set is denoted as T_e . The initial input layer of DGMDH is obtained, and new candidate models are generated by combining pairs of models from the previous layer. Below are the basic steps of feature selection and high-dimensional index tracking model using DGMDH [15], [23]:

Step 1 Randomly divide the sample data into training and testing sets.

Step 2 Generate the initial model, which is a linear $K - G$ polynomial: $f(x_1, x_2, \dots, x_n) = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n$. For DGMDH multilayer neural network modeling, use all sub-terms of the $K - G$ polynomials as the initial input model of the network: $v_0 = a_0, v_1 = a_1x_1, \dots, v_n = a_nx_n$.

Step 3 Obtain the first intermediate model by combining the initial models in pairs, using the linear reference function $y = f(v_i, v_j) = a_0 + a_iv_i + a_jv_j$.

Step 4 Adopt the least squares method to estimate the parameters of the candidate model on the training set.

Step 5 Use external criteria to evaluate the performance of candidate models on the training set. Select the best models as input for the next layer.

Step 6 Repeat steps (3)-(5) to generate the second, third, \dots , n^{th} layer intermediate network. Obtain intermediate candidate models with increased model complexity. The termination rule of the algorithm is the optimal complexity principle [23]. As the model complexity increases gradually, the external criteria, which have a supplementary property, will initially decrease and then increase. The global minimum value of the external criteria corresponds to the high-dimensional index tracking model of optimal complexity.

D. HIGH-DIMENSIONAL INDEX TRACKING BASED ON LSTM-DGMDH NETWORK

Our high-dimensional stock index tracking model, LSTM-DGMDH, combines LSTM and deep evolutionary GMDH-type neural network. Our model mainly includes: (1) A stock index data preprocessing model based on LSTM network with attention mechanism, which optimizes the original data into continuous, complete, and sequential stock index data. (2) We utilize DGMDH to execute feature selection of high-dimensional stock index data and develop an evolutionary stock index tracking model. (3) For a high dimensional

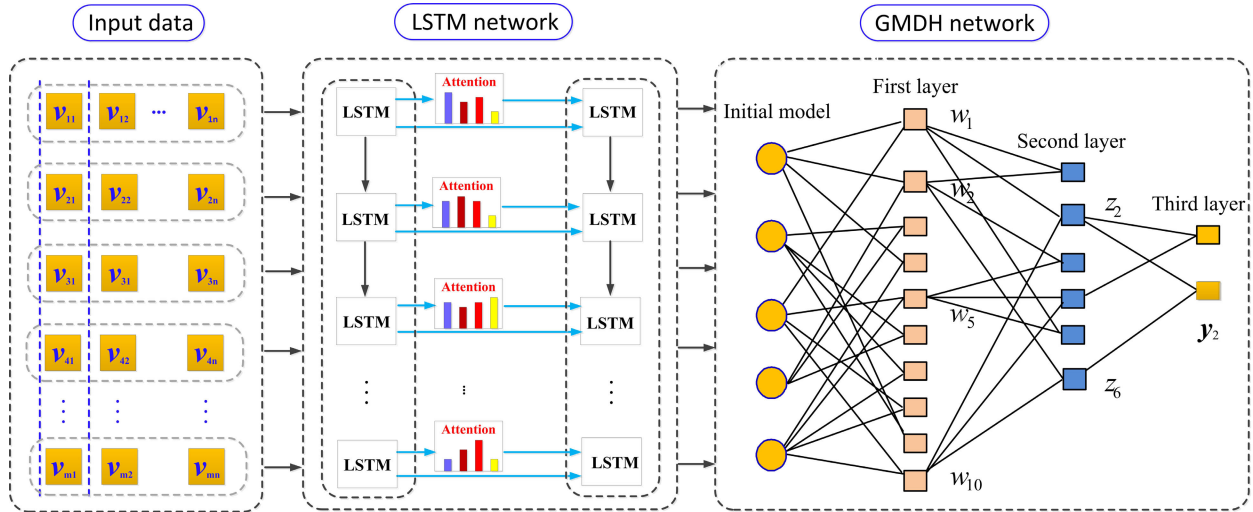


FIGURE 3. Structure diagram of high-dimensional index tracking model.

stock index dataset, the traditional external criteria can no longer meet the needs of reality. Consequently, to better analyze external criterion of DGMDH, TEEC is used to select optimal complex model. The structure diagram of our high-dimensional stock index tracking model is shown in Figure 3.

IV. EXAMPLE OF LSTM-DGMDH FOR INDEX TRACKING

Next, the index tracking algorithm of LSTM-DGMDH for high-dimensional financial data is shown in Algorithm 1. Let us consider a concrete example of index tracking [23]. Assume that each sample of dataset D after LSTM pretreatment with N samples includes five attributes v_1, v_2, v_3, v_4, v_5 and the sample label Y . Then the procedure of feature selection and high-dimensional index tracking model with LSTM-DGMDH can be illustrated as in the following steps (see Figure 4).

(1) Split dataset D equally and horizontally into training set A and testing set B .

Algorithm 1 Algorithm Description

Input: Dataset T (closing prices or yield rates after LSTM pretreatment)

Output: The optimal model Y_{opt}

- 1: Utilize closing prices or yield rates of stocks v_i ($i = 1, 2, \dots, N$) to get initial input features of DGMDH;
- 2: Combine pairs of initial features to gain C_N^2 medium alternative models in first layer;
- 3: Calculate the external criterion value TE for each output y_i^l according to Equation (8) in i -th layer;
- 4: Get the minimum value of TEEC TE_{min} in i -th layer;
- 5: Repeat steps 3-4 with $l = l + 1$, and if $TE_{min} \geq TE$, then STOP, else set $TE = TE_{min}$ and CONTINUE;
- 6: **Return** Y_{opt} .

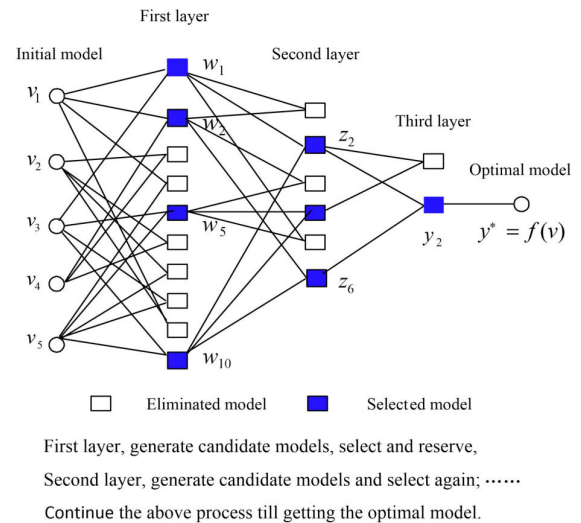


FIGURE 4. Structure diagram of GMDH-type neural network.

(2) Build a degree 1 polynomial, $K - G$, to represent the relationship between dependent variables Y and initial independent variables $v_1, v_2, v_3, v_4, v_5 : Y = f(v_1, v_2, v_3, v_4, v_5)$ in the input layer of DGMDH (Figure 4 (Initial model)).

(3) Combine pairs of initial models to generate novel potential models y_i ($i = 1, 2, \dots, C_N^2$) in the first layer, such as $y_1 = V_1 + V_2 = a_1v_1 + a_2v_2, y_2 = V_1 + V_3 = a_1v_1 + a_3v_3$ (model parameters calculated in set A).

(4) Select the TEEC as the external criterion for neurons selection. The neurons of each layer is evaluated by the TEEC. The selected neurons are retained as input of the next layer. The unselected neurons will be eliminated (Figure 4 (First layer)).

(5) Generate the local model of the first layer by defining the transfer function $w_k = f_k(v_i, v_j)$ with $k = 1, 2, \dots, 10$.

The parameters of w_k are estimated using the training set A . These models are then evaluated on the testing set B based on an external criterion, and the models w_k ($k = 1, 2, 5, 10$) are selected as input variables for the next layer. The selected models from the initial layer are combined to create potential models for the second layer, for example, $z_2 = b_1w_1 + b_{10}w_{10}$ (refer to Figure 4 (Second layer)).

(6) Calculate the TEEC value for each model in the second layer using the model selecting set A . Choose the model with the minimum external criterion as the input for the third layer (Figure 4 (Third layer)).

(7) Repeat the process described in (5) and (6) to obtain the TEEC value in the third layer. According to the theoretical framework of optimal complexity, the smallest TEEC value corresponds to the optimal complexity model, and subsequently, we find $y_2 = Y_{opt} = f(v)$ (Figure 4 (Optimal layer)).

(8) Once Y_{opt} is identified, search for the initial models included in Y_{opt} . In this example, it includes four initial models: v_1, v_3, v_4, v_5 . In other words, only these four features (v_1, v_3, v_4, v_5) are part of the optimal complexity selection.

(9) Take v_1, v_3, v_4, v_5 into Y_{opt} , to obtain the ultimate high-dimensional index tracking model for test set B .

V. EXPERIMENTAL SETUP AND DATASET

In the presented experiments, we utilize DGMDH for feature selection and high-dimensional index tracking of stock index data. The DGMDH's potential for high-dimensional index tracking, performance, and stability will be explored and evaluated.

CSI 300 index datasets are employed for testing our DGMDH. The experimental evaluation comprises two parts. Firstly, we compare our method against five commonly used index tracking techniques: nonnegative adaptive elastic-net (NAEN) [2], nonnegative minimax concave penalty estimator (NMCPE) [3], multiple spline regression (MSR) [29], Gaussian processes regression (GPR) [30], and index tracking with cardinality constraints (ITCC) [6]. Secondly, we assess the effectiveness of our approach through statistical analysis. To mitigate random effects, a 5-fold cross-validation is conducted, with four folds for training and one for testing. Ten independent runs of all experiments are performed, and the average results are reported. The execution platform consists of a PC with a 3.60 GHz Intel(R) Core(TM) i7-7700M CPU, 8GB RAM, and Microsoft Windows 10 operating system.

Our datasets include the closing prices and yield rates of the CSI 300 index and all its constituent stocks from July 1 to November 30, 2021. Notably, the constituents of the index remained unchanged during this period [2], [3]. Moreover, our datasets are comprised of the following three aspects:

(1) The data characteristics of some stocks are missing within a period of time or within a few days. Since these data contain time characteristics, we exploit LSTM to fill in these missing data. The dataset for our experiments contains a total of 102 observations (corresponding to the 102 work days in the interval) and 300 covariates. The data is divided

into two parts: 80% of the observations are allocated for training, while the remaining 20% are reserved for testing. Consequently, the resulting model is highly dimensional due to its training on this dataset.

(2) In addition to the closing prices, we also consider the yield rates of stocks to generate new datasets. Meanwhile, it should be noted that the CSI 300 index adjusts its constituents semi-annually, so our data set contains data in half a year. This is completely different from the previous approaches, which only consider data across the years.

(3) In order to evaluate our model fully, we usually select a small subset (such as half of the total samples) of all constituent stocks to track the target index. In this paper, we generate four datasets from the CSI 300 index. (i) with 102 observations and 300 covariates named ("Clopri102"); (ii) 51 observations with closing prices (Observations on odd days were selected by averaging the interval of one day, this is abbreviated as "Clopri51"); (iii) 102 observations with yield rates ("Yierat102"); (iv) 51 observations with yield rates ("Yierat51").

To understand these datasets more clearly, we show the statistical distribution chart for these datasets. In Figure 5, we showcase the distribution of closing prices or yield rates of first nine stocks. In first line subgraphs, the horizontal axes represent the date. The y-coordinate is the closing prices of Clopri102 and Clopri51, respectively. In the second line of subgraphs, we also set the x-coordinate as the date. The y-coordinate represents the yield rates of Yierat102 and Yierat51, respectively.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

A. ERROR ESTIMATE

In this study, three different model performance evaluation indexes are selected, namely the root mean square error (RMSE), the absolute mean percent error (MAPE), and the relative square root error (RRSE).

$$RMSE = \sqrt{\sum_{i=1}^m (\hat{y}_i - y_i)^2 / m},$$

$$MAPE = \sum_{i=1}^m \left| \frac{\hat{y}_i - y_i}{y_i} \right| / m,$$

$$RRSE = \sqrt{\sum_{i=1}^m (\hat{y}_i - y_i)^2 / \sum_{i=1}^m (y_i - \bar{y})^2}.$$

Here, y_i represents the actual value of the i -th instance, while \hat{y}_i is its corresponding predicted value. The test set size is denoted as m , and \bar{y} denotes the average value of y_i across the m test instances. A smaller evaluation index indicates better prediction performance for the model.

Additionally, in finance fields, TE given in Equation 8 is often measured as the variance of the difference between portfolio return and index return. The error estimate (rank) results of the six methods on four datasets can be found in Table 1.

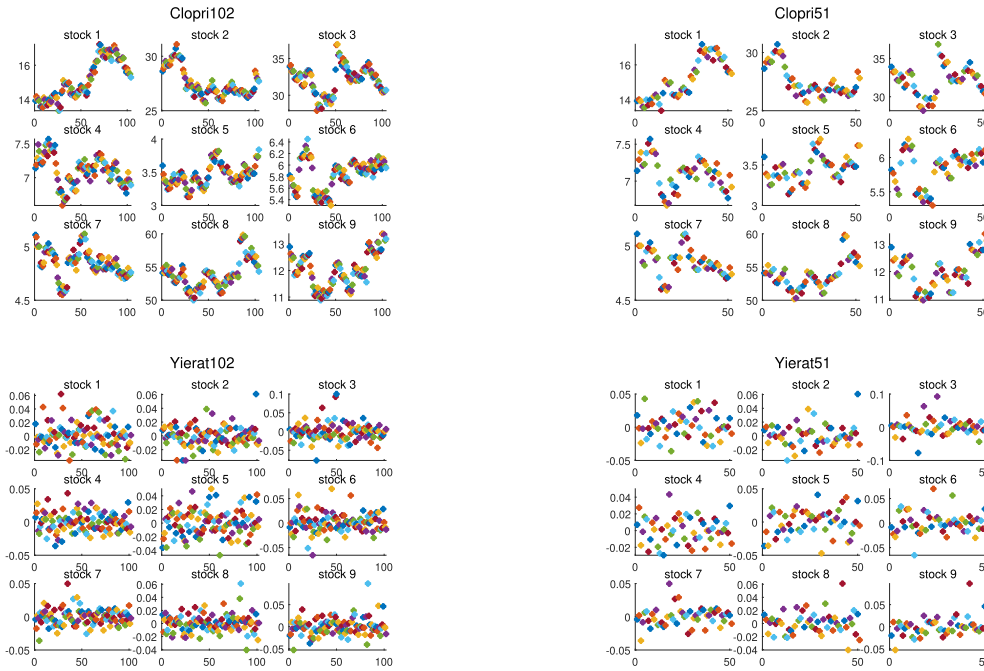


FIGURE 5. Visualization of Clopri102, Clopri51, Yierat102 and Yierat51 datasets, respectively.

TABLE 1. Error estimate (rank) on CSI 300 index. For the RMSE, MAPE, RRSE and TE, the evaluation results of six different methods are shown in lines 1-7, lines 8-14, lines 15-21 and lines 22-28 on four datasets, respectively.

RMSE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
Clopri102	25.768(4)	17.922(1)	64.650(5)	67.974(6)	23.071(3)	18.068(2)
Clopri51	25.910(4)	17.427(2)	39.361(5)	75.289(6)	24.202(3)	16.949(1)
Yierat102	0.002(1)	0.003(3)	0.004(4)	0.008(6)	0.005(5)	0.002(1)
Yierat51	0.001(1)	0.002(3)	0.003(4)	0.009(6)	0.005(5)	0.001(1)
Total error	51.861	35.354	104.018	143.28	47.283	35.02
Total rank	10	9	18	24	16	5
MAPE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
Clopri102	0.004(3)	0.003(1)	0.012(5)	0.014(6)	0.006(4)	0.003(1)
Clopri51	0.005(3)	0.003(1)	0.005(3)	0.014(6)	0.007(5)	0.004(2)
Yierat102	0.487(3)	0.503(4)	0.053(1)	0.995(6)	0.704(5)	0.428(2)
Yierat51	0.473(3)	0.604(5)	0.045(1)	1.063(6)	0.562(4)	0.434(2)
Total error	0.969	1.113	0.115	2.086	1.279	0.869
Total rank	12	11	11	24	18	7
RRSE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
Clopri102	0.575(4)	0.412(3)	0.721(5)	1.155(6)	0.308(2)	0.220(1)
Clopri51	0.531(4)	0.357(3)	0.705(5)	1.500(6)	0.294(2)	0.235(1)
Yierat102	0.284(2)	0.390(4)	0.293(3)	1.050(6)	0.412(5)	0.283(1)
Yierat51	0.205(1)	0.263(3)	0.298(4)	1.069(6)	0.419(5)	0.221(2)
Total error	1.595	1.422	2.017	4.774	1.433	0.959
Total rank	11	13	17	24	14	5
TE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
Clopri102	27.162(4)	18.891(2)	66.678(6)	62.014(5)	21.705(3)	17.724(1)
Clopri51	26.583(4)	17.880(2)	40.458(5)	52.900(6)	19.073(3)	15.043(1)
Yierat102	0.002(1)	0.003(3)	0.004(5)	0.008(6)	0.003(3)	0.002(1)
Yierat51	0.001(1)	0.002(3)	0.005(5)	0.008(6)	0.004(4)	0.001(1)
Total error	53.748	36.776	107.145	114.93	40.785	32.77
Total rank	10	10	21	23	13	4

We find that on the Clopri102 dataset in terms of RMSE, our DGMDH is almost as good as NMCPE, and outperforms the other regression approaches including NAEN, MSR, GPR and ITCC. For the Clopri51 dataset, DGMDH obtains the

lowest RMSE result of all methods. On the Yierat102 dataset, the NAEN and our DGMDH perform the same and better than the other regression approaches. On the Yierat51 dataset, our DGMDH is again the best. In a word, in terms of RMSE, DGMDH performs similar to NMCPE. MSR and GPR are relatively poor on our four datasets. Our method also has the smallest total error sum and total rank sum. In terms of the MAPE, our methodology and NMCPE are again the best on the Clopri102 dataset and outperform the other approaches. NAEN and ITCC are also good. For the Clopri51 dataset, our DGMDH is slightly worse than NMCPE but outperforms the other methods. GPR is the worst possible method. For the Yierat51 and Yierat102 dataset, the MSR is the best and our method is second. For the dataset, the results of the regression analysis are somewhat similar to the Yierat102 dataset above. DGMDH comes the second, and is superior to the other approaches including NAEN, NMCPE, GPR and ITCC. GPR is the worst. For the total error, The MSR clearly has the minimum error in all regression approaches. Our DGMDH resembles the NAEN, both outperform all of the rest. For the total rank, it distinctly shows that our DGMDH is superior to any other regression method.

In the RRSE of Table 1, for the DGMDH methodology, it is the best and outperforms the other approaches including NAEN, NMCPE, MSR, GPR and ITCC regression approaches on Clopri102, Clopri51 and Yierat102 datasets. For the Yierat51 dataset, the result of the regression analysis of NAEN is the best, and somewhat similar to the Yierat102 dataset above. DGMDH comes second, which is superior to other approaches including NAEN, NMCPE, GPR and ITCC. The NMCPE and MSR methods are comparable

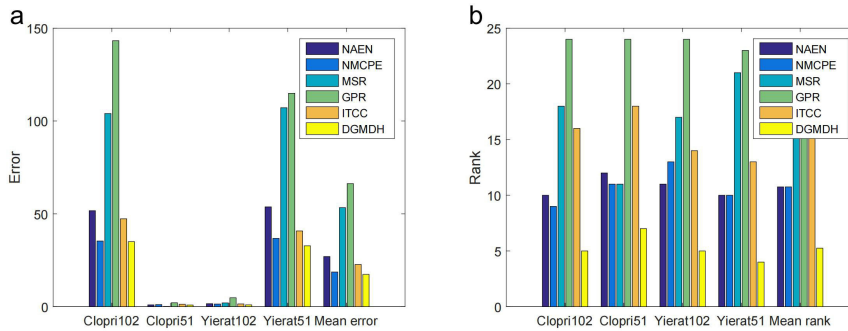


FIGURE 6. The results of error estimate (rank) on CSI 300 index. The error estimate and rank value of six different methods are shown in Figure 6 (a) and (b), respectively.

to our DGMDH, while GPR is the worst approach in our experiments. In terms of both the total error and total rank, it evidently shows that our DGMDH outperforms the other regression approaches.

The TE in Table 1 illustrates that DGMDH gets better classification results for all of the datasets employed in the experiments for most of the datasets used. In particular, our method is obviously superior to NAEN, MSR, GPR and ITCC on Clopri102 and Clopri51 datasets. Moreover, on Yierat102 and Yierat51 datasets, our DGMDH methodology is similar to the NAEN method, which are superior to the other approaches studied in the paper. The second best approach is NMCPE, and the worst is GPR, while the rest are strikingly similar. With respect to both total error and total rank, it evidently shows DGMDH and NAEN have the same level of TE. They are slightly lower than other methods. The NMCPE and ITCC methods are similar, and the GPR method has the biggest TE.

B. COMPARISON OF TOTAL ERROR AND RANK

In Table 2 and Figure 6, for the total results and mean result of error estimate, we can see that our method is equal to NMCPE, which represent the best level of the regression analysis algorithms. The MSR and GPR are relatively poor. In terms of the total results and mean result of rank value, it is clear that our DGMDH methodology is superior to all of the others. It is worth mentioning that the NAEN and NMCPE are strikingly similar. The GPR proves to be the worst in our experiments.

C. STATISTICAL ANALYSIS

The general comprehensive evaluation methods can be divided into two types: absolute value and ranking value. The absolute value directly represents the situation of the evaluated objects, while the ranking value represents their priority order. Correspondingly, the combination evaluation methods are also classified based on their forms: The first type uses the absolute numerical form of evaluation results for calculation and combination. For instance, the least square method is used as the combination evaluation method. The second type uses the ranking type of evaluation results as

TABLE 2. For the RMSE, MAPE, RRSE and TE, the total and mean results of error estimates (ranks) on six different methods are shown. The total results and mean results of error estimate of six different methods are shown in line 2-5 and line 6, respectively. The total results and mean results of rank value of six different methods are shown in line 8-11 and line 12, respectively.

Method	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
RMSE	51.681	35.354	104.018	143.280	47.285	35.020
MAPE	0.969	1.113	0.115	2.086	1.279	0.869
RRSE	1.595	1.422	2.017	4.774	1.433	0.959
TE	53.748	36.776	107.145	114.930	40.785	32.770
Mean error	26.998	18.666	53.324	66.268	22.696	17.405
Method	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
RMSE	10	9	18	24	16	5
MAPE	12	11	11	24	18	7
RRSE	11	13	17	24	14	5
TE	10	10	21	23	13	4
Mean rank	10.750	10.750	16.750	23.750	15.250	5.250

the basis for calculation. The Copeland method, used in this paper, belongs to this type.

1) COPELAND SCORING SORTING METHOD

The Copeland Scoring Sorting Method is a combination evaluation approach that assigns scores based on a program’s merits. It was introduced by A.H. Copeland from the University of Michigan during a mathematics seminar. This method emphasizes the principle that the minority is subordinate to the majority [31]. It is parameter-free and uses global comparisons instead of simple scores. Due to its simplicity and effectiveness, it finds extensive application in elections.

Let’s consider a scenario with m evaluation objects, denoted as X_i ($i = 1, 2, \dots, m$), each having p evaluation indexes. The value of the j th evaluation index of the i th evaluation object is represented as a_{ij} .

Suppose there are m evaluation objects, where X_i ($i = 1, 2, \dots, m$) represents the i th evaluation object, and each evaluation object has p evaluation indexes, where a_{ij} ($i = 1, 2, \dots, m$) represents the j th evaluation index value of the i th evaluation object.

The evaluation index values a_{ij} ($j = 1, 2, \dots, p$) of the evaluation object X_i ($i = 1, 2, \dots, m$) will be compared with the evaluation index values a_{kj} ($j = 1, 2, \dots, p$) of another

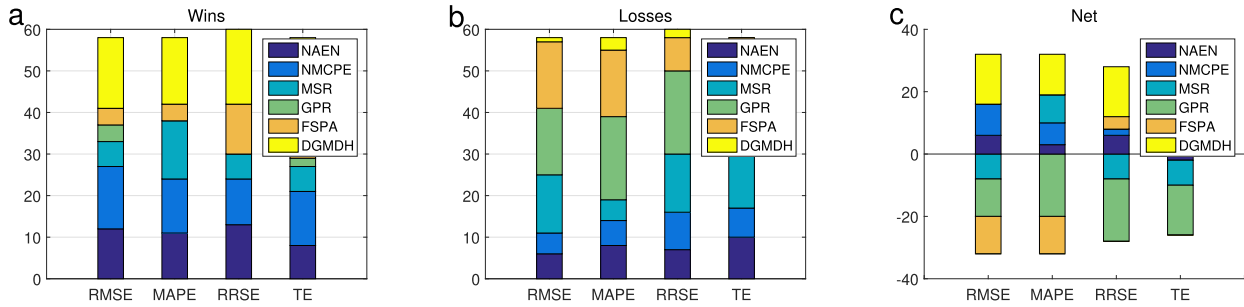


FIGURE 7. The stacking histograms of Copeland scoring sorting method for the total of Wins/Losses/Net are shown in Figure 7 (a) , (b) and (c), respectively.

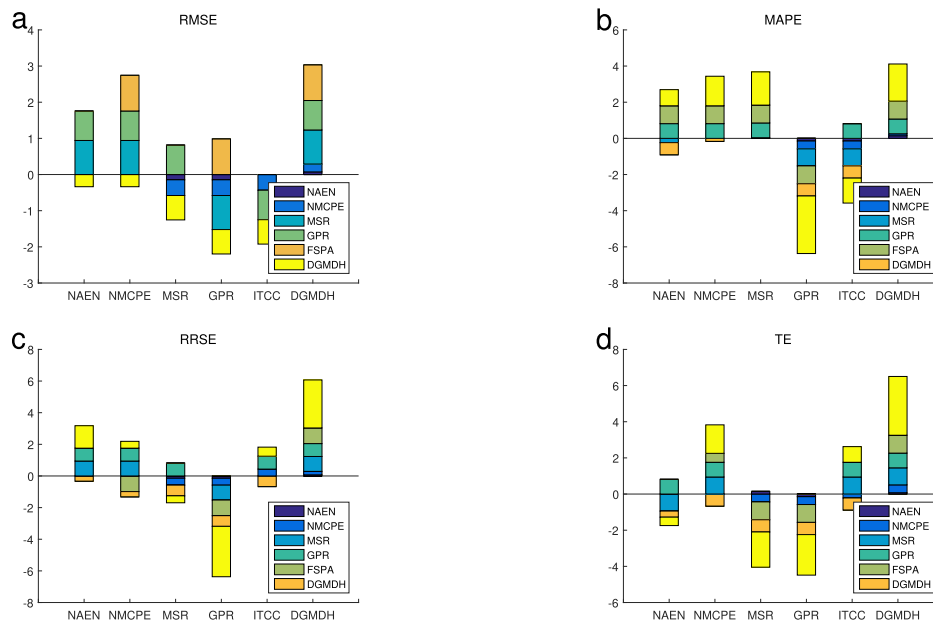


FIGURE 8. The histograms of the six algorithms using the random weighted Copeland scoring sorting method are depicted in Figure 8 (a) , (b), (c), and (d) respectively.

evaluation object X_k ($k = 1, 2, \dots, m$).

$$y_j = \begin{cases} +1 & a_{ij} > a_{kj} \\ 0 & a_{ij} = a_{kj} \\ -1 & a_{ij} < a_{kj} \end{cases} \quad j = 1, 2, \dots, p. \quad (9)$$

where y_j represents the score of the evaluation object X_i when compared with the j -th evaluation index value of the evaluation object X_k , then $\sum_{j=1}^p y_j$ is the comprehensive score of the evaluation object X_i when compared with all indexes of evaluation object X_i , thus the comparison between the two evaluation objects is completed, and then according to this method, the scores of X_i in comparison to X_1, X_2, \dots, X_m are calculated respectively, and m scores can be obtained. The sum of these m scores is the final score of evaluation object X_i . Similarly, calculate the final scores for the remaining $m - 1$ evaluation objects. Higher scores indicate more serious ecological harm, while lower scores indicate less harm. Finally, rank the evaluation objects based on their scores.

Table 3 presents the results of the Copeland scoring sorting method for six different methods. Additionally, Figure 7 (a),

(b), and (c) display the stacking histograms of the Copeland scoring sorting method for Wins/Losses/Net. In order to compare the performances of different error evaluation methodologies in pairs, we assign an index value of +1 (Wins) to a higher score, -1 (Losses) to a lower score, and 0 if the scores are equal. Net represents the difference between Wins and Losses. Based on the information in Table 3 and Figure 7, it is evident that our DGMDH method achieves the highest score, surpassing other state-of-the-art regression analysis methods. The second score is the NMCPE methodology in the evaluation of RMSE and TE, while the NAEN is the second in RMSE, and the GPR methodology is the worst in all datasets.

2) RANDOM WEIGHTED COPELAND SCORING SORTING METHOD

To validate the decision results of the Copeland scoring sorting method, a weighted operation is employed to differentiate the importance and authority of each expert. The weight w_k is assigned to the k -th expert, enabling the definition of the Copeland scoring sorting method for the evaluation object

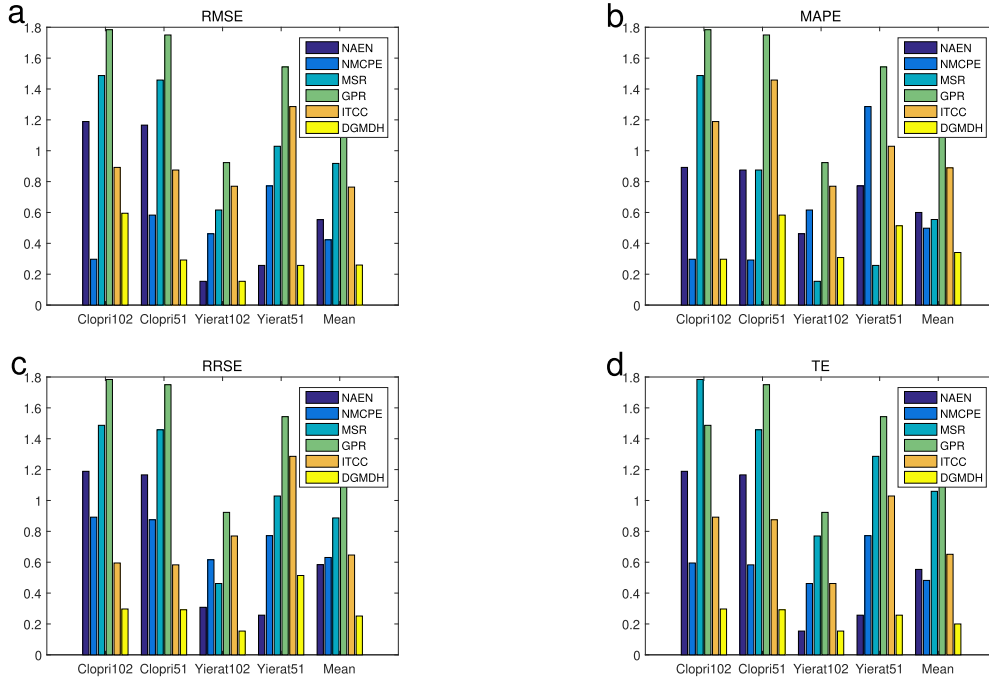


FIGURE 9. The histograms of random weighted rank sorting of six algorithms are shown in Figure 9 (a) , (b), (c) and (d), respectively.

TABLE 3. Copeland scoring sorting method, Wins/Losses/Net.

RMSE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH	Total
NAEN	0/0/0	2/2/0	4/0/4	4/0/4	2/2/0	0/2/-2	12/6/6
NMCPE	2/2/0	0/0/0	4/0/4	4/0/4	4/0/4	1/3/-2	15/5/10
MSR	0/4/-4	0/4/-4	0/0/0	4/0/4	2/2/0	0/4/-4	6/14/-8
GPR	0/4/-4	0/4/-4	0/4/-4	0/0/0	4/0/4	0/4/-4	4/16/-12
FSPA	2/2/0	0/4/-4	2/2/0	0/4/-4	0/0/0	0/4/-4	4/16/-12
DGMDH	2/0/2	3/1/2	4/0/4	4/0/4	4/0/4	0/0/0	17/1/16
MAPE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH	Total
NAEN	0/0/0	2/2/0	1/2/-1	4/0/4	4/0/4	0/4/-4	11/8/3
NMCPE	2/2/0	0/0/0	2/2/0	4/0/4	4/0/4	1/2/-1	13/6/7
MSR	2/1/1	2/2/0	0/0/0	4/0/4	4/0/4	2/2/0	14/5/9
GPR	0/4/-4	0/4/-4	0/4/-4	0/0/0	0/4/-4	0/4/-4	0/20/-20
ITCC	0/4/-4	0/4/-4	0/4/-4	4/0/4	0/0/0	0/4/-4	4/16/-12
DGMDH	4/0/4	2/1/1	2/2/0	4/0/4	4/0/4	0/0/0	16/3/13
RRSE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH	Total
NAEN	0/0/0	2/2/0	4/0/4	4/0/4	2/2/0	1/3/-2	13/7/6
NMCPE	2/2/0	0/0/0	4/0/4	4/0/4	4/0/4	1/3/-2	11/9/2
MSR	0/4/-4	0/4/-4	0/0/0	4/0/4	2/2/0	0/4/-4	6/14/-8
GPR	0/4/-4	0/4/-4	0/4/-4	0/0/0	0/4/-4	0/4/-4	0/20/-20
ITCC	2/2/0	4/0/4	2/2/0	4/0/4	0/0/0	0/4/-4	12/8/4
DGMDH	3/1/2	3/1/2	4/0/4	4/0/4	4/0/4	0/0/0	18/2/16
TE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH	Total
NAEN	0/0/0	2/2/0	0/4/-4	4/0/4	2/2/0	0/2/-2	8/10/-2
NMCPE	2/2/0	0/0/0	4/0/4	4/0/4	3/1/2	0/4/-4	13/7/6
MSR	4/0/4	0/4/-4	0/0/0	2/2/0	0/4/-4	0/4/-4	6/14/-8
GPR	0/4/-4	0/4/-4	2/2/0	0/0/0	0/4/-4	0/4/-4	2/18/-16
ITCC	2/2/0	1/3/-2	4/0/4	4/0/4	0/0/0	0/4/-4	11/9/2
DGMDH	2/0/2	4/0/4	4/0/4	4/0/4	4/0/4	0/0/0	18/0/18

y_i as follows: $b_i = \sum_{k=1}^m w_k b_{ij}^k$ where $\sum_{k=1}^n w_k = 1$, $0 \leq w_k \leq 1$. Here, the *Net* of Table 3 is only taken into account. Random weights within the range of 0-1 are generated to implement the random weighted Copeland scoring sorting method for the *Net* value. The results are presented in Table 4 and Figure 8, yielding similar conclusions as the

forementioned Copeland scoring sorting method. Clearly, our DGMDH method is the most preferred among all error evaluation methods. Moreover, for the NMCPE regression analysis methods, it is the second level on RMSE, MAPE and Tracking error evaluation methods. However, for the RRSE, the NMCPE method is the third level and the second on the list is NAEN method. For the most error evaluation methods, the regression analysis performance of MSR, GPR and ITCC methods is relatively weak.

3) RANDOM WEIGHTED RANK SORTING

In this section, in order to further verify the decisive results of our method, a weighted operation is used to distinguish the importance and authority of each rank sorting of Table 1 for six methods. Table 5 and Figure 9 present the results of random weighted rank sorting of six algorithms on each dataset. The results indicate that DGMDH outperforms the other regression analysis approaches. More specific observations can be made here. Firstly, in terms of the total random weighted rank sorting, it is clear that DGMDH may obtain the lowest error results than NAEN, NMCPE, MSR, GPR and ITCC approaches studied. What's more, the NAEN and NMCPE are rather similar, and obviously superior to MSR, GPR and ITCC. Secondly, for the RMSE, MAPE and Tracking error, although the NAEN is slightly weaker than the NMCPE, however, the NAEN is obviously superior to the NMCPE in our experiments. Thirdly, the GPR is consistently the worst in all of the error estimate methods. In addition, the MSR and ITCC are obviously similar, they may outcome the

TABLE 4. Random weighted copeland scoring sorting method, Wins/Losses/Net.

RMSE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH	Total
NAEN	0.000	0.000	0.942	0.815	0.000	-0.337	1.420
NMCPE	0.000	0.000	0.942	0.815	0.988	-0.337	2.408
MSR	-0.146	-0.434	0.000	0.815	0.000	-0.675	-0.440
GPR	-0.146	-0.434	-0.942	0.000	0.988	-0.675	-1.210
ITCC	0.000	-0.434	0.000	-0.815	0.000	-0.675	-1.924
DGMDH	0.073	0.217	0.942	0.815	0.988	0.000	3.035
MAPE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH	Total
NAEN	0.000	0.000	-0.236	0.815	0.988	-0.675	0.892
NMCPE	0.000	0.000	0.000	0.815	0.988	-0.169	1.634
MSR	0.037	0.000	0.000	0.815	0.988	0.000	1.839
GPR	-0.146	-0.434	-0.942	0.000	-0.988	-0.675	-3.185
ITCC	-0.146	-0.434	-0.942	0.815	0.000	-0.675	-1.382
DGMDH	0.146	0.109	0.000	0.815	0.988	0.000	2.057
RRSE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH	Total
NAEN	0.000	0.000	0.942	0.815	0.000	-0.337	1.420
NMCPE	0.000	0.000	0.942	0.815	-0.988	-0.337	0.433
MSR	-0.146	-0.434	0.000	0.815	0.000	-0.675	-0.440
GPR	-0.146	-0.434	-0.942	0.000	-0.988	-0.675	-3.185
ITCC	0.000	0.434	0.000	0.815	0.000	-0.675	0.574
DGMDH	0.073	0.217	0.942	0.815	0.988	0.000	3.035
TE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH	Total
NAEN	0.000	0.000	-0.942	0.815	0.000	-0.337	-0.465
NMCPE	0.000	0.000	0.942	0.815	0.494	-0.675	1.577
MSR	0.146	-0.434	0.000	0.000	-0.988	-0.675	-1.950
GPR	-0.146	-0.434	0.000	0.000	-0.988	-0.675	-2.242
ITCC	0.000	-0.217	0.942	0.815	0.000	-0.675	0.866
DGMDH	0.073	0.434	0.942	0.815	0.988	0.000	3.252

TABLE 5. The results of random weighted rank sorting.

RMSE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
Clopi102	1.189	0.297	1.487	1.784	0.892	0.595
Clopi51	1.166	0.583	1.458	1.750	0.875	0.292
Yierat102	0.154	0.462	0.616	0.923	0.770	0.154
Yierat51	0.257	0.772	1.029	1.543	1.286	0.257
Mean	0.692	0.529	1.148	1.500	0.956	0.323
MAPE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
Clopi102	0.892	0.297	1.487	1.784	1.189	0.297
Clopi51	0.875	0.292	0.875	1.750	1.458	0.583
Yierat102	0.462	0.616	0.154	0.923	0.770	0.308
Yierat51	0.772	1.286	0.257	1.543	1.029	0.514
Total	3.000	2.491	2.772	6.000	4.446	1.703
RRSE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
Clopi102	1.189	0.892	1.487	1.784	0.595	0.297
Clopi51	1.166	0.875	1.458	1.750	0.583	0.292
Yierat102	0.308	0.616	0.462	0.923	0.770	0.154
Yierat51	0.257	0.772	1.029	1.543	1.286	0.514
Total	2.921	3.154	4.435	6.000	3.233	1.257
TE	NAEN	NMCPE	MSR	GPR	ITCC	DGMDH
Clopi102	1.189	0.595	1.784	1.487	0.892	0.297
Clopi51	1.166	0.583	1.458	1.750	0.875	0.292
Yierat102	0.154	0.462	0.770	0.923	0.462	0.154
Yierat51	0.257	0.772	1.286	1.543	1.029	0.257
Total	2.767	2.411	5.297	5.703	3.257	1.000

GPR method, but may not superior to the NAEN and NMCPE for most error estimation method yet.

4) COMPARISON WITH DIFFERENT INDEX TRACKING METHOD

The NAEN and NMCPE belong to partially linear models. It is an important semi-parametric regression model, which adds more non-parametric parts than the linear model. Therefore, the partial linear models not only inherit the characteristics of linear ones that are easy to interpret, but also retain the flexibility of non-parametric regression models.

Meanwhile, they overcome the curse of dimensionality confronted by non-parametric regression model. The MSR is a non-linear and non-parametric regression method, which is a local regression method that uses a spline function to simulate complex non-linear relationships. It divides the entire non-linear model into several regions, which are fitted by a linear regression line in each specific region. The slope of the spline function is constant in each specific area, but varied in different areas. However, this locality may affect the regression performance of time series data. The GPR belongs to non-parametric function approximation, which relies more on sample data where predicting and estimating are concerned. Therefore, it is the number of variables that determines the calculation amount and the effect of regression. The ITCC method: the main disadvantages of the principal component analysis algorithm are: on the one hand, it does not consider the relationship between independent variables and dependent variables in the process of data dimensionality reduction, resulting in the ambiguity of the meaning of each characteristic dimension of the principal components, so when there are many explanatory variables, the extracted principal components are generally difficult to explain. On the other hand, non-principal components with small variance may also contain crucial information on sample differences, because dimension reduction and discarding can impact subsequent regression analysis. DGMDH algorithm is a combined method of data processing based on $K - G$ polynomials to identify non-linear systems through continuous screening and combination. By finding the best balance point between the fitting accuracy of the sample and the prediction accuracy of the new data set, the algorithm can reflect the real internal relationship of the system to the greatest extent even when the sample data is small or the data noise is large. The layer by layer selection of the model structure and variables in the modeling process ensures the convergence speed of the calculation, which also greatly reduces the impact of the subjective factors.

VII. CONCLUSION

We propose a high-dimensional stock index tracking method called LSTM-DGMDH network. It utilizes LSTM and deep evolutionary GMDH-type neural networks. Our method competes well with other approaches for the high-dimensional datasets of the CSI 300 index. It is feasible for three reasons: (1) Our stock index data preprocessing model optimizes the original data using the attention mechanism. This ensures the data is continuous, complete, and sequential. (2) DGMDH selects relevant input variables and generates a concise model structure through the incomplete induction method. This avoids concerns about variable multicollinearity. (3) Traditional external criteria are inadequate for high-dimensional stock index datasets. Hence, we employ TEEC for better external criterion analysis and optimal complex model selection. The heuristic self-organizing method dynamically selects the model structure and estimates parameters without requiring a specific form of the model. Researchers can

choose the best solution from the model scheme results. We also utilize DGMDH for feature selection and index tracking in other high-dimensional financial datasets. In the future, we foresee the LSTM-DGMDH network extending its applications to a wider range of financial datasets, beyond the CSI 300 index. This advancement will position our method to contribute to the analysis and prediction of global stock market trends, offering valuable insights for investors and researchers worldwide.

REFERENCES

- [1] D. T. Tran, A. Iosifidis, J. Kannianen, and M. Gabbouj, "Temporal attention-augmented bilinear network for financial time-series data analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1407–1418, May 2019, doi: [10.1109/TNNLS.2018.2869225](https://doi.org/10.1109/TNNLS.2018.2869225).
- [2] N. Li and H. Yang, "Nonnegative estimation and variable selection under minimax concave penalty for sparse high-dimensional linear regression models," *Stat. Papers*, vol. 62, pp. 661–680, Apr. 2019.
- [3] N. Li, H. Yang, and J. Yang, "Nonnegative estimation and variable selection via adaptive elastic-net for high-dimensional data," *Commun. Statist.-Simul. Comput.*, vol. 50, no. 12, pp. 4263–4279, Dec. 2021.
- [4] L. Li, D. Li, T. Song, and X. Xu, "Actor-critic learning control with regularization and feature selection in policy gradient estimation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1217–1227, Mar. 2021, doi: [10.1109/TNNLS.2020.2981377](https://doi.org/10.1109/TNNLS.2020.2981377).
- [5] X. Wu, X. Xu, J. Liu, H. Wang, B. Hu, and F. Nie, "Supervised feature selection with orthogonal regression and feature weighting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 1831–1838, May 2021, doi: [10.1109/TNNLS.2020.2991336](https://doi.org/10.1109/TNNLS.2020.2991336).
- [6] Y. Zheng, B. Chen, T. M. Hospedales, and Y. Yang, "Index tracking with cardinality constraints: A stochastic neural networks approach," in *Proc. 34th Conf. Artif. Intell. (AAAI), 32nd Innov. Appl. Artif. Intell. Conf. (IAAI), 10th Symp. Educ. Adv. Artif. Intell. (EAAI)*, New York, NY, USA, 2020, pp. 1242–1249. [Online]. Available: <https://aaai.org/ojs/index.php/AAAI/article/view/5478>
- [7] K. Ren, H. Yang, Y. Zhao, W. Chen, M. Xue, H. Miao, S. Huang, and J. Liu, "A robust AUC maximization framework with simultaneous outlier detection and feature selection for positive-unlabeled classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3072–3083, Oct. 2019, doi: [10.1109/TNNLS.2018.2870666](https://doi.org/10.1109/TNNLS.2018.2870666).
- [8] S. Yi, Z. He, X.-Y. Jing, Y. Li, Y.-M. Cheung, and F. Nie, "Adaptive weighted sparse principal component analysis for robust unsupervised feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2153–2163, Jun. 2020, doi: [10.1109/TNNLS.2019.2928755](https://doi.org/10.1109/TNNLS.2019.2928755).
- [9] P. Zhou, X. Hu, P. Li, and X. Wu, "OFS-density: A novel online streaming feature selection method," *Pattern Recognit.*, vol. 86, pp. 48–61, Feb. 2019, doi: [10.1016/j.patcog.2018.08.009](https://doi.org/10.1016/j.patcog.2018.08.009).
- [10] S. Munoz-Romero, A. Gorostiaga, C. Sogueru-Ruiz, I. Mora-Jiménez, and J. L. Rojo-Alvarez, "Informative variable identifier: Expanding interpretability in feature selection," *Pattern Recognit.*, vol. 98, Feb. 2020, Art. no. 107077, doi: [10.1016/j.patcog.2019.107077](https://doi.org/10.1016/j.patcog.2019.107077).
- [11] H. Yuan, J. Li, L. L. Lai, and Y. Y. Tang, "Low-rank matrix regression for image feature extraction and feature selection," *Inf. Sci.*, vol. 522, pp. 214–226, Jun. 2020, doi: [10.1016/j.ins.2020.02.070](https://doi.org/10.1016/j.ins.2020.02.070).
- [12] A. Malhi and R. X. Gao, "PCA-based feature selection scheme for machine defect classification," *IEEE Trans. Instrum. Meas.*, vol. 53, no. 6, pp. 1517–1525, Dec. 2004, doi: [10.1109/TIM.2004.834070](https://doi.org/10.1109/TIM.2004.834070).
- [13] Y. Kim, W. N. Street, G. J. Russell, and F. Menczer, "Customer targeting: A neural network approach guided by genetic algorithms," *Manage. Sci.*, vol. 51, no. 2, pp. 264–276, Feb. 2005.
- [14] M. Sheikhan and N. Mohammadi, "Time series prediction using PSO-optimized neural network and hybrid feature selection algorithm for IEEE load data," *Neural Comput. Appl.*, vol. 23, nos. 3–4, pp. 1185–1194, Sep. 2013, doi: [10.1007/s00521-012-0980-8](https://doi.org/10.1007/s00521-012-0980-8).
- [15] L. Mo, L. Xie, X. Jiang, G. Teng, L. Xu, and J. Xiao, "GMDH-based hybrid model for container throughput forecasting: Selective combination forecasting in nonlinear subseries," *Appl. Soft Comput.*, vol. 62, pp. 478–490, Jan. 2018, doi: [10.1016/j.asoc.2017.10.033](https://doi.org/10.1016/j.asoc.2017.10.033).
- [16] J. Xiao, H. Cao, X. Jiang, X. Gu, and L. Xie, "GMDH-based semi-supervised feature selection for customer classification," *Knowl.-Based Syst.*, vol. 132, pp. 236–248, Sep. 2017, doi: [10.1016/j.knsys.2017.06.018](https://doi.org/10.1016/j.knsys.2017.06.018).
- [17] S. Jeddi and S. Sharifian, "A hybrid wavelet decomposer and GMDH-ELM ensemble model for network function virtualization workload forecasting in cloud computing," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 105940, doi: [10.1016/j.asoc.2019.105940](https://doi.org/10.1016/j.asoc.2019.105940).
- [18] R. A. Gupta and M.-Y. Chow, "Networked control system: Overview and research trends," *IEEE Trans. Ind. Electron.*, vol. 57, no. 7, pp. 2527–2535, 2009.
- [19] A. G. Ivakhnenko, "Heuristic self-organization in problems of engineering cybernetics," *Automatica*, vol. 6, no. 2, pp. 207–219, Mar. 1970.
- [20] F. Lemke and J. Müller, "Self-organizing data mining," in *Proc. Data Mining Und Data Warehousing, Workshop im Rahmen der GI-Jahrestagung Informatik, Magdeburg, Deutschland*, R. Kruse and G. Saake, Eds. Otto-von-Guericke-Univ. Magdeburg, Fakultät für Informatik, Sep. 1998, pp. 107–118.
- [21] C. Leake, "Multicriterion decision in management: Principles and practice," *J. Oper. Res. Soc.*, vol. 52, no. 5, p. 603, May 2001, doi: [10.1057/palgrave.jors.2601200](https://doi.org/10.1057/palgrave.jors.2601200).
- [22] Ehab. E. Elattar, J. Y. Goulermas, and Q. H. Wu, "Generalized locally weighted GMDH for short term load forecasting," *IEEE Trans. Syst., Man, Cybern., C*, vol. 42, no. 3, pp. 345–356, May 2012, doi: [10.1109/TSMCC.2011.2109378](https://doi.org/10.1109/TSMCC.2011.2109378).
- [23] J. Xiao, C. He, and X. Jiang, "Structure identification of Bayesian classifiers based on GMDH," *Knowl.-Based Syst.*, vol. 22, no. 6, pp. 461–470, Aug. 2009, doi: [10.1016/j.knsys.2009.06.005](https://doi.org/10.1016/j.knsys.2009.06.005).
- [24] Y. Wei, Y. Tang, and P. D. McNicholas, "Flexible high-dimensional unsupervised learning with missing data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 610–621, Mar. 2020, doi: [10.1109/TPAMI.2018.2885760](https://doi.org/10.1109/TPAMI.2018.2885760).
- [25] T. Zhou, S. Gao, J. Wang, C. Chu, Y. Todo, and Z. Tang, "Financial time series prediction using a dendritic neuron model," *Knowl.-Based Syst.*, vol. 105, pp. 214–224, Aug. 2016, doi: [10.1016/j.knsys.2016.05.031](https://doi.org/10.1016/j.knsys.2016.05.031).
- [26] B. Moews, J. M. Herrmann, and G. Ibikunle, "Lagged correlation-based deep learning for directional trend change prediction in financial time series," *Exp. Syst. Appl.*, vol. 120, pp. 197–206, Apr. 2019, doi: [10.1016/j.eswa.2018.11.027](https://doi.org/10.1016/j.eswa.2018.11.027).
- [27] M. Han, S. Feng, C. L. P. Chen, M. Xu, and T. Qiu, "Structured manifold broad learning system: A manifold perspective for large-scale chaotic time series analysis and prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1809–1821, Sep. 2019, doi: [10.1109/TKDE.2018.2866149](https://doi.org/10.1109/TKDE.2018.2866149).
- [28] J. Mei, Y. de Castro, Y. Goude, J. Azaïs, and G. Hébrail, "Nonnegative matrix factorization with side information for time series recovery and prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 3, pp. 493–506, Mar. 2019, doi: [10.1109/TKDE.2018.2839678](https://doi.org/10.1109/TKDE.2018.2839678).
- [29] C. R. Madan, "Prism: Multiple spline regression with regularization, dimensionality reduction, and feature selection," *J. Open Source Softw.*, vol. 1, no. 3, p. 31, Jul. 2016, doi: [10.21105/joss.00031](https://doi.org/10.21105/joss.00031).
- [30] F. Yin, L. Pan, T. Chen, S. Theodoridis, Z. T. Luo, and A. M. Zoubir, "Linear multiple low-rank kernel based stationary Gaussian processes regression for time series," *IEEE Trans. Signal Process.*, vol. 68, pp. 5260–5275, 2020, doi: [10.1109/TSP.2020.3023008](https://doi.org/10.1109/TSP.2020.3023008).
- [31] Y. Yang and L. Wu, "Nonnegative adaptive lasso for ultra-high dimensional regression models and a two-stage method applied in financial modeling," *J. Stat. Planning Inference*, vol. 174, pp. 52–67, Jul. 2016.



HE TONG was born in Liaoning, China, in 1980. She received the B.Eng. degree in communication engineering from Liaoning University, in 2004, and the M.E. degree in communication and information from Northeastern University, China, in 2007. From 2007 to 2008, she was a Teacher with the Liaoning University of Petroleum and Chemical Technology. Since 2008, she has been a Teacher of computer science with the Chinese People's Liberation Army Aviation Institute. She has published about 40 articles in journals and has edited over ten textbooks in computer science and electronic technique. Her current research interests include DSP, embedded technology, and AI.



YUSHENG LIU is currently pursuing the master’s degree with the School of Artificial Intelligence and Big Data, Hefei University. His research interests include graph neural networks, graph contrastive learning, and recommender systems.



SIBAO CHEN (Member, IEEE) received the B.S. and M.S. degrees in probability and statistics and the Ph.D. degree in computer science from Anhui University, Hefei, China, in 2000, 2003, and 2006, respectively. From 2006 to 2008, he was a Postdoctoral Researcher with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei. Since 2008, he has been a Teacher with Anhui University. From 2014 to 2015, he was a Visiting Scholar with The University of Texas at Arlington, Arlington, TX, USA. His research interests include image processing, pattern recognition, machine learning, and computer vision.



LIN LIU received the bachelor’s degree in communication engineering from the Hefei University of Technology, in 2005, and the master’s degree in circuit and systems from the University of Science and Technology of China, in 2008. He is currently pursuing the Ph.D. degree in electronic information with the China University of Science and Technology. In August 2021, he was awarded the senior professional title of Electronic Information Engineering. He is currently the Chief Engineer with Hefei iFLYTEK Digital Technology Company Ltd. He has applied for 36 invention patents and won the 17th “China Patent Excellence Award” in 2016. He has been engaged in long-term research on artificial intelligence technology in signal and information processing fields, such as speech, image, and electromagnetic. He is also a Standing Committee Member of Information Security Committee, CAAI.



LIXIANG XU (Member, IEEE) received the B.Sc. and M.Sc. degrees in applied mathematics, in 2005 and 2008, respectively, and the Ph.D. degree from the School of Computer Science and Technology, Anhui University, Hefei, China, in 2017. He has been awarded a scholarship to pursue his study in Germany, as a joint Ph.D. student, from 2015 to 2017. He was with Huawei Technologies Company Ltd., in 2008, before joining Hefei University, China, in the following year, where he is currently a Professor. He is also doing a postdoctoral research with the University of Science and Technology of China. His current research interests include structural pattern recognition, machine learning, graph spectral analysis, image and graph matching, especially in kernel methods and complexity analysis on graphs and networks.



NING LI received the Ph.D. degree. He was a Supervisor of master’s students. He is a Statistician. He has been a Principal Investigator of projects, such as the Anhui Natural Science Foundation Youth Project and the Anhui University Natural Science Research Key Project. Furthermore, he has contributed to several academic articles published in journals, such as *Statistical Papers* and the *Journal of Statistical Computation and Simulation*. His research interests include parameter and semi-parameter model selection along with its practical applications.



YUANYAN TANG (Life Fellow, IEEE) is the Chair Professor with the Faculty of Science and Technology, University of Macau, and a Professor/an Adjunct Professor/a Honorary Professor with several institutes, including Chongqing University, China; Concordia University, Canada; and Hong Kong Baptist University, Hong Kong, respectively. He has published more than 400 academic papers and the author/coauthor of over 25 monographs/books/bookchapters. His current research interests include wavelets, pattern recognition, and image processing. He is an IAPR Fellow. He served as the general chair, the program chair, and a committee member for many international conferences. He is the Founder and the Chair of the Pattern Recognition Committee, IEEE SMC. He is the Founder and the General Chair of the Series International Conferences on Wavelets Analysis and Pattern Recognition (ICWAPRs). He is the Founder and the Chair of the Macau Branch of International Associate of Pattern Recognition (IAPR). He is the Founder and the Editor-in-Chief of *International Journal on Wavelets, Multiresolution and Information Processing* and an associate editor of several international journals.

...