ITC 4/53

Information Technology
and Control
Vol. 53 / No. 4/ 2024
pp. 1236-1250
DOI 10.5755/j01.itc.53.4.37053

Internet Finance Non-stationary Time Series Prediction Algorithm Based on Deep Learning

Received 2024/04/23 | Accepted after revision 2024/07/10

HOW TO CITE: Li, Y. (2024). Internet Finance Non-stationary Time Series Prediction Algorithm Based on Deep Learning. *Information Technology and Control*, 53(4), 1236-1250. https://doi.org/10.5755/j01.itc.53.4.37053

Internet Finance Non-stationary Time Series Prediction Algorithm Based on Deep Learning

Yangyi Li

Southern Power Grid Digital Enterprise Technology (Guangdong) Co., Ltd, Southern Power Grid Digital Grid Research Institute Co., Ltd, Guangzhou, 510000, China

Corresponding author: iuiuuifuif@163.com

Inaccurate prediction results of financial time series will lead to wrong investment decisions. Therefore, a prediction algorithm for Internet financial non-stationary time series based on deep learning is proposed. EMD (empirical mode decomposition) method is used to divide the collected historical Internet financial non-stationary time series information into high-frequency and low-frequency parts, and remove the noise in the decomposed high-frequency components to obtain the financial non-stationary time series without noise. The knowledge map method is used to mine the transaction characteristics and market characteristics of Internet finance from the financial non-stationary time series without noise, and the two are fused as the input of the improved CNN (convolutional neural network) prediction model. The prediction results of Internet financial time series are obtained through CNN. The experimental results show that after setting the CNN parameters, the predicted results are consistent with the actual market trends. The highest RSE of the predicted result is 0.551, The highest RAE is 0.443, which is relatively low, the CORR value is 0.864, which is relatively high, indicating that the relative square root error, relative absolute error, and relevant empirical coefficients of the prediction results are all good, making it a highly applicable algorithm.

KEYWORDS: EMD; CNN; Knowledge map; Non-stationary time series; Internet finance; Prediction algorithm.

1. Introduction

Network finance [3], also known as E-finance, refers to various financial activities based on the achievements of financial computerization through the Internet platform. This covers multiple aspects such as the operation of online financial institutions [1], the conduct of online financial transactions, and the operation and regulation of online financial markets [8]. More specifically, online finance involves a series of financial businesses carried out on the Internet, such as online banking services [16], online securities trading, online insurance business, online stock trading and options trading. These services together constitute a wide range of content in online finance, providing people with a more convenient and efficient financial service experience. In a broad sense, online finance is based on network technique. For support, the general term of all financial activities in the global scope includes not only narrow content, but also network financial security [17] , Internet financial supervision and various other factors stand apart from the conventional financial endeavors that manifest in physical form. Instead, it represents a financial activity that resides in the electronic realm, taking on a virtual existence and operating in a networked manner. This activity is a testament to the advancements in information technology [10], particularly the Internet, and serves as a finance tailored to meet the evolving demands of e-commerce operations in the digital age.

1.1. Literature Review

When it comes to Internet finance time series forecasting, research centers on analyzing the variables within the market. By employing mathematical modeling, researchers delve into the market's evolutionary patterns to anticipate future trends. Scholars like Fatima [6] have utilized methods like the DCC-GARCH model and multi-layered neural networks to forecast multiple financial time series. The DCC-GARCH approach captures volatility and covariance across multiple time series, while the hybrid approach combining multivariate artificial neural networks (MANN) enhances prediction accuracy and validates time-varying correlations. However, DCC-GARCH involves complex parameter settings (e.g., lag order, distribution assumptions), and neural networks' performance hinges on factors like network structure, activation functions, and training algorithms. Despite their intricacies, these methods contribute significantly to financial time series forecasting in the Internet finance realm. Luca et al. [9] explored Boltzmann entropy as a financial indicator for time series prediction, comparing it to traditional analysts' tools. They assessed its effectiveness on stocks and encrypt stocks. LSTM networks excel in forecasting but risk overfitting. Pinto et al. [12] proposed SODA-t2fts, leveraging interval type-2 fuzzy sets for data-driven partitioning, enhancing prediction with lower error, complexity, and noise sensitivity. However, interval type-2 fuzzy sets' complexity poses challenges in parameter setting, potentially impacting accuracy. Pradeepkumar and Ravi [13] introduced a financial forecasting method blending time series motifs with neural networks. EP-C detects motifs, and GMDH, MLP, and GRNN predict based on motif info. The Motif+GMDH model excelled in euro/USD, INR/USD, and crude oil price forecasting. This offers a novel approach but EP-C's parameter sensitivity may hinder motivation detection accuracy. Nikolaos et al. [11] propose a robust deep adaptive input normalization method for financial time series prediction. It learns input distribution, applies optimal normalization, and runs on sliding windows to handle non-stationarity. Accurate prediction is achieved. Proper sliding window size is key to capturing features, avoiding information loss or noise. Sidekerskienė et al. [15] used Hankel matrices & ABC optimization for time series modeling, improving prediction. ARX's exogenous variables complexity risks over/underfitting. Chen et al. [4] introduced 2CFastICA for EMG decomposition, validated in simulations & real data. Financial time series noise/outliers may hinder decomposition, affecting predictions. Damaševičius et al. [5] used attention RNN with time series decomposition & metaheuristic optimization to predict renewable energy. RNN structure must be tuned for tasks; inappropriate settings may cause info loss or poor performance.

Deep learning in time series focuses on data-driven, handling nonlinearities. A proposed algo for Internet financial time series uses EMD to decompose into IMFs. Removing noise & preserving lows yields clearer data. Integrated with CNN, enhanced by residuals & multilevel conv/pooling for accurate predictions, aiding Internet finance.

2. Internet Finance Non-stationary Time Series Prediction

2.1. Decomposition and Noise Reduction of Internet Financial Non-stationary Time Series Based on EMD Method

Internet financial institutions will establish various financial platforms, such as online financial services, crowdfunding P2P and third-party payment platforms have a large amount of transaction data, such as wealth management product data, insurance product data, product portfolio data, payment data, etc. The above online financial data, especially those related to the financial market, such as stock prices, exchange rates, bond prices, etc., are usually non-stationary data. Moreover, due to market fluctuations, policy changes, and unexpected events, financial time series data often contains various types of noise, which can interfere with the performance of prediction models. EMD decomposes signals based on the time scale characteristics of the data itself, without the need to set any basis functions in advance. Through the iterative screening process, EMD can decompose a complex signal into several IMFs and a residual term. Each IMF represents a signal component with a specific time scale, while the residual term reflects trends or long-term changes in the signal. Therefore, The EMD method can decompose financial time series into several IMF components and a residual term, and reduce noise interference by removing noise components or merging similar components. Meanwhile, The IMF components have low correlation, which makes the denoising process more effective, thereby improving the accuracy and stability of the prediction model.

2.1.1. Decomposition of Non-stationary Time Series of Internet Finance

Empirical Mode Decomposition (EMD) is an efficient method for handling non-stationary nonlinear sequences, which has the advantage of not requiring preset basis functions and demonstrating excellent adaptability. Essentially, EMD is a stabilization process that decomposes a sequence into multiple stationary fluctuation terms and residual trend terms of different scales through a fixed model, each of which is referred to as the intrinsic mode function (IMF), thus achieving effective analysis of complex sequences. The determination of IMF component needs to meet two conditions:

1. The number of extreme points and zero crossing points of the component should be the same or only differ by one.
2. At each time point, the local maximum and minimum points of the component form upper and lower envelopes, respectively. The two envelopes obtained through cubic spline interpolation are locally symmetric about the time axis, that is, their average value is zero.

EMD decomposes the non-stationary time series of Internet finance [7], the detailed steps are as follows:

1. Determine the sequence to be decomposed $X_0(t)$ of all local maximum and minimum points which are connected the upper envelope formed by cubic spline interpolation is represented as $u_0(t)$, all local minimum points are also formed into the lower envelope $l_0(t)$.
2. Find the mean value of the upper and lower envelope lines $m_0(t)=\dfrac{u_0(t)+l_0(t)}{2}$, the sequence to be decomposed $X_0(t)$ subtract mean $m_0(t)$ and get component $h_1(t)$, i.e.:

$$h_1(t)=X_0(t)-m_0(t). \tag{1}$$

3. Judgment $h_1(t)$ whether the IMF conditions are met, and if so $h_1(t)$ is the first IMF, otherwise, do the same according to the non-stationary time series of Internet finance to $h_1(t)$. Get new components for $h_2(t)=h_1(t)-m_1(t)$, make the same judgment and treatment until $h_k(t)$ if the IMF conditions are met or the stop criteria are met, the stop criteria are as follows:

$$SD=\sum_{t=0}^{T}\frac{|h_{k-1}(t)-h_k(t)|^2}{h_{k-1}^2(t)}. \tag{2}$$

   When the standard deviation is less than a specific threshold, it is considered to have reached the stopping standard, and the specific value is generally selected between [0.2, 0.3]. So far, the first IMF component of the non-stationary time series of Internet finance has been obtained imf_1.
4. Set the remaining component $r_1=X_0(t)-imf_1$ as a new sequence to be decomposed, repeat the above steps until the component imf_n or residual component r_n is less than predetermined value or residual component r_n ends when it becomes a monotone function. At this time:

$$X(t)=\sum_{k=1}^{n}imf_k+r_n\ . \tag{3}$$

In this case, $X(t)$ that is, the decomposed non-stationary time series of Internet finance. IMF component imf_k along with k the IMF component decomposed first is the high-frequency part of the original Internet financial non-stationary time series, and also the part with high noise. Residual component r_n is a trend item that reflects the trend of non-stationary time series of Internet finance.

2.1.2. Noise Reduction of Non-stationary Time Series

Step 1: IMF energy calculation.

The core of EMD denoising idea is to decompose the non-stationary time series of Internet finance in detail $X(t)$ high frequency components typically contain high levels of noise. To clearly distinguish between high-frequency and low-frequency components, it is necessary to calculate the energy of each IMF:

$$E(imf_k) = X(t)\sqrt{\sum_{t=1}^{T} imf_{kt}^2} \,. \tag{4}$$

Among them, imf_{kt} represents IMF component of Internet financial non-stationary time series imf_k stay t value of time, T is the sequence length.

Step 2: Determine the mutation point.

After EMD decomposition of the non-stationary time series of Internet finance, the high-frequency IMF component shows a lower energy level, while the low-frequency IMF component has a higher energy performance. There is a significant abrupt change in the energy of the IMF between the high-frequency and low-frequency components. To define the range of high-frequency components, it is necessary to determine this mutation point. The estimation of mutation or breakpoint can be expressed as:

$$R(K) = |\frac{K(n-K)}{n^2}\left[\frac{1}{K}\sum_{k=1}^{K} E(imf_k) - \frac{1}{n-K}\sum_{k=K+1}^{n} E(imf_k)\right]| \tag{5}$$

$$K_0 = \arg\max_K \left[R(K)\right] \tag{6}$$

That is, to traverse IMF energy as a breakpoint K, where $R(K)$ the point that reaches the maximum is the catastrophe point K_0.

Step 3: Denoising processing.

After determining the mutation point K_0, right $imf_1 \sim imf_{K_0}$ perform denoising processing respectively to obtain the denoised components $imf_1' \sim imf_{K_0}'$. After denoising, high-frequency components, low-frequency components and trend items will be reconstructed to obtain the Internet financial non-stationary time series after noise reduction $X(t)$ denoised sequence $X'(t)$:

$$X'(t) = \sum_{k=1}^{K_0} imf_k' K_0 + \sum_{k=K_0+1}^{n} imf_k + r_n \,. \tag{7}$$

This paper adopts empirical mode decomposition method for denoising processing:

$$imf_{kt}' = \begin{cases} \mathrm{si}\,gn(imf_{kt})(|\,img_{kt}\,| - \lambda), |\,imf_{kt}\,| \geq \lambda \\ 0, |\,imf_{kt}\,| < \lambda \end{cases}, \tag{8}$$

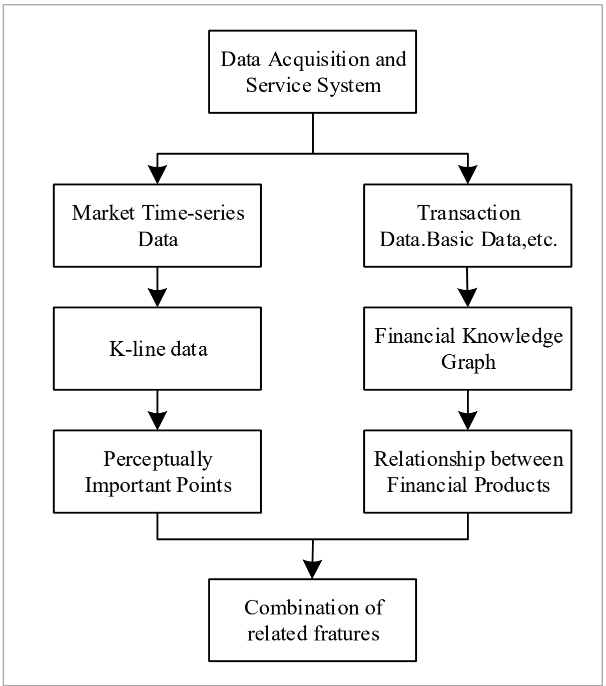
where, imf_{kt}' represents denoising the value of component imf_{kt}' at t time, $sign(\ )$ represents the characteristic function, threshold value of imf_k is $\lambda = \sigma_k\sqrt{2\ln T}$, $\sigma_k = \frac{median(|\,imf_k\,|)}{0.6745}$, $median(\ )$ represents the median.

2.2. Financial Non-stationary Time Series Feature Mining and Fusion

After EMD denoising, information and rules in non-stationary time series can be obtained, but the denoised information and rules are relatively scattered, which means that in certain specific financial time series, The EMD method may not be able to completely remove noise, or it may lose some useful information during the denoising process. Financial time series have non stationarity, which means their statistical characteristics change over time. By mining and fusing these non-stationary features, can better understand the dynamic changes of time series and make more accurate predictions and decisions. Therefore, it is necessary to mine and fuse the characteristics of Internet financial non-stationary time series to improve the prediction accuracy. The traditional trend prediction method does not fully consider the local trend characteristic mode of Internet financial market and the correlation analysis of trading behavior. To address the aforementioned issues, this paper innovatively introduces the method of knowledge graph [14] and effectively applies it to the fusion process of heterogeneous information. The integration of transaction information and market information based on the financial knowledge map is shown in Figure 1.

Figure 1

Composite diagram of trading information and market information



2.2.1. Feature Mining

Knowledge graph G is a network structure formed by the interconnection of entities (nodes) and relationships (edges) between entities in the real world. Generally, the complex relationship between different elements is expressed in the form of triple, and each piece of knowledge in the atlas can be intuitively expressed as $G=\langle head, relation, tail\rangle$, where $head$, $tail$ the first entity and the last entity of the triple belong to collection of entities G, $relation=\{r_1, r_2, \cdots, r_{|n|}\}$ is the relationship set of G, including n different relationships.

The spot stock market and futures market are often related, and futures prices (especially stock index futures) tend to lead stock prices to rise and fall. For example, the trend of related varieties in the overnight futures market has a huge impact on related stock market sectors on the second day. Taking the CSI 300 stock index futures contract as an example, in the first iteration, the entity type includes investors, contract codes, etc., and the relationship type includes preference relationship, profit and loss relationship, etc. In the second iteration, entity attributes such as company, industry, region and other entity attributes of the relevant stocks involved, as well as the shareholding, association and subordination relationships among companies.

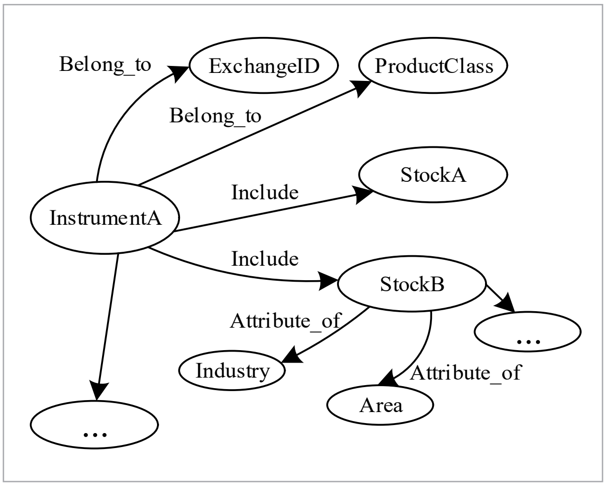
This paper adopts a top-down strategy to construct a knowledge graph, based on investor transaction orders and related basic data, carefully constructing a knowledge graph in the financial field. Discretized trading and market data and conducted in-depth semantic analysis to extract discrete triplets. On this basis, a knowledge graph of financial time series was further constructed to deeply explore the internal connections and potential characteristics between financial entities, in order to reveal the deep laws and dynamic changes of the financial market.

Financial time series knowledge map FinKG is a directed label map $FinKG_t = imf'_{kt}(T_s, T_e, E, R, \tau)$, where E is the vertex set representing a knowledge graph, which is used to represent the entity set; R is the edge set of the knowledge map, which is used to represent the fact relationship set; τ is the function of $E \times E \to R \mid k$ which represents all tuples in the knowledge map; k indicates in the time period $[T_s, T_e]$ in the list of knowledge map triplets sorted by time, there is a k secondary relation R between the two entities. An example of the concept of financial knowledge mapping is shown in Figure 2.

The financial market is a complex and nonlinear dynamic system, influenced by macroeconomic factors,

Figure 2

Example of Internet financial knowledge graph



underlying fundamentals, investor behavior, and other factors. Financial time series data not only contains time series information, but also implies complex relationships between various assets, market participants, and trading behaviors. The construction of the financial temporal knowledge graph shown in Figure 2 can represent these complex relationships in a structured manner, clearly representing the entities in financial data and their relationships, facilitating subsequent deep learning models to gain a deeper understanding of the inherent laws and structures of financial data.

Trans R (transformation model) for each triplet (h,r,t) each relationship in defines a projection matrix $M_r \in R^{k*d}$, the entity vector in the entity space is passed through M_r the project to relationship r subspace, l_{hr} and l_{tr} expressed as:

$$l_{hr} = l_h M_r \tag{9}$$

$$l_{tr} = l_t M_r . \tag{10}$$

The corresponding loss function is:

$$f_r(h,t) = \left\| l_{hr} + l_r - l_{tr} \right\|_{L1/L2} . \tag{11}$$

Using TransR transformation model to map knowledge *FinKG* period selected in $[T_s, T_e]$ triple inside (E_i, R, E_j) embedded in low dimensional space *FinKG* when graph embedding is performed, each element is converted into vector representation. So, *FinKG* period selected in $[T_s, T_e]$ the entities (affiliated stocks, etc.) within are embedded into k in a dimensional hyperplane, relationships are embedded into d dimension hyperplane. It can be selected according to actual experience k and d.

According to TransR principle k the dimension vector distance is relatively close. By calculating the i branch stock vector v_i and jth stock vector v_j, the Euclidean distance between stocks as shown in Formula (12) is used to measure the similarity between stocks.

$$d(v_i, v_j) = \sqrt{(v_i - v_j)(v_i - v_j)^T} . \tag{12}$$

The above method can be used to calculate the similarity between any pair of stocks, and the related stocks are those with $top-K$ Similar stocks.

2.2.2. Feature Fusion

In order to better reflect the price trend of Internet financial targets (such as stocks), this paper selects the characteristic indicators of financial targets and related financial products within a certain period to build their market information according to the similarity between stocks $d(v_i, v_j)$.It covers a wealth of market data, including core indicators such as opening price, highest price, lowest price, closing price, and trading volume, as well as derivative feature data such as Z-shaped shape, MA moving average, MACD index, and KDJ index, collectively forming its comprehensive and in-depth feature information system.

1 Basic data

The most basic data is OHLC (opening price, highest price, lowest price, closing price) market data. Other K-line technical indicators are derived data based on these basic data, such as high opening and low closing prices, closing prices, and trading volume, which have been calculated.

$Open(t)$: The opening price at time t.

$High(t)$: High prices at time t.

$Low(t)$: Low prices at time t.

$Close(t)$: Closing price at time t.

$Volume(t)$: Transaction volume at time t.

2 Derived Data Features

In addition to the basic data, also consider some derivative indicators most relevant to the yield. The commonly used derivative data mainly include Zigzag, MA moving average, MACD, KDJ and other data.

In addition to these financial indicator data, other data will also be used as derivative data. For example, transaction time will be divided into more detailed parameters such as year, month, day and week according to the format. The company's stock code, name, industry and region will be included in the analysis model as parameters.

For each target financial product, its feature information feature vector will be constructed according to the above indicators i day j of the market information vector of only relevant stocks is expressed as m_{ij}. Then, use knowledge atlas and graph embedding technology to select financial products related to the target stock, and combine their feature information to obtain a comprehensive feature information vector. If select the target stock and the most relevant k

stocks to jointly predict the future trend of the underlying stock, will get i day's market information vector based on related stocks $M=\left[m_{i1},m_{i2},\cdots,m_{i(k+1)}\right]$. They are used together as the input data of the Internet financial non-stationary time series prediction model, so as to enrich the semantic information of the input data and improve the prediction accuracy.

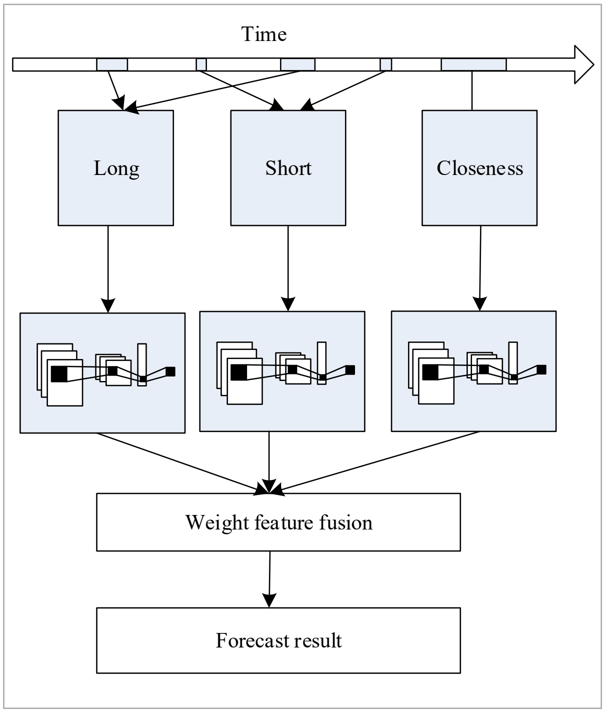
2.3. CNN Structure Improvement

Although the EMD method is highly adaptive, it may require a lot of computing resources and time when dealing with complex Internet financial time series. Meanwhile, The EMD method requires extreme points of data, and for data that does not meet basic conditions, preprocessing or transformation may be required. In order to integrate data from different sources and types and provide more comprehensive and comprehensive information, CNN is used to predict the non-stationary time series of Internet finance. The parallel computing ability of CNN makes it highly efficient in processing long sequences [2]. Meanwhile, the convolutional kernel of CNN can be seen as a learnable filter that can capture specific patterns or structures in time series. The size, number, and network depth of convolutional kernels can be adjusted and optimized according to specific tasks to adapt to different data features and requirements. However, CNN may have some limitations in dealing with long-term dependence and the internal dynamic characteristics of the series, especially for the Internet financial non-stationary time series with strong long-term dependence. This means that CNN may have difficulty capturing distant information in time series that has a significant impact on prediction. The residual network allows the network to learn the residual information between input and output by introducing residual blocks, which makes it easier to optimize the deep network, further improves the prediction accuracy of the Internet financial non-stationary time series, prevents gradient diffusion and gradient explosion, and solves the problem that the nonlinear model is insensitive to the changes in the scale of input data.

This paper uses the optimized CNN structure to predict the non-stationary time series of Internet finance. The prediction model takes specific time series features as inputs and is divided into three parts: current time point features, short-term non-stationary features, and long-term non-stationary features. These three features are input into three different convolutional components, and finally the final prediction result is obtained by weighting and fusing the outputs of these three convolutional components. Figure 3 shows the structural design of the improved CNN.

Figure 3

Improving the CNN structure



2.3.1. Design of Nearby Convolution Components

The first part of CNN uses the convolutional neural network with pooling layer to extract the correlation between short-term patterns and variables in the non-stationary time series of Internet finance. This part has a particularly significant impact on the prediction results. In addition, in this part of the entire network, each convolutional layer has a width of w and height is a composed of multiple filters, wherein w refers to the time interval set independently, a is equal to the number of variables in the sequence, that is, the number of observation points in general. To reduce the output scale and effectively avoid overfitting issues, a pooling layer is introduced in this component. Definition to calculate k the input vector of the nearest convolution component of the filter X_{cT}, and produce:

$$H_{ck} = Mf\left(W_{ck} * X_{cT} + b_{ck}\right). \quad (13)$$

Among $*$ represents convolution operation, H_{ck} is the output vector, W_{ck} and b_{ck} are weight parameter, f is the activation function. With each convolution operation, the size of the output matrix diminishes, leading to a significant loss of edge position information, which in turn, negatively impacts subsequent convolution operations. Therefore, when the input matrix X_{cT}, there is zero filling on the left side of, so the size of the output matrix after the convolution layer operation is always $a \times k_c$. Finally, the size of the output matrix calculated from the entire convolution integral is $d_c \times a \times k_c$, where d_c is the number of filters in the last convolution layer. Therefore, the output after convolution can be expressed as $V_c \in R^{d_c \times a \times k_c}$, complete the design of the nearest convolution component in CNN.

2.3.2. Design of Periodic Partial Convolution Component

The convolution component in this section resembles its nearest counterpart, but a key distinction lies in the absence of a pooling layer. This omission is due to the fact that extracting features from long and short-term internet financial non-stationary time series is more challenging compared to the latest information at the predicted time point. Consequently, eliminating the pooling layer in this part allows us to maximize the extraction of periodic characteristics corresponding to the predicted time points of the internet financial non-stationary time series. Specifically, this component comprises two parts: one dedicated to the short-cycle model and the other to the long-cycle model.

For the short period component of the non-stationary time series characteristics of Internet finance p_s is the short cycle length k_s for short cycle span, can make $\left[X_{(T+h)} - k_s * p_s, X_{(T+h)} - (k_s - 1) * p_s, \cdots, X_{(T+h)} -\right.$ as a dependency sequence. Similarly, with p_l as the long period length, k is the long period span, and the characteristics of the model's long period non-stationary time series can be obtained, as shown below: $X_{(T+h)} - k_l * p_l, X_{(T+h)} - (k_l - 1) * p_l, \cdots, X_{(T+h)}$. Here, only the daily and weekly sequence features need to be extracted for experimental modeling, but the model framework proposed in this paper allows users to customize other types of cycle time periods (for example, every month, every season, and every year).

Similar to the initial component, define the necessary calculations to determine the output vector for both the long-period and short-period components within each convolution layer:

$$H_{sk} = V_c f\left(W_{sk} * X_{sT} + b_{sk}\right) \quad (14)$$

$$H_{lk} = V_c f\left(W_{lk} * X_{lT} + b_{lk}\right). \quad (15)$$

Among H_{sk} refers to the number of the k short period input matrix of filter pairs X_{sT}, the result after scanning, and H_{lk} is represents the outcome of the long-period convolution component operation. W_{sk}, b_{sk}, W_{lk} and b_{lk} are the corresponding weight parameter. Therefore, the output of the short period convolution operation can be expressed as $V_s \in R^{d_s \times a \times k_s}$, the output of the long period convolution operation is expressed as $V_l \in R^{d_l \times a \times k_l}$. Here, d_s and d_l represent the number of filters in the final convolution layer of both the short-period and long-period convolution components for the characteristics of internet financial non-stationary time series are denoted by their respective symbols. This completes the design of the CNN's medium-period partial convolution component.

2.3.3. Residual Network Design

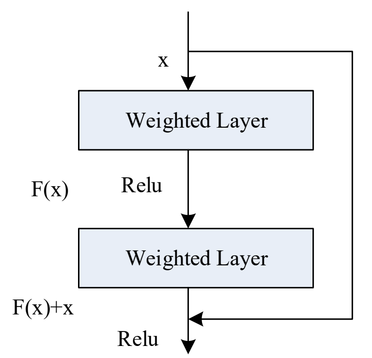
After completing the design of the nearest part of the convolution component and the periodic part of the convolution component, it is necessary to further strengthen the performance of the CNN structure. The residual learning method is helpful to solve the degradation problem and to build a more powerful and effective CNN structure. In short, two or more consecutive neural network layers form a stack layer, and a shortcut connection is added to the stack layer, that is, input skips the stack layer and connects directly to the output position of the stack layer. The process of implementing quick connections is called IdentityMapping. A ResidualUnit consists of a stack layer and a quick connection. In particular, the residual unit's output can be expressed as $H(x)$, where x represents the input of residual unit, $F(x) = H(x) - x$ represents a residual mapping. This paper focuses on $F(x)$ Fit. Compared with direct $H(x)$ fitting, residual mapping $F(x)$ is exhibits greater sensitivity to output variations, resulting in a wider adjustment range for parameters. This acceleration in learning speed significantly enhances the network's optimization performance.

The complete structure of a residual unit is depicted in Figure 4, and its forward propagation process can be formally defined as follows:

$$y = V_s V_l F(x, W_i) + x. \tag{16}$$

Among them, x and y input vector and output vector of the residual unit are represented accordingly, function $F(x, W_i)$ represents the residual mapping to be learned. The stack layer in Figure 4 has only two layers, so $F = W_{2\sigma}(W_1 x)$, σ indicates the ReLU activation function. Through quick connection $F + x$ operation, and ReLU activation again, that is $\sigma(y)$.

Figure 4
Residual Unit Structure



In Formula (17), the shapes of x and F must be equal. If they are not equal (such as when the input or output channels change), linear projection can be performed through quick connection W_s, to make shape matching:

$$y' = F(x, W_i) + W_s\left[y - V_s V_l F(x, W_i)\right]. \tag{17}$$

Complete the design of residual network according to Formula (17).

2.3.4. Internet Finance Non-stationary Time Series Prediction Results

The final output result of the characteristics after input fusion in the CNN network with added residual network is:

$$y' = F(x, W_i) + W_s\left[y - V_s V_l F(x, W_i)\right]. \tag{18}$$

where, $\hat{Y}$ represents the prediction results of Internet financial non-stationary time series finally output by CNN, $\otimes$ is the Hadamard product, that is, the element-by-element multiplication of the tensor. Y_c, Y_s, Y_l, are the non-stationary time series feature information of Internet finance extracted from the nearest part of components, short period branches and long period branches, respectively. W_c, W_s, W_l are adjustable parameter, which can adjust the degree of influence by different branches.

3. Experimental Analyses

3.1. Experimental Objects

This study utilizes the closing data of the Shanghai A-share index, spanning from January 1, 2010, to January 1, 2020, encompassing a decade of comprehensive trading information. Meanwhile, trading information from the same time period was selected in the foreign exchange market. Thus, the data from the stock market and foreign exchange market are summarized to form an experimental dataset. During the experiment, 80% of the data was used as the training set to learn the historical trends and patterns of Shanghai trading. The remaining 20% of the data, 10% used as the test set to validate the effectiveness of method, and 10% used as the validation set for hyperparameter adjustment and model selection during training to avoid overfitting.

3.2. Experimental Data

The relevant parameter settings during the experimental process are shown in Table 1.

In the parameters shown in Table 1, three convolutional layers are set to capture multi-level features and improve prediction accuracy. Use convolution kernels of different sizes to capture features at different time scales in financial time series. The pooling kernel size is (2,2) to reduce data dimensionality, prevent overfitting, and introduce translation invariance. 128 nodes are set up in the fully connected layer to integrate features and generate predictions, while avoiding overfitting. Choose a learning rate of 0.001 to ensure stable convergence of the model and avoid

Table 1
Experimental Parameter Settings

The parameter name		Parameter values
EMD	MaxNumIMF	5
	SiftRelativeTolerance	0.2
CNN	Number of convolutional layers	3
	Convolutional kernel size	[3, 5, 7, 9]
	Number of convolution kernels	[32, 64, 128]
	Pooled Core Size	(2, 2)
	Number of fully connected layer nodes	128
	Learning rate	0.001
	Batch size	64
	Lterations	100

oscillations during gradient descent. Set the batch size to 64 to strike a balance between computational efficiency and memory usage. Set the number of iterations to 100, allowing the model to fully learn data features to achieve optimal performance.

It can be inferred from this that, the convolutional kernel in CNN is a key component of feature extraction. The size of the convolutional kernel determines the features of different scales in the time series that the network can capture. In financial non-stationary time series, these characteristics may include short-term market volatility, medium-term trend changes, and long-term cyclical patterns.

To assess the impact of CNN convolution kernel size on predicting non-stationary time series in this study, conducted predictions using kernel sizes of 3, 5, 7, and 9, respectively. The target data was the non-stationary time series of internet finance from January 1, 2010, to January 1, 2011. Figure 5 illustrates the prediction accuracy achieved with different convolution kernel sizes for both training and test sets. Additionally, Table 2 presents the prediction results, including root mean square error, determination coefficient, and correlation coefficient, for each kernel size.

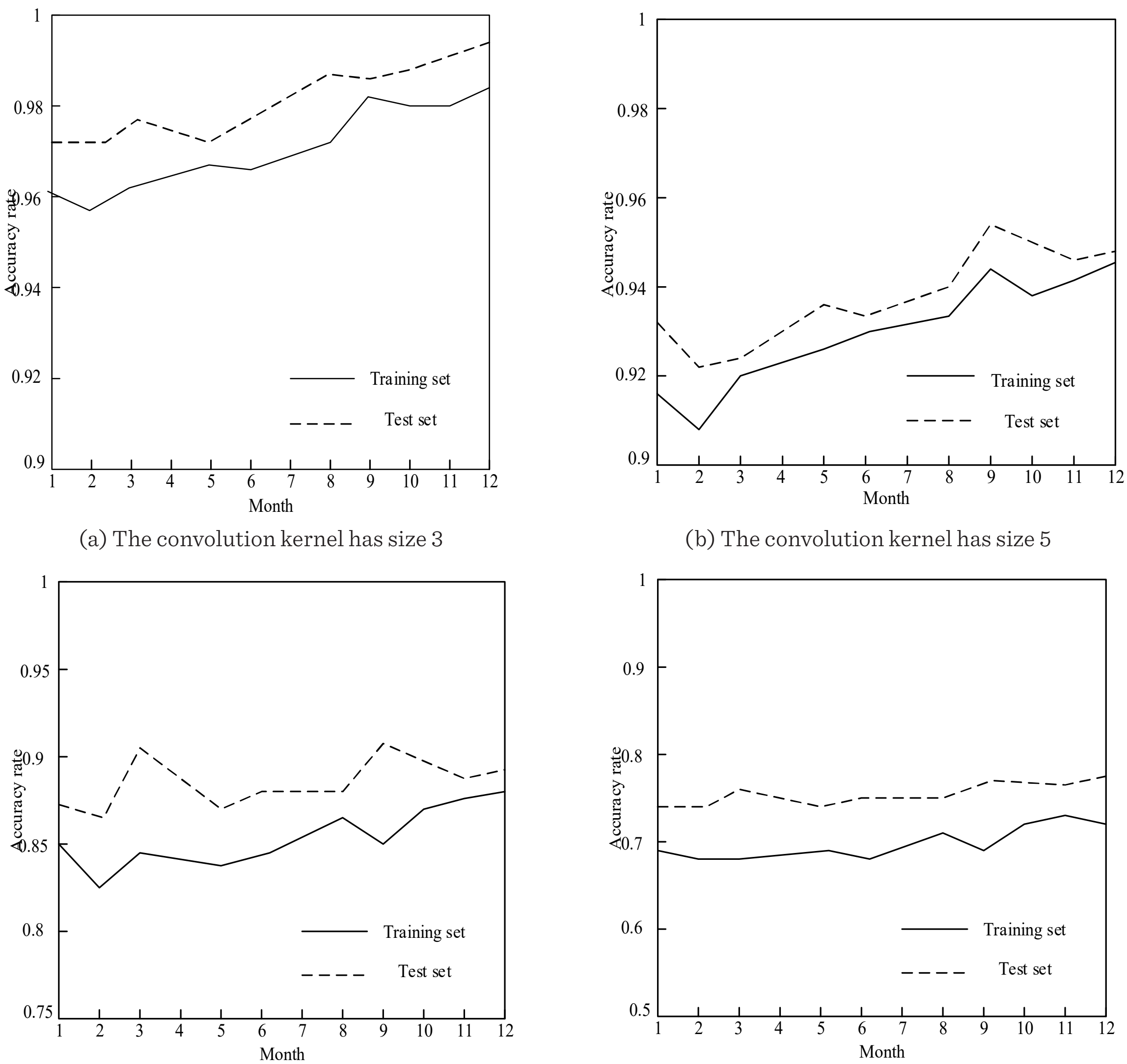
From Figure 5 and Table 2, it is evident that as the size of the convolution kernel decreases, the root mean square error also diminishes. Specifically, when the kernel size is 3, the root mean square error reaches its minimum value. Similarly, as the kernel size decreases, the correlation coefficient increases, approaching a value closer to 1. This trend suggests that with smaller kernel sizes, the predicted stock trends align more closely with the original changes. On the other hand, the determination coefficient decreases as the kernel size gets smaller, yet it remains closer to 1, indicating that the prediction results exhibit a better fit when the kernel size is small. In summary, smaller convolution kernel sizes lead to superior prediction outcomes. Based on these experimental findings, a convolution kernel size of 3 yields the most optimal prediction results for the Internet financial non-stationary time series within the convolution neural network prediction model.

Therefore, a statistical test is conducted on the method proposed in this article. During the inspection process, the proposed method will be used as the benchmark model before application, and mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) will be used as evaluation indicators. Among them, using MSE as an evaluation metric helps to compare the performance of the model on the training and testing sets, in order to check whether the model is overfitting or underfitting. RMSE provides a more intuitive measure of error, which helps determine the degree of deviation between predicted and actual values. The MAE directly provides the average degree of deviation between predicted values and actual values.

Verify the results in Table 3. Among them, t-tests were conducted on the MSE before and after the application of the proposed method, and the p-value was less than 0.05, indicating that the MSE index of the proposed method was significantly better after application than before. The t-test was conducted on the RMSE before and after the application of the proposed method, and a p-value less than 0.05 was obtained, indicating that the RMSE index of the proposed method is significantly better after application than before. Perform a U-test on the MAE before and after the application of the proposed method, and obtain a p-value less than 0.05, indicating that the proposed method is effective in MSE after application Lower error values were achieved in evaluation indicators such as RMSE and MAE, and the statistical test results also showed that these performance differences were significant. This validates the effec-

Figure 5

Prediction accuracy of training set and test set



(a) The convolution kernel has size 3

(b) The convolution kernel has size 5

(c) The convolution kernel has size 7

(d) The convolution kernel has size 9

Table 2

Indicators of prediction results of different convolution kernel

Convolution kernel size	Root mean square error	Correlation coefficient	Coefficient of certainty
3	6.40E-04	0.8996	1.0046
5	7.14E-04	0.8916	1.2133
7	7.21E-04	0.8863	1.1515
9	7.23E-04	0.8784	1.1293

Table 3

Statistical test results

Evaluation indicators	Before the application of the proposed method	After the application of the proposed method
MSE	0.016	0.010
RMSE	0.126	0.100
MAE	0.102	0.080

tiveness of the deep learning based network finance non-stationary time series prediction algorithm.

To validate the effectiveness of algorithm in predicting non-stationary time series related to Internet finance, present short-term predictions for a specific stock over the period from January 1, 2012, to January 1, 2013 in Figure 6. Additionally, Figure 7 displays the long-term prediction results for the overall Shanghai A-share index, covering the period from January 1, 2010, to January 1, 2020. These results demonstrate the algorithm's capability in capturing trends and patterns in both short- and long-term scenarios.

As observed in Figures 6-7, the algorithm employed in this paper exhibits remarkable performance in predicting the non-stationary time series of a stock within a year. The predicted outcomes align closely with the actual results, strongly indicating that the algorithm demonstrates a highly evident short-term prediction effect for the non-stationary time series related to Internet finance. In the long-term non-stationary time series prediction of the Shanghai Stock Exchange Index for 10 years, there were some fluctuations in the second half of 2014 and 2017-2019, and the prediction results for the rest of the time were very accurate, which proved that the algorithm in this paper is also very accurate in the long-term prediction of Internet finance.

This is because the EMD method used in this article can decompose complex non-stationary time series information into a series of intrinsic mode functions (IMFs), which contain different frequency components from high frequency to low frequency. By removing noise from high-frequency components, key information and structures in the data can be effectively preserved and enhanced, making subsequent analysis and prediction more accurate.

To validate the practicality of the algorithm proposed in this paper, conducted a comparison study with var-

Figure 6

Non-stationary time series prediction results of a stock in one year

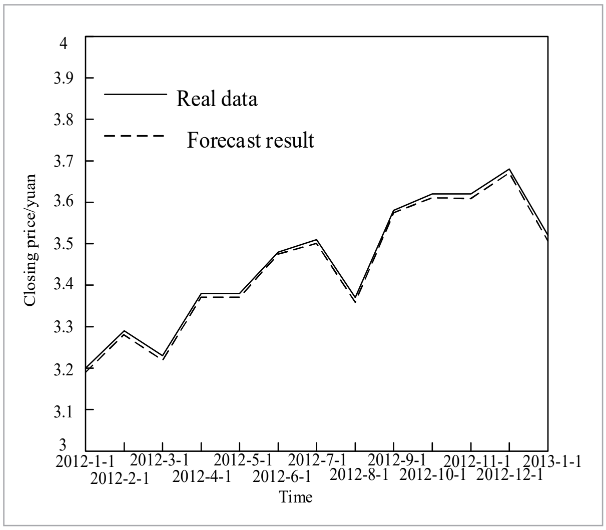
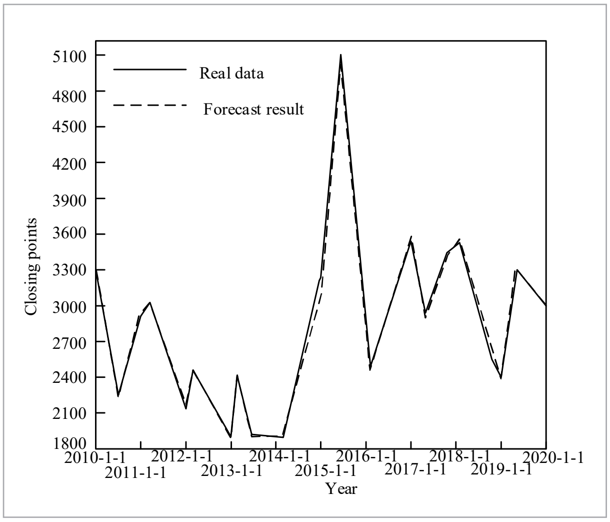


Figure 7

Non-stationary time series prediction results of Internet finance in 10 years



ious other prediction methods, including the DCC-GARCH model combined with a multivariate neural network (reference [6] algorithm), Boltzmann entropy neural network for financial time series prediction (reference [9] algorithm), interval 2 fuzzy set time series data-driven partition prediction (reference [12] algorithm), the method of basic information and neural network (reference [13] algorithm), and robust depth adaptive input normalization for financial time series prediction (reference [11] algorithm). The comparison results are summarized in Table 4. In this experiment, employed three key parameters to comprehensively assess the prediction outcomes: root relative square error (RSE), correlated absolute error (RAE), and correlated empirical coefficient (CORR). Among them, RSE is a commonly used error evaluation metric used to measure the performance loss between predicted and actual results. It represents the proportional difference between the actual value and the predicted value. RAE is one of the metrics used to measure the accuracy of predictive models. It represents the ratio of the absolute difference between the predicted value and the actual value to the average of the actual values. CORR is a statistical indicator that measures the strength and direction of the linear relationship between two variables. Its value range is between -1 and 1, where 1 represents complete positive correlation, -1 represents complete negative correlation, and 0 represents no linear relationship. Using RSE The three parameters RAE and CORR are used to validate the practicality of the algorithm, which can comprehensively evaluate the predictive performance and accuracy of the algorithm from different perspectives. RSE and RAE focus on the degree of difference between predicted and actual results, while CORR focuses on the strength and direction of the linear relationship between predicted and actual values. When evaluating experimental results, lower RSE and RAE indicate better performance, while higher CORR indicates better model performance. The combination of these three parameters can provide more comprehensive and accurate algorithm performance evaluation results.

Table 4

Comparison of prediction results of different methods

Method	Index	Forecast duration		
		Week	Month	Year
Textual algorithm	RSE	0.505	0.545	0.551
	RAE	0.356	0.421	0.433
	CORR	0.864	0.853	0.829
References [6]	RSE	0.598	0.609	0.631
	RAE	0.452	0.473	0.501
	CORR	0.779	0.757	0.663
References [9]	RSE	0.583	0.611	0.627
	RAE	0.441	0.466	0.484
	CORR	0.808	0.762	0.749
References [12]	RSE	0.601	0.648	0.733
	RAE	0.576	0.608	0.771
	CORR	0.694	0.653	0.642
References [13]	RSE	0.539	0.557	0.593
	RAE	0.417	0.458	0.593
	CORR	0.799	0.752	0.739
References [11]	RSE	0.602	0.658	0.713
	RAE	0.588	0.543	0.426
	CORR	0.671	0.681	0.601

It can be seen from Table 4 that the RSE and RAE indicators of other algorithms are higher than those of the algorithm in this paper. Among them, the RSE and RAE values of the long-term Internet financial non-stationary time series predicted by the algorithm in reference [12] are the highest, 0.733 and 0.771, respectively. It is much higher than 0.551 and 0.433 of this algorithm. The CORR value predicted by the algorithm in this paper is also the highest among all algorithms. The lowest CORR value predicted by the method in reference [11] for the long-term Internet financial non-stationary time series is 0.601, which shows that the prediction performance of the algorithm in this paper for Internet financial non-stationary time series is far superior to other algorithms. This is because the knowledge graph method was used in the paper to extract transaction and market features from noise free financial non-stationary time series, and the two were integrated. This feature extraction and fusion method can capture key factors that affect financial time series, providing more comprehensive and rich information for prediction models. On this basis, the structurally optimized CNN is used to automatically learn complex patterns and features in the data, and can effectively process multi-dimensional input data. The CNN model used in the paper, after appropriate adjustment and optimization, can adapt well to the task of financial time series prediction, thereby improving the accuracy of prediction.

4. Conclusion

The algorithm introduced in this paper utilizes deep learning to predict the non-stationary time series associated with Internet finance. By incorporating the residual network concept within the CNN, enhance the depth of the CNN network layer, thereby enhancing the prediction accuracy for non-stationary time series in the realm of Internet finance. Experiments show that the algorithm in this paper is more accurate for single stock or the whole stock market, long-term or short-term prediction results, and has higher prediction ability and stronger practicability, which can improve the prediction ability of financial non-stationary time series. Although the proposed method demonstrates high predictive ability and practicality, the training and optimization of the model heavily rely on a large amount of annotated data, which may be a challenge in the financial field as high-quality historical financial data is often difficult to obtain and costly. In addition, the performance of the model in predicting extreme market events has not been fully validated, which may involve more complex nonlinear and dynamic relationships. To address the aforementioned issues, future approaches can include outlier handling, ensemble learning, developing specialized models, real-time updates and adjustments, introducing external information, and implementing risk management and stress testing. These strategies are expected to improve the generalization ability of the model and the accuracy of predicting extreme events.

References

1. Al-Haija, Q. A., Alsulami, A. A. High-Performance Classification Model to Identify Ransomware Payments for Heterogeneous Bitcoin Networks. Electronics, 2021, 10(17), 2113. https://doi.org/10.3390/electronics10172113
2. Aryal, S., Nadarajah, D., Rupasinghe, P. L., Jayawardena, C., Kasthurirathna, D. Comparative Analysis of Deep Learning Models for Multi-Step Prediction of Financial Time Series. Journal of Computer Science, 2020, 16(10), 1401-1416. https://doi.org/10.3844/jcssp.2020.1401.1416
3. Ashraf, S., Abdullah, S., Khan, S. Fuzzy Decision Support Modeling for Internet Finance Soft Power Evaluation Based on Sine Trigonometric Pythagorean Fuzzy Information. Journal of Ambient Intelligence and Humanized Computing, 2020, 12(2), 3101-3119. https://doi.org/10.1007/s12652-020-02471-4
4. Chen, M., Zhou, P. 2CFastICA: A Novel Method for High-Density Surface EMG Decomposition Based on Kernel Constrained FastICA and Correlation Constrained FastICA. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2024. https://doi.org/10.1109/TNSRE.2024.3398822
5. Damaševičius, R., Jovanovic, L., Petrovic, A., Zivkovic, M., Bacanin, N., Jovanovic, D., Antonijevic, M. Decomposition-Aided Attention-Based Recurrent Neural Networks for Multistep-Ahead Time-Series Forecasting of Renewable Power Generation. PeerJ Computer Science, 2024, 10. https://doi.org/10.7717/peerj-cs.1795
6. Fatima, S., Uddin, M. On the Forecasting of Multivariate Financial Time Series Using Hybridization of DCC-GARCH Model and Multivariate ANNs. Neural Computing and Applications, 2022, 34(24), 21911-21925. https://doi.org/10.1007/s00521-022-07631-5
7. Hu, Y. M., Pan, Q., Tian, N. L. Source Enumeration Method Based on Ensemble Empirical Mode Decomposition Under Low SNR. Computer Simulation, 2022, 39(9), 186-189, 207.
8. Jain, A., Sharma, Y., Kishor, K. Solid-State Technology Financial Supervision and Management System Using ML Algorithm. Solid State Technology, 2020, 63(6), 18974-18982.
9. Luca, G., Domenico, S. Forecasting Financial Time Series with Boltzmann Entropy Through Neural Networks. Computational Management Science, 2022, 19(4), 131-147. https://doi.org/10.1007/s10287-022-00430-2
10. Neena, P., Rahul, D., Mn, R. Improving the Governance of Information Technology: Insights from the History of Internet Governance. Journal of Informa-

tion Technology, 2022, 37(3), 266-287. https://doi.org/10.1177/02683962211054513

11. Nikolaos, Passalis, J., Kanniainen, M., Gabbouj, A., Iosifidis, A., Anastasios, T. Forecasting Financial Time Series Using Robust Deep Adaptive Input Normalization. Journal of Signal Processing Systems, 2021, 93(10), 1235-1251. https://doi.org/10.1007/s11265-020-01624-0
12. Pinto, A. C. V., Fernandes, T. E., Silva, P. C. L., Guimares, F. G., Wagner, C., Eduardo, P. D. A. Interval Type-2 Fuzzy Set-Based Time Series Forecasting Using a Data-Driven Partitioning Approach. Evolving Systems, 2022, 13(5), 703-721. https://doi.org/10.1007/s12530-022-09452-2
13. Pradeepkumar, D., Ravi, V. Financial Time Series Prediction: An Approach Using Motif Information and Neural Networks. International Journal of Data Science, 2020, 5(1), 79-109. https://doi.org/10.1504/IJDS.2020.109489
14. Rožanec, J. M., Fortuna, B., Mladenić, D. Knowledge Graph-Based Rich and Confidentiality Preserving Explainable Artificial Intelligence (XAI). Information Fusion, 2022, 81, 91-102. https://doi.org/10.1016/j.inffus.2021.11.015
15. Sidekerskienė, T., Woźniak, M., & Damaševičius, R. Nonnegative Matrix Factorization Based Decomposition for Time Series Modelling. In Computer Information Systems and Industrial Management: 16th IFIP TC8 International Conference, CISIM 2017, Bialystok, Poland, June 16-18, 2017, Proceedings 16, 604-613. Springer International Publishing. https://doi.org/10.1007/978-3-319-59105-6_52
16. Talib, M. A., Alsaafin, A., Medjden, S. M. Application of Quality in Use Model to Evaluate the User Experience of Online Banking Software. Journal of Cases on Information Technology, 2020, 22(2), 34-51. https://doi.org/10.4018/JCIT.2020040103
17. Zimon, G., Sobolewski, M., Lew, G. An Influence of Group Purchasing Organizations on Financial Security of SMEs Operating in the Renewable Energy Sector-Case for Poland. Energies, 2020, 13(11), 1-17. https://doi.org/10.3390/en13112926