JEAS
Journal of Economics and Administrative Sciences
E-ISSN: 2148-1792 P- ISSN: 1302-2024
Volume 3, Supplement Issue 3
www.jeasweb.org

A Hybrid Approach of the Empirical Mode Decomposition and Factorization Machine based Neural Network and Filtering of Information in the Capital Market

Alireza Danesh
M.Sc. Student of Software Engineering, Kooshyar University of Rasht, Rasht, Iran.

ABSTRACT
The capital market has become a suitable platform for investment because of its features such as not needing high capital and high profitability. For these reasons, these markets are growing rapidly. This has led to higher demand for information, more effort for prediction and the development of new models to predict the future of the market. Predicting the capital market is a difficult and challenging task due to the large number of investors with different views and the effectiveness of a large number of variables which all cannot be studied in practice. For these reasons, new prediction models are presemted and previous prediction models are upgraded or combined. In this regard, in order to improve the forecast accuracy of stock market prices, a new hybrid approach has been introduced, in which two empirical mode decomposition (EMD) and the Empirical Mode Decomposition and Factorization Machine based Neural Network (EMD2FNN) are combined with long short-term memory (LSTM). In technical analysis, analysis by financial time series models is a kind of nonlinear and non- stationary random signal, that using EMD and the empirical mode decomposition in a set with Gaussian white noise, empirical mode decomposition based neural network (EMD2NN) model and the wavelet de-noising-based back propagation (WDBP) neural network model can be combined in several different time through Factorization Machine based Neural Network based on frequency modulation to ensure the effect of historical data on pre-results Nasal, LSTM prediction models have been developed for each characteristic series of sediment, CEEMDAN (EMD) and EMD2FNN. The final results of prediction are obtained by reconstructing each forecast series.
Keywords: Prediction, Long Short-Term Memory, Time Series Analysis, Neural Network, EMD2FNN & LSTM

Introduction
Financial markets have become one of the most popular areas of investment in recent times due to their unique characteristics such as large capital needless, simplicity and low cost of their transactions and lack of default risk. Governments and states by providing huge budgets for state affairs by pooling small and large capital, have always helped to expand these markets by trying to attract savings by passing various laws, including tax exemptions. For these reasons, a large number of investors have entered to these markets and these markets are growing rapidly. This large number of investors have entered these markets to make a profit and therefore have always been looking for ways to increase their profit. This has led investors to always anticipate future events and prices in the market and thus make a profit; that is why, as these markets grow, so many different models have emerged and are expanding for prediction (Hafezi, 2015).

Stock price changes are nonlinear and non-stationary. Therefore, it is very difficult to predict price fluctuations. Since the stock price forecast is related not only to the *current available data* but also to the Historical data, if only the Historical data is available, the information performed by the data disappear at the early time. Recurrent neural networks, unlike traditional neural networks, establish a connection between hidden units, which allows the network to retain recent events in memory and deal with specifications related to pre- and post-memory features and is very convenient to predict time series. This is an improved model of RNN, LSTM (Pradipakumar 2017)

Stock price forecasting models can be generally divided into three categories: technical analysis, fundamental analysis, and artificial intelligence algorithms analysis. Technical analysis predicts the trend of past stock price changes by assuming that past events are repeated in the future. In fact, chartists or technical employees believe that the only thing that changes the stock price is the amount of demand and supply in the market. In fact, they believe that stock prices are affected by any other fundamental economic factor that changes, these factors only affect supply and demand in the market, and therefore by predicting these values can also predict prices.

The two general models used by this category are pattern matching and the use of indicators; Patterns are in fact price trends in the past that can be used to predict the future, given the belief that they will be repeated in the future. However, this is not easy due to the involvement of various market factors and the relationships between them. (Hey 2016) It seems that the use of more sophisticated computational tools and algorithms such as fuzzy neural networks in modeling nonlinear processes that result in price and stock trends can be very useful.

Therefore, in this study, we try to use capital market variables (total index, P/E ratio, earnings per share, etc.), economic variables (exchange rate, oil prices, gold prices, etc.) and technical analysis indicators. (RSI, SO, MACD, etc.). A neural network should be designed that has the ability to achieve an optimal answer close to the real answer. Indicators are mathematical models that use price indicators such as opening and closing prices and trading volume to predict supply and demand, and ultimately price.

Artificial intelligence algorithms which is rapidly growing among investors, is in fact a combination of all the prediction methods mentioned with the ability to fit high-grade nonlinear curves. Among artificial intelligence algorithms, the use of neural networks in the subject of prediction is very high. This is due to the ability of the neural network to work with a large number of variables, very precise fit of the time series, not being affected by outliers, no limit for a certain degree of nonlinearity and flexibility of the network against changes in model parameters. Kara et al. (2011) prefered neural networks to classical models and other artificial intelligence algorithms in their study.

Lou et al. (2014) after examining the reasons for overfitting of poor generalization of neural networks, by applying changes in the neural network and applying a constructive Decay RBF at neural networks succeeded in building the neural network with more precision and, of course, fewer neurons in the hidden layer of the neural network, while testing the results in the real world.

In another article, Nguyen introduced the use of HMM in trading stocks in 2018 depending on stock price forecasts. This process begins with the use of four criteria as a way to identify the optimal number of states for HMM. Next, it was used to predict monthly closing prices. The results clearly confirm that HMM performed better than the traditional approach in stock trading and forecasting.

In 2019, Chandar introduced a new hybrid method called wavelet-adaptive network-based fuzzy inference system (WANFIS) that used the past information of several firms. The results showed that this method has received higher accuracy than predicted compared to other methods.

In previous scientific research, many models have been proposed to predict stock prices, but each of the models presented in previous research has different advantages and disadvantages. Therefore, we intend to improve the shortcomings of previous models and eliminate their problems as much as possible.

The aim is to introduce a new, two-step hybrid approach nof Empirical Mode Decomposition and Factorization Machine based Neural Network (EMD2FNN) and to filter information through the LSTM "gateway" to extract more useful information from historical data.

Research Methods

Programming Languages

Neural network tools were selected in MATLAB software to implement neural networks. Then, in order for the data to be received directly from the MetaTrader software and the results to be visible on the same software, the Metaweb Query Language 4 was examined and used.

Data Collection and Database Formation

The data required for this study are high, low, clos, open, bid, ask, and trading volume; technical analysis indicators are calculated and used for forecasting using this data. We also test the model among several major Forex currency pairs for better validation. For each pair of currencies, we select one currency pair and do the following research for it. Currency pairs are described in Table 1. The data required to enter the network and train and test it were provided by MetaTrader software.

Table 1: The Main Currency Pairs of the Forex Market Selected for Research

No.	The name of the currency pair	Abbreviation	Alias in Forex
1	Euro - US Dollar	EUR/USD	Fiber
2	British Pound - US Dollar	GBP/USD	Cable
3	Australian Dollar - US Dollar	AUD/USD	Aussie
4	New Zealand Dollar - US Dollar	NZD/USD	Kiwi
5	US Dollar - Canadian Dollar	USD/CAD	Loonie
6	US Dollar - Japanese Yen	USD/JPY	Gopher
7	US Dollar - Swiss Franc	USD/CHF	Swissy

Data Preprocessing

According to the data mining methods, first the data should be pre-processed and a line reduction should be done, the necessary information should be made and the variables and parameters should be identified, and then a column reduction should be based on the data mining methods to prevent over-fitting of the neural network.

Removing Outliers

Bollinger bands are used to identify confusing data. Bollinger bands are actually volatility that are priced on a moving average. As the data scatter increases, these bands open, and as the scatter decreases, the bands converge. The general formula for Bollinger upper and lower bands in this study is as follows:

$$Upper\ Band = 20 - day\ SMA + (20 - day\ standard\ deviation * 2) \qquad 1$$

$$Lower\ Band = 20 - day\ SMA - (20 - day\ standard\ deviation * 2) \qquad 2$$

In the present study, after removing outlier data through the bollinger bands, this data is removed from the database and therefore the record that contains this data is also completely deleted from the data.

Making the Required Information

In this research, technical analysis indicators as well as similar time series are used as neural network inputs. In this section, first the indicators studied in this research and then the similar time series that are identified by time series data mining are identified. This step is done after reducing the line because this information is made or identified based on the records obtained in the first step, and therefore if there is confusing data or skewed data among the initial data. They also change this information and variables and confuse the data.

The indicator was tested in the tester strategy in MetaTrader software. After testing the indicator in the tester strategy, it was also tested online to see possible structural and programming defects.

After several months of testing, the existing errors were determined and then the defect elimination indicator was performed to prepare for the final test.

At this stage, technical indicators for forecasting stock prices are created using stock price information including high, low, clos and volume information including trading volume.

Column data reduction using stepwise regression method: To determine which indicators can predict the share price, we use the stepwise regression method. Stepwise regression method, both forward and backward, measures the ability of each indicator to predict the price.

Choose from Similar Time Series

In the previous step, the final indicators were selected. At this stage, one must choose one of the time series with the optimal lag. The correlation criterion will also be used for this purpose. Based on this, the time series of the stock price is selected that has the highest correlation for the closing prices with the desired share. Therefore, at the end of this stage, the last input of the neural network to predict the stock price is identified and finalized.

Network Structure

In this paper, we use a 4-layer neural network as shown in the figure below.

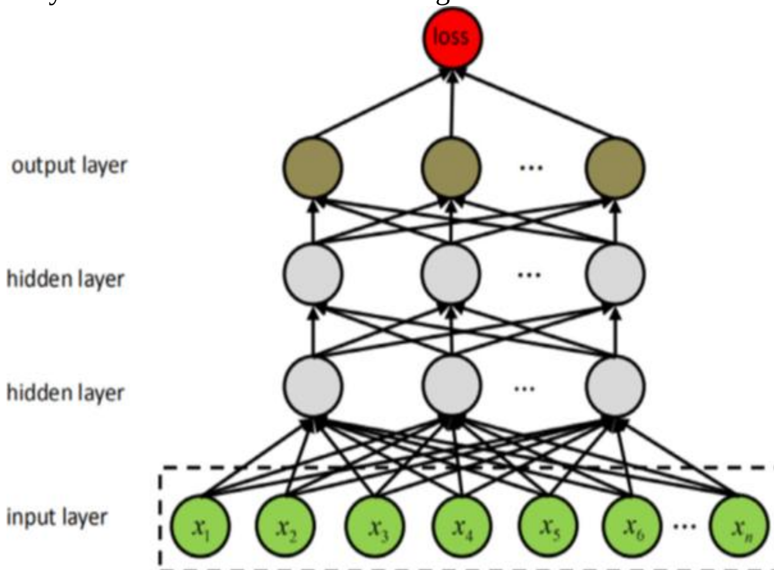


Figure 1. Neural Network Architecture

The four layers include an input layer, two hidden layers and an output layer. We want to note that even and odd layers, which are often added when using neural network (CNN) in image processing, are not included in the NN structure.

All indeterminate weights in the NN model are frequently updated according to the Stochastic Gradient Descent (SGD) method in the reverse diffusion process for gradient. This trend can be described as a specific case of the improved FNN model.

FM was originally introduced for participatory recommendations; FMs, like backup vector machines, constitute a general class of predictors that are able to estimate reliable parameters assuming a very high deficit. FM has a fundamental advantage in learning reciprocity due to learning in the hidden space, and this is one of the main reasons why we choose FM in our model.

In the neural network above, the input layer neurons determine the input properties, each neuron coming out of the hidden layers is calculated by combining the multiplied binding weights with the input values and the linear passive function. The objective function is to consider nonlinearity in the model. The layer calculates the error loss between the exercise and the estimated values for adjusting the connection weights. Suppose there are n inputs, output neurons m_1 in hidden layer 1, output neurons m_2 in hidden layer 2 And output neurons o In the output layer, the neural network prediction process can be described in the following four steps:

1. Hidden layer 1: Nerve cells in the first hidden layer are divided into two parts and the outputs of the first activation layer are calculated by the following scheme:

$$y_j^I = f(\textstyle\sum_{i=1}^{n} w_{ij}^I x_i), (j = 1, 2, \ldots, m_1 - k_1)$$

$$y^I_{m_1-k_1+j} = f\left(\frac{1}{2}\left(\left(\sum_{i=1}^{n} v^I_{ij} x_i\right)^2 - \sum_{i}^{n} (v^I_{ij})^2 x_i^2\right)\right), \ (j = 1, 2, \ldots, k_1),$$

Where, x_i represents the property i, y_j^I I is the value of the node j, w_{ij}^I is the indeterminate linear weight of the input property i and the output neurons j, $v^I{}_i \in R^{k_1}$ is the unspecified hidden weight corresponding to the property x_i, k_1 1 is the user specified dimensions of v$v^I{}_i$, and f is the activation function.

2. Hidden Layer 2: The output of the second hidden layer includes the following two:

•single-FNN:

$$y_j^H = f(\sum_{i=1}^{m_1} w_{ij}^H y_i^I), (j = 1, 2, \dots, m_2),$$

•multi-FNN:

$$y_j^H = f(\sum_{i=1}^{m_1} w_{ij}^H y_i^I), (j = 1, 2, \dots, m_2 - k_2),$$

$$y_{m_2-k_2+j}^H = f\left(\frac{1}{2}\left(\left(\sum_{i=1}^{m_1} v_{ij}^H y_i^I\right)^2 - \sum_{i=1}^{m_1} \left(v_{ij}^H\right)^2 \left(y_i^I\right)^2\right)\right), (j = 1, 2, \dots, k_2),$$

Where, w_{ij}^H is the indeterminate linear weight for input neurons I and output neurons j, $v_i^H \in R^{k_2}$ is the undefined hidden weight is related to y_i^I, k_2 is specified By the user. Dimensions v_i^H.

Output layer: The output layer in FNN is the same as in the NN model, ie

$$y_j^O = f(\sum_{i=1}^{m_2} w_{ij}^O y_i^H), (j = 1, 2, \dots, o),$$

Where, w_{ij}^O is the indeterminate connection weight, y_j^O is the value of node j of the output layer.

Loss layer: FNN can be applied to a variety of predictive tasks by performing various loss tasks l, including classification, regression and ranking, for regression, a commonly used loss function, loss is a square defined by

$$l(y_i^O, y_i) = \frac{1}{2}(y_i^O - y_i)^2$$

Considering a valuable feature $x \in R^n$,FM estimates the objective by modeling the whole interaction between each pair of features through modulated frequency parameters:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i \cdot x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i \cdot x_j,$$

By specifying input characteristics, studies show that FM can mimic many specific frequency modulation models, such as auditing standard frequency modulation, parallel factor analysis, and SVD. On the other hand, the term factor modulation can be rewritten,

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i \cdot x_j = \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{if} \cdot x_i \right)^2 - \sum_{i=1}^{n} v_{if}{}^2 \cdot x_i{}^2 \right),$$

This means that Equation (16-3) can be calculated at a linear time $O(k \cdot n)$. Due to this property, FM is known as one of the most effective methods for predicting scattered data. This collection has brought successful applications in the industry, and has promised good results in a variety of predictive tasks, such as re-reading, classification and ranking.

Rival Models and Measurement Criteria

Because in this study, the neural network uses past prices and indicators, it will have the same function as time series models and therefore will be compared with ARMA time series models. In addition, the performance of this network is similar to the performance of multivariate regression using the same indicators and prices, and therefore the performance of the neural network will be compared with multivariate regression.

Also, the measurement criteria will be MSE and MAD. Since two variables will be predicted in each of the three competing models, the MSE and MAD of each observation will be equal to the sum of the two outputs, MSE and MAD. The formulas for calculating these two criteria are as follows:

$$MSE = \sum_t \frac{(y_t - \hat{y}_t)^2}{n-1}$$

$$MAD = \sum_t \frac{|y_t - \hat{y}_t|}{n-1}$$

Neural Network Design

The number of neural network inputs includes 8 indicators, two high and low prices of the same time series and two high and low prices of the desired share. Therefore, a total of 12 inputs are considered for the neural network. It was also said that the objective is to forecast high and low prices for a period ahead. Therefore, the number of outputs will be equal to 2.

First, we use EMD, which is very efficient at working with non-stationaryu data, to generate the main series of financial time in several components called intrinsic mode functions (IMF). Each extracted IMF contains narrow-range oscillating patterns and can be viewed as a quasi-stationary component. To reduce the impact of noise on forecasting, EMD and its advanced version, CEEMDANare used to forecast financial time series. Because EMD is a Fourier transform-based signal analysis method, it processes any nonlinear and non-stationary signals adaptively. The main time series is broken down into several subsets under different frequencies using EMD processing, and these sub-series are predicted as model input data, respectively.

Secondly, the FM-based neural method (FNN) is constructed, using values of international monetary fund as input to predict future stock price trends. Due to its technology combination, FM allows FNN to achieve factor interactions between inputs and is efficient in computing due to its linear complexity.

In this study, we support this idea by inserting FM in other hidden layers. We use single-FNN to indicate that FM is used only in hidden layer 1, while multi-FNN is used to indicate that FM is used in other layers. Assume that there are n inputs, output neurons m_1 in hidden layer 1, output neurons m_2 in hidden layer 2, and output neurons O in output layer. Next, we describe the FNN forecasting process.

Results

Neural Network Implementation

Once the neural network is built, it is time for learning. For this purpose, the data in each of the databases are used.

The forecast results obtained from each of international monetary fund and the balance are reconstructed to obtain the final forecast time series.

Table 2 shows the results of statistical analysis of four stock price data: number of indices, average deviation, minimum, maximum, standard of neural networks.

Table 2: Results of statistical analysis of four stock price data

Standard deviation	Max	Min	Mean	Count	Index
477.02	2664.11	676.53	1607.46	2518	S&P500
3068.58	30003.49	11015.84	22000.22	2465	HSI
2372.72	13478.86	3666.41	8198.87	2535	DAX
648.06	5497.90	1706.70	2802.20	2433	SSE

To test the performance of the proposed forecast model, the daily close prices of the S & P500[1], HSI[2], DAX[3] and SSE are selected as the main data, all of which are obtained from Yahoo Finance (https://finance.yahoo.com). Figure 12 shows the main data of the financial time series, which is non-stationary in the short time. Statistical analysis of the main time series data is shown in Table 2. Data for all indicators are from December 13, 2007 to December 12, 2017.

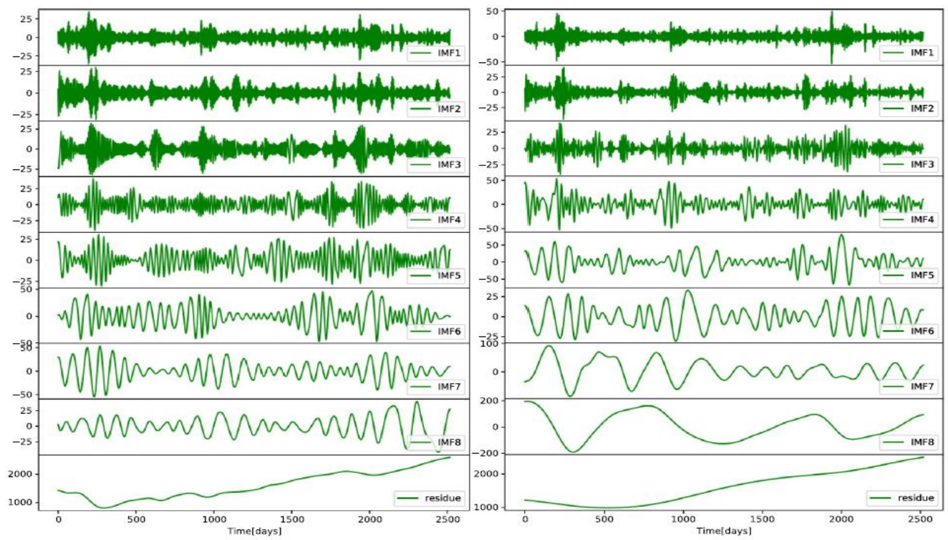
The top 90% data from each time series is selected as the learning set and the second 10% data is selected as the test set. The main financial time series is decomposed into several international money funds and the remainder through EMD and CEEMDAN. According to experimental results, the number of IMFs generated by CEEMDAN is often less than the number generated by the EMD algorithm. To compare the decomposition effect of the two algorithms and reduce the computation rate, we limit the number of IMFs generated by EMD, such as CEEMDAN. Figure 2 shows the results of the analysis of the S & P500 index series.

Each international money fund is organized from high frequency to low frequency. Several international money funds represent high-frequency components or noise in the main series, although difficult to predict, they are also important parts of the overall forecast.

The results of EMD algorithm decomposition have the phenomenon of "mode combination", the international money fund obtained by the CEEMDAN algorithm have a clear frequency difference.

---

[1] *Standard and Poor's Index*
[2] *Hang Sang indexes*
[3] *DAX*



(a) Decomposed by EMD (b) Decomposed by CEEMDAN

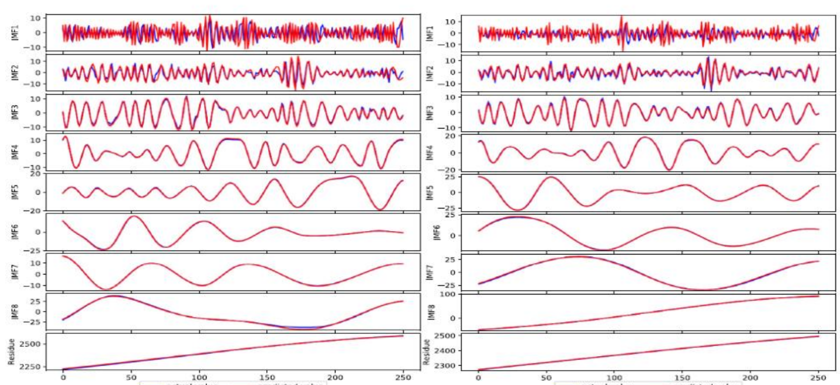
Figure 2- Results of S & P500 index analysis.

Learning Process and Results Forecasting

After decomposition, each subset is divided into a learning set and an experimental set, and then prediction models are constructed for each subset. To obtain the best prediction result for different index data and different subsets, the optimal hyperparameters are selected through experiments, the "window" shows the length of each sample series, and the "period" is the number of learning and the learning period for the rest it is less than other international money fund, because the rest is a smooth curve, this model can easily acquire the features.

Figure 3 shows the prediction results of subsets of S & P500 test dataset. The prediction accuracy of high frequency components IMF1 and IMF2 is relatively low due to the high amplitude and high frequency of components. The remainder shows the long-term trend of the index data and the predicted residual value is close to the actual value. Finally, the prediction results of the four indicators are shown in Figure 4 (a) (c) (e) (g). According to the pictures, the predicted values of the four financial time series are very close or even correspond to the original values.

To evaluate the performance of the proposed models more accurately, three error methods have been adopted, which include the mean absolute error (MAE), the root mean square error (RMSE) and the mean absolute percentage error (MAPE).



(a) EMD-LSTM (b) CEEMDAN-LSTM

Figure 3. Forecast results of Subset for the S & P500 index.

If the MAE, RMSE and MAPE values are smaller, the deviation between the predicted value and the original value is smaller. The MAPE value is more stable than other error measurements, so it is used as the main evaluation indicator.

The prediction error of these two models is shown in Table 4. According to Table 3, the CEEMDAN-LSTM forecast error is four indicators lower than the EMD-LSTM.
Since, CEEMDAN extracts more efficient features from the original data, the prediction error is smaller.

Table 3 Prediction error of two models

Index	Model	MAE	RMSE	MAPE
S&P500	EMD-LSTM	5.0611	6.2794	0.2098
	CEEMDAN-LSTM	3.9177	4.8291	0.1617
HSI	EMD-LSTM	89.2689	118.9504	0.3418
	CEEMDAN-LSTM	71.8757	98.2410	0.2768
DAX	EMD-LSTM	36.1955	45.2663	0.2924
	CEEMDAN-LSTM	24.8473	33.3531	0.2010
SSE	EMD-LSTM	8.3783	10.8478	0.2587
	CEEMDAN-LSTM	6.8621	8.7431	0.2116

To better explain the performance of the prediction model, the prediction value of the CEEMDAN-LSTM model and the principal value are analyzed using linear regression.

Table 4- Parameters in linear regression analysis

Parameter	S&P500	HIS	DAX	SSE
a	0.997	1.0115	1.0007	0.998
b	10.029	-306.57	-12.211	8.8263
R^2	0.9985	0.9982	0.996	0.9932

If a is close to 1, the deviation between the predicted value and the original value is smaller. The linear regression of the experimental results is shown in Figure 15 (b) (d) (f) (h). The parameters in linear regression analysis are also given in Table 5. The gradient a for linear regression of each index is close to 1 and the coefficient of determination R^2 is close to 1, which indicates the predicted value of this model is vry close to the original value.

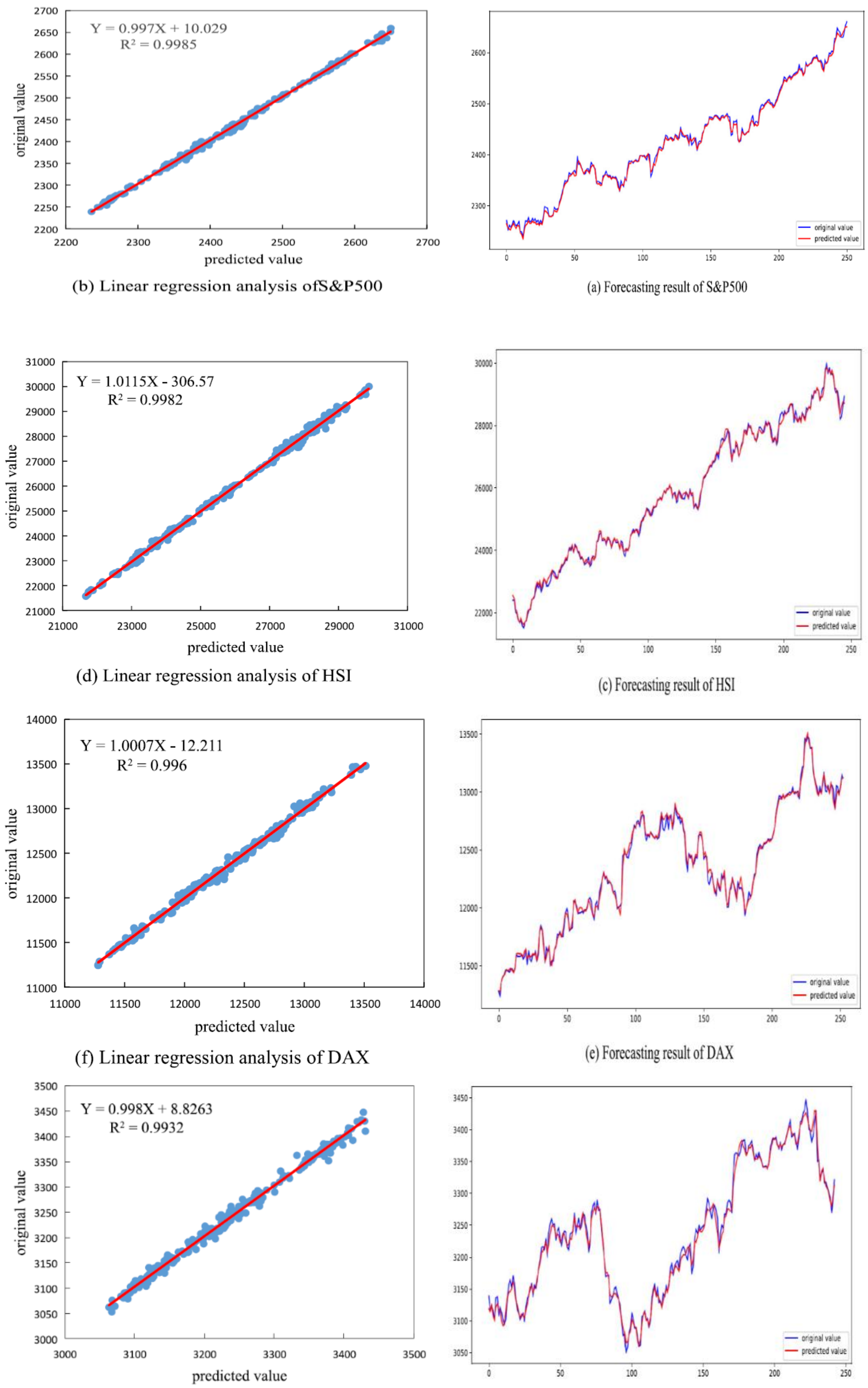
Comparison with other models

In this section, to evaluate the performance of the proposed method, we compared the prediction effect of several models including LSTM, SVM, CEEMDAN-SVM, CEEMDAN-MLP and CEEMDAN-LSTM. All models use the same dataset.

The original data is used directly as input data for the LSTM and SVM prediction models. The other three hybrid models use the CEEMDAN algorithm with series decomposition. SVM has been widely used for time series forecasting. SVM maps input features to a high-dimensional space and uses a linear regression model in this high-dimensional space to achieve it.

Table 6- Comparison of prediction measures of different models

Index	Model	MAE	RMSE	MAPE
S&P500	LSTM	14.8064	18.2321	0.6115
	SVM	7.2209	10.2448	0.3993
	CEEMDAN-SVM	5.4011	6.8118	0.2213
	CEEMDAN-MLP	5.0199	6.1412	0.2051
	CEEMDAN-LSTM	3.9177	4.8291	0.1617
HSI	LSTM	169.0257	215.4280	0.6472
	SVM	147.6551	192.2751	0.5678
	CEEMDAN-SVM	102.6629	129.8097	0..3984
	CEEMDAN-MLP	89.7364	114.6355	0.3416
	CEEMDAN-LSTM	71.8757	98.2410	0.2768
DAX	LSTM	83.7904	109.5599	0.6731
	SVM	57.7602	80.8611	0.4678
	CEEMDAN-SVM	37.5861	49.0196	0.3043
	CEEMDAN-MLP	48.9322	58.3381	0.3981
	CEEMDAN-LSTM	24.8473	33.3531	0.2010
SSE	LSTM	14.8829	18.6694	0.4586
	SVM	17.4754	21.5559	0.5385
	CEEMDAN-SVM	10.1280	12.7900	0.3128
	CEEMDAN-MLP	10.3604	12.5282	0.3192
	CEEMDAN-LSTM	6.8621	8.7431	0.2116



(b) Linear regression analysis ofS&P500

(a) Forecasting result of S&P500

(d) Linear regression analysis of HSI

(c) Forecasting result of HSI

(f) Linear regression analysis of DAX

(e) Forecasting result of DAX

(h) Linear regression analysis of SSE

(g) Forecasting result of SSE

Figure 4. Four-series prediction results by CEEMDAN-LSTM.

Prediction of nonlinear data. The SVM model used in this article is implemented through the "scikit-Learn" machine learning library. We use conventional radial basis functions (RBF) as the core function. MLP has good generalizability and is widely used in classification and forecasting. In this paper, we optimize various parameters of MLP through experiments and use a 4-layer network, an input layer, two hidden layers, and an output layer. Table 6 shows the forecast actions of S & P500, HSI, DAX and SSE indices using these models.
The error values of MAE, RMSE and MAPE measurements from S & P500 index by CEEMDAN-LSTM model are 3.9177, 4.8291 and 0.1617, respectively, which are much smaller than other models.
CEEMDAN-LSTM also performs better in predicting HSI, DAX and SSE indices than other models and the predicted value is closer to the original value.

Conclusion

The aim is to introduce a new, two-step hybrid approach nof Empirical Mode Decomposition and Factorization Machine based Neural Network (EMD2FNN) and to filter information through the LSTM "gateway" to extract more useful information from historical data. In the first chapter, the importance of accurate forecasting to reduce investment risk in the stock market was defined and explained. In the second chapter, the feasibility of stock market forecasting and Fama capital market efficiency theory was criticized and it was concluded that the real market should be considered as a combination of efficient and inefficient and therefore the real market should be forecastable.
In the following, general explanations about research tools such as data mining, neural network and technical analysis were given and previous researches were reviewed. During the review of these studies, it was pointed out that the time series data mining field is pristine in predicting the series of stock prices in the capital market and the inefficiency of the models developed and introduced to predict the price of a period ahead so far. In the third chapter, stating that both technical analysis and forecasting using classical time series methods have a high ability to predict future prices and stock behavior; A model was developed in which both models were used in the form of an extensive neural network to predict stock prices.
For this purpose, first the data is received directly from the MetaTrader software and the results can be seen on the same software. The Metaweb Query Language 4 is examined and used and the required information is formed. In the fourth chapter, according to the data preprocessing methods and the combination of tools, a purposeful method was performed and then data mining of time series and identification of the most similar time series to the target time series in each database was performed.
Finally, the remaining criteria were entered into the databases as neural network inputs with a hidden layer and the EMD2FNN algorithm was selected for network learning. The results obtained from the constructed network are compared with the predicted results by multivariate regression and time series methods and the criteria such as LSTM, SVM, CEEMDAN-SVM, CEEMDAN-MLP and CEEMDAN-LSTM are compared and examined. The superiority of neural network function in price prediction was well visible in the proposed model and other models.
Although the proposed hybrid models are satisfactory in predicting performance, still *needs some improvements*. For example, the daily clos price is approved only as input data. To improve accuracy and strength, more financial parameters such as trading volume and the highest price and different time scale series may be input. In addition, the time series forecasting method presented in this paper is also able to predict other time series such as weather and traffic.

References

[1] Hafezi, R.; Shahrabi, J.; Hadavandi, E. A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price. Appl. Soft Comput. 2015, 29, 196–210. [CrossRef]
[2] D. Pradeepkumar, V. Ravi, Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network, Appl. Soft Comput. 58 (2017) 35–52.
[3] He, X. , & Chua, T. (2017). Neural factorization machines for sparse predictive analyt- ics. arXiv.org , 355–364.
[4] Kara, Y.; Boyacioglu, M.A.; Baykan, Ö.K. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. Expert Syst. Appl. 2011, 38, 5311–5319. [CrossRef]
[5] Lu, C.-J. Integrating independent component analysis-based denoising scheme with neural network for stock price prediction. Expert Syst. Appl. 2010, 37, 7056–7064. [CrossRef]
[6] Chandar, S.K. Fusion model of wavelet transform and adaptive neuro fuzzy inference system for stock market prediction. J. Ambient Intell. Humaniz. Comput. 2019, 1–9. [CrossRef].