Financial Time Series Forecasting based on Novel CEEMDAN-LSTM-BN Network

Shunwei Dou
Xi'an Jiaotong-liverpool University, Suzhou, China

Abstract: Long-short term memory(LSTM) is a state-of-art and widely used model to forecast financial time series. However, primitive LSTM networks do not perform well due to over-fitting problems of the deep learning model and non-linear and non-stationary characteristics of financial time series data. Thus, this paper proposed a novel hybrid network based on LSTM to solve the two problems. To avoid over-fitting, the modified LSTM-BN network consists of two LSTM layers, two Batch Normalization(BN) layers following each LSTM layer and a dropout layer. Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) is an outstanding data frequency decomposition technique which can decompose original time series into several intrinsic mode functions and a residue. Each of the intrinsic mode function and the residue would be processed by LSTM-BN and the final prediction results are obtained by reconstructing each predictive series. The advantages of the proposed networks are verified by comparing to primitive LSTM, other hybrid models and some famous machine learning model such as Support Vector Machine(SVM), Autoregressive(AR) and Random Forest. Moreover, the robustness of the networks are assessed by numerical experiments on different stock indices datasets including "Standard's & Poor's 500 Index", "Nikkei 225", "Heng Seng Index" and "Deutscher Aktienindex Index".

Keywords: LSTM Networks; SVM; CEEMDAN

1. Introduction

Forecasting stock index price is a significant issue to the investors and professional researchers as it plays an important role in the economy of any region and country.[1][2] Given fact that stock market price data is dynamic, no-stable, non-stationary, nonlinear, noise and chaotic, it is tremendously difficult for researchers to analyse and forecast.[3][7] Although precise prediction of the financial market is nearly impossible, a lot of researchers put forward various ideas and methods to resolve this problem. In fact, with the incessant progress of artificial intelligence, it is possible to forecast financial market more precisely.

In traditional, researchers use machine learning techniques such as the support vector machine (SVM)[5], its advanced version the support vector regression (SVR)[6], auto regression (AR), moving regression (MR), linear regression and linear discriminant analysis (LDA). However, although statistical model has been widely used and very helpful in a variety of research areas, most of machine learning model are based on the stable and linear assumptions so that the efficiency and accuracy of the performance in predicting stock price is poor.[8]

Compared to machine learning techniques, deep learning which can process non-linear and dynamic financial time series has a better performance.[9][10] The well-known deep learning techniques include Artificial Neural network (ANN)[11][12], Multi Layer Perceptron (MLP)[13], Recurrent Neural Network (RNN) [14], Long Short-Term Memory (LSTM)[15], Convolutional Neural Network (CNN)[16] and so on. Some traditional ANN such as Auto Regressive Integrated Moving Average (ARIMA) are used by researchers to predict the stock price. Yet despite there are certain advantages of those neural networks, they are still unable to forecast the fluctuation of stock market accurately because there is no regression and traditional neural networks have only shallow architecture. Forecasting financial market requires time series analysis. To be more precise, the prediction is not only related to the data at the latest time or the current time but also at the earlier time. Compared to traditional ANN, RNN and LSTM, which both capable to extract noisy and non-linear data features, enable to keep memory of earlier information so that improves the forecasting accuracy. Specifically LSTM is widely used by scholars in time series prediction such as pedestrian trajectory prediction[18] and power generation prediction[19] as it solves the long term dependence problem in time series analysis. Researches show that LSTM performs better than ARIMA.[17] Thus, LSTM is select to forecast stock price index in this paper.

Although LSTM are very effective in data analysis[23], sometimes time series data could be so volatile and stochastic that results of performance are still unsatisfying. To tackle this problem, the frequency decomposition such as Empirical mode decomposition (EMD) is used to improve the data analysis process. EMD can decomposes time series data with noise adaptively based on its own scale characteristics while dose not required any basic functions in advance, which has great advantages in dealing with non-stationary and nonlinear data. Moreover, EMD can be a method to reduce the volatility of time series and transform the non-stationary data to stationary data. The performance would be better by using the stationary time series data with low volatility rather than non-stationary data with high volatility.[24] In order to decease the effect of noise and increase the accuracy of prediction, LSTM are combined with EMD to forecast the financial time series.

However,EMD remains an unignorable problem of mode mixing which refers to oscillations of dramatically disparate scales consisted in IMF. To resolve it, several advanced versions have been put forward, for instance, ensemble empirical mode decomposition (EEMD)[20], complementary ensemble empirical mod decomposition (CEEMD)[21] and complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN)[22] which has obvious advantages in avoiding mode mixing and reducing noise in the mode. The EEMD algorithm reduces mode mixing effect by adding normally distributed white noise to original series at the base of EMD algorithm. The CEEMDAN algorithm resolves EEMD's problems of incompleteness and reconstruction error due to adding white noise. Thus, CEEMDAN is adopted to combined with LSTM to improve the accuracy of the forecasting. In the final, this paper uses global stock price indices for practical evaluation and compares the proposed model (CEEMDAN-LSTM) with other model(EMD-LSTM, EMD-SVM, LSTM and SVM).The original financial data comes from four major global stock indices, including Nikkei 225 Stock Index (N225), Standard & Poor 500 Index (S&P500), Hang Seng Index (HSI) and Deutscher Aktien Index(DAX).

The remainder of the paper is organized as follow. The methodology is elaborated in the Section 2. Section 3 discusses the experimental analysis of the prediction. The final Section 4 makes a conclusion.

2. Methodology

2.1 Empirical Mode Decomposition (EMD)

EMD is put forward by Huang et al., which decompose the complex time series accordingly into a series of intrinsic mode functions (IMFs) by characteristic time scales of data. EMD has the power to preprocess the data, transforming non-stationary and non-linear characteristics into stationary and linear. An IMF must follow two conditions.

(1) For each IMF, the difference between the number of minima and the number of maxima at most is one.

(2) The mean of each IMF is zero.

It is worth to noting that the second condition means that an IMF is stationary.

The steps of decomposing are as follow:

(1) Draw the upper envelop $U(t)$ and lower envelop $L(t)$ based on the maximum and minimum of the original financial time series $S(t)$ by the cubic spline interpolation function.

(2) Calculate the mean function $m(t)$ by $m(t) = (U(t) + L(t))/2$.

(3) Calculate $h(t)$ by subtracting $m(t)$ from the original $S(t)$

(4) Repeating steps (1) through (4) with $h(t)$ as the new input series until $h(t)$ satisfies the two conditions of IMF gets the IMF_1, *denoted as* $C_1(t)$, $C_1(t) = h(t)$.

(5) Subtracting IMF_1 from the original

series $S(t)$ gets a new series $R(t) = S(t) - IMF_1$, get IMF_2 by repeating steps (1) through (4).

(6) Repeating the all five steps above with $R_i(t)$ as new series input gets a series of IMFs, denoted $C_i(t)$ until $R_i(t)$ is a constant or satisfies monotonicity.

The are constructed by these IMFs: $S(t) = \sum_{i=1}^{n} C_i(t) + R_n(t)$, where $R_n(t)$ is a residue, representing the trend of the time series $S(t)$.

2.2 Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN)

Although the EMD algorithm as a type of frequency decomposition has great advantages in processing non-stationary series not requiring basic functions in advance, it has obvious mode mixing problems. As an advanced version or EMD, EEMD largely overcomes the mode mixing problems by adding Gaussian withe noise to original data. As a result, however, EEMD faces the problem of incompleteness which means it can not eliminate the white noise after signal reconstruction. To tackle this problem, CEEMDAN is put forward as an advanced version of EEMD. Its reconstruction error is almost zero and it has a faster calculate speed.

Define $E_i(x)$ as the function producing i th mode by EMD and $w_i(t)$ as Gaussian white noise with normal distribution $N(1,0)$.

The implementation steps of CEEMDAN are as follow:

(1) Adding white noise $w_i(t)$ with a standard normal distribution to $S(t)$, $S_i(t) = S(t) + w_i(t)$, $i = 1,2,3...I$, and decomposing each $S_i(t)$ to get IMF_1^i gets the first mode as $\overline{IMF_1} = \frac{1}{I}\sum_{i=1}^{I} IMF_1^i$ and the residue $r_1(t) = S(t) - \overline{IMF_1}$.

(2) Decomposing the $r_1(t) + E_1(w_i(t))$ obtains the second mode $\overline{IMF_2} = \frac{1}{I}\sum_{i=1}^{I} E_1(r_1(t) + E_1(w_i(t)))$ and residue $r_2(t) = S(t) - \overline{IMF_2}$.

(3) Repeat steps above to obtain $\overline{IMF}$ and corresponding residue. The original signal can be reconstructed as follow: $S(t) = \sum_{i=1}^{m} \overline{IMF_i} + r_m(t)$, where m is total number of IMFs and the residue $r_m(t)$ shows the trend or original signal.

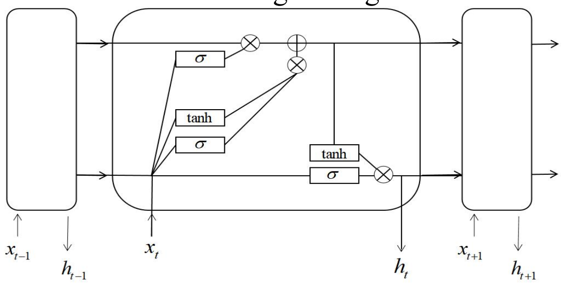


Figure 1. Internal Structure in an LSTM Memory Unit

2.3 Long Short-Term Memory (LSTM)

By the advantage of self feedback mechanism, RNN technique can be utilized in one-step-ahead prediction of financial series using the latest data and previous data. However, although RNN has an advantage in deal with long-term dependence problems, it is nearly practically useless because of the exploding or vanishing gradient problem. To tackle this problems, Hochreiter S, Schmidhuber J. put forward LSTM in 1997.[26] Additionally, LSTM uses the gate mechanism on the base of RNN to largely solve the problem that effective historical information in the previous data can not be preserved for a long time.

LSTM includes of three gates: the forget gate, input gate and output gate. The structure of LSTM unit is showed in Fig. 1. For each LSTM unit at time t, x_t is the input data, x_{t-1} is the input of previous unit, h_t is the output of this LSTM unit and h_{t-1} is the output of previous unit. The steps of LSTM are showed in the following:

(4) The forget gate determines which information will be kept from the previous memory cell C_{t-1} and which will be discarded. The range of f_t is [0,1] and the more the value of f_t gets close to 1, the more information in C_{t-1} will be kept

and vice versa.

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

Where σ is a sigmoid function, W_f is Weight matrix and b_f is bias.

(5) The input gate determines which new information will be retained. Then by the output value of h_{t-1} and input value of x_t, use $\tanh$ function whose range is $[-1,1]$ to produce a candidate value which will be multiplied with i_t and added into memory cell in the next step.

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

$$\overline{c_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Where $\overline{c_t}$ is the candidate memory cell, W_i and W_c are Weight matrix and b_i and b_c are bias.

(6) Calculate the current value of memory cell c_t.

$$c_t = f_t * c_{t-1} + i_t * \overline{c_t}$$

Where c_{t-1} is state value of latest LSTM unit, "*" is represents the dot product, f_t is forget gate and i_t is input gate and $\overline{c_t}$ is candidate memory cell.

(7) Use the value of current memory cell c_t to determine the current LSTM unit's output h_t. The output gate o_t determines which information will be output and using a $\tanh$ activation function gets the candidate value of output.

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

$$h_t = o_t * \tanh(c_t)$$

2.4 Batch Normalization Transform(BN)

Another technologies to solve over-fitting is Batch Normalization proposed by Ioffe and Szegegy in 2015.[28] Batch normalization works by normalizing the input of each layer to have a mean of zero and a variance of one. BN layers involved four steps:(1)Calculate Batch Statistics: Compute the mean and variance of the mini-batch. (2)Normalize the Batch: Subtract the mean and divide by the square root of the variance. (3)Scale and Shift: Apply learned parameters (gamma and beta) to scale and shift the normalized values, which allows the network to represent the identity transformation if necessary. (4)Update Parameters: During back propagation, update the gamma and beta parameters along with the rest of the network parameters.

2.5. Dropout

Over-fitting is an non-ignorable issue in time series prediction, which means the trained model might have a satisfying performance in train set, while poorly fit the test set. To deal with this problem dropout was proposed by Hinton G E.[27] In the forward propagation, some hidden nodes would be block with a preset rate by setting their output values as zero and in the back propagation the parameters in the nodes will not be updated.

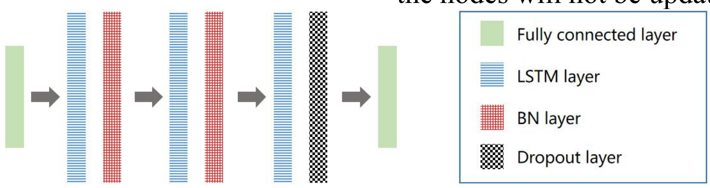


Figure 2. Basic Structure of Proposed Model

2.6 Proposed Model

2.6.1 Basic structure of LSTM-BN

In order to improve the performance in financial time series prediction, it is necessary to combine LSTM with BN layers and dropout layers. Based on this principle, the paper puts forward an LSTM-BN hybrid deep learning network whose framework is showed in Fig. 2. It consists of three LSTM layers, two BN layers following the first two LSTM layers and dropout layer.In this LSTM-BN model, the "ReLU" function is adopted as the activation function of the fully connected layer and we adopt the mean square error (MSE) as the loss function.

$$Loss = MSE = \frac{1}{N}\sum_{n=1}^{N}(d_n - y_n)^2$$

Where N is the total number of the days, d_n is actual value and y_n is predictive value.

2.6.2 Structure of CEEMDAN-LSTM-BN

The CEEMDAN technique is used to

decompose the original series in advance to get smooth sequences. The proposed CEEMDAN-LSTM-BN model is applied to forecast some notable stock price indices and the implementation steps are showed in following:

(1) Use CEEMDAN to decompose the original series $S(t)$ into IMFs sequences $C_i(t)(i=1,2,...,M)$ and a residue $R_M(t)$;

(2) The IMFs obtained and residue are used as the inputs of the LSTM-BN model for training and get the predicted results respectively. The predicted result of the test set is $\overline{C}_i(t)(i=1,2,...,M)$ and $\overline{R}_M(t)$.

(3) Get the final predicted result by the following formula according to each IMF and the residue obtained.

$$S_i(t) = \sum_{i=1}^{M} \overline{C}_i(t) + \overline{R}_M(t)(t=1,2,..,L)$$

Where $S_i(t)$ is the final predictive series of the test set and L is the length of the test series.

3. Analysis of Experimental Results

3.1 Data Preparation

Compared to a single stock, stock market indexes are generally regarded as the best performance indicator of the financial market and in order to demonstrate the robustness of the proposed hybrid model, the daily closing price of the S&P500, N255, HSI and DAX are selected as the original data. Moreover, for better performance of the prediction, Realized volatility (RV) would be calculated from the original data and used as input data of the hybrid model. RV is a measure used in financial time series analysis to quantify the amount of variation or fluctuation in the prices of an asset over a specific time period. Incorporating RV into deep learning models involves using it as an input variable to capture the intrinsic market dynamics better. This approach helps in training models to learn from historical volatility patterns and improve the accuracy of predictions. The method for calculating RV is defined as:

$$r_{t,i} = \ln\left(\frac{p_{t,i}}{p_{t,i-1}}\right)$$

$$RV_t = \sqrt{\sum_{i=1}^{n} r_{t,i}^2}$$

Where $p_{t,i}$ is the close price at interval i on day t, $p_{t,i-1}$ is the price at the previous interval on day t and n is the number of intervals within the day.

Aiming for accelerate the training speed and strengthen the generalization ability of the model, the following standardization method is applied to the RV:

$$P = (RV - \mu_{RV}) / \sigma_{RV}$$

Where μ_{RV} and σ_{RV} are the mean value and standard deviation of RV and is the data set of standardized realized volatility as the input data to the proposed model.

Figure. 3 shows the normalized realized volatility data of financial time series and the statistical analysis of the normalized RV data is shown in Table. 1. The data of all indices are from October 1, 2004 to April 4, 2024. Specially, the RVs of all four indices are relatively higher in 2008 and 2019 which respectively due to financial crisis and COVID-19 epidemic. Moreover, top 90% data would be selected as training set and the rest 10% would serve as test set.

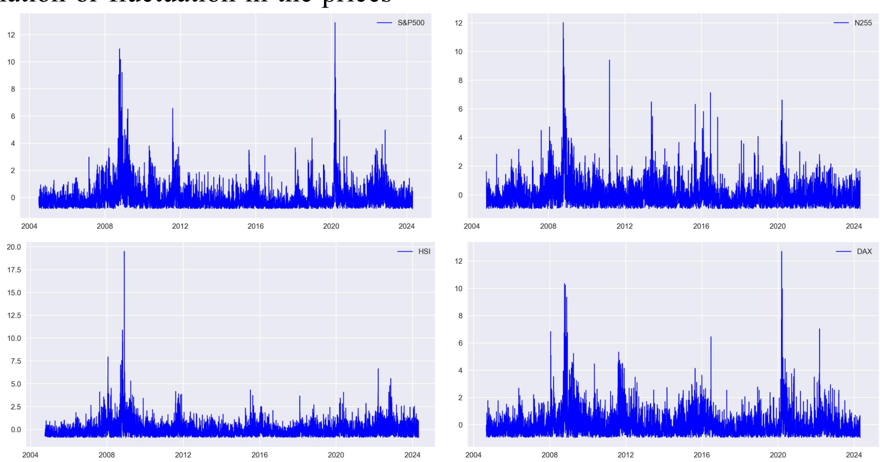


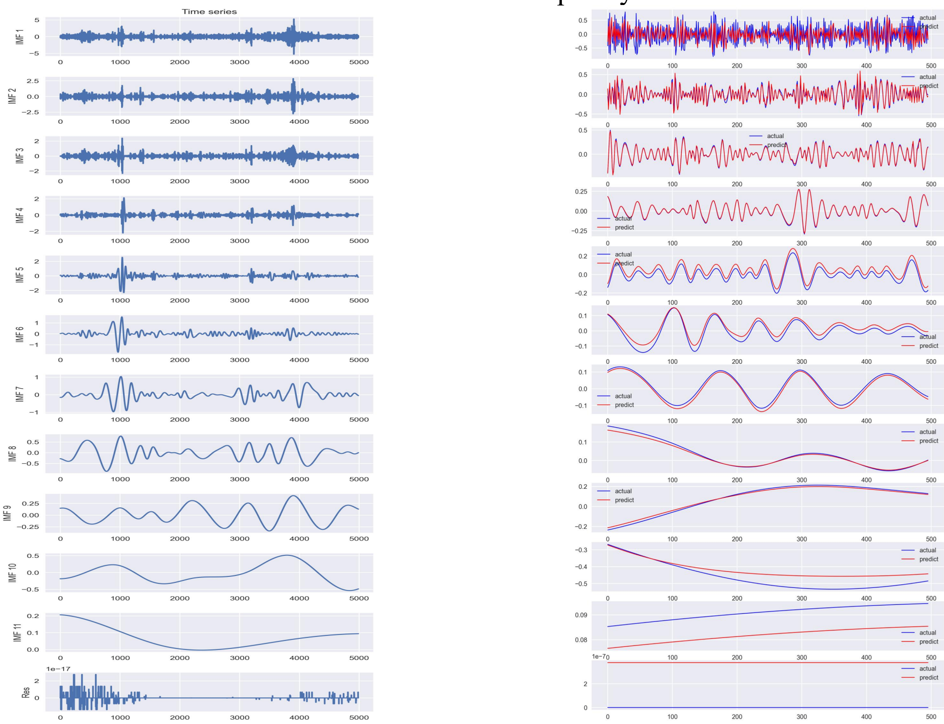
Figure 3. The Normalized Realized Volatility of the Indices

Table. 1 The Statistic Analysis Result Normalized Realized Volatility of the Four Indices

Index	Count	Mean	Min	Max	Standard deviation
S&P500	4987	0	-0.833	12.889	1
N255	4816	0	-0.970	12.016	1
HSI	4850	0	-0.884	19.493	1
DAX	4999	0	-0.934	12.711	1

3.2 CEEMDAN of RV

The normalized RV time series is decomposed by CEEMADAN into several IMFs and one residue. Fig. 4(a) shows the decomposition results of S&P500 index series and each IMFs is arranged from high frequency to low frequency.

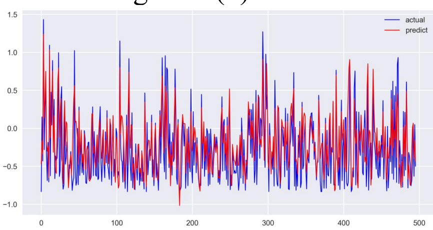


(a) decomposed by CEEMDAN **(b) predicted by CEEMDAN-LSTM-BN**

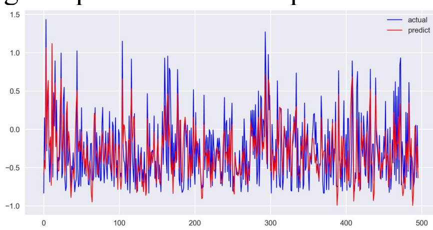
Figure 4. Decomposition and Its Prediction Results of S&P500

3.3 Training process and prediction results

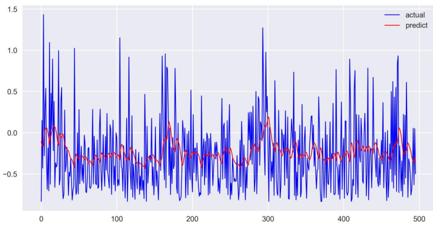
After decomposition, each sub-series consisting of IMFs and residue is divided into a training set and a test set. Figure. 4(b) shows the results of sub-series for S&P500 index data. The prediction performance of high frequency IMF1 and IMF2 is relatively low due to the high amplitude of the components.



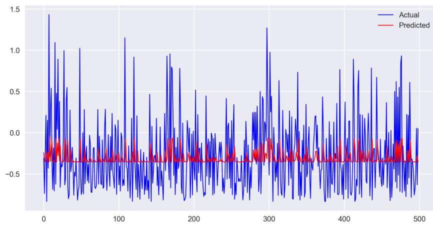
(a) CEEMDAN-LSTM-BN



(b) CEEMDAN-LSTM



(c) LSTM



(d) SVM

Figure. 5 The prediction results of S&P500

The forecast results of S&P500 index by the proposed CEEMDAN-LSTM-BN model are shown in Fig. 5(a). In accordance with this graph, the prediction accuracy of the proposed models is outstanding and to evaluate the performance with more accuracy, two error measures are adopted, including Mean Squared Error (MSE) and Mean Absolute Error(MAE). MSE would also serve as loss function, which has been illustrate above, the remaining criteria is calculated as follows:

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

where y_i is the actual value, $\hat{y}_i$ is the predicted value and n is the number of observations

The smaller the value of MSE and MSA, the smaller the deviation between the predicted value and the original value, suggesting better performance of the model. The MSE value is more stable and solid than so it is used as the main evaluating criteria. Table. 2 shows shows the prediction error of the models. According to Table. 2 the prediction performance of CEEMDAN-LSTM-BN is better than LSTM, LSTM-BN and CEEMDAN-LSTM since CEEMDAN extracts more effective features from original data deemphasizing the effect of noise and BN improve the prediction accuracy of LSTM network.

Table 2. Comparison of Prediction Error of Different Models

(a) Results of S&P500

Model	CEEMDAN-LSTM-BN	CCEMDAN-LSTM	LSTM-BN	LSTM	HAR	AR	SVM	Random Forest
MSE	0.095	0.104	0.204	0.207	0.260	0.453	0.423	1.029
MAE	0.233	0.253	0.368	0.375	0.430	0.454	0.545	0.661

(b) Results of N255

Model	CEEMDAN-LSTM-BN	CCEMDAN-LSTM	LSTM-BN	LSTM	HAR	AR	SVM	Random Forest
MSE	0.216	0.240	0.468	0.499	0.537	0.539	0.714	1.147
MAE	0.343	0.371	0.548	0.567	0.599	0.619	0.602	0.514

(c) Results measures of HSI

Model	CEEMDAN-LSTM-BN	CCEMDAN-LSTM	LSTM-BN	LSTM	HAR	AR	SVM	Random Forest
MSE	0.114	0.123	0.227	0.229	0.314	0.380	0.799	1.506
MAE	0.259	0.253	0.399	0.404	0.480	0.549	0.586	0.728

(d) Results of DAX

Model	CEEMDAN-LSTM-BN	CCEMDAN-LSTM	LSTM-BN	LSTM	HAR	AR	SVM	Random Forest
MSE	0.168	0.170	0.367	0.367	0.419	0.449	0.549	1.222
MAE	0.283	0.300	0.477	0.480	0.548	0.578	0.523	0.741

3.4 Comparison with Other Models

To verify the performance of the proposed hybrid deep learning method, I compared the prediction results of several famous machine learning models, including Support Vector Machine (SVM), Random Forest, AutoRegressive (AR) and Heterogeneous Autoregressive (HAR). All models use the same data set. SVM is widely used to time series prediction.[29] SVM would cast the input data to high-dimensional space, and applies Linear Regression model in high-dimension space to predict the nonlinear data and in this paper I use radial basis function as the kernel function and implement the SVM through the “sklearn” library in Python. The AR model is a famous time series model that uses the dependencies between an observation and a number of past values to make predictions. HAR model is an extension of the AR model that includes components to capture the heterogeneous nature of financial time series data and HAR are widely used in finance to model and predict volatility. Random Forests have good performance in various field including financial market prediction and Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes or mean prediction of the individual trees. Table. 2 shows the prediction measures

of the four indices using these models.
The error measures MSE and MAE values of S&P500 index by the proposed hybrid CEEMDAN-LSTM-BN model are 0.095 and 0.233 respectively, which is obviously the smallest among the models. Additionally, the result of CEEMDAN-LSTM is highly better than LSTM and other models and BN layers also have a important positive effect on hybrid models. The proposed model also performs better than the other models in predicting the N255, HSI and DAX, and the predictive value is very close to the original value.

4. Conclusion

The paper develops a novel CEEMDAN-LSTM-BN hybrid deep learning model to forecast the realized volatility of stock index in financial time series data. In this study, BN layer and dropout layer are adopted to avoid the over-fitting problems in primitive LSTM network. In addition, CEEMDAN signal decomposition technique is combined with LSTM-BN to further enhance the performance. The robustness and effectiveness of the hybrid models are verified by a set of numerical experiments using different stock indices. At the same time, we compare the proposed model with primitive LSTM, LSTM-BN, CCEMDAN-LSTM and other four famous machine learning models. Although the proposed hybrid models have a satisfying performance in forecasting financial indices, there are still some places to improve in the future. For instance, I plan to study data fusion methods to involved not only stock price as input data, but also such as trading volume, different time-scale series and macroeconomic and microeconomic data. In addition, more advanced forecasting models will be studied by introducing the state-of-art deep learning method such as xLSTM and Transformer.

References

[1]Introduction to time series and forecasting[M]. New York, NY: Springer New York, 2002.
[2]Poon S H, Granger C W J. Forecasting volatility in financial markets: A review[J]. Journal of economic literature, 2003, 41(2): 478-539.
[3]Abu-Mostafa Y S, Atiya A F. Introduction to financial forecasting[J]. Applied intelligence, 1996, 6: 205-213.
[4]Wall L D. Some financial regulatory implications of artificial intelligence[J]. Journal of Economics and Business, 2018, 100: 55-63.
[5]Huang W, Nakamori Y, Wang S Y. Forecasting stock market movement direction with support vector machine[J]. Computers & operations research, 2005, 32(10): 2513-2522.
[6]Dash R K, Nguyen T N, Cengiz K, et al. Fine-tuned support vector regression model for stock predictions[J]. Neural Computing and Applications, 2023, 35(32): 23295-23309.
[7]Qin S J, Chiang L H. Advances and opportunities in machine learning for process data analytics[J]. Computers & Chemical Engineering, 2019, 126: 465-473.
[8]Chen Y, Hao Y. A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction[J]. Expert Systems with Applications, 2017, 80: 340-355.
[9]Guresen E, Kayakutlu G, Daim T U. Using artificial neural network models in stock market index prediction[J]. Expert systems with Applications, 2011, 38(8): 10389-10397.
[10]Rezaei H, Faaljou H, Mansourfar G. Stock price prediction using deep learning and frequency decomposition[J]. Expert Systems with Applications, 2021, 169: 114332.
[11]Guresen E, Kayakutlu G, Daim T U. Using artificial neural network models in stock market index prediction[J]. Expert systems with Applications, 2011, 38(8): 10389-10397.
[12]Kim K, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index[J]. Expert systems with Applications, 2000, 19(2): 125-132.
[13]Devadoss A V, Ligori T A A. Forecasting of stock prices using multi layer perceptron[J]. International journal of computing algorithm, 2013, 2(1): 440-449.
[14]Kamijo K, Tanigawa T. Stock price pattern recognition-a recurrent neural network approach[C]//1990 IJCNN international joint conference on neural networks. IEEE, 1990: 215-221.
[15]Fischer T, Krauss C. Deep learning with

long short-term memory networks for financial market predictions[J]. European journal of operational research, 2018, 270(2): 654-669.

[16]Chen S, He H. Stock prediction using convolutional neural network[C]//IOP Conference series: materials science and engineering. IOP Publishing, 2018, 435: 012026.

[17]Siami-Namini S, Tavakoli N, Namin A S. A comparison of ARIMA and LSTM in forecasting time series[C]//2018 17th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2018: 1394-1401.

[18]Xue H, Huynh D Q, Reynolds M. SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction[C]//2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2018: 1186-1194.

[19]Han S, Qiao Y, Yan J, et al. Mid-to-long term wind and photovoltaic power generation prediction based on copula function and long short term memory network[J]. Applied energy, 2019, 239: 181-191.

[20]Wu Z, Huang N E. Ensemble empirical mode decomposition: a noise-assisted data analysis method[J]. Advances in adaptive data analysis, 2009, 1(01): 1-41.

[21]Yeh J R, Shieh J S, Huang N E. Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method[J]. Advances in adaptive data analysis, 2010, 2(02): 135-156.

[22]Colominas M A, Schlotthauer G, Torres M E. Improved complete ensemble EMD: A suitable tool for biomedical signal processing[J]. Biomedical Signal Processing and Control, 2014, 14: 19-29.

[23]Nelson D M Q, Pereira A C M, De Oliveira R A. Stock market's price movement prediction with LSTM neural networks[C]//2017 International joint conference on neural networks (IJCNN). Ieee, 2017: 1419-1426.

[24]Hao W, Sun X, Wang C, et al. A hybrid EMD-LSTM model for non-stationary wave prediction in offshore China[J]. Ocean Engineering, 2022, 246: 110566.

[25]Huang N E, Shen Z, Long S R, et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis[J]. Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 1998, 454(1971): 903-995.

[26]Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[27]Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. arXiv preprint arXiv:1207.0580, 2012.

[28]Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International conference on machine learning. pmlr, 2015: 448-456.

[29]Sapankevych N I, Sankar R. Time series prediction using support vector machines: a survey[J]. IEEE computational intelligence magazine, 2009, 4(2): 24-38.