Physica A 557 (2020) 124907

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction

Yan Hu, Jian Ni *, Liu Wen

School of Finance, Southwestern University of Finance and Economics, Chengdu 611130, China

ARTICLE INFO

Article history:
Received 1 December 2019
Received in revised form 7 June 2020
Available online 6 July 2020

Keywords:
Copper price volatility
Deep learning
GARCH
LSTM-ANN

ABSTRACT

Forecasting the copper price volatility is an important yet challenging task. Given the nonlinear and time-varying characteristics of numerous factors affecting the copper price, we propose a novel hybrid method to forecast copper price volatility. Two important techniques are synthesized in this method. One is the classic GARCH model which encodes useful statistical information about the time-varying copper price volatility in a compact form via the GARCH forecasts. The other is the powerful deep neural network which combines the GARCH forecasts with both domestic and international market factors to search for better nonlinear features; it also combines the long short-term memory (LSTM) network with traditional artificial neural network (ANN) to generate better volatility forecasts. Our method synthesizes the merits of these two techniques and is especially suitable for the task of copper price volatility prediction. The empirical results show that the GARCH forecasts can serve as informative features to significantly increase the predictive power of the neural network model, and the integration of the LSTM and ANN networks is an effective approach to construct useful deep neural network structures to boost the prediction performance. Further, we conducted a series of sensitivity analyses of the neural network architecture to optimize the prediction results. The results suggest that the choice between LSTM and BLSTM networks for the hybrid model should consider the forecast horizon, while the ANN configurations should be fine-tuned depending on the choice of the measure of prediction errors.

1. Introduction

As a key material in various industrial applications, copper is the most actively traded base metal. However, copper prices are highly volatile and depend on many external factors. Such inherent high volatility makes the prediction modeling more challenging. In fact, the application of traditional prediction models (e.g., ARMA and GARCH) has shown their underperformance on the volatility forecasts of copper prices. Despite the challenges, improving the volatility forecasts of copper prices is a worthy endeavor since volatility is an important proxy for market risk. In particular, Kristjanpoller and Minutolo [1] and García and Kristjanpoller [2] have adopted volatility for measuring the risk of commodity market. Ederington and Lee [3] and Fuertes et al. [4] have also applied volatility to measure the risk in the stock market. Other applications of volatility as a measure of market risk can be found in [5–7], and references therein. Therefore, the ability to predict the volatility of copper prices with greater precision is critical for market participants.

* Corresponding author.
E-mail addresses: huyan0910@126.com (Y. Hu), kevinni0326@outlook.com (J. Ni), wenliu7374@163.com (L. Wen).

https://doi.org/10.1016/j.physa.2020.124907
0378-4371/

How to accurately predict volatility is still an unsolved issue. To address this issue, a hybrid volatility prediction model is developed in this study by synthesizing the state-of-art deep learning technique with the classic GARCH *(Generalized Autoregressive Conditional Heteroskedasticity)* model.

In the finance industry, volatility is usually predicted by various GARCH-type models. As a useful generalization of the *autoregressive conditional heteroskedasticity (ARCH)* model, the GARCH model proposed by Bollerslev [8] provides a way to model a change in variance *in a time series*. Later on, different extensions on the basic ARCH and GARCH models have been developed, including EGARCH [9], APGARCH [10], SWARCH [11], FIAPGARCH [12], HYGARCH [13], and more. However, due to the existence of complex nonlinear correlation structure among variables and larger mass of data sets, the prediction results of these GARCH-type models may not be very satisfactory.

The recent development of the deep learning methodology originates from the artificial neural network (ANN) models, which are designed to mimic the knowledge-acquisition and organizational skills of the human brain [14,15]. Many comparative studies between ANNs and traditional prediction models (e.g., ARMA and GARCH) have been conducted with regard to their performance on predictions. High accuracy of ANNs on volatility prediction for various commodities have been demonstrated, see [16–18] and [19].

Furthermore, hybrid ANN and GARCH-type models are usually found to have advantages in comparison with ANNs or time series models. Bildirici and Ersin [20] used a neural network augmented GARCH to predict oil prices. They concluded that neural networks models were promising. Kristjanpoller et al. [5] applied a hybrid ANN-GARCH model to forecast volatility in three Latin American indexes from Brazil, Chile, and Mexico. They demonstrated that neural network models can improve the predictions from GARCH models. Similar studies about hybrid models with ANNs can be seen as Kristjanpoller and Minutolo [1], Cui et al. [21], Lu et al. [22], Lahmiri [23] and Kristjanpoller and Hernández [6].

Extensions of ANNs also improve the accuracy of copper price prediction. A recurrent neural network (RNN) is one of the extensions of ANNs. In the RNN, connections between nodes form a directed graph along a temporal sequence. This allows the neural network to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition [24] or speech recognition [25]. Various forms of RNNs have performed well for prediction as related to time series data [26], such as the long short-term memory (LSTM) and bidirectional LSTM (BLSTM). The LSTM network, which was firstly introduced by Hochreiter and Schmidhuber [27], is widely perceived as a successful RNN model in dealing with the time series data, see [28,29] and [7,30]. BLSTM is proposed to access both the past and future information by combining a forward hidden layer and a backward hidden layer [31]. Sardelich and Manandhar [32] used a model combining the BLSTM and stacked LSTM neural networks to predict daily stock volatility. The proposed model outperformed the well-known GARCH (1,1) model in many sectors (e.g., financial, health care, etc.). Eapen et al. [33] found that CNN layers combining with BLSTM showed a better prediction performance than the traditional SVM regressor.

Therefore, the recent advancement of the AI techniques, characterized by the so-called deep learning methodology, opens up possibilities for us to develop new prediction models that can better forecast the volatility of copper prices. To enhance the prediction power, the best forecasts of the GARCH model are used as inputs for the hybrid models that combine GARCH and neural networks. Six neural network models (i.e., ANN, LSTM-ANN, BLSTM-ANN, GARCH-ANN, GARCH-LSTM-ANN, and GARCH-BLSTM-ANN) will be investigated in this study. A comparative study on the proposed models will be conducted to find the best approach for the prediction of copper price volatility.

The remainder of this paper is structured as follows. In Section 2, we introduce six prediction models. In Section 3, we conduct an empirical analysis to assess the prediction results. In Section 4, we summarize the main results and conclude this study.

2. Methodology

2.1. Data preprocessing method

Return estimation

The closing price of copper has a very crucial role in the metal market. Moreover, investors are more concerned with return rather than the price. Therefore, the volatility of returns is forecasted in this paper. The returns of copper on day t are computed by using the following equation:

$$R_t = (\log p_t - \log p_{t-1}) \times 100\% \tag{1}$$

where R_t is the return of the time series on day t, P_t and P_{t-1} are the closing prices of the financial time series on day t and $t-1$, respectively.

Volatility estimation

Volatility plays a very important role in the metal market. As a classic metric of volatility, variance reflects how much copper has varied during a certain period. In this paper, we compare the predicted and realized volatilities (i.e., variances) to assess the performance of various models. Realized volatility (RV_t) on day t during T trading days is calculated by the following equation:

$$RV_t = \frac{1}{T}\sum_{j=t}^{j=t+T-1}(R_j - \overline{R})^2 \tag{2}$$

where T is the number of trading days after day t, R_j is the return of copper on day j, and $\overline{R}$ is the average return of copper during T trading days. RV_t is the actual volatility. In this study, we will consider three cases, i.e., $T = 10, 15, 20$. For $T = 10$, RV_t is the realized volatility over the next 10 trading days (two weeks) from day t. Similarly, RV_t values for $T = 15$ and $T = 20$ then correspond to the realized volatilities over the next 15 trading days (three weeks) and 20 trading days (four weeks) from day t, respectively.

Data standardization

Standardization is very important before putting data into the ANN and LSTM networks. It can enhance the training efficiency of the prediction models. In this paper, we choose the MinâMax method to normalize the raw input data. Following this method, the input variables are calculated as follows:

$$\tilde{x}_{i,t} = \frac{x_{i,t} - x_{i,\min}}{x_{i,\max} - x_{i,\min}} \tag{3}$$

where $\tilde{x}_{i,t} \in [0, 1]$ is the standardized value of the ith feature on day t, $x_{i,t}$ is the actual value of the ith feature on day t, $x_{i,\min}$ and $x_{i,\max}$ is the minimum and maximum value of the ith feature, respectively.

Measures of prediction errors

To evaluate and compare the performances of the prediction models, four common measures of prediction errors are used, including the mean squared error (MSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE), and the root mean squared error (RMSE). They are defined as follows:

$$MSE = \frac{1}{N}\sum_{\mathrm{i}=1}^{N}(PV_t - RV_t)^2 \tag{4}$$

$$MAE = \frac{1}{N}\sum_{\mathrm{i}=1}^{N}|PV_t - RV_t| \tag{5}$$

$$MAPE = \frac{1}{N}\sum_{\mathrm{i}=1}^{N}|1 - PV_t/RV_t| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{\mathrm{i}=1}^{N}(PV_t - RV_t)^2} \tag{7}$$

where PV_t and RV_t are the predicted and realized volatilities of copper return series, respectively; N is the number of predictions. A lower value of the measure indicates a better prediction.

We note that the four measures of prediction errors (i.e., MSE, MAE, RMSE, and MAPE) introduced above are widely-applied in the literature. For example, Fuertes et al. [4] and Kim and Won [7] have applied MSE, MAE, and MAPE in analyzing stock volatility forecasts. Kristjanpoller et al. [5] has utilized the four measures (MSE, MAE, RMSE, and MAPE) to examine the volatility prediction results of hybrid neural network models for three Latin-American stock exchange indexes. Other applications of these four measures in analyzing prediction models of commodity market include Bentes [34], Zhang et al. [35], and Kristjanpoller and Hernández [6]. Besides, these four measures have also been adopted in studying prediction models in the foreign exchange market [36–38]. To be in line with the literature, all the four measures of prediction errors (i.e., MSE, MAE, RMSE, and MAPE) are utilized in this study to compare different volatility prediction models of copper price.

2.2. Prediction models

GARCH model

GARCH model describes the variance of the current error term as a function of the error terms of the previous periods. The specification for the GARCH*(p, q)* is defined as:

$$R_t = \mu + a_t \tag{8}$$

$$a_t = \varepsilon_t\sigma_t \tag{9}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p}\alpha_i a_{t-i}^2 + \sum_{j=1}^{q}\beta_j\sigma_{t-j}^2 \tag{10}$$

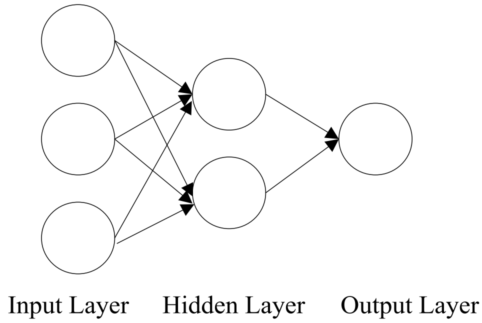

Fig. 1. An illustrative graph of the ANN architecture.

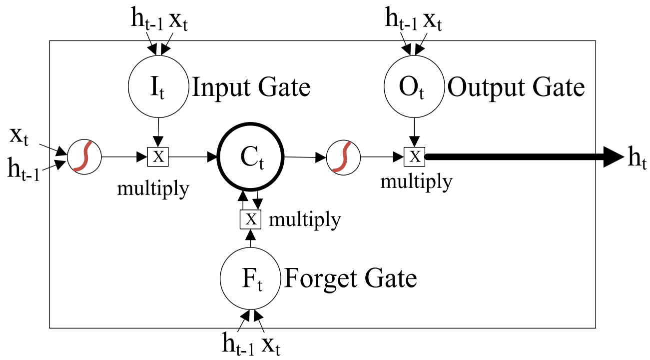

Fig. 2. An illustrative graph of the LSTM unit.

where $\alpha_0 > 0, \alpha_i \geq 0, i = 1, \ldots, p, \beta_j \geq 0, j = 1, \ldots, q$ which guarantee that the conditional variance of GARCH*(p, q)* is always positive. $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$ which ensures that the variance is finite.

ANN model

ANN is a network of artificial neurons, which can receive inputs, change their internal states according to the inputs, and then compute outputs based on the inputs and internal states. These artificial neurons have weights that can be modified by a process called learning. Fig. 1 shows a neural network model that sequentially calculates the value of the output layer from the input layer by using the output of the previous layer as the input for the current layer.

The ANN follows the following formulation:

$$y = f(f(x_i w_{ij} + b_j) w_{jk} + b_k) \tag{11}$$

Eq. (11) shows that the input variable x_i is multiplied by weight w_{ij} and summed with bias b_j, $f(.)$ is the activation function, the result of this layer is the inputs of the next layer, and the final result y is the forecast value.

LSTM model

LSTM is a classic type of RNNs which can deal with the exploding and vanishing gradient problems. It is normally augmented by recurrent gates called forget gates. Different from previous neural networks such as ANN, LSTM can learn very deep learning tasks that require a long-time memory of events. LSTM can also handle inputs or signals that have both low and high-frequency components. For more technical details, please see [27,39]. The structure of an LSTM unit is demonstrated in Fig. 2.

The equations for an LSTM unit are demonstrated in (12)–(16):

$$f_t = \sigma(W_{fx} x_t + W_{fh} h_{t-1} + b_f) \tag{12}$$

$$i_t = \sigma(W_{ix} x_t + W_{ih} h_{t-1} + b_i) \tag{13}$$

$$o_t = \sigma(W_{ox} x_t + W_{oh} h_{t-1} + b_o) \tag{14}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx} x_t + W_{ch} h_{t-1} + b_c) \tag{15}$$

$$h_t = o_t \odot \tanh(c_t) \tag{16}$$

As Fig. 2 illustrates, the LSTM contains a memory cell (c_t) and three gates: an input gate (i_t), a forget gate (f_t), and an output gate (o_t). In Eqs. (12)–(16), the initial values are $c_0 = 0$ and $h_0 = 0$, the operator $\odot$ denotes the element-wise

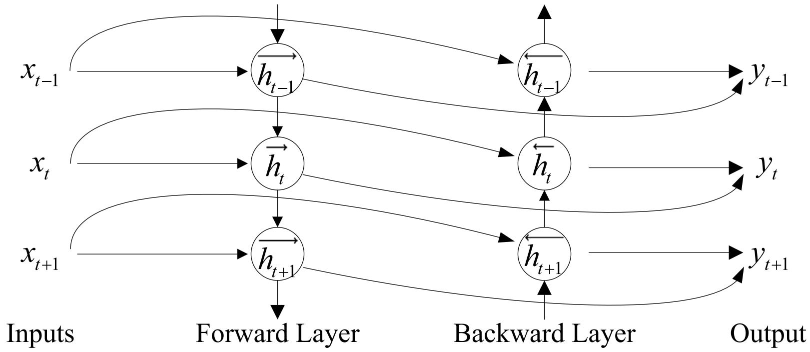

Fig. 3. An illustrative graph of the BLSTM architecture.

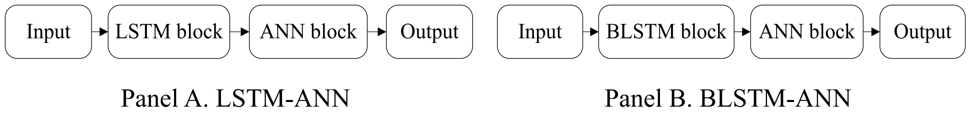

Fig. 4. An illustrative graph of the hybrid neural network.

product (i.e., Hadamard product). At time t, x_t represents the input vector, h_t is the hidden state vector which is also known as the output vector of the LSTM unit. W and b are weight matrices and bias parameters which need to be learned during training. $\sigma(.)$ is the sigmoid function and tanh(.) is hyperbolic tangent function.

Bidirectional LSTM model

The design of BLSTM is to access both the past and future information by combining a forward hidden layer and a backward hidden layer [31]. As shown in Fig. 3, BLSTM can access long-range information in two opposite directions: one processing the sequential data from top to bottom, the other one from bottom to top. The equations for a BLSTM unit are given in (17)–(19):

$$\overrightarrow{h_t} = f(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h_{t-1}} + \overrightarrow{b_h}) \tag{17}$$

$$\overleftarrow{h_t} = f(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h_{t+1}} + \overleftarrow{b_h}) \tag{18}$$

$$y_t = g(W_{\overrightarrow{h}y} \overrightarrow{h_t} + W_{\overleftarrow{h}y} \overleftarrow{h_t} + b_y) \tag{19}$$

where x_t represents the input vector, h_t is the hidden state vector. W and b are weight matrices and bias parameters which need to be learned during training. Fig. 3 shows the structure of the BLSTM unit.

LSTM-ANN and BLSTM-ANN networks

LSTM-ANN and BLSTM-ANN are hybrid network models that combine LSTM and BLSTM with ANN model. A simple illustration of the LSTM-ANN (BLSTM-ANN) structure is shown in Fig. 4.

Six hybrid prediction models

We now proceed to formally introduce the six prediction models of copper price volatility that will be tested and analyzed in detail. By integrating the aforementioned LSTM-ANN and BLSTM-ANN networks with the GARCH model, we can construct two hybrid prediction models, namely, GARCH-LSTM-ANN and GARCH-BLSTM-ANN; they use the GARCH forecasts as extra inputs in an attempt to provide useful complementary information to the neural network model. The effectiveness of incorporating GARCH forecasts is then assessed by benchmarking these two hybrid models with models of the same network structure but without using the GARCH forecasts (i.e., LSTM-ANN and BLSTM-ANN). On the other hand, to test the effectiveness of introducing the memory units (LSTM or BLSTM) into the hybrid model, the memory-free ANN structure is also introduced for comparison purpose, resulting in two additional benchmark models (ANN and GARCH-ANN). Thus, we have six different types of hybrid prediction models, namely, ANN, LSTM-ANN, BLSTM-ANN, GARCH-ANN, GARCH-LSTM-ANN, and GARCH-BLSTM-ANN. We will use real-world data to compare and test their prediction performances from different angles to develop insights on the development of effective hybrid prediction models.

Table 1
Variables and descriptions.

Group	Variable	Description
Metal markets	CP	Copper spot price of Yangtze River nonferrous metals, China
	AP	Aluminum spot price of Yangtze River nonferrous metals, China
	ZP	Zinc spot price of Yangtze River nonferrous metals, China
	GP	Gold spot price of Shanghai nonferrous metals, China
	LCP	Copper spot price of London Metals Exchange
	LAP	Aluminum spot price of London Metals Exchange
	LZP	Zinc spot price of London Metals Exchange
Exchange rates	USD	CNY — US Dollar exchange rate
	EUR	CNY — Euro exchange rate
	YEN	CNY — 100JPY exchange rate
Stock markets	DJIA	DJIA index of American stock market
	FTSE	FTSE 100 index of British stock market
	CSI	CSI300 index of Chinese stock market
Futures markets	LMCP	Copper futures price of London Metals Exchange
	LMAP	Aluminum futures price of London Metals Exchange
	LMZP	Zinc futures price of London Metals Exchange
	OIL	WTI crude oil futures price
	HCP	Copper futures price of Shanghai Futures Exchange
	HAP	Aluminum futures price of Shanghai Futures Exchange
	HZP	Zinc futures price of Shanghai Futures Exchange
Interest rate	SHI	Shanghai Interbank Offered Rate

Table 2
Descriptive statistics for the returns of the variables.

Variable	Mean (%)	Std. Dev. (%)	Min (%)	Max (%)	Jarque–Bera	ADF	ARCH (12)
CP	−0.0075	1.33	−9.06	8.05	6823.15	−22.23	634.97
AP	−0.0103	0.85	−6.01	4.29	4389.83	−44.49	505.86
ZP	0.0049	1.29	−7.04	7.27	1809.67	−47.59	303.92
GP	0.0137	0.96	−7.35	6.55	7295.08	−43.90	219.09
LCP	−0.0038	1.72	−19.47	11.37	12 323.63	−53.89	397.11
LAP	−0.0086	1.49	−11.00	8.50	2294.83	−53.88	153.24
LZP	0.0024	1.89	−15.46	9.80	1695.34	−52.65	328.49
USD	−0.0023	0.15	−0.93	1.84	41 083.53	−46.70	680.29
EUR	−0.0115	0.62	−6.93	3.34	7425.28	−51.18	124.67
YEN	−0.0020	0.64	−5.15	3.63	3387.94	−38.22	182.25
DJIA	0.0213	1.20	−11.27	10.51	17 587.75	−40.42	792.44
FTSE	0.0018	1.22	−10.33	9.38	10 556.78	−24.65	633.30
CSI	−0.0217	1.75	−9.15	8.93	1807.47	−50.32	334.43
LMCP	−0.0045	1.73	−19.76	11.88	12 873.50	−23.71	383.21
LMAP	−0.0103	1.39	−10.56	7.54	1780.60	−53.79	212.34
LMZP	0.0001	1.88	−15.27	9.78	1680.29	−54.27	331.47
OIL	−0.0294	2.51	−19.66	16.41	4000.99	−55.11	483.10
HCP	−0.0071	1.39	−7.50	6.71	1531.31	−34.52	453.46
HAP	−0.0100	0.99	−6.94	5.40	4616.02	−52.67	450.73
HZP	0.0041	1.53	−9.40	5.96	1149.04	−54.66	251.10
SHI	−0.0210	7.70	−62.48	77.52	63 021.49	−36.95	228.08

Note: The critical value at 5% for Jarque–Bera test is 5.99. ADF is the stationarity test, and the critical value at 5% is −2.86. The ARCH(12) statistic corresponds to the ARCH-LM test with 12 lags, where the probability distribution is $\chi^2(12)$ and the critical value at the 5% level is 21.03.

3. Empirical analysis

3.1. Data description

We collect data from January 1, 2008 to December 31, 2018, totaling eleven years. The data source is the WIND database. A group of explanatory variables are utilized, which could potentially improve the volatility forecasts. The detailed description of the explanatory variables is given in Table 1. The variables are related to the main metal prices, the main stock market indices, currency, the main metal futures, and interest rate.

We use the daily closing prices of these variables to compute their daily returns. Missing values are filled with the last valid values of previous trading days. Then, each data set is sequentially divided into two sets, i.e., the training set which contains 70% of the data, and the testing set which contains the remaining 30% data. Table 2 presents the common statistics for returns of the variables.

From Table 2, it is shown that the mean copper returns (CP) is close to 0 (−0.0075%), and its standard deviation is close to 1%. The ADF test is significant at 5%, which indicates that the time series are stationary. However, the Jarque–Bera

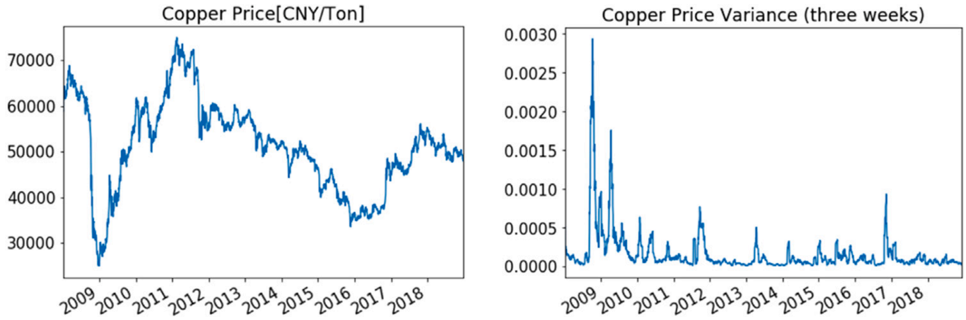

Fig. 5. Time series plots for daily closing price and the variance of three weeks.

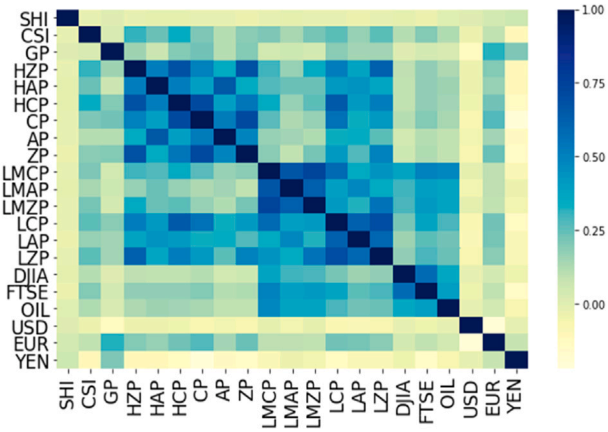

Fig. 6. Heat map of return correlation coefficient matrix.

(normality) test shows that the copper returns are not normally distributed. Besides, ARCH (12) is the heteroskedasticity test used to identify the presence of ARCH effects for copper returns, and the null hypothesis is rejected, indicating the existence of ARCH effects. Moreover, Table 2 shows that other explanatory variables also exhibit some complex statistical features, such as non-normality and ARCH effects. These complex statistical features imply that it is difficult to deal with these time series with conventional statistical methods. It is therefore meaningful to investigate the effective combination of both conventional statistical methods (e.g., GARCH) and the state-of-art deep learning techniques (e.g., LSTM) to improve the copper price volatility prediction.

Next, we depict the copper price, as well as its variance of three weeks from 2008 to 2018 in Fig. 5.

From Fig. 5, it can be observed that, in the second half of 2008, the price of copper suddenly dived, coinciding with the subprime crisis. This unexpected crash greatly increased the subsequent market volatility in the second half of 2008 and the beginning of 2009. Furthermore, high volatility is also observed in 2011, 2013, and 2016.

The lower the correlation between the explanatory variables indicating more information can be taken by the models to make a better fit. For this reason, the study on explanatory variables is conducted with the correlation analysis and principal component analysis. Both analyses are investigated in terms of returns. The heat map of return correlation coefficient matrix is shown in Fig. 6.

From Fig. 6, we can observe that the correlation coefficients disperse. It means that we need some methods to extract useful information. Also, from the principal component analysis (please see Table A.1 in the Appendix) we can find that with 19 eigenvectors, 98.85% of explained variance ratio is achieved. Furthermore, 99.45% is achieved with 20 eigenvectors. These results also show that the relationship is not high.

3.2. Empirical results

In this section, six different types of hybrid prediction models are empirically tested and compared, including ANN, LSTM-ANN, BLSTM-ANN, GARCH-ANN, GARCH-LSTM-ANN, and GARCH-BLSTM-ANN. The empirical study aims to unravel four main questions: (1) Is the inclusion of GARCH forecasts as network inputs an effective way to enhance the prediction performance? (2) Can the integration of RNNs with GARCH forecasts significantly reduce the prediction errors? (3) Which neural network architecture, particularly the choice between LSTM and BLSTM, works better for the task of copper price volatility prediction? (4) How to fine-tune the ANN configuration (e.g., the number of layers and neurons) in the hybrid network to optimize the prediction results?

To this end, different choices of the number of hidden layers and neurons in each layer are tested and compared. Each ANN configuration is denoted by ANN *(l, n)*, where l corresponds to the number of hidden layers, and n corresponds to the number of neurons in each layer. We set the hidden layers and neurons of ANN according to the results of Kristjanpoller and Hernández [6]; thus, $l \in \{4, 5, 6\}$ and $n \in \{10, 20\}$. As for the GARCH model, we have tested the model performances of various parameters values (p and q, see Eq. (10)) using the copper price data, and found that in general the best choice is $p = 1$ and $q = 1$. Thus, GARCH*(1,1)* is adopted in the numerical experiments.

In the first set of experiments, we use one year trading data (i.e., one-year input window, including 252 trading days) to forecast the volatility in the next two weeks (two-week ahead forecasts, or $T = 10$ in Eq. (2)). Four measures of prediction errors (i.e., MSE, MAPE, MAE, and RMSE) are utilized to evaluate the prediction performances. To facilitate the prediction error analysis, all returns are scaled up by a factor of 100. The results are presented in Table 3.

From Table 3, we have the following key observations. First, although the network models can outperform the GARCH model, incorporating GARCH forecasts as inputs can enhance the prediction power of the network models. For example, the MSE of GARCH-LSTM-ANN(6,20) is 3.56E−03, which is smaller than LSTM-ANN(6,20) with a MSE of 4.35E−03. The MAPE of GARCH-LSTM-ANN (6,10) is 4.2640, being smaller than that of LSTM-ANN(6,10), of which the MAPE is 4.4011. Similar results can be found in the cases of MAE and RMSE in spite of a few exceptions. Second, both LSTM and BLSTM can improve the prediction performance of memory-free ANN. In other words, both GARCH-LSTM-ANN and GARCH-BLSTM-ANN models outperform the GARCH-ANN model, and both LSTM-ANN and BLSTM-ANN models demonstrate better performances than the ANN model, with only few exceptions.

Next, GARCH-LSTM-ANN architecture is better than the other five network architectures. The MSE, MAPE, MAE, and RMSE of GARCH-LSTM-ANN models are the lowest in general. There is only one exception. The MAE of LSTM-ANN(6,10) and GARCH-LSTM-ANN are almost the same. Then, let us focus on GARCH-LSTM-ANN models. In terms of MSE, the best prediction model is GARCH-LSTM-ANN(6,10) with a lowest MSE of 2.91E−03. However, the smallest MAPE (3.5922) is achieved by GARCH-LSTM-ANN(5,20). Besides, the GARCH-LSTM-ANN architecture with the 6-layer and 20-neuron in ANN configuration is the best prediction model in terms of MAE. These results suggest that the ANN configuration should be optimized depending on the choice of the measure of prediction errors.

We next investigate the case of three-week ahead forecasts ($T = 15$ in Eq. (2)). The results are presented in Table 4.

From Table 4, we have the following key observations. First, we can find that the GARCH model underperforms the network models and incorporating GARCH forecasts as inputs can still enhance the prediction power of the network models in general. This can be illustrated by the following comparisons. Let us compare LSTM-ANN and GARCH-LSTM-ANN models. The MSE of GARCH-LSTM-ANN(5,20) is 3.36E−03, which is smaller than that of LSTM-ANN models. As for the comparison between BLSTM-ANN and GARCH-BLSTM-ANN, the MSE of GARCH-BLSTM-ANN(4,20) is 3.60E−03, which is significantly smaller than that of BLSTM-ANN models. Similar results can be found for other error measures (MAPE, MAE and RMSE) and the comparison between ANN and GARCH-ANN models. Besides, GARCH-BLSTM-ANN architecture appears to be better than the other five network architectures, since the MSE, MAPE, MAE, and RMSE of GARCH-BLSTM-ANN models are always the lowest.

We next examine the best network structure. It is clear that both LSTM-ANN and BLSTM-ANN models demonstrate better performances than ANN models. Thus, the incorporation of LSTM or BLSTM is effective. As for the ANN configurations (hidden layers and neurons), it still appears that the best configuration should be dependent on the choice of error measures.

After the investigation of two-week and three-week ahead forecasts, we then turn to four-week ahead forecasts ($T = 20$ in Eq. (2)) to analyze the robustness of the results. The results are summarized in Table 5.

From Table 5, we have the following key observations. First, we can find that incorporating GARCH forecasts as inputs can enhance the prediction power of the network models in general. For example, the MSE of GARCH-BLSTM-ANN(5,20) is 3.65E−03, being smaller than that of BLSTM-ANN(5,20), of which the MSE is 4.08E−03. Besides, it is clear that both LSTM and BLSTM can improve the prediction performance of ANN, which is consistent with former observations for the cases of the two-week and three-week forecasts.

Next, GARCH-LSTM-ANN architecture still presents the best predictive power than the other five network architectures in general. For example, the MAPE of GARCH-LSTM-ANN(6,10) is 0.9237; while this is the worst among all GARCH-LSTM-ANN model configurations, it is still better than other prediction models. Then, let us examine the influence of network configuration to optimize the prediction results. We find that when prediction errors are measured in terms of MSE, MAPE, MAE, and RMSE, the best prediction models are GARCH-LSTM-ANN(6,10), GARCH-LSTM-ANN(4,20), GARCH-LSTM-ANN(6,10) and GARCH-LSTM-ANN(6,10), respectively. Thus, the best configurations (hidden layers and neurons) of the prediction model are a bit different for different error measures.

To sum up, we can find the following conclusions from Tables 3–5. First, incorporating GARCH forecasts as inputs can enhance the prediction power of the network models in general. Second, the integration of memory networks (i.e., LSTM and BLSTM) with the classical ANN can improve the prediction power of the entire neural network. Third, the best model architecture for volatility prediction is GARCH-LSTM-ANN for two-week and four-week ahead forecasts. However, for

Table 3
Prediction performances of two-week ahead forecasts for different hybrid neural network configurations.

Panel A							
Measure of prediction errors		MSE			MAPE		
Network architecture	Ranking	*(l,n)*	MSE	Var%	*(l,n)*	MAPE	Var%
GARCH			1.03E−02	−253.95%		14.7829	−311.53%
S1	1	(6,10)	4.26E−03	−46.39%	(6,10)	5.8649	−63.27%
	2	(6,20)	4.60E−03	−58.08%	(6,20)	7.4989	−108.76%
	3	(4,20)	4.93E−03	−69.42%	(4,20)	8.3441	−132.28%
S2	1	(6,20)	4.35E−03	−49.48%	(6,10)	4.4011	−22.52%
	2	(6,10)	4.41E−03	−51.55%	(6,20)	6.5909	−83.48%
	3	(4,20)	4.60E−03	−58.08%	(4,10)	7.474	−108.06%
S3	1	(6,10)	4.38E−03	−50.52%	(6,10)	6.6012	−83.76%
	2	(4,20)	4.56E−03	−56.70%	(6,20)	7.3599	−104.89%
	3	(6,20)	4.57E−03	−57.04%	(4,20)	7.3894	−105.71%
GARCH-S1	1	(4,10)	4.53E−03	−55.67%	(6,10)	5.6488	−57.25%
	2	(5,10)	4.66E−03	−60.14%	(5,10)	5.9565	−65.82%
	3	(5,20)	4.68E−03	−60.82%	(5,20)	7.722	−114.97%
GARCH-S2	**1**	**(6,10)**	**2.91E−03**	**Base case**	**(5,20)**	**3.5922**	**Base case**
	2	(5,10)	3.55E−03	−21.99%	(6,20)	3.8292	−6.60%
	3	(6,20)	3.56E−03	−22.34%	(4,20)	4.2919	−19.48%
GARCH-S3	1	(6,20)	3.53E−03	−21.31%	(6,10)	4.264	−18.70%
	2	(5,10)	3.56E−03	−22.34%	(6,20)	4.3101	−19.98%
	3	(5,20)	3.63E−03	−24.74%	(5,10)	4.6163	−28.51%
Panel B							
Measure of prediction errors		MAE			RMSE		
Network architecture	Ranking	*(l,n)*	MAE	Var%	*(l,n)*	RMSE	Var%
GARCH			9.26E−02	−192.11%		1.01E−01	−87.38%
S1	1	(6,10)	3.72E−02	−17.35%	(6,10)	6.53E−02	−21.15%
	2	(6,20)	4.53E−02	−42.90%	(6,20)	6.78E−02	−25.79%
	3	(4,20)	5.04E−02	−58.99%	(4,20)	7.02E−02	−30.24%
S2	1	(6,10)	3.15E−02	0.63%	(6,20)	6.60E−02	−22.45%
	2	(6,20)	4.06E−02	−28.08%	(6,10)	6.64E−02	−23.19%
	3	(4,20)	4.52E−02	−42.59%	(4,20)	6.78E−02	−25.79%
S3	1	(6,10)	4.05E−02	−27.76%	(6,10)	6.62E−02	−22.82%
	2	(4,20)	4.44E−02	−40.06%	(4,20)	6.75E−02	−25.23%
	3	(6,20)	4.45E−02	−40.38%	(6,20)	6.76E−02	−25.42%
GARCH-S1	1	(4,10)	3.86E−02	−21.77%	(4,10)	6.73E−02	−24.86%
	2	(6,10)	3.92E−02	−23.66%	(5,10)	6.82E−02	−26.53%
	3	(5,10)	4.39E−02	−38.49%	(5,20)	6.84E−02	−26.90%
GARCH-S2	**1**	**(6,20)**	**3.17E−02**	**Base case**	**(6,10)**	**5.39E−02**	**Base case**
	2	(5,20)	3.35E−02	−5.68%	(5,10)	5.96E−02	−10.58%
	3	(4,20)	3.36E−02	−5.99%	(6,20)	5.96E−02	−10.58%
GARCH-S3	1	(6,20)	3.20E−02	−0.95%	(6,20)	5.94E−02	−10.20%
	2	(6,10)	3.38E−02	−6.62%	(5,10)	5.97E−02	−10.76%
	3	(5,10)	3.43E−02	−8.20%	(5,20)	6.02E−02	−11.69%

Var% is the percentage variation of the error measures. S1, S2 and S3 represent the ANN, LSTM-ANN and BLSTM-ANN respectively.

three-week ahead forecasts, the best architecture is GARCH-BLSTM-ANN. Fourth, the ANN configuration in the hybrid model should be optimized depending on the choice of the measure of prediction errors.

Further, the predicted and realized copper price volatilities in both the training and testing sets are shown in the Fig. 7. The cases of two-week, three-week, and four-week ahead forecasts are all considered. Fig. 7 shows that the deviation between the predicted and realized volatilities will generally become larger as the forecast horizon grows from two weeks to four weeks. This is consistent with the intuition that it should be easier to make predictions about the nearer future. Moreover, from Fig. 7, it seems that the forecasting accuracy is not dependent on volatility level. For instance, for the two-week and three-week ahead forecasts in the testing set, the realized volatilities around November 2016 are extremely high, whereas the realized volatilities around August 2017 are relatively low. However, there are significantly large prediction errors around both November 2016 and August 2017. Thus, the size of the prediction error is not necessarily dependent on the level of the realized volatility.

Table 4
Prediction performances of three-week ahead forecasts for different hybrid neural network configurations.

Panel A							
Measure of prediction errors		MSE			MAPE		
Network architecture	Ranking	*(l,n)*	MSE	Var%	*(l,n)*	MAPE	Var%
GARCH			1.10E−02	−237.42%		5.1202	−413.82%
S1	1	(5,20)	4.26E−03	−30.67%	(5,20)	2.3503	−135.86%
	2	(4,20)	4.67E−03	−43.25%	(4,20)	2.6689	−167.83%
	3	(4,10)	4.68E−03	−43.56%	(4,10)	2.6963	−170.58%
S2	1	(6,10)	3.81E−03	−16.87%	(5,20)	1.7711	−77.73%
	2	(5,20)	3.85E−03	−18.10%	(6,10)	1.9936	−100.06%
	3	(6,20)	4.03E−03	−23.62%	(6,20)	2.0984	−110.58%
S3	1	(6,10)	3.87E−03	−18.71%	(6,10)	1.8797	−88.63%
	2	(5,20)	3.90E−03	−19.63%	(5,20)	1.8994	−90.61%
	3	(4,20)	4.05E−03	−24.23%	(4,10)	2.0935	−110.09%
GARCH-S1	1	(6,20)	4.51E−03	−38.34%	(5,20)	1.1374	−14.14%
	2	(5,20)	4.65E−03	−42.64%	(6,20)	2.5523	−156.13%
	3	(5,10)	4.71E−03	−44.48%	(5,10)	2.7035	−171.30%
GARCH-S2	1	(5,20)	3.36E−03	−3.07%	(5,20)	1.1095	−11.34%
	2	(5,10)	3.88E−03	−19.02%	(4,10)	1.8645	−87.10%
	3	(4,10)	3.89E−03	−19.33%	(5,10)	1.867	−87.36%
GARCH-S3	**1**	**(4,10)**	**3.26E−03**	**Base case**	**(6,10)**	**0.9965**	**Base case**
	2	(5,10)	3.46E−03	−6.13%	(4,20)	1.0071	−1.06%
	3	(4,20)	3.60E−03	−10.43%	(4,10)	1.0619	−6.56%
Panel B							
Measure of prediction error		MAE			RMSE		
Network architecture	Ranking	*(l,n)*	MAE	Var%	*(l,n)*	RMSE	Var%
GARCH			9.83E−02	−248.58%		1.05E−01	−83.89%
S1	1	(5,20)	4.63E−02	−64.18%	(5,20)	6.52E−02	−14.19%
	2	(4,20)	5.18E−02	−83.69%	(4,20)	6.83E−02	−19.61%
	3	(4,10)	5.22E−02	−85.11%	(4,10)	6.84E−02	−19.79%
S2	1	(5,20)	3.77E−02	−33.69%	(6,10)	6.17E−02	−8.06%
	2	(6,10)	4.09E−02	−45.04%	(5,20)	6.21E−02	−8.76%
	3	(6,20)	4.25E−02	−50.71%	(6,20)	6.35E−02	−11.21%
S3	1	(6,10)	3.91E−02	−38.65%	(6,10)	6.22E−02	−8.93%
	2	(5,20)	3.95E−02	−40.07%	(5,20)	6.25E−02	−9.46%
	3	(4,10)	4.24E−02	−50.35%	(4,20)	6.36E−02	−11.38%
GARCH-S1	1	(5,20)	3.68E−02	−30.50%	(5,20)	6.71E−02	−17.51%
	2	(6,20)	4.97E−02	−76.24%	(6,20)	6.82E−02	−19.44%
	3	(5,10)	5.23E−02	−85.46%	(5,10)	6.86E−02	−20.14%
GARCH-S2	1	(5,20)	2.81E−02	0.35%	(5,20)	5.79E−02	−1.40%
	2	(5,10)	3.90E−02	−38.30%	(5,10)	6.23E−02	−9.11%
	3	(4,10)	3.99E−02	−41.49%	(4,10)	6.23E−02	−9.11%
GARCH-S3	**1**	**(4,20)**	**2.82E−02**	**Base case**	**(4,10)**	**5.71E−02**	**Base case**
	2	(6,10)	2.92E−02	−3.55%	(5,10)	5.88E−02	−2.98%
	3	(4,10)	3.01E−02	−6.74%	(4,20)	6.00E−02	−5.08%

Var% is the percentage variation of the error measures. S1, S2 and S3 represent the ANN, LSTM-ANN and BLSTM-ANN respectively.

4. Conclusions

We have investigated the integration of deep learning methodology and GARCH model to improve the prediction of copper price volatility. Facing the long memory phenomenon in time series data, recurrent neural networks (RNNs) are inherently suitable to distill information from sequences of inputs. Thus, the conventional ANN is combined with two RNNs (LSTM and BLSTM) to generate hybrid neural networks. Besides, GARCH models are trained and their best forecasts are used as extra inputs to augment the training data for the hybrid neural networks. As a result, six different types of hybrid neural network models (i.e., ANN, LSTM-ANN, BLSTM-ANN, GARCH-ANN, GARCH-LSTM-ANN, and GARCH-BLSTM-ANN) are developed and tested. We further conduct a series of sensitivity analysis of the ANN configuration (i.e., hidden layers and neurons) for each neural network architecture to optimize the prediction performance.

Through the empirical analyses, we have several major findings that can contribute to the literature. First, we find the GARCH forecasts can serve as informative features to substantially boost the volatility prediction, which complements the results of Kristjanpoller and Hernández [6]. We also find that incorporating RNNs (LSTM and BLSTM) into the hybrid

Table 5
Prediction performances of four-week ahead forecasts for different hybrid neural network configurations.

Panel A							
Measure of prediction errors		MSE			MAPE		
Network architecture	Ranking	*(l,n)*	MSE	Var%	*(l,n)*	MAPE	Var%
GARCH			1.35E−02	−258.09%		4.8537	−639.89%
S1	1	(5,10)	4.19E−03	−11.14%	(5,10)	1.6659	−153.95%
	2	(4,20)	4.71E−03	−24.93%	(4,20)	2.1890	−233.69%
	3	(4,10)	4.92E−03	−30.50%	(4,10)	2.2993	−250.50%
S2	1	(4,10)	4.05E−03	−7.43%	(4,10)	1.4334	−118.51%
	2	(5,20)	4.22E−03	−11.94%	(5,20)	1.6980	−158.84%
	3	(4,20)	4.36E−03	−15.65%	(5,10)	1.7951	−173.64%
S3	1	(5,20)	4.08E−03	−8.22%	(4,10)	1.5288	−133.05%
	2	(4,10)	4.16E−03	−10.34%	(5,20)	1.6633	−153.55%
	3	(5,10)	4.27E−03	−13.26%	(5,10)	1.7930	−173.32%
GARCH-S1	1	(5,20)	4.62E−03	−22.55%	(5,20)	1.8373	−180.08%
	2	(6,10)	4.70E−03	−24.67%	(6,10)	2.1475	−227.36%
	3	(4,20)	4.95E−03	−31.30%	(6,20)	2.2763	−247.00%
GARCH-S2	**1**	**(6,10)**	**3.77E−03**	**Base case**	**(4,20)**	**0.656**	**Base case**
	2	(5,10)	3.87E−03	−2.65%	(5,10)	0.8552	−30.37%
	3	(6,20)	3.90E−03	−3.45%	(6,10)	0.9237	−40.81%
GARCH-S3	1	(5,20)	3.65E−03	3.18%	(6,20)	1.1977	−82.58%
	2	(5,10)	4.16E−03	−10.34%	(4,20)	1.2629	−92.52%
	3	(6,20)	4.17E−03	−10.61%	(5,10)	1.3316	−102.99%
Panel B							
Measure of prediction errors		MAE			RMSE		
Network architecture	Ranking	*(l,n)*	MAE	Var%	*(l,n)*	RMSE	Var%
GARCH			1.10E−01	−280.62%		1.16E−01	−88.93%
S1	1	(5,10)	4.12E−02	−42.56%	(5,10)	6.47E−02	−5.37%
	2	(4,20)	5.04E−02	−74.39%	(4,20)	6.86E−02	−11.73%
	3	(4,10)	5.30E−02	−83.39%	(4,10)	7.01E−02	−14.17%
S2	1	(4,10)	3.77E−02	−30.45%	(4,10)	6.36E−02	−3.58%
	2	(5,20)	4.18E−02	−44.64%	(5,20)	6.50E−02	−5.86%
	3	(5,10)	4.35E−02	−50.52%	(4,20)	6.60E−02	−7.49%
S3	1	(4,10)	3.93E−02	−35.99%	(5,20)	6.39E−02	−4.07%
	2	(5,20)	4.10E−02	−41.87%	(4,10)	6.45E−02	−5.05%
	3	(5,10)	4.31E−02	−49.13%	(5,10)	6.54E−02	−6.51%
GARCH-S1	1	(5,20)	4.44E−02	−53.63%	(5,20)	6.80E−02	−10.75%
	2	(6,10)	4.99E−02	−72.66%	(6,10)	6.85E−02	−11.56%
	3	(6,20)	5.25E−02	−81.66%	(4,20)	7.03E−02	−14.50%
GARCH-S2	**1**	**(6,10)**	**2.89E−02**	**Base case**	**(6,10)**	**6.14E−02**	**Base case**
	2	(5,10)	2.92E−02	−1.04%	(5,10)	6.22E−02	−1.30%
	3	(4,20)	2.96E−02	−2.42%	(6,20)	6.25E−02	−1.79%
GARCH-S3	1	(6,20)	3.50E−02	−21.11%	(5,20)	6.04E−02	1.63%
	2	(5,10)	3.64E−02	−25.95%	(5,10)	6.45E−02	−5.05%
	3	(5,20)	3.69E−02	−27.68%	(6,20)	6.45E−02	−5.05%

Note: Var% is the percentage variation of the error measures. S1, S2 and S3 represent the ANN, LSTM-ANN and BLSTM-ANN respectively.

GARCH-ANN network can further improve the volatility prediction performance; such a finding highlights the efficacy of deep learning methodology and advances the results of Kristjanpoller and Hernández [6]. Besides, we have conducted a variety of experiments to find the best architecture of the hybrid neural network. In particular, we find that the choice between LSTM and BLSTM networks should be dependent on the forecast horizon. It appears that BLSTM works better for three-week ahead volatility forecasts while LSTM works better for other cases. Further, the empirical results suggest that the ANN configuration in the hybrid model should be optimized depending on the choice of the measure of prediction errors. From these findings, an effective hybrid approach is developed to construct useful volatility prediction models through neatly integrating GARCH forecasts with ANN, LSTM, and BLSTM networks. This proposed hybrid approach, we hope, will motivate further investigation into the exploration and exploitation of using deep learning methodology in solving time series forecasting problems.

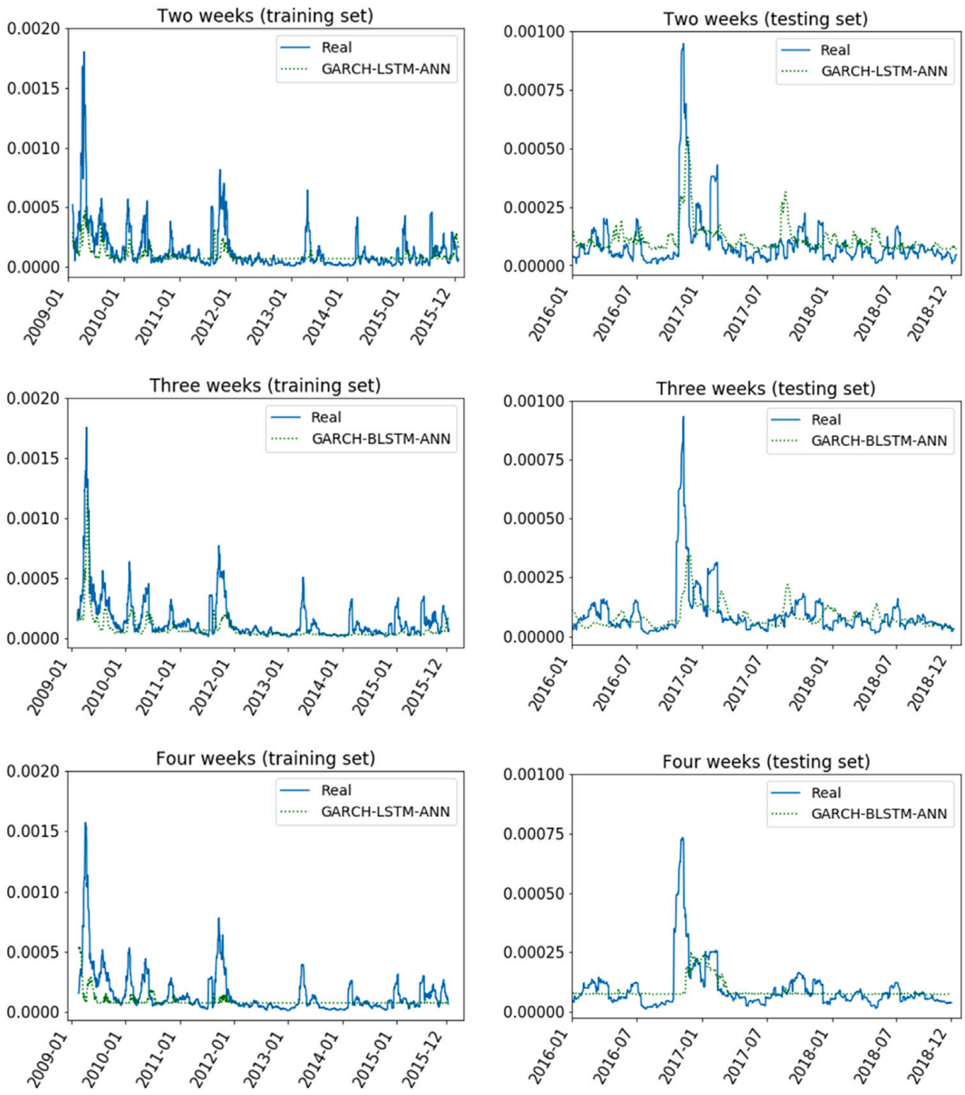

Fig. 7. Predicted and realized copper price volatilities.

CRediT authorship contribution statement

Yan Hu: Data curation, Software, Formal analysis, Visualization, Writing - original draft. **Jian Ni:** Conceptualization, Methodology, Supervision, Resources, Writing - review & editing. **Liu Wen:** Software, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank the Editor and the anonymous referees for their insightful comments and suggestions for the revision of this paper. Any remaining errors are our responsibility. This research was supported by National Social Science Fund of China (No. 19ZDA074), Fundamental Research Funds for the Central Universities (No. JBK2007029), and National Natural Science Foundation of China (No. 71801112).

Appendix

See Table A.1.

Table A.1
Principal component analysis results.

Panel A

Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
AP	0.21	−0.24	−0.14	−0.10	0.57	0.05	0.09	0.14	−0.13	0.08
CP	0.27	−0.27	−0.05	−0.23	−0.06	0.08	0.10	−0.20	−0.05	0.16
CSI	0.15	−0.01	−0.12	−0.05	−0.40	−0.04	−0.28	0.73	0.14	0.18
DJIA	0.14	0.32	−0.01	−0.47	0.00	0.17	0.17	0.08	0.09	−0.39
EUR	0.13	−0.13	0.48	−0.20	0.01	−0.04	−0.11	0.01	0.19	0.34
FTSE	0.20	0.32	−0.04	−0.40	−0.03	0.07	0.07	0.04	0.15	−0.16
GP	0.09	−0.13	0.55	−0.08	−0.07	0.07	0.31	0.00	0.17	0.24
HAP	0.25	−0.16	−0.14	0.07	0.47	0.03	−0.02	0.37	0.02	−0.05
HCP	0.29	−0.23	−0.06	0.01	−0.17	0.07	0.01	0.04	0.05	0.01
HZP	0.29	−0.21	−0.07	0.11	−0.24	0.06	0.04	0.05	−0.20	−0.21
LAP	0.26	0.02	0.09	0.23	0.24	−0.18	−0.12	−0.11	0.48	−0.23
LCP	0.32	0.00	0.04	0.09	−0.08	−0.04	−0.05	−0.17	0.27	−0.10
LMAP	0.22	0.34	0.09	0.23	0.20	−0.10	−0.06	−0.03	−0.05	0.22
LMCP	0.27	0.33	0.00	0.10	−0.02	0.01	0.01	−0.04	−0.17	0.24
LMZP	0.25	0.31	0.02	0.26	−0.10	−0.03	−0.01	−0.02	−0.35	0.16
LZP	0.30	−0.01	0.03	0.26	−0.20	−0.07	−0.05	−0.16	0.06	−0.31
OIL	0.17	0.30	0.01	−0.17	0.09	0.13	0.10	0.00	−0.15	0.19
SHI	−0.01	0.00	0.11	0.07	0.07	0.81	−0.54	−0.13	0.01	−0.02
USD	−0.04	0.04	−0.36	0.31	−0.09	0.42	0.55	0.01	0.40	0.26
YEN	−0.07	0.02	0.50	0.28	0.04	0.20	0.33	0.37	−0.22	−0.37
ZP	0.25	−0.30	−0.06	−0.11	−0.14	0.08	0.15	−0.18	−0.35	−0.03
CuP (%)	30.39	42.42	49.41	54.77	59.76	64.62	69.16	73.39	77.12	80.67

Panel B

Variable	PC11	PC12	PC13	PC14	PC15	PC16	PC17	PC18	PC19	PC20	PC21
AP	0.03	−0.14	0.05	0.19	0.28	−0.20	−0.08	0.29	0.46	0.05	−0.04
CP	−0.12	−0.05	−0.32	0.31	−0.02	0.01	0.36	−0.19	−0.25	0.42	−0.30
CSI	−0.09	−0.07	0.10	0.31	0.06	−0.07	−0.06	0.01	−0.02	0.01	−0.01
DJIA	0.17	−0.10	−0.01	0.05	−0.39	−0.47	−0.02	0.00	0.00	0.01	0.05
EUR	0.59	0.40	−0.08	−0.06	0.01	−0.05	−0.03	0.01	0.06	−0.02	0.00
FTSE	0.13	−0.16	0.01	−0.11	0.43	0.61	0.11	0.02	0.05	−0.04	−0.01
GP	−0.33	−0.44	0.37	−0.19	−0.01	−0.07	−0.01	0.00	−0.02	−0.01	−0.01
HAP	0.04	0.03	0.05	−0.43	−0.05	0.01	0.07	−0.41	−0.41	−0.07	0.04
HCP	−0.21	0.05	−0.39	−0.26	−0.28	0.08	0.34	0.24	0.29	−0.44	0.14
HZP	0.14	0.08	0.23	−0.31	−0.28	0.24	−0.28	0.16	0.19	0.47	−0.20
LAP	−0.06	0.08	0.16	0.27	−0.16	0.13	0.07	0.46	−0.31	0.02	−0.04
LCP	−0.21	0.05	−0.32	−0.06	0.26	−0.15	−0.41	−0.21	0.08	0.25	0.48
LMAP	0.07	−0.22	−0.02	0.28	−0.44	0.28	−0.03	−0.39	0.33	−0.01	0.04
LMCP	−0.03	−0.11	−0.31	−0.14	0.08	−0.12	−0.42	0.22	−0.25	−0.23	−0.48
LMZP	0.23	−0.14	0.09	−0.11	0.15	−0.20	0.43	0.25	−0.15	0.22	0.39
LZP	0.08	0.08	0.26	0.04	0.30	−0.28	0.22	−0.34	0.23	−0.25	−0.37
OIL	−0.47	0.67	0.29	0.03	−0.01	0.01	0.04	−0.01	0.05	0.02	0.03
SHI	−0.02	−0.09	0.05	0.02	0.01	0.01	0.00	0.01	−0.01	0.00	0.00
USD	0.22	0.08	0.03	0.05	0.01	0.00	−0.01	−0.01	0.01	0.00	0.01
YEN	−0.02	0.14	−0.31	0.22	0.11	0.12	0.03	0.00	−0.03	0.03	−0.01
ZP	0.15	−0.02	0.22	0.36	−0.02	0.15	−0.27	−0.04	−0.28	−0.41	0.31
CuP (%)	83.88	86.87	89.58	91.99	93.95	95.71	96.94	97.92	98.85	99.45	100

Note: CuP represents cumulative proportion of explained variance.

References

[1] W. Kristjanpoller, M. Minutolo, Gold price volatility: A forecasting approach using the artificial neural network–GARCH model, Expert Syst. Appl. 42 (5) (2015) 7245–7251.

[2] D. García, W. Kristjanpoller, An adaptive forecasting approach for copper price volatility through hybrid and non-hybrid models, Appl. Soft Comput. 74 (2019) 466–478.

[3] L.H. Ederington, J.H. Lee, How markets process information: News releases and volatility, J. Finance 48 (4) (1993) 1161–1191.

[4] A.M. Fuertes, M. Izzeldin, E. Kalotychou, On forecasting daily stock volatility: the role of intraday information and market conditions, Int. J. Forecast. 25 (2) (2009) 259–281.

[5] W. Kristjanpoller, A. Fadic, M.C. Minutolo, Volatility forecast using hybrid neural network models, Expert Syst. Appl. 41 (5) (2014) 2437–2442.

[6] W. Kristjanpoller, P.E. Hernández, Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors, Expert Syst. Appl. 84 (2017) 290–300.

[7] H.Y. Kim, C.H. Won, Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models, Expert Syst. Appl. 103 (2018) 25–37.
[8] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, J. Econometrics 31 (3) (1986) 307–327.
[9] D.B. Nelson, Conditional heteroscedasticity in asset returns: A new approach, Econometrica 59 (2) (1991) 347–370.
[10] Z. Ding, R. Engle, C. Granger, A long memory property of stock market returns and a new model, J. Empir. Financ. 1 (1993) 83–106.
[11] J.D. Hamilton, R. Susmel, Autoregressive conditional heteroscedasticity and changes in regime, J. Econometrics 64 (1994) 307–333.
[12] Y.K. Tse, The conditional heterscedasticity of the yen-dollar exchange rate, J. Appl. Econometrics 13 (1) (1998) 49–55.
[13] J. Davidson, Moment and memory properties of linear conditional heteroscedasticity models, and a new model, J. Bus. Econ. Stat. 1 (2004) 16–29.
[14] K. Bergerson, D.C. Wunsch, A commodity trading model based on a neural network-expert system hybrid, in: Proceedings of the International Joint Conference on Neural Networks, Vol. 91, 1991, pp. 289–293.
[15] R. Sharda, R.B. Patil, A connectionist approach to time series prediction: an empirical test, J. Intell. Manuf. 3 (1992) 317–323.
[16] S.A. Hamid, Z. Iqbal, Using neural networks for forecasting volatility of S & P 500 index futures prices, J. Bus. Res. 57 (10) (2004) 1116–1125.
[17] A. Parisi, F. Parisi, D. Díaz, Forecasting gold price changes: Rolling and recursive neural network models, J. Multinational Financ. Manag. 18 (5) (2008) 477–487.
[18] A. Azadeh, M. Moghaddam, M. Khakzad, V. Ebrahimipour, A flexible neural network-fuzzy mathematical programming algorithm for improvement of oil price estimation and forecasting, Comput. Ind. Eng. 62 (2) (2012) 421–430.
[19] A. Yazdani-Chamzini, S.H. Yakhchali, D. Volungeviciene, E.K. Zavadskas, Forecasting gold price changes by using adaptive network fuzzy inference system, J. Bus. Econ. Manag. 13 (5) (2012) 994–1010.
[20] M. Bildirici, Ö.Ö. Ersin, Forecasting oil prices: Smooth transition and neural network augmented GARCH family models, J. Pet. Sci. Eng. 109 (2013) 230–240.
[21] L. Cui, K. Huang, H.J. Cai, Application of a TGARCH-wavelet neural network to arbitrage trading in the metal futures market in China, Quant. Finance 15 (2) (2015) 371–384.
[22] X. Lu, D. Que, G. Cao, Volatility forecast based on the hybrid artificial neural network and GARCH-type models, Procedia Comput. Sci. 91 (2016) 1044–1049.
[23] S. Lahmiri, Modeling and predicting historical volatility in exchange rate markets, Physica A 471 (2017) 387–395.
[24] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, J. Schmidhuber, A novel connectionist system for improved unconstrained handwriting recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (5) (2009) 855–868.
[25] X.G. Li, X.H. Wu, Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4520–4524.
[26] M. Herman, B. Schrauwen, Training and analyzing deep recurrent neural networks, Adv. Neural Inf. Proces. Syst. (NIPS) 26 (2013).
[27] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[28] N. Maknickiene, A. Maknickas, Application of neural network for forecasting of exchange rates and forex trading, in: Proceedings of the 7th International Scientific Conference on Business and Management, 2012, pp. 10–11.
[29] D.M.Q. Nelson, A.C.M. Pereira, R.A.d. Oliveira, Stock market's price movement prediction with LSTM neural networks, in: International Joint Conference on Neural Networks, 2017, pp. 1419–1426.
[30] Y.X. Wu, Q.-B. Wu, J.Q. Zhu, Improved EEMD-based crude oil price forecasting using LSTM networks, Physica A 516 (2019) 114–124.
[31] G. Liu, J. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification, Neurocomputing 337 (2019) 325–338.
[32] M. Sardelich, S. Manandhar, Multimodal deep learning for short-term stock volatility prediction, 2018, arXiv:1812.10479.
[33] J. Eapen, D. Bein, A. Verma, Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction, in: IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), 2019, pp. 0264–0270.
[34] S.R. Bentes, Forecasting volatility in gold returns under the GARCH, IGARCH and FIGARCH frameworks: New evidence, Physica A 438 (2015) 355–364.
[35] J. Zhang, Y. Zhang, L. Zhang, A novel hybrid method for crude oil price forecasting, Energy Econ. (2015) 649–659.
[36] G. Sermpinis, C.L. Dunis, J. Laws, C. Stasinakis, Forecasting and trading the EUR/USd exchange rate with stochastic neural network combination and time-varying leverage, Decis. Support Syst. 54 (1) (2012) 316–329.
[37] A. Petropoulos, S.P. Chatzis, V. Siakoulis, N. Vlachogiannakis, A stacked generalization system for automated FOREX portfolio trading, Expert Syst. Appl. 90 (2017) 290–302.
[38] J. Henriquez, W. Kristjanpoller, A combined independent component analysis–neural network model for forecasting exchange rate variation, Appl. Soft Comput. (2019).
[39] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM, Neural Comput. 12 (10) (2000) 2451–2471.