Forecasting WTI Futures Prices Using Recurrent Neural Networks

Review of Integrative Business & Economics Research

Kohei Matsuoka
Kobe University, Kobe, Japan

Shigeyuki Hamori*
Kobe University, Kobe, Japan

ABSTRACT

Within the context of crude oil futures pricing, this study compares the forecast performance of recurrent neural network (RNN)-based models to that of an autoregressive-generalized autoregressive conditional heteroscedasticity (AR-GARCH) model and a vector autoregression (VAR) model. For the RNN-based models, we use simple recurrent network, long short-term memory, and gated recurrent unit models, as well as hybrids thereof. Our empirical results indicate that RNN-based models outperform the AR-GARCH model and VAR model. We also find that neither adding the same type of layer nor combining different types of layers statistically significantly improves forecast performance.

Keywords: Crude oil futures prices forecasting; Recurrent neural network; Simple recurrent network; Long short-term memory; Gated recurrent unit; Hybrid model.

Received 28 July 2019 | Revised 1 December 2019 | Accepted 20 January 2020.

1. INTRODUCTION

At present, crude oil is essential to the world economy. Many oil products contribute to our daily lives and have become necessities. However, the prices of crude oil are highly unstable and respond to various factors such as U.S. monetary policy, reduced cooperation among Organization of the Petroleum Exporting Countries in terms of oil production, and stock price movements. The complex interrelationships among these factors have had disastrous effects on the dynamics of crude oil prices. According to Bildirici (2019), crude oil prices exhibit nonlinear and chaotic behaviors, making it difficult for one to forecast their prices. To stabilize oil exporters and importers’ economies — and consequently that of the world — in the face of related risks, we need to build an accurate forecasting model that can capture the nonlinear and chaotic behavior of oil prices.

Several studies have attempted to build such models. Moshiri and Foroutan (2006) built feedforward multilayer artificial neural network (ANN), standard autoregressive moving average (ARMA), and exponential generalized autoregressive conditional heteroscedasticity (EGARCH) models, and found that the ANN model outperforms the other two. Wen *et al.* (2006) found that a support vector machine outperforms ARMA and a back-propagation neural network (BPNN) in terms of forecasting performance. Wang and Wang (2016) propose an Elman recurrent neural network (ERNN) and use an ERNN with a stochastic time-effective (ST-ERNN) function; they found that ST-ERNN is superior to both ERNN and BPNN. Chen *et al.* (2017) used a long short-term memory (LSTM) model to forecast crude oil prices.

In the current study, we compare the forecast performance of recurrent neural network (RNN)-based models, an autoregressive-generalized autoregressive conditional heteroscedasticity (AR-GARCH) model, and a vector autoregression (VAR) model. We use four types of RNN-based models—namely, simple recurrent network (SRN), LSTM, and gated recurrent unit (GRU) models, as well as hybrids thereof. The hybrid models include different types of RNNs for the first and second layers.

Our empirical results indicate that RNN-based models outperform AR-GARCH and VAR models. We also found that neither adding the same type of layer nor combining different layer types has a statistically significant effect on forecast performance.

This paper is organized as follows. In Section 2, we briefly describe the AR-GARCH, VAR, SRN, LSTM, and GRU models. Then, in Section 3, we provide information on our data. We describe our forecasting models and empirical results in Section 4. Finally, Section 5 concludes.

2. METHODOLOGY

2.1 Autoregressive-Generalized Autoregressive Conditional Heteroscedasticity

AR-GARCH models can capture linear time relationships with heteroscedasticity. For this reason, this model is used to analyze stock returns (Sembiring *et al.*, 2016), for example. An AR(k)-GARCH(p, q) model is expressed as follows.

$$Y_t = \theta + \sum_{l=1}^{k} \psi_l Y_{t-l} + \varepsilon_t , \tag{1}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \sigma_{t-i}^2 + \sum_{j=1}^{q} \beta_j \varepsilon_{t-j}^2 , \tag{2}$$

$$\varepsilon_t \sim WZ(0, \sigma_t^2). \tag{3}$$

Here, ε_t represents the error term at time *t*, and $WZ(0, \sigma_t^2)$ is the white noise whose mean is 0 and variance is σ_t^2. In this study, we assume $WZ(0, \sigma_t^2)$ as Student's t distribution.

2.2 Vector Autoregression

A VAR model is a linear multivariate autoregressive model that can capture linear time interrelationships among multiple variables. The model with *p* lags is

$$\mathrm{Y_t} = \mathrm{C} + \sum_{\mathrm{i=1}}^{\mathrm{p}} \Phi_i Y_{t-i} + \epsilon_t \,, \tag{4}$$

where $\mathrm{C}, \Phi,$ and ϵ_t represent a constant vector, coefficient matrix, and error term vector at time *t*, respectively, which satisfies the following conditions:

$$\begin{aligned} \mathrm{E}[\epsilon_\mathrm{t}] &= 0 \\ \mathrm{V}[\epsilon_\mathrm{t}] &= \Sigma \\ E[\epsilon_t \epsilon_{t-s}] &= 0 \quad for\ s > 0. \end{aligned} \tag{5}$$

Here, E[ ] is the expectations operator, V[ ] is the variance operator, and Σ is a nondiagonal covariance matrix of ϵ_t.

2.3 Fully Connected Layer

A fully connected feedforward neural network (FNN) is a traditional type of ANN often used to aggregate the output of the last layer and transform it into an adequate shape. Let Y_t and X_t denote the output vector and the input vector at time *t*, respectively. The output vector is then defined as

$$\mathrm{Y_t} = f(WX_t + b), \tag{6}$$

where $f(\cdot)$ represents the activation function and W is a weight matrix.

2.4 Recurrent Neural Network

An RNN (Rumelhart *et al.*, 1986) is an extension of an FNN. Unlike FNNs, RNNs can maintain time-series dynamics by using their internal memory. This structure is given by:

$$\mathrm{M_t} = f(X_t,\ \ M_{t-1}), \tag{7}$$

where M_t and X_t are the internal memory and input vector at time *t*, respectively, and $f(\cdot)$ is the activation function. A sigmoid function is a common activation function, and it enables RNNs to capture nonlinear interrelationships among multiple variables of an input—something that VAR cannot do. Eq. (7) implies that the internal memory at time *t* – 1 is fed back to that at time *t* for $t \geq 1$. This dependency enables RNNs to hold time-series dynamics.

An SRN is a basic RNN model whose output is computed as follows.

$$Y_t = \tanh(W_Y Y_{t-1} + W_X X_t + b_Y)\,. \tag{8}$$

Here, Y_t and X_t denote the output vector and input vector at time *t*, respectively. *W* and

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

b are a weight matrix and bias vector, respectively, and $\tanh(\cdot)$ is a hyperbolic tangent. Eq. (8) introduces internal memory, which enables the SRN to store time-dependent relationships.

2.5 Long Short-Term Memory

While an SRN can learn short-term dependency, it has difficulty learning long-term dependency, on account of the vanishing gradient problem. Hochreiter and Schmidhuber (1997) first developed LSTM to overcome this problem, and Gers *et al.* (2000) later introduced LSTM with a forget gate. In the current study, we follow the latter.

Let $F_t, I_t, O_t, C_t, and\ Y_t$ denote the forget gate vector, input gate vector, output gate vector, cell state (a state of a memory cell) vector, and output vector at time *t*, respectively. The forward pass is

$$F_t = sigmoid(W_{FX}X_t + W_{FH}Y_{t-1} + b_F), \tag{9}$$
$$I_t = sigmoid(W_{IX}X_t + W_{IH}Y_{t-1} + b_I), \tag{10}$$
$$\mathrm{O_t} = sigmoid(W_{OX}X_t + \mathrm{W_{OH}Y_{t-1}} + b_O), \tag{11}$$
$$C_t = F_t * C_{t-1} + I_t * \tanh(W_{CX}X_t + W_{CH}Y_{t-1} + b_C), \tag{12}$$
$$Y_t = O_t * \tanh(C_t), \tag{13}$$

where $\mathrm{sigmoid}(\cdot)$ and $*$ represent a sigmoid function and Hadamard product operator, respectively. *W* and *b* are a weight matrix and bias vector, respectively.

Unlike SRN, LSTM has four types of time-recurrent structure—namely, a forget gate, input gate, output gate, and memory cell. A forget gate decides when to reset the information from the previous cell state, while an input gate decides when to add the information from the input and the previous output to the current cell state; an output gate decides when to output the information from the current cell state. A memory cell stores information from the previous inputs, current inputs, and previous outputs. Given the structure of a memory cell, as stated in Eq. (12), LSTM precludes the vanishing gradient problem.

2.6 Gated Recurrent Unit

LSTM is used successfully in many tasks, such as machine translation (Sutskever *et al.*, 2014; Yao *et al.*, 2015; Wu *et al.*, 2016). However, owing to its structural complexity, LSTM does have a drawback in terms of its learning speed. Cho *et al.* (2014) therefore propose GRU to reduce the complexity.

Let $U_t, R_t, X_t,$ and Y_t denote the update gate vector, reset gate vector, input vector, and output vector at time *t*, respectively. The forward pass is

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

$$\mathrm{U_t} = sigmoid(W_{UX}X_t + W_{UY}Y_{t-1} + b_U), \tag{14}$$
$$\mathrm{R_t} = sigmoid(W_{RX}X_t + W_{RY}Y_{t-1} + b_R), \tag{15}$$
$$\mathrm{Y_t} = (1 - U_t) * Y_{t-1} + U_t * \tanh(W_{YX}X_t + W_{YR}(R_t * Y_{t-1}) + b_Y), \tag{16}$$

where W and b represent a weight matrix and bias vector, respectively. In comparing Eqs. (9–10) and (12) to Eqs. (14) and (16), we see that GRU integrates a forget gate and input gate into an update gate and thus reduces the complexity of LSTM.

3. DATA

In this study, we use daily data pertaining to West Texas Intermediate (WTI) futures, NASDAQ-100, and US soy oil futures prices. We use US soy oil futures prices and NASDAQ-100 as explanatory variables. The data, which we obtained from Investing.com,[1] covers the March 31, 1983–February 21, 2019 period, and consists of 9,004 observations after removing data pertaining to days with missing values. Fig. (1) shows the original WTI futures prices, NASDAQ-100, and US soy oil futures prices, and Fig. (2) shows their serial correlation. As we see in Fig. (2), the data seems to have serial correlation. Since VAR is not appropriate for modeling data with serial correlation, we must remove it first. Among the available methods, we use first-log differencing:

$$\Delta_t = \log(X_t) - \log(X_{t-1}), \tag{17}$$

where Δ_t and X_t represent the new data and original data at time *t*, respectively. We illustrate their serial correlations in Figs. (3) and (4), after applying first-log differencing. Fig. (4) indicates that the method seems to have removed the serial correlation in the data.

Feature scaling can promote better performance and faster training, and we therefore rescale the data using standardization. Let Z and Δ denote the standardized data and the data before standardization, respectively. The standardization is described by:

$$Z = \frac{\Delta - \mu}{\sigma}, \tag{18}$$

where σ and μ represent the sample standard deviation and sample mean of attribute Δ, respectively. Using this method, we can transform the mean and standard deviation of the data into 0 and 1. Fig. (5) depicts the standardized data. In comparing Fig. (3) to Fig. (5), we can confirm that we rescaled the data successfully.

[1] See https://www.investing.com/ (accessed February 22, 2019).

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

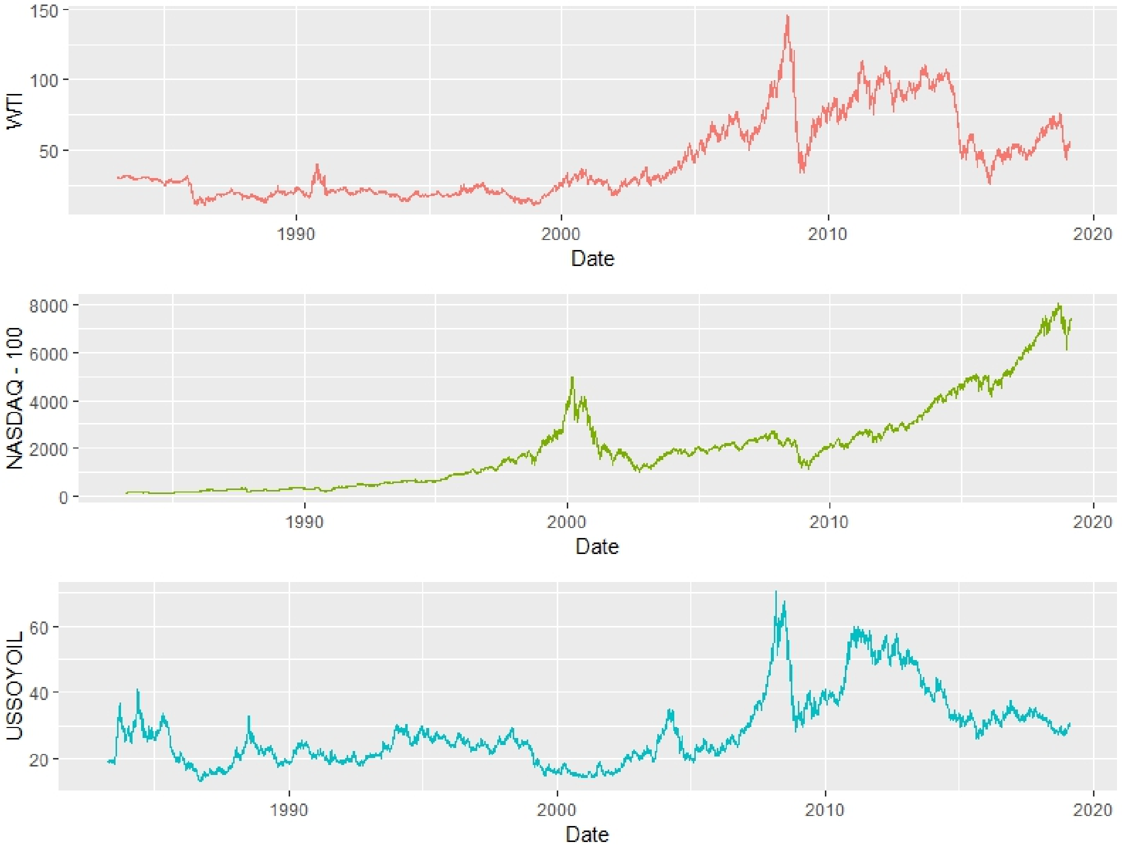

Figure 1. Original WTI futures prices (**top**), NASDAQ-100 (**middle**), US soy oil futures prices (**bottom**).

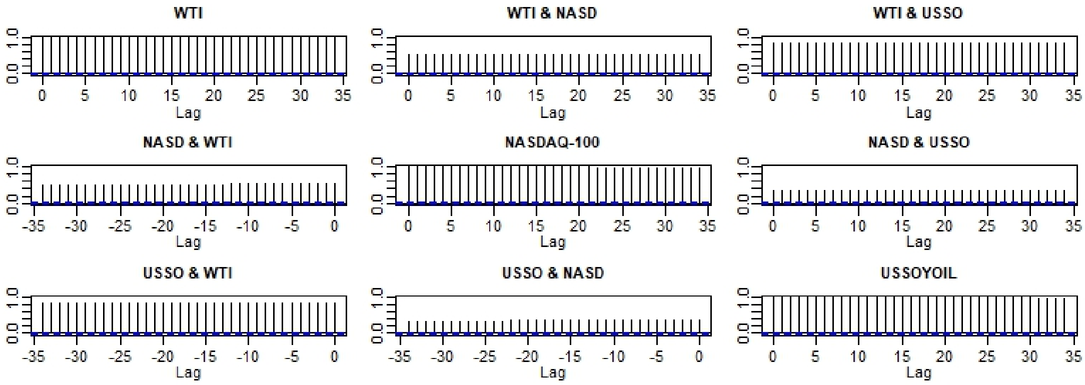

Figure 2. Serial correlation of the original data. In this figure, NASD and USSO stand for NASDAQ-100 and US soy oil futures prices, respectively.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

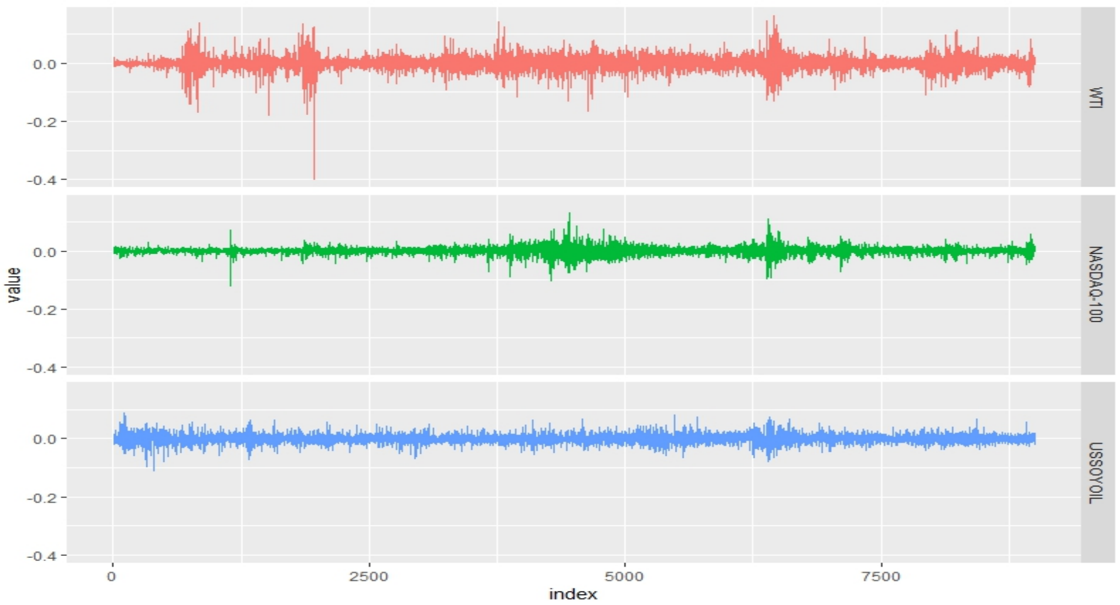

Figure 3. WTI futures prices (**top**), NASDAQ-100 (**middle**), and US soy oil futures prices (**bottom**), after first-log differencing.

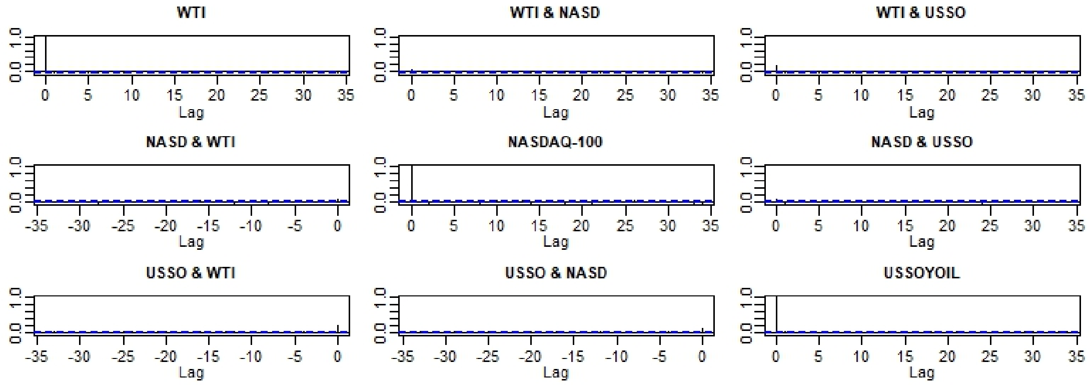

Figure 4. Serial correlation of the data after first-log differencing.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

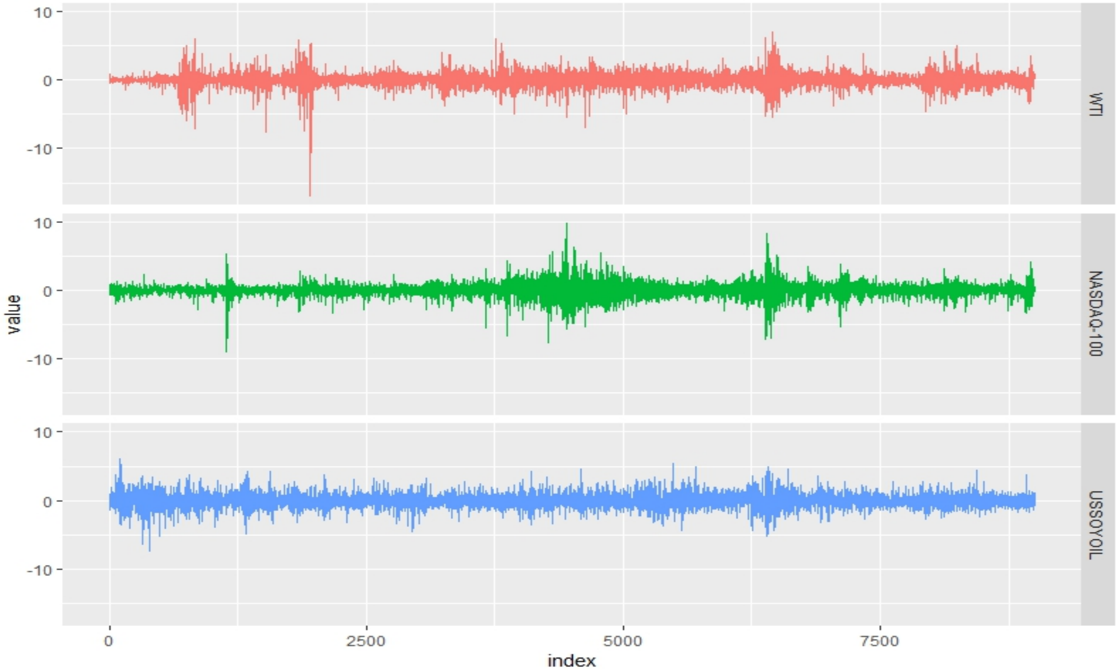

Figure 5. Standardized WTI futures prices (**top**), standardized NASDAQ-100 (**middle**), and standardized US soy oil futures prices (**bottom**). Standardization is performed after first-log differencing.

4. EMPIRICAL RESULTS

4.1 Evaluation Criteria

To evaluate the forecasting models, we use the mean absolute error (MAE) and mean directional accuracy (MDA), and apply the Diebold–Mariano test (DM test) to the forecasting models to examine whether or not there are statistically significant differences in forecast performance.

4.1.1 Mean Absolute Error

MAE is a measure of the differences between actual and predicted values. We calculate it as follows.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |f_i - Z_i| \,, \tag{19}$$

where Z_i, f_i, and N are the actual and predicted values of the standardized data at time *i* and the sample size, respectively. For this criterion, a model with lower MAE values have better forecasting ability.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

4.1.2 Mean Directional Accuracy

MDA is a measure of the differences between the forecast direction and the direction in which the actual value fluctuates. This criterion is defined as follows.

$$\text{MDA} = \frac{1}{N}\sum_{i}^{N} \eta_i\,, \tag{20}$$

$$\eta_i = \begin{cases} 1, & sign(X_i - X_{i-1}) = sign(\widehat{X}_i - X_{i-1}) \\ 0, & otherwise \end{cases}\,, \tag{21}$$

where $sign(\cdot)$ represents the sign function, and X_i and $\widehat{X_\iota}$ are the observed and predicted values of the original data at time *i*, respectively. Note that we can transform $\text{sign}(X_i - X_{i-1})$ as follows.

$$\begin{aligned} sign(X_i - X_{i-1}) &= sign(\log(X_i) - \log(X_{i-1})) \\ &= sign(\Delta_\text{i}) && \because \Delta_\text{i} = \log(X_i) - \log\,(X_{i-1}) \\ &= sign\left(Z_i + \frac{\mu}{\sigma}\right) && \because Z_i = \frac{\Delta_\text{i} - \mu}{\sigma}, \sigma > 0, \end{aligned} \tag{22}$$

where μ and σ represent the sample mean and standard deviation of attribute $\varDelta$, respectively. We also can transform $sign(\widehat{X_\iota} - X_{i-1})$ as follows:

$$\begin{aligned} sign(\widehat{X}_\text{i} - X_{i-1}) &= sign(\log(\widehat{X}_i) - \log(X_{i-1})) \\ &= sign(\widehat{\Delta}_\text{i}) && \because \widehat{\Delta}_\text{i} = \log(\widehat{X}_i) - \log\,(X_{i-1}) \\ &= sign\left(f_i + \frac{\mu}{\sigma}\right) && \because f_i = \frac{\widehat{\varDelta}_i - \mu}{\sigma}, \sigma > 0. \end{aligned} \tag{23}$$

Finally, using Eqs. (22) and (23), we can transform Eq. (21) into the following equation:

$$\eta_\text{i} = \begin{cases} 1, & sign(\text{Z}_i + \frac{\mu}{\sigma}) = sign(f_i + \frac{\mu}{\sigma}) \\ 0, & otherwise \end{cases}. \tag{24}$$

As Eq. (24) shows, we need to check the signs of $Z_\text{i} + \frac{\mu}{\sigma}$ and $f_i + \frac{\mu}{\sigma}$ when calculating MDA. With respect to this criterion, a higher MDA indicates better forecasting ability.

4.1.3 Diebold–Mariano Test

The DM test examines whether or not the null hypothesis (i.e., that the competing model has the same predictive accuracy) is statistically true. According to Diebold and Mariano (1995), the forecast error ϵ_{it} is

$$\epsilon_{it} = f_{it} - Z_t \;\; \text{i} = 1, 2, \tag{25}$$

where f_{it} and Z_t are the predicted and actual values at time *t*, respectively.
Let $\text{g}(\epsilon_{it})$ denote the loss function. The loss differential d_t is then

$$d_t = g(\epsilon_{1t}) - g(\epsilon_{2t}). \tag{26}$$

In this study, we use $g(\epsilon_{it}) = |\epsilon_{it}|$.

In this test, the null hypothesis states that $\mathrm{H}_0\colon \mathrm{E}[d_t] = 0 \ \ st$, while the alternative hypothesis is that $\mathrm{H}_1\colon E[d_t] \neq 0 \ \forall t$. We finally define the statistic for the DM test as

$$\mathrm{DM} = \frac{\bar{d}}{\sqrt{\frac{s}{N}}}, \tag{27}$$

where $\bar{d}, s,$ and N represent the sample mean, the variance of d_t, and the sample size, respectively. If the null hypothesis is true, then the DM statistic is asymptotically distributed as *N*(0, 1). Here, *N*(0, 1) represents the standard normal distribution.

To overcome the problem of multiple comparisons, we use the Bonferroni method (Bonferroni 1935, 1936), or set the significance level to (5%)/*n* and control the family-wise error at or under 5%. Additionally, we perform *n* number of tests.

4.2 Forecasting Models

We construct six main types of forecasting model—namely, AR-GARCH, VAR, SRN, LSTM, GRU, and hybrid models.

We construct the AR(1)-GARCH(1,1) model and VAR model with five lags, as benchmarks. The number of VAR lags is determined within 10 or fewer lags, according to the Akaike information criterion. We use the ordinary least squares method to estimate the model.

Each SRN, LSTM, and GRU model consists of one or two RNN layers of the same kind and one fully connected layer; meanwhile, each of the hybrid models consists of two different kinds of recurrent layers and one fully connected layer. An RNN layer is an SRN, LSTM, or GRU layer. We describe the hybrid models as, for example, SRNLSTM, which indicates that the first recurrent layer is an SRN layer and the second layer is an LSTM layer. In this study, we use the hard-sigmoid function as the sigmoid function for an LSTM or GRU layer, and the linear function as the activation function on a fully connected layer. The number of units in each recurrent layer is 20, 40, 60, or 80, and that of the fully connected layer is one. We set RMSprop as the optimization algorithm, MAE as the target of optimization, and the batch size to 32. Moreover, we use the weight that minimizes MAE on the validation data within 100 iterations, to preclude both overfitting and underfitting.

We divide the dataset into two parts, with 80% of it being used for training and the remaining 20% for evaluating the models. For the training models other than AR(1)-GARCH(1,1) and VAR, we use 10% of the training data as validation data.

Finally, we implement these models by using the Keras, TensorFlow, random, and NumPy modules in Python 3.2.6. To obtain reproducible results, we set the random seed in Python to 0, that in the NumPy module to 42, that in the random module to 12345, and that in the TensorFlow module to 1234; we also make the TensorFlow module use a single thread.

4.3 Empirical Results

Tables 1–5 report the MAE, MDA, and DM test results. As Tables 1–4 show, RNN-based models outperform the AR-GARCH model and VAR model in terms of both MAE and MDA—that is, nonlinear models are more suitable for forecasting WTI futures prices than are linear models. This implies that there may be nonlinear interdependencies among WTI futures prices, the NASDAQ-100, and US soy oil futures prices. Table 5 shows that all of the differences in forecast performance between the AR-GARCH and RNN-based models and between the VAR and RNN-based models are significant at the (5%)/31 level. Additionally, the two-layered SRN model with 60 units on the first layer and 40 units on the second layer has the lowest MAE, while the two-layered GRU model with 60 units on the first layer and 40 units on the second layer and the GRUSRN model with 80 units on the first layer and 20 units on the second layer have the highest MDA. The models that include an SRN layer tend to have better forecasting performance with respect to both MAE and MDA than models lacking one. This tendency indicates that the interrelationships among WTI futures prices, NASDAQ-100, and US soy oil futures prices may be short-term dependent, rather than long-term dependent.

Table 1. MAE and MDA results for AR(1)-GARCH(1,1), VAR, and the best SRN, LSTM, and GRU models, based on MAE

Model	Units	Recurrent Layers	MAE	MDA
AR(1)-GARCH(1,1)	–	–	0.9167	48.11%
VAR(5)	–	–	0.6351	47.50%
SRN	60	1	0.6269	52.05%
LSTM	40	1	0.6279	51.66%
GRU	40	1	0.6282	51.55%
SRN2	[60, 40]	2	0.6268	52.27%
LSTM2	[80, 80]	2	0.6280	51.72%
GRU2	[40, 20]	2	0.6283	51.16%

Notes: VAR(5) represents the VAR model with five lags. SRN, LSTM, GRU, SRN2, LSTM2, and GRU2 denote the best one-layered SRN, one-layered LSTM, one-layered GRU, two-layered SRN, two-layered LSTM, and two-layered GRU models, based on MAE. [i, j] means that the first and second layers' units are i and j, respectively.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

Table 2. MAE and MDA results for the hybrid models, based on MAE

Model	Units	Recurrent Layers	MAE	MDA
SRNLSTM	[20, 60]	2	0.6273	52.16%
SRNGRU	[80, 60]	2	0.6271	52.16%
LSTMSRN	[60, 40]	2	0.6282	51.44%
LSTMGRU	[40, 20]	2	0.6277	51.77%
GRUSRN	[80, 60]	2	0.6276	52.05%
GRULSTM	[60, 80]	2	0.6282	51.27%

Notes: SRNLSTM, SRNGRU, LSTMSRN, LSTMGRU, GRUSRN, and GRULSTM denote the best SRNLSTM, SRNGRU, LSTMSRN, LSTMGRU, GRUSRN, and GRULSTM models, based on MAE. [i, j] means that the first and second layers' units are i and j, respectively.

Table 3. MAE and MDA results of AR(1)-GARCH(1,1), VAR, and the best SRN, LSTM, and GRU models based on MDA

Model	Units	Recurrent Layers	MAE	MDA
AR(1)-GARCH(1,1)	–	–	0.9167	48.11%
VAR(5)	–	–	0.6351	47.50%
SRN	60	1	0.6269	52.05%
LSTM	80	1	0.6280	52.00%
GRU	80	1	0.6283	52.16%
SRN2	[60, 40]	2	0.6268	52.27%
LSTM2	[60, 60]	2	0.6282	51.83%
GRU2	[60, 40]	2	0.6287	52.33%

Notes: VAR(5) represents the VAR model with five lags. SRN, LSTM, GRU, SRN2, LSTM2, and GRU2 denote the best one-layered SRN, one-layered LSTM, one-layered GRU, two-layered SRN, two-layered LSTM, and two-layered GRU models, based on MDA. [i, j] means that the first and second layers' units are i and j, respectively.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

Table 4. MAE and MDA results of the best hybrid models, based on MDA

Model	Units	Recurrent Layers	MAE	MDA
SRNLSTM	[60, 80]	2	0.6276	52.22%
SRNGRU	[80, 60]	2	0.6271	52.16%
LSTMSRN	[40, 60]	2	0.6283	52.22%
LSTMGRU	[20, 40]	2	0.6277	52.05%
GRUSRN	[80, 20]	2	0.6277	52.33%
GRULSTM	[20, 80]	2	0.6287	51.66%

Notes: SRNLSTM, SRNGRU, LSTMSRN, LSTMGRU, GRUSRN, and GRULSTM denote the best SRNLSTM, SRNGRU, LSTMSRN, LSTMGRU, GRUSRN, and GRULSTM models, based on MDA. [i, j] means that the first and second layers' units are i and j, respectively.

Table 5. DM test results

	AR(1)-GARCH(1,1) vs. VAR(5)	AR(1)-GARCH(1,1) vs. SRN
Statistic	16.920	17.114
p-value	$<2.2*10^{-16}$	$<2.2*10^{-16}$
	AR(1)-GARCH(1,1) vs. SRN2	AR(1)-GARCH(1,1) vs. LSTM
Statistic	16.751	17.093
p-value	$<2.2*10^{-16}$	$<2.2*10^{-16}$
	AR(1)-GARCH(1,1) vs. LSTM2	AR(1)-GARCH(1,1) vs. GRU
Statistic	17.157	17.169
p-value	$<2.2*10^{-16}$	$<2.2*10^{-16}$
	AR(1)-GARCH(1,1) vs. GRU2	AR(1)-GARCH(1,1) vs.
Statistic	17.256	16.997
p-value	$<2.2*10^{-16}$	$<2.2*10^{-16}$
	AR(1)-GARCH(1,1) vs. SRNGRU	AR(1)-GARCH(1,1) vs.
Statistic	17.103	17.061
p-value	$<2.2*10^{-16}$	$<2.2*10^{-16}$
	AR(1)-GARCH(1,1) vs.	AR(1)-GARCH(1,1) vs. GRUSRN
Statistic	17.215	17.052
p-value	$<2.2*10^{-16}$	$<2.2*10^{-16}$
	AR(1)-GARCH(1,1) vs.	
Statistic	17.196	
p-value	$<2.2*10^{-16}$	

	VAR(5) vs. SRN	VAR(5) vs. SRN2
Statistic	4.162	3.831
p-value	$3.302*10^{-5}$	$1.316*10^{-4}$
	VAR(5) vs. LSTM	VAR(5) vs. LSTM2
Statistic	3.411	3.598
p-value	$6.593*10^{-4}$	$3.293*10^{-4}$
	VAR(5) vs. GRU	VAR(5) vs. GRU2
Statistic	3.388	3.364
p-value	$7.184*10^{-4}$	$7.826*10^{-4}$
	VAR(5) vs. SRNLSTM	VAR(5) vs. SRNGRU
Statistic	3.824	4.000
p-value	$1.359*10^{-4}$	$6.575*10^{-5}$
	VAR(5) vs. LSTMSRN	VAR(5) vs. LSTMGRU
Statistic	3.450	3.673
p-value	$5.723*10^{-4}$	$2.461*10^{-4}$
	VAR(5) vs. GRUSRN	VAR(5) vs. GRULSTM
Statistic	3.793	3.503
p-value	$1.531*10^{-4}$	$4.705*10^{-4}$
	SRN vs. SRN2	LSTM vs. LSTM2
Statistic	0.1043	–0.2251
p-value	0.9169	0.8219
	GRU vs. GRU2	SRN2 vs. SRNGRU
Statistic	–0.1662	–3.622
p-value	0.8679	0.7172
	LSTM2 vs. SRNLSTM	GRU2 vs. SRNGRU
Statistic	0.9534	1.4273
p-value	0.3405	0.1537

Notes: SRN, LSTM, GRU, SRN2, LSTM2, GRU2, SRNLSTM, and SRNGRU denote the best one-layered SRN, one-layered LSTM, one-layered GRU, two-layered SRN, two-layered LSTM, two-layered GRU, SRNLSTM, and SRNGRU models, based on MAE.

As Table 1 shows, adding the same type of layer can improve the MAE of the best one-layered SRN model by 0.0001. Meanwhile, we can improve the MDA of the best one-layered SRN model and that of the best one-layered GRU model by 0.22 and 0.17, respectively (Table 3). The results of the DM test (Table 5) indicate that at the (5%)/31 level, there are no statistically significant differences in forecast performance between the best one-layered model and the best two-layered model. This result implies that adding the same type of layer does not have a statistically significant effect in terms of improving forecast performance.

In comparing Tables 1 and 2, we find that we can lower the MAE of the best two-layered GRU and LSTM models by making appropriate hybrid models. Making the first layer of the two-layered GRU models an SRN layer, for example, can reduce the MAE by 0.0012 from the best two-layered GRU model's score. As Tables 3 and 4 report, we can increase the MDA in the best two-layered SRN and LSTM models by building appropriate hybrid models. Substituting the first layer of the two-layered LSTM models with an SRN layer, for example, can increase the MDA by 0.39 from the best two-layered LSTM model's score.

As the results of the DM test indicate (Table 5), at the (5%)/31 level, there are no statistically significant differences in forecast performance between the best two-layered models and the best hybrid models. This finding implies that combining different types of layers does not have a statistically significant effect in terms of improving forecast performance.

5. CONCLUSION

Crude oil is one of the world's most important resources, and so its price fluctuations directly affect the world economy. To mitigate the effects of spreading risks, we need to create an accurate forecasting model. In this study, we test methods by which to accurately forecast West Texas Intermediate (WTI) futures prices, a benchmark of crude oil pricing, using an autoregressive-generalized autoregressive conditional heteroscedasticity (AR-GARCH) model, a vector autoregression (VAR) model, and four main types of recurrent neural network (RNN)-based models (i.e., simple recurrent network, long short-term memory, gated recurrent unit, and hybrid models). In comparing the forecasting performance of RNN-based models to those of an AR-GARCH model and a VAR model, we find that RNN-based models outperform both with respect to mean absolute error (MAE) and mean directional accuracy (MDA). Additionally, our Diebold–Mariano test (DM test) results confirm statistically significant differences in forecast performance between RNN-based models and an AR-GARCH model and between RNN-based models and a VAR model. This finding implies that the interrelationships among WTI futures prices, the NASDAQ-100, and US soy oil futures prices may be nonlinear. We expect that in the future, the use of RNN models to forecast crude oil futures prices will become more important; our findings can help inform and thus contribute to this trend.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

ACKNOWLEDGEMENT

An early version of this paper was presented at the Workshop of Big Data and Machine Learning (Kobe University) and the SIBR 2020 Sydney Conference on Interdisciplinary Business & Economics Research. We are grateful to Prof. Zheng Zhang and conference participants for their helpful comments and suggestions. This work was supported by JSPS KAKENHI Grant Number (A) 17H00983. We thank the anonymous reviewers' for their comments.

REFERENCES

[1] Bildirici, M. (2019), "The chaotic behavior among the oil prices, expectation of investors and stock returns: TAR-TR-GARCH copula and TAR-TR-TGARCH copula", *Petroleum Science*, 16, 217–228.

[2] Bonferroni, C.E. (1935), "Il calcolo delle assicurazioni su gruppi di teste", *Studi in onore del Professore Salvatore Ortu Carboni*, Rome, Italy, 13–60.

[3] Bonferroni, C.E. (1936), Teoria statistica delle classi e calcolo delle probabilità, *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8, 3–62.

[4] Chen, Y., He, K., Tso, G.K.F. (2017), "Forecasting crude oil prices: A deep learning based model", *Procedia Computer Science*, 122, 300–307.

[5] Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y. (2014), "On the properties of neural machine translation: Encoder–decoder approaches", in arXiv:1409.1259 [cs.CL].

[6] Diebold, F.X., Mariano, R.S. (1995), "Comparing predictive accuracy", *Journal of Business & Economic Statistics*, 13(3), 253–263.

[7] Gers, F.A., Schmidhuber, J., Cummins, F. (2000), "Learning to forget: Continual prediction with LSTM", *Neural Computation*, 12(10), 2451–2471.

[8] Hochreiter, S., Schmidhuber, J. (1997), "Long short-term memory", *Neural Computation*, 9(8), 1735–1780.

[9] Moshiri, S., Foroutan, F. (2006), "Forecasting nonlinear crude oil futures prices", *The Energy Journal*, 27(4), 81–95.

[10] Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986), "Learning representations by back-propagating errors", *Nature*, 323, 533–536.

[11] Sembiring, M.F., Paramita, S.V., Malik, P.A. (2016), "The estimation model for measuring performance of stock mutual funds based on ARCH/GARCH model", *Review of Integrative Business and Economics Research*, 5(2), 215–225.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

[12] Sutskever, I., Vinyals, O., Le, Q.V., (2014), "Sequence to sequence learning with neural networks", in *Proc. Advances in Neural Information Processing System*, 27, 3104–3112.

[13] Wang, J., Wang, J. (2016), "Forecasting energy market indices with recurrent neural networks: Case study of crude oil price fluctuations", *Energy*, 102, 365–374.

[14] Wen, X., Yu, L., Xu, S., Wang, S. (2006), "A new method for crude oil price forecasting based on support vector machines", *Computational Science – ICCS 2006*, 444–451.

[15] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J. (2016), "Google's neural machine translation system: Bridging the gap between human and machine translation", in arXiv:1609.08144 [cs.CL].

[16] Yao, K., Cohn, T., Vylomova, K., Duh, K., Dyer, C. (2015), "Depth-Gated LSTM", in arXiv:1508.03790 [cs.NE].

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)