Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 2557865, 12 pages
https://doi.org/10.1155/2022/2557865

Research Article

A Novel Methodology for Credit Spread Prediction: Depth-Gated Recurrent Neural Network with Self-Attention Mechanism

Xiao Liu,[1] **Rongxi Zhou**,[2] **Daifeng Qi**,[3] **and Yahui Xiong**[4]

[1]*School of Economics and Management, North China University of Technology, Beijing 100144, China*
[2]*School of Banking and Finance, University of International Business and Economics, Beijing 100029, China*
[3]*Peking University HSBC Business School, Shenzhen 518055, Guangdong, China*
[4]*No.8 Department, 32180 Army, Beijing 100072, China*

Correspondence should be addressed to Yahui Xiong; yy11xyh@163.com

Received 27 August 2021; Revised 3 January 2022; Accepted 3 May 2022; Published 9 August 2022

Academic Editor: Yiwen Zhang

This paper develops a depth-gated recurrent neural network (DGRNN) with self-attention mechanism (SAM) based on long-short-term memory (LSTM)\gated recurrent unit (GRU) \Just Another NETwork (JANET) neural network to improve the accuracy of credit spread prediction. The empirical results of the U.S. bond market indicate that the DGRNN model is more effective than traditional machine learning methods. Besides, we discovered that the Depth-JANET model with one gated unit performs better than Depth-GRU and Depth-LSTM models with more gated units. Furthermore, comparative analyses reveal that SAM significantly improves DGRNN's prediction performance. The results show that Depth-JANET neural network with SAM outperforms most other methods in credit spread prediction.

1. Introduction

Credit spread is the risk premium demanded by investors of credit bonds over the yield of risk-free bonds of the same maturity, which is the basis of credit bond pricing and risk management. By grasping the future trend of credit spread, stakeholders involved in the bond market can make decisions more scientifically. For instance, investors can improve the accuracy of transactions. Financiers can choose time scientifically, and regulators can properly prevent and control financial risks. Besides, credit spread can be utilized to monitor macroeconomics and warn governments. However, because the bond market is usually regarded as a complex system [1], existing techniques cannot perform well in predicting the credit spread accurately. Therefore, it is necessary to theoretically and empirically discuss how to improve the credit spread's prediction accuracy.

As a representative technology of artificial intelligence, deep learning methods have developed rapidly in recent years [2–4]. Deep neural networks have become the most advanced forecasting method in finance due to its outstanding performance in time-series prediction [5, 6]. They have been widely used to predict indicators, such as stock prices, exchange rates, gold prices, and housing prices [5, 7–9].

Many studies show that deep neural networks can effectively fit complex nonlinear relationships between input variables with a higher fitting degree, reducing the overfitting of shallow foundations and local extremum problems. Besides, deep neural networks have no restrictions on the form of input variables. Therefore, all relevant information can be included. Particularly, deep neural networks can perform generalized learning based on data characteristics, weakening irrelevant information while learning heterogeneous information.

Existing literature on the prediction of the credit spread is mainly based on linear models [6, 10]. Although the deep learning methods can help improve the accuracy of credit spread prediction, it is not clear which algorithm has the best prediction performance according to the "No Free Lunch Theorem" proposed by Wolpert. Thus, it is also worthy of in-depth investigation of the performance of deep learning algorithms in credit spread prediction [11].

This paper aims to construct a depth-gated recurrent neural network with self-attention mechanism (SAM-DGRNN) to predict credit spreads in the U.S. corporate bond market. The main contributions are as follows: (1) to apply the XGBoost algorithm to integrate the selected credit spread determinants and extract the feature variables with the highest importance of prediction. (2) To construct depth-gated recurrent neural networks based on LSTM/GRU/JANET and compare them with three traditional nonlinear machine learning models (e.g., support vector regression, multilayer perception, and the random forest approaches) and a linear model (e.g., the vector autoregressive model). The comparative analysis of their prediction effects supports the superiority of deep learning methods in predicting credit spreads. (3) To construct a depth-gated recurrent neural network with self-attention mechanism (SAM-DGRNN) to explore the effectiveness of SAM in credit spread prediction.

The remainder of this paper is organized as follows. Section 2 provides a brief background. Section 3 discusses literature review related to the deep learning field. Section 4 introduces the theoretical methods and methodology for constructing the model of predicting the credit spread. Section 5 presents the experimental results. Section 6 is the conclusion section of this paper.

2. Background

2.1. Long-Short-Term Memory (LSTM) Neural Network. LSTM neural network has three gated units: input gate, forget gate, and output gate. The gated units allow information to affect recurrent neural networks at each moment selectively. Each gate outputs a value between 0 and 1. The value refers to how much information can be passed (0 means "no information can pass and one means "all information is allowed to pass"). The forget gate controls what information is discarded or saved from the cell state, the input gate controls how much new information is added to the cell state, and the output gate controls which part of the cell state will be output. The schematic diagram of LSTM neural network structure is shown in Figure 1. The update rules are shown in equations (1) to (6).

First, the forget gate discards useless historical information:

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

Second, the input gate updates the state with input data and historical information:

$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + b_i\right), \tag{2}$$

$$\tilde{c}_t = \tanh\left(W_{\tilde{c}x}x_t + W_{\tilde{c}h}h_{t-1} + b_{\tilde{c}}\right), \tag{3}$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t, \tag{4}$$

Third, the output gate outputs current information:

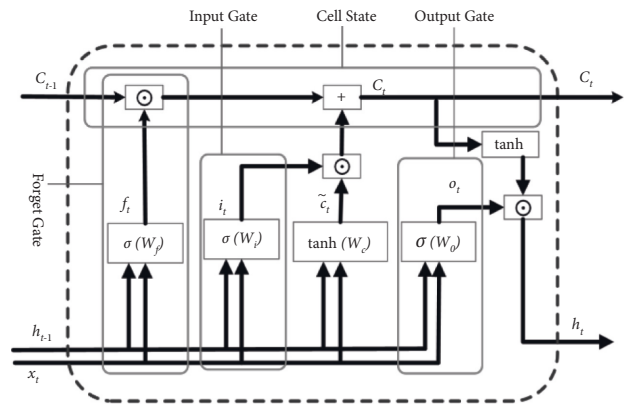

Figure 1: Structure of LSTM neural network.

$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right), \tag{5}$$

$$h_t = o_t \times \tanh\left(c_t\right), \tag{6}$$

where x_t is the input vector at time t, h_{t-1} is the output vector at time $t-1$, $W_{fx}, W_{fh}, W_{ix}, W_{ih}, W_{\tilde{c}x}, W_{\tilde{c}h}, W_{ox}, W_{oh}$ are the weight matrixes, $b_f, b_i, b_{\tilde{c}}, b_o$ are the bias vectors, σ is the logistic sigmoid function with the form of $\sigma(x) = (1+e^{-x})^{-1}$, tanh is the hyperbolic tangent activation function, f_t, i_t, o_t are the states of forget gate, input gate, and output gate at time t, respectively, and c_t is the state of the memory unit at time t.

2.2. Gated Recurrent Unit (GRU) Neural Network. GRU neural network consists of two gated units. The update gate is used to control the degree to which previous state information is brought into the current state. The smaller its value is, the less information it brings and the smaller the impaction the current hidden layer is. The reset gate is used to control the degree of state information that is ignored at the previous moment. The larger its value is, the less information is overlooked. GRU neural network synthesizes the input gate and the forget gate in LSTM neural network into a single update gate and combines the cell state and the hidden state. These features not only maintain the advantages of LSTM in solving long-term dependency problems but also lead to a more straightforward structure, with fewer parameters and higher training efficiency. The schematic diagram of GRU neural network structure is shown in Figure 2. Update rules are shown in equations (7) to (10).

First, the reset gate determines the degree of the alternative state $\tilde{h}_t$ depending on the previous state h_{t-1}:

$$r_t = \sigma\left(W_{rx}x_t + W_{rh}h_{t-1} + b_r\right), \tag{7}$$

$$\tilde{h}_t = \tanh\left(W_{\tilde{h}x}x_t + W_{\tilde{h}h}r_t h_{t-1} + b_{\tilde{h}}\right). \tag{8}$$

Second, the update gate determines the weights of historical information inheriting from the previous state h_{t-1} and new information the current alternative state accepts:

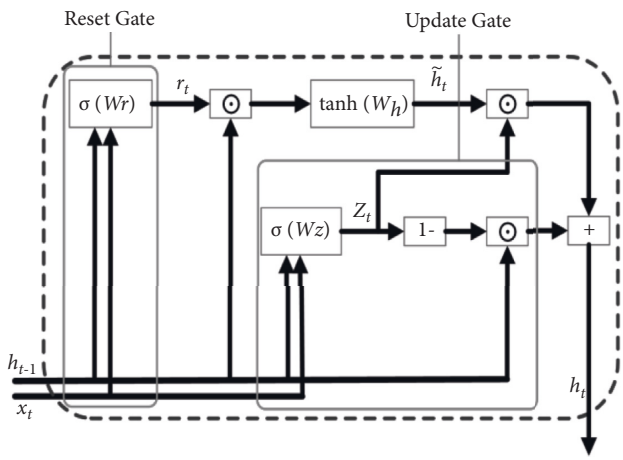

Figure 2: Structure of GRU neural network.

$$z_t = \sigma\left(W_{zx}x_t + W_{zh}h_{t-1} + b_z\right), \tag{9}$$

$$h_t = \left(1 - z_t\right) \times h_{t-1} + z_t \times \tilde{h}_t, \tag{10}$$

where x_t is the input vector at time t, h_{t-1} is the output vector at time $t-1$, $b_r, b_{\tilde{h}}, b_z$ are the bias vectors, $W_{rx}, W_{rh}, W_{\tilde{h}x}, W_{\tilde{h}h}, W_{zx}, W_{zh}$ are the weight matrixes, σ is the logistic sigmoid function, tanh is the hyperbolic tangent activation function, and r_t, z_t are the output state of the reset gate and update gate at time t, respectively.

2.3. Just Another NETwork (JANET) Neural Network. JANET neural network, with dramatically less training time, performs better on multiple benchmark data sets than LSTM neural network. The schematic diagram of JANET neural network structure is shown in Figure 3. The update rules are shown in equations (11) to (14).

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right), \tag{11}$$

$$\tilde{c}_t = \tanh\left(W_{\tilde{c}x}x_t + W_{\tilde{c}h}h_{t-1} + b_{\tilde{c}}\right), \tag{12}$$

$$c_t = f_t \times c_{t-1} + \left(1 - f_t\right)\tilde{c}_t, \tag{13}$$

$$h_t = c_t, \tag{14}$$

where x_t is the input vector at time t, h_{t-1} is the output vector at time $t-1$, $W_{fx}, W_{fh}, W_{\tilde{c}x}, W_{\tilde{c}h}$ are the weight matrixes, $b_f, b_{\tilde{c}}$ are the bias vectors, σ is the logistic sigmoid function, tanh is the hyperbolic tangent activation function, f_t is the output state of the forget gate at time t, and c_t is the state of memory unit at time t.

2.4. Self-Attention Mechanism (SAM). SAM is an improvement in the basic attention mechanism. It uses an attention mechanism to dynamically generate the weights of different connections between the input and output of the same layer (not the model's final output) to obtain this layer's output. SAM considers the logical connection between the upper and lower sequences. Assume that the input

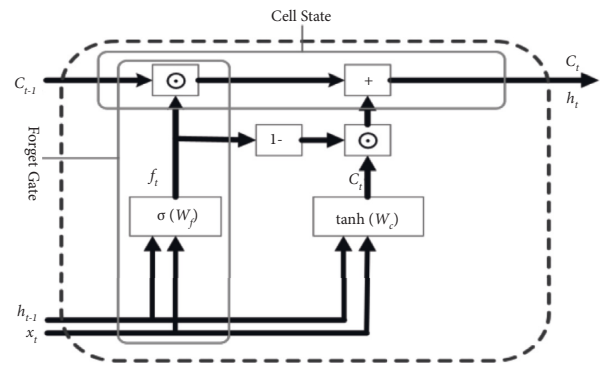

Figure 3: Structure of JANET neural network.

sequence of a layer is $X = [x_1, x_2, \ldots, x_N]$, and the output sequence is $H = [h_1, h_2, \ldots, h_N]$ with the same length. First, deduce three vector sequences through a linear transformation:

$$\begin{aligned} Q &= W_Q X \in R^{d_3 \times N}, \\ K &= W_K X \in R^{d_3 \times N}, \\ V &= W_V X \in R^{d_3 \times N}, \end{aligned} \tag{15}$$

where Q, K, V are the query vector sequence, key vector sequence, and value vector sequence, respectively, and W_Q, W_K, W_V are the learning parameters.

The output vector is

$$\begin{aligned} h_i &= att\left((K, V), q_i\right) = \sum_{j=1}^{N} a_{ij} v_j = \sum_{j=1}^{N} \operatorname{soft\,max}\left(s\left(k_j, q_i\right)\right) v_j \\ &= \sum_{j=1}^{N} \frac{\exp\left(s\left(k_j, q_i\right)\right)}{\sum_i \exp\left(s\left(k_i, q_i\right)\right)} v_j. \end{aligned} \tag{16}$$

where $(K, V) = [(k_1, y_1), (k_2, y_2), \ldots (k_N, y_N)]$ is the key-value pair, representing the input information. $i, j \in [1, N]$ are the positions of input and output vector sequences, and the connection weight α_{ij} is dynamically generated by the attention mechanism. softmax ensures that the sum of all weights is 1.

3. Related Work

Deep learning methods have been widely used in many fields. As one of the classic deep learning models, long-short term memory (LSTM) neural network has great advantage in mining long-term dependencies of sequence data. It was first proposed by Hochreiter and Schmidhuber to solve long-term memory problems in recurrent neural network by considering the "gated units" [12]. Wang et al. applied LSTM to speech enhancement and proposed a LSTM convolution network, which includes transpose convolution and jump connection [13]. Ma et al. introduced convolution operation into traditional LSTM and proposed a CLSTM learning algorithm to extract time-frequency information and obtained features through convolution [14]. Petmezas et al.

combined LSTM with convolutional neural networks (CNN) and proposed the CNN-LSTM model. Through CNN, signal features are transmitted to LSTM to realize dynamic memory [15]. Yu et al. applied LSTM to a nonlinear system model and proposed an improved depth LSTM. Combining the strengths of LSTM and multilayer perception, the stability of the training method is verified by the Lyapunov function. At the same time, the model is preferable to other existing models in a nonlinear system [16].

Due to many parameters involved, the LSTM neural network performs a lower training efficiency. To improve this drawback, Cho et al. proposed a more simplified gated recurrent unit (GRU) neural network based on LSTM neural network and proved that the prediction performance of the GRU Neural network is better than that of standard LSTM neural network [17]. Particularly, GRU can significantly simplify the structure of LSTM, reduce the number of parameters, and greatly shortens the training time. Liu et al. used GRU to replace the LSTM in the neural programmer interpreter for changing the core structure [18]. Based on the classification results of LSTM and full convolution network LSTM-FCN, Nelsayed et al. found that GRU has higher classification accuracy and simpler hardware implementation in time-series classification problems, which are of smaller architecture and less computation [19]. Wu et al. combined GRU with CNN to propose a GRU-GNN hybrid neural network model. In the GRU-GNN model, GRU is responsible for extracting the feature vector of time-series data, and CNN extracts the feature vector of high-dimensional data [20]. Pan et al. applied the GRU-GNN combined model to the water level prediction of the Yangtze River. Through the 30-year water level data of the Yangtze River and comparative analysis, it is confirmed that the model is superior to wavelet neural network, LSTM, and statistically integrated moving average autoregressive model ARIMA [21]. Given the excellent performance of the GRU neural network after eliminating redundant gates, Westhuizen and Lasenby further explored the necessity of three gated units in the LSTM neural network to build more efficient models [22]. They proposed a JANET (Just Another NETwork) with only a forget gate and chronologically initialized bias terms.

Attention mechanisms are widely used in neuroscience and computational neuroscience. This common mechanism comes from the fact that many animals only focus on specific parts of their vision to give enough response. Therefore, many neural computing studies have concluded that people only need the most relevant information, rather than all information, for further neural processing. In recent years, this mechanism has also been widely used in deep learning research, such as image re-rolling and voice recognition. Recent studies have found that considering the self-attention mechanism in deep learning can effectively extract the most critical information for current tasks to enhance predictive power. Attention mechanism has become one of the most important topics in the deep learning literature following the research by Vaswani et al. [23]. Zhao et al. designed a long-short term memory (LSTM) neural network structure model with attention mechanism based on the dynamic sequence in the internet financial market [24]. The empirical results showed that their attention mechanism model outperformed others. Chen and Ge applied an LSTM neural network with attention mechanism to predict the stock price trend in Hong Kong and achieved satisfactory prediction results [25].

In the training of deep neural network models, the gradient vanishing and overfitting often result in unsatisfactory learning effects. Studies have shown that batch-normalization (B. N.) method can alleviate the gradient vanishing by pulling the data back to a standard normal distribution with a mean of 0 and a variance of 1 [26]. Furthermore, Dropout can prevent overfitting to a certain extent by preventing neuronal coadaptation during the training phase [27]. However, improper use of both methods will generate the opposite effect. Li et al. found that placing Dropout in all B. N. layers or modifying Dropout's formula to reduce the sensitivity of variance could improve the coordination between B. N. and Dropout [28]. Luo et al. suggested that by adopting differentiable learning, the switchable-normalization method (S. N.) could determine the appropriate normalization operation for each normalization layer in a deep network [29]. As a result, it is more advantageous than B. N. in avoiding gradient disappearance. Therefore, in our deep neural network design, we add the S. N. layer and Gaussian Dropout layer to optimize its structure. A reasonable combination of the S. N. layer and Dropout layer will improve the performance of the neural networks.

4. Methodology

4.1. Depth-Gated Recurrent Neural Network with Self-Attention Mechanism (SAM-DGRNN). We add the S. N. layer and Gaussian dropout layer to optimize its structure in the deep neural network. A reasonable combination of the S. N. layer and dropout layer will improve the performance of the neural networks. Specifically, the main structure of the depth-gated recurrent neural network constructed in this paper includes an attention mechanism layer, a three-layer LSTM/GRU/JANET neural layer, and two fully connected layers (of which the first neural layer has 128 neurons, the second has 64 neurons, the third has 32 neurons, and the two fully connected layers have 32 neurons and one neuron, respectively). An S.N. layer is added in front of each LSTM/GRU/JANET neural layer. A Gaussian Dropout layer is added at the back of the LSTM/GRU/JANET neural layer, and the drop rate is set to 0.2. The structure of the deep LSTM/GRU/JANET neural network is shown in Figure 4, and the neural network structure is shown in the dotted box.

4.2. Training Method, Loss Function, and Optimizer Selection. We apply the mini-batch gradient descent method to train the deep learning neural network. In order to predict future credit spreads, the mean square error (MSE) is selected in the loss function. We choose Adam optimizer (adaptive moment estimation) to perform optimization training. Compared with other self-adaptive learning rate algorithms, the Adam algorithm is more robust in selecting hyperparameters, with higher training efficiency, and can generate more effective results [30]. The experimental environment of this paper is shown in Table 1.

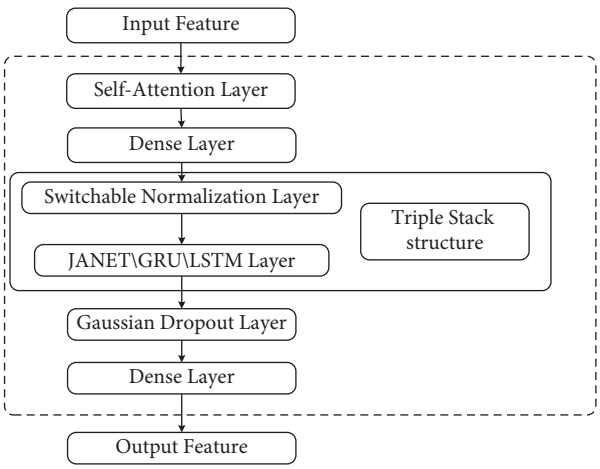

Figure 4: Depth-gated recurrent neural network with self-attentional mechanism.

Table 1: Experiment environment.

Operating system	Windows 7
Processor	Intel(R) core (TM) i7-2760QM
Random-access memory	8.00 GB
Version of python	Python 3.7.0
Version of spyder	Spyder 3.3.1
Version of keras	Keras 2.2.4
Version of tensorflow	Tensorflow 1.13.1
Optimizer	Adam optimizer

4.3. Control Models. In the depth-gated recurrent neural network model, we employ the rolling-window prediction method and use the data of n trading days to predict the credit spread on the next day. This paper first integrates indicators with XGBoost (extreme gradient boosting) algorithm, extracting the predictor variables with higher importance ranking. XGBoost algorithm combines random forest algorithm, which can further reduce calculation complexity. When dealing with a large amount of data, XGBoost can operate parallel and divide the data according to different characteristics to form a tree sequence. This algorithm is simpler and more effective. It can transfer complex data to an orderly and concise arrangement form. Then the feature variables with a higher importance ranking are selected as the model input.

To comprehensively evaluate the prediction effect of depth-gated recurrent neural network, one benchmark deep learning model RNN and three traditional machine learning models (support vector machines (SVR), multilayer perceptron (MLP), and random forest (RF)) in financial prediction are selected as nonlinear control models. Research has put VAR as a linear control model. [6]. Deep RNN selection and depth-gated recurrent neural network have the same structure. The parameter combination in SVR is set as "Radial Basis Function (RBF), penalty parameter $C = 1$, gamma = auto". We also select the classic MLP neural network with three layers. The prediction methods of RNN, SVR, MLP, and R. F. are consistent with depth-gated recurrent neural network. The prediction idea of a VAR model is as follows: first, the stationarity of all sequences is comprehensively judged by the ADF test, KPSS test, and P. P. test and decide whether to carry out the corresponding order difference to obtain the stationary sequence according to the test results. Second, the VAR model is established. The order of the VAR model is determined by integrating AIC and BIC information criteria. Third, the VAR model was estimated, and the model's stability was tested. Finally, the credit spread sequence is predicted based on the stable VAR model. The prediction flowchart is shown in Figure 5.

5. Empirical Prediction Analysis

5.1. Variable Selection. We collect daily closing data from 2009 to 2019. The 2517 trading days during this period are divided into a training set (includes the first 85% of trading days) and a test set (includes the remaining 15% of trading days).

Table 2 shows variables in the literature that have been verified as significant credit spread determinants. The credit spread sequence is a forecast indicator, and additional variables are used as the characteristics to predict credit spreads. The detailed indicators are discussed as follows.

Risk-free interest rate term structure: the risk-free interest rate is an important variable in the structural model. The information contained in the shape of the riskless yield curve can improve the prediction performance of credit spreads [6].

Credit spread term structure: the credit spread curve's level, slope, and curvature are the principal variables for predicting the future credit spread [6].

Fama-French factor returns: credit spreads indicate the extra compensation of holding risky assets as an analogy to stock risk premiums. Therefore, financial markets would transfer the explanatory power of stock returns, represented by Fama-French factor returns, to the bond market [31].

Return on Stock Index: stocks are also yield-producing securities. The equity market is the most plausible alternative to the fixed-income market, and equity market indexes measure capital market investment levels. Therefore, the return on the stock index could be relevant to corporate bond credit spreads [32].

Volatility of Stock Index: VIX Index, often referred to as the market's "fear gauge," can be correlated with credit spreads, which capture the future probability of default as a common forward-looking risk metric. As a result, stock market volatility is a significant variable for explaining credit spread changes [33].

Exchange rate: the prevailing economic theory, such as uncovered interest rate parity, suggests that there should be an empirical relationship between exchange rates and interest rates. Given exchange rate fluctuations, foreign investors will be attracted to invest in U.S. corporate bonds. The foreign exchange rate is a heretofore overlooked variable for explaining credit spread changes [33].

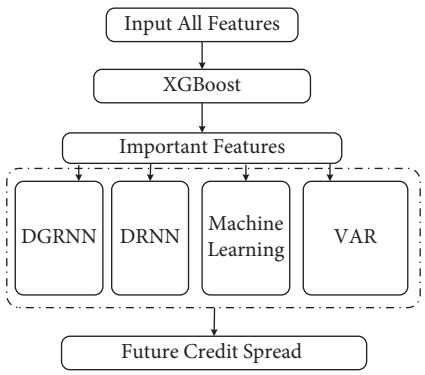

Figure 5: Prediction flowchart.

Table 2: Variables and data.

Variables			Data
Credit spread			ICE BofAML US corporate master option-adjusted spread[$]
Financial market factors	Credit spread term structure	Level factor	The first principle component extracted From a large cross section of credit spreads of various maturities and various ratings[$]
		Slop factor	The second principal component extracted From a large cross section of credit spreads of various maturities[$]
		Curve factor	The third principle component extracted From a large cross section of credit spreads of various maturities[$]
	Risk-free interest rate term structure	Level factor	The long-term factor of Nelson and Siegel term structure decomposition extracted from a large cross section of treasury yields of various maturities[$$]
		Slop factor	The short-term factor of Nelson and Siegel term structure decomposition extracted from a large cross section of treasury yields of various maturities[$$]
		Curve factor	The mid-term factor of Nelson and Siegel term structure decomposition extracted from a large cross section of treasury yields of various maturities[$$]
	Fama-French factor returns		Excess return on the market
			Small-minus-big return
			High-minus-low return
	Return on stock index		Return on S&P 500
	Volatility of stock index		Volatility of S&P 500
	Exchange rate		U.S. Dollar index
	Oil prices		Crude oil prices: West Texas intermediate (WTI)
	TED spread		Difference between 3-month LIBOR based on U.S. dollars and 3-month treasury bill
	Swap spread		Difference between10-year swap rate and 10-year treasury yield
	Commodity price index		RJ/CRB index

Note: Data are obtained from https://fred.stlouisfed.org; **$** ICE BofAML US Corporate 1–3\3–5\5–7\7–10\10–15\15+ \AAA\AA\A\BBB\BB\B\CCC Option-Adjusted Spread;**$$** 3\6-month and 1\2\3\5\10-year treasury yields.

Oil Prices: energy prices, as the cost of economic activities, are captured by oil prices to study their influence on credit spreads.

TED spread: TED spread captures additional macroeconomic and interest rate information from international fixed-income markets. LIBOR and U.S. treasury yields are often used to price complex financial derivative products, and the difference is an important predictive variable [34].

Swap spread: swap spread is highly correlated with credit spreads because it is a proxy for credit rate. Since the swap market is more well developed and liquid than corporate bonds, swap rates may provide a forward indication of credit spreads. Credit spreads will increase with swap spreads [35].

Commodity Price Index: it is widely used to analyze price fluctuations in commodity markets and macroeconomy. CPI index is a better indicator of inflation [6].

The importance score based on the XGBoost algorithm is shown in Figure 6. The y-axis represents the feature, x-axis represents the importance score, and the score is between 0 and 1.

Figure 6 shows the mutual information of selected features. To avoid disturbance from insignificant features, with 0.01 as the cut-off point of importance score, we select ten features with the highest mutual information from the

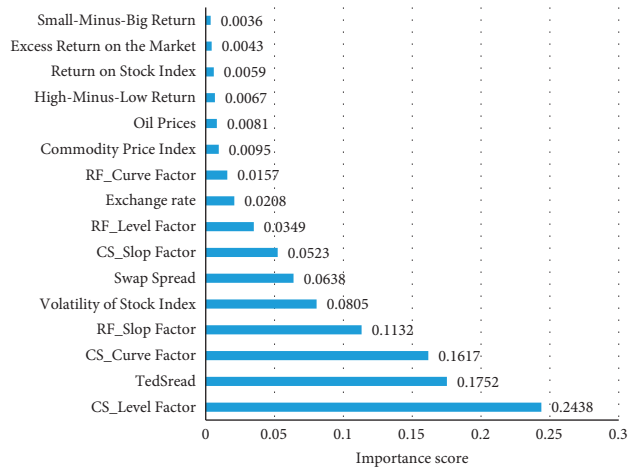

Figure 6: Mutual information for selected features. (There is no doubt that the term structure of credit spread is the most important factor to predict credit spread. In order to visualize the importance of the other factors in the diagram, we only examine the importance of the remaining factors here.)

raw feature set. Features with higher mutual information are more helpful to determine future spreads.

5.2. Model Evaluation and Result Comparison

5.2.1. Evaluation Indexes for Prediction Results. In this paper, we apply three indicators, including MAE (mean absolute error), MAPE (mean absolute percentage error) and RSR (The classification of RSR values by Moriasi et al. (2007): when $\mathrm{RSR} \leq 0.5$, the prediction performance is excellent; when $0.5 \leq \mathrm{RSR} \leq 0.6$, the prediction performance is good; when $0.6 \leq \mathrm{RSR} \leq 0.7$, the prediction performance is at an average level; when $\mathrm{RSR} > 0.70$, the prediction performance is poor) (root mean square error (RMSE) divided by standard deviation), to evaluate prediction accuracy. The smaller the value is, the higher accuracy the prediction will have. SDAPE (standard deviation of mean absolute percentage error) is used to evaluate the prediction stability. The smaller the SDAPE value is, the better the prediction stability will be. The calculation formulas of evaluation indicators are shown in formulas (17)–(21):

$$\mathrm{MAE} = \frac{1}{N} \times \sum_{i=1}^{N} (y_i - \widehat{y}_i), \tag{17}$$

$$\mathrm{MAPE} = \frac{1}{N} \times \sum_{i=1}^{N} \left| \frac{y_i - \widehat{y}_i}{y_i} \right| \times 100\%, \tag{18}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2}, \tag{19}$$

$$\mathrm{RSR} = \frac{\mathrm{RMSE}}{\mathrm{STD}}, \tag{20}$$

$$\mathrm{SDAPE} = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} \left(\left| \frac{y_i - \widehat{y}_i}{y_i} \right| - \mathrm{MAPE} \right)^2} \times 100\%, \tag{21}$$

where y_i and $\widehat{y}_i$ represent the actual value and the predicted value of credit spreads, respectively, STD is the standard derivation of the actual value, and N is the sample size.

Although the indicators are widely used to compare prediction accuracy, their values alone are insufficient to determine models' prediction ability. We also conduct D. M. statistical tests with these indicators as the basic loss function to analyze statistical significance [36]. The idea of the D. M. statistical test is as follows. For a set of actual time series$\{y_t\}_{t=1}^{T}$, the estimated values for the two models are $\{\widehat{y}_{it}\}_{t=1}^{T}$ and $\{\widehat{y}_{jt}\}_{t=1}^{T}$, whose error sequences are $\{e_{it}\}_{t=1}^{T}$ and $\{e_{jt}\}_{t=1}^{T}$, and whose loss functions are $g(y_t, \widehat{y}_{it}) \equiv g(e_{it})$ and $g(y_t, \widehat{y}_{jt}) \equiv g(e_{jt})$, respectively. As a result, their relative loss functions can be expressed as $d_t = g(e_{it}) - g(e_{jt})$. The null hypothesis is that the two models' prediction abilities are not different, expressed as $E(d_t) = 0$. If the loss-differential series $\{d_t\}_{t=1}^{T}$ is covariance stationary and short memory, then standard results may be used to deduce the asymptotic distribution of the sample mean loss differential.

We have $\sqrt{T}(d_{\mathrm{mean}} - \mu) \longrightarrow^{d} N(0, 2\pi f_d(0))$, where $d_{\mathrm{mean}} = 1/T \sum_{t=1}^{T} d_t$ is the sample mean loss differential. $f_d(0) = 1/2\pi \sum_{\tau=-\infty}^{\infty} \gamma_d(\tau)$ is the spectral density of the loss differential at frequency 0, and $\gamma_d(\tau) = E[(d_t - \mu)(d_{t-\tau} - \mu)]$ is the autocovariance of the loss differential at displacement γ, and τ is the population mean loss differential. The formula of $f_d(0)$ shows that the correction for serial correlation can be substantial, even if the loss differential is only weakly serially correlated, due to the accumulation of the autocovariance terms.

Because in large samples the sample means loss differential d_{mean} is approximately normally distributed with mean μ and variance $2\pi f_d(0)/T$, the obvious large-sample $N(0,1)$ statistic for testing the null hypothesis of equal forecast accuracy is $\mathrm{DM} = d_{\mathrm{mean}}/(2\pi \widehat{f}_d(0)/T)$, where $\widehat{f}_d(0)$ is a consistent estimate of $f_d(0)$. If the absolute value of D. M. statistic is significantly greater than the critical value, the null hypothesis is rejected, indicating that the two models' predictive abilities are significantly different.

5.2.2. Prediction Performance of Depth-Gated Recurrent Neural Network (DGRNN). We have the following discoveries about repeated experiments. (a) The deep learning model is sensitive to the number of traversals (the value of hyperparameter epochs); the prediction effectiveness of the same model is in a U-shaped relationship with epochs value; when epochs = 100 ± 10, the deep learning models perform best in the experiments; the machine learning models are not sensitive to hyperparameter epochs. In this paper, the number of traversals is 100 (epochs = 100). (b) Several representative values (1, 5, 20, 60, 120, 180, and 250) were selected to test the sensitivity of the hyperparameter look_back, which determines the number of trading days

Table 3: Statistical performance indicators of the prediction models.

	DGRNN				Machine learning			VAR
	DJANET	DGRU	DLSTM	DRNN	SVR	MLP	RF	
MAE	**5.6770**	7.7149	8.4230	11.9192	12.0704	10.1098	9.8887	12.7386
DM	—	1.8761*	1.9799**	2.5902***	2.7406***	2.4763**	2.1659**	3.1721***
MAPE	**0.0479**	0.0635	0.0728	0.1046	0.2271	0.0863	0.0763	0.3826
D.M.	—	1.6973*	1.9477*	3.7901***	5.6809***	2.7217***	2.0705**	8.3050***
RSR	**0.4198**	0.5549	0.5995	0.8277	1.6920	0.7263	0.7076	1.8340
SDAPE	**0.0316**	0.0437	0.0442	0.0570	0.1154	0.0526	0.0471	0.2008

Note:*, **, and *** indicate that D.M. statistics are significant at the 10%, 5%, and 1% levels, respectively; DJANET is the benchmark in the D.M. test; boldface represents the optimal value under different evaluation criteria; the following tables are the same.

Table 4: Results of the prediction models (loss function = "MAE")

	True value	DJANET	DGRU	DLSTM	DRNN	SVR	MLP	RF	VAR
9/2017	112.89	114.51#	109.59	118.56	115.36	108.56	116.97	117.694	121.44
10/2017	107.45	106.21	107.89#	113.95	112.15	106.05	113.28	112.76	115.04
11/2017	104.42	105.73	105.31#	110.77	108.05	101.11	106.83	109.22	112.30
12/2017	100.40	100.97#	102.42	107.45	105.83	97.19	105.64	105.71	111.58
1/2018	94.71	94.03	98.69	101.68	100.32	94.72#	101.68	100.51	106.28
2/2018	96.52	95.70#	103.21	106.88	103.22	99.78	98.77	111.14	103.21
3/2018	109.66	104.35	111.59	116.20	111.28	110.32#	110.56	112.88	114.01
4/2018	112.47	112.63#	118.29	123.56	116.87	117.08	115.34	124.02	116.36
5/2018	115.95	116.33#	110.80	113.83	110.56	113.03	117.52	111.91	122.35
6/2018	123.95	122.61	117.84	121.34	118.13	117.28	124.91#	116.67	131.73
7/2018	122.95	124.95#	117.51	119.97	117.96	117.32	127.17	117.46	131.63
8/2018	117.78	116.48#	113.22	115.88	115.23	112.03	122.08	115.09	127.66
9/2018	115.94	117.47	112.53	113.81	113.92	109.91	121.54	116.97#	125.37
10/2018	117.09	115.53	113.99	113.60	113.21	110.01	116.28#	123.65	122.99
11/2018	132.20	130.91#	125.84	130.23	125.76	124.81	130.04	130.63	135.14
12/2018	151.47	150.50#	149.44	158.19	146.64	150.15	149.32	152.88	152.51
1/2019	150.14	151.70	150.50#	159.20	148.31	148.73	147.88	149.66	154.39

Table 5: Results of the different lengths of time steps (loss function = "SDAPE")

	DGRNN			
	DJANET	DGRU	DLSTM	DRNN
5	0.0467	0.0593	0.0949	0.1176
20	0.0738	0.1313	0.1321	0.1400
60	0.0783	0.1005	0.1019	0.1105
120	0.0856	0.0836	0.0977	0.1155
180	0.0719	0.0894	0.0925	0.0976

used to predict the credit spreads of the next day. We find that the prediction effect of the same model and the value of look_back show W-type characteristics. All models perform best when look_back = 5, indicating that the historical data of the previous five trading days already contain enough information. Too few trading days result in insufficient information, while too many bring extra noise. Therefore, we set the look_back parameter to 5 in the subsequent analysis. The prediction results with the parameter combination [epochs, look_back] as [100, 5] are shown in Tables 3 and 4.

As can be seen from Table 3, all D. M. test results reject the null hypothesis, indicating that the predictive power of these models is significantly different. Depth-gated recurrent neural networks (DJANET/DGRU/DLSTM) are superior to nongated deep recurrent neural networks (DENN), such as the traditional machine learning models (SVR/MLP/RF) and linear prediction model (VAR), in credit spread prediction of U.S. bond market in terms of accuracy and stability. Furthermore, the RSR value of the DGRNN model is less than 0.6, suggesting that the DGRNN model also performs better in absolute dimensions according to the classification standard of RSR metric value by Moriasi et al. [37]. Besides, D.M. test results also show that the null hypothesis is rejected at least at the significance level of 10%, indicating that the predictive power of the DJANET model is significantly different from that of the other models. In the deep learning model with the gated unit mechanism, the DJANET model with one gated unit performs better than the DGRU and DLSTM models, which have more gated units. Furthermore, the DGRU model with two gated units is better than the DLSTM model with three gated units.

Table 6: Impact of SAM on prediction results

	SAM-DJANET	DJANET	SAM-DGRU	DGRU	SAM-DLSTM	DLSTM
MAE	**2.8122**	5.6770	3.2107	7.7149	5.8179	8.4230
DM	2.7254***		3.2561***		2.2883**	
MAPE	**0.0215**	0.479	0.0275	0.0635	0.0502	0.0728
DM	3.1083***		4.1697***		2.1168**	
RSR	**0.3009**	0.4198	0.3733	0.5549	0.4311	0.5995
SDAPE	**0.0219**	0.0316	0.0260	0.0437	0.0322	0.0442
Time (s)	389.65	306.24	494.68	391.04	574.88	471.60

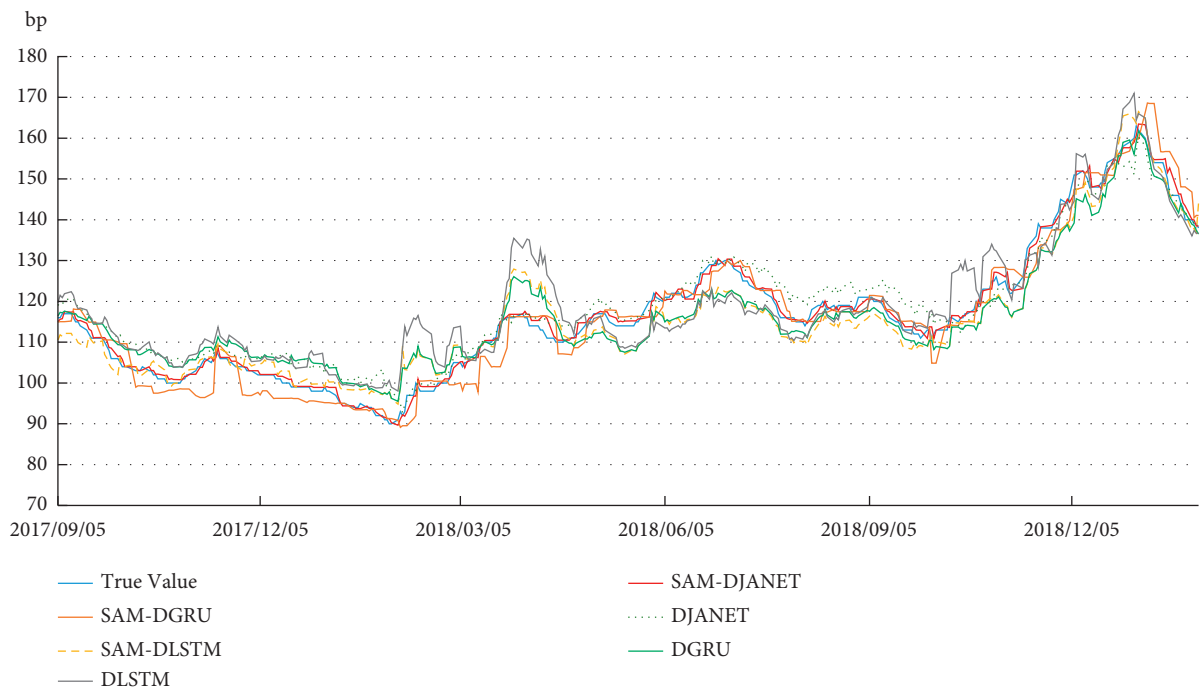

Figure 7: Comparison of DGRNN prediction curves on the prediction set.

As shown in Table 4, in 366 trading days (17 months), there are nine best-predicted values (the value with "#" in the table, the deviation of them from their actual values is the smallest) in the DJANET model and 12 best-predicted values in DGRNN model, accounting for 70.59% of the total values, showing that DGRNN model has a better prediction effect.

We compared the performance of models DJANET, DGRU, DLSTM, and DRNN at different lengths of time steps in Table 5, where SDAPE (standard deviation of mean absolute percentage error) is used to evaluate the prediction stability. The smaller the SDAPE value is, the better the prediction stability will be. As we can see, the values of DJANET are larger when the lengths of time steps are from 5 to 120 in Table 5. However, the value of DJANET is lower when the length of time steps is 180. DJANET has only a forget gate and chronologically initializes bias terms. The short-term prediction is effective and relatively loses the long-term information content. Credit spreads are cyclical, so the results will be similar to those in the short term when the ultralong term is 180 days. At the same time, there are different performances on DGRU and DLSTM. For example, the values of DGRU are lower when the lengths of time steps are from 20 to 120, and the values of DLSTM are lower when the lengths of time steps are from 20 to 180. DGRU is a variant of DLSTM, and their performance is equal in many tasks. DLSTM can learn the characteristics of long-term trend series, so its prediction efficiency is improved with the increase in time length. We discover that no matter how much the length of time steps is, the Depth-JANET model with one gated unit performs better than Depth-GRU and Depth-LSTM models, which have more gated units. However, as the length of time steps increases, the prediction accuracy will decrease.

5.3. Effectiveness of SAM. Table 6 reports the predicted performance of the depth-gated recurrent neural network with SAM. It can be seen from Table 6 that the four evaluation indicators of the SAM-DGRNN model are all smaller than the DGRNN model, and the D. M. test results also show that the null hypothesis is rejected at least at the significance level of 5%, indicating that depth-gated recurrent neural network with SAM performs better than the models without the mechanism. It also suggests that SAM can improve the

Table 7: Prediction performance of DGRNN models (train_ratio = 0.75).

	SAM-DJANET	DJANET	SAM-DGRU	DGRU	SAM-DLSTM	DLSTM
MAE	**3.031**	6.202	3.924	7.747	7.056	10.560
DM	2.977***		2.635***		2.472**	
MAPE	**0.026**	0.058	0.032	0.067	0.063	0.086
DM	3.063***		2.894***		2.117**	
RSR	**0.311**	0.452	0.380	0.602	0.477	0.790
SDAPE	**0.022**	0.032	0.025	0.046	0.034	0.053

Table 8: Prediction performance of DGRNN models (sample = [2015, 2018]).

	SAM-DJANET	DJANET	SAM-DGRU	DGRU	SAM-DLSTM	DLSTM
MAE	**2.949**	5.942	3.645	7.814	6.879	9.329
DM	2.819***		3.108***		2.066**	
MAPE	**0.024**	0.057	0.030	0.070	0.061	0.079
DM	3.326***		2.934***		1.947**	
RSR	**0.307**	0.451	0.378	0.632	0.468	0.706
SDAPE	**0.022**	0.031	0.026	0.046	0.033	0.051

performance of depth-gated recurrent neural network in predicting credit spreads.

Figure 7 further shows the fitting curve of the DGRNN model on the prediction set in the U.S. credit spread. It can be seen that the SAM-DJANET curve fits best.

Li believed that simply assigning statistical data to random without testing its certainty and randomness will lead to large deviations between the predicted results and the actual values [38]. In other words, the prediction outcome and accuracy are largely dependent on a reasonable prediction model and the randomness of the original data of the predicted variable. If the original data exhibits a certain logical change and is less random, adopting an appropriate prediction model will inevitably improve prediction performance and have higher accuracy. Therefore, following the method in Tang et al. [39], this paper applies the Ljung-Box statistic to conduct a random independence test on the original credit spread data. The test results show that its Ljung-Box statistic value is 3519.1, and the p-value is 0.00, and the original hypothesis is rejected at a 1% significant level. The results indicate that the original data are not randomly independent, laying a statistical foundation for exploring the best fitting model. The proper models can be extrapolated extensively, and prediction results in the test set should be very close to the actual real value. What is more, these models can extract the original data information. Furthermore, their residual sequences are white noise sequences, which meet random independence.

To further test the prediction extrapolation ability of the model, we perform a Ljung-Box test on the residual sequence of the prediction set in the SAM-DJANET model. The Ljung-Box statistic value is 3.07, and the p-value is 0.19, indicating that the null hypothesis cannot be rejected even at the 10% significance level. The results suggest that the residual sequence is a white noise sequence, confirming that depth-gated recurrent neural network with self-attention mechanism (SAM-DGRNN) has good predictive extrapolation ability and rationality.

5.4. Robustness Test. This paper conducts a robustness test from the following two aspects: (1) setting the cutting point of the training set and prediction set to 3 : 1, and the results are shown in Table 7; (2) shortening the sample interval to 2015–2018 and the results are shown in Table 8. According to the evidence in Tables 7 and 8, the empirical results of this paper are robust.

6. Conclusions

Traditional prediction data and technologies are insufficient to forecast the credit spread accurately, particularly when using big data. Considering the nonlinear changes in credit spreads, this paper introduces a deep learning algorithm to build depth-gated recurrent neural network with a self-attention mechanism. Additionally, it compares various prediction methods. We choose multiple evaluation indicators and randomness tests for predicted variables' original data to conduct a comparative analysis. The conclusions are as follows.

First, traditional intelligent algorithms such as machine learning and deep learning can capture nonlinear relationships better than linear algorithms. The prediction results of credit spread prediction indicate that deep learning models (LSTM, GRU, and JANET) and traditional machine learning models (SVR, MLP, and RFR) are better than the VAR model. Second, deep learning models with gated unit mechanisms are extremely advantageous in mining the long-term dependence of sequence data. The results show that the deep learning models with gated unit mechanisms have better prediction accuracy and higher stability than those without gated. Third, when predicting credit spreads, JANET, the latest deep learning model that has only one gated unit, excels in prediction efficiency, accuracy, and stability compared with the earlier models, which have more gated units, such as LSTM and GRU. GRU model with two gated units is superior to the LSTM model with three gating units. Fourth, deep learning models with SAM can efficiently

filter out critical information to the current task. The prediction of the credit spreads in the U.S. bond market shows that the deep learning model based on the attention mechanism has better prediction performance than that without the mechanism.

In summary, by comparing each model's prediction results and robustness tests through statistical performance indicators, we confirm that depth-gated recurrent neural network (DGRNN) is an effective prediction method for the U.S. bond credit spread. SAM-DGRNN model can further improve prediction performance. Among the three gated recurrent neural network models, the SAM-DJANET has the highest prediction accuracy, stability, and efficiency. The prediction results can provide a reference for the decision-making of market participants and regulatory authorities in the U.S. bond market.

However, we consider some factors that may affect the credit spread, and there may be other influencing factors. So, we can explore adding more relevant variables to improve the forecasting effect. We can further find a more accurate model in a certain type of credit spread according to the maturity, rating, and industry. In addition, these principles and forecasting methods can extend to the relevant problems of financial time series.

Data Availability

The data used to support the findings of the study were obtained from https://fred.stlouisfed.org.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research was supported by the National Natural Science Foundation of China under Grant no. 71871062.

References

[1] R. X. Zhou, S. N. Du, M. Yu, and F. Yang, "Pricing credit spread option with Longstaff-Schwartz and GARCH models in Chinese bond market," *Journal of Systems Science and Complexity*, vol. 28, no. 6, pp. 1363–1373, 2015.

[2] A. Shresthaand and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, Article ID 53040, 2019.

[3] G. Shan, H. Wang, W. Liang, and K. Chen, "Robust encoder-decoder learning framework for offline handwritten mathematical expression recognition based on a multi-scale deep neural network," *Science China Information Sciences*, vol. 64, no. 3, Article ID 139101, 2021.

[4] J. Wu and H. Li, "Deep ensemble reinforcement learning with multiple deep deterministic policy gradient algorithm," *Mathematical Problems in Engineering*, vol. 2020, Article ID 4275623, 12 pages, 2020.

[5] A. S. Temür, M. Akgün, and G. Temür, "Predicting housing sales in Turkey using ARIMA, LSTM and hybrid models," *Journal of Business Economics and Management*, vol. 20, no. 5, pp. 920–938, 2019.

[6] C. N. V. Krishnan, P. H. Ritchken, and J. B. Thomson, "Predicting credit spreads," *Journal of Financial Intermediation*, vol. 19, no. 4, pp. 529–563, 2010.

[7] J. Zhai, C. Yang, and G. Tao, "Deep learning for price movement prediction using convolutional neural network and long short-term memory," *Mathematical Problems in Engineering*, vol. 2020, Article ID 2746845, 13 pages, 2020.

[8] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[9] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: a hybrid model integrating lstm with multiple garch-type models," *Expert Systems with Applications*, vol. 103, no. 8, pp. 25–37, 2018.

[10] G. Tauchen and H. Zhou, "Realized jumps on financial markets and predicting credit spreads," *Journal of Econometrics*, vol. 160, no. 1, pp. 102–118, 2011.

[11] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] Z. Y. Wang, T. Zhang, Y. Y. Shao, and B. Ding, "LSTM-convolutional-BLSTM encoder- decoder network for minimum mean-square error approach to speech enhancement," *Applied Acoustics*, vol. 172, Article ID 107647, 2021.

[14] M. Ma and Z. Mao, "Deep-convolution-based LSTM network for remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1658–1667, 2021.

[15] G. Petmezas, K. Haris, L. Stefanopoulos et al., "Automated atrial fibrillation detection using a hybrid CNN-LSTM network on imbalanced ECG datasets," *Biomedical Signal Processing and Control*, vol. 63, Article ID 102194, 2021.

[16] W. Yu, J. Gonzalez, and X. O. Li, "Fast training of deep LSTM networks with guaranteed stability for nonlinearsystem modeling," *Neurocomputing*, vol. 422, pp. 85–94, 2021.

[17] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, and H. Schwenk, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[18] W. Liu, Q. Wang, Y. L. Zhu, and H. Chen, "GRU: optimization of NPI performance," *The Journal of Supercomputing*, vol. 76, no. 5, pp. 3542–3554, 2020.

[19] A. Nelsayed, S. Maida, and M. Bayoumi, "Gated recurrent neural networks empirical utilization for time series Classification," in *Proceedings of the IEEE Int Congress on Cybermat*, pp. 1207–1210, IEEE, Atlanta. New York, NY, USA, July 2019.

[20] L. Z. Wu, C. Kong, X. H. Hao, and W. Chen, "A short-term load forecasting method based on GRU-CNN hybrid neural network model," *Mathematical Problems in Engineering*, vol. 2020, Article ID 1428104, 10 pages, 2020.

[21] M. Y. Pan, H. N. Zhou, J. Y. Cao et al., "Water level prediction model based on GRU and CNN," *IEEE Access*, vol. 8, Article ID 60090, 2020.

[22] J. Westhuizen and J. Lasenby, "The unreasonable effectiveness of the forget gate," 2018, https://arxiv.org/abs/1804.04849.

[23] A. Vaswani, N. Shazeer, and N. Parmar, "Attention is all you need," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 6000–6010, 2017.

[24] H. K. Zhao, L. K. Wu, and Z. Li, "Dynamic forecast of Internet financial market based on deep neural network structure," *Journal of Computer Research and Development*, vol. 56, no. 8, pp. 1621–1631, 2019.

[25] S. Chen and L. Ge, "Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction," *Quantitative Finance*, vol. 19, no. 9, pp. 1507–1515, 2019.
[26] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, https://arxiv.org/abs/1502.03167.
[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[28] X. Li, S. Chen, X. L. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2682–2690, Long Beach, CA, USA, June 2019.
[29] P. Luo, J. M. Ren, Z. L. Peng, R. Zhang, and J. Li, "Differentiable learning-to-normalize via switchable normalization," 2018, https://arxiv.org/abs/1806.10779.
[30] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2017, https://arxiv.org/abs/1412.6980.
[31] E. J. Elton, M. J. Gruber, D. Agrawal, and C. Mann, "Explaining the rate spread on corporate bonds," *The Journal of Finance*, vol. 56, no. 1, pp. 247–277, 2001.
[32] I. Loncarski and P. G. Szilagyi, "Empirical analysis of credit spread changes of us corporate bonds," *International Review of Financial Analysis*, vol. 24, no. 3, pp. 12–19, 2012.
[33] E. Clark and S. Baccar, "Modelling credit spreads with time volatility, skewness, and kurtosis," *Annals of Operations Research*, vol. 262, no. 2, pp. 431–461, 2018.
[34] L. Wang and C. H. Nie, "Forecasting ability of Chinese bond spread to macroeconomic indicators," *Management Review*, vol. 31, no. 5, pp. 66–76, 2019.
[35] D. Duffie and M. Huang, "Swap rates and credit quality," *The Journal of Finance*, vol. 51, no. 3, pp. 921–949, 1996.
[36] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," *Journal of Business & Economic Statistics*, vol. 20, no. 1, pp. 134–144, 2002.
[37] D. N. Moriasi, J. G. Arnold, and M. W. VLiew, "Model evaluation guidelines for systematic quantification of accuracy in watershed simulations," *Transactions of the ASABE*, vol. 50, no. 3, pp. 885–900, 2007.
[38] J. C. Li, "On the randomness of statistical data," *China Statistics*, vol. 66, no. 7, pp. 24-25, 2018.
[39] X. B. Tang, R. Zhang, and L. X. Liu, "Forecast of second-hand housing price in Beijing based on bat algorithm SVR model," *Statistical Research*, vol. 35, no. 11, pp. 71–81, 2018.