WEP Warwick Evans Publishing

Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 9
AIDML 2025

Forecasting USD/CNY Exchange Rates Using Machine Learning

Ruihua Gong*

School of Computer Engineering and Science, Shanghai University, Shanghai, China

*Corresponding author: Jerrygong666@shu.edu.cn

Abstract. Exchange rate prediction is important in global trade, investment, and financial decision-making. The USD/CNY exchange rate in particular has received much attention due to China's growing role in the global economy. However, predicting exchange rates is difficult because the data is complex, nonlinear, and often influenced by many factors. This research focuses on predicting the USD/CNY exchange rate using machine learning-based approaches. Three models are evaluated and compared in this work: SVR, LSTM, and GRU. Historical exchange rate data is from January 1, 2015, to March 20, 2025. Model performance is evaluated by the R^2score, MSE, MAE, and RMSE. The results show that all three models can capture exchange rate trends. GRU achieves the highest performance. It reaches an R^2score of 0.9311 and the lowest error metric. LSTM reaches an R^2 of 0.9141.SVR reaches an R^2 of 0.9005 The findings suggest that deep learning approaches offer better results in financial time series forecasting and it can provide valuable insights for future applications in exchange rate prediction.

Keywords: Exchange rate forecasting; USD/CNY; Time series prediction; Deep learning; Financial modeling.

1. Introduction

Foreign exchange rates play a critical role in global financial stability, international trade, and investment decision-making. Accurate forecasting of exchange rates is essential for policymakers, investors, and multinational corporations to manage risk and develop effective strategies. However, exchange rate time series are known to be highly nonlinear, non-stationary, and noisy, making them difficult to model using traditional statistical methods.

Classical approaches such as Autoregressive Integrated Moving Average (ARIMA) [1] and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) [2] have been widely used in the past for time series forecasting. While these models are effective under certain assumptions, they struggle to capture the complex patterns and structural breaks that often exist in financial data [3].

In recent years, machine learning and deep learning techniques have emerged as powerful alternatives for financial forecasting tasks. Among these, SVR has shown effectiveness in handling nonlinear regression problems with limited data and high dimensionality [4]. The LSTM and GRU architectures, as two common types of RNNs, are well-suited for capturing extended temporal relationships in sequential data. These models have been extensively utilized in applications such as stock market prediction, exchange rate estimation, and economic time series modeling [5].

This study selects SVR, LSTM, and GRU as representative models to explore their performance in forecasting the USD/CNY exchange rate. By training and evaluating all three models on the same historical dataset, this paper aims to compare their predictive accuracy and identify the most suitable method for this specific financial forecasting task.

2. Dataset

In this study, publicly available historical exchange rate data were collected from Yahoo Finance. The dataset focuses on the USD/CNY exchange rate and spans a period of more than ten years, covering daily records from January 1, 2015, to March 20, 2025. Each entry in the dataset contains key financial indicators, including the transaction date, the opening and closing values, and the intraday maximum and minimum exchange rates for each trading session.

To prepare the data for modeling, the exchange rate series was first examined for missing or abnormal values. The records were then organized chronologically to maintain the temporal order of the financial time series. Following this, the "Close" price was selected as the target variable, while other features such as Open, High, Low, and Volume were used as input variables. To improve model convergence and performance, all feature values were normalized using MinMax scaling.

The dataset forms the basis for model development and assessment in subsequent parts of the study. Its structure facilitates the examination of both trend characteristics and volatility behavior within the USD/CNY exchange rate.

Spanning from January 1, 2015, to March 20, 2025, the dataset offers uninterrupted daily observations of the USD/CNY exchange rate throughout the entire period. Prior to model development, several preprocessing procedures were applied to ensure data quality and enhance model performance. The data were first examined for missing or null values. Subsequently, all input features were normalized to a [0, 1] scale using the MinMaxScaler method, which contributes to faster model convergence and maintains consistency across different learning algorithms [6]. Finally, the dataset was chronologically divided into training and testing subsets, with 80% of the data used for training and the remaining 20% reserved for evaluation purposes.

The dataset provided a solid foundation for training and comparing the performance of SVR, LSTM, and GRU models in predicting the closing exchange rate of USD/CNY.

3. Method

3.1. Support Vector Regression

SVR derived from the foundational principles of SVM, originally developed for classification problems by Vapnik under the Statistical Learning Theory framework. In the context of financial time series prediction, SVR has gained popularity for its strong generalization capability when dealing with high-dimensional, nonlinear, and small-scale datasets. Unlike conventional regression models, which attempt to minimize the overall prediction error, SVR introduces the concept of an ϵ-insensitive margin (also known as theε-tube), which allows the model to ignore small deviations between predicted and actual values within a specified tolerance zone [4].

In SVR, a regression function f(x) is constructed in such a way that its deviation from the true target values y_i does not exceed the ϵ margin across all training instances, while ensuring the function remains flat. The function can be defined as:

$$f(x) = w^T \phi(x) + b \tag{1}$$

Here, $\phi(x)$ denotes a nonlinear transformation that projects the original input into a higher-dimensional feature space, while w and b correspond to the model's weight vector and bias term, respectively. The optimization objective is formulated as:

$$\min_{w,\xi,\xi^*} \frac{1}{2} \left| |w||^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^* \right) \tag{2}$$

Subject to:

$$\begin{cases} y_i - w^T\phi(x_i) - b \le \epsilon + \xi_i \\ w^T\phi(x_i) + b - y_i \le \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \quad (3)$$

Where, ϵ defines the width of the ϵ-insensitive tube; ξ_i, ξ_i^* are slack variables for penalizing predictions outside the tube; The balance between the capacity of the model and its permissible deviation from the ϵ-insensitive margin is governed by the regularization parameter C.

In addressing non-linear distributions within the data, Support Vector Regression (SVR) utilizes kernel-based transformations that project the original input into a latent space with higher dimensionality. Among various kernel choices, the RBF kernel is the most commonly used in time series applications. The RBF kernel is defined as:

$$K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2) \quad (4)$$

Where$\gamma$is a parameter that defines the spread of the kernel. A smallerγimplies a smoother decision boundary, while a largerγallows the model to fit more complex patterns [7].

In this study, the SVR model was developed using the SVR class from the scikit-learn Python library. Based on parameter tuning through grid search and reference to existing literature, the following configuration was adopted: the kernel function was set to radial basis function (RBF) to capture non-linear relationships within the data. The regularization parameter C was set to 100, imposing a relatively high penalty on prediction errors outside the ϵ margin, thereby encouraging the model to fit the data more closely. The γ parameter was set to 0.01, which controls the width of the RBF kernel; a lower value of gamma promotes a smoother estimation function. Furthermore, the ϵ-insensitive margin was specified as 0.1, establishing a threshold below which deviations between predicted and actual values are not penalized.

This parameter configuration was designed to balance the trade-off between bias and variance, aiming to achieve both generalization and flexibility in modeling the fluctuations of the USD/CNY exchange rate. The SVR model is particularly suited for regression tasks involving noisy and non-linear data. Its key advantages include strong resistance to overfitting due to the regularization term C, the ability to ignore minor fluctuations through the ϵ-insensitive loss function, the flexibility of modeling complex relationships via kernel methods without explicitly increasing the dimensionality, and the sparsity of the solution, as only a subset of training instances becomes support vectors, improving computational efficiency. In the context of this research, SVR serves as a strong baseline to compare with more complex deep learning models. Despite its relative simplicity, SVR demonstrates strong performance in capturing trends in the USD/CNY exchange rate, especially when the input features are carefully selected and scaled.

3.2. Long Short-Term Memory

LSTM, a variant of RNNs, was proposed in 1997 to overcome difficulties in capturing distant temporal dependencies within sequential data. Traditional RNN models often encounter difficulties in preserving information across distant time steps due to the vanishing gradient phenomenon, a limitation that LSTM was specifically designed to overcome. LSTM networks utilize a more sophisticated internal mechanism consisting of memory cells and gating functions that enable the selective retention and update of information across time steps [5, 8].

A distinguishing feature of the LSTM architecture is its gated memory structure, which plays a crucial role in modeling sequential data with extended temporal dependencies. This design incorporates three primary gating mechanisms responsible for regulating how information is retained or discarded during processing. The first of these, the forget gate, evaluates which components of the prior cell state are preserved or eliminated, thereby allowing the network to maintain relevant contextual information across time steps. Second, the input gate controls the extent to which new information from the current input is written into the cell state, enabling the model to incorporate important features while filtering out noise. Finally, the output gate decides which part of the updated cell state should be output as the hidden state at the current time step, thereby influencing the prediction or further propagation in the sequence. These gates collectively allow LSTM networks to mitigate the vanishing gradient problem and maintain stable learning over long sequences, making them particularly suitable for time series forecasting tasks such as exchange rate prediction.

The equations governing the internal dynamics of an LSTM cell at time step t are as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{5}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{6}$$

$$\tilde{C}_t = \tanh(W_C \cdot [ht - 1, x_t] + b_C) \tag{7}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}t \tag{8}$$

$$o_t = \sigma(W_o \cdot [ht - 1, x_t] + b_o) \tag{9}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{10}$$

In the LSTM framework, the input at time step t is represented by x_t, while h_{t-1}refers to the hidden state from the preceding time point, and C_t designates the internal cell state. The gating operations employ the sigmoid activation $\sigma(\cdot)$ and the hyperbolic tangent $\tanh(\cdot)$. These gates enable LSTM networks to learn when to retain information, when to update it, and when to forget irrelevant details, allowing them to handle noisy or irregular sequences—common in financial time series such as exchange rate data.

This study utilizes historical USD/CNY univariate time series data from Yahoo Finance to build an LSTM-based prediction framework. A fixed-length sliding window was implemented to transform the temporal data into a supervised format, enabling next-step forecasting. Input sequences were formatted into three-dimensional tensors—comprising sample count, timestep window, and feature dimension—to satisfy the input configuration required by Keras LSTM layers.

Model construction was carried out through the Keras interface, with TensorFlow serving as the computational backend. Training was performed using the Adam optimization algorithm, which adjusts the learning rate dynamically. For objective minimization, the MSE loss was employed, while the R^2 score served as the primary evaluation criterion.

The network architecture was designed sequentially, incorporating a single LSTM layer with 64 units and sequence output disabled. This layer accepted inputs shaped according to a predefined sliding window and feature size. A final fully connected layer comprising one neuron was appended to yield the predicted output. The compilation process utilized the Adam optimizer for weight updates and applied MSE as the loss function, a standard choice for regression-based modeling tasks.

The training process was conducted over 50 epochs with a batch size of 32, aiming to strike a balance between computational efficiency and stable convergence. The input to the model was structured using a sliding window approach, where each sample consisted of a sequence of 10 consecutive days used to predict the exchange rate of the following day. Prior to training, all input features were normalized to the range [0, 1] using the MinMaxScaler method, ensuring consistent feature scaling and improving model convergence behavior.

The dataset was divided into 80% training and 20% testing subsets. Early stopping and validation monitoring were applied to prevent overfitting. Experimental results demonstrated that LSTM was effective at capturing both short-term fluctuations and medium-term trends in the exchange rate series. Its ability to model temporal dynamics without explicit feature engineering makes it particularly suited for volatile and nonlinear financial time series.

In comparison to classical regression models, the LSTM model offered superior performance in terms of both R^2 score and prediction smoothness. This finding aligns with existing literature, which supports the robustness of LSTM models in financial forecasting domains [9].

3.3. Gated Recurrent Unit

GRU, proposed in 2014, represents a streamlined variant of RNNs that was formulated to mitigate the vanishing gradient issue commonly observed in traditional RNN architectures. Unlike LSTM, GRU simplifies the gating structure by utilizing only two control gates—reset and update—which regulate information transmission without requiring a separate memory cell. GRUs have demonstrated improved computational efficiency while maintaining comparable forecasting performance in domains such as temporal sequence modeling, speech analysis, and language understanding [10].

Given an input sequence $x_t \in \mathbb{R}^n$ and the previous hidden state $h_{t-1} \in \mathbb{R}^n$, the GRU updates its hidden state at time t using the following operations:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{11}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{12}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h). \tag{13}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{14}$$

In GRU networks, the σ serves as the non-linear activation, and element-wise multiplication is represented by $\odot$ denotes element-wise multiplication. The update gate z_t regulates the proportion of past hidden information that is preserved. r_t is the reset gate. $\tilde{h}_t$ is the candidate activation. h_t is the final updated hidden state at time step t.

The GRU's architecture enables it to selectively forget or retain information, which is particularly important for modeling temporal sequences such as exchange rates that exhibit seasonality, trends, and abrupt shocks [11].

In this experiment, the GRU model was implemented using the Keras API with a TensorFlow backend to forecast the USD/CNY exchange rate based on a univariate time series composed of daily closing prices. A sliding window technique was employed to construct the supervised learning dataset, where each input sample consisted of 60 consecutive days used to predict the exchange rate of the following day. The model architecture included an input layer accepting sequences of shape (60, 1), followed by a GRU layer with 30 units configured to output a single hidden state. To reduce the risk of overfitting, a Dropout layer with a rate of 0.2 was applied. The output layer consisted of a single dense neuron to generate the final prediction. The model was compiled using the Adam optimizer, and the MSE was used as the loss function.

The model underwent 50 full training iterations (epochs), with each iteration processing data in batches of 32 samples. To maintain the temporal integrity inherent in time series data, the dataset was partitioned sequentially without shuffling. Chronological splitting allocated 80% of the samples to training and the remaining 20% to evaluation. Such a setup helps preserve sequential patterns, mitigates data leakage, and contributes to consistent model behavior during forecasting.

4. Experiments and Results

4.1. Results and Analysis

The forecasting capability of SVR, LSTM, and GRU models on the USD/CNY exchange rate is examined in this section. Performance assessment was conducted using identical test data and evaluated through standard statistical indicators, including the coefficient of R^2, MSE, MAE, and RMSE. A comparative overview of these results is provided in Table 1.

Table 1. Evaluation metrics for all models on the test set

Model	R^2 Score	MSE	MAE	RMSE
SVR	0.9005	0.0009	0.0246	0.0308
LSTM	0.9141	0.0008	0.0199	0.0286
GRU	0.9311	0.0007	0.0163	0.0256

When receiving the paper, we assume that the corresponding authors grant us the copyright to use the paper for the book or journal in question. When receiving the paper, we assume that the corresponding authors grant us the copyright to use.

As shown in Table 1, all three models performed well in terms of R^2 score and low prediction errors. SVR, while being a classical machine learning model, achieved a respectable R^2 of 0.9005. However, the deep learning models outperformed SVR in all metrics, demonstrating their superior capacity in capturing temporal patterns and non-linear dynamics inherent in financial time series.

Among the neural models, GRU achieved the best performance, with the highest R^2 score (0.9311) and the lowest MSE, MAE, and RMSE values. This suggests that the GRU model not only captures complex relationships effectively but also generalizes well to unseen data with minimal prediction error. LSTM also performed strongly and closely followed GRU, indicating that both architectures are suitable for exchange rate forecasting, with GRU offering a more lightweight and computationally efficient alternative.

The actual vs. predicted exchange rate curves for all three models are illustrated in Figure 1, Figure 2, and Figure 3, respectively. These plots provide visual insights into how accurately each model tracks the real exchange rate traject

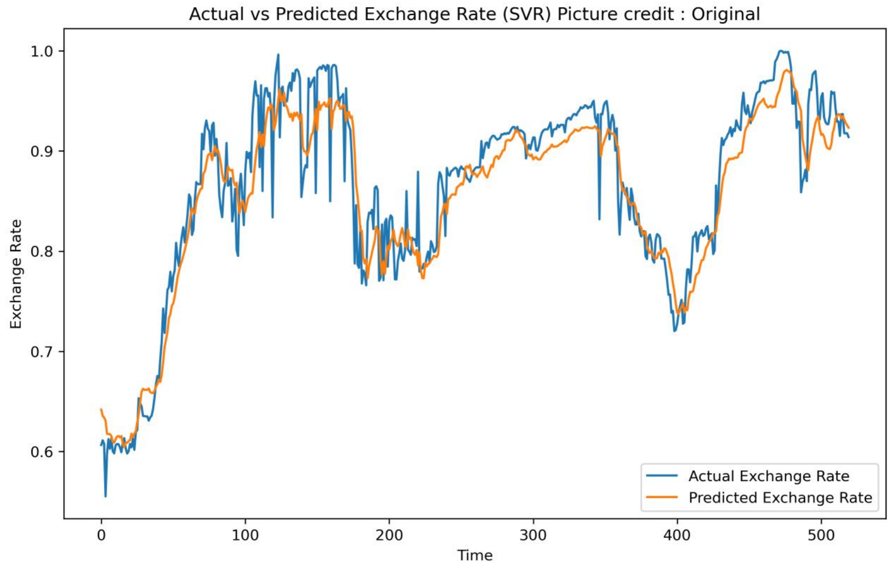

Fig. 1 SVR Prediction vs Actual (Picture credit: Original)

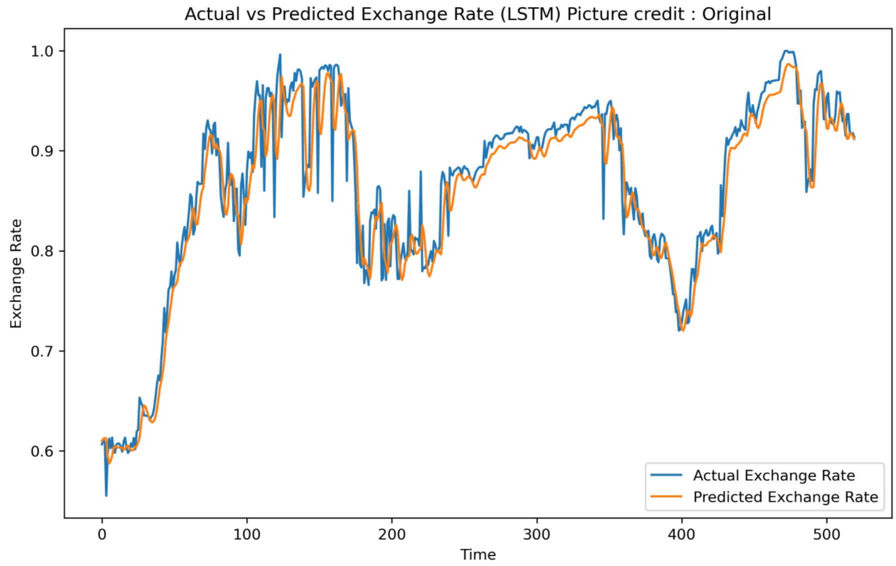

Fig. 2 LSTM Prediction vs Actual (Picture credit: Original)

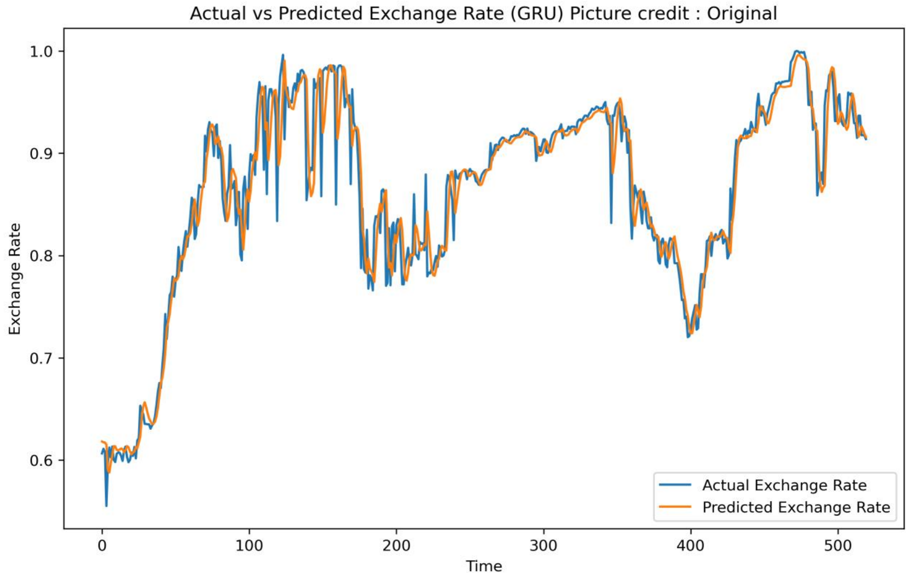

Fig. 3 GRU Prediction vs Actual (Picture credit: Original)

Each figure clearly shows that the predicted values from the GRU and LSTM models are more closely aligned with the actual exchange rate, particularly in capturing rapid fluctuations and short-term reversals. SVR, by contrast, tends to produce smoother predictions and is less responsive to sudden market changes, reflecting its structural limitations in temporal modeling.

From the above studies, it can be concluded that, first, the characteristics of the USD/CNY exchange rate data—marked by non-linear trends, seasonal fluctuations, and sudden market shocks—favor models capable of capturing complex temporal dynamics. Deep learning models such as LSTM and GRU are inherently well-suited for this task due to their recurrent architectures, which enable them to learn dependencies across time more effectively than traditional machine learning methods. Second, differences in model complexity and training requirements also played a role. SVR offers simplicity and fast training, but it lacks the sequential memory necessary for modeling long-term patterns. In contrast, LSTM provides strong modeling power but at the cost of higher computational overhead, while GRU strikes a balance by delivering comparable accuracy with reduced training time and a more compact architecture. Finally, in terms of generalization capability, GRU's streamlined structure helps mitigate overfitting risks, leading to better performance on unseen data. Although LSTM is more expressive and flexible, it typically requires additional regularization and careful parameter tuning to achieve similar levels of robustness.

5. Conclusion

This research evaluated the performance of three forecasting models—SVR, LSTM, and GRU—for predicting the USD/CNY exchange rate based on historical data. All models demonstrated satisfactory results, with GRU outperforming others in terms of accuracy and error reduction. LSTM was closely followed, while SVR, though a traditional method, served as a solid baseline.

The results suggest that deep neural architectures—particularly the GRU model—demonstrate strong suitability for capturing intricate temporal patterns in exchange rate sequences, while also exhibiting robust generalization performance. Moreover, the evaluation underscores the balance that must be struck between predictive precision, structural complexity, and computational resource demands.

However, the study is not without limitations. It focused solely on univariate time series data, excluding macroeconomic indicators or external market influences. Moreover, real-time adaptability and latency were not considered. Future work could extend this research by incorporating multivariate inputs, testing adaptive online learning algorithms, and exploring hybrid models such as Transformer-based architectures.

In summary, the research enhances the understanding of financial time series prediction by empirically validating the effectiveness of GRU-based neural architectures in modeling exchange rate dynamics.

References

[1] Shumway R H, Stoffer D S. ARIMA models. Time Series Analysis and Its Applications: With R Examples, 2017: 75–163.

[2] Bauwens L, Laurent S, Rombouts J V K. Multivariate GARCH models: a survey. Journal of Applied Econometrics, 2006, 21(1): 79–109.

[3] Tay F E H, Shen L. Economic and financial prediction using rough sets model. European Journal of Operational Research, 2002, 141(3): 641–659.

[4] Basak D, Pal S, Patranabis D C. Support vector regression. Neural Information Processing-Letters and Reviews, 2007, 11(10): 203–224.

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

[6] Sola J, Sevilla J. Importance of input data normalization for the application of neural networks to complex industrial problems. IEEE Transactions on Nuclear Science, 1997, 44(3): 1464–1468.

[7] Awad M, Khanna R. Support vector regression. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, 2015: 67–80.

[8] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 2020, 404: 132306.

[9] Siami-Namini S, Tavakoli N, Namin A S. The performance of LSTM and BiLSTM in forecasting time series. IEEE International Conference on Big Data (Big Data), 2019: 3285–3292.

[10] Dey R, Salem F M. Gate-variants of gated recurrent unit (GRU) neural networks. IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 2017: 1597–1600.

[11] Fu R, Zhang Z, Li L. Using LSTM and GRU neural network methods for traffic flow prediction. 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2016: 324–328.