Technology Stock Forecasting Based on Hybrid Model and Data Migration

Yong Lu*, Lumiao Hu

School of Information Engineering, Minzu University of China, Beijing, 100081, China;

*Corresponding author: 2006153@muc.edu.cn

Abstract. The forecasting of financial time series has long been recognized as a challenging and important task for both economists and computer scientists. In recent years, the integration of deep learning and artificial intelligence has significantly advanced the field of financial time series forecasting. Deep learning has been demonstrated to yield superior results in handling nonlinear trends, which are difficult to model using traditional linear models. In this study, we propose a two-part method that integrates linear and deep learning models to forecast financial time series. In the first part, we utilize the Autoregressive Integrated Moving Average (ARIMA) model to filter the linear trend of the stock and pass the residual value to the second part. In the second part, we employ a Convolutional Neural Network (CNN)-Bidirectional Gated Recurrent Unit (BIGRU) neural network with an attention mechanism to effectively process the residual value and make predictions.The dataset we selected consists of stocks from the technology sector, and there is a certain similarity in the trends of these stocks. To improve performance, we use data migration to improve its performance. To evaluate the proposed model, we use Mean Squared Error (MSE) and Mean Absolute Error (MAE) to measure its performance. We compare our proposed method with a benchmark approach, and the experimental results demonstrate that our method has higher prediction accuracy. Our approach thus presents significant advantages for forecasting financial time series.

Keywords: Hybrid model; ARIMA; Deep Learning; Data Migration.

1 Introduction

Forecasting has numerous applications across various areas of production and daily life. For instance, in the retail sector, probabilistic forecasting of product supply and demand can be employed for effective inventory management, staff scheduling, and layout planning. In meteorology, forecasting future rainfall and typhoon routes, wind directions, and other factors can make people's lives more convenient. In the transportation sector, traffic flow forecasting can lead to the development of intelligent transportation systems. In the financial sector, forecasting plays a significant role and can be used to predict interest rates, foreign exchange risks, stock

C. Chen et al. (eds.), *Proceedings of the 3rd International Conference on Digital Economy and Computer Application (DECA 2023)*, Atlantis Highlights in Computer Sciences 17,
https://doi.org/10.2991/978-94-6463-304-7_41

market volatility, and other important variables. Financial projections are crucial for decision-makers and business professionals to make informed decisions regarding production, procurement, market sustainability, resource allocation, and other crucial issues [1,2]. Accurate forecasting is essential in data-driven decision-making, allowing for anticipation of events such as stock market volatility and making informed decisions based on predictions.

In financial analysis, the analysis of financial time series plays a critical role in understanding how financial variables, such as investment returns, evolve over time. Since all financial time series are subject to uncertainty, statistical theory and methods are fundamental in analyzing financial time series. Financial asset time series can be regarded as a sequence of random and unknown variables that change over time. In most cases, it is assumed that the random variable sequence is only defined at specific locations along the time axis, which constitutes a discrete random process. For example, a stock's daily rate of return is a discrete time series. In quantitative investing, the primary objective is to statistically model the time series of investment product returns to identify any patterns in the returns on various trading days and forecast future returns, resulting in trading signals. This involves analyzing the statistical properties of the time series to develop forecasting models that provide insights into the behavior of the underlying assets [3].

The remainder is arranged as follows. After introducing our model in Section 3, Section 4 describes the dataset and the specifics of the experiment, and Section 5 wraps up this paper, Section 2 discusses some related work.

2 Related Work

The ARIMA model [4] is a widely recognized and extensively studied method for linear univariate time series prediction, incorporating Autoregressive (AR), Moving Average (MA), and Autoregressive Moving Average (ARMA). Linear support vector regression [5, 6] (SVR) is also commonly used as a conventional regression technique with time-varying parameters. However, both of these models are limited in their ability to handle nonlinear time series data.

The nonlinear interdependencies that deep neural networks can capture have been the focus of much research in recent years. Among the various types of recurrent neural networks, the GRU has shown promise in numerous nonlinear programming tasks. In particular, the BiGRU model, which comprises both forward and reverse GRU, has become widely used in stock prediction due to its ability to gather data-related information from both directions. The model's limitation lies in its suboptimal prediction performance when solely employing BIGRU without incorporating CNN and an attention mechanism.

The DeepAR [7] algorithm is a recommended prediction model for unifying the modeling of related time series. By utilizing deep learning technology and training autoregressive recurrent network models on a vast number of time series, it can effectively learn the global model from related time series, capture complex patterns such as seasonality, and account for uncertainty growth of data over time to predict each time series accurately. The algorithm's disadvantages is the lack of an attention mechanism, which limits its ability to capture information regarding long periods and seasons. Consequently, the algorithm may suffer from memory loss when dealing with extended time series.

3 Models

In the realm of time series analysis, it is often assumed that the data can be decomposed into two primary components: a linear component and a nonlinear component. The linear portion is generally characterized by a trend that follows a straight line or an exponential curve, and possibly includes seasonal or cyclic fluctuations. On the other hand, the nonlinear portion encompasses all other types of temporal variability, including irregular or erratic fluctuations, sudden changes, and irregular oscillations [8]. The ARIMA model is a widely used mathematical model for statistical forecasting. However, it is best suited for capturing linear trends in the data. To address the limitation of ARIMA in modeling nonlinear trends, the CNN-BiGRU model with an attention mechanism has been developed. This model is capable of capturing nonlinear trends present in the data. To take advantage of the strengths of both models, an integrated approach has been proposed. This involves combining the ARIMA model with the CNN-BiGRU model with an attention mechanism. Specifically, the ARIMA model is used in the front portion of the integrated model to capture linear trends, while the back portion of the model is a CNN-BiGRU model with an attention mechanism to capture nonlinear trends. By incorporating both linear and nonlinear trends into the model, the integrated approach has the potential to improve the accuracy of statistical forecasting.

Therefore, this paper introduces an attention-driven ARIMA-CNN-BIGRU model, which utilizes trends from representative stocks within the same technology sector as assistance. The objective is to enhance forecasting quality and achieve more favorable investment outcomes.

The ARIMA (p,d,q) model is a common time-series analysis technique that is represented mathematically by the parameters p, d, and q, where p, d, and q are non-negative integers. The parameter p denotes the order or number of time lags in the autoregressive model, while d represents the degree of differencing, which refers to the number of times the data have been subtracted from their past values. Lastly, the parameter q is the order of the moving-average model.The mathematical representation of the ARIMA (p,d,q) model can be expressed as follows:

$$\left(1-\sum_{i=1}^{p}\phi_i L^i\right)(1-L)^d X_t=\left(1+\sum_{i=1}^{q}\theta_i L^i\right)\varepsilon_t \tag{1}$$

Where L is the Lag operator. Once the model is fitted, it can be used to forecast future values of the time series.

Box and Jenkins [4] introduced a widely used strategy for creating ARIMA models, which consists of three fixed steps. The first step involves identifying and selecting the appropriate type of model, such as AR (p), MA (q), ARMA (p, q), or ARIMA (p, d, q). The second step involves estimating the model parameters to optimize the model fit. Lastly, in the third step, residual analysis is conducted to further enhance the model's accuracy.

In this paper, we chose to estimate the parameters using the AIC metric.

$$\mathrm{AIC} = -2\ln(\hat{L}) + 2k \tag{2}$$

The $\ln(\hat{L})$ notation is the value of the likelihood function, and k is the degree of freedom, that is, the number of parameters used. Models with smaller AIC values are generally considered to be better models. There are different ways to calculate the likelihood function, $\ln(\hat{L})$。 We use maximum likelihood estimation of calculation. This method is slow, but the result is accurate [9].

In the CNN-BiGRU model, the data are initially processed by the CNN component, and the one-dimensional results are then input into the BiGRU model. Figure 1 illustrates the one-dimensional convolution process. Typically, hidden layers include convolutional layers, activation functions, and pooling layers. CNN layers can transfer local features from high-level inputs to lower layers for more complex features [10]. The formula and schematic diagram of CNN are as follows:

$$Y[i] = \sigma\left(\sum_{k=1}^{K} W[k]X[i+k-1] + b\right) \tag{3}$$

X is the input signal or sequence, Y is the output feature map at position i, W is the weight vector of the convolutional filter, b is the bias term, K is the size of the convolution kernel. The size of the convolution kernel, K, is typically an odd number to ensure that it can symmetrically cover every position in the input sequence. Moreover, since the size of the convolution kernel is fixed, the length of the output sequence will be $K-1$ units shorter than the input sequence.

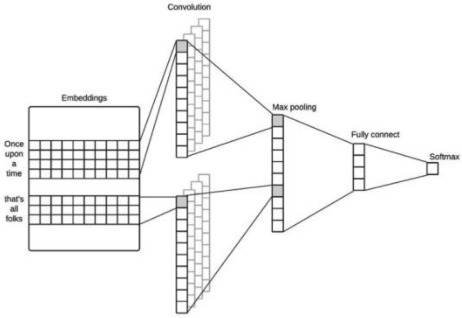

Fig. 1. One-dimensional convolution example diagram

The GRU [11] is a type of RNN that is specifically designed to handle sequential data. It possesses the ability to retain both short and long-term memory and can selectively forget irrelevant information, thereby enhancing its ability to capture long-range dependencies. Compared to its counterpart, the LSTM network, GRU is less susceptible to overfitting and has a faster training speed.The formula and schematic diagram of GRU are as follows:

$$R_t = \sigma(W_r[H_{t-1}, X_t] + b_r) \quad (4)$$

Where R_t is the reset gate at time step t, σ is the sigmoid activation function, W_r is the weight matrix for the reset gate, b_r is the bias term for the reset gate, $H_{(t-1)}$ is the hidden state at the previous time step, X_t is the input at time step t, and the update gate:

$$Z_t = \sigma(W_z[H_{t-1}, X_t] + b_z) \quad (5)$$

Where Z_t is the update gate at time step t, W_z is the weight matrix for the update gate, b_z is the bias term for the update gate. Next, the candidate hidden state:

$$\tilde{H}_t = \mathrm{Tan}h\,(W[R_t * H_{t-1}, X_t] + b) \quad (6)$$

Where $\tilde{H}_t$ is the candidate hidden state at time step t, Tanh is the hyperbolic tangent activation function. Finally, the new hidden state:

$$H_t = (1 - Z_t) * H_{t-1} + Z_t * \tilde{H}_t \quad (7)$$

Where H_t is the new hidden state at time step t. However, one of the limitations of the GRU model is that it is unable to encode information in a backward direction, which hampers its ability to handle bidirectional information. To overcome this limitation, this paper employs the BiGRU model, which is a combination of forward and backward GRU. By leveraging both forward and reverse directions, BiGRU is able to more effectively manage bidirectional information, leading to superior results.

BIGRU consists of two independent GRUs, so the formula is generally consistent with that of a GRU. However, the BIGRU model introduces an additional mathematical formula to its architecture:

$$H_t = [H_t^f, H_t^b] \quad (8)$$

In this formula, H_t is the hidden state at time step t, H_t^b is the hidden state in the backward pass at time step t, and H_t^f is the hidden state in the forward pass at time step t. The BIGRU architecture uses both forward and backward passes to learn dependencies in both directions, while selectively updating and resetting hidden states using GRU. Figure 2 depicts the architecture of the BiGRU model utilized in this study.

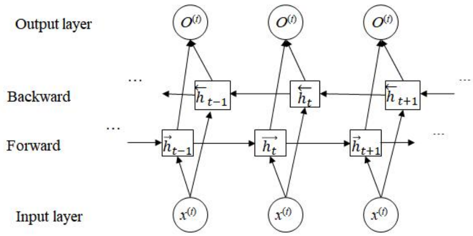

Fig. 2. One-dimensional convolution example diagram

Attention plays a crucial role in improving performance on time series tasks. Attention mechanisms allow the model to selectively focus on specific parts of the input sequence that are most relevant for making predictions at each time step. By doing so, attention mechanisms can improve the model's ability to capture long-range dependencies in the input sequence and make more accurate predictions.

4 Applications and Experiments

In this study, we utilized the annual data of stocks of the technology sector from January 1, 2018, to December 31, 2022. We conducted our analysis using the "adjusted closing price" of the stock data. Our experimental data was sourced from Yahoo Finance.In stock technical analysis, it is more common to use the Adjusted Closing Price instead of the regular closing price. This preference arises because the Adjusted Closing Price provides more accurate price data, reflecting the true market performance. Therefore, we favor the Adjusted Closing Price as it offers a more precise representation of market conditions. The codes for the technology stocks we have chosen are AAPL, MSFT, GOOG, and AMZN, corresponding to the companies Apple Inc., Microsoft Corporation, Alphabet (the parent company of Google), and Amazon. These are all highly valuable stocks in the US stock market and are highly representative.

In forecasting time series data, there are generally three approaches: daily forecasting, short-term forecasting, and long-term forecasting. This article primarily focuses on the case of short-term forecasting because the ARIMA model is more suitable for linear predictions in such scenarios. We selected a test set consisting of stock trading days from December 21, 2022, to December 31, 2022, totaling seven days. In the ARIMA model, a validation set is generally not required. In the process of deep learning, we further split the residual sequences from the training data into a training set and a validation set for training and validating the deep learning model. To maintain consistency, we also chose a seven-day duration for the validation set.

We begin by applying the ARIMA model to the original stock data to obtain linear predictions and residuals. We determine the three ARIMA parameters using AIC. Typically, the ARIMA parameters p range from 0 to 4, d ranges from 0 to 2, and q ranges from 0 to 4. Within this range, we search for the minimum AIC value to obtain the best ARIMA model parameters.The ARIMA parameters are shown in the following table 1.

Table 1. ARIMA parameter values for stocks.

Stock	P	D	Q
APPL	2	2	3
MSFT	3	1	3
GOOG	3	1	3
AMZN	2	1	2

We calculate the residual values of the data by subtracting the predicted values obtained from the ARIMA model from the actual values. These residual values are then used in the deep learning model. To improve prediction accuracy, we transform these residual values into percentages of the actual values, which helps enhance the performance of subsequent forecasts. Figure 3 is our residual percentage distribution plots.

Before feeding the residual percentages into the deep learning model, we perform data standardization. This process involves scaling the data to a specific range and transforming it into a specific interval. Normalization can significantly enhance the model's convergence speed and accuracy. The following is the normalization function formula we use:

$$X^* = (X - Min)/(Max - Min) \tag{9}$$

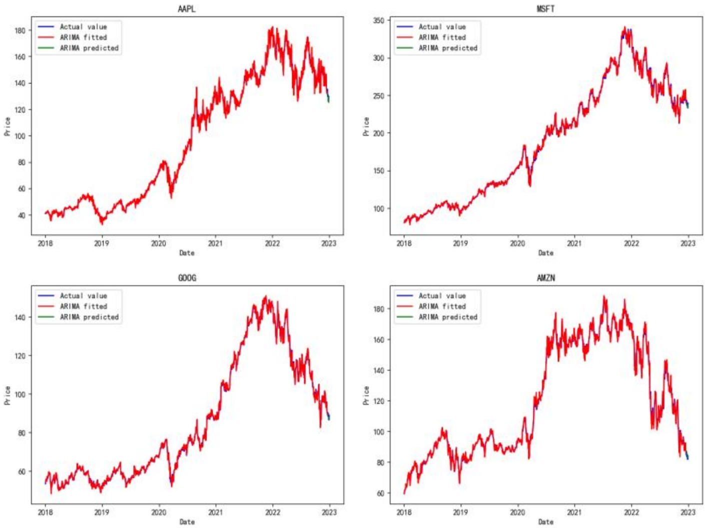

Fig. 3. ARIMA forecast plot

We further divided the training data into a training set and a validation set as per the previous plan. This helps in selecting the best model configuration to enhance the model's performance and generalization capability. To ensure consistency, our validation set also contains 7 stock trading days. We processed our dataset using the sliding window method, which segments time series data into continuous, non-overlapping subsequences, each with a fixed length. These subsequences are used as input sequences, enabling the model to capture temporal dependencies and sequence patterns. Given the characteristics of our dataset, we set the time window size to 5 and the stride to 1, which will contribute to improving the model's fitting ability.

Next, we compare our utilized deep learning model, the CNN-BiGRU model with attention, to the baseline models. We divide the comparison process of each stock into two groups according to whether there is data migration from other technology sectors. There are 4 stocks in total, so we have a total of 8 groups of experiments to compare the deep learning model we proposed with the baseline models. The model with the lowest MSE is chosen, and the MAE is also investigated for further analysis. The chosen optimal model is then evaluated for the two most recent time intervals.Additionally, we compare the anticipated value obtained after residual restoration to the actual value. Figure 4 is the final stock comparison chart in which we compare a number of baseline models [12-17]. Table 2 is the performance data value of our model comparison.

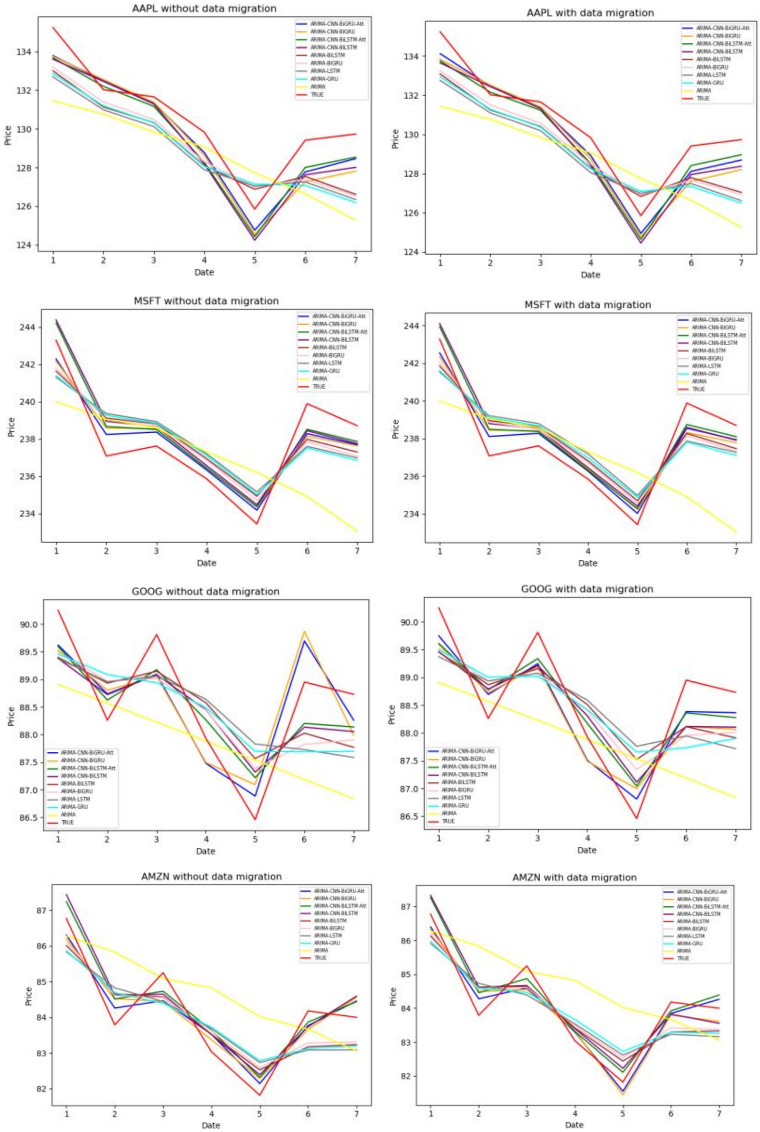

Fig. 4. Comparison chart of predicted value and actual value

Table 2. Model Performance Results and Its Comparison

Stock	Model	MSE	MAE	Stock	Model	MSE	MAE
AAPL without data migration	ARIMA-CNN-BiGRU-Att	1.2929	1.0427	MSFT without data migration	ARIMA-CNN-BiGRU-Att	1.0427	0.9641
	ARIMA-CNN-BIGRU	2.0073	1.2678		ARIMA-CNN-BIGRU	1.3911	1.1235
	ARIMA-CNN-BiLSTM-Att	1.5199	1.1189		ARIMA-CNN-BiLSTM-Att	1.1284	1.0138
	ARIMA-CNN-BILSTM	2.0662	1.3139		ARIMA-CNN-BILSTM	1.4754	1.1630
	ARIMA-BILSTM	3.5595	1.7471		ARIMA-BILSTM	2.4784	1.5395
	ARIMA-BIGRU	3.4275	1.6816		ARIMA-BIGRU	2.3769	1.4921
	ARIMA-LSTM	4.4772	1.9751		ARIMA-LSTM	3.4046	1.8072
	ARIMA-GRU	4.4309	1.9305		ARIMA-GRU	3.2837	1.7653
	ARIMA	7.2980	2.3955		ARIMA	11.7324	3.0096
AAPL with data migration	ARIMA-CNN-BiGRU-Att	0.8472	0.8540	MSFT with data migration	ARIMA-CNN-BiGRU-Att	0.7047	0.7911
	ARIMA-CNN-BIGRU	1.4581	1.0914		ARIMA-CNN-BIGRU	1.0866	0.9881
	ARIMA-CNN-BiLSTM-Att	1.0302	0.9063		ARIMA-CNN-BiLSTM-Att	0.8001	0.8404
	ARIMA-CNN-BILSTM	1.5378	1.1396		ARIMA-CNN-BILSTM	1.1523	1.0159
	ARIMA-BILSTM	2.8750	1.5786		ARIMA-BILSTM	1.8954	1.3438
	ARIMA-BIGRU	2.7637	1.5127		ARIMA-BIGRU	1.7848	1.2907
	ARIMA-LSTM	3.8738	1.8362		ARIMA-LSTM	2.7558	1.6275
	ARIMA-GRU	3.7688	1.7821		ARIMA-GRU	2.6130	1.5745
GOOG without data migration	ARIMA-CNN-BiGRU-Att	0.3280	0.5590	AMZN without data migration	ARIMA-CNN-BiGRU-Att	0.2524	0.4851
	ARIMA-CNN-BIGRU	0.4815	0.6784		ARIMA-CNN-BIGRU	0.3449	0.5690
	ARIMA-CNN-BiLSTM-Att	0.3665	0.5845		ARIMA-CNN-BiLSTM-Att	0.2733	0.5093
	ARIMA-CNN-BILSTM	0.5182	0.7040		ARIMA-CNN-BILSTM	0.3722	0.5983
	ARIMA-BILSTM	0.7194	0.8320		ARIMA-BILSTM	0.5893	0.7529

	ARIMA-BIGRU	0.6824	0.8070		ARIMA-BIGRU	0.5429	0.7304
	ARIMA-LSTM	0.9999	0.9670		ARIMA-LSTM	0.8407	0.9067
	ARIMA-GRU	0.9411	0.9415		ARIMA-GRU	0.7902	0.8822
GOOG with data migration	ARIMA	1.7487	1.1443	AMZN with data migration	ARIMA	1.9402	1.1607
	ARIMA-CNN-BiGRU-Att	0.2161	0.4576		ARIMA-CNN-BiGRU-Att	0.1604	0.3770
	ARIMA-CNN-BIGRU	0.3784	0.6020		ARIMA-CNN-BIGRU	0.2431	0.4692
	ARIMA-CNN-BiLSTM-Att	0.2643	0.4996		ARIMA-CNN-BiLSTM-Att	0.1724	0.3927
	ARIMA-CNN-BILSTM	0.4179	0.6305		ARIMA-CNN-BILSTM	0.2792	0.5048
	ARIMA-BILSTM	0.6103	0.7656		ARIMA-BILSTM	0.4589	0.6602
	ARIMA-BIGRU	0.5739	0.7420		ARIMA-BIGRU	0.4226	0.6415
	ARIMA-LSTM	0.8467	0.8951		ARIMA-LSTM	0.6938	0.8209
	ARIMA-GRU	0.7971	0.8597		ARIMA-GRU	0.6338	0.7915

5 Conclusions

The objective of the present investigation is to suggest a novel stock market prediction model that surpasses the efficacy of prevailing approaches. At the same time, combined with the technology sector dataset we have chosen, a data migration method has been proposed to effectively improve model performance. Specifically, this paper proposes a novel hybrid model called ARIMA-CNN-BiGRU, which filters out linear trends during the ARIMA process, and utilizes the residual values to feed into the attention-based CNN-BiGRU model for prediction and evaluation. This approach is complemented by a data migration method for comprehensive assessment. To evaluate the performance of our proposed model, we employ MSE and MAE metrics. The results demonstrate that our model outperforms existing methods, contributing to portfolio optimization and enhancing investment returns.

It is important to note that our predictive models have limitations in accounting for all possible trends. Additionally, untrained trends that could potentially affect the model's predictive accuracy may be present. Hence, it is crucial to acknowledge that financial irregularities and noise are inherent, making it challenging for any model to consider all conceivable unique patterns. Furthermore, the ability of the model to predict unexpected events still needs to be addressed in future research.

References

1. Seyedimany A. Stock price reactions on nasdaq stock exchange for special dividend announcements. Emerging Science Journal, 2019, 3(6), 382-388.
2. Ilyas Q M, Iqbal K, Ijaz S, et al. A Hybrid Model to Predict Stock Closing Price Using Novel Features and a Fully Modified Hodrick–Prescott Filter. Electronics, 2022, 11(21), 3588.
3. Liu W, Yang Z, Cao Y, et al. Discovering the influences of the patent innovations on the stock market. Information Processing & Management, 2022, 59(3), 102908.
4. Box G E P, Jenkins G M, Reinsel G C, et al. Time series analysis: forecasting and control. John Wiley & Sons, 2015.
5. Cao L J, Tay F E H. Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on neural networks, 2003, 14(6), 1506-1518.
6. Tay F E H, Cao L J. Modified support vector machines in financial time series forecasting. Neurocomputing, 2002, 48(1-4), 847-861.
7. Salinas D, Flunkert V, Gasthaus J, et al. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 2020, 36(3), 1181-1191.
8. Dave E , Leonardo A , Jeanice M , et al. Forecasting Indonesia Exports using a Hybrid Model ARIMA-LSTM. Procedia Computer Science, 2021, 179(1), 480-487.
9. Moschini G , Houssou R , Bovay J , et al. Anomaly and Fraud Detection in Credit Card Transactions Using the ARIMA Model. Engineering Proceedings, 2021.
10. Agga A, Abbou A, Labbadi M, et al. Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models. Renewable Energy, 2021, 177, 101-112.
11. Cho K , Merrienboer B V , Gulcehre C , et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Association for Computational Linguistics, 2014.
12. Pirani M, Thakkar P, Jivrani P, et al. A Comparative Analysis of ARIMA, GRU, LSTM and BiLSTM on Financial Time Series Forecasting. 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). IEEE, 2022, 1-6.
13. Mangalampalli R, Pandey V, Khetre P, et al. Stock prediction using hybrid ARIMA and GRU models. International Journal of Engineering Research & Technology (IJERT), 2020, 9(5), 2278-0181.
14. She D, Jia M. A BiGRU method for remaining useful life prediction of machinery. Measurement, 2021, 167, 108277.
15. Fan D, Sun H, Yao J, et al. Well production forecasting based on ARIMA-LSTM model considering manual operations. Energy, 2021, 220, 119708.
16. Ji L, Zou Y, He K, et al. Carbon futures price forecasting based with ARIMA-CNN-LSTM model. Procedia Computer Science, 2019, 162, 33-38.
17. Lu W, Li J, Wang J, et al. A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications, 2021, 33, 4741-4753.