

An improved convolutional recurrent neural network for stock price forecasting

Pham Hoang Vuong^{1,2}, Lam Hung Phu², Le Nhat Duy¹, Pham The Bao², Tan Dat Trinh²

¹Department of Computer Science, Faculty of Information Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

²Department of Computer Science, Faculty of Information Science, Sai Gon University, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Jan 22, 2024

Revised Feb 28, 2024

Accepted Mar 8, 2024

Keywords:

Attention mechanism
Convolutional neural network
Convolutional recurrent neural network
Long short-term memory
Stock price forecasting

ABSTRACT

Stock price forecasting is a challenging area of research, particularly due to the complexity and unpredictability of financial markets. The accuracy of prediction models is influenced by various factors, including nonlinearity, seasonality, and economic shocks. Deep learning has demonstrated better forecasts of stock prices than traditional approaches. This study, therefore, proposed a new approach to improve forecasting system based on an end-to-end convolutional recurrent neural network (CRNN) with attention mechanism. Our approach first investigates local stock price features using 1D convolutional neural network, and then employs a bidirectional long short-term memory (Bi-LSTM) network for forecasting. This model stands out by effectively utilizing contextual data and representing the temporal character of data. The Bi-LSTM is helpful for understanding the history and future contextual information since it uncovers both past and future contexts of stock data. Furthermore, integrating attention mechanism within the CRNN represents a significant improvement. This allows our model to handle long input sequences more effectively and capture the inherent stochasticity in stock prices, which is often missed by traditional models. The effectiveness of our approach is investigated using data on 10 stock indexes from Yahoo Finance. The results show that our method outperforms autoregressive integrated moving average (ARIMA), long short-term memory (LSTM), and conventional methods.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Tan Dat Trinh

Department of Computer Science, Faculty of Information Science, Sai Gon University

273 An Duong Vuong Street, Ward 3, District 5, Ho Chi Minh City, Vietnam

Email: trinhtandat@sgu.edu.vn

1. INTRODUCTION

Financial markets, known for their complex and fluctuating nature, are often described as nonlinear, dynamic, and stochastic systems [1], [2]. This complexity makes understanding and predicting market behavior, particularly stock prices, a challenging in the financial sector. Among various financial analyses, stock price forecasting (SPF) stands out as an important task. The SPF involves using time-series data analysis to predict future stock prices based on historical trends and data [2], [3]. This task presents challenges such as dealing with incomplete, distorted, or non-stationary data, as noted in the literature [4], [5]. Furthermore, stock prices are influenced by a multitude of external factors. These range from global events like the COVID-19 pandemic to fluctuations in oil prices, seasonal market shifts, changes in economic policy, the performance of individual firms, and even political upheavals [1], [6]. Each of these elements can significantly impact the data used in SPF. This study focuses on predicting future stock prices using historical market data. The objective is

to develop a more robust and accurate method for SPF that can effectively address the challenges presented by the dynamic nature of financial markets. It aims to provide valuable insights into the field of financial analysis and forecasting.

The literature review of SPF methods over recent decades highlights the progression from traditional statistical models to advanced machine learning and deep learning techniques [2], [3], [7]. Initially, traditional statistical models like the autoregressive integrated moving average (ARIMA) and its variants [8], [9] were effective for short-term predictions. However, their assumption of linear relationships limits their applicability to nonlinear market data.

To address these limitations of traditional systems, machine learning and deep learning approaches were introduced. The traditional machine learning approaches, including artificial neural networks (ANNs) [10], k-nearest neighbor (KNN) [11], support vector machines (SVM) [12], [13], and bayesian networks (BN) [14], have been successfully used in SPF systems. Nevertheless, they often require handcrafted feature extraction and might struggle with the complexity or non-linear nature of market data.

Deep learning approaches, including convolutional neural networks (CNNs), long short-term memory (LSTM), and bidirectional long short-term memory (Bi-LSTM), are being widely applied. Khanna *et al.* [15] implemented a 1D CNN for pre-training purposes. After pre-training, the model was fine-tuned and evaluated utilizing a dataset of NIFTY 200 stocks. The results showed an enhancement in the model's efficacy relative to baseline models without pre-training. Akşehir and Kiliç [16] introduced a new rule-based labeling algorithm to address data imbalance in stock price prediction. The effectiveness of their method was tested using a 2D-CNN deep neural network. The results showed that their proposed approach enhanced the model's predictive performance. The CNNPred model was used to extract features for forecasting the future of various markets [17]. The experimental results were evaluated using indices such as the S&P 500, NASDAQ, DJI, NYSE, and RUSSELL. Authors claimed that their method outperforms state-of-the-art baseline methods. Studies have shown enhancements in the effectiveness of models using CNNs for feature extraction. Although CNNs are highly effective in processing spatial data, they are not ideally suited for temporal sequences. This limitation can result in the neglect of long-term dependencies, which are essential for analyzing stock data.

The recurrent neural network (RNN) and LSTM were specifically designed to handle sequential data. They were proposed for SPF [18]. They can capture and learn from temporal dependencies in time series data, which is a important characteristic of stock prices. Ghosh *et al.* [19] introduced LSTM networks and random forests to forecast directional movements of stock prices from the S&P 500 index for intraday trading. Gao *et al.* [20] proposed an approach for predicting stock prices using LSTM and gated recurrent neural network (GRU) models. The authors employed deep learning least absolute shrinkage and selection operator (LASSO) and principal component analysis (PCA) for dimensionality reduction, focusing on various factors influencing stock prices. The experiments demonstrated the efficiency of both LSTM and GRU models in stock price prediction.

LSTMs process sequences sequentially, but they can be slow and might be difficult to capture very long-term dependencies. Bi-LSTM [21] has also been explored, combining two LSTMs for improved forecasting accuracy [22], [23]. A stacked technique was suggested [21] to forecast the stock price. They first applied wavelet transform and stacked auto-encoder to reduce noise and unnecessary features. The future stock price was then predicted using the RNN and Bi-LSTM. Liu *et al.* [22] introduced the influence of air pollutants on stock price predictions. This research concentrates on the shanghai stock exchange (SSE) enterprises index and integrates six categories of air pollutants as input features. These features, along with conventional financial data, are utilized to construct a predictive model using Bi-LSTM. Mootha *et al.* [23] proposed a Bi-LSTM to forecast the future open, high, close, and low (OHCL) value of a stock.

Furthermore, hybrid models were also introduced to enhancing the accuracy of the SPF [24], [25]. Recently, the model was trained end-to-end into a single deep learning model using new robust techniques based on a combination of CNN and RNN [26], [27]. The CNN-LSTM was proposed [26] as a method of predicting stock closing prices. The features from the input data were extracted by the authors using CNN, and the features were then fed to the LSTM for prediction. To predict the stock price, the CNN model and Bi-LSTM were combined [27]. Lu *et al.* [28] proposed a combination of CNN-BiLSTM and attention mechanism. The CNN is used in the study to extract features from the input data. The Bi-LSTM was then fed the feature vectors for prediction. To improve performance, they also used attention mechanisms to perform how feature vectors affected the stock data. To enhance the performance of SPF, the CNN-BiLSTM based on improved attention mechanism was introduced [29]. The authors suggested an effective channel attention (ECA) module to increase the network's sensitivity to important features and data. Results from experiments demonstrated that the suggested CNN-BiLSTM-ECA model outperforms baseline approaches.

In this study, we propose an end-to-end deep learning model via convolutional recurrent neural network (CRNN) combined with attention mechanism to enhance the accuracy of the SPF system. To examine the local stock closing price feature learning, we first employ the 1D-CNN layer. The Bi-LSTM was then fed

these features to take advantage of the input time stock price data's past and future contexts. For learning historical and upcoming contextual knowledge, the Bi-LSTM is helpful. Finally, CRNN is combined with an attention mechanism to help it focus on important information of the input data. The contribution of this study is described as follows:

- Our method explores the efficiency of deep CNNs, Bi-LSTM, and attention mechanisms. Our framework can learn local stock data features through 1D-CNN, exploring sequence modeling, and comprehending past and future contextual features via Bi-LSTM. Additionally, it decodes frames into a sequence of data by employing an attention mechanism.
- The attention mechanism plays a important role in enhancing the accuracy of forecasts by focusing on relevant information, thereby assigning importance weight to specific data. Moreover, attention mechanism helps reduce the vanishing gradient problem. So, the attention mechanism not only improves the overall predictive capability of the model but also ensures a more efficient and effective learning process.
- We utilize actual data consisting of 10 stock prices obtained from Yahoo Finance. Our evaluation involves comparing our model with other well-known models. The results confirm that the CRNN+attention model shows promise for SPF system.

The remainder of this paper is structured as follows: in section 2, a method for the achievement of SPF is proposed. Section 3 presents the experimental results of the proposed approach and the comparison with the other approaches. Finally, section 4 concludes the paper.

2. METHOD

In this section, we introduce the details of the proposed CRNN model. The approach consists of four components such as input layer, convolutional layer, recurrent layer and output layer as illustrated in Figure 1. The raw input data is first normalized in the preprocessing step. To extract feature sequence from the preprocessed data, the data is then put into a convolutional layer. In this study, one-dimensional CNN has been applied to SPF. For predicting the stock price, we first employ a 1D CNN model, then a Bi-LSTM. Finally, we use attention mechanisms to improve SPF performance.

Preprocessing is necessary before the prediction method to be performed. Data normalization is first applied to reduce the large gap and inconsistency in the input data to make our forecasting system more correct. In this study, the MinMaxScaler technique [29] is adopted to scale and transform the value of data in to [0, 1] interval. This may enhance the model's performance and rate of convergence. In (1) describe the normalization function,

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

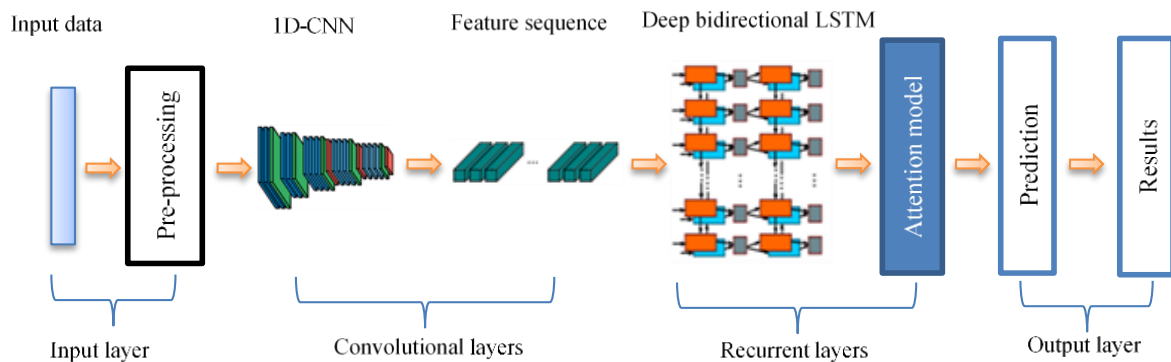


Figure 1. The proposed approach for the SPF based on CRNN and attention mechanism

2.1. Convolutional neural network for feature extraction

To extract features from the input, a CNN model is utilized [30], which includes layers with max-pooling, batch normalization, and rectified linear unit (ReLU) activation [31], [32]. Common two-dimensional (2D) CNNs have been effectively used in different applications, including text recognition, image classification, and speech processing [30], [31]. Given each 1D input data x of size n and 1D convolution kernel w size of k , the non-causal convolution between x and w for stride s in the 1D CNN is calculated as (2),

$$y(n) = \begin{cases} \sum_{i=0}^k x(n+i)w(i), & \text{if } n = 0 \\ \sum_{i=0}^k x(n+i+(s-1))w(i), & \text{otherwise} \end{cases} \quad (2)$$

In our study, there is no padding at all. We calculate 1D CNN by using Keras framework. The length of the output o for stride s is defined as (3),

$$o = \left\lfloor \frac{(n-k)}{s} \right\rfloor + 1 \quad (3)$$

2.2. Recurrent layer for prediction

The convolutional layer in this study is followed by a recurrent layer that forecasts stock price using RNN. The RNN can capture the information of previous and next elements. Using adjacent information in the feature sequence will help the RNN to "capture the context" of the data. Specifically, we use LSTM [6] to produce the prediction of feature sequence or series from CNN layer. The LSTM is RNN extension that lessens the impact of the problem with vanishing gradients. Significant contextual information in a sequence or series can be understood by the LSTM model. Additionally, the LSTM network is likely to gather and record data for an output sequence based on previous and upcoming circumstances. Additionally, an end-to-end network is created by combining the LSTM and convolutional layers. As a result, the LSTM works effectively and is stable for time series data of any length. The LSTM models pick up on long input dependencies.

The essential components of LSTM architecture are a memory cell, three gates—an input gate, an output gate, and a forget gate. Both the past and the future have value, and both are recorded in the memory cell. The cell can store and retrieve data for extended periods of time. The input gate determines the amount of additional information that will be added to the cell. The output gate regulates how much the value of the memory cell is used to determine the output. The cell's memory is cleared using the forget gate, which is also used to determine which value is still present in the cell. It chooses which data is erased from memory. As a result, the LSTM has a propensity to detect long-term dependency, which frequently appears in time-series. Regarding LSTM in this case, combining both of past and futures contexts can produce better outputs and, thereby, complement better information to the adjacent elements. To this end, we defined two LSTMs in two opposite directions, one forward and one backward, and combined them into a single component, named Bi-LSTM, to exploit information of the input feature sequence or series in both directions [30], [31].

2.3. Attention mechanism for stock price forecasting

In this study, we apply the attention-based approach [29], [30], [32] incorporated into the CRNN model to enhance the performance of the SPF system. The advantage of an attention mechanism lies in its ability to enable the network to dynamically weigh the importance of different time steps in the stock time series data. This helps focus more on relevant information, improving forecasting accuracy by assigning greater weight to important patterns. Moreover, the CRNN combined with an attention mechanism can adapt to varying lengths of historical data, enabling forecasting models to be more flexible and accommodating for different time series lengths. The Attention-based approach does not use conditional independence constraints. Instead, the Attention mechanism directly estimates the posterior probability of the output sequence C , $p(C|y)$ based on (4),

$$p(C|y) = \prod_l p(c_l | c_1, c_2, \dots, c_{l-1}, y) \quad (4)$$

where $p(c_l | c_1, c_2, \dots, c_{l-1}, y)$ is estimated by (5)-(8):

$$h = \text{Encoder}(y) \quad (5)$$

$$a_t = \text{Attention}(\{a_{l-1}\}, q_{l-1}, h_t) \quad (6)$$

$$r_t = \sum_t a_{lt} h_t \quad (7)$$

$$p(c_l | c_1, c_2, \dots, c_{l-1}, y) = \text{Decoder}(r_l, q_{l-1} c_{l-1}) \quad (8)$$

The encoder network transforms the input feature y_t into h_t values, where h_t is the hidden vector. Convolutional features are used to implement a content-based attention mechanism. The context vector r_t for the decoder is computed using the attention weight, a_{lt} , which denotes the soft alignment of h_t for each output c_l . The encoded string will produce output for the decoder network. Finally, the output stock price is calculated using the whole connection layer.

Utilizing the Keras framework on Google Colab, we implement our SPF model into practice. The summary of network configuration for SPF is shown in Table 1. When using the backpropagation algorithm to train our network, we utilize the Adam optimizer, a batch size of 32, and 150 epochs. The learning rate was set at 1e-3. The mean squared error (MSE) loss, or mean square error loss, is used to calculate the loss function. The proposed CRNN's model structure with attention mechanism for SPF is shown in Figure 2.

Table 1. The network configuration for SPF

Hyperparameters	Value
Input time-step	10
Number of features of the input dimension	1
Number of filters of 1D Conv layer	256
Kernel size of Conv layer	1
Activation function of Conv layer	ReLU
Bi-LSTM time-step	256
Number of features per time-step	128
Activation function of Bi-LSTM layer	ReLU
Dropout rate	0.2
Number of units of dense layers	128, 64, 1

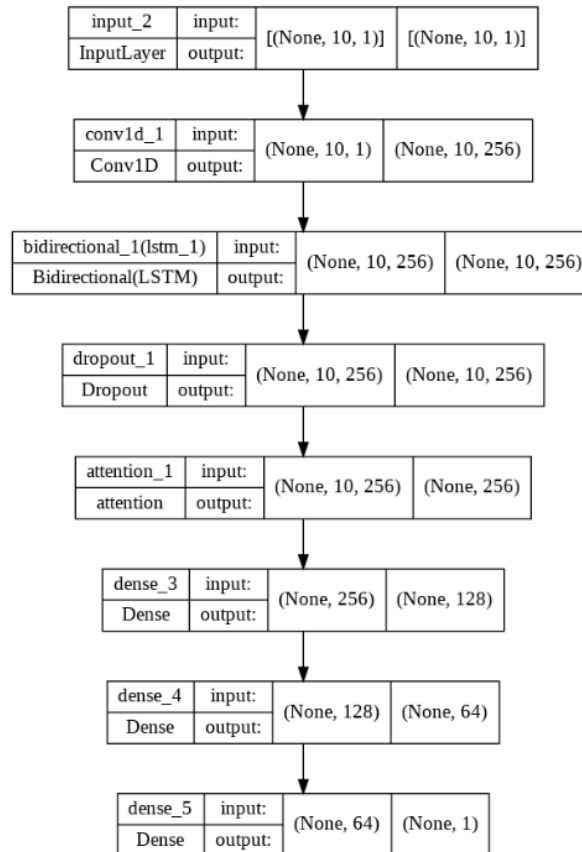


Figure 2. The proposed CRNN with attention mechanism using Keras framework

3. RESULTS AND DISCUSSION

3.1. Dataset

In the research, we evaluate the effectiveness of the proposed approach for SPF using a dataset of 10 stock prices obtained from Yahoo Finance (<https://finance.yahoo.com/>), including companies like Apple (AAPL), Amazon (AMZN), Nvidia (NVDA), Tesla (TSLA), Netflix (NFLX), Tencent (TCEHY), Microsoft (MSFT), Intel (INTC), Alphabet Inc (GOOG), and Baidu (BIDU). We collect and use information from 2012-01-03 to 2020-12-24. There are 2261 observations in all. A record of stock data includes six numbers for each trading day of the stock, including the open price, closing price, highest price, lowest price, adjusted close, and volume.

Figure 3 presents an example of a candlestick chart of the historic daily stock price of the AAPL dataset from 2020-09-01 to 2020-12-24. In this figure, the vertical axis shows the stock price, the horizontal axis shows trading dates, the green bar stands for the closing price was higher than the open price and the red bar stands for the opening price was higher than the closing price. In the figure, we realize that the noise occurred during data collection which is represented by the little candles. While these little candles might seem like noise, some traders specifically focus on these intraday movements to make short-term trades or capture quick profits. However, for longer-term investors, these fluctuations might be more of a distraction from the overall trend. So, the SPF is a complex problem.

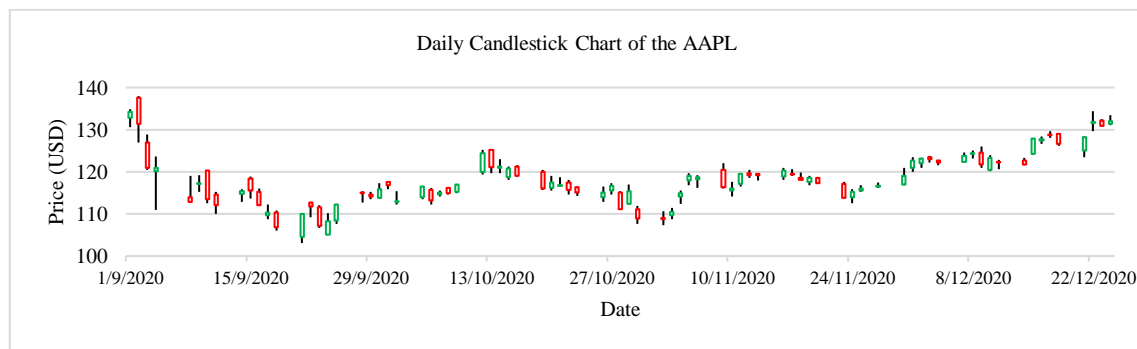


Figure 3. Candlestick chart historic daily stock price of the AAPL dataset from 2020-09-01 to 2020-12-24

The closing price of stock data refers to the last price at which a stock trades during a regular trading session. This amount is frequently used to calculate if a profit or loss is made. Furthermore, it supplies especially valuable information for investors to use to assess changes in stock prices over time. So, as a prediction objective in the study, we use the closing price of stock data. The proposed approach is implemented to forecast the stock data's closing price.

Figure 4 shows the daily closing price of AAPL stock dataset and Figure 5 illustrates the daily closing price of AMZN stock dataset. In our analysis of market trends, we pay special attention to the performance of leading tech companies. These figures illustrate the fluctuations in the stock prices of AAPL and AMZN over the selected time. They highlight key trends and anomalies that are important for both investors and analysts to consider. From these figures, we realize that stock data is high-dimensional nonlinear time series with large fluctuations. Otherwise, the closing price had an increase trend. In particular, the price gradually increased from 2012 to the end of 2018 and then showed a slight decline in the beginning of 2019. Noticeably, there was a sharp growth of the stock price from 2020 to 2021. These discrepancies provide compelling evidence that developing accurate high-performance stock price predictions is a challenging task.

We divided the stock price dataset into training and test data for our research. The first 80% is used for training, and the final 20% is used for testing. Particularly, 1808 samples are used for training and 453 samples are used for testing. We use 361 samples from the training set (or about 20% of the training data) for the validation set for building the training network.

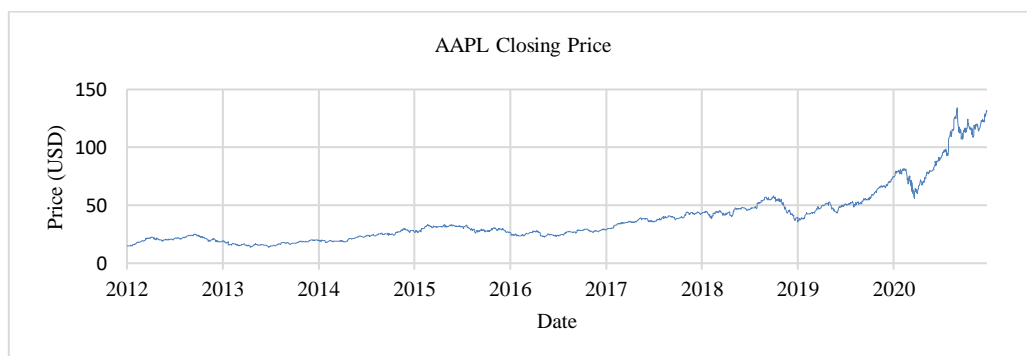


Figure 4. The daily closing of AAPL stock price dataset from 2012-01-03 to 2020-12-24



Figure 5. The daily closing of AMZN stock price dataset from 2012-01-03 to 2020-12-24

3.2. Parameters analysis

In time series analysis based on neural network, the time step is one of important parameters which affects the performance of the forecasting system. Selecting a suitable time step for SPF is quite a challenging task. Too small of a time step can lead to large deviation of the results and exceptionally long computation times. This will neglect the influence of global factors. In contrast, too large of a time step can lead to high fluctuation which reduces the accuracy of prediction results. We base our experimentation and findings [29] on a time step of 10. To be more precise, we train our suggested CRNN model using the closing price from the eleventh day as the label and the stock data from the previous 10 days as the input layer to our CNN layer. To anticipate the closing price of the following day, the closing prices from the previous 10 days are used as an input.

Based on the experiments, the most suitable hyperparameters are as follows: the output of the CNN layer is set at 256. Additionally, the Bi-LSTM utilized a time-step of 256 with 128 features per time-step (number of hidden units), serving as a component of the RNN layer. The dropout is used to reduce the overfitting. The dropout layer's parameter is 0.2. To enhance the performance, we added the attention mechanism. The three dense layers that are fully connected are utilized and have, respectively, 128, 64, and 1 neuron as shown in Table 1 and Figure 2. ReLU function serves as the activation function for each layer. The SPF system's accuracy is lastly evaluated using the metrics of mean squared error (MSE), root mean square error (RMSE), mean absolute percentage error (MAPE), and mean absolute error (MAE). The most precise system often has the lowest values.

3.3. Experiment results and discussion

This study primarily focuses on comparing the effectiveness of deep learning approaches with baseline classical algorithms in forecasting market prices. We also investigate the effects of the attention mechanism and compare the proposed approach to other approaches. Various methods such as ARIMA [6], [8], LSTM [18], CNN+LSTM [26], [27], CNN+Bi-LSTM [27], [28] and proposed CRNN+attention method are compared across ten stock price datasets. The CNN+LSTM and CNN+Bi-LSTM models are combinations of the CNN and LSTM and the CNN and Bi-LSTM models, respectively. The CRNN with attention mechanism is described by the proposed CRNN+attention. Our experimental results found the limitations of the ARIMA model in handling non-linear and non-stationary data, a common characteristic in financial time series.

Table 2 compares the performance of various approaches in forecasting stock prices using different metrics like MSE, RMSE, MAE, and MAPE across different ten stock datasets such as AAPL, AMZN, NVDA,

TSLA, NFLX, TCEHY, MSFT, INTC, GOOG, and BIDU. The ARIMA performs reasonably well but is outperformed by more complex models like LSTM, CNN+LSTM, CNN+Bi-LSTM, and CRNN+attention in most cases. The performance of the CRNN model, which combines the CNN with LSTM/Bi-LSTM architectures, obtains a significant improvement in prediction accuracy. The ARIMA model is not suitable for nonlinear and non-stationary data, limiting their applicability as financial time series frequently exhibit nonlinear behaviors [8]. Furthermore, determining the correct differencing order and the appropriate number of AR and MA terms (the p , d , q parameters) can be challenging and requires expertise [6]. The results claimed that the deep learning techniques consistently outperform the traditional ARIMA model [19], [23]. These models are capable of handling the inherent nonlinear and non-smooth characteristics of stock price data [22]. Furthermore, we've observed that the combined CNN+LSTM/Bi-LSTM (CRNN model) can significantly improve prediction results compared to LSTM alone. The CRNN consistently performs well across various datasets, often achieving lower error metrics compared to the LSTM model, although it's not consistently the best model. The CRNN-based approach led to better capture of temporal and spatial dependencies in the data.

Table 2. Performance comparison of our approach and other approaches on various stock price datasets

Metrics	Model				
	ARIMA	LSTM	CNN+LSTM	CNN+Bi-LSTM	CRNN+attention (proposed)
AAPL stock dataset					
MSE	0.00414125	0.00096229	0.00111956	0.0016502	0.00058514
RMSE	0.06435258	0.03102075	0.03345975	0.04062271	0.0241897
MAE	0.04488899	0.02471271	0.02327488	0.02995746	0.01705691
MAPE	4.48889914	2.47127130	2.32748838	2.99574607	1.70569113
AMZN stock dataset					
MSE	0.00144367	0.00061734	0.00140791	0.00133806	0.0006067
RMSE	0.03799568	0.02484636	0.03752215	0.03657946	0.02463131
MAE	0.02635121	0.01873357	0.02704039	0.02604606	0.01702218
MAPE	2.63512148	1.87335703	2.70403948	2.60460576	1.70221833
NVDA stock dataset					
MSE	0.00209091	0.0011401	0.00275545	0.00506704	0.00463466
RMSE	0.0457265	0.03376543	0.05249234	0.07118313	0.0680783
MAE	0.03224868	0.02275946	0.03693821	0.04828172	0.04469303
MAPE	3.22486794	2.27594588	3.69382138	4.82817243	4.46930276
TSLA stock dataset					
MSE	0.05500308	0.00056591	0.00107664	0.0004636	0.00047298
RMSE	0.23452736	0.02378885	0.0328122	0.02153127	0.02174803
MAE	0.13197235	0.01338222	0.02395707	0.01635219	0.01412555
MAPE	1.319723453	1.33822189	2.39570686	1.63521933	1.41255506
NFLX stock dataset					
MSE	0.00121343	0.00169071	0.00151416	0.00128446	0.00140912
RMSE	0.0348343	0.04111822	0.03891219	0.0358393	0.03753826
MAE	0.02517076	0.02913156	0.02827703	0.02595907	0.02761907
MAPE	2.51707564	2.91315622	2.82770265	2.59590731	2.76190669
TCEHY stock dataset					
MSE	0.00127932	0.00119615	0.00123172	0.00105299	0.0011031
RMSE	0.03576755	0.03458537	0.03509587	0.03244977	0.03321294
MAE	0.02774635	0.02503279	0.02634682	0.02417637	0.02484925
MAPE	2.77463492	2.50327876	2.63468177	2.41763666	2.48492500
MSFT stock dataset					
MSE	0.00294903	0.00148967	0.00113022	0.00105081	0.0010748
RMSE	0.05430499	0.03859618	0.03361875	0.03241627	0.03278418
MAE	0.03733583	0.02689209	0.0233195	0.02230774	0.02216237
MAPE	3.73358330	2.68920942	2.33194993	2.23077434	2.21623747
INTC stock dataset					
MSE	0.00465329	0.00386873	0.00474968	0.00373726	0.00503339
RMSE	0.068215	0.06219909	0.06891793	0.0611331	0.07094637
MAE	0.04693407	0.03912231	0.04444593	0.03814838	0.0422491
MAPE	4.69340699	3.91223136	4.44459287	3.81483810	4.22491005
GOOG stock dataset					
MSE	0.00157086	0.00153489	0.00137259	0.0012681	0.00131233
RMSE	0.03963404	0.03917764	0.03704849	0.03561035	0.03622606
MAE	0.02834307	0.02778249	0.0272164	0.02503106	0.02609333
MAPE	2.83430702	2.77824910	2.72163980	2.50310562	2.60933311
BIDU stock dataset					
MSE	0.00069584	0.00122399	0.0014525	0.0011243	0.00121801
RMSE	0.02637884	0.03498551	0.03811165	0.03353057	0.03490006
MAE	0.01815237	0.0225333	0.02542847	0.02307138	0.02359329
MAPE	1.81523716	2.25332959	2.54284657	2.30713825	2.35932860

Upon comparing the proposed approach with other methods, it becomes evident that the combined CNN and RNN models yield promising results [26]. Additionally, CRNN+attention consistently performs well across various datasets, frequently yielding the lowest MSE, RMSE, MAE, and MAPE values. It shows robustness and better accuracy over other models in most cases. Regarding long-term time-series prediction, the incorporation of attention mechanisms helps the model in focusing on relevant information, potentially enhancing its forecasting capability [28]. The performance of SPF systems can be considerably improved by using our technique, which uses the trend in stock price as a time series forecasting metric. We realize that the effectiveness of models varies across different stocks, indicating that certain architectures might better capture unique characteristics of specific stocks.

Figures 6 and 7 display examples of the comparison of forecast outcomes for several techniques on the AAPL and AMZN stock datasets, respectively. The red curve and blue curve in the illustration represent, respectively, the prediction value and ground truth the closing price. The stock price is shown on the vertical axis, and trading dates are shown on the horizontal axis. Figure 6(a) and Figure 6(b) show the performance of ARIMA and LSTM models on the AAPL stock dataset, respectively. Figure 6(c) and Figure 6(d) show the performance of CNN+LSTM and CNN_Bi-LSTM models on the AAPL stock dataset, respectively. Figure 6(e) illustrates the performance of the proposed model on the AAPL stock dataset. Figure 7(a) and Figure 7(b) show the performance of ARIMA and LSTM models on the AMZN stock dataset, respectively. Figure 7(c) and Figure 7(d) show the performance of CNN+LSTM and CNN+Bi-LSTM models on the AMZN stock dataset, respectively. Figure 7(e) illustrates the performance of the proposed model on the AMZN stock dataset.

In these figures, the predicted closing prices obtained for all methods were close to the target values. Therefore, all these methods are suitable for yielding high forecasting accuracy. First, we show that model-based deep learning algorithms as shown in Figures 6(b) to 6(e) and Figures 7(b) to 7(e) perform better than the standard ARIMA model. The results of the LSTM or Bi-LSTM and the proposed method are compared to the ARIMA model as clear for enhancing the accuracy. Because the Bi-LSTM can identify both the past and future contexts of the input stock price data and because it is helpful for learning the past and future contextual information, it is also more accurate than the LSTM [23]. In addition, we apply the CNN layer for investigating local stock price feature learning and obtaining nonlinear information of the stock data. The performance is like that of the LSTM techniques when CRNN are applied. Finally, the proposed method using attention mechanism as shown in Figure 6(e) and Figure 7(e) often achieves the best prediction results in comparison to the others.

Figure 8 presents the performance metrics in terms of MAPE value of CRNN-based approach for SPF for various companies. The models mentioned are CRNN+attention, CRNN (LSTM), and CRNN (Bi-LSTM), which are variations of RNNs with different architectures. The proposed model consistently outperforms other models across various datasets, suggesting its potential as a robust forecasting model for diverse stock behaviors. These findings suggest that selecting the appropriate neural network architecture for each specific stock might be important for SPF. Moreover, combining the predictions of different models or using ensemble techniques might offer potential enhancements in forecasting accuracy across various stocks [33].

Although the proposed method yields promising performance, it still has some limitations. In our study, we utilized a predefined model configuration to train across varied datasets. However, stock data often exhibit significant variations in characteristics like volatility, liquidity, and market dynamics. This variability requires different parameter configurations for each dataset, making it a challenge to find an optimal model configuration that is effective across all types of stock data. Furthermore, finding the optimal hyperparameters for each dataset is a complex and time-consuming task [34]. This complexity arises from the high dimensionality of the parameter space, which significantly impacts the model's performance.

Time consumption is a secondary limitation in this study. Table 3 illustrates a comparison of various methods in average time computation cost. The ARIMA consumes 19.1 seconds. It was lower computation cost than deep learning-based approaches and did not require graphics processing unit (GPU) computing. It is efficient for linear relationships but may struggle with complex patterns. The other deep learning models consume more time due to capturing long-term dependencies or spatial and temporal patterns simultaneously, leading to higher computational complexity [35]. The proposed method consumes 506 seconds. The time consumption of deep learning models, particularly the proposed method, presents a challenge for real-time applications. Upgrading hardware offers a short-term fix, but further research is necessary for a long-term, sustainable solution. Finally, the choice of forecasting method involves a trade-off between accuracy and computational efficiency. More complex models like CRNN and attention-based models might offer improved predictive capabilities to capture complex relationships and patterns in stock price data but at the expense of significantly higher computational costs compared to simpler models like ARIMA [36]. Finding the right balance between accuracy and computational resources is important when selecting a method for SPF.

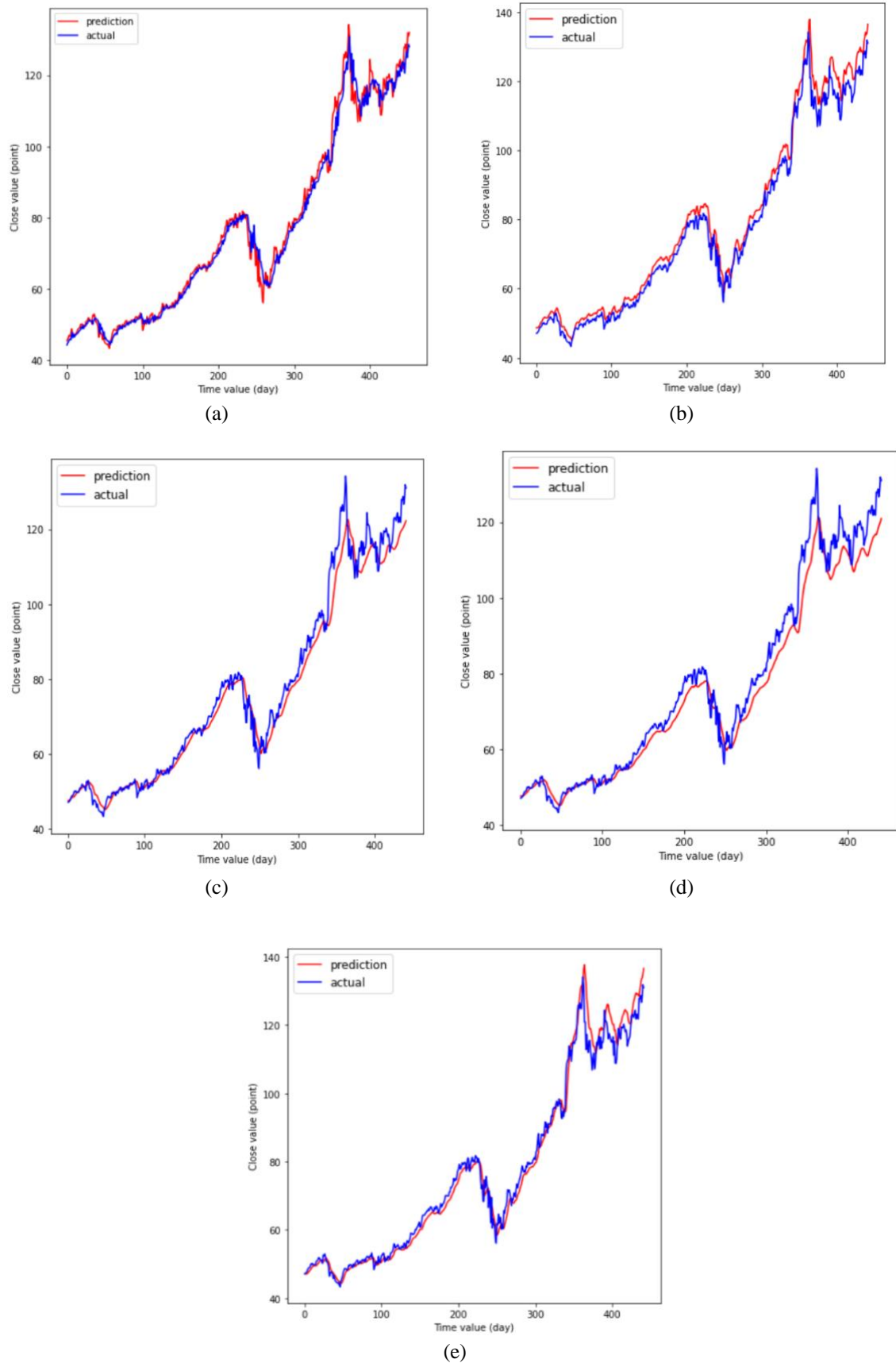


Figure 6. The results of various methods on AAPL stock price dataset: (a) ARIMA, (b) LSTM, (c) CNN+LSTM, (d) CNN+Bi-LSTM, and (e) proposed CRNN+attention

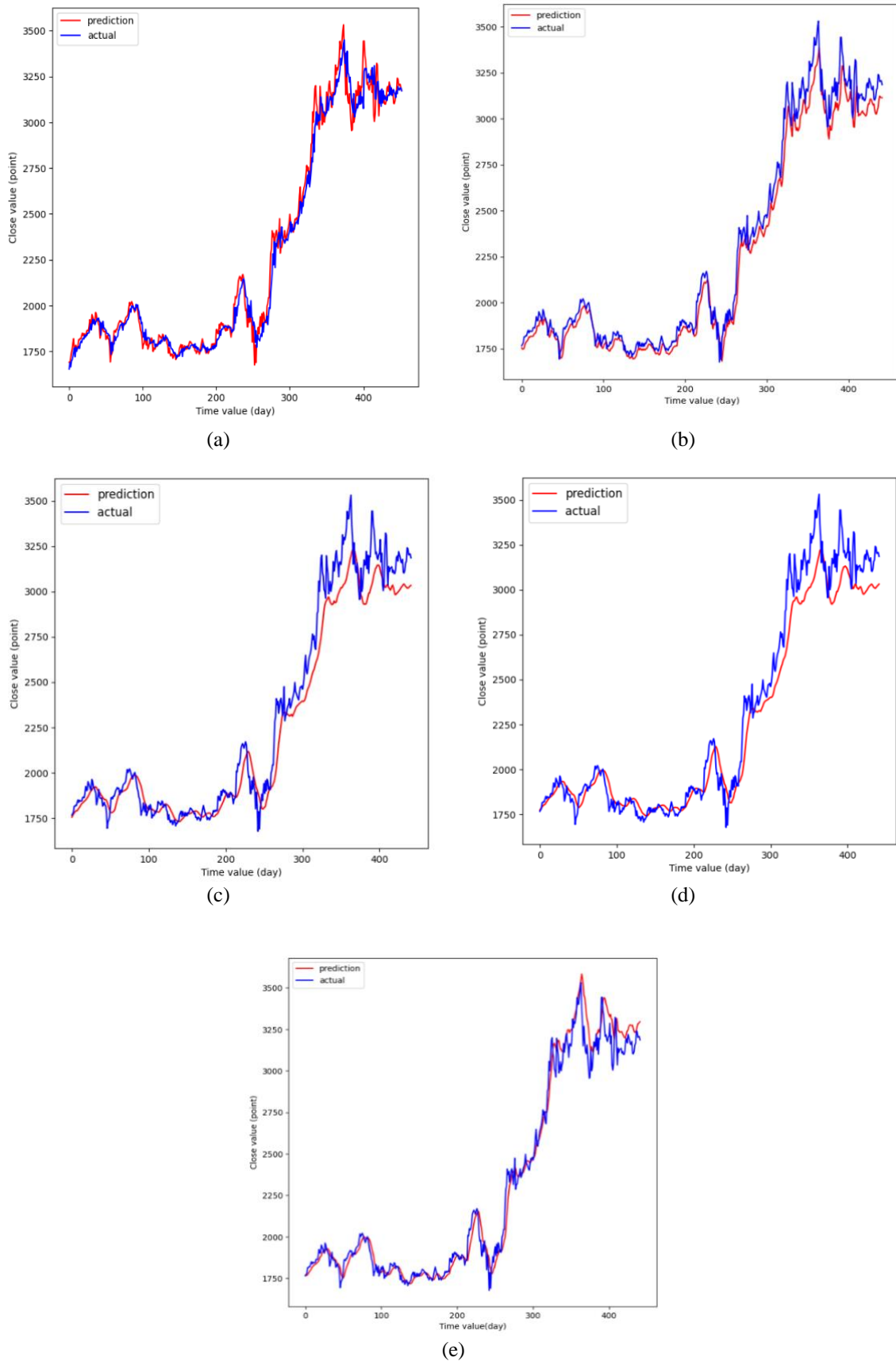


Figure 7. The results of various methods on AMZN stock price dataset: (a) ARIMA, (b) LSTM, (c) CNN+LSTM, (d) CNN+Bi-LSTM, and (e) proposed CRNN+attention

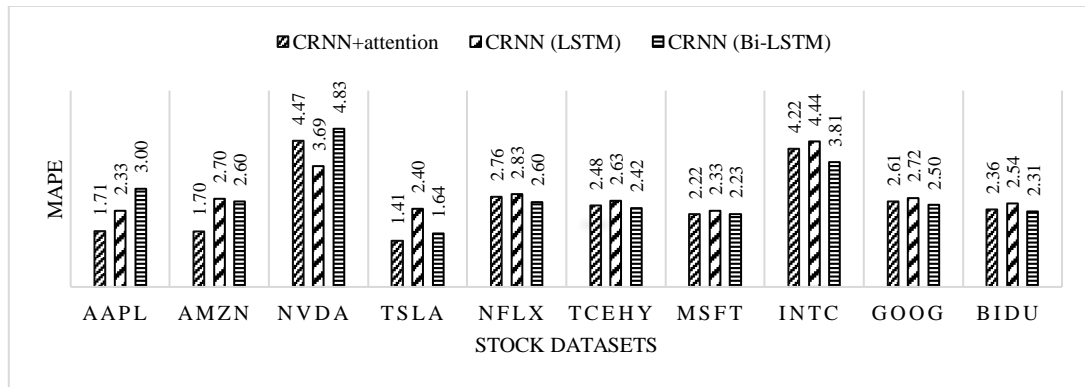


Figure 8. The comparison of our method and CRNN approaches on various stock datasets

Table 3. Comparison of various methods in average time computation cost

	ARIMA	LSTM	CRNN (LSTM)	CRNN (Bi-LSTM)	CRNN+attention
Time consumed	19.1s	322s	349.4s	502s	506s

4. CONCLUSION

By fusing the advantages of CNN, Bi-LSTM, and attention mechanism, we have proposed an improved CRNN model for SPF. First, we described how to use 1D CNN as a feature extraction approach to build important features from a high-dimensional dataset. Second, the features are fed into the Bi-LSTM and attention models to assess how well the forecasting system is performing. The experiment results confirmed that our method can outperform baseline approaches and significantly enhances the accuracy of forecasting system. In addition, the balance between accuracy and computational efficiency remains an important consideration in selecting appropriate models for SPF. In the future, we will explore new hybrid models that integrate different neural network architectures, including multi-head attention and transformer models, potentially leading to enhanced performance of systems. Additionally, we intend to explore the effects of ensemble models to achieve more accurate and robust predictions, and we will also investigate the development of optimal lightweight models suitable for real-time applications.




REFERENCES

- [1] Srivinay, B. C. Manujakshi, M. G. Kabadi, and N. Naik, "A hybrid stock price prediction model based on pre and deep neural network," *Data*, vol. 7, no. 5, pp. 1–11, 2022, doi: 10.3390/data7050051.
- [2] N. Rouf *et al.*, "Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions," *Electronics*, vol. 10, no. 21, pp. 1–25, 2021, doi: 10.3390/electronics10212717.
- [3] Z. Hu, Y. Zhao, and M. Khushi, "A survey of forex and stock price prediction using deep learning," *Applied System Innovation*, vol. 4, no. 1, pp. 1–30, 2022, doi: 10.3390/asi4010009.
- [4] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *Journal of Supercomputing*, vol. 76, no. 3, pp. 2098–2118, 2020, doi: 10.1007/s11227-017-2228-y.
- [5] J. Chen, S. Yang, D. Zhang, and Y. A. Nanehkar, "A turning point prediction method of stock price based on RVFL-GMDH and chaotic time series analysis," *Knowledge and Information Systems*, vol. 63, no. 10, pp. 2693–2718, 2021, doi: 10.1007/s10115-021-01602-3.
- [6] P. H. Vuong, T. T. Dat, T. K. Mai, P. H. Uyen, and P. T. Bao, "Stock-price forecasting based on XGBoost and LSTM," *Computer Systems Science and Engineering*, vol. 40, no. 1, pp. 237–246, 2022, doi: 10.32604/CSSE.2022.017685.
- [7] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021, doi: 10.1016/j.ijforecast.2020.06.008.
- [8] M. R. Islam and N. Nguyen, "Comparison of financial models for stock price prediction," *Journal of Risk and Financial Management*, vol. 13, no. 8, pp. 1–19, 2020, doi: 10.3390/jrfm13080181.
- [9] D. S. Kumar, T. B. C., S. N. R. C., V. A., A. S. Devi, and D. Kavitha, "Analysis and prediction of stock price using hybridization of SARIMA and XGBoost," in *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 2022, pp. 1–4, doi: 10.1109/IC3IoT53935.2022.9767868.
- [10] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock closing price prediction using machine learning techniques," *Procedia Computer Science*, vol. 167, pp. 599–606, 2020, doi: 10.1016/j.procs.2020.03.326.
- [11] Q. Yunneng, "A new stock price prediction model based on improved KNN," in *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, 2020, pp. 77–80, doi: 10.1109/ICISCE50968.2020.00026.
- [12] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, "A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting," *Annals of Data Science*, vol. 10, no. 1, pp. 183–208, 2023, doi: 10.1007/s40745-021-00344-x.
- [13] P. K. Illa, B. Parvathala, and A. K. Sharma, "Stock price prediction methodology using random forest algorithm and support vector




- machine,” *Materials Today: Proceedings*, vol. 56, pp. 1776–1782, 2022, doi: 10.1016/j.matpr.2021.10.460.
- [14] R. Chandra and Y. He, “Bayesian neural networks for stock price forecasting before and during COVID-19 pandemic,” *PLoS ONE*, vol. 16, pp. 1–32, 2021, doi: 10.1371/journal.pone.0253217.
- [15] V. Khanna, J. Joshua, and R. Pramila, “Implementation of supervised pre-training methods for univariate time series forecasting,” in *2023 2nd International Conference for Innovation in Technology (INOCON)*, 2023, pp. 1–8, doi: 10.1109/INOCON57975.2023.10101327.
- [16] Z. D. Akşehir and E. Kiliç, “How to handle data imbalance and feature selection problems in CNN-based stock price forecasting,” *IEEE Access*, vol. 10, pp. 31297–31305, 2022, doi: 10.1109/ACCESS.2022.3160797.
- [17] E. Hoseinzade and S. Haratizadeh, “CNNpred: CNN-based stock market prediction using a diverse set of variables,” *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019, doi: 10.1016/j.eswa.2019.03.029.
- [18] Q. Jiang, C. Tang, C. Chen, X. Wang, and Q. Huang, “Stock price forecast based on LSTM neural network,” in *Proceedings of the Twelfth International Conference on Management Science and Engineering Management*, 2019, pp. 393–408, doi: 10.1007/978-3-319-93351-1_32.
- [19] P. Ghosh, A. Neufeld, and J. K. Sahoo, “Forecasting directional movements of stock prices for intraday trading using LSTM and random forests,” *Finance Research Letters*, vol. 46, 2022, doi: 10.1016/j.frl.2021.102280.
- [20] Y. Gao, R. Wang, and E. Zhou, “Stock prediction based on optimized LSTM and GRU models,” *Scientific Programming*, vol. 2021, pp. 1–8, 2021, doi: 10.1155/2021/4055281.
- [21] Y. Xu, L. Chhim, B. Zheng, and Y. Nojima, “Stacked deep learning structure with bidirectional long-short term memory for stock market prediction,” in *Neural Computing for Advanced Applications*, 2020, pp. 447–460, doi: 10.1007/978-981-15-7670-6_37.
- [22] B. Liu, Z. Yu, Q. Wang, P. Du, and X. Zhang, “Prediction of SSE Shanghai Enterprises index based on bidirectional LSTM model of air pollutants,” *Expert Systems with Applications*, vol. 204, pp. 1–9, 2022, doi: 10.1016/j.eswa.2022.117600.
- [23] S. Mootha, S. Sridhar, R. Seetharaman, and S. Chitrakala, “Stock price prediction using bi-directional LSTM based sequence to sequence modeling and multitask learning,” in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2020, pp. 78–86, doi: 10.1109/UEMCON51285.2020.9298066.
- [24] N. B. Korade and M. Zuber, “Stock price forecasting using convolutional neural networks and optimization techniques,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 11, pp. 378–385, 2022, doi: 10.14569/IJACSA.2022.0131142.
- [25] P. Lv, Q. Wu, J. Xu, and Y. Shu, “Stock index prediction based on time series decomposition and hybrid model,” *Entropy*, vol. 24, no. 2, pp. 1–18, 2022, doi: 10.3390/e24020146.
- [26] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, “A CNN-LSTM-based model to forecast stock prices,” *Complexity*, vol. 2020, pp. 1–10, 2020, doi: 10.1155/2020/6622927.
- [27] H. Wang, J. Wang, L. Cao, Y. Li, Q. Sun, and J. Wang, “A stock closing price prediction model based on CNN-BiLSTM,” *Complexity*, vol. 2021, pp. 1–12, 2021, doi: 10.1155/2021/5360828.
- [28] W. Lu, J. Li, J. Wang, and L. Qin, “A CNN-BiLSTM-AM method for stock price prediction,” *Neural Computing and Applications*, vol. 33, no. 10, pp. 4741–4753, 2021, doi: 10.1007/s00521-020-05532-z.
- [29] Y. Chen *et al.*, “Stock price forecast based on CNN-BiLSTM-ECA model,” *Scientific Programming*, vol. 2021, pp. 1–20, 2021, doi: 10.1155/2021/2446543.
- [30] T. T. Dat, L. T. A. Dang, V. N. T. Sang, L. N. L. Thuy, and P. T. Bao, “Convolutional recurrent neural network with attention for Vietnamese speech to text problem in the operating room,” *International Journal of Intelligent Information and Database Systems*, vol. 14, no. 3, pp. 294–314, 2021, doi: 10.1504/IJIDS.2021.116476.
- [31] T. T. Dat *et al.*, “An improved crnn for Vietnamese identity card information recognition,” *Computer Systems Science and Engineering*, vol. 40, no. 2, pp. 539–555, 2021, doi: 10.32604/CSSE.2022.019064.
- [32] P. T. Bao, L. T. A. Dang, N. D. Tam, N. N. Truong, P. C. L. T. Vu, and T. T. Dat, “An application of Vietnamese handwriting text recognition for information extraction from high school admission form,” *IAES International Journal of Artificial Intelligence*, vol. 12, no. 2, pp. 568–576, 2023, doi: 10.11591/ijai.v12.i2.pp568-576.
- [33] M. Jiang, J. Liu, L. Zhang, and C. Liu, “An improved stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms,” *Physica A: Statistical Mechanics and its Applications*, vol. 541, pp. 1–16, 2020, doi: 10.1016/j.physa.2019.122272.
- [34] B. Gülmez, “Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm,” *Expert Systems with Applications*, vol. 227, pp. 1–16, 2023, doi: 10.1016/j.eswa.2023.120346.
- [35] A. Kanwal, M. F. Lau, S. P. H. Ng, K. Y. Sim, and S. Chandrasekaran, “BiCuDNNLSTM-1dCNN — a hybrid deep learning-based predictive model for stock price prediction,” *Expert Systems with Applications*, vol. 202, pp. 1–15, 2022, doi: 10.1016/j.eswa.2022.117123.
- [36] T. B. Shahi, A. Shrestha, A. Neupane, and W. Guo, “Stock price forecasting with deep learning: a comparative study,” *Mathematics*, vol. 8, no. 9, pp. 1–15, 2020, doi: 10.3390/math8091441.

BIOGRAPHIES OF AUTHORS






Pham Hoang Vuong    received a B.Sc. degree in Mathematics and Computer Sciences from University of Education – National University of HCM City, Vietnam in 1992. He also received engineering degree from Hanoi University of Science and Technology in 2000 and also received master degree from Ho Chi Minh City University of Science in 2001. He is currently a lecturer at Department of Computer Science, Sai Gon University, Vietnam since 2008. His research areas of interest include stock-price forecasting and computer vision based on deep-learning. He can be contacted at email: phvuong@sgu.edu.vn.






Lam Hung Phu    received a bachelor's degree in information technology from Sai Gon University of HCM City, Vietnam in 2023. He is currently a fresh graduate since 2023. His research areas of interest include predictive modeling, computer vision, and data analysis. He can be contacted at email: pth.a3.30.hungphu@gmail.com.






Le Nhat Duy    received Ph.D. degree in Computer Science from Moscow State Pedagogical University, Russia in 2013. He is currently a lecturer at Department of Computer Science, Industrial University of Ho Chi Minh City, Vietnam since 2013. His research areas of interest include computational intelligence and cryptography. He can be contacted at email: lenhatduy@iuh.edu.vn.



Pham The Bao    received his B.Sc. degree in Algebra from University of Natural Science – National University of HCM City, Vietnam in 1995. He also received M.Sc. degree in Mathematical Foundation of Computer Science and Ph.D. degree in Computer Science from University of Natural Science – National University of HCM City, Vietnam in 2000 and 2009, respectively. He was a lecturer and professor in Department of Computer Science, Faculty of Mathematics Computer Science, University of Natural Science, Vietnam from 1995 to 2018. He is currently dean and professor at Department of Computer Science, Sai Gon University, Vietnam since 2019. He has published over 50 papers in international journals and conferences. His research includes image processing, pattern recognition, and intelligent computing. He can be contacted at email: ptbao@sgu.edu.vn.



Tan Dat Trinh    received a B.Sc. degree in Mathematics and Computer Sciences from University of Natural Science – National University of HCM City, Vietnam in 2010. He also received Master of Engineering (M.Eng.) and Ph.D. degree in Electronics and Computer Engineering from Chonnam National University, Korea in 2013 and 2017, respectively. He is currently a lecturer at Department of Computer Science, Sai Gon University, Vietnam since 2019. He is also an AI researcher at Computer Science Laboratory, Sai Gon University since 2019. His research areas of interest include speaker recognition, speech signal processing, computer vision, and pattern recognition. He can be contacted at email: trinhtandat@sgu.edu.vn.