International Journal of Innovative Computing, Information and Control
Volume **21**, Number **5**, October **2025**

ICIC International ©2025 ISSN 1349-4198
pp. 1401–1414

A NOVEL COMPACT 1D-CNN ARCHITECTURE FOR SHORT-TERM STOCK PRICE PREDICTION: A CASE STUDY ON APPLE STOCK PRICES

Nguyen Thi Huong Giang[1], Le Manh Thanh[2] and Cuong H. Nguyen-Dinh[3,*]

[1]Department of Informatics
University of Education
[2]Department of Information Technology
University of Sciences
Hue University
3 Le Loi Street, Hue 49000, Vietnam
{ nthgiang; lmthanh }@hueuni.edu.vn

[3]Faculty of Business and Technology
Phu Xuan University
176 Tran Phu Street, Hue 49000, Vietnam
*Corresponding author: cuong.nguyen@pxu.edu.vn

Received December 2024; revised April 2025

Abstract. *Short-term stock price prediction is a complex and challenging task due to the volatility and non-linearity of financial markets. This study presents a novel, compact One-Dimensional Convolutional Neural Network (1D-CNN) architecture designed specifically for predicting short-term stock prices, with a focus on Apple Inc. (AAPL) stock. The proposed model leverages historical price data to efficiently capture temporal patterns while maintaining a simple and computationally efficient structure. Comprehensive experiments evaluate its performance against a conventional Long Short-Term Memory (LSTM) model and a hybrid LSTM-1D-CNN model using metrics such as MAE, MSE, RMSE, and* R^2 *score. The findings demonstrate that the compact 1D-CNN model delivers competitive accuracy, particularly in short-term scenarios, achieving* R^2 *scores of 0.9983 for 1-day and 0.9953 for 3-day ahead predictions on AAPL stock, outperforming LSTM and hybrid baselines.*

Keywords: 1D-CNN, LSTM, Hybrid model, Stock price prediction, Short-term price prediction

1. **Introduction.** Short-term stock price prediction is a highly complex and challenging task due to the dynamic, non-linear, and volatile nature of financial markets [1]. The difficulty arises from many factors influencing price movements, such as market sentiment, macroeconomic indicators, or unforeseen geopolitical events [2]. Traditional methods often struggle to capture the intricate dependencies within financial time series data, necessitating advanced techniques for improved forecasting [2]. Furthermore, while specifically modern deep learning models like LSTMs excel at capturing long-term dependencies, their sophisticated architecture might be overly complex or less efficient for effectively extracting the local, short-term features crucial for near-future stock price prediction. To address the challenges of complex designs and high computational costs in existing models, this study introduces a compact 1D-CNN model. Its simple structure and high performance make it a practical and efficient solution for short-term stock price forecasting.

DOI: 10.24507/ijicic.21.05.1401

This research specifically investigates the suitability of a compact 1D-CNN architecture as an alternative to more complex models like LSTMs, particularly for short-term predictions where capturing recent local patterns is crucial and computational efficiency is desirable. Furthermore, the study targets a comparison of the advantages of the compact 1D-CNN model with those of hybrid models that integrate 1D-CNN and other recurrent neural networks, such as LSTM. By doing so, this research seeks to highlight the specific contexts in which 1D-CNN models outperform or complement other approaches in financial time series forecasting.

To be more specific, this research contributes a compact 1D-CNN model, which is only composed of a 1D-convolutional layer, a max pooling layer and a fully connected layer. The performance of this proposed model is evaluated by three distinct configurations of predicting next 1-day prices, next 3-day prices and next 5-day prices. This model's performance is also compared with those of LSTM model and hybrid model of 1D-CNN and LSTM. The experimental results, which used historical trading time series data of Apple Inc. stock, confirmed the advantages of 1D-CNN model in short-term scenarios including predicting next 1-day prices and next 3-day prices.

The rest of this paper is structured as follows. Section 2 reviews state-of-the-art research in the field of stock price prediction using deep learning models. Section 3 details the architecture of the proposed 1D-CNN model, while Section 4 presents and discusses the experimental settings and results. Finally, Section 5 summarizes the key achievements and outlines directions for future research.

2. **Related Work.** Economists and computer scientists have long been interested in the challenges of stock market prediction. In several decades ago, the regression methods and machine learning models were dominated this research trend [2]. However, the recent emerging deep learning models have pushed this research domain to the new frontiers. Within the scope of this section, we focus solely on elaborating on the state-of-the-art studies of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), with an emphasis on 1D-CNN models, LSTM models, and hybrid models of the two. For a comprehensive overview of research in this domain, readers are encouraged to consult the following surveys for valuable information [1-3].

Convolutional Neural Network (CNN) models, particularly 1D-CNNs, have proven highly effective in handling time series data due to their ability to automatically extract hierarchical features, capture local patterns, and process sequential information with computational efficiency. Hoseinzade and Haratizadeh [4] introduced a CNN-based framework designed to leverage data from various financial markets, incorporating inter-market correlations to enhance feature extraction and predict market indices' next-day movements. While this approach demonstrated significant improvements in prediction performance, it emphasizes multi-market data and complex feature interactions. In contrast, our proposed compact 1D-CNN model focuses on short-term stock price prediction with a simpler structure, optimizing computational efficiency while maintaining high predictive accuracy.

Alhazbi et al. [5] utilized a CNN to predict stock direction in the Qatar Stock Exchange, addressing the challenges of higher volatility in emerging markets. By incorporating external factors such as the S&P index, Nikkei index, and oil prices alongside historical data, the model improved prediction accuracy by 10%. A framework called Deep Candlestick Predictor (DCP) was proposed to forecast price movements by analyzing candlestick charts instead of relying on numerical financial data [6]. The model combined chart decomposition, a CNN-autoencoder for feature extraction, and a 1D-CNN for prediction, demonstrating superior accuracy on daily prices from the Taiwan Exchange Capitalization Weighted Stock Index compared to traditional index-based models. This highlighted the potential of chart-based deep learning approaches for stock prediction tasks. While the former used a traditional CNN model, the latter presented a hybrid method. In our approach, the task of predicting stock prices only focuses on short-term prices and uses a compact 1D-CNN model.

In another effort, Chandar [7] proposed a robust stock trading model integrating Technical Indicators and a Convolutional Neural Network (TI-CNN) to address the limitations such as slow convergence and overfitting in traditional models. The approach involved converting feature vectors derived from ten technical indicators into images using Gramian Angular Field, which are then fed into the CNN. Tested on NASDAQ and NYSE data from 2009 to 2018, the model demonstrated superior prediction accuracy and F1 scores compared to earlier methods, highlighting its effectiveness for stock trading applications. Our approach differs from that of Chandar [7] by using only historical stock transactiondata.

In a similar approach, Yao et al. [8] introduced a hybrid model named MEMD-TCN combining Multivariate Empirical Mode Decomposition (MEMD) and Temporal Convolutional Network (TCN). The model decomposed multivariate time series data (COHLV) into subsequences of different frequencies, used TCN to predict future closing prices, and reconstructed the predictions for improved accuracy. Experimental results demonstrated the MEMD-TCN model's effectiveness, accuracy, and stability compared to traditional methods, highlighting the importance of time series decomposition. Following this hybrid trend, Chen et al. [9] presented the MKP-TemporalNet model composed of an attention mechanism to dynamically weigh feature importance with improved Temporal Convolutional Networks (iTCN) and Bidirectional Gated Recurrent Units (BiGRU) for enhanced accuracy. The iTCN introduced a multi-kernel parallel convolution structure to improve temporal feature extraction across different time scales, outperforming traditional methods. Experimental results confirmed the model's effectiveness in predicting new energy stock indexes compared to other machine learning approaches. In short, these hybrid methods require further resources like external data or preprocessing method, while the proposed method is simpler and saves computational costs.

As a member of CNN model family, 1D-CNN models are especially effective in handling time series data such as stock transaction data. For example, Liu and Si [10] proposed a novel approach using 1D-CNN model for classifying chart patterns in financial time series. The 1D-CNN model outperformed traditional methods such as support vector machines, extreme learning machines, and long short-term memory models on both synthetic and real datasets. Their results demonstrated that 1D-CNN achieved the highest accuracy and provided the most recognizable chart pattern classifications compared to other techniques. In our approach, the proposed 1D-CNN model is more compact than that of Liu and Si [10]. Additionally, the baseline models were compared to other deep learning modelsn, not traditional machine learning models.

Besides the CNN model family, the RNN model family, which excels in capturing temporal dependencies, modeling non-linear patterns, and retaining information from previous time steps for accurate sequential predictions, has been studied thoroughly. For example, Kothari et al. [11] explored the use of LSTM neural networks for predicting the next day's NSE stock closing price using nine carefully selected predictors from market fundamentals, macroeconomics, and technical indicators. A performance evaluation revealed that single-layer LSTM outperformed multilayer models in accuracy, as measured by RMSE, MAPE, and correlation coefficient. By integrating advanced machine learning with financial insights, the Stock Market Prediction App aimed to empower users to navigate the complexities of the modern stock market and make informed decisions. Our work differs from this work in terms of the number of day-price to predict, the time series data and the model.

Recently, researchers have introduced various approaches to using LSTM for stock price prediction in the literature. Examples include predicting next 1-day price [12], extending input data for the LSTM model [13], combining an LSTM-based attention mechanism with a cyclic multidimensional gray model [14], and optimizing the LSTM network [15], to name a few. Our proposed method differs from the aforementioned studies in terms of the model used for research and the input data processing technique.

To enhance the predictive capacity of models, hybrid approaches have emerged and garnered significant research interest. Recently, notable hybrid models include CNN-LSTM [16], LSTM-RNN [17], K-means and LSTM [18], CNN-LSTM-ResNet [19]. While these hybrid approaches have demonstrated their effectiveness, their complex architectures come with higher computational costs. In contrast, the proposed compact 1D-CNN model features a simpler architecture compared to these hybrid models.

A limitation of this study is the scope of the comparative analysis performed. While the proposed compact 1D-CNN was benchmarked against a standard LSTM model and a related hybrid CNN-LSTM architecture, comparisons were not extended to other relevant deep learning methods like Gated Recurrent Units (GRUs) or Transformer-based models, nor to traditional time series forecasting methods such as ARIMA or Support Vector Regression (SVR). Including these diverse baselines would undoubtedly provide a more comprehensive assessment of the proposed model's performance relative to the state-of-the-art.

For instance, GRUs offer a potentially simpler recurrent alternative to LSTMs, sometimes achieving comparable performance [20]. Transformer models, utilizing self-attention, represent the cutting edge for many sequence tasks and excel at capturing long-range dependencies, though they often demand significant computational resources and data [21]. Traditional methods like ARIMA serve as crucial statistical benchmarks but inherently assume linearity (or require data transformation) and may struggle with the complex, non-linear dynamics often found in financial markets [22]. The focus of this study, however, is primarily on evaluating the proposed compact 1D-CNN against closely related deep learning sequence models to specifically assess its viability as a lightweight alternative within that domain.

3. **Compact 1D-CNN Model for Short-Term Stock Prediction.** This section introduces the proposed approach for predicting short-term stock prices using a novel compact 1D-CNN architecture. It is divided into three key subsections: (i) the sliding-window mechanism used to preprocess historical stock price data into windowed time-series inputs; (ii) the design, mathematical foundation, and structure of the compact 1D-CNN model; and (iii) the process of leveraging the trained model to generate accurate short-term stock price predictions.

The proposed 1D-CNN model was intentionally designed for compactness and computational efficiency. This is achieved primarily through a deliberately shallow architecture consisting of minimal core layers necessary for capturing short-term temporal features. The architecture, as illustrated in Figure 1, begins by segmenting historical stock price data into window-based data using a sliding window mechanism. Each window represents a sequence of time-series data points, which are then fed into the convolutional neural network.

The first layer of the network is a 1D convolutional layer with 64 filters and a kernel size of 2, activated by the ReLU function, which extracts local temporal features. Using only one convolutional layer minimizes the depth of feature extraction, and the small kernel size

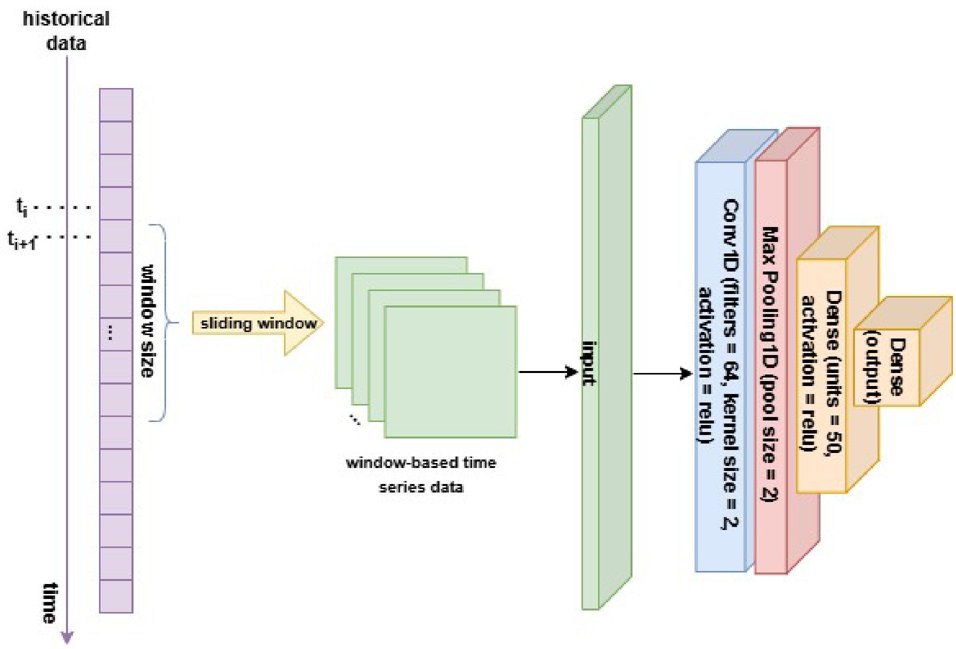

FIGURE 1. The compact 1D-CNN model architecture

reduces the number of parameters and computations required to capture local patterns. This is followed by a max-pooling layer with a pool size of 2, which further enhances efficiency by downsampling the feature maps, reducing dimensionality and computational load for subsequent layers while retaining salient features. Finally, a relatively small dense layer with 50 neurons, also activated by ReLU, processes the pooled features before the output layer predicts the stock price. The limited number of neurons in this layer helps maintain the model's overall compactness. In short, this minimal configuration results in significantly fewer trainable parameters compared to the baseline LSTM and hybrid models, directly contributing to faster training times and lower computational resource requirements during inference.

3.1. **Sliding-window mechanism.** Given the historical stock-price dataset as $X = \{x_t | x_t \in R^n; t = 0, \pm 1, \pm 2, \ldots\}$, in which, $x_t \in R^n$ specifies the stock prices at the time t. The sliding window method is applied to this time series dataset in order to produce window-based inputs for deep learning models.

Generally, the sliding window method is mathematically described as a mapping function in Equation (1).

$$\varphi(x_t, w) : R^n \to R^{w \times n}, \quad \forall x_t \in X \tag{1}$$

where $X \in R^{m \times n}$, w is the window size and $w \ll m$.

Specifically, the function $\varphi(x_t, w)$ maps the original dataset from a tensor with rank 2 ($R^{m \times n}$) to a tensor with rank 3 ($R^{k \times w \times n}$). The result tensors serve the purposed of training model and capturing short-term trends. Algorithm 1 shows the pseudocode of the process of sliding window over historical time series data.

To be more specific, the *window* and *forecast* parameters, which define the number of days used for input and the number of subsequent days used for prediction, are applied to sliding the window throughout the historical time series dataset – *data*. At every sliding step, the corresponding windows are appended to the tensors of X and y. At the end, these tensors are provided as inputs for model training and evaluation.

Algorithm 1: Sliding window

function SLIDING_WINDOW(*data*, *window*, *forecast*)
  $n \leftarrow |data|$
  $X \leftarrow [\,]$
  $y \leftarrow [\,]$
  for i **from** $window$ **to** $n - window - forecast$**:**
    $X \leftarrow X \cup data[i - window - forecast : i - forecast, :]$
    $y \leftarrow y \cup data[i - forecast : i, :]$
  return X, y
end function

3.2. **The compact 1D-CNN model.** The proposed compact 1D-CNN model is composed of convolutional layer, max pooling layer and ReLU activation function. Theoretically, a convolutional operation at the timestep t, which takes the input x and uses the kernel w of size k, yields the output y at the same timestep as in Equation (2).

$$y(t) = (x * w)(t) = \sum_{i=0}^{k-1} x(t+i) \cdot w(i) \tag{2}$$

where $*$ denotes the convolution operation, $x(t+i)$ is the $t+i$-th input element and $w(i)$ is the i-th element of the kernel.

Equation (3) defines the ReLU activation function used in this compact 1D-CNN model.

$$ReLU(z) = \max(0, z) \tag{3}$$

The max pooling, which is applied for the proposed model, is defined as Equation (4).

$$y(t) = \max(x(t), x(t+1), \ldots, x(t+p-1)) \tag{4}$$

In order to adjust the weights of filters during the training phase, the 1D-CNN model uses backpropagation mechanism for the continuous update of the errors. The error value is calculated by a loss function based on the difference between the true value and the predicted value. This error value is then fed into gradient descent during the backpropagation process in order to update the weights. The gradient of convolution is defined as in Equation (5).

$$\frac{\partial L}{\partial w(i)} = \sum_{t} \frac{\partial L}{\partial y(t)} \cdot x(t+i) \tag{5}$$

where $y(t) = \sum_{i=0}^{k-1} x(t+i) \cdot w(i)$ and $\frac{\partial L}{\partial y(t)}$ is the gradient of the loss with respect to the output value.

3.3. **Predicting short-term stock price.** In order to make use of the proposed compact 1D-CNN model, a pipeline process, which is presented in Algorithm 2, is applied. To be more specific, this pipeline is composed of four steps. At the first step, the time series

Algorithm 2: Predict next n day prices

function PREDICT_NEXT_PRICES(*file, window_size, forecast_size*)
  timeseries_data $\leftarrow$ *read_data(file)*
  tensors_data $\leftarrow$ *sliding_window(timeseries_data, window_size, forecast_size)*
  model $\leftarrow$ *initialize_1D_CNN()*
  results $\leftarrow$ *model.predict(tensors_data.X)*
  return *results*
end function

data is loaded into memory. Then, sliding window technique is applied to transforming the original dataset into tensors. Next, the trained 1D-CNN model is initialized. Finally, the model makes and returns predictions based on inputs.

By structuring the prediction process as shown in Algorithm 2, the compact 1D-CNN model can be seamlessly integrated into practical applications. The pseudocode encapsulates the functionality of the proposed architecture and demonstrates how it can efficiently process historical data to deliver accurate and timely short-term price predictions.

4. **Experiment.** In this study, we conducted experiments using historical stock data of Apple Inc. (AAPL) collected from Yahoo Finance[1] via the yfinance[2] package. The dataset spans from December 12, 1980, to October 23, 2024, providing a comprehensive view of AAPL's price trends over several decades. To preprocess the data, we employed the sliding window technique with three distinct configurations to capture short-term and medium-term trends. The first short-term configuration utilized a window size of 5 days to forecast the closing price of the next day (forecast size = 1). The second short-term configuration extended the window size to 10 days with a forecast size of 3 days, while the third configuration focusing on medium-term used a window size of 15 days to predict the closing price for the subsequent 5 days. These window sizes (5, 10, 15 days) were chosen to provide a balance between capturing sufficient historical context for the prediction task and maintaining a focus on recent data, which is often more influential for short-to-medium term forecasting. Smaller windows prioritize recency for short forecasts, while larger windows incorporate more context for medium-term predictions. After generating the processed datasets based on these configurations, the data was split into training, validation, and test sets with proportions of 70%, 15%, and 15%, respectively, ensuring a robust framework for model training and evaluation.

To evaluate the effectiveness of different model architectures in stock price forecasting, three deep learning models were employed: the proposed compact 1D-CNN, an LSTM, and a hybrid model combining 1D-CNN and LSTM. All models were implemented using the TensorFlow[3] framework and trained in the Google Colaboratory[4] environment, ensuring efficient utilization of computational resources.

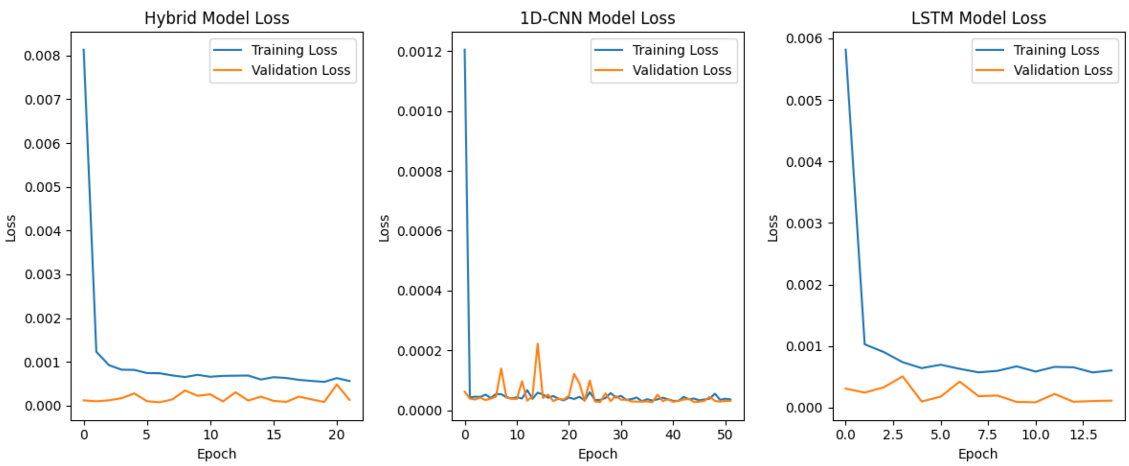

FIGURE 2. Loss curves of models predicting next day price

[1]https://finance.yahoo.com/
[2]https://pypi.org/project/yfinance/
[3]https://www.tensorflow.org/
[4]https://colab.google/

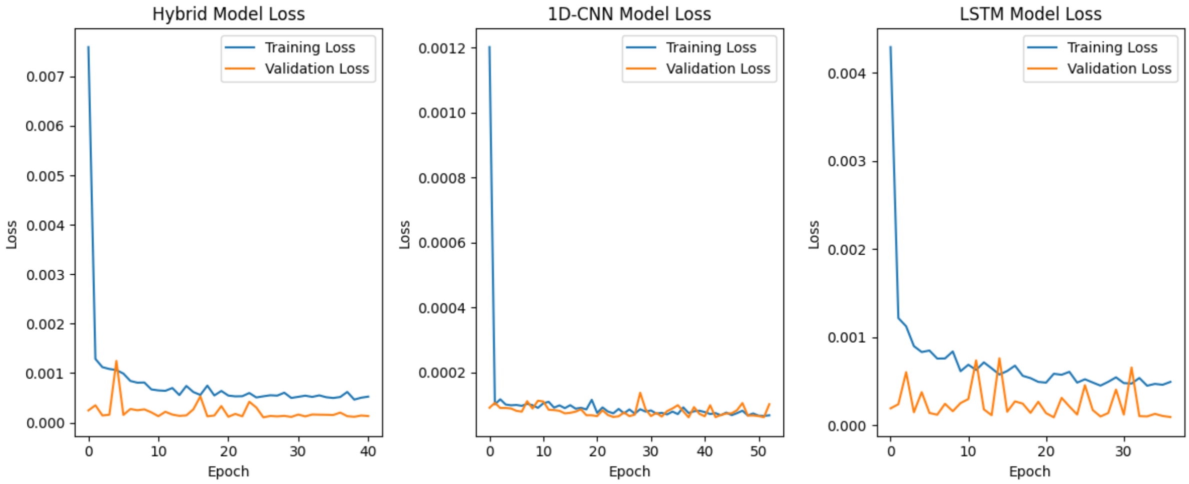

FIGURE 3. Loss curves of models predicting next 3-day prices

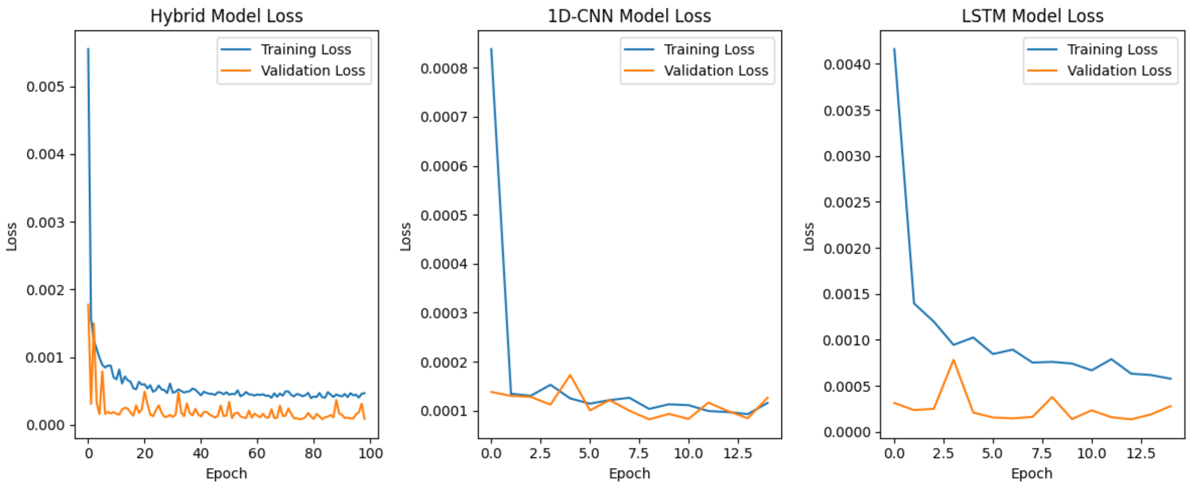

FIGURE 4. Loss curves of models predicting next 5-day prices

The training process for all three models was configured with 100 epochs, and early stopping was applied with a patience of 15 epochs, monitored via the validation loss, to prevent overfitting. Figure 2, Figure 3 and Figure 4 illustrate the training and validation results corresponding to the three distinct sliding window configurations, highlighting the performance differences among the models under varying data setups. From the provided loss curves, all three models (1D-CNN, LSTM, and the Hybrid 1D-CNN-LSTM) demonstrate a consistent decrease in training and validation losses across the experiments, indicating effective learning.

In order to evaluate the models' prediction performances, the following metrics were applied: (i) Mean Squared Error – MSE (Equation (6)); (ii) Mean Absolute Error – MAE (Equation (7)); (iii) Root Mean Squared Error – RMSE (Equation (8)); and (iv) R^2 score (Equation (9)).

$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2 \tag{6}$$

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \tag{7}$$

$$RMSE(y, \hat{y}) = \sqrt{MSE(y, \hat{y})} \tag{8}$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n-1} (y_i - \bar{y})^2} \tag{9}$$

where n is the number of vector elements; y and $\hat{y}$ are true values and predicted values, respectively.

Figures 5, 6, and 7 illustrate the predicted versus actual values for Configurations 1, 2, and 3, respectively. The performance scores for the three models across these configurations are presented in Table 1.

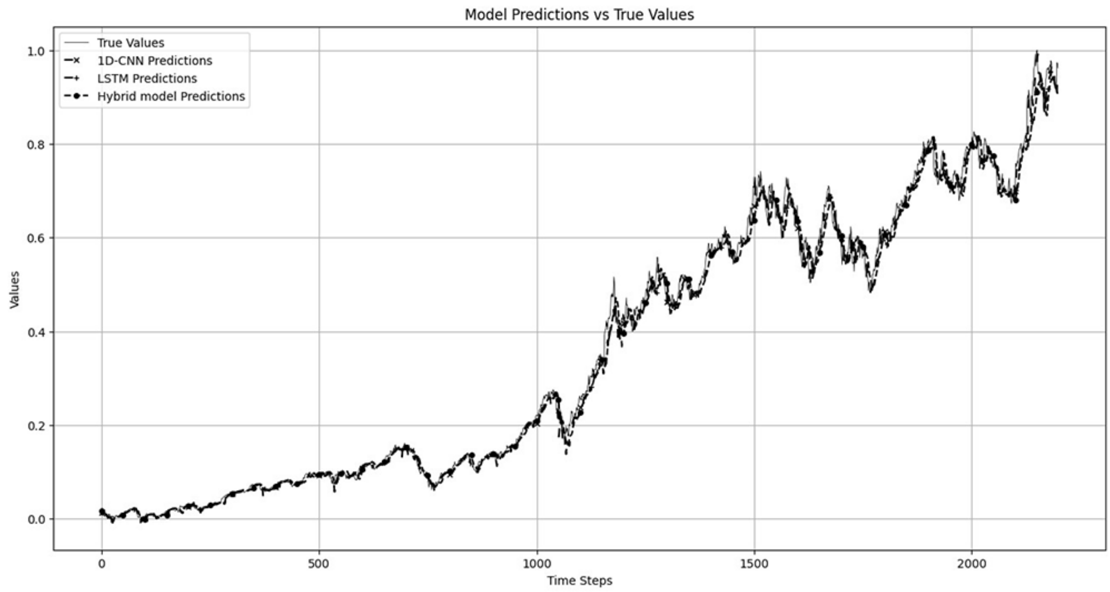

FIGURE 5. Models' predictions of next 1-day price vs. true values

The experimental results presented in Table 1 and Figure 5, Figure 6, and Figure 7 reveal distinct performance characteristics for each model across the different prediction horizons.

For short-term predictions (Configuration 1: 1-day forecast, 5-day window; Configuration 2: 3-day forecast, 10-day window), the proposed compact 1D-CNN model demonstrated superior performance, achieving the lowest MSE/RMSE and highest R^2 scores. This suggests that for predicting immediate future movements, capturing local temporal patterns within the recent price history is crucial. The 1D-CNN architecture, with its convolutional filters (especially the small kernel size of 2 used here), excels at automatically extracting these localized features and short-term dependencies. Its relatively simple and computationally efficient structure appears well-suited to modeling these dynamics without the overhead of more complex recurrent architectures, potentially leading to better generalization on these specific tasks. The max-pooling layer likely further contributes by focusing on the most salient short-term signals detected.

Conversely, in the medium-term scenario (Configuration 3: 5-day forecast, 15-day window), the LSTM model yielded the best results, although the compact 1D-CNN remained competitive. Predicting further into the future inherently involves greater uncertainty and potentially requires modeling dependencies over longer periods present in the larger 15-day

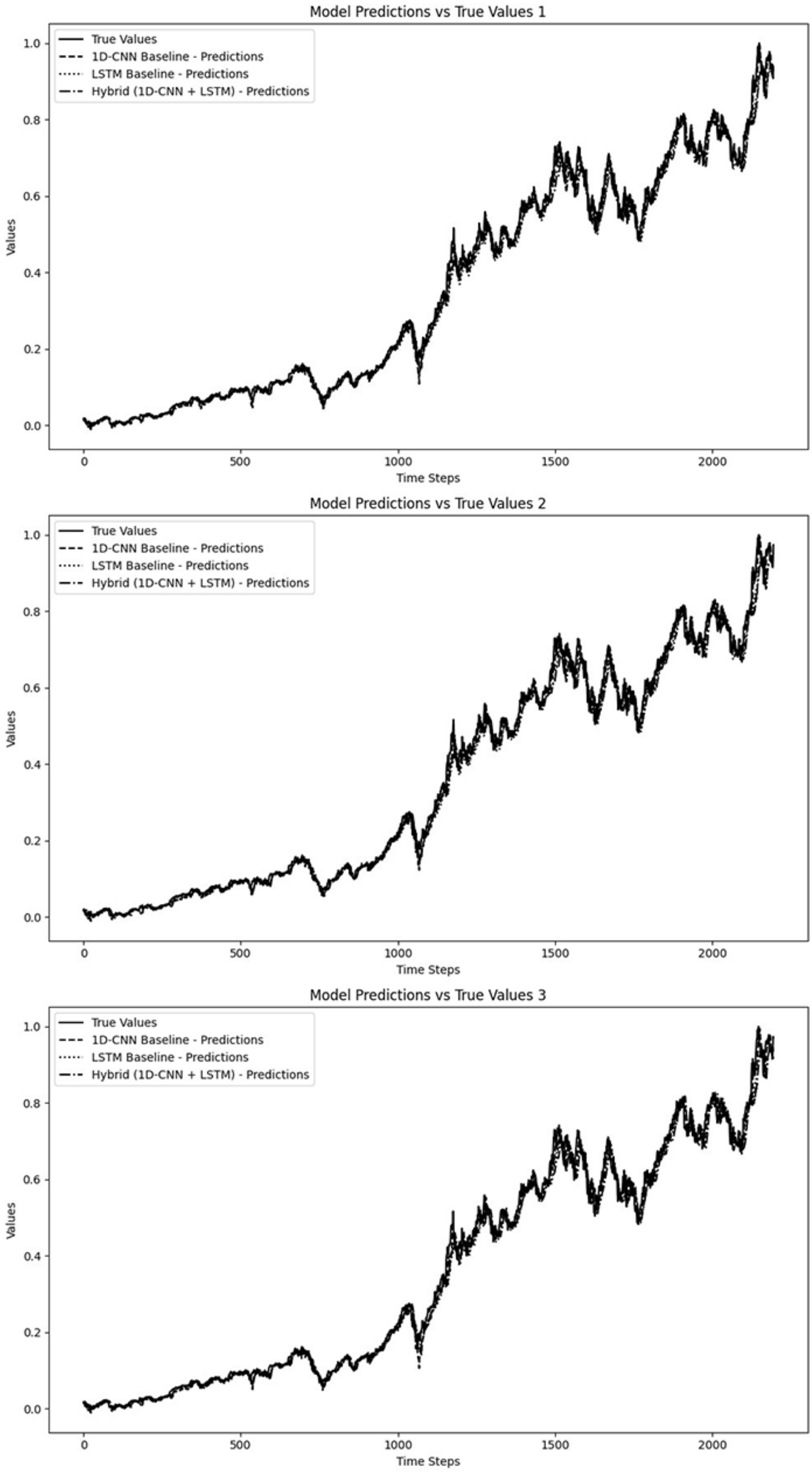

FIGURE 6. Model's predictions of next 3-day prices vs. true values

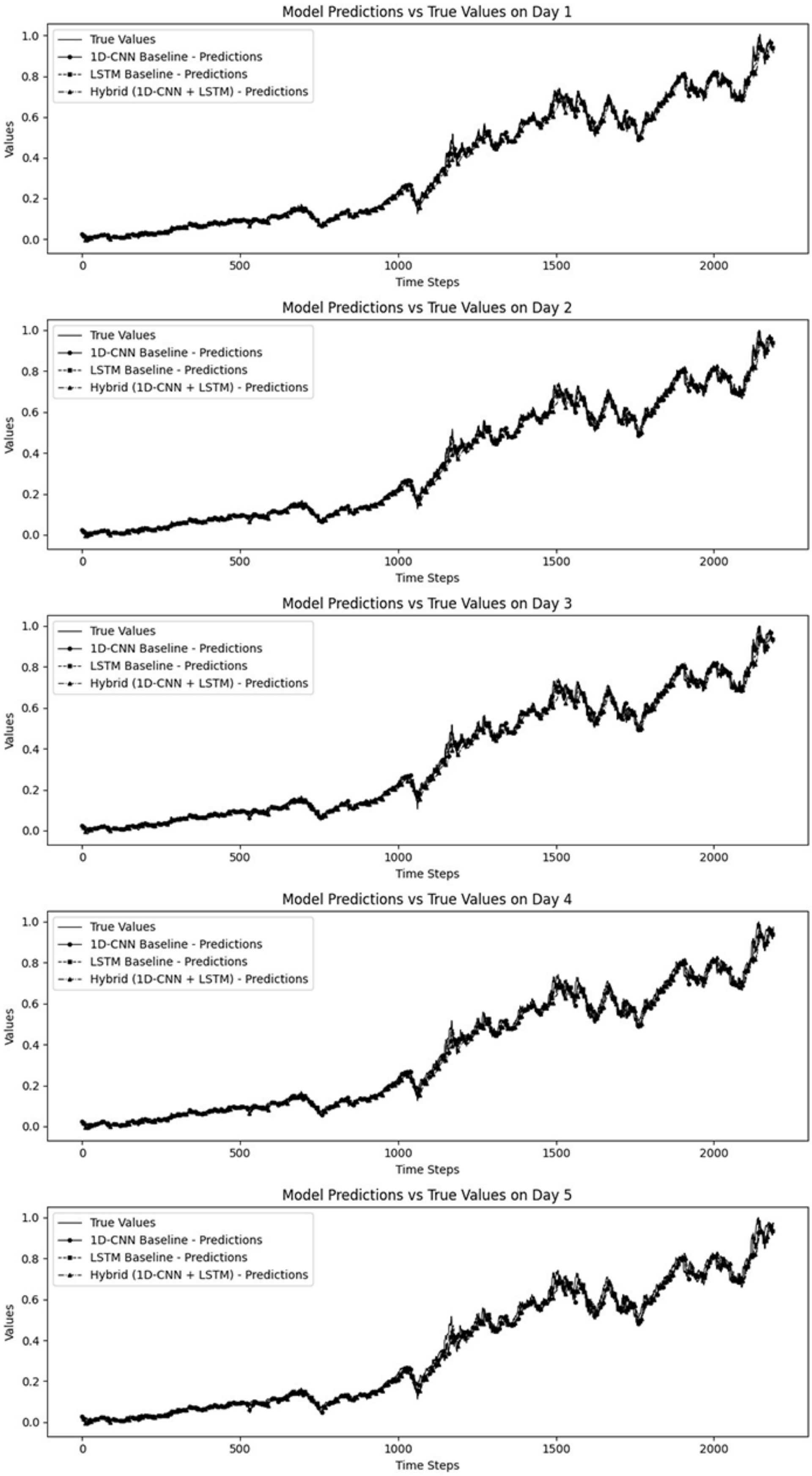

FIGURE 7. Models' predictions of next 5-day values vs. true values

TABLE 1. Experimental result metrics

	Model	MSE	MAE	RMSE	R^2
Configuration 1	Hybrid	0.000399	0.013607	0.019981	0.995207
	1D-CNN	**0.000143**	**0.007928**	**0.011941**	**0.998288**
	LSTM	0.000878	0.020833	0.029638	0.989454
Configuration 2	Hybrid	0.000713	0.017562	0.026693	0.991410
	1D-CNN	**0.000394**	**0.013628**	**0.019842**	**0.995254**
	LSTM	0.000700	0.017492	0.026466	0.991555
Configuration 3	Hybrid	0.002202	0.037596	0.046921	0.887864
	1D-CNN	0.002081	0.036826	0.045618	0.893961
	LSTM	**0.001442**	**0.028632**	**0.037974**	**0.926428**

input window. The LSTM model likely benefits from leveraging its memory capabilities to capture longer-range temporal dynamics necessary for the 5-day forecast horizon.

These findings highlight a potential trade-off between model complexity and forecasting horizon: the lightweight 1D-CNN excels when recent local patterns dominate (short-term), while the more complex model like LSTM shows advantages when longer-term sequential dependencies become more critical (medium-term).

5. **Conclusion.** Short-term stock price prediction is a complex task due to the dynamic and non-linear nature of financial markets. This study introduced a compact 1D-CNN architecture that addresses the challenges of computational efficiency and complex model structures, demonstrating its effectiveness in capturing temporal patterns and predicting next 1-day, 3-day, and 5-day prices using historical data from Apple Inc. (AAPL). Comprehensive experiments revealed its competitive accuracy compared to traditional LSTM and hybrid 1D-CNN-LSTM models, particularly for short-term predictions where simplicity and speed are critical. The proposed model's practical and efficient design makes it a promising tool for various financial applications where timely short-term predictions are valuable. Specifically, its demonstrated accuracy on 1-to-3-day horizons and computational efficiency lend themselves well to integration into short-term algorithmic trading strategies or real-time decision support systems for traders and portfolio managers. Furthermore, its compactness suggests potential for deployment in resource-constrained environments or as an efficient predictive module within larger financial analysis platforms. Future work will focus on integrating advanced feature engineering techniques, incorporating external factors such as market sentiment and macroeconomic indicators, and expanding evaluations to a broader range of financial instruments and markets to enhance the model's generalizability and performance.

REFERENCES

[1] W. Jiang, Applications of deep learning in stock market prediction: Recent progress, *Expert Syst. Appl.*, vol.184, 115537, 2021.

[2] J. V. R. Ferro, R. J. R. Dos Santos, E. de Barros Costa and J. R. da Silva Brito, Machine learning techniques via ensemble approaches in stock exchange index prediction: Systematic review and bibliometric analysis, *Appl. Soft Comput.*, 112359, 2024.

[3] A. O. Ige and M. Sibiya, State-of-the-art in 1D convolutional neural networks: A survey, *IEEE Access*, 10609371, 2024.

[4] E. Hoseinzade and S. Haratizadeh, CNNpred: CNN-based stock market prediction using a diverse set of variables, *Expert Syst. Appl.*, vol.129, pp.273-285, 2019.

[5] S. Alhazbi, A. B. Said and A. Al-Maadid, Using deep learning to predict stock movements direction in emerging markets: The case of Qatar stock exchange, *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp.440-444, 2020.

[6] C. C. Hung, Y. J. Chen, S. J. Guo and F. C. Hsu, Predicting the price movement from candlestick charts: A CNN-based approach, *Int. J. Ad Hoc Ubiquitous Comput.*, vol.34, no.2, 111, DOI: 10.1504/IJAHUC.2020.107821, 2020.

[7] S. K. Chandar, Convolutional neural network for stock trading using technical indicators, *Autom. Softw. Eng.*, vol.29, no.1, 16, DOI: 10.1007/s10515-021-00303-z, 2022.

[8] Y. Yao, Z. Zhang and Y. Zhao, Stock index forecasting based on multivariate empirical mode decomposition and temporal convolutional networks, *Appl. Soft Comput.*, vol.142, 110356, DOI: 10.1016/j.asoc.2023.110356, 2023.

[9] W. Chen, N. An, M. Jiang and L. Jia, An improved deep temporal convolutional network for new energy stock index prediction, *Inf. Sci.*, vol.682, 121244, DOI: 10.1016/j.ins.2024.121244, 2024.

[10] L. Liu and Y.-W. Si, 1D convolutional neural networks for chart pattern classification in financial time series, *J. Supercomput.*, vol.78, no.12, pp.14191-14214, DOI: 10.1007/s11227-022-04431-5, 2022.

[11] A. Kothari, A. Kulkarni, T. Kohade and C. Pawar, Stock market prediction using LSTM, in *Smart Trends in Computing and Communications*, T. Senjyu, C. So-In and A. Joshi (eds.), Singapore, Springer Nature, 2024.

[12] S. K. Chandar, Deep learning framework for stock price prediction using long short-term memory, *Soft Comput.*, vol.28, no.17, pp.10557-10567, DOI: 10.1007/s00500-024-09836-3, 2024.

[13] H. G. T. Thu, T. N. Thanh and T. L. Quy, Dynamic sliding window and neighborhood LSTM-based model for stock price prediction, *SN Comput. Sci.*, vol.3, no.3, 256, DOI: 10.1007/s42979-022-01158-1, 2022.

[14] Q. Liu, Y. Hu and H. Liu, Enhanced stock price prediction with optimized ensemble modeling using multi-source heterogeneous data: Integrating LSTM attention mechanism and multidimensional gray model, *J. Ind. Inf. Integr.*, vol.42, 100711, DOI: 10.1016/j.jii.2024.100711, 2024.

[15] B. Gülmez, Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm, *Expert Syst. Appl.*, vol.227, 120346, DOI: 10.1016/j.eswa.2023.120346, 2023.

[16] F. Bodaghi, A. Owhadi, A. K. Nasr and M. K. Sameni, A novel CNN-LSTM model for predicting the railway sector stock price in the Tehran stock exchange, *2023 9th International Conference on Web Research (ICWR)*, pp.108-114, 2023.

[17] H. Harbaoui and E. A. Elhadjamor, Enhancing stock price prediction: LSTM-RNN fusion model, *Procedia Comput. Sci.*, vol.246, pp.920-929, DOI: 10.1016/j.procs.2024.09.511, 2024.

[18] Y. Chen, J. Wu and Z. Wu, China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach, *Expert Syst. Appl.*, vol.202, 117370, DOI: 10.1016/j.eswa.2022.117370, 2022.

[19] P. Khodaee, A. Esfahanipour and H. M. Taheri, Forecasting turning points in stock price by applying a novel hybrid CNN-LSTM-ResNet model fed by 2D segmented images, *Eng. Appl. Artif. Intell.*, vol.116, 105464, DOI: 10.1016/j.engappai.2022.105464, 2022.

[20] Y. Gao, R. Wang and E. Zhou, Stock prediction based on optimized LSTM and GRU models, *Sci. Program.*, vol.2021, pp.1-8, DOI: 10.1155/2021/4055281, 2021.

[21] C. Wang, Y. Chen, S. Zhang and Q. Zhang, Stock market index prediction using deep Transformer model, *Expert Syst. Appl.*, vol.208, 118128, 2022.

[22] M. M. R. Majumder and M. I. Hossain, Limitation of ARIMA in extremely collapsed market: A proposed method, *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp.1-5, 2019.

Author Biography

Nguyen Thi Huong Giang received her Master's degree in Computer Science from Hue University, Vietnam, in 2006. She is currently a lecturer in the Department of Informatics, University of Education, Hue University, Vietnam. Her primary research interests include deep learning, ontology, database systems, and data analysis.

Le Manh Thanh received his Ph.D. degree from the University of Szeged, Hungary, in 1994. He is currently a lecturer in the Department of Information Technology, University of Sciences, Hue University, Vietnam. His research interests include advanced methods and technologies for agent and multi-agent systems, intelligent information and database systems, ontology, and deep learning.

Cuong H. Nguyen-Dinh received his Doctorate in Computer Science from Khon Kaen University, Thailand, in 2016. Currently, he serves as the Dean of the Faculty of Business and Technology at Phu Xuan University, Vietnam. His primary research interests encompass deep learning, text mining, knowledge management systems, and recommender systems.