Financial Engineering and Risk Management (2022)
Clausius Scientific Press, Canada

DOI: 10.23977/ferm.2022.050703
ISSN 2523-2576 Vol. 5 Num. 7

Prediction stock price based on CNN and LSTM models

He Kexin[1,a,*], **Zhang Zhijin**[1,b]

[1]*The School of Business, Wenzhou-kean University, Li'ao Street, Wenzhou, China*
[a]*hekex@kean.edu*, [b]*zhanzhij@kean.edu*
[*]*Corresponding author: He Kexin*

Keywords: CNN method, LSTM method, compare, accuracy

Abstract: To compare the ability of efficiently prediction stock price based on CNN and LSTM method, we take the data from Tata Consultancy Services as our object to study. We run 35 times for their training and test loss and errors. Afterwards we made a comparison on the data, and getting the result that in contract to LSTM method, CNN method has a better ability on prediction accuracy in short time.

1. Introduction

Financial markets are constantly changing, and people have always been very interested in stock forecasts. One of the great characteristics of stocks is uncertainty, and it is difficult to predict what will happen in the future and cause losses. The financial market is a dynamical system that is intricate, evolving, and non-linear. Data intensity, noise, non-stationarity, unstructuredness, high levels of uncertainty, and hidden relationships are characteristics of the field of financial forecasting [1].

The law of the stock is difficult to be discovered, and the possibility of exploring it by traditional methods is small, which will cost a lot of time and energy. Aiming for large returns utilizing clearly defined trading techniques, stock market forecasters concentrate on creating approaches to accurately foresee index values or stock prices. The key to accurate stock market forecasting is to produce the best outcomes with the fewest possible inputs and the simplest possible stock market model [2]. The basic analysis method and the technical analysis method are the primary components of the traditional analysis method, which is based on economics and finance[3].

Due to technological advancements in computer hardware and advancements in machine learning techniques in recent years [4], more creative methods are proposed. These algorithms include ANNs, evolutionary computation, swarm intelligence, artificial immune systems, and fuzzy systems. Bio-inspired algorithms, which are designed after the central nervous systems of the brain, have demonstrated significant success in the field of artificial intelligence [5]. Deep Learning has emerged as one of the most alluring research areas in these years thanks to the extensive usage of powerful computers and the availability of vast amounts of data [4]. Given the unpredictability of the stock market model, soft computing techniques are strong contenders for capturing the nonlinear relationships in the stock market and producing meaningful forecasting results without necessarily requiring prior knowledge of the statistical distributions of the input data [2].

2. Literature Review

According to big data and historical experience, it is not difficult to find that it is not easy to predict stock returns, and its volatility and uncertainty often cause difficulties in prediction. Technical analysis is a strategy that is frequently used to model and forecast the stock market. It is based on historical data from the market, primarily price and volume[6]. Political developments, broader economic conditions, and traders' expectations are just a few of the many variables at play in the world of finance. As a result, forecasting price changes in the financial market is challenging. Academic studies indicate that market price changes are increasingly not random[1]. As computer technology has moved deeper into finance, more forecasting and analysis methods have been developed to improve accuracy. To estimate a function by minimizing an upper bound of the generalization error, support vector machine (SVM), a highly particular form of learning algorithms, based on the unique theory of the structural risk minimization principle[1]. A time series forecasting method based on neural networks was put forth by Sun et al. in 2005. This approach combines the radial basis function (RBF) neural network with the optimal partition algorithm (OPA) [7]. In 2014, Adhikari et al. proposed a strategy combining arbitrary walk (RW) and manufactured neural organize (ANN) to anticipate four budgetary time arrangement information, the accuracy of prediction is greatly improved.[8] In 2018, the exploratory comes about of Hu et al. appear that convolutional neural organize can anticipate time arrangement, and profound learning is more appropriate for understanding the issue of time arrangement[3]. CNN is a type of machine learning algorithm that is used to solve problems like recognizing images and extracting features [8].

The most potent deep learning methods, CNNs, have been employed extensively for a variety of computer vision tasks. Convolution layers and pooling layers alternately make up the usual CNN architecture. In contrast to conventional machine learning methods, the primary goal of this layering sequence is to derive high-level features, which avoids the human feature engineering stage [9]. CNN is frequently used in feature engineering because it has the characteristic of focusing on the most evident elements in the line of sight. LSTM, which is frequently employed in time series, has the property of growing in accordance with the passage of time [3]. Based on a stock forecasting model of CNN-LSTM, this essay lists a lot of experience and data, with the help of the computer to calculate the rule, and constantly according to the actual situation to detect, analyze and correct, finally get a more accurate prediction result.

3. Method

3.1. CNN

In 1998, Lecun et al [10] proposed a network model called CNN, which is a kind feedforward neural network with good performance in image and natural language processing [11]. It can be effectively applied to the forecasting of time series. CNN can improve the efficiency of model learning by greatly reduce the number of parameters through local perception and weight sharing[12]. CNN is mainly consisted of two parts: the convolution layer and the pooling layer. Each convolution layer contains a plurality of convolution kernels, and its calculation formula is shown in formula (1).After the convolution operation of the convolutional layer, the features of the data are extracted, but the extracted feature dimension is very high, so in order to solve this problem and reduce the cost of training the network, a pooling layer is added after the convolutional layer to reduce Feature dimension: After the convolution operation of the convolutional layer, the features of the data are extracted, but the extracted feature dimension is very high, so in order to solve this problem and reduce the cost of training the network, a pooling is added after the convolutional layer layer to reduce the feature dimensionality:

$$l_t = \tanh(x_t * k_t + b_t) \tag{1}$$

where l_t represents the output value after convolution, tanh is the activation function, x_t is the input vector, k_t is the weight of the convolution kernel, and b_t is the bias of the convolution kernel.

3.2. LSTM

At time t, x_t is the input data of the LSTM cell, h_{t-1} is the output of the LSTM cell at the previous moment, c_t is the value of the memory cell, h_t and is the output of the LSTM cell. The calculation process of the LSTM unit can be divided into the following steps

(1) First, calculate the value of the candidate memory cell $\tilde{c}_t$,W_c is the weight matrix b_c is the bias.

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2}$$

(2) Calculate the value of the input gate i_t, the input gate controls the update of the current input data to the state value of the memory cell, σ is sigmoid function, W_i is the weight matrix, b_i is the bias.

$$i_t = \sigma(W_i \cdot \left[h_{t-1}, x_t\right] + b_i \tag{3}$$

(3) Calculate the value of the forget gate ft, the forget gate controls the update of the historical data to the state value of the memory cell, Wf is the weight matrix, bf is the bias.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{4}$$

(4) Calculate the value of the current moment memory cell c_t, andc_{t-1} is the state value of the last LSTM unit.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{5}$$

Where “*” represents dot product.The update of the memory element depends on the state value of the last element and the candidate element, controlled by the input gate and the forget gate

(5) Calculate the value of the output gate o_t, the output gate controls the output of the state value of the memory cell, W_o is the weight matrix, b_o is the bias.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

(6) Finally, calculate the output of LSTM unit h_t.

$$h_t = o_t * \tanh c_t \tag{7}$$

Three control gates and memory units, LSTM is convenient for saving, reading, resetting and updating long-term information. It should be noted that due to the sharing mechanism of LSTM internal parameters, the size of the output can be controlled by setting the size of the weight matrix. LSTMs build in long delays between input and feedback. Gradients neither burst nor vanish because the internal state of the memory cells in this architecture maintains a continuous stream of errors.

4. Experimental results

We take the data from Tata Consultancy Services as our object of study. Figure 1 shows the loss of the CNN method, we take the loss as the y-axis and epochs as the x-axis, and we run 35 times. Comparing the training loss and validation loss, we found that with the epochs going through, the trend of training loss and validation loss approaches down and keeps a similar trend. In the beginning, the validation loss keeps a level of 0.5; after 18 epochs, the validation training loss decreases and trends to a level of 0.1. It can be found that when we run 27 times later, we can see a slight change in the loss. As for training loss, it keeps a similar trend as the validation loss line in that it continues at a level of 1.58 loss or so before 17 epochs, then it decreases quickly and trends to a 0.3 loss level. Finally, it keeps a stable level. The LSTM method is shown in Figure 2. Both Training loss and Validation loss fall immediately from 1 to 5 Epochs. After that, both of them tend to a low and stable line, approximately 0.00 loss.

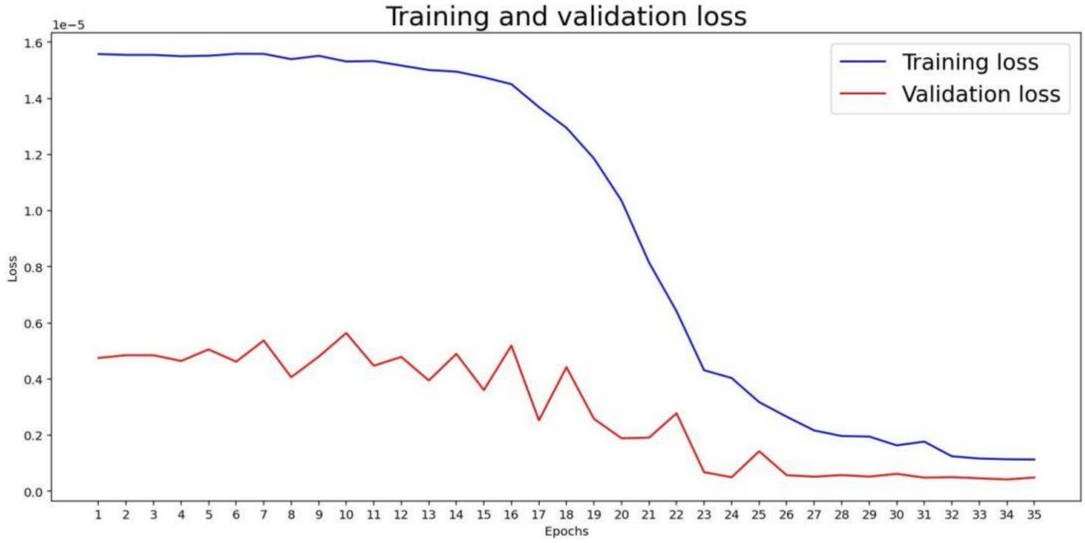

Figure 1: The training and validation loss of the CNN

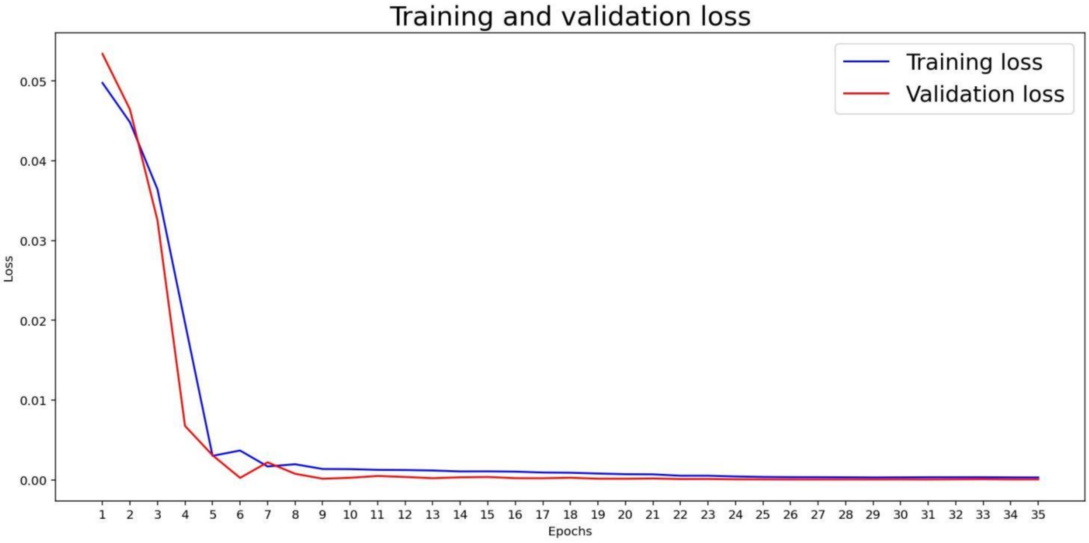

Figure 2: The training and validation loss of the LSTM

Figure 3 and Figure 4 represent the training and validation errors of the CNN method and LSTM method separately. The following two graphs take Error as the y-axis and Epochs as the x-axis. What we can see here in figure 3 is that from 1 to 5 Epochs, the Error of Training and Validation falls steeply. After five epochs, training error and validation error fall stably. In the end, the training error trend to a level of 0.03 error, and the validation error trends to 0.02 error.

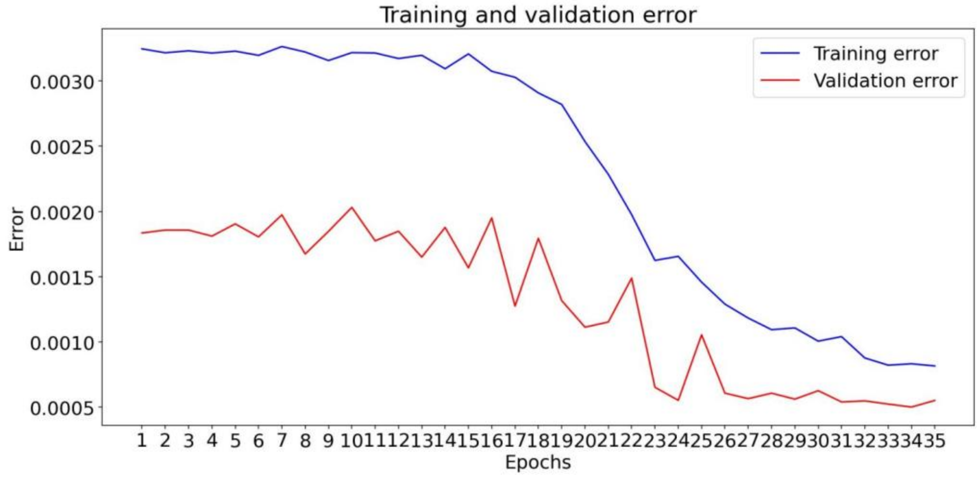

Figure 3: The training and validation error of the CNN

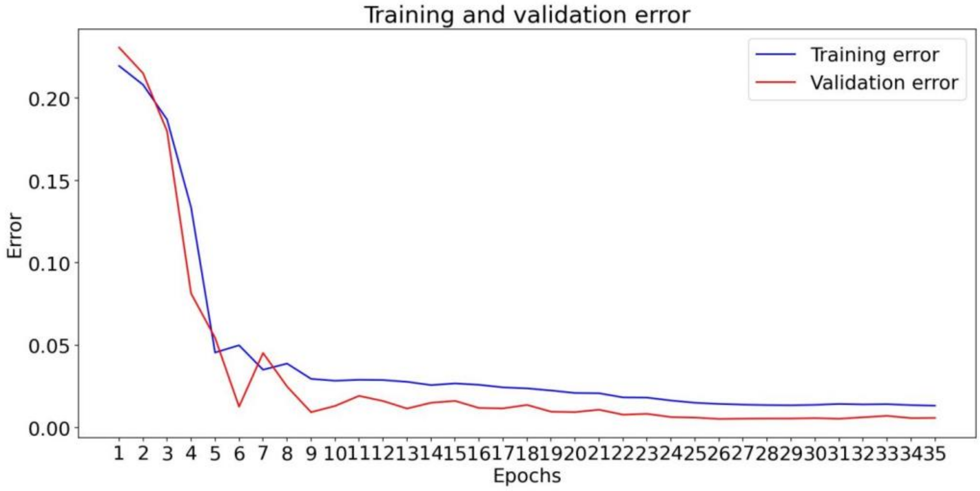

Figure 4: The training and validation error of the LSTM

We predict the Close Value of the CNN method and LSTM and put them in the next two graphs, Figure 5 and Figure 6 in these graphs, the number of Trading Days was designed as the x-axis, Close Value or scaled as the y-axis. Comparing the prediction result of the CNN method with the prediction result of the LSTM method, we can find though the shape of them are similar, it none the less be found the accuracy of the CNN method is much higher for its Close Value is much smaller.

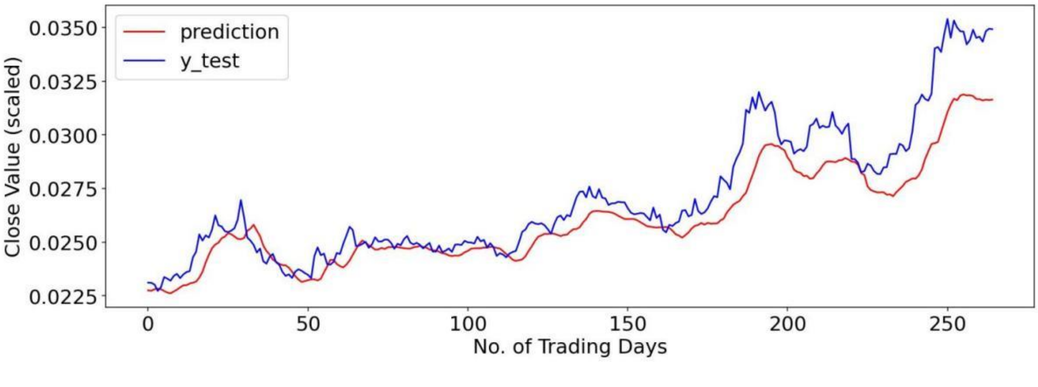

Figure 5: The prediction result of the CNN

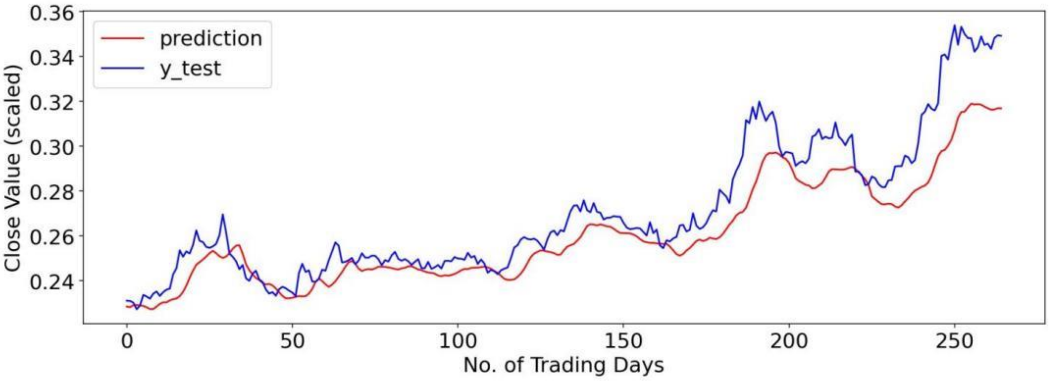

Figure 6: The prediction result of the LSTM

Now we conformity all lines into Figure 7 and Figure 8. in the following graphs, we can see the prediction on training samples, validation samples, and y-train. By studying the CNN method in Figure 7 and LSTM- method in Figure 8, we can see the CNN method has a better fit on prediction both on training samples and validation samples since the close value is ten times smaller than LSTM-method.

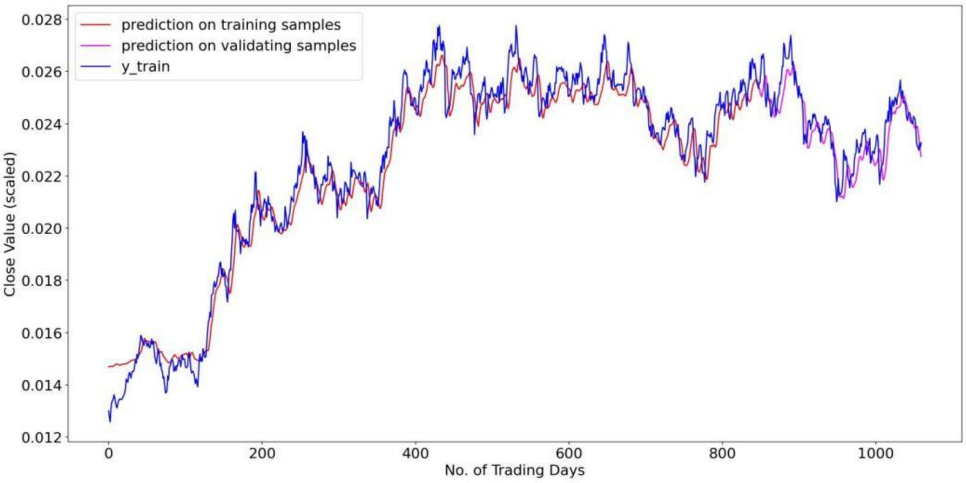

Figure 7: Integration of forecasts result of the CNN

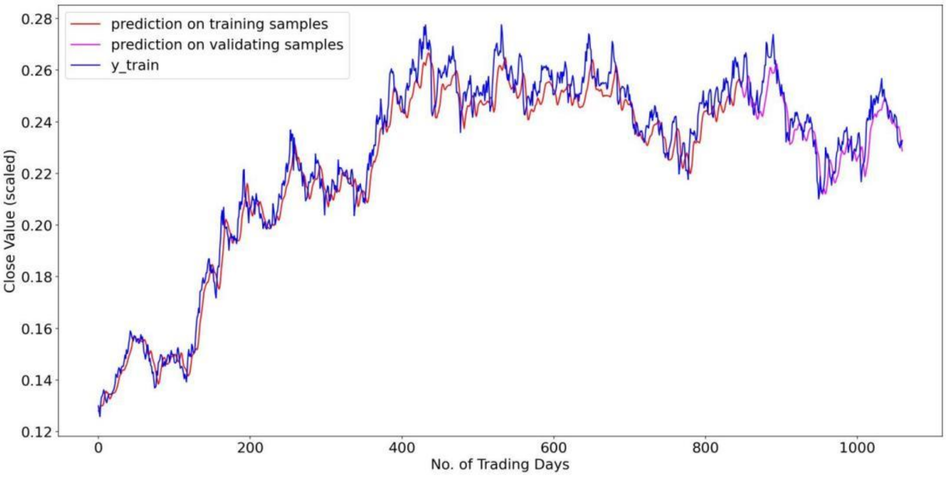

Figure 8: Integration of forecasts result of the LSTM

It is discovered that CNN has 5 times higher accuracy on Train RMSE, and 4 times higher accuracy on Test RMSE. Comparing the Mean absolute percentage error of CNN and LSTM by table1, we can clearly find that CNN has a 5 to 6 times smaller number of mean absolute errors, which means it has a greater accuracy on the stock price prediction. Then we take the test MAE, getting the result that LSTM has 4 times larger error than CNN.

Table 1: The comparison of CNN and LSTM result

	Train RMSE	Train MAE	Train MAPE	Test RMSE	Test MAE	Test MAPE
CNN	0.0007	0.0006	0.0014	0.0010	104.6630	3.5933
LSTM	0.0037	0.0030	0.0057	0.0048	479.4051	16.7314

5. Conclusion

In this paper, we investigate the use of an LSTM network system to predict future trends in stock prices based on historical empirical prices as well as technical analysis indicators in an attempt to improve the reliability of the predictions. To achieve this goal, we perform a series of experiments using a large collection of historical data to build a relevant database and forecasting model, and continuously revise the predicted trends based on the generated results. The models make full advantage of the time sequence properties and various stock data comparisons. The features of the input data are extracted by CNN, and LSTM is then used to learn the feature data to forecast the ending price of the stock. In order to validate the experimental findings, this research uses pertinent data from Tata Consultancy Services as an example. We run 35 epochs for CNN and LSTM method, comparing their training and testing loss, training and testing error. We finally find the higher accuracy of the CNN method compering with the LSTM method in short term prediction. In the end of the experience, we list a table for comparison. It can be found easily that CNN has 4 to 6 times more accurate. Thus, we conclude that the CNN method has a better ability on the accuracy of prediction in short time. We hope our article can help more people to learn about CNN and LSTM method.

References

[1] Huang, W., Nakamori, Y., & Wang, S.-Y. (2004). Forecasting stock market movement direction with support vector machine. Computers & Operations Research, 32(10), 2513–2522.

[2] Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods. Expert Systems with Applications, 36(3), 5932–5941.

[3] Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-Based Model to Forecast Stock Prices. Complexity, 2020, 1–10.

[4] Sismanoglu, G., Onde, M. A., Kocer, F., & Sahingoz, O. K. (2019). Deep Learning Based Forecasting in Stock Market with Big Data Analytics. 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT).

[5] Vui, C. S., Soon, G. K., On, C. K., Alfred, R., & Anthony, P. (2013). A review of stock market prediction with Artificial neural network (ANN). 2013 IEEE International Conference on Control System, Computing and Engineering.

[6] Nelson, D. M. Q., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. 2017 International Joint Conference on Neural Networks (IJCNN).

[7] Y. Sun, Y. Liang, and W. Zhang, "Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting," Neural Computing and Applications, vol. 14, pp. 1441–1449, 2005.

[8] Y. Hu, "Stock market timing model based on convolutional neural network – a case study of Shanghai composite index," Finance& Economy, vol. 4, pp. 71–74, 2018.

[9] Rasheed, J., Jamil, A., Hameed, A. A., Ilyas, M., Özyavaş, A., & Ajlouni, N. (2020, October). Improving stock prediction accuracy using cnn and lstm. In 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) (pp. 1-5). IEEE.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document

recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[11] B. S. Kim and T. G. Kim, "Cooperation of simulation and data model for performance analysis of complex systems," International Journal of Simulation Modelling, vol. 18, no. 4, pp. 608–619, 2019.
[12] L. Qin, N. Yu, and D. Zhao, "Applying the convolutional neural network deep learning technology to behavioural recognition in intelligent video," Tehnicki Vjesnik-Technical Gazette, vol. 25, no. 2, pp. 528–535, 2018.