International Applied Business and Engineering Conference 2021
August 25, 2021, Riau, Indonesia

Price Difference Embedded Multivariate Long Sort-Term Memory for Stock Movement Prediction

1st Wawan Yunanto
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
d10615811@mail.ntust.edu.tw

***Abstract*—The course of the capital market is complicated, unpredictable, and volatile for investors to formulate. Fundamental and technical analysis are the main common approaches to predict stock prices used by the economic experts nowadays. The fundamental aspect is determined by the internal factor of the companies, but the technical one is clearly represented on a daily basis in the stock market. Stock price is not the only important item for investors to make investment policy. The fluctuation in the trading floor has become the most important issue to be considered. In this research, we propose a prediction framework, namely Price Difference Embedded Multivariate Long Sort-Term Memory (PDEM-LSTM), to combine stock price and movement prediction into a single pipeline. We employ recurrent deep learning modeling technics into stock market forecasting since there are sequential properties in the technical components. Our work solely based on these sequences or timeseries features to simplify the experimental setting and more focus on the improvement compared to most previous studies. Our benchmark compares results from univariate scheme on the same sequences with 3 difference features which are current day and next day price along with the price difference between those days. We use 5 stock issuers from 5 different stock indices and the market data taken from January 1, 2000, to December 31, 2020. The results showed that price difference feature embedded into LSTM in multivariate setting greatly improve stock movement prediction without degrading stock price forecasting too much. It is simple and robust; it can be attached on most stock prediction techniques in the feature engineering phase.**

Keywords—stock price, stock movement, LSTM, univariate, multivariate

I. INTRODUCTION

Capital market is a high-risk investment due to its broad factors that can affect the profit taking strategy. There are two major aspects noticed by investors before they make trading decisions. Company performance become the first aspect to make a good justification about how secure the investment will be [1]. These fundamental statistics tend to attract long term investors to spend their capital and participate in every decision making made by the company. On the other hand, there are sort-term investors who pay attention more on the second aspect, namely technical aspect, in trading. Technical aspect mostly driven by external factors, such as: price trend, investor sentiment, political situation, or event global pandemic just like we are all experiencing today. Hence, investors and traders are prone to severe monetary loss if their do not maintain the financial risk management carefully. Conversely, careful data-driven investment decisions can maximize profits and minimize loss greatly [2].

The vast majority research in financial studies rely on the sequential data that widely available for public. In stock market there are several time-series features to choose for building robust prediction method for traders. The most common features are stock prices, e.g., opening and closing price. Prediction the next day price seems to be the most important thing for investors because in most people's view money is closely related to price. The problem is that even the lowest error price prediction for the next day does not give us any clue whether we should buy or sell. In a statistical perspective, low error means a good prediction but what the investors really need is the direction of the next day price, is it better or worse than today's price. If next day price predicted to be lower than today, regardless the error, traders will likely sell their share and vice versa, as simple as that. Therefore, prediction of stock movement direction considered to be significant in financial studies and plays a key role in determining to buy and sell stocks [3].

Unlike stock price prediction, which focus on predicting the next day price by minimizing the mean squared error and usually resulting a really low errors, stock movement prediction is much harder to cope so that it can be implement in trading strategy. We cannot just use the price differences between real and predicted prices to justify the stock movement direction because oftentimes they are pointing to the opposite direction. Some researchers engineered new features [4], utilized technical analysis [2], or even formulated a new loss function (in deep learning) to get better stock movement prediction [5]. In this study, we focus on how to build a simple framework to get significant improvement, only using single time-series feature in the stock market data. Our goal is building a method that can be generalized to be use in any stock movement prediction to improve the prediction result. We are not designing a production level end to end method but more on the enhancement of any deep learning-based predictor using our strategy.

In brief, the main contributions of this work can be summarized as follows:

1. We extend the use of price differences and embed them in LSTM-based predictor in multivariate manner.
2. We improve stock movement prediction result while keeping stock price prediction errors at the acceptable level using single pipeline network.
3. We provide extensive experiments in univariate manner as well to benchmark our proposed framework against common prediction scenarios.

The rest of paper is organized as follows. Previous works in this area are introduce in Section 2. In Section 3 we explain about the methodology of this research. Our results and its analysis are provided in Section 4. Finally, the conclusion is drawn in Section 5.

II. Related Work

Stock movement prediction is widely considered challenging due to the high stochasticity of the stock signal and sometimes chaotic. Basically, it is a time-series problem where the temporal dependencies are crucial [6]. Research in stock market field, most of previous works are usually a comprehensive research to get better result for traders. They use all features available, fundamental and technical analysis, or even external data such as social media.

Technical indicators become the most popular features to use by researcher to forecast next day stock price and movement. These features used by predictors, mostly machine learning models, to extract more features in order to improve prediction ability [7]. Shallow model machine learning methods such as k-Nearest Neighbor (kNN), Support Vector Machine (SVM), and Random Forest are often utilized to capture the effect of using the features from the technical analysis and price movement. These models use the deterministic trends of stock price index movement to learn the correlation between current and previous stock data, but still get undesirable outcome [4].

A deep learning approach has been used in several research in this area. The usage of Convolutional Neural Network combined with LSTM has been proposed in [3]. Although this method performs well in detecting stock price but struggle in forecasting stock movement. The best performance is only 0.6144 in average on 5 market indices using their proposed network with CNN3D-DR+LSTM. An extensive deep learning model based on Factorization Machine and Attention Mechanism was utilized to predict stock price movement as presented in [5]. The idea of this paper is to construct a convolutional neural network graph based on a deep factorization machine and attention mechanism (FA-CNN) to improve the prediction accuracy of stock price movement via enhanced feature learning. The average accuracy obtained is below 60%, approximately equal to others. Other researchers try to involve external aspects related to investment field, namely social media comments and company correlations, in order to overcome the non-stationary nature of stock market [8]. They best accuracy is 0.608 on MAN-SF with Fundamental Analysis models. It is slightly better than the others but still not far from flip of a coin.

One of the advantages in using deep learning method is that we can design our input vector to be suitable to our need. Combination of time-series features and engineered features can capitalize the prediction power of the model in multivariate scheme. Our work is building a feature engineering method to benefit from this advantage.

III. Methodology

A. Dataset

In this research, we utilize 5 stock issuers namely Abbott Laboratories (ABT), Walmart Inc. (WMT), Intel Corporation (INTC), Boeing Company (BA) and HSBC HOLDINGS (0005.HK), from these international stock market indices respectively: S&P 500, DJIA, NASDAQ, NYSE and HKEX. We collect the data between January 1, 2000, to December 31, 2020, from Yahoo Finance (https://finance.yahoo.com). As mentioned earlier, our work focusses on finding performance improvement by employing time-series feature contained in stock market data, that is closing stock price.

Our proposed experiment scheme is a multivariate LSTM that takes multi-dimensional vectors as input. Before creating the input sequences, we preprocess the original dataset to generate next day closing price signal and extract the price movement by capturing the difference between today and next day prices. We keep the price difference values as signed real number and not convert them to boolean because we want our model to not only learning the movement but also the magnitude of the movement. When the next day price is going up, we want to know how much it is, likewise if next price is going down. As a result, our engineered dataset is 3-dimensional with each one of the features hold a sequential property.

Let i be the trading day index, P be today closing price, Q be next day closing price and D be the difference between next day and today's price. We engineer the new features as follows:

$$P = \{p_1, p_2, p_3, \dots, p_{i-1}\} \tag{1}$$

$$q_i = p_{i+1}, q \in Q \tag{2}$$

$$d_i = q_i - p_i = p_{i+1} - p_i, d \in D \tag{3}$$

We utilize holdout method to partition the dataset into 3 parts, which are train, validation and test set, with each portion is 80%, 10% and 10% respectively. We build and train the predictor using train set and observe the result for convergence using validation set. Finally, we test our method and analysis the result using test set, to represent data that have not seen before by the predictor.

B. Experimental Setup

Recently, deep learning gains better performance in various tasks compared to more conventional methods due to improved computational power, breakthrough abilities to learn non-linear

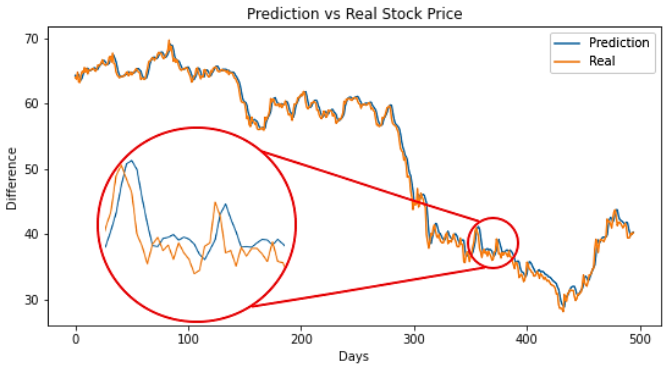

Fig. 1. Univariate LSTM prediction of the HSBC HOLDINGS closing day price.

connection and feature extraction in high-dimensional space. LSTM in particular, designed to learn the sequential observations from previous time steps and capture future trends. Its feedback connection not only can process a single data point, but also entire sequences of data. LSTM introduces the memory mechanism to enables long sequence dependency between time steps [9]. Sequence or time-series forecasting like stock price and stock movement prediction is greatly benefit from the use of LSTM.

In this study, 2 experimental setups are tested and compared, namely univariate and multivariate manner. We use simple univariate LSTM using 3 sequential data points that we have in our modified dataset as the baseline. It represents any method that use the same setting in their experiment. Although we utilize deep learning who has the capability to take multidimensional input, univariate scenario is a good way to represent shallow learning methods with limitation in their input vector shape. In this scenario, we examine a learning method that solely based on the raw sequential property, i.e., today and next day stock price.

In stock price prediction, a simple univariate LSTM without any complicated adjustment like regularization, with only today stock price alone as its dataset, already achieve a good fit, with very low error. In Fig. 1, stock price prediction for HSBC HOLDINGS in Hong Kong's stock index, from 495 days prior to the end of year 2020, shows a very promising result, with 1.002840 mean squared error. However, as we can see in a zoomed region of the graph, the prediction approximates the real price with delayed pattern, means that although the graph is very close, it does not match for the same day trading. Moreover, there are known evidence that for any given trading day we will likely have opposite direction of price movement. This is reinforced by the movement prediction based on this graph, it is only 0.56, not very different than random guessing. It will not good enough to earn lots of profit in real trading.

In addition, we also use the generated price difference data in this univariate scenario to give a broader view about this baseline measurement. The result is almost the same, it only gives random guessing range prediction. We presume that learning solely based on the individual sequential property only capture the temporal feature, connection between current and previous values and exclude the spatial feature, the other property that might present in the surroundings. The model only cares about the statistical values of the input sequence when learning how to minimize error. This particular goal is well achieved, shown by the low error value, but the other goal, detecting price movement, is fall miserably due to lack of movement sequences to be learn simultaneously.

Deep Learning known for its ability to take input in almost any shape. This advantage can be used to create a mechanism how a sequence learning method, let say LSTM, learns not only from the temporal property but also from the spatial property as well. Normally, to predict next value in a long sequence, we need to consider the temporal property given by the historical factor in the sequential feature. The model will find the sequential pattern given by the training set as its knowledge and then use it to predict the next value. Since our augmented data is multidimensional as in (1), (2) and (3), we believe that the spatial properties are beneficial to build a robust stock movement prediction model.

Our proposed method is different from others in term of managing multi-features in building model. Mostly, all previous works mentioned previously use these features in a conventional supervised manner [4][5][8]. They accommodate all features and pass them to the training pipeline without considering the relationship of these features to each other. In machine learning, more features mean more possibility to get better or worse, not to mention the impact caused by the computational cost. Surely, they have curate and evaluate the pros and cons in features selection in their studies to make the most out of spatial properties among features.

We take the same direction as the previous researchers did but with different feature selection, and then make prediction of stock price and stock movement in one single pipeline. We emphasize the power of temporal property in the sequence P and Q with our generated spatial property, i.e., price difference D. Our proposed framework is trying to overcome the delayed pattern problem depicted in Fig. 1. In multivariate, we try to find the prior knowledge in the previous sequence, termed temporal features, and at the same time, expand the search area to the surrounding environment, let say the next feature found, and use them to predict the next value.

We do not regard prediction of stock movement as a supervised binary classification task since it will only predict the trends, it might be similar to the real movement but not align at any given day. Besides, we still have to build another model to predict the stock price. The goal of our proposed framework is to make predicted stock prices as close as possible to the real prices without any delayed movement pattern. We calculate the ups and downs movement based on the predicted prices, and not by building a new model, all task in a single pipeline. We formulate our predicted value $\hat{Y}$ at day i as follows:

$$(\hat{y}_i \approx y_i) \wedge (\hat{y}_i^{movement} = y_i^{movement}) \tag{4}$$

There are two ways to determine $Y^{movement}$, i.e., look-behind and look-ahead price difference. At any given day i, we want to determine the price movement, if we compare today's price against previous day's price, we term it as look-behind price difference. This is not very intuitive since what we are looking for is next day price, but for benchmarking purpose we will accommodate this scenario. Look-ahead price difference seems more suitable for our problem since we consider next day price to determine the price difference. Every data point in the sequence will have a look-behind price difference feature except the first, and vice versa, the last data point will have no look-ahead price difference feature. Whichever scenario that we use in our model, our input sequences will be 1 less than overall sequences from the dataset.

Although we already have equation for the price difference D, in practice we need to extent the formulation in (3) to get $Y^{movement}$. We use binary to indicate movement, 0 for down and 1 for up. In look-behind scenario, we determine stock price movement as follows:

$$y_i{}^{movement} = \begin{cases} 0, & p_i < p_{i-1} \\ 1, & p_i \geq p_{i-1} \end{cases} \quad (5)$$

In line with (5), here is the stock price movement in look-ahead scenario:

$$y_i{}^{movement} = \begin{cases} 0, & p_i > p_{i+1} \\ 1, & p_i \leq p_{i+1} \end{cases} \quad (6)$$

We put it all together in a multivariate LSTM graph. The spatial feature, which is price difference, is embedded to the temporal feature, which is stock price, to capitalize the predictive power of the resulting model. Our input is a 2D tensor with time steps 60 which consist of 2 sequences. We use simple LSTM layer with 60 nodes and Dropout 0.2 to avoid overfitting [10]. Since this is basically a regression task, we utilize MSE (Mean Squared Error) as our loss function and Adam optimizer with learning rate 0.001 to control our model convergence [11]. All the comparative models were implemented in slightly the same setting, except the Look-ahead scenario.

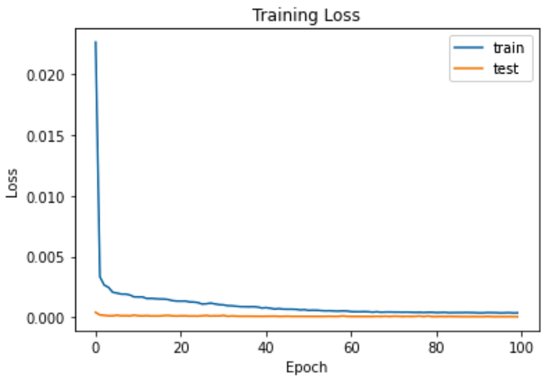

Fig. 2. Look-ahead PDEM-LSTM Training Loss

Price difference in Look-ahead scenario obtained by comparing today's price with next day's price. We can easily get the next day value from our dataset, but this is not applicable in the real world. Let say today is Tuesday and we need to predict Wednesday's price. We create a sequence consist of price historical data and its Look-ahead price difference. The problem is that we cannot calculate price difference for today because we have not known Wednesday's price yet. Just like we mentioned before, the last data point in Look-ahead scenario do not have price difference. In fact, Wednesday's price is the goal that we trying to predict. So, we need to make adjustment in the implementation. We shift the input sequence to exclude the last data point, which is Tuesday's price. Monday's price will be the last data point in our input sequence to predict Wednesday's price, there are 2 days gap. However, we need to maintain 1 day gap in our prediction to align with our input sequence. We employ sequence to sequence prediction, 60-time steps input sequence will be used to predict 2-time steps result. We adjust the final layer of our model with 2 nodes fully connected layer. The first node will contain Tuesday's predicted price that we will ignore since we already have Tuesday's real price. And the second will contain Wednesday's price, the one that we record to calculate the stock movement accuracy.

IV. Result and Analysis

A. Metric

Our proposed framework is actually a regression model, so the suitable performance metric is MSE. However, the price predicted is only an intermediate goal to be further used for determining its movement each day, as in (6). The MSE is used to ensure that our price prediction is acceptable, not too far away from the actual stock price. Afterwards, Movement Accuracy (MA) is calculated by comparing $y_i{}^{movement}$ and $\hat{y}_i{}^{movement}$ as follows:

$$MA = \frac{\sum_{i=1}^{n}[\hat{y}_i^{movement} = y_i^{movement}]}{n} \times 100\% \quad (7)$$

We utilize Iverson bracket notation to count how many predicted movements match the real stock movements, all the way from day 1 to day n.

B. Model Convergence

We observe the learning curve of our model using loss graph at training phase. The train and validation/test loss should converge after a certain number of epochs. We need to ensure that our model has a decreasing loss and constant slope by the end of the training process. Fig. 2 shows our train loss decrease as intended and converge along with the test/validation loss at the end of the training phase. Ideally, the validation loss will be higher than the train loss, unlike our loss graph, but our model is considered acceptable since we do not tune the hyperparameter excessively. After all, both losses are converged eventually. The gap between these losses describes the generalization ability of the model. There is no sign of overfitting, our model is ready to predict data that has never seen before.

Fig. 3 depicts Look-ahead PDEM-LSTM for 0005.HK stock price. Our proposed model manages to successfully predict the stock prices with very low error. Although the error considered lower than the other models, but the value difference is extremely small, so it does not upgrade the stock price prediction significantly. The prediction graph matches the real price perfectly, no sign of excessive delayed movements. The numbers shown in Table I confirm that while maintaining price prediction as precise as possible, Look-ahead PDEM-LSTM surprisingly achieve a very good accuracy on stock movement prediction as well. All of this is obtained with single integrated pipeline, no extra computational cost. Furthermore, this great performance also applies to other datasets that we explore in this study.

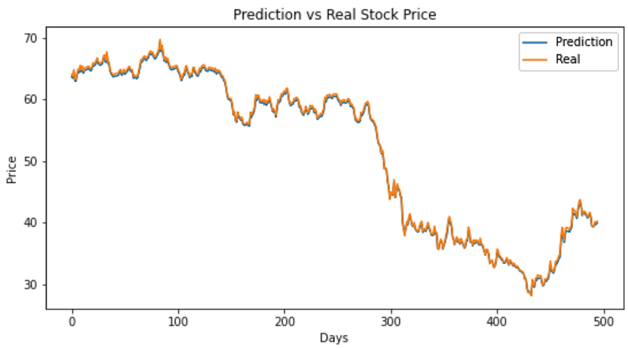

Fig. 3. Look-ahead PDEM-LSTM prediction of the HSBC HOLDINGS closing day price.

C. Comparative Study

Stock movement prediction is considered a very challenging task. A small portion of improvement can lead to a significant potential profit. In this field, the accuracy of 56% is generally reported as a satisfying result [12]. Most of the previous studies accommodate as many factors as possible in their models, namely fundamental analysis, technical analysis, political situation, social media comments and more. This complex information truly influences the model performance. This is the best practice to build an end-to-end production level application. Meanwhile, this research does not go to that direction. We focus on maximizing the use of temporal and spatial features resided in the stock market datasets. We may not have an end-to-end production level application, but we still give a noteworthy contribution in this research area.

Our main focus is the improvement that our proposed framework can achieve with only using simple model. Table I exhibits the overall performance conducted in our experiments using 0005.HK stock price. Sequential properties in price data alone does not perform well enough to predict stock movement although the prediction error considered very low as shown in the first and second row. The movement accuracy score is in par with most of the previous studies. The third row tells the performance of our engineered feature in univariate manner to do the same task. Although we make an exception, the MSE noted is not related to stock price but price difference instead. Anyway, the movement accuracy is slightly better than other univariate models.

The combination of temporal and spatial features in multivariate manner makes a notable improvement. It is a surprise that with more data involved, the results are very different. Look-behind scenario struggle to perform better even get the worst performance. The idea to engineer price difference using previous day's price does not affect movement accuracy at all. The accuracy of 55% is considered no improvement whatsoever, it is still on the same level as other previous studies. This is allegedly because in Look-behind scenario, the price difference does not describe the future values that our model should train on. We determine the price difference by looking at the past, the previous day's price. Therefore, the model learning direction points to the opposite direction, generate more delayed movements despite the low price MSE. Our model should learn the future, it must be taught using future directed features, price difference should consider the next day and not the previous day. Look-ahead scenario is appropriate for this task since it is created with future direction in mind.

TABLE I. MOVEMENT ACCURACY ON VARIOUS DATASET USING SVM

Dataset	Movement Accuracy	Price MSE
S&P 500: ABT	46.31%	7.241039
DJIA: WMT	46.51%	11.67176
NASDAQ: INTC	48.70%	2.013702
NYSE: BA	44.91%	80.23296
HKEX: 0005.HK	55.42%	0.904179

LSTM is known for its ability to deliver previous knowledge from the past sequence and use it to predict current value based on current input sequence. As a baseline, we build another predictor with slightly the same approach to benchmark our proposed method. We utilize SVM with RBF (Radial Basis Function) kernel. This kernel accepts two parameters, the first is gamma which defines how far the influence of a single example reaches, means that this parameter act just like the short-term memory in LSTM in carrying previous knowledge to predict the current situation. The second parameter determines the margin being accepted as a regularization function in SVM.

Table I shows the overall results on various datasets that we examine in this research. We can draw several conclusions based on these numbers. In term of MSE, SVM did a good job in predicting the stock price although the Price MSEs are no match against more advance method like LSTM. In term of movement accuracy, SVM experienced delayed pattern, so the result is slightly equal to random guess, somewhat the same as experienced by Univariate LSTM.

TABLE II. 0005.HK MOVEMENT ACCURACY AND PRICE MSE

Model	Movement Accuracy	Price MSE
Univariate – Today's Price	56.48%	1.002840
Univariate – Next Day's Price	56.48%	1.424277
Univariate – Price Difference	64.98%	0.544506*
Look-behind PDEM-LSTM	55.87%	0.070927
Look-ahead PDEM-LSTM	**97.77%**	**0.070927**

a. Third row represents Price Difference MSE, not Price MSE*

TABLE III. MOVEMENT ACCURACY ON VARIOUS DATASET

Dataset	Movement Accuracy	Price MSE
S&P 500: ABT	97.18%	0.278237
DJIA: WMT	97.99%	0.265340
NASDAQ: INTC	98.99%	0.017432
NYSE: BA	96.38%	4.184101
HKEX: 0005.HK	97.77%	0.070927

Last row of Table II indicates the best performance of our proposed model without sacrificing price MSE too much. It is harder to implement than the others, but the result is very promising. The accuracy of 97% is a huge improvement and everyone should consider this approach to predict stock movement. This simple framework can be attached easily to whatever model that one already has, and it is ready to be developed further to become production level application.

Generalization ability is very important since no one knows what kind of data will be processed by our model. A good model must generalize better even with a broad kind of data throw at it. We have to ensure that our model behave as expected, not only good in predicting test set but also good in predicting arbitrary input. Table III shows the performance of our model against data from 5 stock issuers in 5 stock indices. The performance remains excellent in all datasets that we use, this indicates that our model has a sufficient generalization ability. It is simple, robust, yet easy to implement.

V. CONCLUSION

In this paper, we proposed Price Embedded Multivariate Long Short-Term Memory to predict stock movement using only feature that has sequential property in it. The main contributions of this work can be summarized as follows:

1. Our engineered price difference feature makes a huge impact on the movement accuracy using Look-ahead scenario and embed them in Multivariate LSTM predictor.
2. Look-ahead PDEM-LSTM performs surprisingly good at predicting stock movement while maintain predicted price error at an acceptable level.
3. Our model not only perform great but also generalize better against all datasets that we use in this research.

One of the reasons for this good performance is that our datasets contain sufficient amount of data, from 2000 to 2020. We have not test against stock data from new companies which definitely much less. In the future, we will examine our model using short sequences so it will be useful to be implemented using relatively newly listed companies which only has a very little stock data.

ACKNOWLEDGMENT

The author would like to thank Google LLC for providing a helpful IDE. Most of the work in this research is conducted on Google Colaboratory.

REFERENCES

[1] K. Adam, A. Marcet, and J. P. Nicoli, "Stock Market Volatility and Learning", The Journal of Finance, Vol. 71, pp. 33-82, 2016

[2] M. J. S. de Souza, D. G. F. Ramos, M. G. Pena, V. A. Sobreiro, and H. Kimura, "Examination of the profitability of technical analysis based on moving average strategies in brics", . Examination of the profitability of technical, Vol. 4, 2018

[3] C. Yang, J. Zhai, and G. Tao, "Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory", Mathematical Problems in Engineering, 2020

[4] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," Expert Systems with Applications, vol. 42, no. 1, pp. 259–268, 2015

[5] X. Zhang, S. Liu, and X. Zheng, "Stock Price Movement Prediction Based on a Deep Factorization Machine and the Attention Mechanism", Mathematics, 2021

[6] Y. Xu, and S. B. Cohen, "Stock Movement Prediction from Tweets and Historical Prices", Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pp. 1970-1979, 2018

[7] Y. Shynkevich, T. M. McGinnity, S. A. Coleman, A. Belatreche, and Y. Li, "Forecasting price movements using technical indicators: investigating the impact of varying input window length," Neurocomputing, vol. 264, pp. 71–88, 2017

[8] R. Sawhney, S. Agarwal, A. Wadhwa, and R. R. Shah, "Deep Attentive Learning for Stock Movement Prediction From Social Media Text and Company Correlations", Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 8415–8426, 2020

[9] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory," Journal of Neural Computation, Vol. 9, 1997

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting" J. Mach. Learn. Res. 15, pp. 1929–1958, 2014

[11] D. P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization", International Conference for Learning Representations, 2015.

[12] T. H. Nguyen and K. Shirai, "Topic modeling based sentiment analysis on social media for stock market prediction", In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Vol. 1, pp. 1354–1364.