Iranian Journal of Economic Studies, 9(2) 2020, 527-548

Iranian Journal of Economic Studies

Journal homepage: ijes.shirazu.ac.ir

The Role of Feature Engineering in Prediction of Tehran Stock Exchange Index Based on LSTM

Amin Aminimehr[a*], Ali Raoofi[b], Akbar Aminimehr[c], Amirhossein Aminimehr[d]

a. Department of Management, Ershad Damavand University, Tehran Branch, Tehran, Iran.
b. Faculty of Economics , Allameh Tabatabaei University, Tehran, Iran.
c. Faculty of Accounting, Management and Economic, Payame Noor University, Tehran, Iran.
d. School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

Article History

Received date: 21 February 2021
Revised date: 17 May 2021
Accepted date: 16 June 2021
Available online: 10 July 2021

JEL Classification
C32
C45
C52
C53
C58

Keyword
Tehran Stock Exchange
Price Prediction
Deep Neural Network
Feature Engineering
Knowledge Extraction

Abstract

In this research, the impact of different preprocessing methods on the Long-Short term memory in predicting the financial time series was examined. At first, the model was implemented on the Tehran stock exchange index by utilizing the Principal Component Analysis (PCA) model on 78 technical indicators. Then, the same model was implemented by the advantage of the random forest to select features rather than the PCA to extract them. In the next step, other technical strategy dummy variables were added to the model to examine the changes in its performance. Finally, two deep learning methods with the advantage of only target lags were deployed to compare the accuracy to the other models. The first deep model was plain but the second one was with the advantage of the Wavelet denoising process. The results of the MSE, MAE, MAPE, and R2 score on unseen test sequences showed that applying the Long Short-Term Memory with its own deep feature extraction procedure and the wavelet's denoising process leads to the best accuracy in prediction of the Tehran stock exchange index. Finally, the Diebold Mariano test exposed a significant difference between the accuracy of the best model and the rest. This result implied that although the application of deep learning gains accurate results, it can be alleviated by feeding the model with creatively extracted and denoised features.

Highlights

- The performance of different knowledge extraction methods on financial time-series data.
- Using technical indicators as input variables of deep neural networks.
- Enhancing the impact of technical strategies in the stock price prediction.
- The role of the preprocessing features for training neural networks.

* aminaminimehr@outlook.com
DOI: 10.22099/ijes.2021.39877.1739

1. Introduction

Stock price prediction has always been a concern to investors due to its commercial benefits and the debates upon the Efficient Market Hypothesis (EMH). Within research works held on this topic, many scholars have incorporated the advanced linear and non-linear methods to predict the price. Besides, the quality of the data fed into the prediction models has not been adequately discussed, while the stock market is well known for its noisy data because of human interactions. Fortunately, with the advanced developments in programming languages, especially Python and its specific libraries in data science such as Sikitlearn, Tensorflow, Scipy, etc, running various kinds of deep learning models has become extremely convenient. That is why the main concern of this paper is not to seek a new model or to create one, but to manage a unique methodology, consisting of preprocessing methods, in order to achieve the optimal procedure of the stock price prediction. In this research, attempts were made to improve the accuracy of a specific machine learning model step by step, by modifying the procedure of the feature selection, feature extraction, and denoising. Meanwhile, advanced deep learning methods were deployed to see that the features extracted by the model itself can more precisely predict the price. Moreover, there is an emphasis on the feature engineering and procedure of the pre-knowledge extraction. The main question is to find which pre-knowledge extraction methods result in more accurate predictions.

Artificial Neural Networks (ANN) are a class of complex computing models inspired from the human neural network and a class of Machine Learning (ML) algorithms. They have been used to solve various problems from different subjects for about 80 years McCulloch and Pitts (1943). ANN is an experience-based model, which is on gaining by the available data propagating through stacked layers. This can be analogized to the way human gains experiences and uses them in future situations. This kind of complicated model can fit nonlinear high-dimensional problems with an unprecedented amount of error. It has been proven that ANNs, as a black box, could perform an effective feature detection and relation interpretation and overcome the traditional statistical models.

There are many applications of AANs in finance (Abbasian et al., 2019; Sayadi & Omidi, 2020); for instance, Nicholas et al. (1994) and Odom and Sharda (1990) used ANN to model stock performance and bankruptcies, respectively. From the other point of view, ANNs have become popular in recent years because of using Recurrent Neural Networks (RNNs) by big firms in order to solve many problems in image and speech recognition. Every year attempts are increasing to apply deep learning to the stock market forecasts because using the artificial intelligence (AI)-based models lead to better performance compared to the conventional statistical models when the characteristics of the financial data are studied. They are specifically employed to predict the stock price and return and even Sharpe ratio, which can be used for asset prioritization and portfolio management (Cipiloglu Yildiz et al., 2020; Fernández-Rodríguez et al., 2000; Tay & Cao, 2001; Vukovic et al., 2019; Wang et al., 2020). In further steps, the integrated models were utilized to solve many problems. For example, Liu et al. (2017) utilized the Convolutional Neural Networks (CNN) and LSTM to develop a framework and compared their quantitative and momentum strategies to each other in their gained returns. The results obtained showed that their framework can achieve a better return in comparison with the momentum strategy and market benchmark. In another research conducted by Zhao et al. (2017), a time-weighted function was added to an LSTM neural network, and the prediction results concerning the Capitalization-weighted stock market index (CSI 300) surpassed those of the other models. Finally, Jin et al. (2020) took use of investors' emotional tendency through the sentiment analysis and the introduced empirical modal decomposition (EMD) combined with the LSTM to obtain more accurate stock forecasts.

In terms of volatility and risk modeling, machine learning applications play a vital role. In a research project in 2018, a hybrid model of the GARCH combined with the LSTM was proposed to predict the stock price fluctuations (Kim & Won, 2018). Huang and Guo (2020) developed an early warning system using a fuzzy approach combined with a support vector machine, and their model could deal with noisy data better than the earlier version of the twin support vector machine.

Financial time series, especially those that are generated in the stock market, are the results of the elaborate factors affecting them mainly under the topics like macroeconomic factors, corporate finance, market movements and patterns, and even behavioral finance factors. These factors and some of their subfactors are presented in Table 1. Each of them consists of numerous subfactors which individually were proven to be statistically significant in recent literature for defining the market movements. Although the eminent Nobel prize holder, Fama, believed that the stock prices in efficient markets cannot be predicted by analyzing the historical data (Fama, 1970), attempts are still made in order to predict the stock market using the historical patterns.

Table 1. Variables affecting the market price

Macroeconomic factors	**Corporate finance**	**Market Factors**	**Behavioral Finance Factors**
GDP (Gross Domestic Product)	Capital Structure	Technical Indicators	Self Deception
GNP (Gross National Product)	Debt Ratio, Current ratio, property ratio, …	Technical Oscillators	Heuristic Simplification
Inflation	Financial leverages	Price Actions	Social Influence
Economic Growth Rate	Bankruptcy Risk	Support levels	Emotion
Price Level	Profit to Asset Ratio	Resistance Level	Risk Aversion
National Income	Expected Return	…	…
Unemployment Level	Manegerial performance		
…	…		

Source: *Authors' Compilation*

While studying many of these factors by the conventional statistical methods, scholars encountered several statistical and hardware limitations. The limitations are as follows: multidimensional correlations between features, losing a degree of freedom, nonstationarity, and nonlinearity. In order to have a better prediction, many researchers started using modern Data Science methods like deep neural networks. Furthermore, the financial time series are extremely noisy, which is mainly as a result of human interactions in the market. In fact, there are various approaches in Data Science and AI to deal with the aforementioned obstacles.

Moreover, studying the factors and subfactors mentioned earlier was not feasible due to the lack of data, efficient statistical models, and even the lack of computational infrastructure. However, with the advantages of recent developments in quantum computers, deep learning models, data science techniques, and vast databases in the 21st Century it is not unlikely to deploy extremely high dimensional models in finance. Furthermore, the study of these factors in addition to their commercial benefits can help researchers to find out which of these elements has more contribution in defining market movements. For instance, there has always been a struggle between technical analyzers and corporate finance analyzers on the subject of the factors defining the movement of the markets. Even today, behavioral finance analyzers are struggling with the traditional financial theories to define the market price and return.

The model implemented in this research was fed by two subfactors of Technical Indicators and Technical Oscillators (see Table 1). The novelty of this research is to additionally use specific trading strategies, which are extracted from the integration of the technical indicators and oscillators in the form of dummy variables, to check their impact on the accuracy. This kind of analysis is vastly utilized by professional technical traders to predict price movements.

In the following, there is a comprehensive description of the data and specific variables used to implement the suggested methodology in Section 2. Section 3 covers the methodology materials, including the feature engineering methods used. In section 4, the evaluation of the metrics utilized to compare the models and the statistical test is covered. Section 5 covers the experiments conducted and the results obtained from them. Finally, the research is ended up with section 6, stating the conclusions and further suggestions.

2. Data and Variables

The study period of this research is in the range of December-2008 to February-2021. The research sample was extracted from tsetmc.com with a daily time scale. The Tehran stock exchange index includes about 400 firms registered in 2021 and its market capital worth is about 500´000´000 billion Rials. In this study, the daily closing price of the Tehran stock exchange index was considered as the target value to be predicted. Initial feature data used in this research are trading volume, open, high, and low price. Moreover, many financial researchers used lags of target observations to predict the future (Fischer & Krauss, 2018; Paiva et al., 2019). Accordingly, the prices of the lagged target were also employed as the features. In order to adequately feed the LSTM model with long-time data, the whole data sequence is split into 85% training data and 15% test data.

Technical indicators and oscillators can be analogized to footprints of price. They form signals, which are either bounded like the Relative Strength Index (RSI), or unbounded like the trend-having simple moving average. Unlike fundamental analysis, which is based on the intrinsic value evaluation of security relying on earnings, revenue, profit margins, and macroeconomic factors, the focus of the technical discipline is on the price movements and trading signals in order to evaluate the temporary strength or weakness of security.

Technical analysis can be employed on any security with historical trading data. This includes stocks, futures, commodities, fixed-income, currencies, and other securities. In fact, technical analysis is more prevalent in commodities and forex markets where traders focus on short-term price movements.

The profitability of the technical indicators and their role in price prediction, portfolio optimization, and risk management are debated in academic and practical circles. Even though many traders and investors make use of analyses based on them, there is still a controversy about different academic kinds of literature of finance (Gehrig & Menkhoff, 2006; Stanković et al., 2015).

On one side, stands the eminent Fama who implies that the historical data-based technical analysis is completely against the widely accepted EMH. Along with that, certain studies were conducted to show that technical trading strategies cannot always provide an acceptable level of profitability; this results in many doubts about their widely used applications by traders (Fama, 1970; Fama & Blume, 1966; Jensen & Benington, 1970; Seelenfreund et al., 1968).

Several researchers have debated the presence of the EMH assumptions on different markets with different properties, yet there is not an explicit answer given. They mainly concur on the fact that the profitability of the technical analysis cannot be generalized. By studying the Eastern European market, they stated that the profitability of the technical indicators is in respect to the size of the market. They found that the historical data can be used to predict small and medium-sized capitalization markets (Todea et al., 2009; Todea et al., 2009; Metghalchi et al., 2012).

On the other end of the spectrum, some researchers believed that the technical analysis can predict very well in the large capitalized markets too (Fifieldet al., 2005; Heyman et al., 2012; McKenzie & Takaoka, 2007). By reviewing technical trading rules, Coakley et al. (2016) studied a wide range of technical trading approaches to check their profitability; their results implied that many approaches are significantly profitable up to 30% annually. In more specific surveys, trading strategies based on moving average's (MA) crossovers lead to decreasing the investment risk (Barbulescu, 2014); also, the strategies based on the optimized Moving Average Convergence Divergence (MACD) and Relative Vigor Index (RVI) indicators result in significantly increasing the investment profitability (Eric et al., 2009). Research held by Chen and Hao in 2018 also showed that the indicators such as the Exponential Moving Average (EMA), RSI, and Momentum Index (MoM) are correlated with the changes that occurred in the stock market (Chen & Hao, 2018). Furthermore, in terms of using the technical data in the neural network models, Kara et al. (2011) selected ten technical indicators as the definitive input features.

In summary, the technical analysis provoked many debates over the past few decades, but its profitability is highly dependent on the presence of initial assumptions of the EMH theory within each market. As a result, the data science's role is to find the most contributing features on each market, so studying the characteristic of the financial time series should be specified to a particular market, index, and even particular period.

3. Methodology

3.1 LSTM Networks

The LSTM networks were introduced by Hochreiter and Schmidhuber (1997) as an improved method to learn the sequential patterns rather than the simple RNNs. The LSTM networks are a specific type of RNNs, which is utilized with the memory cell, so it has the advantage of memorizing information over a long period compared to the RNNs (Fischer & Krauss, 2018; LeCun et al., 2015). Sang and Di Pierro (2019) showed that the LSTM can be effective in improving the performance of the trading algorithms in financial data. In research conducted by Graves and Schmidhuber (2005), they showed that the LSTM networks are better at memorizing long-term knowledge, and have no problem regarding the vanishing gradient. Furthermore, Jiang et al. (2019) compared a simple RNN to an LSTM and realized that the latter can better forecast the financial time series.

The structure of the LSTM network is shown in Fig. 1. The LSTM networks have a simple but effective structure. Each cell is individually a system performing a precise function. Without considering the inner sub-functions of the LSTM cell, the main function is to get three inputs simultaneously:

1. The input vector (input from current time step)
2. The memory passed from the previous time step
3. The output of the previous layer

and to pass two information elements simultaneously:

1. The memory (cell state) of the current block
2. The output vector of the current block.

There are three gates inside an LSTM cell: 1) input gate, 2) forget gate, and 3) output gate. The input gate is in charge of deciding which data can be involved in the cell state; the output gate is activated to prepare the output from the cell state, and the forget gate decides which data should be dismissed from the cell state.

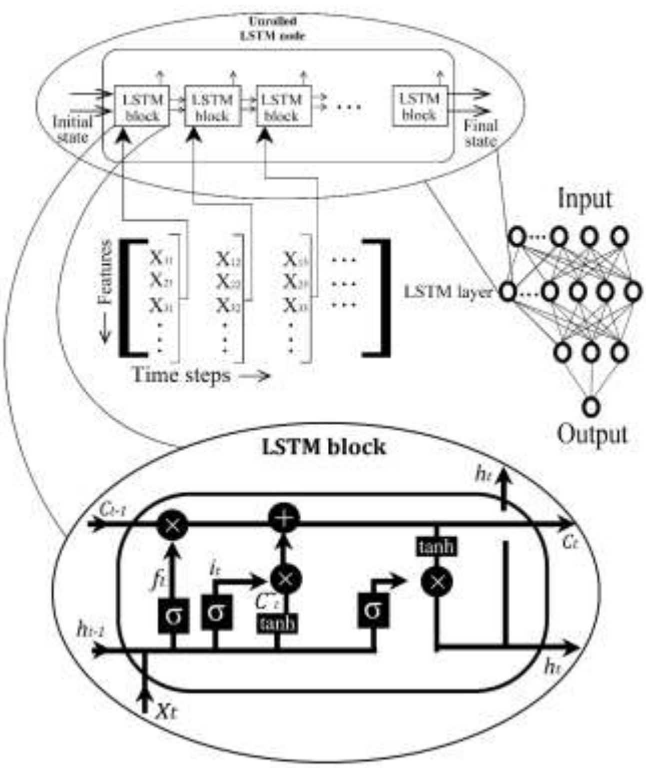

Figure 1. Structure of LSTM cell
***Source**: Authors' Compilation*

The described procedure inside each LSTM cell is based on the specific sub-functions described in formulas numbers 1 to 6.

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

$$\mathrm{i_t} = \sigma\left(\mathrm{w_{i^2}}\left[\mathrm{h_{t-1,}x_t}\right] + \mathrm{b_i}\right) \quad (2)$$

$$\mathrm{O_t} = \sigma(\mathrm{W_0} \cdot [\mathrm{h_{t-1}}, x_\mathrm{t}] + \mathrm{b_0}) \quad (3)$$

$$\mathrm{c_t} = \tanh(\mathrm{W_c} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_c}) \quad (4)$$

$$\mathrm{g_t} = \tanh(\mathrm{x_t U^g} + \mathrm{h_{t-1} W^g}) \quad (5)$$

$$\mathrm{h_t} = \tanh(\mathrm{c_t}) \cdot 0 \quad (6)$$

3.2 Principal Component Analysis (PCA)

The principal component analysis is an approach to reduce the data dimensionality and is within the class of the feature extraction methods. The PCA forms the basis in many ways for multivariate data analysis and is a methodology through which an optimal number of series are generated to include the maximum amount of knowledge and least dimensionality. Wold et al. (1987) recommended the data pre-scaling to unit variance, which generally causes a better performance of knowledge retainment in the PCA. In an overview, the PCA provides various services such as simplification, data reduction, modeling, outlier detection, variable selection, classification, prediction, and unmixing. The Principal Component Analysis generates new features, which are the linear combination of the initial features; also, it has a procedure of matrix diagonalization through itself. Through a PCA, the maximum amount of data variance is saved with the least amount of data series. The PCA proved to be successful for complex spaces, and various researchers have used this technique for denoising and dimensionality reducing in the high dimensional data. It was shown that the PCA has a positive impact on the classification algorithm of intrusion detection (Keerthi Vasan & Surendiran, 2016). In this study, the PCA is used in particular experiments to reduce the dimension of the features, which is generated from applying technical indicators on close, open, high, low price, and trading volume.

3.3 Random Forest (RF)

The RF is a class of ML algorithms, and they can be used to select the most contributing features in a prediction problem. Tyralis and Papacharalampous (2017) compared two sets of time series in their one-step-ahead prediction model to introduce an optimal set of predictor variables using the Random Forest. The Random Forests seem to provide reliable results as they are easily interpretable and less often overfit. The RF interpretability is because of its structure in such a way that the importance of each variable can be conveniently derived. The RF can provide two services. As it is categorized as an embedded method, it combines the qualities of the features and the wrapping method. The random Forests feature selection can be performed through algorithms having built-in functions. Here are some advantages of the RF algorithm:

- They provide relatively precise results
- They provide a good generalization

- Their results are easily interpretable

The random forests are built from several decision trees. Each of the trees is only presented by several random samples, which are not presented in the other trees. This guarantees that the trees are decor-related and thus less probable to overfit. The procedure of the feature selection in the random forest is by the rule of similarity and difference. In other words, the model has nodes dividing the data into 2 sets. Each set hosts the observations with maximum similarity among themselves and maximum difference to the alternative set. In this research, the random forest is applied to S&P 500 features, and those features with greater importance than mean are selected to implement the model.

3.4 Wavelet Transforms

The wavelet transforms are vastly used by scholars in order to denoise various types of data. The wavelet approach and the frequency discretization can discretize within various time intervals. This capability of wavelet is a further step in comparison with furrier transform discretization. In another word, a wavelet is a function that can discretize a sequence of time series in to scale components considering time and frequency. This discretization approach was used by Sharif et al. (2020) to analyze the connection between the recent Coronavirus spreading with oil price volatility shock. As financial time series are extremely noisy due to many factors, wavelet is used in financial predictions more. Chang et al. (2019) used the wavelet transform to stabilize their features in their hybrid LSTM model to predict the Electricity price. Lu (2010) also employed the advantage of the wavelets to denoise the features fed to their neural network model that predicts the stock price. This helps a machine-learning algorithm to face a simpler form of the data fed into in order to achieve more accuracy in training and prediction. Li and Tang (2020) also proposed a model combining the wavelet and the Gated Recurrent Unit (GRU) network to predict S&P 500 price, and their model had the least error in prediction comparing to the ARIMA and multi-lag SVR.

4. Evaluation and statistical test

To evaluate each experiment, MAE, MSE, MAPE, and R^2 scores were used. The formulas for each of the scores are mentioned in formulas numbers 7 to 10.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |A_t - F_t|}{n} \tag{7}$$

$$\mathrm{MSE} = \frac{1}{n} \sum_{t=1}^{n} (A_t - F_t)^2 \tag{8}$$

$$\mathrm{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \tag{9}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{n} (A_t - F_t)^2}{\sum_{t=1}^{n} (A_t - \bar{A})^2} \tag{10}$$

where $\boldsymbol{A_t}$ is the actual value and $\boldsymbol{F_t}$ is the fitted value.

Furthermore, the Diebold-Mariano (DM) statistical test was applied to verify the difference of accuracy on each pair of the methods introduced in this research. This test takes in sequences of out-of-sample time series, an actual value, and two individual predictions, and finally tests the difference of the predictions established upon the target value. The null hypothesis is the condition at which the expected difference between the two predictions loss is statistically zero, keeping in mind that the loss differences can be related to any given loss function. The DM statistics are calculated as shown in formulas numbers 11 and 12:

$$\mathrm{DM} = \frac{\bar{d}}{\sqrt{\frac{2\pi \hat{f}_d(0)}{n}}} \tag{11}$$

$$\bar{d} = {}^{1}/_{n} \sum_{t=1}^{n} (d_t) \tag{12}$$

where d_t is the loss differential according to any function selected for the test, such as MSE or MAE, etc. And $\hat{f}_d(0)$ denotes the spectral density of the loss differential at frequency 0.

5. Experiments and Results

This research was conducted to introduce a sequence of techniques in order to improve the process of stock price prediction. In another word, the aim was to take leading steps towards maximizing knowledge extraction from the financial time series. There is a very important question that which type of feature engineering works best in predicting the financial time series in the stock market.

The LSTM neural network on its own performs a nonlinear feature interpretation within its training procedure. This feature learning process is compared to the other types of feature extraction and feature selection methods such as the PCA and Random Forest. Since financial stock values have a time-based sequential definition, there must be a time-based interpretation on each price value. To do so, the LSTM's own deep learning procedure is introduced as a feature extraction procedure in the two experiments held in this research.

First, the LSTM model is deployed to predict the target value by the technical features generated from open, high, low daily prices, including their daily trading volume. There are 78 generated Indicators and Oscillators in the following titles: Overlap studies, Momentum, Volume, Volatility, Price transform Indicators, and Oscillators. Meanwhile, historical partial correlation is studied on the target value up to 200 lags and 95 percent confidence level. This shows whether there are signs of a historical linear impact on target value or not. Figure 2 illustrates the partial autocorrelation of the target value. It is worthy to note that the signs of significant linear relations between the target value and its lags are apparent as shown in this figure. The two dash lines above and below zero are the significance margins. Indeed, in places at which the P-value crosses the margins, the corresponding lag is considered as a non-zero influencer on the target value. It is visually significant even at lags higher than 100 that there are significant correlations in them, so all significant linear correlated lags are fed as features to the model.

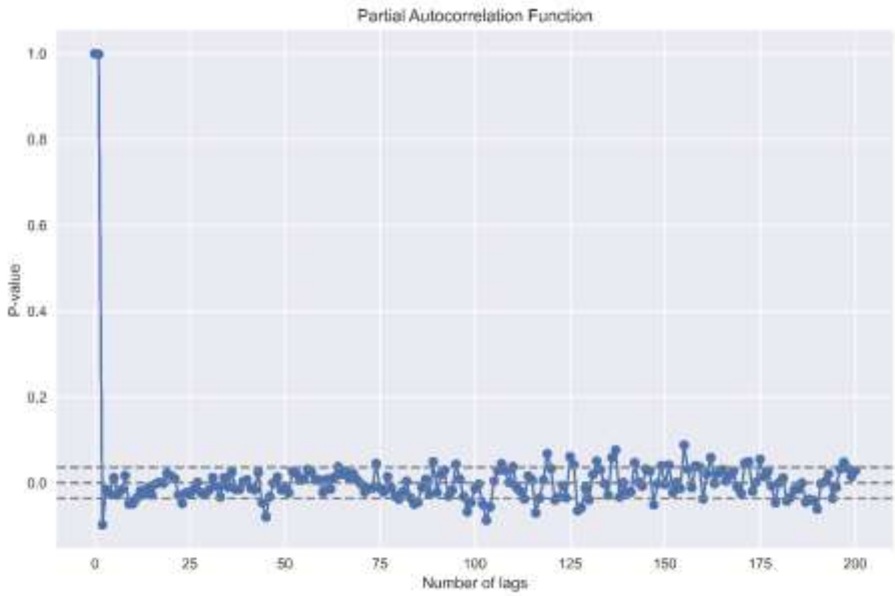

Figure 2. Partial Auto Correlation function on 200 lags of S&P 500 close price
Source: *Authors' computation*

Based on the literature, the intrinsic of the LSTM is that it can retain information over a very long history. Typically, the input gate has activation functions by which it gives rates of importance to the recent blocks cell state and the current input to the block. In other words, although each block of the LSTM does not meet the far historically significant lags at a point, it yet will meet a survived representation of historically significant lags of close price.

There are five experiments implemented in this research. The core model is constructed by 2 layers of the LSTM cell. In both hidden layers, units are utilized with the rectified linear unit (ReLU) activation function. The model is optimized with the Mean Absolute Error (MAE), but the Mean Squared Error (MSE) and the Mean Absolute Percentage Error (MAPE) are also recorded. The model was experimented with different learning rates on TensorFlow. Experiments showed that a large learning rate causes the gradient to be bounced and exploded. On the other side, a small learning rate prevents the loss function from converging to its global minimum and it gets trapped into a local minimum. Learning callback, with a high resolution of learning rate variation, is used to find the best learning rate in each experiment individually. Alongside, Different dropout layers were experimented in order to prevent overfitting problems and improve the robustness of the models. Srivastava et al. (2014) used this dropout procedure in 2014 and showed that the performance of the neural networks significantly improves. Finally, Adam optimizer, as a class of the Standard Gradient Descend (SGD), is used to train the models.

5.1 Application of LSTM and PCA

In the first experiment, 78 technical indicators and oscillators were fed to the PCA, and 17 features were extracted through them. Figure 3 shows the amount of cumulative explained variance within the extracted features. These 17 features can explain the 99.99% of cumulative variance. As a result, these sets of features are fed into an LSTM model with two hidden layers. 23 hidden units in the first layer, a drop-out layer, and 12 units in the second hidden LSTM layer constructed the architecture of the first experiment.

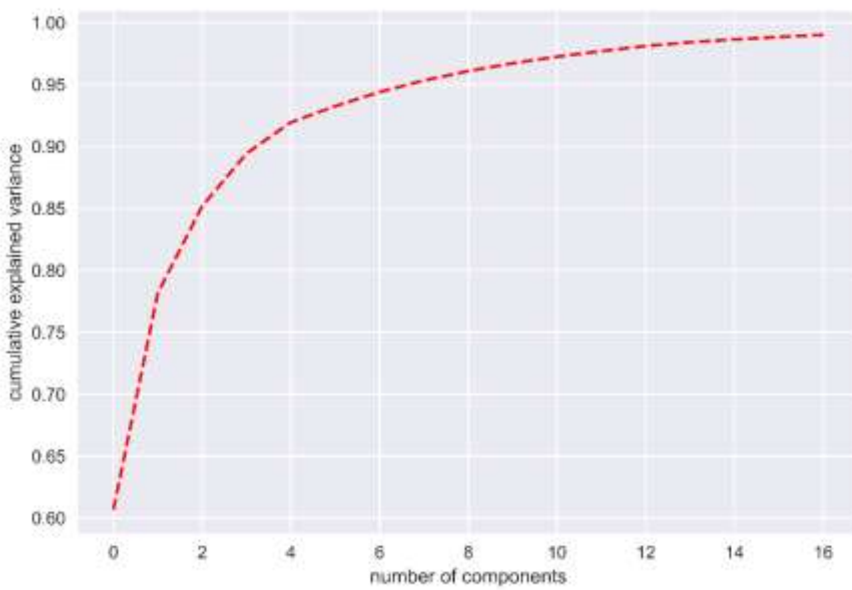

Figure 3. The PCA result on 78 features fed into
Source: *Authors' computation*

5.2 Application of LSTM and PCA (Technical Strategy Dummy Variables)

In the next experiment, there are four technical strategy features added to the model in order to maximize the knowledge extracted from this set of data. These features are a series of dummy variables in which labels one and zero express the presence and absence of a specific condition in technical indicators.

The logic behind the first Series of dummy variables is to give the label "one" when the 7 days moving average of close price crosses the 14 days moving average of close price and more than that This means the return of the recent 7 days is performing better than the entire recent 14 days, so the price will be more likely to grow.

The Second Series of the dummy variables are mainly generated to capture the small-scale volatilities by applying two queries on the features. First, the days with higher 2 days moving average than the 3 days one is selected and filled in a data table. On the composing data table, those days with a higher close price than 2 days moving average are labeled as “one”, so the other remaining days are labeled as “zero”. The logic behind this strategy is to predict the detailed ascending movements.

The third series of the dummy variables are generated by selecting those days with higher 14 days moving average than 21 days one. From the selected dataset, those with the Relative Strength Index (RSI) of lower than 50 are labeled as “one” and the rest remain “zero”.

In the next strategy, to capture small variations of close price, the days with higher 3 days moving average are selected compared to those with 7 days moving average. Furthermore, to have a long-term perspective simultaneously, the days in which 28 days RSI is lower than 50 are selected and labeled as “one”, and the rest remain “zero”.

In this experiment, the PCA extracts 20 components as input features for the LSTM model. Figure 4 shows the amount of variance explained through the number of the PCA components.

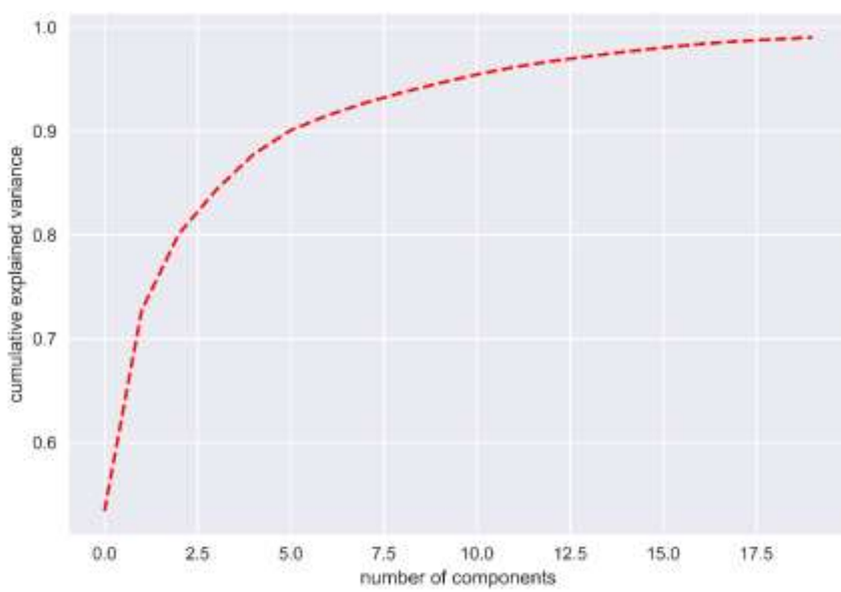

Figure 4. PCA result on 78 features and 4 dummy variables fed into
Source: *Authors’ computation*

5.3 Application of LSTM and Random Forest

In this experiment, the Random Forest classifier is used to select the most contributing features among the 78 technical indicators and oscillators. The difference between these two feature selection and PCA feature extraction methods is that the selected features in the RF are unchanged series, which are chosen from the initial features fed into the RF, but in the PCA, the extracted components are not exactly the same as the initial components. In other words, the PCA gives a series of values containing the knowledge of the whole feature dataset by studying the variance in them. The random forest with 80 units introduced 41 numbers of features, and in the next step, the selected features are fed into a model with 1 hidden LSTM layer, as well as a dropout layer. There are 55 units in the LSTM layer.

5.4 Application of Deep Learning

In this experiment, an application of deep learning was implemented on close price, but in contrast with the earlier ones, there is no need to extract technical indicators or select features, and all this procedure is held by the model itself. The LSTM models can distinguish and take the most contributing features to train the model. The procedure happening in a deep learning model is based on saving the most contributing features, as well as diminishing the low contributing features. This procedure is held by two activation functions in each LSTM cell, one in the input gate and the other in the forget gate. In this experiment, a sliding window of 12-time lags of the target value is fed into the model to extract features and give a prediction. Keep in mind that the features extracted by the network itself are not absolutely interpretable for humans, but for the network itself.

5.5 Application of Deep Learning and Wavelet

Finally, an experiment is held with the advantage of the wavelet decompose. Similarly, in this experiment, there is no need to extract the features like technical indicators. In this experiment, a 12-time step lag of close price is fed into the LSTM, but this time the feature values are denoised using wavelet. This is done to avoid the network from struggling with unnecessary complexity of data to predicate more accurately. To choose the best wavelet configuration, many experiments were done with different thresholds and finally, Debauchees 3 with threshold 1.01 was selected as the best.

Deep learning can achieve very good accuracy throughout its procedure; it is mainly because of plenty of features extracted through the process of training. Moreover, the application of deep learning with the advantage of wavelet leads to achieving better accuracy and that is because of a slight smoothness in the features fed into the model. The enhancive effect of wavelets is mainly as a result of human interactions in the market causing extra complexity in data. Table 2 reports the detailed final results of each experiment at the last epoch. The MAE experienced ups and downs through the experiments held in this research, but the application of deep learning and wavelet can achieve the best fitting and present the best accuracy on the data tested. Moreover, it stands for the application of the random Forest in decreasing the test MAE. The results of the MAE metric show that even though the application of the PCA reduces the dimension of the data and preserves the cumulative explained variance in the features, it cannot beat the application of deep learning. Not only this but also the application in which dummy variables were included either, cannot perform better than the deep learning methods. On the other hand, the impact of the technical strategies on the accuracy of the first model is completely significant. It is shown that the dummy variables provided the network with valuable knowledge lead to better results of the knowledge extraction.

Table 2. Experiment results

Experiment	Wavelet threshold / name	PCA componen ts	Test MAE	Test MSE	Test MAPE	Test R^2
Application of LSTM and PCA	-	17	199683. 4006	85329979 846.5103	30.247 11	0.8026
Application of LSTM and PCA (Including technical strategy dummy variables)	-	20	74311.9 960	13301610 870.4125	9.0298	0.9626
Application of LSTM and Random Forest	-	-	145762. 5472	46430104 828.9132	17.586 6	0.8411
Application of Deep Learning	-	-	51786.2 118	69609434 56.0665	4.9599	0.9828
Application of Deep Learning and Wavelet	1.0 / db3	-	47334.8 561	51291281 93.2003	4.8274	0.9849

Source: *Authors' computation*

The Diebold Mariano test was applied to the test accuracy of all 5 experiments to check the significance of differences in the prediction ability of the models. It is shown that the difference between the accuracy of predictions of all models and the application of deep learning and wavelet is significant with a 99% confidence level. Even though, the application of deep learning with wavelet obtains very close results to its next competitor, Diebold Mariano shows a significant difference between their predictions.

Table 3. Diebold Mariano Test

D-M test	Application of LSTM and PCA	Application of LSTM and PCA (Including technical strategy dummy variables)	Application of LSTM and Random Forest	Application of Deep Learning	Application of Deep Learning and Wavelet
Application of LSTM and PCA		P-value = 1.523e-39	P-value = 2.460e-09	P-value = 6.882e-54	P-value = 6.453e-56
Application of LSTM and PCA (Including technical strategy dummy variables)			P-value = 1.032e-28	P-value = 0.0001	P-value = 3.882e-12
Application of LSTM and Random Forest				P-value = 7.627e-37	P-value = 6.375e-46
Application of Deep Learning					P-value = 5.353e-27
Application of Deep Learning and Wavelet					

Source: *Authors' computation*

Figure 5 shows the fitted plots of the predictions performed from each experiment in this research on the data tested. It is manifested that the application of the LSTM and PCA has very unstable predictions, but the dummy variables result in an enhancive effect on them. Meanwhile, the predictions performed by the application of deep learning and wavelet showed a very close behavior to the actual price value of the target data.

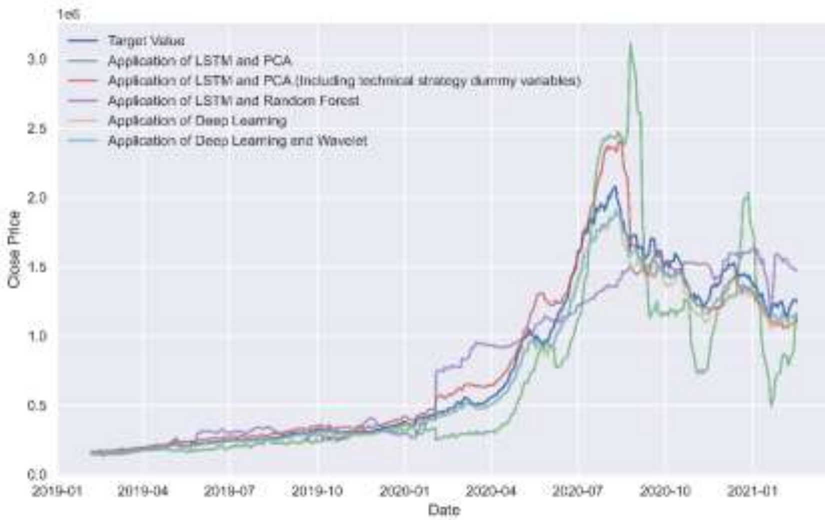

Figure 5. Prediction plot of validated methods on test sequence
Source: *Authors' plot*

6. Conclusion

Nowadays, neural network models have been improved vastly, and they have been used in the stock market very much. The fact in which the financial data have to go through different pre-processing methods to be suitable to train the network was considered very much before, but preprocessing is held by the model itself with the advantage of high computational powers and deep learning models. Keep in mind that the deep networks can be still improved by some denoising methods like wavelet. In other words, the wavelet can provide a higher quality of definitive features for the prediction models to gain accurate results. Moreover, this research concluded that the deep feature extraction of the LSTM is more promising and accurate than the other separate feature engineering tools like the PCA and Random Forest. Finally, this research showed that feeding the model with creative definitive features like technical strategy dummy variables can enhance the procedure of the price prediction. One of the main fields that should be studied through deep learning is an integration of technical indicators and oscillators to be fed into the models through a multivariate sliding window. Studying the combination of the linear composition of technical indicators and oscillators through deep learning may effectively help scholars and investors to find the turning points of the price movement with higher accuracy. Moreover, as mentioned earlier, there are many subfactors affecting the financial time series, and researchers can extend this field by engaging exogenous factors to increase the accuracy of the predictions. Specifically, there are many models introduced in which a group of those subfactors are studied, but building a unit mega-model to study and give an instant prediction considering all these factors is a desired vision in the field of AI and finance.

References

Abbasian, M., Sardar Shahraki, A., & shahraki, J. (2019). The modeling and comparison of GMDH and RBF artificial neural networks in forecasting consumption of petroleum products in the agricultural sector. *Iranian Journal of Economic Studies*, 8(1), 91-105.

Barbulescu, R. (2014). *Remus gabriel anghel (2013), Romanians in Western Europe. Migration, status dilemmas, and transnational connections [Book Review]*. Central and Eastern European Migration Review, 2.

Chang, Z., Zhang, Y., & Chen, W. (2019). Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. *Energy*, 187, 115804.

Chen, Y., & Hao, Y. (2018). Integrating principle component analysis and weighted support vector machine for stock trading signals prediction. *Neurocomputing*, 321, 381-402.

Cipiloglu Yildiz, Z., & Yildiz, S. B. (2020). A portfolio construction framework using LSTM-based stock markets forecasting. *International Journal of Finance & Economics*, n/a(n/a).

Coakley, J., Marzano, M., & Nankervis, J. (2016). How profitable are FX technical trading rules? *International Review of Financial Analysis*, 45, 273-282. doi:https://doi.org/10.1016/j.irfa.2016.03.010

Eric, D., Andjelic, G., & Redzepagic, S. (2009). Application of MACD and RVI indicators as functions of investment strategy optimization on the financial market. *Zbornik Radova Ekonomskog Fakultet au Rijeci*, 27.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383-417.

Fama, E. F., & Blume, M. E. (1966). Filter rules and stock-market trading. *The Journal of Business*, 39(1), 226-241. Retrieved from http://www.jstor.org/stable/2351744.

Fernández-Rodríguez, F., González-Martel, C., & Sosvilla-Rivero, S. (2000). On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. *Economics Letters*, 69(1), 89-94.

Fifield, S., Power, D., & Sinclair, C. D. (2005). An analysis of trading strategies in eleven European stock markets. *The European Journal of Finance*, 11(6), 531-548. Retrieved from https://EconPapers.repec.org/RePEc:taf:eurjfi:v:11:y:2005:i:6:p:531-548.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654-669.

Gehrig, T., & Menkhoff, L. (2006). Extended evidence on the use of technical analysis in foreign exchange. *International Journal of Finance & Economics*, 11(4), 327-338.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5), 602-610.

Heyman, D., Inghelbrecht, K., & Pauwels, S. (2012). Technical trading rules in emerging stock markets. *SSRN Electronic Journal*. doi:10.2139/ssrn.1998036.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

Huang, X., & Guo, F. (2020). A kernel fuzzy twin SVM model for early warning systems of extreme financial risks. *International Journal of Finance & Economics*, n/a(n/a).

Jensen, M. C., & Benington, G. A. (1970). Random walks and technical theories: Some additional evidence. *The Journal of Finance*, 25(2), 469-482.

Jiang, Q., Tang, C., Chen, C., Wang, X., & Huang, Q. (2019). *Stock price forecast based on LSTM neural network.* Paper Presented at the Proceedings of the Twelfth International Conference on Management Science and Engineering Management, Cham.

Jin, Z., Yang, Y., & Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing and Applications*, 32(13), 9713-9729.

Kara, Y., Acar Boyacioglu, M., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange. *Expert Systems with Applications*, 38(5), 5311-5319.

Keerthi Vasan, K., & Surendiran, B. (2016). Dimensionality reduction using principal component analysis for network intrusion detection. *Perspectives in Science*, 8, 510-512.

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, 103, 25-37.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Li, X., & Tang, P. (2020). Stock index prediction based on wavelet transform and FCD-MLGRU. *Journal of Forecasting*, 39(8), 1229-1237.

Liu, S., Zhang, C., & Ma, J. (2017). *CNN-LSTM neural network model for quantitative strategy analysis in stock markets.* Paper Presented at the Neural Information Processing, Cham.

Lu, C. -J. (2010). Integrating independent component analysis-based denoising scheme with neural network for stock price prediction. *Expert Systems with Applications*, 37(10), 7056-7064.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115-133.

McKenzie, C. R., & Takaoka, S. (2007). EViews 5.1. *Journal of Applied Econometrics*, 22(6), 1145-1152.

Metghalchi, M., Marcucci, J., & Chang, Y. -H. (2012). Are moving average trading rules profitable? Evidence from the European stock markets. *Applied Economics*, 44(12), 1539-1559.

Nicholas Refenes, A., Zapranis, A., & Francis, G. (1994). Stock performance modeling using neural networks: A comparative study with regression models. *Neural Networks*, 7(2), 375-388.

Odom, M. D., & Sharda, R. (1990). *A neural network model for bankruptcy prediction.* Paper Presented at the 1990 IJCNN International Joint Conference on Neural Networks.

Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, 115, 635-655.

Sang, C., & Di Pierro, M. (2019). Improving trading technical analysis with TensorFlow long short-term memory (LSTM) neural network. *The Journal of Finance and Data Science*, 5(1), 1-11.

Sayadi, M., & Omidi, M. (2020). Prediction-based portfolio optimization model for Iran's oil dependent stocks using data mining methods. *Iranian Journal of Economic Studies*, 8(2), 225-252.

Seelenfreund, A., George, G. C. P., & Van Horne, J. C. (1968). Stock price behavior and trading. *The Journal of Financial and Quantitative Analysis*, 3(3), 263-281.

Sharif, A., Aloui, C., & Yarovaya, L. (2020). Covid-19 pandemic, oil prices, stock market, geopolitical risk and policy uncertainty nexus in the US economy: Fresh evidence from the wavelet-based approach. *International Review of Financial Analysis*, 70, 101496.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929-1958.

Stanković, J., Marković, I., & Stojanović, M. (2015). Investment strategy optimization using technical analysis and predictive modeling in emerging markets. *Procedia Economics and Finance*, 19, 51-62.

Tay, F. E. H., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, 29(4), 309-317.

Todea, A., Ciupac-Ulici, M., & Silaghi, S. (2009). Adaptive markets hypothesis - evidence from Asia-Pacific financial markets. *The Review of Finance and Banking*, 01(1), 007-013. Retrieved from https://EconPapers.repec.org/RePEc:rfb:journl:v:01:y:2009:i:1:p:007-013.

Todea, A., Zoicas-Ienciu, A., & Filip, A. -M. (2009). Profitability of the moving average strategy and the episodic dependencies: Empirical evidence from European stock. *European Research Studies Journal*, XII, 63-72.

Tyralis, H., & Papacharalampous, G. (2017). Variable selection in time series forecasting using random forests. *Algorithms*, 10, 114.

Vukovic, D., Vyklyuk, Y., Chernova, N., & Maiti, M. (2019). Neural network forecasting in prediction sharpe ratio: Evidence from EU debt market. *Physica A: Statistical Mechanics and its Applications*, 542.

Wang, W., Li, W., Zhang, N., & Liu, K. (2020). Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications*, 143, 113042.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1), 37-52.

Zhao, Z., Rao, R., Tu, S., & Shi, J. (2017). *Time-weighted LSTM model with redefined labeling for stock trend prediction.* Paper Presented at the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI).