applied sciences

MDPI

Article

Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Time Scale Feature Learning

Yaping Hao * and Qiang Gao

School of Electronic and Information Engineering, Beihang University, Beijing 100191, China; gaoqiang@buaa.edu.cn

* Correspondence: haoyaping@buaa.edu.cn; Tel.: +86-138-1096-8583

Received: 25 April 2020; Accepted: 4 June 2020; Published: 7 June 2020

Abstract: In the stock market, predicting the trend of price series is one of the most widely investigated and challenging problems for investors and researchers. There are multiple time scale features in financial time series due to different durations of impact factors and traders' trading behaviors. In this paper, we propose a novel end-to-end hybrid neural network, a model based on multiple time scale feature learning to predict the price trend of the stock market index. Firstly, the hybrid neural network extracts two types of features on different time scales through the first and second layers of the convolutional neural network (CNN), together with the raw daily price series, reflect relatively short-, medium- and long-term features in the price sequence. Secondly, considering time dependencies existing in the three kinds of features, the proposed hybrid neural network leverages three long short-term memory (LSTM) recurrent neural networks to capture such dependencies, respectively. Finally, fully connected layers are used to learn joint representations for predicting the price trend. The proposed hybrid neural network demonstrates its effectiveness by outperforming benchmark models on the real dataset.

Keywords: stock market index trend prediction; multiple time scale features; deep learning; convolutional neural network; long short-term memory neural network

1. Introduction

The trend of the stock market index refers to the upward or downward movements of price series in the future. Accurately predicting the trend of the stock market index can help investors avoid risks and obtain higher returns in the stock exchange [1]. Hence, it has become a hot field and attracted many researchers' attention.

Unitl now, many techniques and various models have been applied to predict the stock market index. Such as traditional statistical models [2,3], machine learning methods [3,4], artificial neural networks (ANNs) [5–8], etc. With the development of deep learning, there are lots of methods based on deep learning used for stock forecasting and have drawn some essential conclusions [9–22].

The above studies were conducted from single time scale features of the stock market index, but it is also meaningful for studying from multiple time scale features. There are multiple time scale features in the stock market index. On the one hand, the stock market is affected by many factors such as economic environment, political policy, industrial development, market news, natural factors and so on. And the durations of factors are different from each other. On the other hand, each investor has a different investment cycle, such as long- and short-term investment. Therefore, we can observe the features of multiple time scales in the stock market index. Among them, the features of a long time scale can reflect the long-term trend of the price, while the features of short time scale can reflect the short-term fluctuation of the price. The combination of multi-scale features facilitates accurate prediction.

In recent years, the convolutional neural network (CNN) has shown high power in feature extraction. Inspired by existing research, we use a CNN to extract multiple time scale features for more comprehensive learning of price sequences. For instance, the daily closing price series in Figure 1a is learned by a two-layer convolutional neural network. The outputs of the two layers of the CNN are called Feature map1 and Feature map2, respectively. As illustrated in Figure 1b, each point of the feature map corresponds to a region of the original price series (termed the receptive field [23]), and it can be considered as the description for the region. Due to different receptive fields, Feature map1 and Feature map2 describe the input price series from two-time scales. Compared with Feature map1, Feature map2 describes the original price sequence on a larger time scale. Therefore, we regard the outputs of different layers of the CNN as features of varying time scales of the original price series.

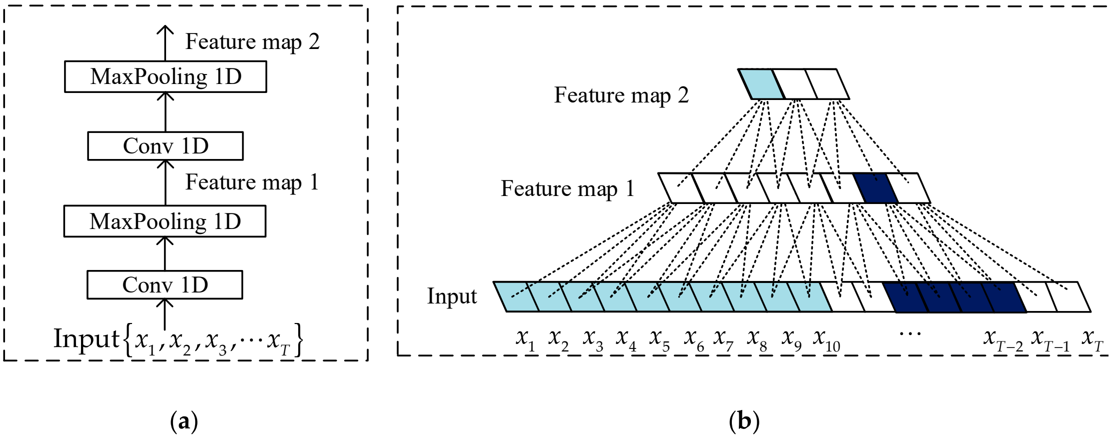

Feature map 2
MaxPooling 1D
Conv 1D
Feature map 1
MaxPooling 1D
Conv 1D
Input$\{x_1, x_2, x_3, \cdots x_T\}$
Feature map 2
Feature map 1
Input
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} $\cdots$ x_{T-2} x_{T-1} x_T
(a)
(b)

Figure 1. An example showing that CNN can extract features of different time scales. (**a**) The daily closing price series learned by a convolutional neural network; (**b**) Description of the time scale of the features.

Meanwhile, the Long Short-Term Memory (LSTM) network works well on sequence data with long-term dependencies due to the internal memory mechanism [24,25]. Many studies use LSTM networks to learn the long-term relationship of features extracted by the CNN. In this way, we utilize LSTMs to learn long-term dependencies of multiple time scale feature sequences obtained by the CNN.

In this paper, we present a novel end-to-end hybrid neural network to learn multiple time scale features for predicting the trend of the stock market index. The network first combines the features obtained from different layers of the CNN with daily price subsequences to form multiple time scale features, reflecting the short-, medium-, and long-term laws of price series. Subsequently, three LSTM recurrent neural networks are utilized to capture time dependencies in multiple time scale features obtained in the previous step. Then, several fully connected layers combine features learned by LSTMs to predict the trend of the stock market index. The experimental analysis of real datasets demonstrates that the proposed hybrid network outperforms a variety of baselines in terms of trend prediction in accuracy.

The rest of the paper is organized as follows. Section 2 presents related work. In Section 3, we present the proposed hybrid neural network based on multiple time scale features of the stock market index. Section 4 reports experiments and results, and the paper is discussed in Section 5.

2. Related Work

From traditional methods to deep learning models, there are numerous techniques on forecasting financial time series, among which deep learning has received widespread attention due to its outperformance.

LSTM is the most preferred deep learning model in studies of predicting financial time series. LSTM can extract dependencies in time series by the internal memory mechanism. In [11], LSTM networks were used for predicting out-of-sample directional movements for the constituent stocks of the S&P 500. They found LSTM networks outperform memory-free classification methods. Si et al. [12] constructed a trading model for the Chinese futures market through DRL and LSTM. They used deep neural networks to discover market features. Then an LSTM was applied to make continuous trading decisions. Bao et al. [13] use Wavelet Transforms and Stacked Autoencoders to learn useful information in technical indicators and use LSTMs to learn time dependencies for the forecasting of stock prices. In [14], limit order book and history information was input to the LSTM model for the determination of the stock price movements. Tsantekidis et al. [15] utilized limit order book and the LSTM model for the trend prediction. These works prove that LSTMs can successfully extract time dependencies in the financial sequence. However, these works do not consider the multiple time scale features in the price series.

Several studies focused on utilizing CNN models inspired by their remarkable achievements in other fields, such as image recognition [26], speech processing [27], natural language processing [28], etc. Convolutional neural networks can directly extract features of the input without sophisticated preprocessing, and can efficiently process various complex data. Chen et al. [16] used one-dimensional CNN with an agent-based RL algorithm to study Taiwan stock index futures. In [17], Siripurapu et al. convert price sequences into pictures and then use the CNN to learn useful features for prediction. In [18], a new CNN model was proposed to predict the trend of the stock prices. Correlations between instances and features are utilized to order the features before they are presented as inputs to the CNN. These works use CNNs to extract features of a single time scale in the price series. But in financial time series, multiple time scale features are ubiquitous, and it is meaningful to study them.

Besides, there are some studies combine the advantages of CNN and LSTM to form hybrid networks. In [19], the proposed model makes the stock selection strategy by using the CNN and then makes the timing strategy by using the LSTM. Wang et al. [20] proposed a Deep Co-investment Network Learning (DeepCNL) model, which combined convolutional and recurrent neural layers. Both [19,20] take advantage of the combination of CNN and LSTM. However, they ignore the multiple time scale features that exist in the financial time series. In [21], numerous pipelines combining CNN and bi-directional LSTM for improved stock market index prediction. In [22], both convolutional and recurrent neurons are integrated to build the multi-filter structure, so that the information from different feature spaces and market views can be obtained. Although the authors in [21,22] proposed models based on multi-scale features, they used multiple pipelines or networks to extract multi-scale features, which makes the model complex and vast, which is not conducive to training or obtaining useful information.

Differing from previous work, we propose a hybrid neural network that mainly focuses on multiple time scale features in financial time series for trend prediction. We innovatively use a CNN to extract features on multiple time scales, simplifying the model and facilitating better predictions. Then we use several LSTMs to learn time dependencies in feature sequences extracted by the CNN, and fully connected layers for higher-level feature abstraction.

3. Hybrid Neural Network Based on Multiple Time Scale Feature Learning

In this section, we provide the formal definition of the trend learning and forecasting problem. Then, we present the proposed hybrid neural network based on multiple time scale feature learning.

3.1. Problem Formulation

In the stock market, there are 5 trading days in a week and 20 trading days in a month. Investors are usually interested in price movements after a week or a month. Therefore we use a series of closing prices for 40 consecutive days to predict the trend of the closing price in n trading days, where the values of n are 5 (a week) and 20 (a month). Formally, we define the sequence of historical closing prices as $X_i = \{x_{i+1}, x_{i+2}, \cdots x_{i+t}, \cdots x_{i+40}\}$, where x_{i+t} is the value of the closing price on the $(i + t)$-th day. Meanwhile, the upward or downward trend to be predicted is defined by the following rule:

$$Y_i = \begin{cases} 1 & x_{i+40} \leq x_{i+40+n} \\ 0 & x_{i+40} > x_{i+40+n} \end{cases} \tag{1}$$

where Y_i denotes the trend of the closing price a week ($n = 5$) or a month ($n = 20$) later, 0 represents the downward trend, and 1 represents the upward trend, x_{i+40} is the closing price value of the $(i + 40)$-th day and x_{i+40+n} is the closing price value of the (i + 40 + n)-th day.

Then, we aim to propose a hybrid neural network to learn the function $f(X)$ to predict the price trend one week or one month later.

3.2. *Hybrid Neural Network Based on Multiple Time Scale Feature Learning*

In this part, we present an overview of the proposed hybrid neural network based on multiple time scale feature learning for the trend forecasting. Then, we detail each component of the hybrid neural network.

3.2.1. Overview

The idea of the proposed hybrid neural network is divided into three parts. The first part is to extract the characteristics of different time scales of price series through different layers of a CNN, and combine them with the original daily price series to reflect the relatively short-, medium- and long-term changes in the price sequence, respectively. The second part is to use multiple LSTMs to learn time dependencies of features of different time scales. The last part is to combine all the information learned by LSTMs through a fully connected neural network to forecast the trend of the closing price in the future. Though the hybrid neural network is composed of different kinds of network architectures, it can be jointly trained with one loss function. Figure 2 shows the structure of the hybrid neural network, which can be viewed as a combination of three models based on single time scale feature learning. The three models are shown in Figure 3. Next, we will introduce each part of the proposed model in detail.

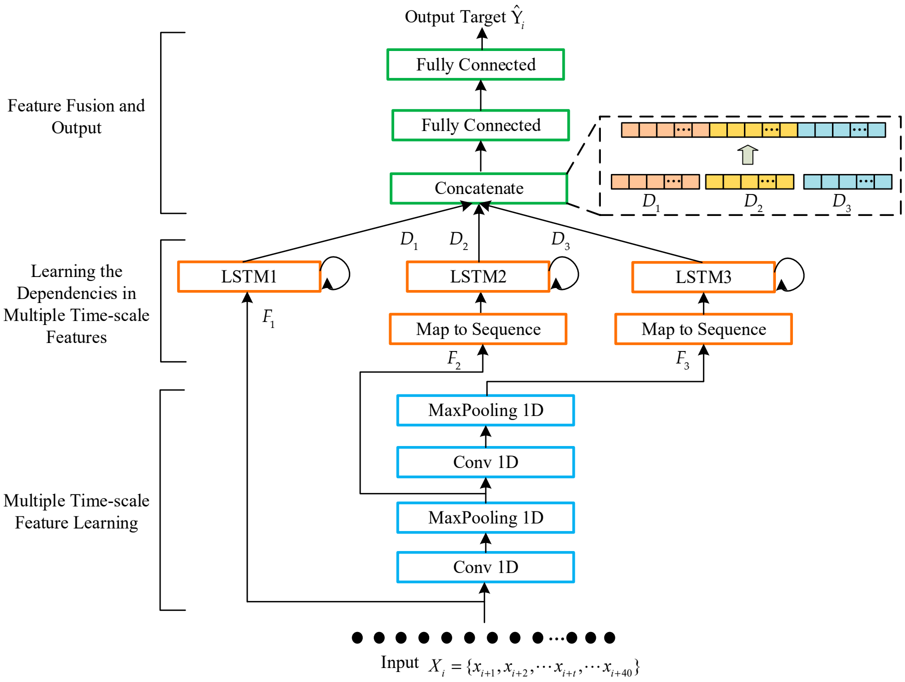

Output Target $\hat{Y}_i$
Fully Connected
Fully Connected
Concatenate
Feature Fusion and Output
D_1
D_2
D_3
Learning the Dependencies in Multiple Time-scale Features
LSTM1
LSTM2
LSTM3
F_1
Map to Sequence
Map to Sequence
F_2
F_3
MaxPooling 1D
Conv 1D
MaxPooling 1D
Conv 1D
Multiple Time-scale Feature Learning
Input $X_i = \{x_{i+1}, x_{i+2}, \cdots x_{i+t}, \cdots x_{i+40}\}$

Figure 2. The proposed hybrid neural network based on multiple time scale feature learning.

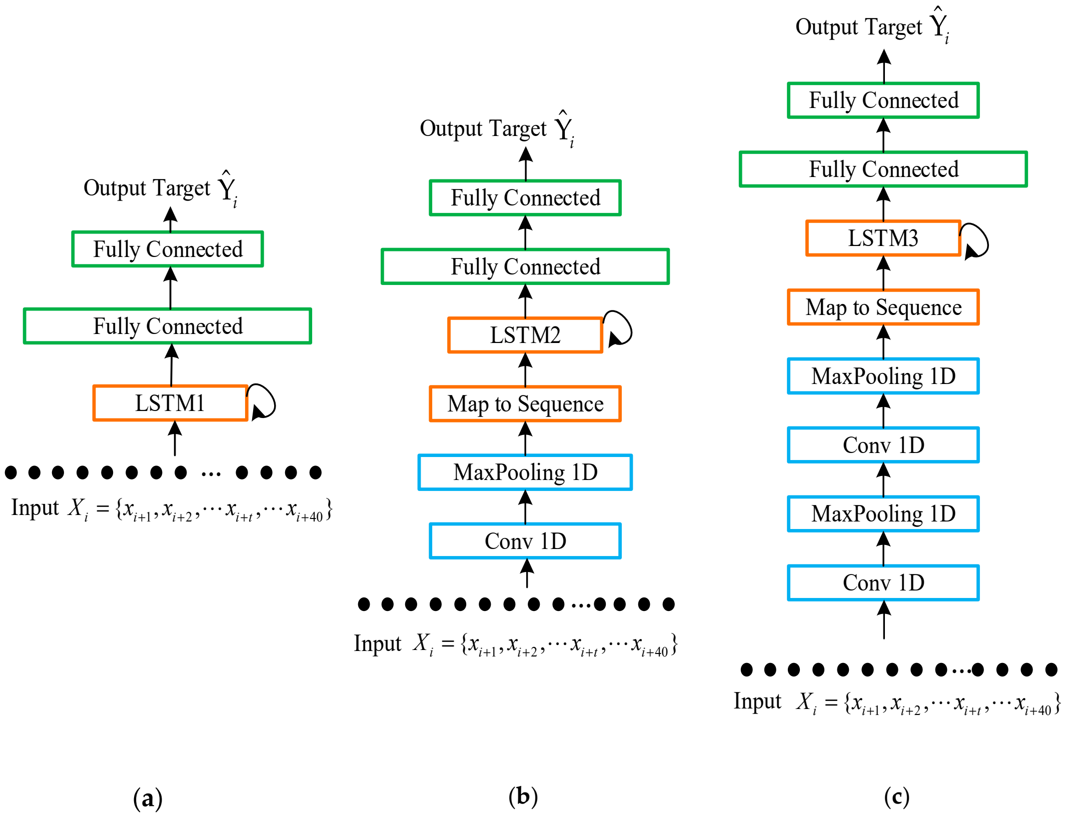

Output Target $\hat{Y}_i$
Fully Connected
Fully Connected
LSTM1
Input $X_i = \{x_{i+1}, x_{i+2}, \cdots x_{i+t}, \cdots x_{i+40}\}$
Output Target $\hat{Y}_i$
Fully Connected
Fully Connected
LSTM2
Map to Sequence
MaxPooling 1D
Conv 1D
Input $X_i = \{x_{i+1}, x_{i+2}, \cdots x_{i+t}, \cdots x_{i+40}\}$
Output Target $\hat{Y}_i$
Fully Connected
Fully Connected
LSTM3
Map to Sequence
MaxPooling 1D
Conv 1D
MaxPooling 1D
Conv 1D
Input $X_i = \{x_{i+1}, x_{i+2}, \cdots x_{i+t}, \cdots x_{i+40}\}$
(**a**)
(**b**)
(**c**)

Figure 3. The models based on single time scale feature learning. (**a**) The structure of Model based on F_1; (**b**) The structure of Model based on F_2; (**c**) The structure of Model based on F_3.

3.2.2. Multiple Time Scale Feature Learning

Considering that there are multiple time scale features in the stock market index sequence and the combination of these features can help to predict the price trend more accurately. In this paper, we research the internal laws of price movement from three time scales. On the one hand, the daily price subsequence represented by F_1 can be regarded as the feature sequence of the minimum time scale. It can reflect local price changes, which are vital to the prediction. On the other hand, the output of different layers of the CNN describes the original price series from different time scales. These outputs can be regarded as the characteristics of different time scales. Since the CNN has two layers, we can get the features on two different time scales, which are represented as F_2 and F_3. In this way, we obtain three kinds of features, F_1, F_2 and F_3, which can reflect relatively short-, medium- and long-term trend changes, respectively.

3.2.3. Learning the Dependencies in Multiple Time Scale Features

We use three LSTMs to learn time dependencies in features of different time scales. We need to convert feature maps extracted by the CNN into feature sequences suitable for LSTMs by map to sequence layer. As shown in Figure 4, feature maps represent features learned by the CNN. Different colors indicate that these feature maps are obtained by different convolution kernels. The points in the feature map are arranged chronologically from left to right. The feature sequence represents the input of LSTM. The feature vector in the feature sequence is represented by fv_t, and the subscript t corresponds to its order in the series. Each feature vector is generated from left to right on feature maps by column. This means the i-th feature vector is the concatenation of the i-th columns of all the maps.

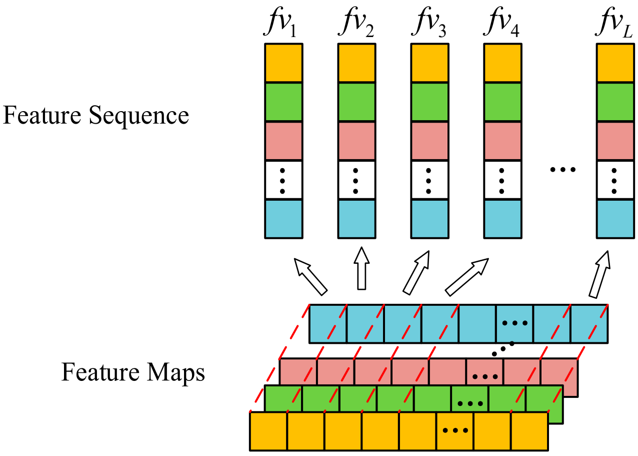


Figure 4. The explanation of map to sequence layer.

Each LSTM network learns the time dependencies in its corresponding feature sequence and the process is described as follows. In the LSTM, each cell has three main gates: the input gate, the forget gate and the output gate. Suppose that the input feature vector at the time t is fv_t and the hidden state at the previous time step is h_{t-1}.

The input gate i_t is calculated by:

$$i_t = sigmoid(W_{ifv}fv_t + W_{ih}h_{t-1} + b_i) \tag{2}$$

The forget gate f_t is calculated by:

$$f_t = sigmoid(W_{ffv}fv_t + W_{fh}h_{t-1} + b_f) \tag{3}$$

The output gate o_t is calculated by:

$$o_t = sigmoid(W_{ofv}fv_t + W_{oh}h_{t-1} + b_o) \tag{4}$$

The principle of the memory mechanism is to control the addition of new information through the input gate, and to forget of the former information through the forget gate. The old information is represented by c_{t-1}, and the latest information is calculated as follows:

$$\widetilde{c_t} = \tanh(W_{cfv}fv_t + W_{ch}h_{t-1} + b_c) \tag{5}$$

The information stored in the memory unit is updated as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \tag{6}$$

Then the output of the LSTM cell is expressed as:

$$h_t = o_t \odot \tanh(c_t) \tag{7}$$

where $\bigodot$ means element-wise product.

After all the feature vectors complete the above process, we will use the final output h_t as the time dependencies learned by the LSTM cell. The time dependences learned by the three LSTMs are denoted by D_1, D_2 and D_3, which are all one-dimensional vectors.

3.2.4. Feature Fusion and Output

The concatenate layer is used to combine the output representations from three LSTM recurrent neural networks. As shown in Figure 2, D_1, D_2 and D_3 are concatenated to form a joint feature. Then, such a joint feature is fed to the fully connected layers to provide the trend prediction. Mathematically, the prediction of the hybrid neural network is expressed as:

$$\hat{Y}_{\mathrm{i}} = f(D_1, D_2, D_3) = \varphi(W(W_1D_1 + W_2D_2 + W_3D_3) + b) \tag{8}$$

where φ is the sigmoid activation function. W_1, W_2 and W_3 are weights for the first fully connected layer. W and b are the weights and bias of the second fully connected layer.

4. Experiments and Results

In this section, we report experiments and detailed results to demonstrate the process of obtaining multiple time scale features and the advantage of the proposed model by comparing it to a variety of baselines.

4.1. Experiment Setup

4.1.1. Experimental Data

The S&P 500 index (formerly Standard & Poor's 500 Index) is a market capitalization-weighted index of the 500 largest U.S. publicly traded companies by market value, and it is widely used in scientific research. In this paper, we study the daily closing price dataset of the S&P 500 index from 30 January 1999 to 30 January 2019 for a total of 20 years obtained from the Yahoo Finance Website [29].

Data normalization is required to transform raw time series data into an acceptable form for applying a machine learning technique. The normalization makes raw closing price series in the interval [0, 1] according to the following formula:

$$X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{9}$$

where X is the original closing price before normalization, X_{min} and X_{max} are the minimum value and the maximum value before X is normalized, respectively. X' is the data after normalization.

After the normalization, data instances are built by combining historical closing prices and the target trend for each time series subsequence. We then take the samples from 30 January 1999 to 30 January 2015 as the training set, and the samples from 1 February 2015 to 30 January 2017 as the validation set, and the remaining samples were used for testing.

4.1.2. Baselines

1. Models based on single time scale features

- Model based on F_1: As shown in Figure 3a, the model directly treats the daily price series as a relatively short-term feature sequence that is subsequently learned by an LSTM and fully connected layers.
- Model based on F_2: As shown in Figure 3b, the model uses the convolutional neural network with one layer to extract the relatively medium-term features, and then predicts the price trend through an LSTM and fully connected layers.
- Model based on F_3: As shown in Figure 3c, the relatively long-term features are extracted by a CNN with two layers. Then it uses an LSTM and fully connected layers to forecast the trend of the closing price.

2. Existing models

- Simplistic Model: This is a simplistic model that directly takes the trend of the last n days of historical price series as the future trend.
- SVM: SVM is a machine learning method commonly used in financial forecasting, such as [3,4]. We adjust parameters c (error penalty), k (kernel function), and γ (kernel coefficient) to make the model reach the best state. In order to make the model perform better, parameters are selected from the sets $c \in \left\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 10^2, 10^3, 10^4, 10^5\right\}$, $k \in \{rbf, sigmoid\}$, $\gamma \in \left\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 10^2, 10^3, 10^4, 10^5\right\}$, respectively.
- LSTM: Because of the time dependencies in financial time series, LSTM is often used in financial forecasting, such as [11,12,15]. We mainly adjusted the parameters L (number of network layers), and N (number of hidden units). We select appropriate parameters in the sets $L \in \{1, 2, 3\}$ and $N \in \{10, 20, 30\}$.
- CNN: Similar to LSTM, CNN is also a common model in this field, such as [16–18]. We mainly adjusted parameters L (number of network layers), S (convolution kernel size) and N (number of convolution kernels). Here we select appropriate parameters in the sets $L \in \{1, 2, 3\}$, $S \in \{3, 5, 7\}$ and $N \in \{10, 20, 30, 40\}$.
- Multiple Pipeline Model: In [21], the authors proposed a new deep learning model that combines multiple pipelines. Each pipeline contains a CNN for feature extraction, a Bi-directional LSTM for temporal data analysis, and a Dense layer to give the output for each individual pipeline. Then they use a Dense layer to combine the different outputs to make predictions.
- MFNN: In [22], the authors proposed a novel end-to-end model named multi-filters neural network (MFNN) specifically for prediction on financial time series. Both convolutional and recurrent neurons are integrated to build the multi-filters structure, so that the information from different feature spaces and market views can be obtained.

4.1.3. Evaluation Metric

Generally, stock index trend prediction can be considered as a classification problem. In order to evaluate the quality of predictions, we use Accuracy as the evaluation metric. Accuracy represents the proportion of samples that be correctly predicted to the total number of samples. The higher the Accuracy, the better the predictive performance of the model. Accuracy is calculated as follows:

$$Accuracy = \frac{N_{correct}}{N_{all}} \times 100\% \tag{10}$$

where $N_{correct}$ represents the samples with the same predicted trend as the actual trend, and N_{all} represents the total number of samples.

4.1.4. Training

The proposed hybrid neural network includes a CNN to extract multiple time scale features, three LSTMs to learn time dependencies, and fully connected layers for higher-level feature abstraction. The CNN has two layers containing 1-d convolution, activation, and pooling operations. Convolution operations of two layers have 10 and 20 filters, respectively. Considering that the data we studied is a one-dimensional price series, 1-d convolution is sufficient here. The activation function is LeakyReLU. The pooling operation is max pooling with size 2 and stride 2. The number of LSTM units is 10, and the number of units in the subsequent fully connected layer is 10 and 1.

Besides, the loss function is binary cross-entropy. An Adam optimizer is used to train the neural network. The learning rate is initialized to 0.001, and decays [30] with the iteration according to Equation (11).

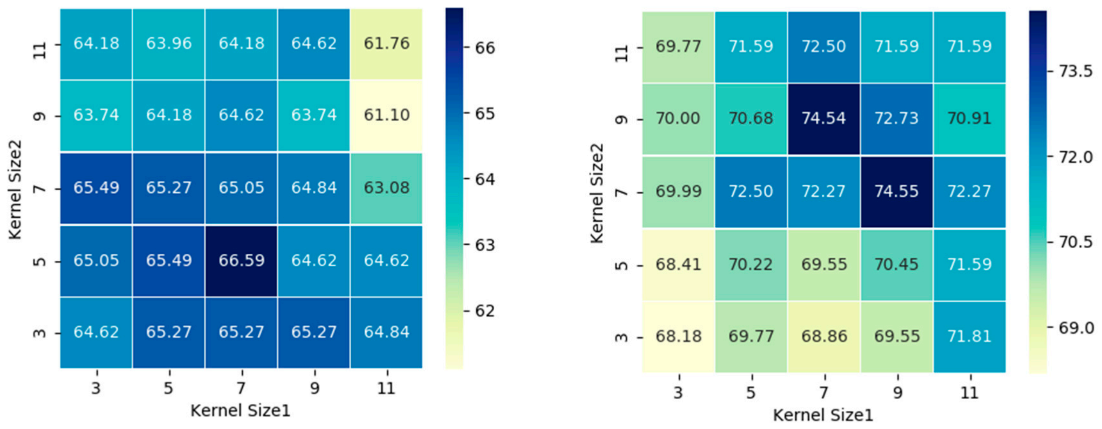
$$lr = lr * \frac{1}{1 + I * d} \tag{11}$$

where lr represents the learning rate, I describes the current iteration, and d represents the attenuation coefficient which takes a value of 10^{-6}. The value of the batch size is 80. We use early stopping to prevent the network from overfitting. That is, training will stop if the accuracy on the validation set has not been improved for 70 epochs. In addition, we use dropout [31] to control the capacity of neural networks to prevent overfitting, and use batchnormalization [32] after convolution operation to reduce internal covariate shift for the better performance of the network.

4.2. Experiment Results

4.2.1. Determination of Time Scales of Features

Considering that we predict the price trend in combination with multiple time scale features (F_1, F_2 and F_3), we need to determine appropriate time scales for these features in order to predict more accurately. Specifically, since we directly treat the daily data as the feature sequence (F_1), we only need to determine time scales for features learned by CNN (F_2 and F_3). Time scales of F_2 and F_3 correspond to the receptive fields, which are determined by kernel size and stride of max-pooling operation and convolution operation. Since we fix other parameters as common values, we only need to adjust kernel sizes of two convolution layers to find more appropriate time scales. The results are shown in Figure 5. Figure 5a,b represent experimental results when predicting price trends one week and one month later, respectively. The abscissa represents the convolution kernel size of the first layer of CNN, while the ordinate represents the convolution kernel size of the second layer of CNN. The larger the convolution kernel size, the larger the time scale of features. The numerical values in the figure represent Accuracy of the model corresponding to the point.



(**a**) (**b**)

Figure 5. The determination of time scales of features. (**a**) Results for predicting the price trend in one week; (**b**) Results for predicting the price trend in one month.

In Figure 5, we observe that when predicting the price trend one week later, the convolution kernel sizes of two convolutional layers are preferably 7 and 5, respectively. Therefore, the time scales of F_2 and F_3 are 8 trading days and 18 trading days, that is, each point in F_2 is obtained by the closing price sequence of 8 trading days, and each point in F_3 is obtained by the closing price sequence of 18 trading days. Similarly, when predicting the price trend one month later, the sizes of two convolution kernels are preferably 9 and 7. In this case, the time scales of F_2 and F_3 are 10 trading days and 24 trading days.

In addition, we can find that time scales of the features used to predict the price trend in a month is larger than that used to predict the price trend in a week. This is consistent with our experience, that is, larger-grained data are used for long-term forecasting, and smaller-grained data are used for short-term forecasting.

4.2.2. Comparisons with Models Based on Single Time Scale Features

In this part, we investigate the advantages of the proposed hybrid neural network in combining multiple time scale features. We compared the proposed network with models based on single time scale features in accuracy. The results are reported in Table 1. The models based on F_1, F_2, and F_3 represent the three models in Figure 3. They are all based on single time scale features.

Table 1. Comparisons with models based on single time scale features in accuracy.

Forecast Horizon	Model	Accuracy (%)
One week	Model based on F_1	64.40
	Model based on F_2	63.30
	Model based on F_3	61.76
	The proposed model	**66.59**
One month	Model based on F_1	71.14
	Model based on F_2	70.23
	Model based on F_3	73.64
	The proposed model	**74.55**

Table 1 shows that combining the features of multiple time scales can promote accurate prediction. The models based on F_1, F_2, and F_3 are based on the features of a single time scale, while the proposed hybrid neural network combines the features of multiple time scales to predict the trend. The predictive performance of the hybrid neural network is better than other networks. Therefore, combining the features of multiple time scales is helpful.

In the next group of experiments, taking forecasting price trends one week later as an example, we visualize the trend prediction using test samples, as shown in Figures 6–8. From these graphs, we can intuitively understand the benefits of combining features of multiple time scales.

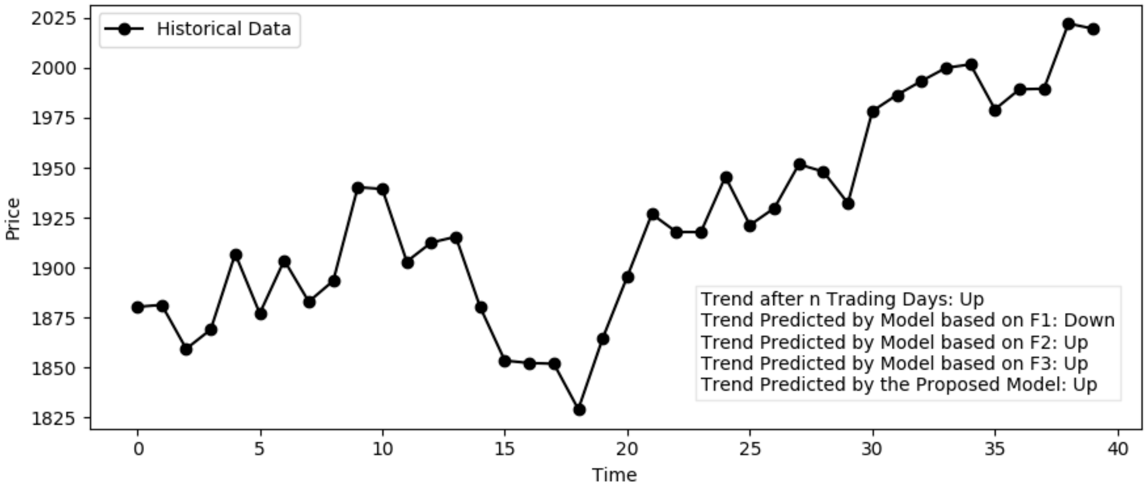


Figure 6. Visualization of the trend prediction by different models on test example 1.

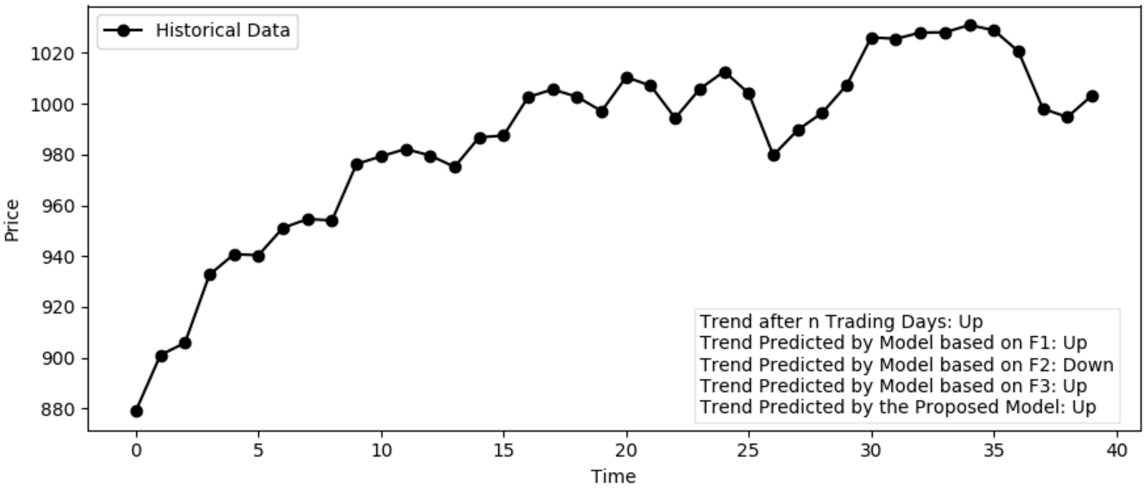


Figure 7. Visualization of the trend prediction by different models on test example 2.

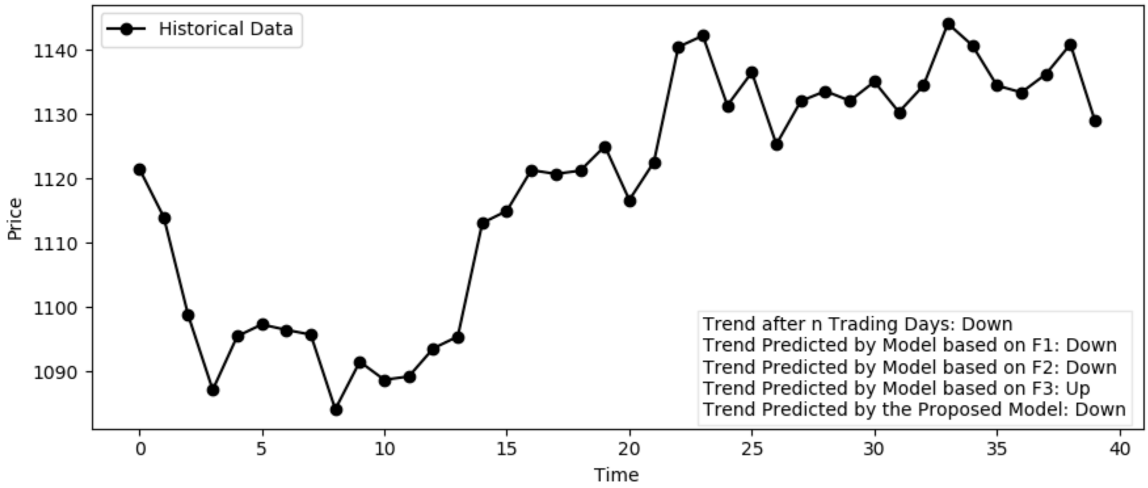


Figure 8. Visualization of the trend prediction by different models on test example 3.

In Figure 6, we can find that due to the influence of short-term fluctuations, in some cases, the model based on F_1 fails to predict the price trend in a week accurately. Similarly, in Figure 7, Model based on F_2 extracts medium-term features to predict the trend, but ultimately fails due to the neglect of long-term and short-term changes in price series. In Figure 8, the model based on F_3 only pays attention to the long-term characteristics in the price series and cannot accurately predict the trend. Therefore, we can conclude that predicting the price trend based on features of a single time scale is not feasible in some cases due to the complexity and variability of price series. It makes sense to combine the features of multiple time scales to predict the future price trend.

4.2.3. Comparisons with Existing Models

Due to the differences in data preprocessing, model training methods, and learning targets, it is not easy to directly compare among existing works, we try to select some models commonly used in financial forecasting and adjust these models to the best state to make relatively fair comparisons. The experimental results are shown in Table 2.

Table 2. Comparisons with existing models in accuracy.

Forecast Horizon	Model	Accuracy (%)
One week	Simplistic Model	54.82
	SVM	61.98
	LSTM	65.05
	CNN	59.34
	Multiple Pipeline Model	63.30
	NFNN	65.93
	The proposed model	**66.59**
One month	Simplistic Model	56.92
	SVM	70.91
	LSTM	71.59
	CNN	67.95
	Multiple Pipeline Model	72.05
	MFNN	72.27
	The proposed model	**74.55**

From Table 2, we can find that the proposed hybrid neural network performs better than other models, whether the forecast horizon is one week or one month. On the one hand, SVM is a commonly used machine learning model in financial forecasting, while CNN and LSTM are commonly used deep learning models. Compared with the Simplistic Model, these models can extract profitable information, but they only learn features from a single scale and ignore some useful information. On the other hand, the Multiple Pipeline Model and MFNN are models based on multi-scale feature learning for financial time series forecasting. They use different branches or different networks to extract different scale features. However, it will increase the complexity of the model. The model we proposed only utilizes a CNN to extract features of different scales simplifying the model and predicting more accurately. Therefore, we can conclude that the proposed hybrid neural network is superior to the commonly used models in the existing works.

5. Discussion

In this paper, we propose a hybrid neural network based on multiple time scale feature learning for stock market index trend prediction. Because there are multi-scale features in financial time series, it makes sense to combine them to predict future trends. First, the proposed model only utilizes one CNN to extract multiple time scale features, instead of using multiple networks like other models. It simplifies the model and makes more accurate predictions. Second, time dependences in the multiple time scale features are learned by three LSTMs. Finally, the information learned by LSTMs is fused through fully connected layers to predict the price trend.

The experimental results demonstrate that such a hybrid network can indeed enhance the predictive performance compared with benchmark networks. Firstly, by comparing with the models based on F_1, F_2 and F_3, we conclude that combining multiple time scale features can promote accurate prediction. Secondly, in comparison with the Simplistic Model, we found that the proposed model can learn valuable information. SVM, CNN, and LSTM all learn the features in price series from a single scale. However, there are multiple scales in financial time series, which makes these methods sometimes unable to perform accurate predictions. Finally, both the Multiple Pipeline Model and MFNN are models based on multi-scale feature learning, but they use several branches or networks to extract multi-scale features, making the network huge and complex. The hybrid neural network we proposed uses only a CNN to extract multi-scale features, simplifying the model and predicting more accurately.

However, we can find that the proposed model cannot accurately predict trends for some data samples. There may be two reasons for this issue. First, the three time scales features are not enough to reflect the internal law of price series. Second, these samples are seriously affected by other factors that we have not considered, such as political policy, industrial development, natural factors, and so on. It provides directions for our future work. We can consider using a multi-layer CNN to extract more scale features for prediction. At the same time, we can extract useful information from more sources, such as macroeconomic indicators, news, market sentiment and so on.

Author Contributions: Y.H. proposed the basic framework of hybrid neural network and completed the model construction, experimental research, and thesis writing. Throughout the process, Q.G. guided and gave a lot of suggestions. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qiu, M.; Song, Y. Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network Model. *PLoS ONE* **2016**, *11*, e0155133. [CrossRef] [PubMed]
2. Adebiyi, A.; Adewumi, A.; Ayo, C.K. Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. *J. Appl. Math.* **2014**, *2014*, 1–7. [CrossRef]
3. Pai, P.-F.; Lin, C.-S. A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega* **2005**, *33*, 497–505. [CrossRef]
4. Kara, Y.; Acar, M.; Baykan, Ö.K. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Syst. Appl.* **2011**, *38*, 5311–5319. [CrossRef]
5. Di Persio, L.; Honchar, O. Artificial neural networks architectures for stock price prediction: Comparisons and applications. *Int. J. Circuits Syst. Signal Process.* **2016**, *10*, 403–413.
6. Masoud, N. Predicting Direction of Stock Prices Index Movement Using Artificial Neural Networks: The Case of Libyan Financial Market. *Br. J. Econ. Manag. Trade* **2014**, *4*, 597–619. [CrossRef]
7. Inthachot, M.; Boonjing, V.; Intakosum, S. Artificial Neural Network and Genetic Algorithm Hybrid Intelligence for Predicting Thai Stock Price Index Trend. *Comput. Intell. Neurosci.* **2016**, *2016*, 1–8. [CrossRef] [PubMed]
8. Qiu, M.; Song, Y.; Akagi, F. Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market. *Chaos Solitons Fractals* **2016**, *85*, 1–7. [CrossRef]
9. Chong, E.; Han, C.; Park, F.C. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Syst. Appl.* **2017**, *83*, 187–205. [CrossRef]
10. Singh, R.; Srivastava, S. Stock prediction using deep learning. *Multimedia Tools Appl.* **2016**, *76*, 18569–18584. [CrossRef]
11. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [CrossRef]
12. Si, W.; Li, J.; Ding, P.; Rao, R. A Multi-objective Deep Reinforcement Learning Approach for Stock Index Future's Intraday Trading. In Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017; IEEE: New York, NY, USA, 2017; pp. 431–436.
13. Bao, W.; Yue, J.; Rao, Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE* **2017**, *12*, e0180944. [CrossRef] [PubMed]
14. Sirignano, J.; Cont, R. Universal features of price formation in financial markets: perspectives from deep learning. *Quant. Financ.* **2019**, *19*, 1449–1459. [CrossRef]
15. Tsantekidis, A.; Passalis, N.; Tefas, A.; Kanniainen, J.; Gabbouj, M.; Iosifidis, A. Using deep learning to detect price change indications in financial markets. In Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos, Greek, 28 August–2 September 2017; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2017; pp. 2511–2515.
16. Chen, C.-T.; Chen, A.-P.; Huang, S.-H. Cloning Strategies from Trading Records using Agent-based Reinforcement Learning Algorithm. In Proceedings of the 2018 IEEE International Conference on Agents (ICA), Singapore, 28–31 July 2018; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2018; pp. 34–37.

17. Siripurapu, A. *Convolutional Networks for Stock Trading*; Stanford University Department of Computer Science: Stanford, CA, USA, 2014.
18. Gunduz, H.; Yaslan, Y.; Cataltepe, Z. Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowl.-Based Syst.* **2017**, *137*, 138–148. [CrossRef]
19. Liu, S.; Zhang, C.; Ma, J. CNN-LSTM Neural Network Model for Quantitative Strategy Analysis in Stock Markets. In Proceedings of the International Conference on Neural Information Processing, Guangzhou, China, 14–18 November 2017; Springer: Cham, Switzerland, 2017; pp. 198–206.
20. Wang, Y.; Zhang, C.; Wang, S.; Yu, P.S.; Bai, L.; Cui, L. Deep Co-Investment Network Learning for Financial Assets. In Proceedings of the 2018 IEEE International Conference on Big Knowledge (ICBK), Singapore, 17–18 November 2018; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2018; pp. 41–48.
21. Eapen, J.; Bein, D.; Verma, A. Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2019; pp. 0264–0270.
22. Long, W.; Lu, Z.; Cui, L. Deep learning-based feature engineering for stock price movement prediction. *Knowl.-Based Syst.* **2019**, *164*, 163–173. [CrossRef]
23. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems, Barcelona, SPAIN, 5–10 December 2016*; MIT Press: Cambridge, MA, USA, 2016; pp. 4898–4906.
24. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
25. Chung, J.; Gulcehre, C.; Cho, K.H.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
27. Jaitly, N.; Zhang, Y.; Chan, W. Very Deep Convolutional Neural Networks for End-To-End Speech Recognition. U.S. Patent No 10,510,004, 1 August 2019.
28. I Widiastuti, N. Convolution Neural Network for Text Mining and Natural Language Processing. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Bandung, Indonesia, 18 July 2019; IOP Publishing: Bristol, UK, 2019; Volume 662, p. 052010.
29. Yahoo Finance, S & P 500 Stock Data. Available online: https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC (accessed on 1 February 2019).
30. Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M.W.; Pfau, D.; Schaul, T.; Shillingford, B.; De Freitas, N. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2016; pp. 3981–3989.
31. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
32. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).