

Stock Prediction Using Convolutional Neural Network

Sheng Chen^{1,*} and Hongxiang He¹

¹School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, People's Republic of China

*Email: chnshn@hotmail.com

Abstract. Stock prediction is a very hot topic in our life. However, in the early time, because of some reasons and the limitation of the device, only a few people had the access to the study. Thanks to the rapid development of science and technology, in recent years more and more people are devoted to the study of the prediction and it becomes easier and easier for us to make stock prediction by using different ways now, including machine learning, deep learning and so on. In this paper, we proposed a deep learning method based on Convolutional Neural Network to predict the stock price movement of Chinese stock market. We set the opening price, high price, low price, closing price and volume of stock deriving from the internet as input of the architecture and then run and test the program. The result has shown that it is a bit reliable to use deep learning method based on Convolutional Neural Network to predict the stockprice movement of China.

1. Introduction

It is recorded that there are about 200 billion stock investors in China. Although the number is very large, the real number of the people who make profit from the stock market is very small. If you want to get the win from the stock market, the analysis of the data is necessary. For example, you can study the movement tendency of the stock price in last few weeks and then determine whether it will be up or down in next few days according to some relations between each day's stock data. Of course, you also can use some mathematical models to help you do the analysis. In fact, stock prediction is really a very nice idea in one way, but because of its complexity and dynamic features, the result of the prediction is not satisfied. Some scientists still doubt whether it is possible for us to make the prediction.

In these years, the main method to predict the stock price movement can be concluded in 2 aspects: using machine learning ways or deep learning ways.

Machine learning is a very useful tool to do some work in financial areas, which can be divided into two parts: supervised machine learning and unsupervised machine learning. The significant difference between the former and the latter is whether we have the label of the training data. The core ideology of machine learning is that we should find a supervised mathematical model to fit the data, and then you should train the model to lower the error or improve the fitting degree. Traditional machine learning way or concept that people use to help them with the job in daily life includes support vector machines, decision trees, random forest and so on. For example, we can use regression analysis to predict the price of the house. We also can use logistic regression to detect credit card fraud. Of course, there is no doubt that it can be used to predict the stock price movement, too.

Although machine learning has a long history and very useful, in recent few years, deep learning enjoys popularity among people all over the world. [1] has shown that the Google team used a special



machine learning way --deep learning to win the contest of image classification. Besides, when the famous robots named Alphago2.0 who was composed of several deep learning models beat the world Weiqi Champion KeJie with 3:0, it attracts more people to make a study on deep learning.

One we should pay attention to is that deep learning is developed on the basis of machine learning, which shows that both concepts are very similar. Traditional deep learning way also includes two parts: Convolutional Neural Network and Recurrent Neural Network.

Traditionally, the input of a Convolutional Neural Network is often a 2D image, but that does not mean that we cannot use the model to help us make predictions. There are two ways to preprocess the data, one we can convert the 1D-input data into a 2D matrix and the other way is to take advantage of 1D function to help us do the computation of convolution. We chose the latter to help us run and test the experiment.

Compared with Convolutional Neural Network, Recurrent Neural Network may be more useful to deal the prediction problem in financial, for the input of the neural network is time series data. But when people use RNN model to help them, they find that the gradient may be vanished. So, they have improved model, naming it as LSTM (Long Short-Term Memory), which are a useful model in Natural Language Process task.

In this paper, a deep learning model based on Convolutional Neural Network is proposed to predict the stock price movement of Chinese stock market. We use TensorFlow to help us design the model. At first, we will preprocess data and normalize them because each feature has a big difference, otherwise it will affect the result. Then, because the stock data belongs to 1D time series data, we have used a 1D function to do the convolution and set five features including volume, high price, low price, closing price and volume as input.

The rest of this paper is organized as follows: Section 2 will have a literature review. Section 3 will introduce how to preprocess the data from the internet and detail in our model. Section 4 will describe the result of our experiment. Section 5 will conclude the paper and propose some suggestion for further study.

2. Literature review

2.1 Literature review in machine learning method

Before deep learning gains its popularity among people, machine learning is a very good way to make prediction in financial area. [2] has used a SVM way to forecast financial time series data, and the experiment result of which has indicated that the use of support vector machine (SVM) to make prediction can be a good idea. Besides, SVM also can do the work of classification. It is used in [3] to predict whether the stock price movement will be up in a short term.

In addition to SVM, the other machine learning methods also can make sense in financial area. [4] has used an artificial neural network to predict the stock values and analyze the result when using more or less hidden layers and different activation function. Besides, the hybrid model, which are used in [5] and [6], has absorbed both data mining techniques and traditional methods to make the prediction of stock price movement. [7] has also used a hybrid model, which combines genetic algorithm with neural network to make prediction, gaining a satisfied result.

2.2 Literature review in deep learning method

In recent years, using deep learning methods to deal problem has shown its superiority when compared with previous machine learning methods, especially in image caption [8] and natural language processing tasks. [9] has compared the efficiency of machine learning and deep learning, which shows that the result of the latter is a bit better than the former.

Traditionally, if we want to solve the problem by deep learning, first we should do is to design an architecture similar with neural network, but it includes more hidden layers and needs a large number of data sets. The more layers and data sets you have, the more features it will be extracted for classification.

The generally used model in deep learning includes two parts: Convolutional neural network [10-13]

and Recurrent neural networks [14-17]. Generally speaking, as the data of stock usually can be seen as sequential data, the frequency of using Recurrent Neural Network to process the data related with time series are higher compared with using Convolutional Neural Network to deal them. [18] and [19] has used LSTM ways to make predictions, which both can get a good result. But that does not mean that LSTM is the first model you should consider when dealing with the problem in financial area. Because the stock data can be seen as a large 2D matrix, [3] has used ANN model to make prediction and gain a satisfied result, both of which have proved that CNN also can be used to do the same thing. Thus, [1] and [9] have tried to use CNN to predict stock price movement. Of course, the result is not inferior to the people who used LSTM to make prediction.

2.3 Convolutional Neural Network

Convolutional Neural Network is a feed-forward neural network. Like the traditional architecture of a neural network including input layers, hidden layers and output layers, convolutional neural network also contains these features and the input of the layer of convolution are the output of the previous layer of convolution or pooling. Of course, they still have some unique features such as pooling layers, full connection layers, etc. The number of hidden layers in a convolutional neural network is more than that in a traditional neural network, which, to some extent, shows that the capability of the neural network. The more the hidden layers are, the higher feature it can extract and recognize from the input. People always use convolutional neural network in computer vision, such as face recognition, image classification [20-21].

3. Method

3.1 Data preprocessing

Data preprocessing is a very important step when you want to get some information from data sets to help you make the prediction. As the initial data may have a lot of noise, it is necessary to reduce them so that they will not interfere with you result. Besides, because some features of data may make no sense, in order to improve efficiency, you should neglect them when you train the data.



Figure 1.Part of the closing price chart from China stock exchange

We got the historical stock data sets of China which covered from January 5th, 2015 to December 29th, 2017 every trading day from the Internet. The data initially contains not only several features, including opening price, closing price, high price, low price, volume, turnover, change rate, etc., but also the name and the code of the stock. The part of closing price chart is shown in Figure 1. Most of them will change once every five minutes. To accelerate the speed of running a program and improve its efficiency, the first thing we should do is to clean the data. [22] has made some experiments with different features for prediction to show that the experiment will achieve the satisfying result when using open, close, high, low and volume as the input features of model. So, we chose the first five features as

our input and neglect irrelevant data such as their names and stock codes. Then, we have removed the data with missing values. Next, as the stock might be suspended one day, it is meaningless to use those data for prediction and they also should be removed from the data sets. One more thing we should note is that there will be a phenomenon in theory that some stock data may not change once every five minutes, so on no account should we regard it as repeated data and remove it from the data sets.

Our task is to do the binary classification instead of regression, so the output of our model only consists of two values—one or zero to show whether the stock price movement will be up or down. Of course, since we adopt supervised way to design the model, labels are of great importance. Unfortunately, the data sets do not contain the labels and we should make the labels ourselves. We have defined a function which will compare the change of closing price between one day and ten days after that day and then get the label. For example, if the closing price would be up after 10 days, the label of that day would be set as 1 to show the stock price would be move up after 10 days and vice versa. People could consider buying the stock when the result of prediction is 1 and selling it after 10 days to get the stock returns. The full data set is divided into two parts: 20% for testing the data and 80% for training the data.

To shorten the computing time, we have vectorized the input data, which means that when we train the model at one time, several numbers of the data will be input into the model. The input form can be defined as following matrix:

$$X = (x^{(1)}, x^{(2)}, \dots, x^{(m)}) \quad (1)$$

Here m indicates the number of samples as inputs. Each sample consist of five features mentioned above, so the input of the neural network is a $m \times 5$ matrix. The output form is basically similar as the input form:

$$Y = (y^{(1)}, y^{(2)}, \dots, y^{(m)}) \quad (2)$$

Here each y is a Boolean value: 1 and 0. Noting that one training sample only has one label.

In order to accelerate the convergence speed when we use gradient descent to find the optimal values, the input data should be normalized because there is too much difference between each feature. We have adopted min-max normalization to keep the value in the range of 0 to 1. The formula is as following:

$$x^* = \frac{x - \min}{\max - \min} \quad (3)$$

Here \max indicates the maximum value of the features and \min indicates the minimum value of the features.

3.2 Convolution function for 1-Dimensional data

Usually, by means of Google's TensorFlow, we can directly use the function "conv2d" to do the computation of convolution in a CNN model which usually accepts a 2D image as its input. But, as our data of the stock belongs to 1D time series data, it is necessary to change the function to help us do the computation. The new function we adopt in the model accepts a number of stock data, which has mentioned above, and other properties such as the number of the filters, width of the filters and stride as its input. Note that the height of the kernel here is not applicable. Then we use a Gaussian distribution to initiate the value of the filters and zero to initiate the biases. The outputs of our function are a number of matrices, and the specific number is decided by the number of the filters. CNN model will extract some features from them and they also will be the input of the next pooling layer after the computation of activation function.

3.3 Convolutional neural network for the stock prediction

By means of conv1d function, we can build a CNN model to help us make the prediction. For getting the final classification values, which will tell us the stock will be up or down in future, we will use

“teaching” number to train the model. Firstly, the vector of stock data will be input into the stock prediction model and we will use the conv1d function to do the computation of convolution. Because our stock prediction model maybe linear and it will be very poor to solve the nonlinear problem, it is necessary to introduce the activation function. Traditionally, the activation function that people always use can be concluded as followings: Tanh, Relu, Sigmoid and Lrelu. To solve the problem of gradient vanishing, we choose the Relu and Lrelu function to do the computation of activation. The Relu function and Lrelu function can be defined as followings:

$$\text{Relu}(x) = \begin{cases} 0 & (x \leq 0) \\ x & (x > 0) \end{cases} \quad (4)$$

$$\text{Lrelu}(x_i) = \begin{cases} a_i x_i & (x \leq 0) \\ x_i & (x > 0) \end{cases} \quad (5)$$

Where x and x_i both are the results of the computation of convolution in the convolutional layer a_i is a hyper-parameter with a very small value and usually we choose 0.01 as its initial value (it also can be fine-tuned to get the best result while training).

Note that Relu function is a very useful activation function to solve the problem of gradient vanishing, but the problem will still exist when the input vectors include negative value [23]. Thus, in the first convolutional layer we choose Relu as our activation function and in the next several convolutional layers it will be replaced by the Lrelu function.

Secondly, the feature maps of the convolutional layer will be the input of pooling layer and they will be pooled to reduce the feature dimension and accelerate the computation. We choose max pooling function to do the work. For example, A pooling layer using a 2×2 filter will subsample its input by the maximum function, that is to say, it will find the maximum in each 2×2 area of the feature maps.

After the computation of several convolutional layers and pooling layers, the output of the last layer will be the input of a fully connected layer, which will help to do the classification tasks. Because our task is to do binary classification, the output of the fully connected layer will be computed by the activation function of SoftMax to get the probability of the stock's movement of up and down and we will choose the bigger to be the final prediction value. Of course, we will also adopt dropout strategy to avoid over-fitting in the fully connected layer and set 0.8 as its value.

The final value of the prediction and the ground-truth will be compared, and we will compute its average error. Our target during the training period is to minimize its error in order to improve its accuracy, so a loss function must be designed and optimized, which can be divided into following two steps:

Firstly, we should compute the result of the SoftMax function used in fully connected layer, which can be represented by the following formula:

$$y_i = \frac{e^{a_i}}{\sum_{k=1}^C e^{a_k}}, i = 2 \quad (6)$$

Where i indicates the number of the classification, a is the output of the fully connected layer and C is the number of a and e is a natural logarithm.

Secondly, the error function will be computed as followings:

$$J(W) = -\sum_{k=1}^n \sum_{i=1}^C t_{ki} \log(y_{ki}) \quad (7)$$

Where n is the number of the samples, C is the number of output of the fully connected layer, t_{ki}

indicates the probability that samples k belongs to class i , y_{ki} indicates the probability that models predict sample k belongs to class i and W is the weightcoefficient of filters. Noting that the loss in each batch will be summed and averaged to get the final loss.

To minimize the loss, the most common way that people use is to take advantage of backpropagation algorithm, which is based on gradient descent [16]. We adopt ADAM algorithm proposed in [17] to do the computation of optimization and use the Equation defined in (8) to update the parameter W .

$$W' = W - \eta \cdot \frac{\partial J(W)}{\partial W} \quad (8)$$

Where W' is the value which will be updated after the computation of each gradient descent step and η is the learning rate set to 0.0001 in the experiment. Figure 2 has visualized the architecture of our model.

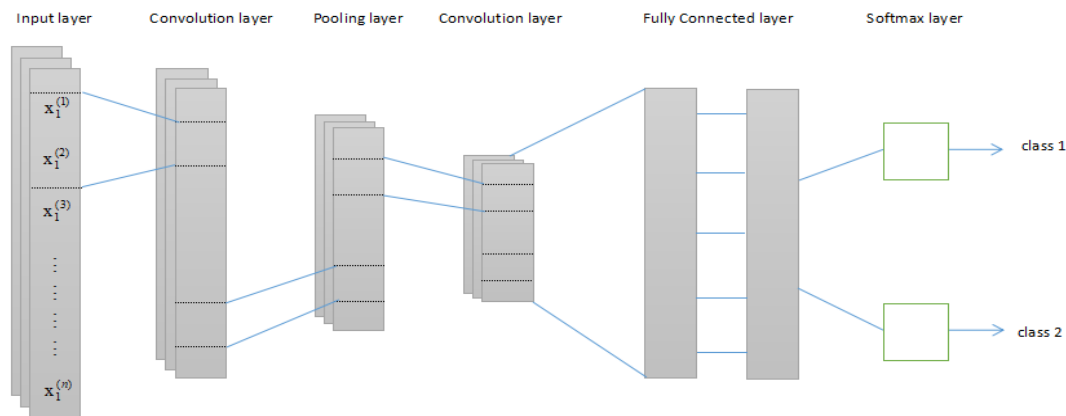


Figure 2.The architecture of our model

4. Result

Some related parameters in our model can be concluded as followings:

(1) Number of layers:6.

Note that too many convolutional layers may lead to complex computation and gradient vanishing or diffusion, and a small number of convolutional layer may make the result unreliable. So, to balance the speed of computation and the efficiency of our model, we have make several tests and decide to choose 6 as our layer number.

(2) Number of fully connection layer:2

We have set 2 fully connection layers and both of them will help to convert the output data into a matrix with different shapes, which will make it easy for us to do the classification by means of SoftMax function.

(3) The number of each filters in convolutional layers and the size of kernel in pooling layers:

32,64 and 128 for the first three convolutional layers and 256 for the others. The size of kernel in pooling layer is 1.

Beside what has been mentioned above, we use batches of 1000 samples and each sample contains 1×4 eigenvector. We have split the data-sets and 80% of that is for training and the rest is for testing. To evaluate the performance, accuracy is far from enough, so we have also proposed another 3 indexes called precision, recall the F1-score to help us make the evaluations.

Accuracy: Given test data-sets, the accuracy is the number of samples correctly classified by the classifier divided by the total number of samples.

Precision and recall: The precision and recall can be calculated by the following formulas from Table 1:

Table 1. Confusion Matrix

Actual class	Prediction class	
	Up	Down
Up	True-positive	False-negative
Down	False-positive	True-negative

$$P = \frac{TP}{TP + FP} \quad (9)$$

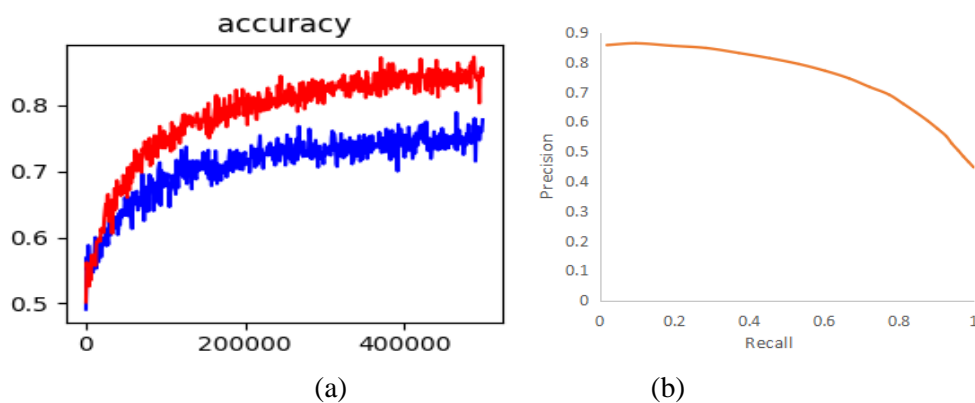
$$R = \frac{TP}{FP + FN} \quad (10)$$

Note that it is hard to find a satisfying result where the value of precision and recall are both high. Generally speaking, when the value of recall is very high, the value of precision is often low, and vice versa. Because of that, to judge which the algorithm is better for the samples easily, we also introduce F1-score, which contains an argument β showing the weight of precision and recall according to the demand. The formula likes the following equations:

$$F_{\omega} = \frac{(1 + \omega^2) \times P \times R}{(\omega^2 \times P) + R} \quad (11)$$

In this paper, we choose $\omega = 1$ to measure the performance.

We have visualized the result of the experiment in Figure 3, including the change between training sets and testing sets in accuracy and precision-recall curve of testing sets.

**Figure 3.** The related evaluation indexes of out model

Where the red or the higher curve indicates the training data and the blue or the lower curve indicates the testing data in picture (a). The curve in picture (b) indicates precision-recall images of testing sets. X-axis is the number of iterations in picture a and recall in picture b while y-axis is the accuracy in picture a and precision in picture b.

Besides, we have used a table to record the number of accuracy in different number of iterations and F1-score using different thresholds to compare whether the model can fit the stock data of different types.

Note the threshold in Table 2 means that firstly the prediction score of test sets should be sorted in descending order and then we choose the biggest number, median and the smallest number of the prediction score named a, b and c as our model's thresholds for classification and calculation.

By comparison of the result, we can conclude that our model is feasible for different stock data.

Besides, it has shown that in addition to Recurrent Neural Network, the convolutional neural network can also solve the problem like stock movement prediction or other predictions in financial areas thanks to its ability of capturing micro-change of data at different time.

Table2. The result of the experiment using different data sources: (a) is the result of accuracy and (b) is the result of F1-score

Data source	1	2	3
Iteration=25000			
Accuracy	73.2%	73.9%	73%
Iteration=500000			
Accuracy	73.8%	75.2%	74.4%

(a)

Data source	1	2	3
Threshold=a			
F1-score	4.5%	3.7%	4.0%
Threshold=b			
F1-score	70.4%	73.0%	72.3%
Threshold=c			
F1-score	60.0%	61.8%	61.6%

(b)

5. Conclusion

In this paper we have introduced a CNN model to make the stock prediction and used a conv1d function to process the 1D data in the convolutional layer. We have also preprocessed stock data which will be input into the model to improve the result of the model. The result has been evaluated by different stock data, and finally indicates that our CNN model is robust and can also be used to make the predictions even if the source data is 1D sequential.

There are some interesting directions for further study:

(1) We can use the stock data with more features to make predictions, for the movement of stock may not only influenced by the features of open, close, high and low prices. Moreover, we can make a comparison about which features are key to the result.

(2) We can use more data sets to judge whether the result will be better, for the number of the data-sets is of great importance in deep learning.

(3) Using other convolutional neural network architectures can also be considered as a good idea, including Google net, Alex net and so on for they have achieved good performance in ILSVRC. We can finetune them to take advantage of them to better solve the problems in financial areas.

References

- [1] J. F. Chen, W. L. Chen, C. P. Huang, S. H. Huang and A. P. Chen, "Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks," *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, Macau, 2016, pp. 87-92.
- [2] L. J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," in *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1506-1518, Nov. 2003.
- [3] Kercheval, Alec N., and Y. Zhang. "Modelling high-frequency limit order book dynamics with support vector machines." *Quantitative Finance* 15.8(2015):1-15.
- [4] K. Abhishek, A. Khairwa, T. Pratap and S. Prakash, "A stock market prediction model using Artificial Neural Network," *Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on*, Coimbatore, 2012, pp. 1-5

- [5] Zhai, Yuzheng, A. Hsu, and S. K. Halgamuge. "Combining News and Technical Indicators in Daily Stock Price Trends Prediction." *International Symposium on Neural Networks: Advances in Neural Networks* Springer-Verlag, 2007:1087-1096.
- [6] Ruiz, Eduardo J., et al. "Correlating financial time series with micro-blogging activity." ACM, 2012:513-522.
- [7] Armano, G., M. Marchesi, and A. Murru. "A hybrid genetic-neural architecture for stock indexes forecasting." *Information Sciences* 170.1(2005):3-33.
- [8] Xu, Kelvin, et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." *Computer Science* (2015):2048-2057.
- [9] A. Tsantekidis, N. Passalis, A. Tefas, J. Kannianen, M. Gabbouj and A. Iosifidis, "Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks," *2017 IEEE 19th Conference on Business Informatics (CBI)*, Thessaloniki, 2017, pp. 7-12.
- [10] Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." *EprintArxiv* (2014).
- [11] Zhang, Xiang, and Y. Lecun. "Text Understanding from Scratch." *Computer Science* (2015).
- [12] Kalchbrenner, Nal, E. Grefenstette, and P. Blunsom. "A Convolutional Neural Network for Modelling Sentences." *EprintArxiv* 1(2014).
- [13] Collobert, Ronan, and J. Weston. "A unified architecture for natural language processing: deep neural networks with multitask learning." *International Conference on Machine Learning* ACM, 2008:160-167.
- [14] Mikolov, Tomas, et al. "Recurrent neural network based language model." *INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September* DBLP, 2010:1045-1048.
- [15] Irsoy, Ozan, and C. Cardie. "Opinion Mining with Deep Recurrent Neural Networks." *Conference on Empirical Methods in Natural Language Processing* 2014:720-728.
- [16] Saad, E. W, D. V. Prokhorov, and D. C. Wunsch. "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks." *IEEE Transactions on Neural Networks* 9.6(1998):1456-70.
- [17] Rather, Akhter Mohiuddin, A. Agarwal, and V. N. Sastry. "Recurrent neural network and a hybrid model for prediction of stock returns." *Expert Systems with Applications* 42.6(2015):3234-3241.
- [18] Chen, Kai, Y. Zhou, and F. Dai. "A LSTM-based method for stock returns prediction: A case study of China stock market." *IEEE International Conference on Big Data* IEEE, 2015:2823-2824.
- [19] Nelson, David M. Q., A. C. M. Pereira, and R. A. D. Oliveira. "Stock market's price movement prediction with LSTM neural networks." *International Joint Conference on Neural Networks* IEEE, 2017:1419-1426.
- [20] A. Raj, S. Gupta and N. K. Verma, "Face detection and recognition based on skin segmentation and CNN," *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, Roorkee, India, 2016, pp. 54-59.
- [21] A. Farooq, S. Anwar, M. Awais and S. Rehman, "A deep CNN based multi-class classification of Alzheimer's disease using MRI," *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, Beijing, 2017, pp. 1-6.
- [22] K. Chen, Y. Zhou and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, 2015, pp. 2823-2824.
- [23] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.