International Journal of Computer Applications Technology and Research
Volume 9–Issue 01, 08-14, 2020, ISSN:-2319–8656

SSE Composite Index Forecasting Model via BP Neural Network with ADAM Optimizer

Yahui Chen
School of Communication Engineering, Chengdu University of Information Technology,
Chengdu, China

Zhan Wen
School of Communication Engineering, Chengdu University of Information Technology,
Chengdu, China

Kangjian Tang*,Wenzao Li
School of Communication Engineering, Chengdu University of Information Technology,
Chengdu, China

Abstract: In China, stock investment is o ne of the important ways for people to manage their finances. Mastering the stock market dynamics can not only bring economic benefits to individuals and enterprises, but also help government to understand Chinese macroeconomic situation. The overall situation of Chinese stock market is mainly displayed through the SSE Composite Index. Therefore, effective prediction of the SSE Composite Index will help investors, enterprises and government agencies to grasp the overall information of the stock market and reduce investment risks in trading. With the development of computer technology, the application of machine learning algorithms to predict stock market volatility has become a hot spot. Among them, BP neural network is the most widely used model, but the gradient descent algorithm used in the model's back propagation has the problem of easily falling into a local minimum. Therefore, the ADAM algorithm to solve this problem was born. This article uses the ADAM optimizer to optimize the BP neural network for SSE Composite Index short-term prediction. The constructed ADAM-BP neural network's Goodness-of-fit index R2 reached 0.986, which means that the model has good prediction performance. In addition, Compared with the BP neural network without the ADAM optimizer, the error evaluation index of the ADAM-BP neural network is significantly reduced.

Keywords: Machine learning; BP neural network; ADAM optimizer; SSE Composite Index; Stock market forecast

1. INTRODUCTION

In China, stock investment is one of the important ways of Chinese citizen's financial management. Since the securities market in mainland China started later than the mature foreign stock market, with greater randomness and worse predictability, Chinese investors urgently need an effective stock market forecasting method to assist investors in trading, reduce risks and blindness in stock investments. Government also need to understand the stock market fluctuations in time to grasp the macro economy. Chinese investors generally observe the overall trend of the Chinese stock market by observing the SSE Composite Index issued by the Shanghai Stock Exchange, because the SSE Composite Index has the earliest compilation date and covers the largest number of listed companies in the Chinese stock market. It can represent the trend of the Chinese stock market. Therefore, the analysis and forecast of the SSE Composite Index is important for investors to fully grasp the overall changes in the Chinese stock market.

In the past, forecasting methods were often: relying on the personal experience of investors, statistical analysis of stock market data by professional teams of investment companies, and researchers using time series models. These prediction methods have indeed achieved certain results, but there will be problems such as information lag and lack of human analysis errors. And the stock market has strong randomness, non-linearity, noise, etc. The prediction itself is difficult, so the accuracy of these prediction methods is not high.

Because machine learning algorithms have the ability to efficiently fit massive data and real-time analysis, they can be used as powerful tools for analyzing and predicting stock market trends. Machine learning algorithms have been applied in the field of quantitative trading and high-frequency trading, and has achieved good returns. For example, Kinjal J and Nisarg A, used BP neural network to predict the Indian stock index and achieved good results (2011)[1];Patel, J, et al. Used a hybrid model of support vector regression, random forest, and support vector machine to make short-term predictions of stock prices, and achieved good results (2015) [2]; Pan, Y. et al. Used a multi-layer support vector machine to predict the weekly price of individual stocks for the next 4 weeks. The accuracy of the prediction results in the first week reached 88%, and the accuracy of the predictions in the following weeks reached 51.3% (2017) [3]; A technology company called Perception in Silicon Valley, U.S.A. uses machine learning algorithms to predict stock market fluctuations, and machines perform stock market trading operations; In China, Flush Software has built an AI stock market forecasting intelligent platform to assist investors in investing.

There are many types of machine learning algorithms. At present, the use of BP neural network as a machine learning algorithm to fit historical market data is a major trend. For example, Ticknor uses financial technical indicators and market prices as input features of neural networks to predict individual stock prices in a certain period of time in the future, thereby obtaining more accurate results than the linear method (2013) [4]. Rather, A. et al. Compared the prediction results of the linear model with the neural network model in the short-term forecast of the stock market and showed that the neural network model has a better fit for the non-linear high-noise system of the stock market (2015) [5] . The principle of this model is simple, and only a single hidden layer can be used to infinitely approximate any non-linear function(2016)[6]. However, its backpropagation strategy is a gradient descent algorithm, which has the problem of easily falling into the local minimum and unable to reach the best point. Therefore, the ADAM algorithm came into being. It can adapt the learning rate and solve the problem of falling into the local optimal solution(2015)[7].

Therefore, this article will use the ADAM optimizer to optimize a single hidden layer BP neural network to build a short-term prediction model of the SSE Composite Index. In order to use artificial intelligence to predict stock market fluctuations, assist Chinese investors to fully grasp the stock market trends and make scientific investment decisions.

2. Stock market forecasting methods

2.1 Introduction of the SSE Composite Index

The Shanghai Stock Exchange Index is an index issued by the Shanghai Stock Exchange. It includes the Shanghai Stock Exchange Composite Index(SSE Composite Index), Shanghai Stock Exchange 50 Index, and Shanghai Stock Exchange Dividend Index. Among them, the SSE Composite Index, as the earliest indicator compiled and issued in China, integrates information on all the stocks listed on the Shanghai Stock Exchange. It is currently the best index to reflect the overall situation of the Chinese stock market.

Since the SSE Composite Index covers the largest stock market information in China's stock market, and has the longest history and the most extensive application, this article will choose to predict the key index of the SSE Composite Index. Among them, the Shanghai Stock Exchange closing price is the most commonly used indicator, and it can most directly reflect the changing trend of the broader market. Therefore, this article chooses the Shanghai Stock Closing Price as the forecast target.

The input features selected the Shanghai Stock Exchange's opening price, closing price , volume, the highest price, the lowest price and Stock price change. The calculation process of the above indicators is more complicated. The calculation is based on the weight of the issued share capital of all sample stocks listed on the Shanghai Stock Exchange. Every day, the SSE calculates and publishes real-time indices through a program.

2.2 BP neural network and ADAM optimizer

The full name of BP neural network is error back propagation neural network. The implementation process of this model is divided into two parts, one is the forward propagation process, and the other is the back propagation process. The forward propagation process is a process of calculating the input features of each layer of the neuron to obtain the result; the back propagation process is a model optimization process that performs parameter correction based on the gap between the obtained result and the true value.

A general back propagation strategy uses a gradient descent algorithm, but the gradient descent algorithm does not have the function of adjusting the learning rate according to the gradient descent rate, it is easy to fall into a local optimal solution. Therefore, this article will use the ADAM optimizer to replace the gradient descent optimizer in the BP neural network to adaptive learning rate, avoid falling into the local optimal solution. And the single hidden BP neural network has been proved to be able to approach arbitrary functions indefinitely, so this paper will build a single hidden layer BP neural network based on the ADAM optimizer to predict the SSE Composite Index. The detailed principle will be introduced below. The figure below shows the structure of a single hidden layer BP neural network:

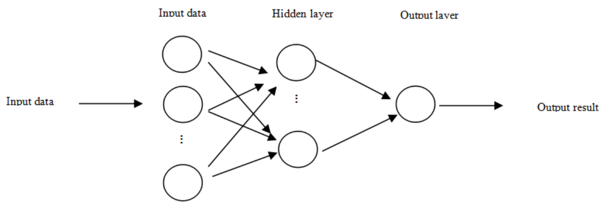

Figure1 : Single hidden layer BP neural network structure

The function that reflects the mapping between the output result and the input data is as follows:

$$F(X) = \omega_2 g(\omega_1 X + b_1) + b_2 \quad (1)$$

Given a set of training data X,

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \quad (2)$$

Each column of the X matrix represents an input vector, and the number of rows represents the number of features.

$$\omega = \begin{bmatrix} \omega_{11} & \cdots & \omega_{1m} \\ \vdots & \ddots & \vdots \\ \omega_{h1} & \cdots & \omega_{hm} \end{bmatrix} \quad (3)$$

ω_1 and ω_2 also corresponds to a matrix form, which represents the weight of each neuron in the hidden layer and the output layer, respectively. The number of rows h is related to the number of neurons in the hidden layer, and the number of columns is the same as the number of input vector features.

The parameter b represents the bias that comes with each neuron. The $g(\cdot)$ in the formula is the activation function of the hidden layer. It is a non-linear function in each neuron in the neural network. The result obtained by the activation function is the output of the neuron. The purpose of its existence is to make the values in the network no longer be a linear combination of inputs, so that the entire network is non-linear and can approximate any non-linear function. The Relu function with a steep gradient is the main choice of the current neural network activation function as well as in this paper. Its analytical formula is as follows:

$$RELU = MAX(0, Z) \quad (4)$$

The process of obtaining the output data after activation by the activation function is the forward propagation process of the BP neural network.

To build a neural network model, you need to get the neural network weights when the error function reaches a minimum. The process of updating the weights, that is, updating the weights to minimize the loss function, is the back propagation process. In this paper, J(·) is the loss function. Generally, the mean square error is used as the loss function in the prediction problem:

$$J = \frac{1}{n}\sum_{1}^{n}(y-\hat{y})^2 \qquad (5)$$

Where is the true value and is the predicted value. n is the number of samples. The loss function represents the average distance between the predicted value and the true value, and is a criterion for measuring the accuracy of the model. The smaller the value, the higher the model accuracy. The ADAM optimizer minimizes the loss function by the following formula for updating weights:

$$dw = \frac{\partial J(\omega_i)}{\partial \omega_i} \qquad (6)$$

$$v_{dw}^1 = dw \qquad (7)$$

$$v_{dw}^2 = \beta_1 v_{dw}^1 + (1-\beta_1)dw$$

$$v_{dw}^3 = \beta_1 v_{dw}^2 + (1-\beta_1)dw$$

… …

$$v_{dw}^{n+1} = \beta_1 v_{dw}^n + (1-\beta_1)dw \qquad (8)$$

dw is the calculation of the partial derivatives of the parameters ω_i in the loss function J(·) . v_{dw}^1 refers to the initial gradient value, and the secondary gradient is modified by the formula. The above formula indicates that the gradient of the n-th generation is not only related to the current gradient, but also to the gradient of the previous n-generation. (The smaller the value of n, the smaller the correlation, because $\beta_1 = 0.9$) In this case, when our gradient suddenly becomes larger or smaller, the improved gradient will not abruptly change, which is conducive to the stability of the model.

The ADAM algorithm also introduces the introduction of differential weighted averages to adjust the size of the learning rate. The calculation method is as follows:

$$s_{dw} = \beta_2 s_{dw} + (1-\beta_2)dw^2 \qquad (9)$$

$$s_{db} = \beta_2 s_{db} + (1-\beta_2)db^2 \qquad (10)$$

The general setting of β_2 in the above formula is 0.999, the second moment of the gradient is calculated as dw^2、db^2. s_{dw}、s_{db} used to adjust the learning rate, after calculating according to the above steps, you can update the parameters, as shown in the following formula:

$$w = w - \alpha\frac{v_{dw}^c}{\sqrt{s_{dw}^c + \varepsilon}} \qquad (11)$$

$$b = b - \alpha\frac{v_{db}^c}{\sqrt{s_{db}^c + \varepsilon}} \qquad (12)$$

α is the learning rate, which represents the magnitude of the gradient descent. If it is too large, it will be difficult for the model to reach the minimum value. If it is too small, the gradient descent rate will be too slow. According to experience, the initial value is generally set to 0.001. v_{dw}^c, s_{dw}^c , v_{db}^c , s_{db}^c called a modified gradient. Because the initialization value is generally 0, the parameters will be biased, so the correction index of the above index needs to be calculated. $\frac{v_{db}^c}{\sqrt{s_{db}^c + \varepsilon}}$ can make the learning rate adaptive. When the gradient declines too fast, the learning rate will become smaller, and when the gradient declines too slowly, the learning rate will increase accordingly.

The above is the implementation principle of BP neural network based on ADAM optimizer.

3. Model implementation

3.1 Data collection

This article collected a total of 1691 pieces of data on SSE Composite Index from 2012 to 2018, provided by the domestic stock market data collection professional website "Forecaster Network". Some historical market data are as follows:

Table1: The Historical SSE Composite Index (Partial)

date	open	Close	lowest	highest	volu-me	cha-nge
2017/12/1	3315.1	3317.62	3302.44	3324.52	13919830000	0.000129628
2017/12/4	3310.38	3309.62	3304.1	3324	14805328800	-0.002411367
2017/12/5	3301.69	3303.68	3300.51	3315.74	20827886200	-0.001794768
2017/12/6	3291.31	3293.96	3254.61	3296.2	15160445200	-0.002942174
2017/12/7	3283.28	3272.05	3259.16	3291.28	13210590000	-0.006651568
2017/12/8	3264.48	3289.99	3258.76	3297.13	13320931400	0.005482801
2017/	3290.	3322.	3288.	3322.67	131965	0.009790

12/11	49	2	29		98400	303
2017/12/12	3320.31	3280.81	3280.33	3320.31	12460482700	-0.012458612
2017/12/13	3278.4	3303.04	3273.32	3304.01	11199864700	0.006775766

3.2 Feature engineering

The forecast target selected in this article is the closing price of the SSE Composite Index. Based on previous research and actual needs, this article chooses to use five input features, including the closing price, opening price, change rate, highest price and lowest price.

And the method called single-step forecast is used to construct input features. The single-step prediction is also called one-step prediction. The specific method is to use the observed value at the previous moment as input data to predict the index fluctuation at the next moment, and the actual observed value at the next moment will become part of the next input data to participate in the prediction. That is, using the SSE Composite Index of the previous four days to predict closing price of the SSE Composite Index next day. The length of the input feature is 20. The Partial input data are as follows:

Table 2: Input feature vector display (partial)

The **date of prediction**	**Opening price 1**	**Opening price 2**	**Opening price 3**	**Opening price 4**	**Closing price 1**
2012/1/10	2212	2160.9	2148.15	2164.74	2169.39
2012/1/11	2160.9	2148.15	2164.74	2221.83	2148.45
2012/1/12	2148.15	2164.74	2221.83	2282.91	2163.4
2012/1/13	2164.74	2221.83	2282.91	2268.74	2225.89
2012/1/16	2221.83	2282.91	2268.74	2277.08	2285.74
2012/1/17	2282.91	2268.74	2277.08	2230.43	2276.05
2012/1/18	2268.74	2277.08	2230.43	2206.53	2275.01
2012/1/19	2277.08	2230.43	2206.53	2298.83	2244.58
2012/1/20	2230.43	2206.53	2298.83	2266.08	2206.19
2012/1/30	2206.53	2298.83	2266.08	2300.5	2298.38
2012/1/31	2298.83	2266.08	2300.5	2324.49	2266.38
2012/2/1	2266.08	2300.5	2324.49	2285.95	2296.08
2012/2/2	2300.5	2324.49	2285.95	2288.07	2319.12
2012/2/3	2324.49	2285.95	2288.07	2273.85	2285.04
2012/2/6	2285.95	2288.07	2273.85	2306.66	2292.61
2012/2/7	2288.07	2273.85	2306.66	2334.25	2268.08
2012/2/8	2273.85	2306.66	2334.25	2319.42	2312.56
2012/2/9	2306.66	2334.25	2319.42	2291.81	2330.41

Closing price 2	**Closing price 3**	**Closing price 4**	**Change rate 1**	**Change rate 2**
2148.45	2163.4	2225.89	-0.013652254	-0.009651561
2163.4	2225.89	2285.74	-0.009651561	0.00695524
2225.89	2285.74	2276.05	0.00695524	0.028887466
2285.74	2276.05	2275.01	0.028887466	0.026889918
2276.05	2275.01	2244.58	0.026889918	-0.00424282
2275.01	2244.58	2206.19	-0.00424282	-0.000455175
2244.58	2206.19	2298.38	0.000455175	0.013375765
2206.19	2298.38	2266.38	-0.013375765	-0.017102086

2298.38	2266.38	2296.08	-0.017102086	0.041783742
2266.38	2296.08	2319.12	0.041783742	-0.013919394
2296.08	2319.12	2285.04	-0.013919394	0.013100604
2319.12	2285.04	2292.61	0.013100604	0.010035822
2285.04	2292.61	2268.08	0.010035822	-0.014695242
2292.61	2268.08	2312.56	-0.014695242	0.00331373
2268.08	2312.56	2330.41	0.00331373	-0.010699596
2312.56	2330.41	2331.14	-0.010699596	0.019609538
2330.41	2331.14	2291.9	0.019609538	0.0077183
2331.14	2291.9	2347.53	0.0077183	0.000313679

Change rate 3	**Change rate 4**	**The lowest price1**	**The lowest price 2**	**The lowest price 3**
0.00695524	0.028887466	2168.64	2145.56	2132.63
0.028887466	0.026889918	2145.56	2132.63	2148.45
0.026889918	-0.00424282	2132.63	2148.45	2218.28
-0.00424282	-0.000455175	2148.45	2218.28	2265.19
-0.000455175	-0.013375765	2218.28	2265.19	2265.26
-0.013375765	-0.017102086	2265.19	2265.26	2225.74
-0.017102086	0.041783742	2265.26	2225.74	2206.05
0.041783742	-0.013919394	2225.74	2206.05	2196.12
-0.013919394	0.013100604	2206.05	2196.12	2257.9
0.013100604	0.010035822	2196.12	2257.9	2259.34

0.010035822	-0.014695242	2257.9	2259.34	2293.89
-0.014695242	0.00331373	2259.34	2293.89	2284.29
0.00331373	-0.010699596	2293.89	2284.29	2277.06
-0.010699596	0.019609538	2284.29	2277.06	2263.34
0.019609538	0.0077183	2277.06	2263.34	2268.69
0.0077183	0.000313679	2263.34	2268.69	2300.98
0.000313679	-0.016830421	2268.69	2300.98	2317.42
-0.016830421	0.024271544	2300.98	2317.42	2278.83

The lowest price 4	**The highest price 1**	**The highest price 2**	**The highest price 3**	**The highest price 4**
2148.45	2217.52	2183.4	2164.32	2226.22
2218.28	2183.4	2164.32	2226.22	2288.63
2265.19	2164.32	2226.22	2288.63	2290.64
2265.26	2226.22	2288.63	2290.64	2295.22
2225.74	2288.63	2290.64	2295.22	2281.53
2206.05	2290.64	2295.22	2281.53	2241.26
2196.12	2295.22	2281.53	2241.26	2298.38
2257.9	2281.53	2241.26	2298.38	2311.58
2259.34	2241.26	2298.38	2311.58	2305.71
2293.89	2298.38	2311.58	2305.71	2322.89
2284.29	2311.58	2305.71	2322.89	2324.49
2277.06	2305.71	2322.89	2324.49	2296.38
2263.34	2322.89	2324.49	2296.38	2305.86
2268.69	2324.49	2296.38	2305.86	2312.56
2300.98	2296.38	2305.86	2312.56	2336.27
2317.42	2305.86	2312.56	2336.27	2341.58

2278.83	2312.56	2336.27	2341.58	2319.71
2286.82	2336.27	2341.58	2319.71	2350.97

3.3 model evaluation criteria

In the experiments in this paper, three model evaluation standards will be selected, namely Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R2 determination coefficient. Mean absolute error MAE (Mean Absolute Error) is also called L1 norm loss, the formula is as follows:

$$\mathrm{MAE} = \frac{1}{M}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (13)$$

MAE refers to the average of the absolute values of the residuals of the actual and predicted values. Although this indicator can better measure the quality of the regression model, the existence of the absolute value causes the function to be non-smooth and cannot be derived at some points. Change the absolute value to the square of the residual, which is the mean square error. However, the mean square error is inconsistent with the dimension of the target variable. In order to ensure the consistency of the dimension, the root mean square error (RMSE) of the root mean square error (RMSE) is obtained by performing the root mean square error. RMSE is also called L2 norm loss, the formula is as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (14)$$

In order to measure how well the model fits the data, this article will also use R2 to determine the coefficient, which reflects the proportion of the total variation of the dependent variable that can be explained by the independent variable through a regression relationship. The calculation formula is as follows:

$$R^2(y_i, \hat{y}_i) = 1 - \frac{\frac{1}{M}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\mathrm{VAR}(y)} \quad (15)$$

R2 is the difference between one and the ratio between the mean square error and the true value variance. If the result is 0, it means that there is no predictive relationship between the model prediction and the predictive dependent variable. The general result is a number between 0-1. The larger the number in this interval, the higher the degree of fit between the model and the data.

3.4 Empirical results of ADAM-BP neural network Forecasting the SSE Composite Index

During each experiment, the data will be randomly scrambled, and the training set and test set will be divided at a 7: 3 ratio according to the division ratio of the training set and the test set commonly used in the literature to avoid data distribution problems Overfitting and low accuracy. And considering the generalization of the model, the performance of the model is mainly reflected by the results on the test set.

The main tuning parameter is the number of hidden neurons. According to the Kolmogrov's theory, the number of theoretical optimal hidden layer neurons is twice the size of the input vector plus one. This article will focus on the theoretically optimal number of neurons, set the range, and choose the best number of hidden layer neurons based on the evaluation results to improve the persuasiveness of the experiment. The following are the evaluation results with the different number of hidden layers' neurons on the test set:

Table 3: ADAM-BP Neural Network's Evaluation Results

Number of neurons	R2	MAE	RMSE
39	0.985	47.915	77.454
40	0.985	46.917	76.748
41	0.986	46.354	74.787
42	0.985	47.63	74.807
43	0.986	46.375	75.601

As can be seen from the table above, the number of best hidden layer neurons when predicting the closing price of the SSE Composite Index is 41.

3.5 Empirical results of BP Neural Network Forecasting the SSE Composite Index

In order to prove the optimization effect of the ADAM optimizer, a BP neural network based on a gradient descent algorithm will be constructed for comparison. The BP neural network model evaluation of the number of different hidden layer neurons is as follows:

Table 4: BP Neural Network's Evaluation Results

Number of neurons	R2	MAE	RMSE
39	0.971	61.833	106.897
40	0.972	67.666	105.081
41	0.979	63.814	91.325
42	0.982	58.187	83.99
43	0.975	59.697	98.736

It can be seen that in the process of predicting the closing price of the SSE Composite Index, the model works best when the number of neurons in the hidden layer is 42.

3.6 Model comparison

The following compares the performance of the two types of the above BP neural network models:

Table 5: Evaluation of the SSE Composite Index Forecasted by Different Neural Network Models

Evaluation indicators	Model	R2	MAE	RMSE
SSE Composite Index closing price	BP Neural Networks	0.982	58.187	83.99
	ADAM-BP Neural Networks	0.986	46.354	74.787

It can be seen that the R2 of the ADAM-BP neural network has increased, which proves that the model's ability to fit the data increases. MAE and RMSE have significantly decreased compared with the BP neural network. This proves that the use of ADAM optimizer is effective for the improvement of BP neural network.

4. Conclusion

This article shows that BP neural networks are actually suitable for the non-linear subject like stock market prediction, and can achieve high accuracy. R2 of the BP Neural network using gradient descent algorithm for back propagation is 0.982. The evaluation indicators representing the errors are 58.187(MAE), 83.99(RMSE). It can be seen that the R2 of the ADAM-BP neural network has increased, which means the model's ability to fit the data has increased. Compared with the BP neural network without the ADAM optimizer, the error of the ADAM-BP neural network is significantly reduced. Therefore, using ADAM optimizer is a efficacious method to optimize BP neural network's ability. And the final model called ADAM-BP neural network for SSE Composite Index short-term prediction is a useful tool to assist investors, enterprises and government agencies to grasp the overall information of the stock market and reduce investment risks in trading.

5. REFERENCES

[1] Kinjal J, Nisarg A. Forecasting of Indian Stock Market Index S&P CNX Nifty 50 Using Artificial Intelligence. Behavioral & Experimental[J]. Finance Journals, 2011（79）：61-79..

[2] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications, 42(4), 2162–2172.

[3] Pan, Y., Xiao, Z., Wang, X., & Yang, D. (2017). A multiple support vector machine approach to stock index forecasting with mixed frequency sampling. Knowledge-Based Systems, 122, 90–102.

[4] Ticknor J L. A Bayesian regularized artificial neural network for stock market forecasting [J]. Expert Systems with Applications, 2013, 40(14):5501-5506.

[5] Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. Expert Systems with Applications, 42(6), 3234–3241.

[6] J. Konečný, J. Liu, P. Richtárik and M. Takáč, "Mini-Batch Semi-Stochastic Gradient Descent in the Proximal Setting," in IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 2, pp. 242-255, March 2016.

[7] Diederik P. KingmaJimmy Lei Ba. Adam: a Method for Stochastic Optimization International Conference On Learning Representations ,2015.