International Journal of Engineering & Technology, 7 (4) (2018) 3485-3488

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.15892
Research paper

Prediction of stock market using cascade correlation neural network with principal component analysis

K.Velusamy [1] *, R. Amalraj [1]

[1] *Department of Computer Science, Sri Vasavi College, Erode, Tamilnadu, India*
**Corresponding author E-mail:kvelusamy85@gmail.com*

Abstract

Financial forecasting has gained a significant attention among the researchers and investors.A cascade correlation neural network (CCNN) with principal component analysis (PCA) is developed for financial time series forecasting in this research work.In this paper, the PCA method is used to extract the vital components of the input data, and then the extracted features give the input to the CCNN to carry out the financial time series prediction.A comparison is made with conventional back propagation neural network (BPNN) and CCNN.The em-pirical result shows that the proposed prediction model demonstrates a superior performance in financial time series forecasting.For evalu-ating the performance of the proposed model, the empirical research is applied to well known stock market data sets such as S&P 50 Sensex and Nifty 50.

Keywords:*Financial Time Series Analysis; Stock Price Prediction; Principal Component Analysis; Cascade Correlation Neural Networks.*

1. Introduction

The stock price prediction is one of the major issues in the financial time series analysis which is the deciding the factor of economical position of the country. In the earlier stages, the stock price prediction was done with the help of statistical prediction model including moving average (MA), autoregressive integrated moving average (ARIMA) [1] models, etc. There is no linear configuration captured by such above statistical methods due to a relationship structure assumed in the time series datasets. Recently, artificial neural networks (ANNs) model have been extensively used in the time series forecasting [2]. The ANNs models are able to approximate various nonlinearities in the data. The major advantages of ANNs are the capability of their flexible nonlinear model, to grant a suitable mapping between input and output, and a data-driven model which is suitable for any empirical datasets [3].

In the literature, various neural network architectures have been developed in order to improve the performance of the prediction of the stock market price. The back propagation neural network is the most powerful in solving neural network training algorithm which is widely applied to financial stock market prediction [4]. However, the BPNNs algorithm have many weaknesses such as learning speed is low, take more computation time, falling into local minima, and etc,.[5]. Additionally, BPNN also easily falls into under fitting / over fitting due to the selection of the inappropriate hidden neurons in the hidden layer [6]. Hence, to avoid under fitting / over fitting of the neural network architecture, the cascade correlation neural networks is used in this paper for financial stock price prediction. The cascade correlation neural network (CCNN) is a famous constructive learning algorithm which is defining the architecture dynamically [7]. The CCNN algorithm is to establish the hidden neurons one by one to the network during training process.The CCNN takes more computation time, slow convergence rate and produce low efficiency due to the high dimensionality of the data [8]. Hence, to overwrite the above mentioned shortcomings, the principal component analysis (PCA) is used as a dimensionality reduction model in this paper to reduce the dimensionality of the input data in order to improve the generalization performance of the cascade correlation neural network. In this paper, the principal component analysis is integrated with cascade correlation neural networks for predicting stock price and the proposed method is to extract the principal components from the technical indicators of stock market data with the help of PCA then use the output of the PCA as the input for CCNN. The proposed PCA-CCNN method is to remove the redundant values of the original data and take away the correlation between the inputs.

Main contributions of this paper are as follows:

- To improve the quality of stock market data
- To decrease the computational efforts
- To avoid the over fitting of the neural network architecture
- To improve the efficiency of the financial stock price prediction

2. Principal component analysis

During the training process of neural network architecture, the learning algorithm can lead to high computation efforts due to the high dimensionality of raw data [9]. Therefore, there is chance for over fitting in learning algorithm that disturbs the efficiency of the solution.The principal component analysis is the most famous linear dimensionality reduction algorithms and a statistical method for extracting a lesser number of interrelated features from a large set of features [10].

The goal of the PCA is to find a set of $'k'$ orthogonal vectors in the dataset based on their variance. In general, $'k'$ is the lesser dimension then the dimension of the original dataset with high variance, maintains most of the intrinsic information in the datasets and it can discard the feature which has low variance [11]. The description of the every step of PCA algorithm is defined as follows

Algorithm 1: Algorithm for PCA

Step 1: Given a Dataset $X=(x_1,x_2,....,x_N)$, where x_i is the ith data points and N is the total number of data points.

Step 2: Calculate the mean values for all data points as follows :

$$\mu=\frac{1}{N}\sum_{i=1}^{N} x_i \quad (1)$$

Step 3: Subtract the mean from all data points as follows,

$$D=\sum_{i=1}^{N}(x_i-\mu) \quad (2)$$

Step 4: Calculate the covariance matrix as follows,

$$\Sigma=\frac{1}{N-1}D\times D^T \quad (3)$$

Step 5: Calculate the eigenvectors V and Eigen values λ of the covariance matrix (Σ).

Step 6: Sort eigenvectors according to their Eigen values

Step 7: Choose the eigenvectors that have the largest Eigen values $W=\{v_1,v_2,...,v_k\}$ and chooses eigenvectors represent the projection space of PCA.

Step 8: All data points are projected on the lesser dimensional space of the PCA as follows,

$$Y=W^T D \quad (4)$$

3. Cascade correlation neural networks

The cascade correlation neural network algorithm (CCNN) is the most popular supervised constructive learning algorithm [7, 12]. the advantage is to determine the size and topology ofthe architecture of the CCNN dynamically instead of fixed size networks and its architecture shown in Fig. 1 [13]. Specifically, the CCNN architecture starts without hidden unit and then adds one by one according to the given performance principle is satisfied[8]. In CCNN algorithm, learning process commences without hidden unit, the direct connection made between input layer and output layer that are trained the whole training set to reach convergence as possible level [14]. After reaching the convergence level, there is no considerable error reduction happened then presents learning process is concluded.Consequently, the learning process starts with two phases such as input phase and output phase.

First, learning process moves to the input phase to take on new hidden neuron. To add the new hidden unit, the learning process of input phase is commenced with a new candidate unit that receives input signals from all input units and pre-existing hidden units.The learning process of input phase is to adjust weight to maximize the correlation between output of the candidate unit and the residual errors. When the correlation value is improved, the learning process of input phase is stopped. Correlation value is defined as follows

$$C=\sum_i\left|\sum_j(v_j-\bar{v})(E_{j,i}-\overline{E_o})\right| \quad (5)$$

Where, i is the output of the networks at which error is calculated, j^{th} training pattern, 'v' is the output value of the candidate units, E_o is the residual error output at node 'o', '$\bar{v}$'-is the value of 'v' averages value of all patterns, $\bar{E}_o$ is the value of E_o averages overall patterns.The gradient value C is obtained as follows,

$$\frac{\partial c}{\partial w_i}=\sum_{j,i}\sigma_i(E_{i,j}-\bar{E}_i)a'_j I_{m,j} \quad (6)$$

Where, σ_i is the sign of correlation values, a'_j is the derivative for j^{th} pattern. $I_{m,j}$ is the candidate unit inputs which are received from the node m and j^{th} pattern. Finally, the present best candidate unit is added to the active network and its weights are frozen. A residual error is smaller than after adding the new candidate unit to the active network.

Second, the learning process of output phase is started with adjustable output weights to reduce the error between actual and predicted output. In this stage, the output phase of weights only is adjusted and other weights are frozen. After completion of present learning process, network error is calculated, it may reach optimal error then learning process is terminated, otherwise the process is shifted to the input phase to recruit a new hidden unit. The aforesaid two phases of learning process is to be continued until it achieves better error or i reach maximum iterations.

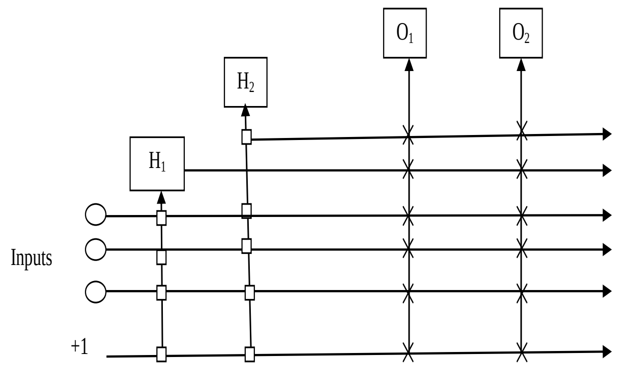

Fig. 1: Architecture of CCNN.

4. Proposed PCA+CCNN prediction model

The robustness of the proposed model is evaluated using two well known stock indices such as Nifty 50 and S&P BSE Sensex that datasets cover total 3350 trading days and the periods from January 3, 2005 to June 30, 2018. The datasets have obtained from NSE and BSE in financial websites. The overview of the proposed model is demonstrated in Fig. 2. In this research works, connection weights are initiated random values in the range between [-1.0, 1.0], its error termination environment is set 0.0005. The original datasets are normalized into the range [-1.0, 1.0] as follows.

$$X'_i=\frac{X_i-X_{min}}{X_{max}-X_{min}}(X'_{max}-X'_{min})+X'_i \quad (7)$$

Where, X'_{max},X'_{min} is the minimum and maximum target value, X_i are the present actual input data, $,X_{max},X_{min}$ maximum and minimum values of scaling factors such as 0, 1 respectively.The sigmoid activation functions is used as activation function of both the input and output phase and it is defined as follows,

$$g(x)=\left(1-\exp(-x)\Big/1+\exp(-x)\right) \quad (8)$$

The performance of the PCA_CCNN is evaluated with the help of statistical methods such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE),

$$RMSE=\sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(t_{(i)}-y_{(i)}\right)^2} \quad (9)$$

$$MAE=\frac{1}{N}\sum_{t=1}^{N}\left(t_{(i)}-y_{(i)}\right) \quad (10)$$

Where,

- $t_{(i)}$ - Target value,
- $y_{(i)}$ - Predicted output of ANN Predictors
- $'N'$ - Total number of data points

The best performance of the neural network architecture is considered a smaller value of RMSE and MAE.The experimental results are conducted using MATLAB 2015.

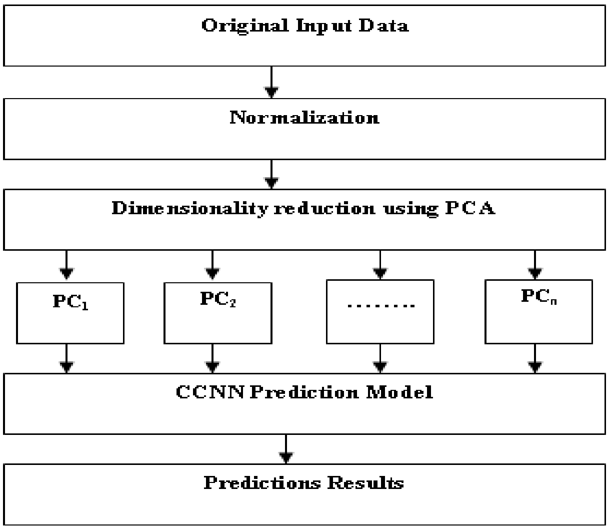

Fig. 2: Overview of the Proposed Prediction Model.

4.1. Dimensionality reduction using PCA

PCA is the most widely used dimensionality reduction method and it is identifying latent structure. The advantage of the PCA is to describe to difficult multidimensional data with less PCs without losing expensive information [15]. Use the PCA method, to extract the principle component from the stock market data which is used as the input variable for the CCNN.

The stock market data contains four variables such as open price, highest price, lowest price, closing price. PCA produced principal components are ordered based on their variance values, then projected in the principal component and the component are transformed to lower dimensional data, which provides input for the CCNN. Here, two well known stock market indices are used to evaluate the performance of the proposed model.

Table 1: Results of the Variance and Cumulative Variance of the Each Component for S &P Sensex Datasets

Components	Eigen Values	Variance (%)	Cumulative Variance (%)
1	2.13535E8	99.98%	99.98%
2	21329.08566	0.01%	99.99%
3	11538.6572	0.01%	100.00%
4	1704.915	0.00%	100.00%

Table 2: Results of the Variance and Cumulative Variance of the each component for Nifty 50 Datasets

Components	Eigen Values	Variance (%)	Cumulative Variance (%)
1	175906.74957	99.99%	99.99%
2	13.66482	0.01%	100.00%
3	4.21066	0.00%	100.00%
4	2.61E-4	0.00%	100.00%

After applying PCA for dimensionality reduction, the amount of features can be reduced according to its most of the variance of the data sets. Table 1 shows the variance values of the S & P BSE Sensex, which is ordered according to their variance values and its graphical representation is shown in Fig. 3, whereas, Table 2 shows details of variance values after applying PCA for Nifty 50 and its graphical representation in Fig. 4.In both datasets, first two principle components are used as input variable for neural network predictor.

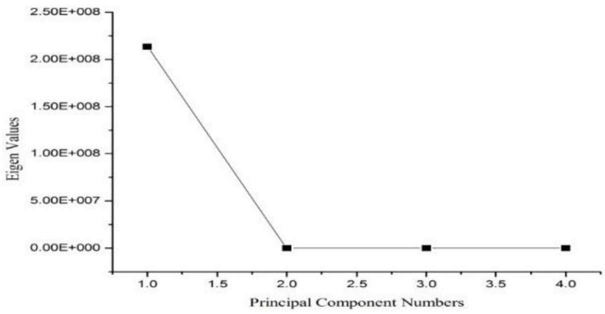

Fig. 3: Variance Values of the Each Principal Component Features for S&P BSE Sensex.

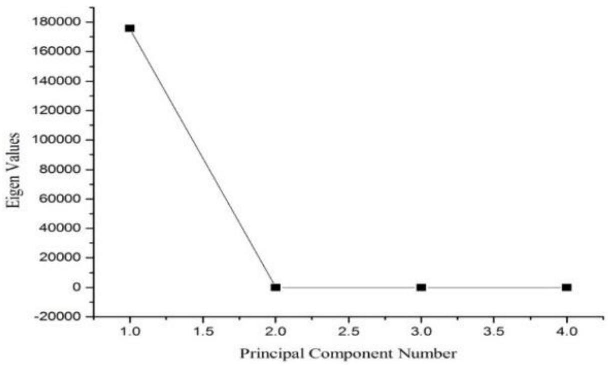

Fig. 4: Variance Values of the Each Principal Component Features for S&P BSE Sensex.

4.2. Prediction using proposed PCA-CCNN

The cascade correlation neural network is a famous constructive neural network architecture which has been applied to many applications.In CCNN algorithm, the weight adjustment is done with the help of Quickprop algorithm in both input and output phase.

Table 3: Performance Results of the S &P BSE Sensex Dataset

Model	MSE	MAE
PCA-CCNN	0.0086	0.0059
CCNN	0.0088	0.0064
BPNN	0.0142	0.0106

Table 4: Performance Results of the Nifty 50 Dataset

Model	MSE	MAE
PCA-CCNN	0.0064	0.0051
CCNN	0.0090	0.0063
BPNN	0.0092	0.0066

4.3. Performance comparisons

In order to evaluate the performance of the proposed PCA_CCNN, it is applied to well known stock market data such as S & P BSE Dataset and Nifty 50. Similarly, we established BPNN, CCNN for comparing the performance of proposed model to predict closing index of the next coming trading day.

The result of the S & P BSE dataset is shown in Table 4, the result of the proposed model achieved higher and exact values compared with CCNN and BPNN and its comparison results as a graphical representation shown in Fig. 5.

In Table 5, the result of the Nifty 50 dataset is produced with higher prediction accuracy while using the proposed model the PCA_CCNN prediction algorithm. Also, graphical representation of the Nifty 50 dataset is illustrated in Fig. 6. Generally, in both stock market indices datasets, the proposed prediction algorithm has produced significant results such as high convergence speed, avoiding over fitting or under fitting compared with CCNN and BPNN.

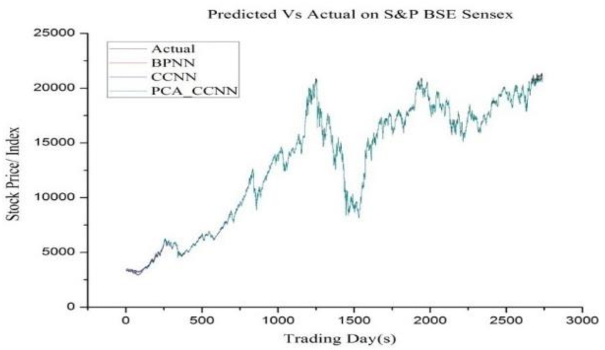

Fig. 5: Performance Comparisons of Actual vs. Predicted Values for S&P BSE Sensex Dataset.

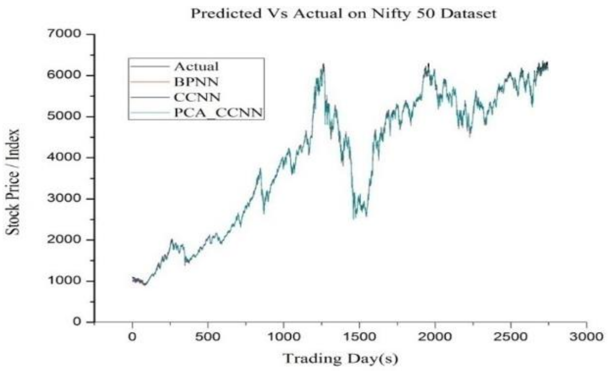

Fig. 6: Performance Comparisons of Actual vs. Predicted Values for Nifty 50 Dataset.

5. Conclusion

The accuracy of the prediction model has disturbed due to the high dimensionality of the stock market data sets. The current research work proposes a method combining PCA techniques for dimensionality reduction based on their variance with the CCNN method for developing the prediction algorithm to generate a stock index prediction for achieving high profit is discussed in details. To further estimate the strength of the proposed method, two well known stock market data is involved. The experimental result shows that the proposed method has enhanced results than conventional CCNN and BPNN.

References

[1] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," Neurocomputing, vol. 50, pp. 159-175, 2003.https://doi.org/10.1016/S0925-2312(01)00702-0.

[2] S. Asadi, E. Hadavandi, F. Mehmanpazir, and M. M. Nakhostin, "Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction," Knowledge-Based Systems, vol. 35, pp. 245-258, 2012.https://doi.org/10.1016/j.knosys.2012.05.003.

[3] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," Applied Soft Computing, vol. 11, no. 2, pp. 2664-2675, 2011.https://doi.org/10.1016/j.asoc.2010.10.015.

[4] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang and S.-P. Guo, "Forecasting stock indices with back propagation neural network," Expert Systems with Applications, vol. 38, no. 11, pp. 14346-14355, 2011.

[5] J. W.-S. Hu, Y.-C. Hu and R. R.-W. Lin, "Applying neural networks to prices prediction of crude oil futures," Mathematical Problems in Engineering, vol. 2012, 2012.

[6] M. Madhiarasan and S. Deepa, "A novel criterion to select hidden neuron numbers in improved back propagation networks for wind speed forecasting," Applied intelligence, vol. 44, no. 4, pp. 878-893, 2016.https://doi.org/10.1007/s10489-015-0737-z.

[7] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in Advances in neural information processing systems, 1990, pp. 524-532.

[8] G. Huang, S. Song, and C. Wu, "Orthogonal least squares algorithm for training cascade neural networks," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 59, no. 11, pp. 2629-2637, 2012.https://doi.org/10.1109/TCSI.2012.2189060.

[9] J. Nadkarni and R. F. Neves, "Combining NeuroEvolution and Principal Component Analysis to trade in the financial markets," Expert Systems with Applications, vol. 103, pp. 184-195, 2018.https://doi.org/10.1016/j.eswa.2018.03.012.

[10] J. Wang and J. Wang, "Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks," Neurocomputing, vol. 156, pp. 68-78, 2015.https://doi.org/10.1016/j.neucom.2014.12.084.

[11] A. Tharwat, "Principal component analysis-a tutorial," International Journal of Applied Pattern Recognition, vol. 3, no. 3, pp. 197-240, 2016.https://doi.org/10.1504/IJAPR.2016.079733.

[12] R. Amalraj and M. Dharmalingam, "A work point count system coupled with back-propagation for solving double dummy bridge problem," Neurocomputing, vol. 168, pp. 160-178, 2015.https://doi.org/10.1016/j.neucom.2015.06.001.

[13] K. Velusamy and R. Amalraj, "Performance of the cascade correlation neural network for predicting the stock price," in Electrical, Computer and Communication Technologies (ICECCT), 2017 Second International Conference on, 2017, pp. 1-6: IEEE.

[14] T. R. Shultz and S. E. Fahlman, "Cascade-correlation," in Encyclopedia of machine learning: Springer, 2011, pp. 139-147.

[15] J. E. Jackson, A user's guide to principal components. John Wiley & Sons, 2005.