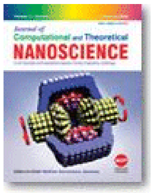

# Neural Network and Principle Component Analysis Based Numerical Data Analysis for Stock Market Prediction with Machine Learning Techniques

**Authors:** Islam, Mohammad Rabiul; Al-Shaikhli, Imad Fakhri; Nor, Rizal Mohd; Tumian, Afidalina

**Source:** Journal of Computational and Theoretical Nanoscience, Volume 16, Number 3, March 2019, pp. 806-812(7)
**Publisher:** American Scientific Publishers
**DOI:** https://doi.org/10.1166/jctn.2019.7958

Abstract | References | Citations | Supplementary Data | Article Media | Metrics | Suggestions

Financial market prediction is gaining attention throughout the market phenomena since various applicable techniques within soft-computational methods have been analyzed to define the optimization. The study of this experimental research focused on two benchmark numerical stock market dataset (S&P 500 index dataset and OHLCV dataset). This structural dataset is analyzed through two main applicable techniques such as Feed-forward Neural Network and Principle Component Analysis for stock market prediction where the remarkable Machine Learning technique hold a variant of features. The architectural neural network is rebuilt based on four layers with neurons that influence on high-dimensional dataset with the performance of popular ReUL activation function. Model specification also embodies the result of precision, recall and "F-score" within the number of twenty epochs. An overall picture of this developing model approaches the maximum level of accuracy which impacts on the academical research philosophy for financial market prediction.

**Keywords:** Neural Network; Numerical Data Analysis; OHLCV Data; Principle Component Analysis; S&P 500 Index; Stock Prediction

**Document Type:** Research Article

**Affiliations:** Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur, 50728, Malaysia

Publication date: March 1, 2019

More about this publication?

*Journal of*
*Computational and Theoretical Nanoscience*
Vol. 16, 1–7, 2019

# Neural Network and Principle Component Analysis Based Numerical Data Analysis for Stock Market Prediction with Machine Learning Techniques

Mohammad Rabiul Islam*, Imad Fakhri Al-Shaikhli, Rizal Mohd Nor, and Afidalina Tumian

*Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur, Malaysia*

Financial market prediction is gaining attention throughout the market phenomena since various applicable techniques within soft-computational methods have been analyzed to define the optimization. The study of this experimental research focused on two benchmark numerical stock market dataset (S&P 500 index dataset and OHLCV dataset). This structural dataset is analyzed through two main applicable techniques such as Feed-forward Neural Network and Principle Component Analysis for stock market prediction where the remarkable Machine Learning technique hold a variant of features. The architectural neural network is rebuilt based on four layers with neurons that influence on high-dimensional dataset with the performance of popular ReUL activation function. Model specification also embodies the result of precision, recall and "F-score" within the number of twenty epochs. An overall picture of this developing model approaches the maximum level of accuracy which impacts on the academical research philosophy for financial market prediction.

**Keywords:** Neural Network, Principle Component Analysis, Numerical Data Analysis, Stock Prediction.

RESEARCH ARTICLE

## 1. INTRODUCTION

Stock market prediction is logical and structural approach within soft computing methods that relies on the analysis of historical data, pattern recognition and data mining techniques which brings the feasible study to determine trading decision. Discovering the knowledge from numerical stock data which is the main goal of this research to define by remodelling for the clear view of additional structure. Different aspects of Neural Network (NN) identify the stock structural data which is configured to discover by pre-processing knowledge, reasoning issues and rebuilding model. The training algorithm focuses on developing computation model that builds up within dimensional array. The collected scarping server dataset as csv files in the array are setup with scaling value which brought the important consequences on this research impact. Sequentially, this paper is organized in accordance with the research design in section two, research methods in section three, NN architecture and mechanism in section four and finally there is a discussion of the result with analysis in section five.

## 2. RESEARCH DESIGN

Numerical data analysis focuses based on NN architecture which come out with the feasible value of stock market. Prediction of stock price relays on quantifiable factors that performed by various soft-computing methods [1]. Basically, numerical data analysis is much dependable on technical methods and factors for stock prediction as have been seen in the trading philosophies of fundamental and technical analysis [2]. Some combinational soft-computing methods are applied on developing platform which come as the consequences of resulting phenomena. The NN architectural diagram build as consequently that figure out the conceptual model of numerical data analysis. However, the analytical results depend on two types of stock data set that was depicted at the end of this experimental research which defined the target value.

## 3. STRUCTURAL DATASET

Structural stock market datasets are revealed from scraping server that figure out and hold relevant stock data about company. Gaining knowledge from market value of numerical data need some manipulation on the shaping of csv file. The subscribed dataset S&P 500 index holds $n = 41266$ minutes of the ranging data from April 2017 on

[*] Author to whom correspondence should be addressed.

1546-1955/2019/16/001/007 doi:10.1166/jctn.2019.7958

500 stocks with the total S&P 500 index price that arrange in wide range of format which is already LOCF'ed (Last observation carried forward) to confirm that dataset out of missing value. Unsubscribed based OHLCV data (Open, High, Low, Close price and Volume) from yahoo finance back from last year to 2018 that hold $n = 252$ minutes of the ranging data. Both structural data types are supportive in a same manner with the developing algorithm.

### 3.1. Dataset Initialization and Dimension

Array in the dataset always configure in two dimensions but this research focused on the index value of data.shape[0] and data.shape[1] is 0&1 that working along with the first dimension array in this dataset. After initializing, the dataset needs to support the arguments of the method in term of data values that can index the array along the given of first dimension as converted by using numpy.

### 3.2. Preparing Training and Testing Data

Collection of datasets need to train for classification by using neural network classifier. In this classifier test stock data and training stock data will be classified and so the dataset needs to be split into two sets as training and testing. Out of total, the training dataset hold between (70%–80%) as setting from the dataset which is not shuffled but sliced sequentially. This training data ranges from April to approximately end of July 2017 and the test dataset end of August 2017. Across the time series cross validation dataset also could be setting on a lot of different approaches such as rolling forecasts with or without refitting or more elaborate concepts such as time series bootstrap resampling. Repetition involves on samples from the reminder of the seasonal decomposition of the time series to simulate samples that follow the same seasonal pattern as the original time series but not exactly the copies of its values.

### 3.3. Data Scaling with Neural Network

Inspired by human brain Neural Network method analyzed and figured out by four layers with feed-forward which consists the input nodes, output nodes and hidden nodes. Initially, these nodes are connected with random weights. A gradient descent algorithm is used to adjust weights during training, so that outputs nodes correctly classify data presented to the input nodes. In this case, Number of architectural neural network (NN) used to get benefit from scaling dataset as input (sometime also output). Because the most common activation function of the network's neurons like tanh or sigmoid are on the $[-1, 1]$ or $[0, 1]$ interval respectively setting on it. Recently, research in data mining use the popular rectified linear unit (ReLU) activations function which are commonly used to unbounded on the axis of possible activation values. However, this project uses the both scaling value as inputs and targets anyways can be easily accomplished in Python by using sklearn MinMaxScaler.

### 3.4. Preparing the Sequence of Setting in Scaleing Value

Data need to be scaled and some caution must be undertaken regarding what part of data and for the time being. Since, different features exit in different dataset so it's necessary to define feature vectors values based on dataset that need to be fining to test data and training data within array. In this case, the usable mistake done through scaling with the whole dataset before training and testing is applied to split. Some mistake commonly happened in this stage, because the invokes of the calculation of the statistics e.g., the min/max of a variables are done through with that mistake. During the timeseries forecasting in real life, much information from future observation might be insufficient. Therefore, calculation of scaling statistics must be conducted on training data and applied on the test data. Other than that future information is not acceptable which is commonly biases in terms of forecasting in a positive direction.

### 3.5. Scaling Value in Principle Component Analysis (PCA)

Linear projection of high dimensional data defined by principle component analysis (PCA) for variance retained that able to maximized and minimized the least square reconstruction error [3]. So, it's important to enhance PCA for dimensionality reduction and scale the features to the dataset before applying to PCA since it's affected. In this case, the Standard Scaler help to standardize the dataset's features onto unite scale ($\text{mean} = 0$ and $\text{variance} = 1$) which is a requirement for the optional performance of many machine learning algorithms. Scikit-learn in python, has a section on the effects of not standardizing the data that help to avoid negative effect of data scaling.

### 3.6. Dimensionality Reduction to Data Scaling

The research of this work focused on the hybrid combination of neural network and dimension reduction method of PCA. New set of variables based on liner combination of original values as describe principle components. By enhance PCA for dimensionality reduction, it had been found that first principle component accounts for most of the possible variation of original data after each succeeding component has the highest possible variance. The principal axes in the feature space, representing the directions of maximum variance in dataset. Table I as stated in line-1, the component is stored by explained variance. In line-2, Then the amount of variance explained by each of the selected components that equal to $n$_components largest eigenvalues of the covariance matrix of $X$. (New version 0.18). Percentage of variance explained by each of the selected components. If $n$_components is not set,

Table I. Data scaling components.

| | |
|---|---|
| Line-1 | Components_: Array, shape ($n$_components, $n$_features) |
| Line-2 | Explained_variance_: Array, shape ($n$_components), |
| Line-3 | Explained_variance_ratio_: Array, shape ($n$_components), |
| Line-4 | Singular_values_: Array, shape ($n$_components), |
| Line-5 | Mean_: Array, shape ($n$_features,) equal to X.mean(axis = 0) |

then all components are stored and the sum of explained variance is equal to 1.0 as stated in line-3. In line-4, the singular values corresponding to each of the selected components. The singular values are equal to the 2-norms of the $n$_components variable in the lower dimensional space. Stated in line-5, the per-feature empirical mean estimated from the training set that equal to $X$.mean(axis $= 0$).

## 4. REVIEW OF PREDICTION TECHNIQUES

Number of prediction techniques are available that belongs to fundamental and technical approaches which based on two trading philosophies that encapsulated by financial research [4]. Artificial Intelligent techniques used for technical analysis to attempt the prediction of financial market. Specially Linear Regression (LR), Genetic Algorithms (GaS), Case based Resoning (CBR), Support Vector Machines (SVMs) and Artificial Neural Network (ANN) are the most influential techniques developing prediction platform [5]. The philosophies of sentimental analysis also determine the behavioural and psychological elements of market value [3]. The analytical quantitative financial data in this research analyse through neural network model as mentioned that brought out with structural architecture.

### 4.1. Neural Network Architecture

Computational ANN model functionally processing the information like human brain as usable on Takagi Sugeno-Kang systems and trained to the model for some collection of output/input data. Multi-layer NN can solve both liner and non-linear classification problem [6]. The rebuild NN model holds the number of connected processing units to process the numerical stock data. As shown in the Figure 1, having four neural layers consisted by this NN architecture. The first is input layer, second dataset layer, third layer is hidden and last one is output layer.

In this developing project, the ANN model is built up with the core component of TesnorFlow in python 3.6.6 environment. Important part of neural network (NN) is the biases number that NN learns by generalizing the existing problem. To overcome such problem, some procedures need to follow as NN formulated in terms of feeding data and received in output layer. First the data come to single layer box for multiplies the weights with data and then adds a bias to the multiplied data as stated in "Eq. (1)"

$$\text{Neuron}_1 = a_1^1 W_1 + a_1^2 W_{1,2}^2 + \cdots + a_n^n W_{n,n}^n \tag{1}$$

This type of execution function template applied for a straight line on single unite as depict on Figure 1. More than two layers could promote the non-linearity within this NN architecture. Computational multiple outputs as expected from this architecture of input data as the consequences from the data of previous layer. There are number of NN generalize the problem by learning through multiplying the weights with the data in optimal biases but the value of Hyper-Parameters manually setting the nobs which change the behaviors of the machine as applied in this Neural Network architecture.

### 4.2. Neural Network Training Process

To perform the specific task the program can define all the rules as required for given input to compute the result of output. The developed framework of NN able to learn to perform the specific task by training itself on a dataset through adjusting the result which computes to be as close to the actual results which have been observed. This process is known as training model which usually perform as described in following manner. When Neural Network model consists OHLCV data as input and to get the expected output then the value of closing price of the next day set on to the model and then the actual value of the output will be represented in $X$ and $Y$ axis in the graph. The training of this model involves adjusting the weights of the variables for all the different neurons present in the neural network.

### 4.3. Neuron Distribution

As seen from the Figure 1 NN model consists of four layers. First is input layer, second is dataset layer, third is hidden layer and fourth is output layer. The first layer contains 1024 neurons, slightly more than double the size of the next inputs. Other subsequent hidden layers are always half the size of the previous layer, which means 512, 256 and finally 128 neurons. A reduction of the number of neurons for each subsequent layer compresses the information of network that identifies in the previous layers. Weight factors for neural network is selected using multi objective optimization algorithm which improves the classification scenario. Number of supportive dataset model in this NN architecture well performed for the outlet configuration by setting up neurons.

### 4.4. Architectural Modle Configuration

In order to fit the model, two placeholders are required where $X$ contains the network's inputs and $Y$ is the network's outputs. (Network's inputs constituents of all S&P 500 at time $T = t$ and output S&P 500 at time $T = t + 1$). It is crucial to understand which input and output dimensions the neural network needs in order to design properly. So, the shape of the placeholders corresponds to [None, $n$_stocks] with [None] which meaning that the inputs are a 2-dimensional matrix and outputs are a 1-dimensional vector as figured in python. The values of the network are

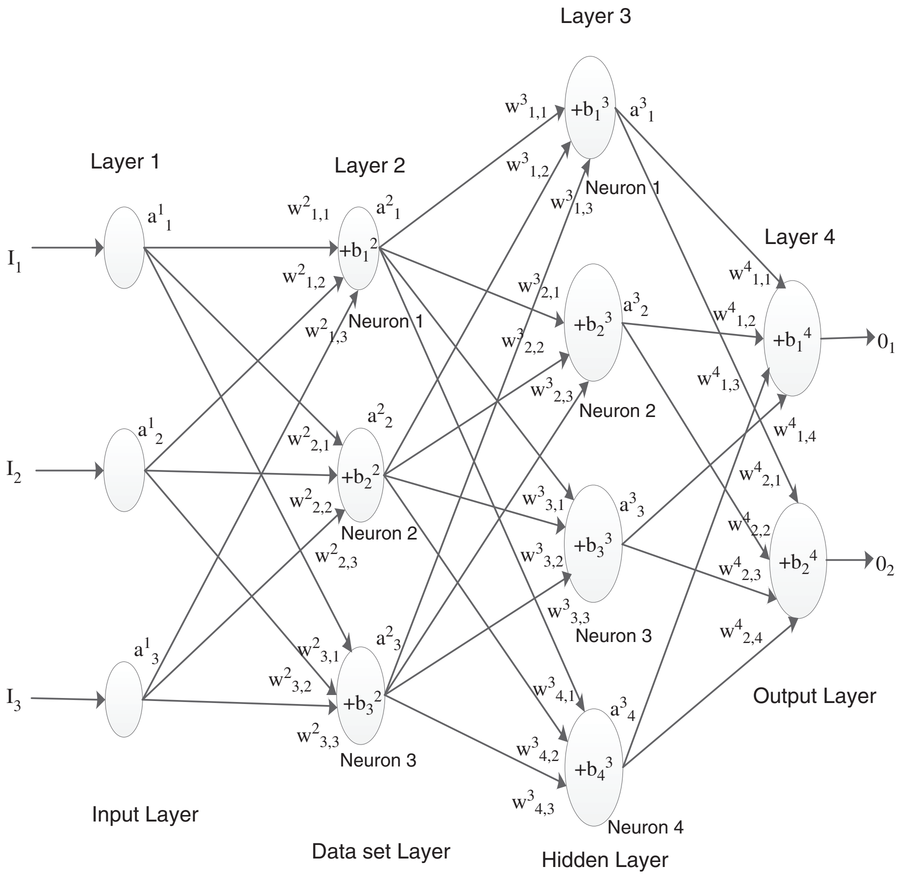

**Fig. 1.** The flow of architectural diagram of neural network.

important due to finds all the weights and bias points of data feeding to the output layer.

### 4.5. Activation Function for Model Specification

It's important to specify the required variables dimensions between input, hidden and output layers. Since the rule of thumb in multilayer perceptrons (MLPs) as used here, where the second dimension of the previous layer is the first dimension in the current layer for weight matrices. Though sound complicated but is essentially just each layer passing its output as input to the next layer. The biases dimension equals the second dimension of the current layer of weight matrix, which corresponds the number of neurons in this layer. The activation function provides different types of nonlinearities that usable in neural network. These type of smooth nonlinearities like ReLU, ReLU6, CReLU and ReLU_x where the regularization (Sometime dropout). ReLU is used as hidden layer with tensorflow as tf in this function.

After definition of the required weight and bias variables, the network topology, the architecture of the network needs to be specified. Hereby, placeholders (data) and variables (weights and biases) need to be combined into a system of sequential matrix multiplications. Furthermore, the hidden layers of the network are transformed by activation functions where the activation functions are important elements of the network architecture since it was introduced the non-linearity to the system. Number of possible activation functions available in NN and one of the most common activation functions rectified linear unit (ReLU) used in this model.

$$\text{ReLU: } \sigma(X) = \max(0, X) \tag{2}$$

This activation function ReLU is simplest non-linear that takes some input on the $x$-axis which is greater than zero and before rectifying its value is always zero as depict on Figure 2. ReLU can drastically improve on NN as research found that it's very popular in recent years due to its activation function as stated on "Eq. (2)" [8]. This activation function on this equation threshold at zero stated on above Figure 2.

### 4.6. Cost Function

A cost function is a single value that rates how good neural network perform overall. The network of the cost function used to generate a measure of deviation between the network's predictions and the actual observation of training targets. It also depends on variable such as weights and biases. For regression problems, the mean squared error (MSE) function is commonly used which is basically MSE

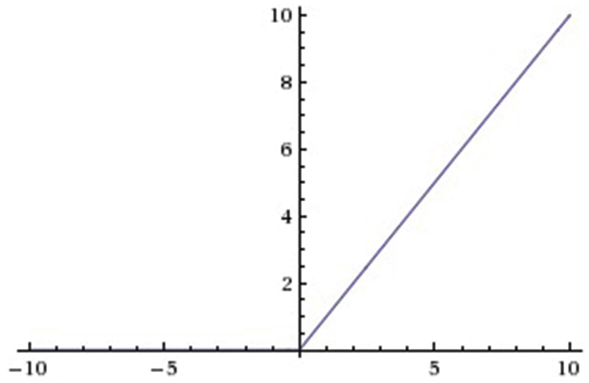

**Fig. 2.** Reul activation function.

computes the average squared deviation between predictions and targets. Any differentiable function can be implemented in order to compute a deviation measure between predictions and targets. However, the MSE and PSNR exhibits certain properties that are advantageous for the general optimization problem to be solved.

### 4.7. Precision and Recall

Precision marked the number of samples of the classifier that correctly identify as true positive out of all positive. Recall marked the number of samples in the classifier that correctly identify out of total samples in the set. Accuracy is the common measure of classification performance or in a word its known as complement error rate. Generally higher accuracy rate depends on better classification. The absolute value needs to find from generated given number of epoch in the system and after that the accuracy could be calculated based on given formula. Precision and recall are widely used in information retrieval (IR) field while metrics for evaluating the performance as shown below in the Table II. The ratio of TP is denoted of samples labeled as belonging to a class that basically belongs to class. In contrast, FP is denoted as the ratio of samples labeled as belonging to a class does not belong to this class. TN denoted the ratio of samples not labeled as belonging to a class which indeed belonging to this class. FN denoted to the ratio of samples not labeled as belonging to a class does not belong to this class. Low and high price define the relation between TN, TP, FN and FP as describe here.

TN: The number of truth negatives, when market price is low as the result found from experiment.

TP: The number of truth positives when market price is high as the result found from experiment.

FN: The number of false negatives. In this case, the market price is high among those result that given prediction of market price having low.

FP: The number of false positive. In this case, the market price is low among those result that given prediction of market price having high.

From the above Table II, define the bottom row that hold the high price with TP as seen from Figure 3 known as recall. It's clear that recall able to give the classification performance with respect to the false negatives. While precision able to generate the performance with respect to false positives.

Precision and recall can be seen as extended versions of accuracy in this experiment by using a combination of these measures for the classifiers dissipates in "Eq. (3)." Precision can be seen as exactness and recall is the measure of completeness.

**Table II.** Confusion matrix.

| | Low price | High price |
|---|---|---|
| Low price | TN | FP |
| High price | FN | TP |

### 4.8. F-Measure

F-Measure is one of the scores that give the balance between recall and precision. Beside the score of precision and recall the F-Measures usable for the use of score which aspects of the classifiers performance and optimize in the system to get the result. Assuming that precision is **P** and recall **R** then formulative way to find the F-Measurement that is F1 score can be interpreted as a weighted average of the precision and recall where an F1 score reaches its best value at 1 and worst score at 0.

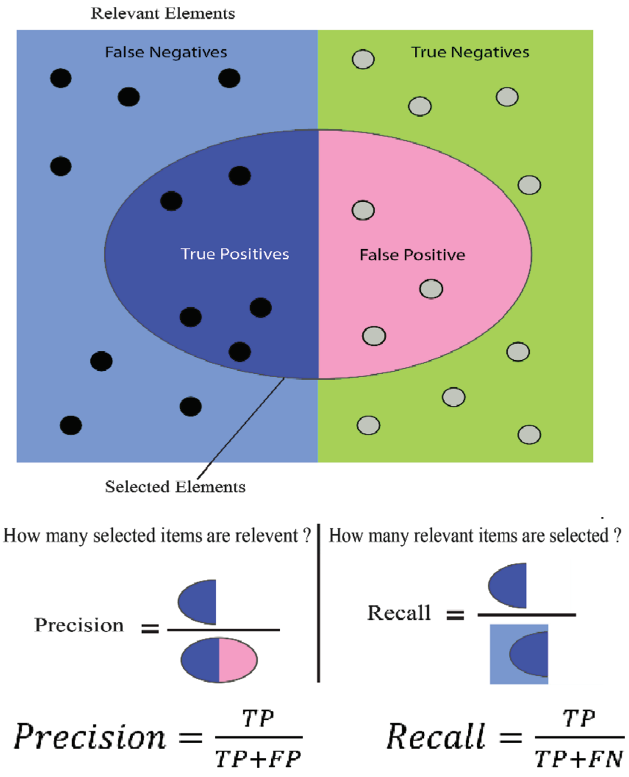
Relevant Elements
False Negatives
True Negatives
True Positives
False Positive
Selected Elements
How many selected items are relevent ?
How many relevant items are selected ?
Precision =
Recall =
$Precision = \frac{TP}{TP+FP}$
$Recall = \frac{TP}{TP+FN}$

**Fig. 3.** Conceptualization model of precision and recall.

The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

$$\text{F-Measures} = \frac{(2 * P * R)}{(P + R)} \tag{3}$$

### 4.9. Experimental Test and Result

Proposed techniques applicable on number of different platforms as python or Jupiter is applied on this experiment from the formulative expected outcome.

$$\text{Accuracy} = \frac{\{(\text{TP}+\text{TN}) * (\text{TP}+\text{TN}+\text{TF}+\text{FN})\}}{25} \tag{4}$$

The value 25 is configured in this "Eq. (4)" as threshold value as found the total number of elements since the accuracy formula is executed through Python 3.6.6 and above version. Here the threshold value considers the minimum value to activate the prediction accuracy. Prediction decision obviously depended on given dataset that will calculate as input data and finding the prediction data. In between prediction and given data calculation found of percentage data through training data. Basically, the accuracy of calculation calculated between prediction data and input value of data. Up to the knowledge of the research this developing algorithm reached the level of accuracy 80% which is new among the calculation from threshold value and number of ecpochs.

As found from the Table III that within three classes how much percentages of frequency level come out from the multi objective algorithm for optimization. The number of epochs found the level of different number of accuracies, MSE Train, MSE Test, PSNR Train and PSNR Test. Based on these results the total average of Precision, Recall and F1-Score value is dependable. The corresponding sores of every class define the accuracy of the classifier in classifying the data points in the particular class compared to all other classes. Class zero defines training precision that one is different for training precision, recall and F1 score. The support is the number of samples of the true response that lie in that class.

### 4.10. Pictorial Result Analysis

The executable total number of epochs is 20 and among those epochs selected epochs will be detecting for high prediction values that showing in plot by using matplotlib. So $x$-axes is the stock data and $y$-axes is prediction data. The epoch range and batch will predict from selecting epoch with optimization techniques for better prediction accuracy with ANN.

**Table III.** Score components.

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| 0 | 0.50 | 1.00 | 0.67 | 1 |
| 1 | 1.00 | 0.50 | 0.67 | 2 |
| 2 | 1.00 | 1.00 | 1.00 | 2 |
| Average/total | 0.90 | 0.80 | 0.80 | 5 |

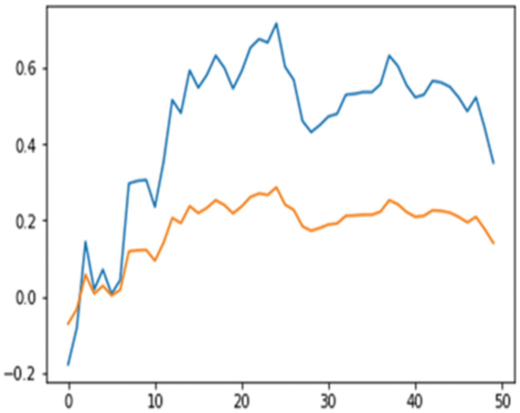

**Fig. 4.** OHLCV dataset depict the prediction along with dataset layer.

The above two pictorial figures represent two types of datasets result. The accuracy of prediction data showing in visualization of graph and mean square error (MSE) of prediction stock data that presented in interpreter mode. The Figure 4 represent based on opening, high, low and closing price with volume and Figure 5 represent the result of S&P 500 index dataset and both are given the same accuracy level. The significant correlation between the feeding of original stock dataset (Generating Blue Line) and generating prediction (orange line) produce the 80% accuracy level from the movements of prediction. The applicable accuracy is formulated as stated below.

$$\text{Accuracy} = \frac{(\text{Number of correct classified elements})}{(\text{Total Elements})}$$

Based on total number of classifications, level of accuracy can lead of the S&P 500 index, meaning, its value is shifted 1 minute into the future as already been done in the dataset. But the prediction operation is required to predict next minute of the index instead of current minute.

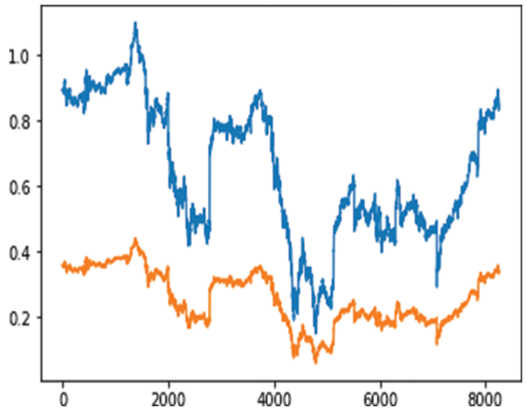

**Fig. 5.** S&P 500 index dataset depict the prediction along with dataset layer.

As seen, current dataset Figure 5 contains the price of the S&P 500 at $(t+1)$ and constituent's price at $(T=t)$.

## 5. CONCLUSION

This experimental research is carried out to analyze the combinational performance of PCA and ANN. It also figures out the correlation between the dataset and prediction result. However, the current result defines the accuracy of the previous result the same but the only difference is with ReUL activation functions in feed-forward NN architecture which makes changes with the distribution of scaling value among the dataset layer with PCA. The result of this achievement influences on academic purpose but not on investment as stated in the previous research [7]. However, the maximum result of accuracy level helps to identify the highest possible value that brings sense of investment in stock market by further research.

**Acknowledgment:** This research work was partially supported by International Islamic University Malaysia, FRGS14-127-0368 and ERGS13-018-0051 from Ministry of Higher Education of Malaysia.

## References

1. Islam, M.R., **2018**. Technical approach in text mining for stock market prediction: A systematic review. *Indonesian Journal of Electrical Engineering and Computer Science, 10*, pp.770–777.
2. Islam, M.R., Al-Shaikhli, I.F. and Abdulkadir, A.A., **2018**. A scientific review of soft-computing techniques and methods for stock market prediction. *Int. J. Eng. Technol., 7*(2.5), pp.27–31.
3. Vinodhini, G. and Chandrasekaran, R., **2014**. Sentiment Classification Using Principal Component Analysis Based Neural Network Model. *2014 International Conference on Information Communication and Embedded Systems (ICICES)*, pp.1–6.
4. Patel, H.R., Suthar, A.B. and Parikh, S.M., **2014**. A proposed prediction model for forecasting the financial market value according to diversity in factor. *International Journal of Computer Technology and Applications, 5*, p.131.
5. Huang, W., Nakamori, Y. and Wang, S.-Y., **2005**. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research, 32*, pp.2513–2522.
6. Uysal, A.K. and Gunal, S., **2012**. A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems, 36*, pp.226–235.
7. Saini, M. and Singh, A., **2014**. Forecasting stock exchange market and weather using soft computing. *International Journal of Advanced Research in Computer Science and Software Engineering, 4*.
8. Krizhevsky, A., Sutskever, I. and Hinton, G.E., **2012**. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pp.1097–1105.

Received: 30 September 2018. Accepted: 12 December 2018.