STOCK MARKET PRICE PREDICTION

CS725: Foundations of Machine Learning

SUBMITTED BY:
ANAND NAMDEV(163050068)
PARTH LATHIYA(163050095)
AWISHA MAKWANA(163050079)

1) Introduction :

Time series data prediction is one of the broad topics in the field of machine learning. The data in the time series is dependent only on time. Stock market prediction is regarded as a challenging task of the financial time series prediction process since the stock market is highly, nonlinear and nonparametric.

In addition, stock market is affected by many macro economical factors such as political events, general economic conditions, investor's expectations, movement of other stock markets and psychology of investors etc.

A various set of different machine learning techniques and statistical techniques have been studied, out of which three different models have been implemented to predict stock market direction. Artificial Neural Networks(ANN), Support Vector Classification(SVC) and Auto Regressive Integrated Moving Average(ARIMA) are the methods studied to achieve the task. Also, a version of ARIMA model is also implemented which has minor variations with respect to ARIMA model. ARIMA and ARMA models are statistical approaches while ANN and SVC are machine learning based methods. The values of parameters according to corresponding model is varied and performance is measured to tune the hyper-parameters.

2) Objective :

The main objective of the project is to predict the direction of stock market. The direction of stock market is defined to be upwards if the closing value of index increase from the previous day and downwards if the closing value of current day is smaller than the previous day.

Various machine learning and statistical methods have been studied and ANN, SVC & ARIMA models have been implemented to achieve this task.

3) Motivation :

The main motivation of doing this project i.e. stock prediction is the association of this problem with one of the most challenging domain that is Time-Series Analysis. In a time-series domain there are no features associated with a output values (here closing value). All values are either varied with respect to the time or in a way dependent on factors which can't be converted into a continuous vector space. For ex: for stock prediction one cannot measure the value of various factors behind the ups and down of the closing price like political shaking, investor's interest in buying the stock etc but one can definitely examine these patterns over a period of time and predict for the future values if a stock values has increased or decreased.

4) Approaches :

Stock prediction task will be achieved here by two methods.
a) Statistical methods
b) Machine learning method

a) Statistical Methods :

These are the mathematical method which define the pattern in the time series with the help of mathematical functions like moving average, exponential moving average etc. One of the most popular ARIMA and its variation ARMA models have been implemented.

b) Machine learning method :

These are the sophisticated supervised machine learning methods. Two most popular methods have been implemented.
a) Artificial Neural Networks
b) Support Vector Classification

5) ARIMA Model :

ARIMA stands for Autoregressive Integrated Moving Average model. It is based on a regression model where the y-closing values of the stock price is matched by drawing a line consisting of previous value and errors.

In ARIMA p defines the previous p-days closing values and q defines the previous q-day's errors values.

$$Y_t = \mu + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q}$$

$Y(t)$: predicted value

ϕ : previous p-days data coefficients

θ : previous q-days error coefficients

5.1) Data Description:

For the prediction of the stock price, we have used the Nikkei-225 which is the stock of Japan. It consists of 8000 rows of closing values in the range of 9k-17k varying with respect of time.

The below figure is the original data.

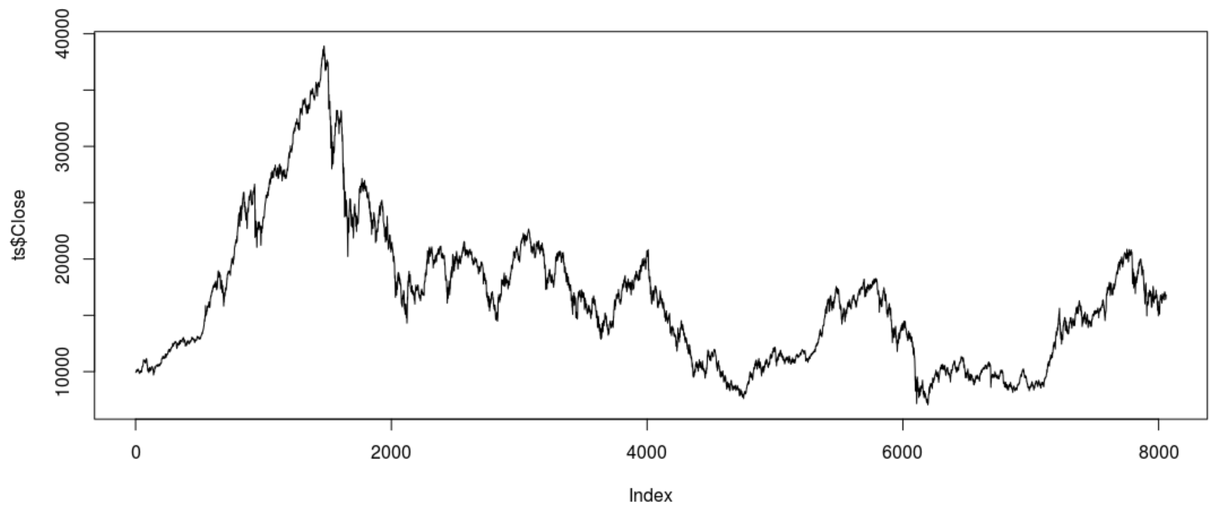

Figure 1 :Original Data

The above graph of the data shows that the data is behaving completely abnormally just like the stock price. That is there is no dependency on any feature except time.

In order to model such a time series, we first have to convert the data into a stationary data. Above series is a highly non-stationary data because it contains non-zero mean and non-zero variance. For a stationary time series the value of mean and variance should be as close to zero as possible.

The data and its various other components have been introduced to study the effect of accuracy on the data stationarity.

a) Original Data
b) Log data
c) Moving average lop difference
d) Exponential average log difference
e) Decomposed data

To convert the above time series into a stationary data, various changes have been done to the original data like logarithm of the data, moving average of the data, exponential average of the data, data decomposing. And for each of the data dickey-fuller test has been applied which returns the p-value. The p-value should be as close to zero as possible for a stationary data.

Below is the figure which shows the mean and variance of the all the types of data setting. From the table it is clearly visible that for decomposed data the mean and variance is closest to zero. It implies that the data for this setting is most stationary. On the other hand the original data and the log data has very high mean and variance which is a property of the non-stationarity.

SNo.	TYPE	MEAN	VARIANCE
1	Original	1.649584	38618370
2	Logarithm	9.645	0.1292
3	Moving Average	0.0002911	0.0005400021
4	Exponentiating	0.001053	0.0015610
5	Decomposing	$-1.55e^{-06}$	0.0002067

Table 1: Mean and Variance for each Dataset

The following graphs shows the plots of above data for ex: log data, moving average data, exponential data, decomposed data.

This is the moving average data. This data is more centered to the zero and has fairly low mean and variance value.

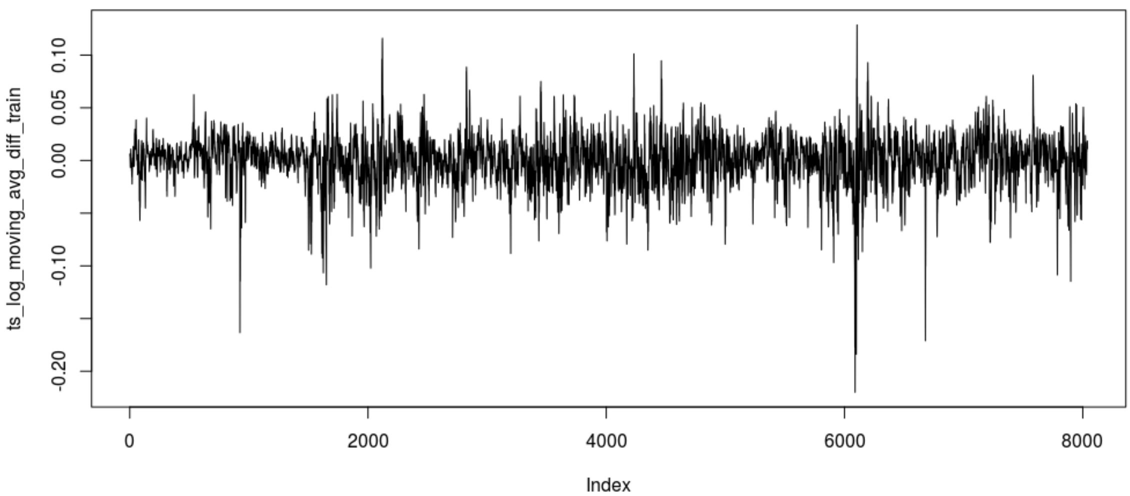

Figure 2: Moving Average Dataset

The below is the decomposed data.

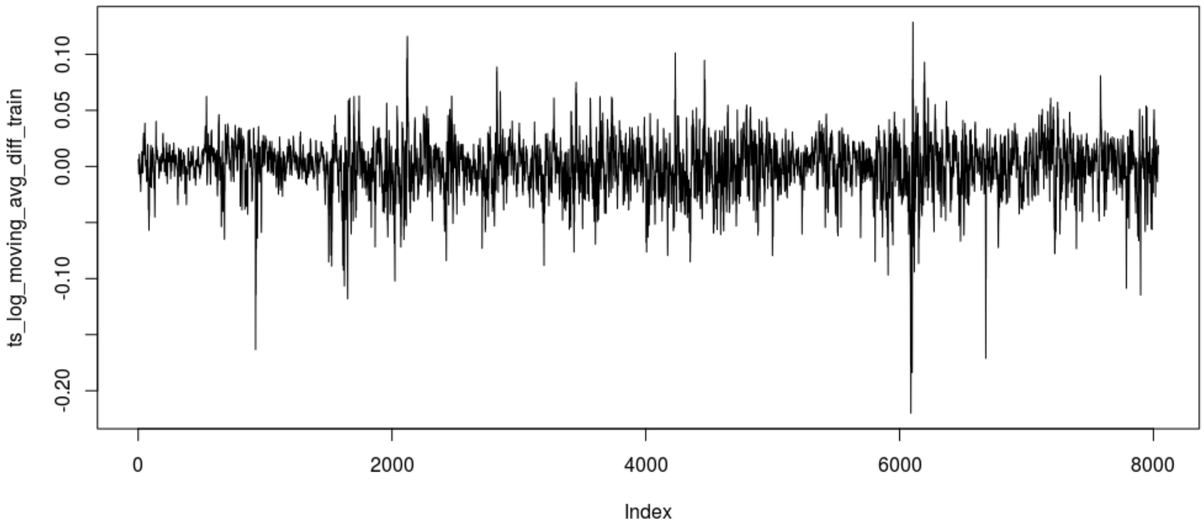

Figure 3 :Decomposed DataSet

5.2) Training of ARIMA :

For the training part of the ARIMA model, various p,i and q values have been optimized by running the experiment extensively. And for each value of p, i and q the error value AIC and BIC is calculated. The value of p,q and i for which this error is minimum will be selected as the final p,q and i value. The similar work has been done for ARMA except that the value of i is kept constant to be = 0.

This is the prediction error on the training data. The training data used here is the original data. The red is the original data and the blue is the predicting data.

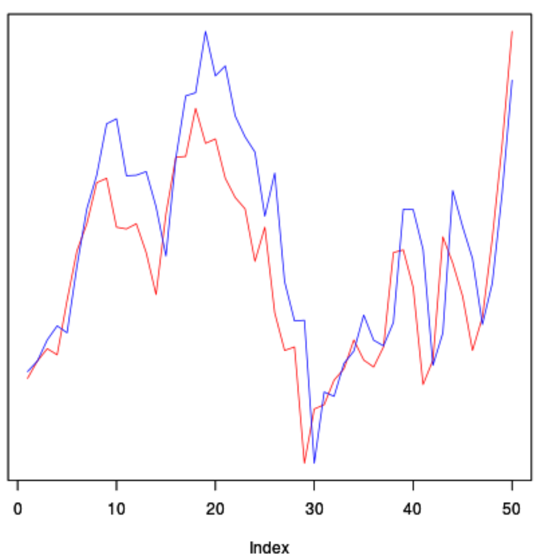

Figure 4 : Training Accuracy for original data

The below is the training prediction on training data based on the model from decomposed data.

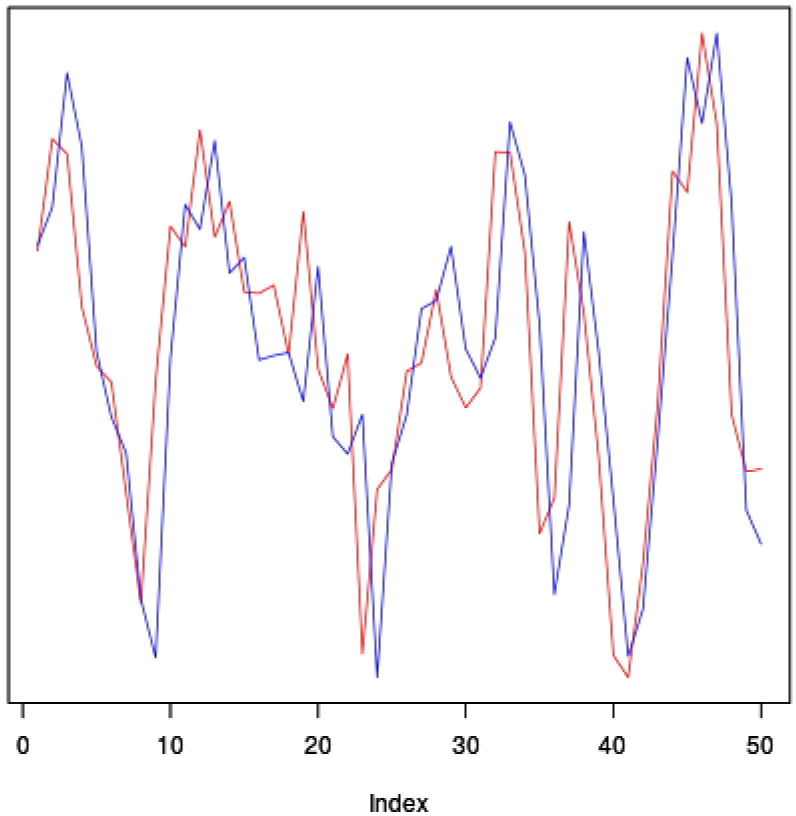

Figure 5 : Training Accuracy for decomposed data

5.3) Test Prediction :

For predicting the values of the stock, next 10 values are predicted.
The below data contains the red value which is the predicting value of stock for next 10 days according to the original data model. Here the data consists of upper and lower bound as well which account for 80% and 95% confidence value.

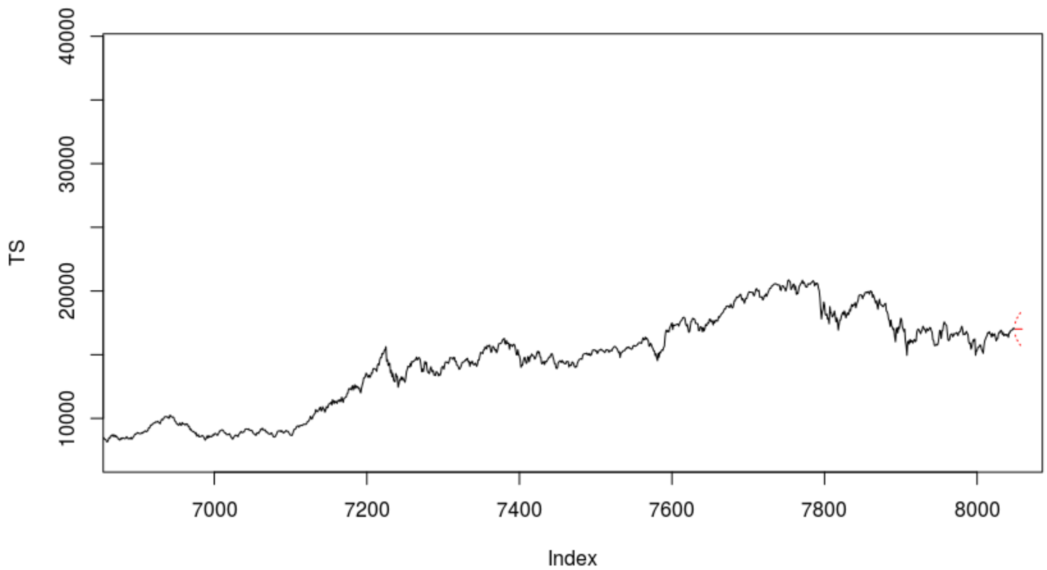

Figure 6 : 10-Days prediction according to Original Data

This is according to the moving average data.

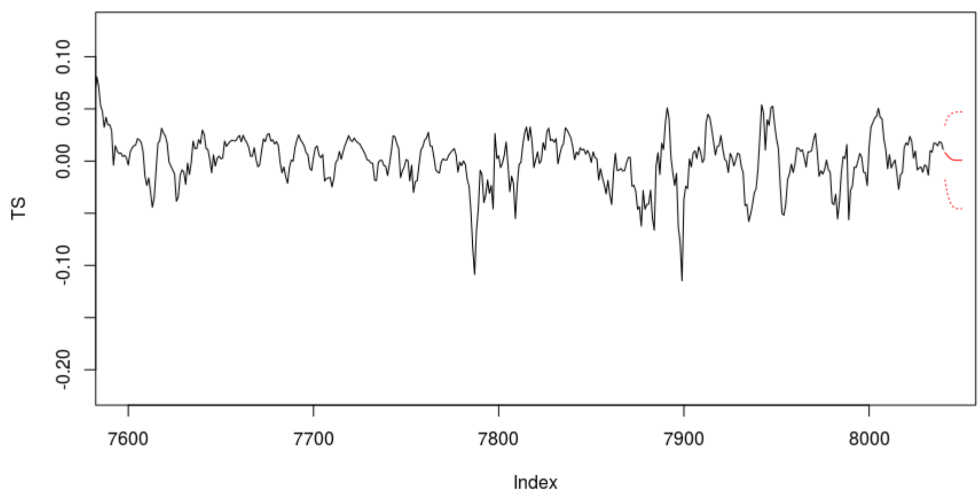

Figure 7 : 10-Days prediction according to Moving Average Data

This is according to the final decomposed data.

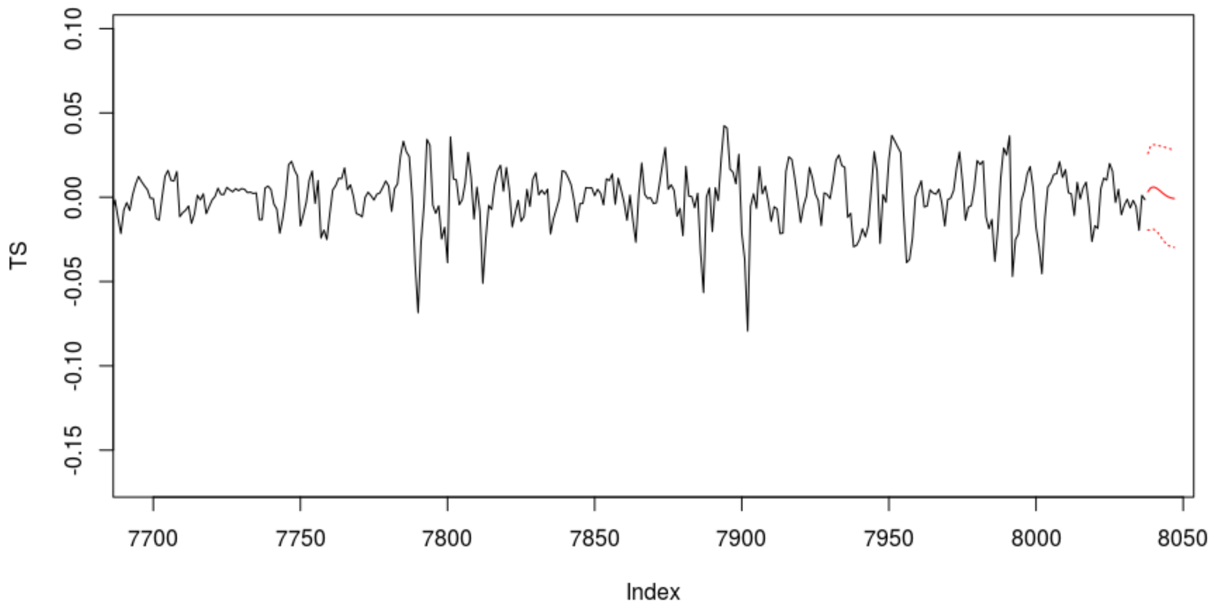

Figure 8: 10-Days prediction according to final decomposed data

While predicting for each type of data model, an error measure called MASE which stands for Mean Absolute Scaled error which is an important error function when different data series are scaled differently. The error here scales down to the common scaling point.

$$q_j = \frac{e_j}{\frac{1}{T-m}\sum_{t=2}^{T} |y_t - y_{t-m}|}$$

Once the error value for each type of data model is calculated, the similar type of work is done for ARMA model where I=0. Here as you can observe the error is least for decomposed data for both ARIMA and ARMA and highest for original data for both ARIMA and ARMA model. This justifies the saying that if a data is non-stationary it can't have very good error.

S. No	ARIMA (I!=0)	ARMA(i=0)
1	0.99968	1.0073
2	0.99987	1.00263
3	0.94	0.9418
4	0.9843	0.9852
5	0.8193	0.8107

Table 2 : Error Comparison on ARIMA & ARMA

The graph between error and the type of the data used for both ARIMA and ARMA.

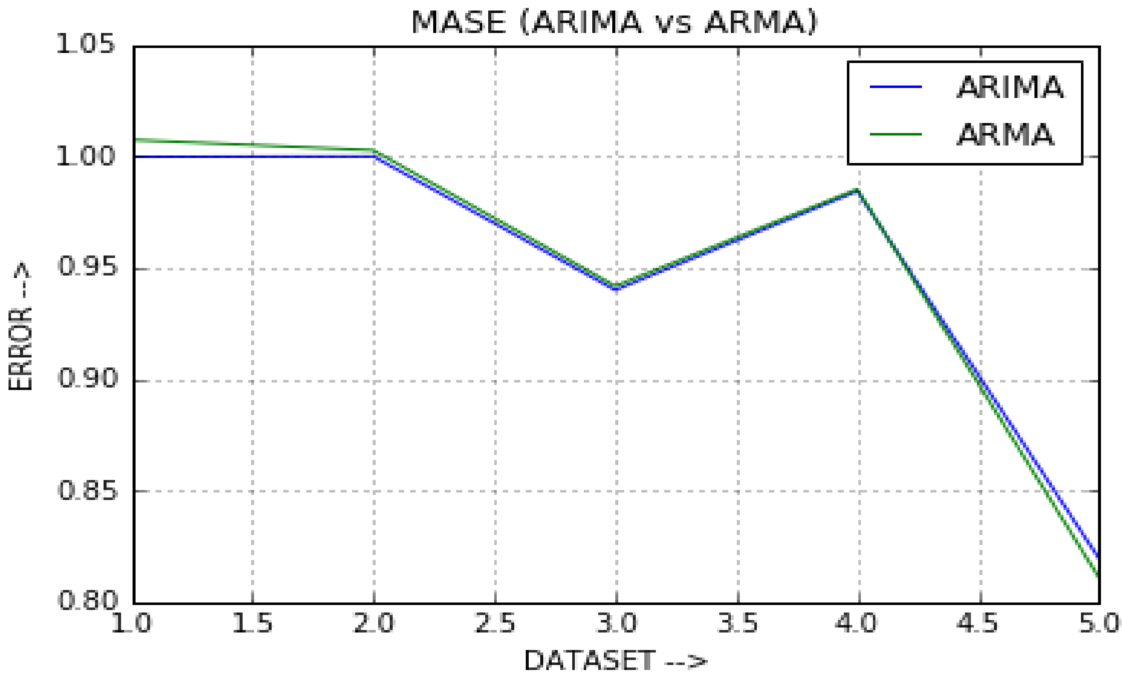

Figure 9 : Graph Error vs DataType (for ARIMA & ARMA)

6) ANN learning model :

ANN has demonstrated their capability in financial modeling and prediction. In this project , a two layered feedforward ANN model has been structured to predict stock price index movement. This ANN model consists of an input layer, 2 hidden layers and an output layer, each of which is connected to the other. In our model there are 10 inputs each of which have been calculated from the attribute values in the dataset used. The calculation involves using 10 functions specified in table 3 .These 10 inputs are the 10 neurons in the input layer. The output layer has 2 neurons for the two class outputs as two patterns(0 or 1) of stock price direction. The architecture of the two-layered feedforward ANN is illustrated in Fig 10.
The number of neurons in the hidden layer has been determined empirically. In an ANN model the neurons of a layer are linked to the neurons of the neighboring layers with connectivity coefficients (weights). These weights are updated to classify the given input patterns correctly for a given set of input-output pairs using a learning procedure. Initially the weights are assigned random values. The back-propagation learning algorithm is used to train the two layered feedforward ANN structure in this project.

To evaluate the performance of the ANN model absolute error is used. The gradient-descent method is used as the weight update algorithm to minimize the absolute error. A sigmoid function is selected on the hidden layer as the activation function. On the other hand, a softmax function is used on the output layer. That is, the outputs of the model will vary between 0 and 1.

If the output values are probabilities of that input belonging to that class(decreasing or increasing direction).

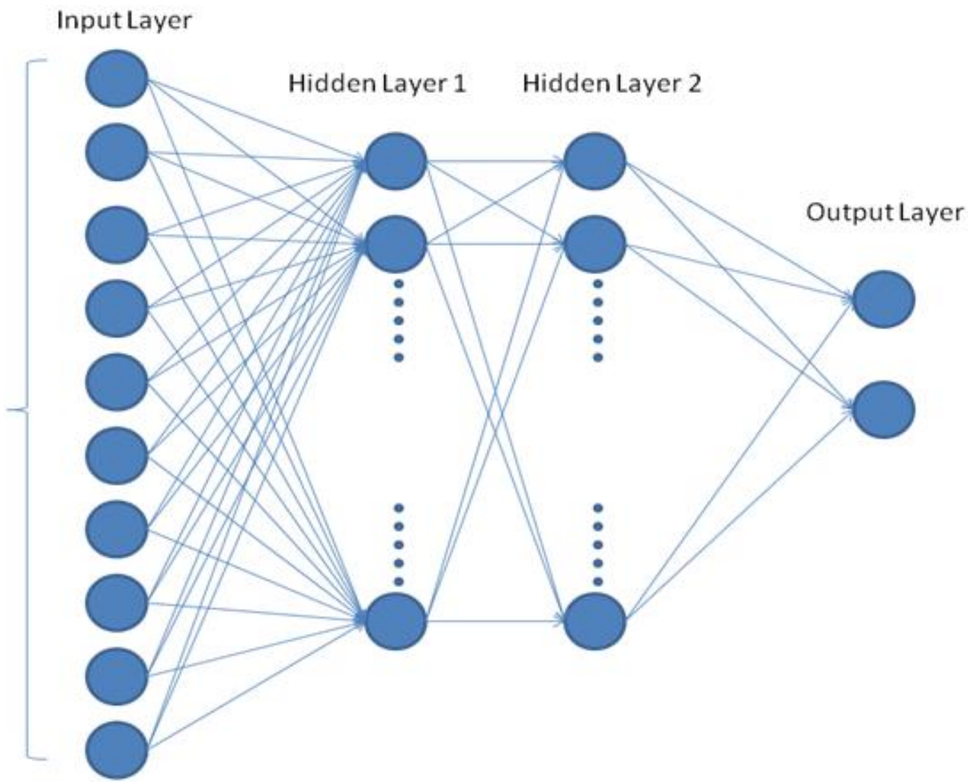

Figure 10 :ANN Model Architecture

Data Preprocessing:
Data used has 4 attributes which have been processed using 10 formulations as follows:

S No.	NAME	FORMULA
1	Simple 10-day moving average	$\frac{C_t + C_{t-1} \cdots + C_{t-10}}{10}$
2	Weighted 10-day moving average	$\frac{((n) \times C_t + (n-1)C_{t-1} \cdots + C_{10})}{(n-(n-1) \cdots +1)}$
3	Momentum	$C_t - C_{t-n}$
4	Stochastic K%	$\frac{(C_t - LL_{t-n})}{(HH_{t-n} - LL_{t-n})} \times 100$

5	Stochastic D%	$\sum_{i=0}^{n-1} K_{t-i}\%$
6	RSI (Relative Strength Index)	$100-\frac{100}{1+(\sum_{i=0}^{n-1}Up_{t-i}/n)/(\sum Dw_{t-i}/n)}$
7	MACD (moving average convergence divergence)	$MACD(n)_{t-1}+2/n+1\times(DIFF_t-MACD(n)_{t-1})$
8	Larry William's R%	$\frac{H_n-C_t}{H_n-L_n}\times 100$
9	A/D (Accumulation/Distribution) Oscillator	$\frac{H_t-C_{t-1}}{H_t-L_t}$
10	CCI (Commodity Channel Index)	$\frac{M_t-SM_t}{0.015D_t}$

Table 3: Technical Indicators and there formula

C_t is the closing price, L_t the low price, H_t the high price at time t, DIFF: $EMA(12)_t$ $EMA(26)_t$, EMA exponential moving average, EMA(k)t: EMA(k)t1 + a(Ct EMA(k)t1), a smoothing factor: 2/1 + k, k is time period of k day exponential moving average, LLt and HHt mean lowest low and highest high in the last t days, respectively, Mt : Ht þ Lt þ Ct=3; Upt means the upward price change, Dwt means the downward price change at time t.

HyperParameter Tuning:

The number of neurons(n) in the hidden layer, value of learning rate(lr) and number of iterations(epochs) are ANN model hyper parameters that must be efficiently determined.

Eight levels of n, 10 levels of learning rate and ten levels of epochs were tested in the hyper parameter tuning. The ANN parameters and their levels are summarized in table 4.

Each parameter combination was applied to the training and holdout data sets and prediction accuracy of the models were evaluated seeing the absolute errors.The parameter combination that resulted in the best performance is selected as the best one for the corresponding model.

Parameter	**Levels**
Number of neurons in hidden layer	10,15,20…..,40
Value of learning rate	0.1,0.2,……,0.9
Number of iterations(epochs)	10,20,…..,100

Table 4:ANN parameter levels tested in hyperparameter tuning

Some observations while hyperparameter tuning:

Learning Rate	Epochs	No. of neurons	Training	Testing
0.1	10	30	0.9113	0.8818
0.1	10	12	0.9134	0.8868
0.1	50	30	0.9141	0.8849
0.9	20	28	0.9146	0.8905

Table 5: Observed accuracies with different hyperparameter values

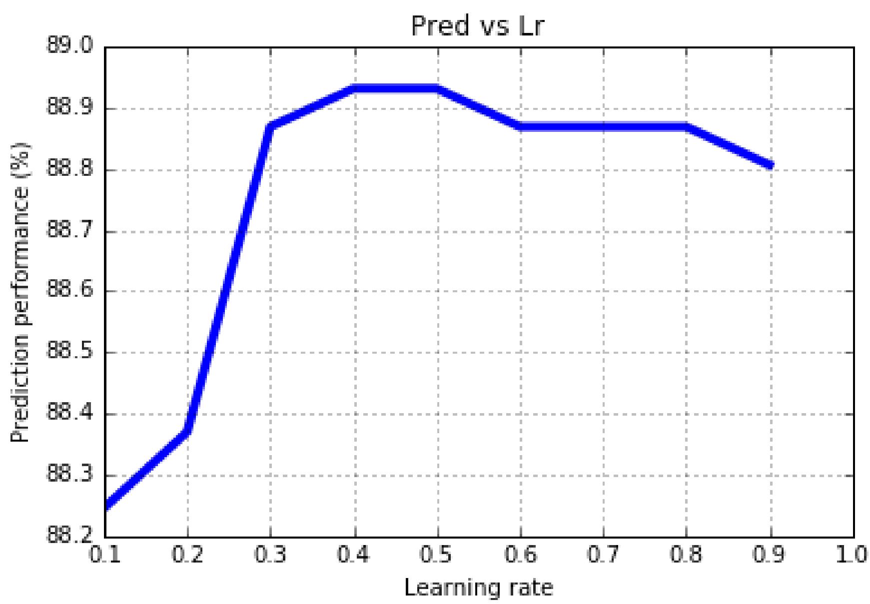

Figure 11 : Learning rate vs prediction performance

Further, Advantages and Disadvantages:
Neural networks are advanced enough to detect any complex relationships between inputs and outputs as well, which is another advantage when using this model. Neural networks are not without their disadvantages. Due to the complicated and advanced nature of the model, they are very difficult to design.

While the adaptability and sensitivity of a neural network is most certainly an advantage, it does also come with problems. Given that a neural network will react to even the smallest change in data, it can often be very hard to model analytically as a result.Running a neural network also requires a huge amount of computing resources, making it expensive, and possibly impractical, for some companies and applications.

7) SVC

Support vector machines (SVM) is a family of algorithms that have been implemented in classification, recognition, regression and time series. SVM emerged from research in statistical learning theory on how to regulate generalization, and find an optimal tradeoff between structural complexity and empirical risk.

SVM classify points by assigning them to one of two disjoint half spaces, either in the pattern space or in a higher-dimensional feature space.

The main idea of support vector machine is to construct a hyperplane as the decision surface such that the margin of separation between positive and negative examples is maximized.

For a training set of samples, with input vectors $x_i \in R^d$ and corresponding labels $y_i \in \{+1,-1\}$, SVM learns how to classify objects into two classes.

The choice of kernel function is a critical decision for prediction efficiency. Both polynomial and radial basis functions were adopted in experiments. Several levels of the degree of polynomial function (d), gamma constant of radial basis function (c) and regularization parameter (c) were tested in the parameter setting experiments. The SVM parameters and their levels are summarized in below table.

Parameters	**Levels (polynomial)**	**Levels (radial basis)**
Gamma in kernel function (c)	0, 0.1, 0.2, … ,5.0	0, 0.1, 0.2, … ,5.0
Regularization parameter (c)	1,10,100	1,10,100

Table 6 :SVM parameter levels tested in parameter setting experiments

No	Kernel function	d	γ	C	Training	Testing	Average
1	RBF	-	2.5	100	0.9125	0.9038	0.9081
2	RBF	-	5.0	100	0.9125	0.9001	0.9063
3	RBF	-	3.1	100	0.9125	0.9041	0.9083
4	Linear	-	-	100	0.9036	0.8992	0.9014
5	Polynomial	1	3.5	100	0.9003	0.8982	0.8992
6	Polynomial	1	0.3	100	0.9033	0.9032	0.9032
7	Polynomial	1	0.5	100	0.9064	0.9002	0.9033

Table 7 : Best three parameter combinations of SVM model

The data sets were applied to the SVM models with three different parameter combinations and the results are given in above table.

8) Advantage and disadvantage

Since the kernel implicitly contains a non-linear transformation, no assumptions about the functional form of the transformation, which makes data linearly separable, is necessary. The transformation occurs implicitly on a robust theoretical basis and human expertise judgment beforehand is not needed.

SVMs provide a good out-of-sample generalization, if the parameters C and r (in the case of a Gaussian kernel) are appropriately chosen. This means that, by choosing an appropriate generalization grade, SVMs can be robust, even when the training sample has some bias.

SVMs deliver a unique solution, since the optimality problem is convex. This is an advantage compared to Neural Networks, which have multiple solutions associated with local minima and for this reason may not be robust over different samples.

The disadvantages of SVM are that the theory only really covers the determination of the parameters for a given value of the regularisation and kernel parameters and choice of kernel. In a way the SVM moves the problem of over-fitting from optimising the parameters to model selection. Sadly kernel models can be quite sensitive to over-fitting the model selection criterion

ARIMA combines auto regression--which fits the current data point to a linear function (usually) of some prior data points--and moving averages--adding together several consecutive data points and getting their mean, and then using that to compute estimations of the next value and advantage is that, with enough elements regressed and averaged, you can fit an approximation to almost any time series you like, to whatever precision you like.

The trouble, of course, is Slutzsky's theorem: Slutzsky showed that, by using ARIMA type computation, and perhaps adding a trend line or two, you can take random noise into any time series you like... The point? Well, it basically means that you may fit the data magnificently