International Journal of Management Research and Emerging Sciences
Homepage: http://ijmres.pk/
Volume 9 Issue 1 (2019) PP. 83-97

PREDICTING STOCK MARKET DIRECTION USING MACHINE LEARNING MODELS

Dr. Faheem Aslam
Assistant Professor, Comsats University, Islamabad.
Corresponding Email: fahimparacha@gmail.com
Muhammad Ather Yaqub
Comsats University, Islamabad. Email: ather.yaqub@comsats.edu.pk
Beenish Bashir
Comsats University, Islamabad. Email: beenish128@hotmail.com

ABSTRACT

Forecasting of stock prices has been a challenging area due to its complex and dynamic nature. There are several evidences that traditional econometrics based predictive models encountered significant challenges due to parameter instability. The aim of this study is to apply three classifiers namely, Random Forest (RF), Support Vector Machines (SVM) and Neural Networks (NN) to predict the Pakistani stock market's direction and to compare the prediction accuracy. Daily closing prices are collected from yahoo server from 2013 to 2018. Famous 30 market indicators are applied to predict the market direction by using Random Forest, Support Vector Machines and Neural Networks. Model accuracy is evaluated using the confusion matrix. The empirical findings reveal that Neural Network performs best with the highest accuracy of 91%. Model specific, top five input indicators are used by applying feature selection in all classifiers. Interestingly, optimization improves the prediction accuracy in case of neural networks (NN) and support vector machine (SVM)) models while Random Forest's (RF) accuracy did not improve. These findings have great importance for institutional investors and management companies having flexibility to accelerate or postpone their investment decisions.

KEYWORDS: Machine Learnings, Stock Market, Neural Networks, Support Vector Machine, Random Forest, Pakistan

1. INTRODUCTION

Direction of stock price prediction is among the most challenging problems and is being extensively studied by academics from diverse backgrounds including computer science, finance, economics and mathematics. Simple time series or regression techniques are very difficult to apply on stock price owing to their volatile nature. Further predicting future direction or performance becomes even more challenging as information regarding the stocks is generally incomplete, uncertain and as stock markets are considered to be semi-strong efficient (Malkiel and Fama, 1970). Efficient market hypothesis states any attempt at market or stock prediction is pointless, as any information which may lead to any change in the price or market index should have already been accounted for recent stock prices. An accurate stock market prediction is important for many reasons, mainly for investor's need, to take hedge decisions against probable opportunities and market risks, and for speculators and arbitrageur, enabling them to earn profits. When it comes to reliability of traditional econometric analysis methods, they mostly rely upon the historical data and due to the problem of correlated evidences, are often questioned. Considerable returns could be earned if a stock future value is efficiently forecasted. Advances in computation techniques has led to the use of Machine learning (ML.) algorithms to estimate future stock prices. Significant number of studies point to strong evidences that these techniques are capable of identifying and predicting the financial markets Takeuchi and Lee (2013), Huck (2009, 2010), Moritz and Zimmermann (2014), Atsalakis and Valavanis (2009), Dixon, Klabjan, and Bang (2015), Krauss, Do, and Huck (2017). Nonetheless, the challenge of stock forecasting has always been there as just a minute changes can increase or decrease returns by millions for stock market participants.

Data analysts now also have a very important and difficult task to forecast the outcome from several input variables. For instance, prediction of an individual to repay the loan, prediction of firm's financial bankruptcy, deciding an email is spam and determining the probability of heart attack. All of them includes the forecasting of a binary categorical outcome (spam/not spam, heart attack/no heart attack, good credit risk/bad credit risk) from a basket of features, with the aim to discover a process to classify new events into one of the two possible outcomes. Numerous classification methods are offered by supervised machine learning which are employed in prediction of categorical results, including decision trees, random forests, logistic regression, support vector machines and neural networks. These models enable us to predict what may transpire in the future based on past observations. Combining this with domain knowledge, expertise, and business logic enables analysts to make data driven decisions, which is the ultimate outcome of predictive analytics.In this study, we tried to tackle an important aspect of prediction of stock market, i.e. Stock direction. We will be looking at the UP and DOWN direction of the KSE-100 index. Where the market is UP denoted by 1 and DOWN, which is denoted by 0. we will be building predictive models using the following ML algorithms:

- Neural networks
- Support vector machines
- Random forests

We have chosen the above algorithms to demonstrate a group of recent machine learning classifiers and compare the model performances using various parameters. Performance of each model is evaluated by comparing with each other and an optimal model for forecasting ability is proposed. Main findings include that in Full Model, Random forest gives accuracy of 92.58% . However, the results changed after hyperparameter optimization and feature selection. In optimized model, Neural Network outperform with prediction accuracy of 91.21%. Top five input indicators were used based on feature selection in all classifiers. It is quite interesting to see that the results in SVM and NN optimized algorithms show improvement in accuracy and precision except in case of RF where both accuracy and precision decreased in the optimized model. It is relieved that addition of more features in optimized RF can improve the model accuracy. Overall, Neural network stands to be the best prediction algorithm that can be used for market direction.

2. LITERATURE REVIEW

There are several methods to predict stock behavior such as Time series analysis, technical analysis, using differential equations for modeling and predicting stock volatility and using machine learning. With technological enhancements, due to high robustness and efficiency, machine learning algorithms have been widely used for stock market forecasting. Using Machine Learning (ML) in stock forecasting is a deviation from traditional analysis techniques namely, multivariate and time series analysis. Researchers have been using several ML algorithms such as Neural networks, Decision trees, support vector machine, random forest etc. for predicting the market trends, both short-term and long-term. Focus of most researchers was the developed economies. There are a very few studies which focus on the comparison of different ML methods, particularly with respect to Pakistani stock exchange. This study focuses on the three most popular classifiers namely; SVM, NN and RF ML algorithms to predict the daily KSE100 index direction using technical indicators.

Historically, the era of machine learning starting somewhere in the 1970's or perhaps 1980's but fact of the matter is that it was first created by McCulloh in 1940 's. he was first known person to perceive idea of Neural networks in his landmark publication in which he developed a computational model for neural networks (McCulloch, W.S. and Pitts, 1943) followed by D. O. Hebb in 1940's further this research and published his book in 1949 by title of "Organization of Behavior " he called it "Hebbian learning". In 1950's Fairly and Clark utilized the Hebbian learning approach to develop first computational machine by the name of "Calculator" (Farley and Clark, 1954). The prediction using non-linear models was earlier put forward by (Wolff, 1988) who concluded that in settings of stock markets the nonlinear models proved better predictors as compared to traditional linear models. One of the main characteristics which defines financial forecasting is noise within the data, non-stationarity, nature of data which is highly un structured along with uncertainty, veiled relationships between most of the variables and most important of all intensity of data. Hoque and Latif (1993) found out that ANNs (Artificial Neural Networks) can substitute the econometric models which are said to predict stock price movement and direction, the main reason for these findings are mainly due fact that ANNs incorporate solutions using incomplete, noisy and complex data variables. Very recently, Ren, Wu and Liu (2019) further add that ANNs are universal function approximators that are able to map out most of nonlinear functions within the data without having to utilize prior assumptions. Due to these reasons they are widely applied to

not only predict the stock market movements but also enable researchers to explore potential scenarios by utilizing complex simulations (Sarantis and Stewart, 1995; A. S. Chen and Leung, 2004).

Most of the studies applied different forms of Neural networks to forecast the asset prices and returns. Recently, Fischer & Krauss (2018) deployed Long Short-Term Memory NN on basic S&P 500 stocks from 1992 till 2015, having sharp ratio of 5.8 excluding transaction cost to predict their out-of-sample directional movements. From a different perspective, Hsu et al. (2016) has explained that prediction of financial markets can be done more accurately by using the machine learning methods instead of econometric methods. He demonstrated that stock market maturity, prediction time and choice of technique affected the forecasting outcome of the financial market. Niaki and Hoseinzade (2013) forecasted S&P index by applying an ANN model with twenty-seven economic inputs. The results of his study confirm that ANN model, significantly enhanced the accuracy of prediction. Adhikari R., & Agrawal RK (2014) used mixed models are also used for stock prediction and proposed random walk (RW), Elman artificial neural network (EANN) and forward artificial neural network (FANN) which had a better predictive power than other models. Likewise, Zbikowski (2015) worked on a Fisher-based SVM model. Oztekin et al. (2016), Gerlein et al. (2016) and Ballings et al. (2015) presented diverse methods, such as Bayesian , ANN, SVM methods to compare the effect of each forecasting technique.

Table 1: Literature of Different Prediction Models

Authors	Algorithm	Data frequency
Patel et al. (2015)	SVR, RF, NN	Weekly
Miró-Julià (2010)	Decision Tree	Weekly
Ballings et al. (2015)	SVM, KNN, NN	Yearly
Zbikowski (2015)	SVM	Weekly
Tiwari et al. (2010)	Decision Tree	Weekly
Qu and Zhang (2016)	SVR	Intraday
Wu et al. (2006)	Decision Tree	Weekly
Tay and Cao (2001)	SVM	Weekly
Choudhury et al. (2014)	SVR	Intraday
Arau´jo et al. (2015)	NN	Per Second
Kim (2003)	SVM	Weekly
Nayak et al. (2015)	SVM, KNN	Weekly

Second important classification algorithm used in similar studies is Support Vector Machine (SVM). The dynamic, nonlinear and evolutionary properties of stock movement make it extremely difficult to predict. Support vector machine (SVM) converts the nonlinear data to quadratic and due to this has been widely utilized for stock market forecasting. Further, SVM gives a unique and optimal solution (Huang, Nakamori and Wang, 2005). Yu et al. (2009) came up with PCA and SVM hybrid models to predict future direction of stock market. The overfitting problem is also reduced by SVM through selection of maximal margin hyperplane during the feature space (Yu et al, 2009). Vapnik (2013) proposed SVM as a supervised learning method which partially addresses the overfitting problem. The problem of nonlinearity could be solved by deploying different kernel functions and projecting the nonlinearity on a high-dimensional feature space. Weekly directional movement of Tokyo Stock Exchange was forecasted by Huang et al (2005) by using SVM . Yu et al. (2009) proposed an evolving least squares SVM and explored the trends in S&P 500 Index, New York Stock Exchange Index and DJIA Index. Istanbul Stock Exchange daily 100 Index was forecasted using SVM by Kara et al. (2011) with an average predicting power of 71.52%. Besides, SVM, jointly with other models achieve a better performance.

A model consisting of SVM integrated with other classification models performed better in forecasting NIKKEI 225 Index (Huang, Nakamori and Wang, 2005). Pai and Lin (2005) used an ARIMA and an SVM mixed model to forecast stock prices. Kim (2003) and Kumar et al. (2016) used SVM-based system with technical indicators and predicted the direction of share prices. Barak & Modarres (2015) used predictive variables extracted from the published financial statements. Zbikowski (2015) used fisher score for variable selection in a modified SVM. He used the Relative Strength Index (RSI), On Balance Volume (OBV) and Williams oscillator indicators in his study and SVM approach was used to generate the results to predict the market. Yeh et al. (2011) and Lu et al. (2009) forecasted

the TAIEX and Nikkei 225 indices through SVR-based analysis using daily data. Patel et al. (2015a)'s model forecasted the market trend instead of price by using the market trend indication as shown by predictive variable in SVM.

Few recent studies also focused on Random Forest (RF). For instance, Basak et al. (2019) proposed that gradient boosted decision trees (using XGBoost) along with RF, using an input of technical indicators predicted medium to long run stock market direction with high accuracy. Kumar et al. (2018) compared the five ML models; RF, K Neural Network, SVM, Naïve Bayes and softmax and predicted trend of Indian stock exchange with Random forest giving highest accuracy for large dataset. Future direction of share prices was forcasted with an accuracy of 85% to 95% by Khaidem et al (2016) by using random forest. Khaidem, Saha and Dey (2016) plugged in 6 stock market indicators to train and test the RF classifier. A summary of the research studies is reported in Table 1. It is clear from the algorithm data frequency that most studies have used daily data sets to forecast direction by using different algorithms.

3. DATA AND METHODOLOGY

The data and methodology part consist on several steps. A brief detail of all the steps is given below.

3.1 Step 01: Reading KSE Data

Daily closing values of KSE-100 index are downloaded from Yahoo Finance Server by using the quantmod package in R. The time span of the daily closing prices ranges from 11-June-2013 to 06-Nov-2018 (which corresponds to 1305 observations).

3.2 Step 02: Calculate Output Variable

The output is the direction of KSE-100 index, which is divided into UP and DOWN. If the price change > 0, it is equal to "1", otherwise "0"

3.3 Step 03: Calculate Indicator Variables-Data Features

A total of 30 indicator variables are calculated by using the TTR package in R. Below in Table 2, the details of the Indicators is reported.

3.4 Step 04: Data Transformation

In this step, Data type conversions, missing values imputation, and scaling & normalization are performed.

3.5 Step 05: Training and Testing Data

The data is divided into two data sets; training data and test data set with 70:30 ratio. The training data (N=) refers to the data is solely used to train the predictive models. The machine learning algorithm picks up the tuples from training dataset and tries to find out patterns and learn from the various observation instances. While the test data is used to get predictions and model performance.

3.6 Step 06: Model Training

In this step, we used three supervised machine learning algorithms and feed the training data features to them and build the predictive model.

Table 2: List of KSE-100 Indicators

Identifier	Indicator Name	Outputs
RSI	Relative Strength Index	1
ATR	Average True Range	4
MACD	Moving Average Convergence/Divergence	1
BB	Bollinger Bands	4
WPR	William's %R	1
Donchian	Donchian Channel	3
CCI	Commodity Channel Index	1
CMO	Chande Momentum Oscillator	1
ROC	Rate of Change	1
Aroon	Aroon	3
EMAcross	Exponential Moving Average	1
SMI	Stochastic Oscillator / Stochastic Momentum Index	1
chaikinAD	Chaikins Volatility	1
CLV	Close Location Value	1
CMF	Chaikin Money Flow	1
ADX	Average Directional Movement Index	1
DPO	Detrended Price Oscillator	1
MFI	Money Flow Index	1
SAR	Paracolic Stop and Reverse	1
OBV	On Balance Volume	1

3.7 Step 07: Predictive Model

We have used SVM, RF and NN for prediction task. A very brief introduction of all four techniques are given below:

3.7.1 Support vector machines

SVMs are set of supervised machine-learning algorithm. It classifies the cases by constructing hyperplanes in a multidimensional space. It is commonly used for classification problem. Recently, SVM are famous, due to its elegance n-1-dimensional space plotting of the data. SVMs are very successful in classification and prediction. Simply the co-ordinates of the tuple are the Support Vectors. An optimal hyperplane divides the data classes in a multidimensional space. Data points, coming to the incorrect side of the margin are weighed down so as to reduce their influencing power and this is called the soft margin compared to the hard margins of separation. The optimal hyperplane maximizes the distance between nearest data point and hyper-plane which will help make a decision regarding the correct hyper-plane. We can visualize how an SVM classifier actually looks much better with the following figure from the official documentation for the SVM library in R.

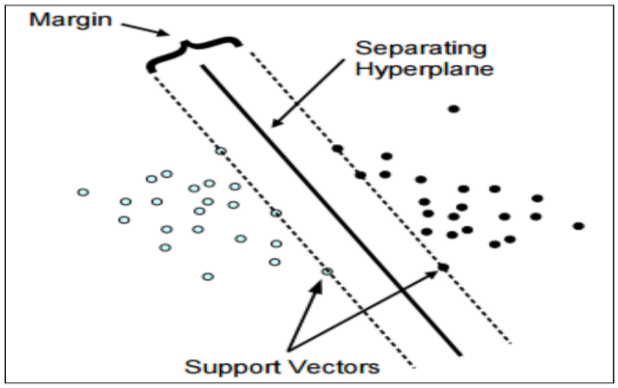


Image source: https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf

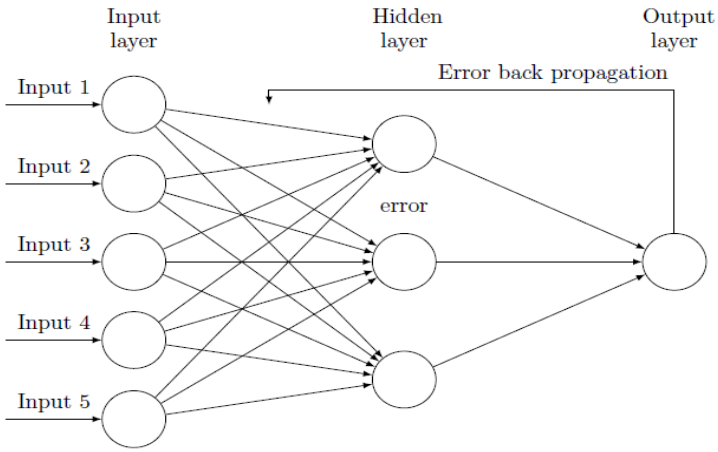
Fig. 1: Support Vector Machine

3.7.2 Random forests

Also known as random decision forests, RFs consists on large number of individual decision trees that work in collaboration with each other. There are several models available to improve the forecasting accuracy.1 These machine learning algorithms can be very easily applied for both classification and regression. This algorithm has a comparative edge of great prediction accuracy, even without hyper-parameter tuning. RF builds multiple decision trees and creates a forest and makes it somehow random. The algorithm adds randomness by sampling with replacement method in the training dataset. Best features are searched from a random subset of features during the growth process of decision trees. This introduction of randomness into the model increases the bias of the model slightly but decreases the variance of the model greatly which prevents the overfitting of models, which is a serious cause of worry in case of the decision trees. Overall, this yields much better performing generalized models. Prediction error is measured by the average forecasting error, i.e. out of bag error (OOB) by using bagging to sub sample training data.Like other ML algorithms, RF also shows the variable importance for feature selection. For analysis purpose, we have used "randomForest()" function in the random-Forest package.

3.7.3 Neural networks

Neural networks are human brain-inspired processors which have ability to learn by training and then store the learned experience for use at later stage when required. They have the ability to derive meaning from complicated data. Neural networks have advantages as compared to the traditional linear models due to their non-linear nature. They have the capability to recognize the non-linear relationship in the predicting data sample without prior knowledge of relationship among the predicting and output variables. Neural network has the capability to change its parameters (weights) when dealing with non-stationary and dynamic data. A special function like sigmoid transforms input variables into output variables.



Fig, 2: Neural Network

In order to minimize the error between predicted (output) and actual (target), error is propagated back into the neural network. Based on the size of initial error, weight adjustment between neurons on each connection is done and the input data is again fed to produce a new output and error. This procedure continues till the acceptable level of error is achieved. Sigmoid is commonly used function in the neural networks and its value ranges from 0 to 1.For analysis purpose, we used packages of "caret" to run the neural network and package "ROCR" for model evaluation.

3.8 Step 08: Model Selection

At start we use 30 features. In model selection step, based on maximum accuracy, we select a predictive model from several iterations of predictive models.

[1]. *Details are available at http://mng.bz/7Nul.*

3.9 Step 09: Hyperparameter Optimization

After feature selection, we try to choose a set of the hyperparameters used by the algorithm in the model such that the performance of the model is optimal with regards to its prediction accuracy.

3.10 Step 10: Model Evaluation

The model evaluation is done with following criteria[2].
Below is the list of parameters used for the evaluation of prediction accuracy.

Table 3: List of Parameters to Evaluate Prediction Accuracy

$$\text{Specificity (TNR)} = \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (1)$$

$$\text{Sensitivity (TPR)/Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \quad (2)$$

$$\text{Precision(PPV)} = \frac{\text{TP}}{\text{FP} + \text{TP}} \quad (3)$$

$$\text{NPV} = \frac{\text{TN}}{\text{FN} + \text{TN}} \quad (4)$$

$$\text{Fallout/FPR}(1 - \text{Specificity}) = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (5)$$

$$\text{Miss Rate/FNR}(1 - \text{Specificity}) = \frac{\text{FN}}{\text{TP} + \text{FN}} \quad (6)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \quad (7)$$

$$\text{F1 Score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad (8)$$

4. EMPIRICAL FINDINGS

4.1 Support Vector Machine

Predictive results of SVM are shown in Table 4. In Figure 3 and Figure 6, overall performance of model is summarized in four-fold confusion matrix plot with the precision score of 66.81% in full model and 89.40% in optimized SVM. The overall accuracy is 77.47%. In optimized model there are top 5 input variables as represented in Figure 04. In Figure 04, greater importance shows more important input variables. The top five variables as per feature selection importance graph are: CLV, ROC, WPR, CCI and PercentageB. Furthermore, we specified 168 (eight γ values * twenty-one cost values) models for 10-fold hyperparameter optimization. The minimum error is achieved at $\gamma = 0.01$ and cost = 100 (Figure 05).

[2] *Confusion matrix which is a nice way to see how the model is predicting the different classes. It reports the number of predicted values in each class against the actual class values in two rows and two columns table. The total number of predictions with the DOWN (0) class label which are actually having the DOWN label is called True Negative (TN) and the remaining DOWN instances wrongly predicted as good are called False Positive (FP). Correspondingly, the total number of predictions with the UP (1) class label that are actually labeled as UP are called True Positive (TP) and the remaining UP instances wrongly predicted as DOWN are called False Negative (FN).*

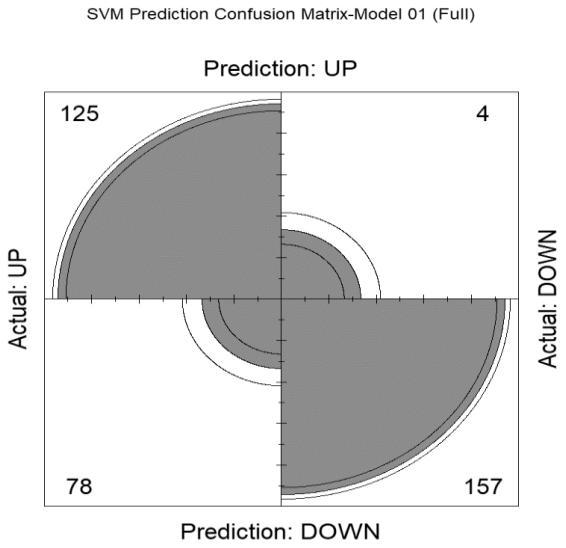


Fig. 3:SVM Prediction using all Features

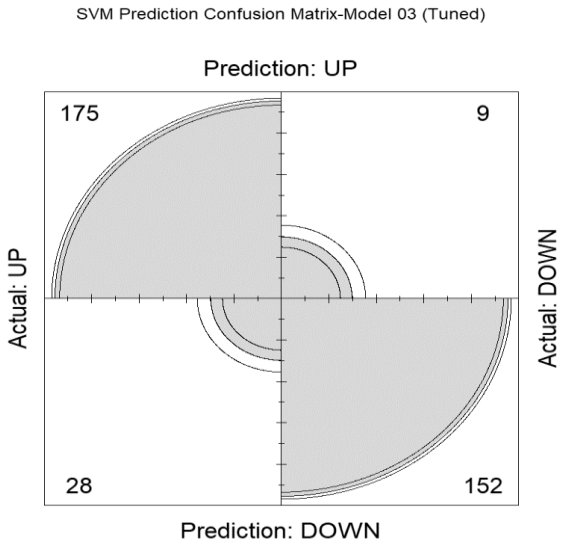


Fig 6: SVM prediction using tuned SVM - Selected Features

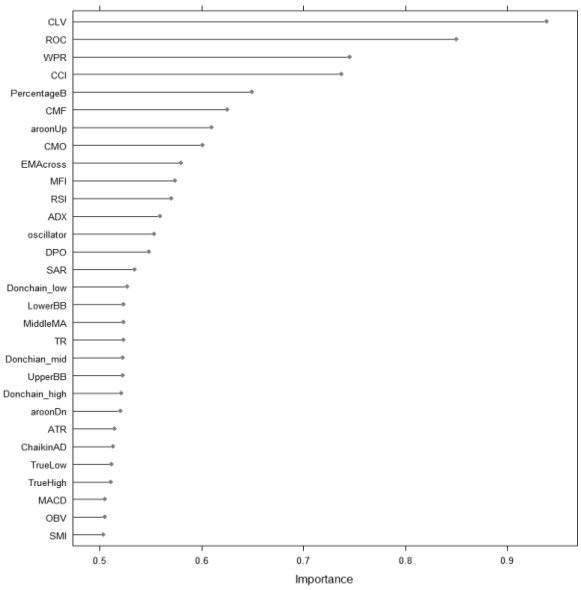


Figure 4: Feature Selection SVM

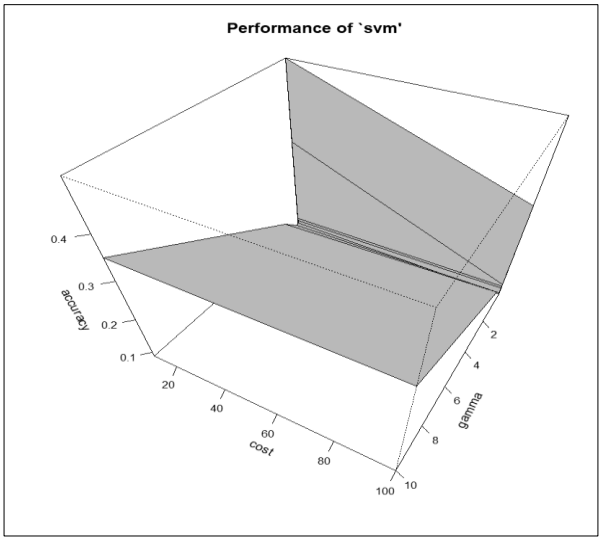


Fig 5: Hyperparameter Optimization – SVM

We clearly notice a significant improvement in the accuracy of the SVM results with hyperparameter optimization (Figure 5) which improves from 77.47% to 89.84% in line with the study of Huang, Nakamori and Wang (2005).In the optimized model SVM shows 92.12% True negative rate and 83.85% True positive rate while False positive and false negative are 7.88% and 16.15% with the model precision of 89.40% and over all accuracy of 88.46%. we can notice a significant improvement in the accuracy of the SVM results with hyperparameter optimization which improves from 77.47% to 88.46%.

Table 4: Prediction using SVM

Measures	SVM Full Model	SVM Optimized (Selected Features)
Specificity (TNR)	0.6158	0.9212
Sensitivity (TPR) / Recall	0.9752	0.8385
Precision (PPV)	0.6681	0.8940
NPV	0.9690	0.8779
Fallout / FPR (1-Specificity)	0.3842	0.0788
MISS RATE / FNR (1-Specificity)	0.0248	0.1615
Accuracy	*0.7747*	0.8846
F1 Score	0.7929	0.8654

4.2 Random Forest (RF)

The results of Random Forest are presented in Table 05. The four-fold confusion matrix plot is presented in Figure 07 and Figure 10. Feature importance according to mean decrease accuracy and mean decrease Gini is represented in Figure 8 and hyper parameter optimization is presented in Figure 09. From Table 5, the RF full model with 30 input vectors are used, overall accuracy is 92.58% with 91.63% of TNR and 93.79% of TPR. In optimized model only top five important variables: CLV, ROC, WPI, CCI and AroonUp are employed. In the optimized model RF shows an approximate 10.34% error in both Specificity (TNR) and Sensitivity (TPR) with the model precision of 87.35% and accuracy of 89.84% comparable with the accuracy range of previous study (Khaidem et al. 2016). 10% error in TNR and TPR means that RF optimized correctly predicted the 90% of output.

Table 5: Prediction using Random Forest

Measures	RF Full Model	RF Optimized (Selected Features)
Specificity (TNR)	0.9163	0.8966
Sensitivity (TPR) / Recall	0.9379	0.9006
Precision (PPV)	0.8988	0.8735
NPV	0.9490	0.9192
Fallout / FPR (1-Specificity)	0.0837	0.1034
MISS RATE / FNR (1-Specificity)	0.0621	0.0994
Accuracy	0.9258	0.8984
F1 Score	0.9179	0.8869

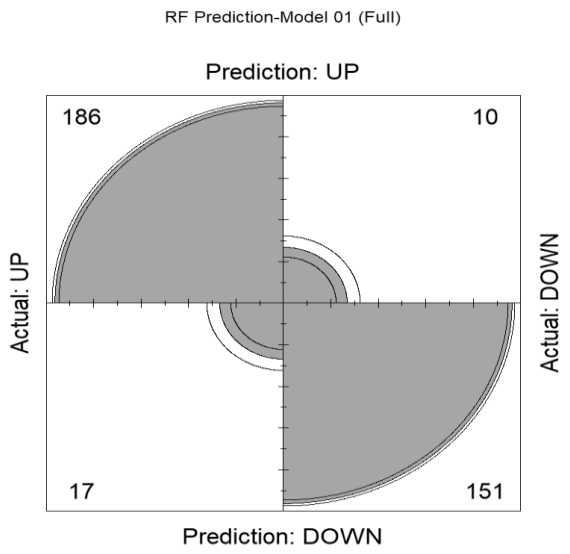


Fig 7: RF Prediction using all Features

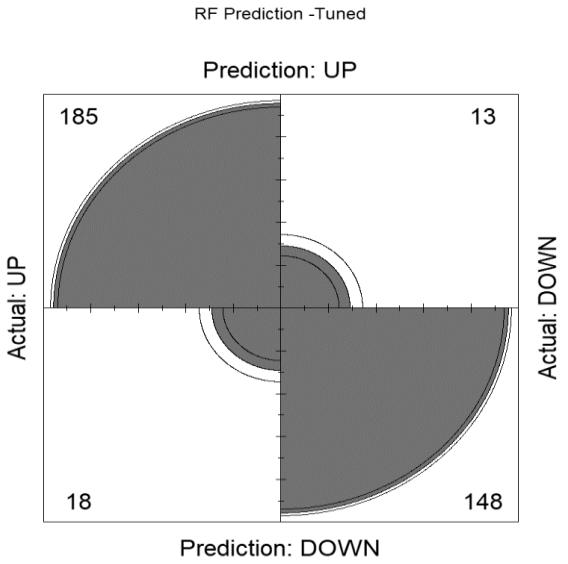
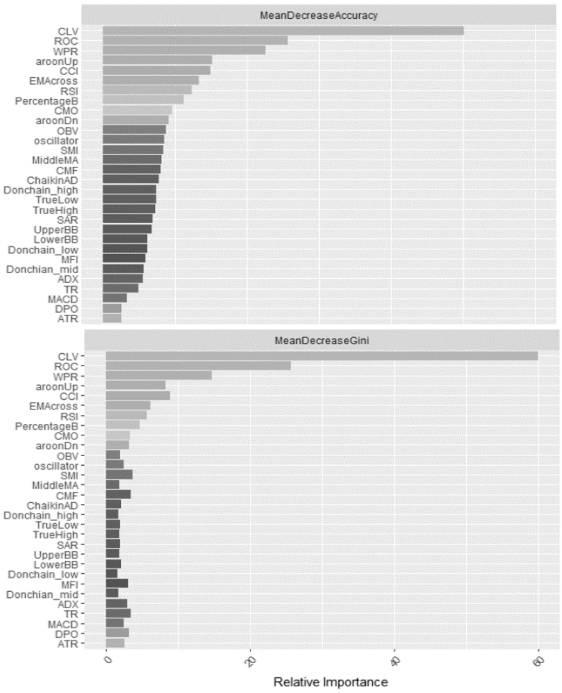


Fig 10: Tuned RF prediction using Selected Features

Fig 8: Feature Selection - RF



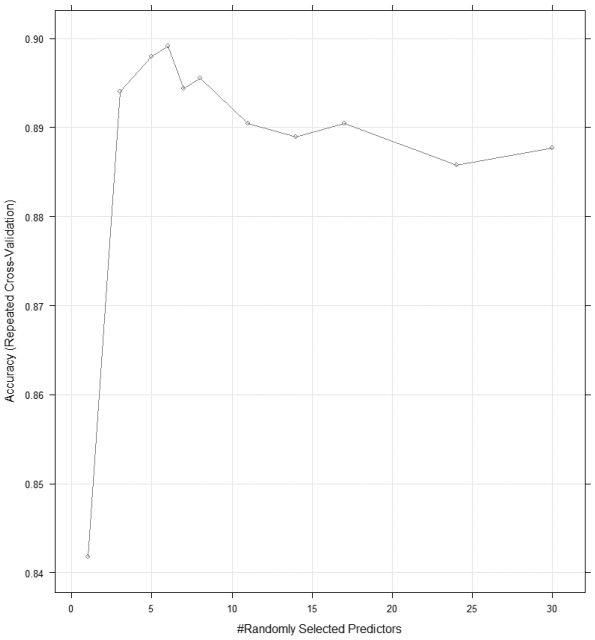


Fig9: Hyperparameter Optimization - RF

4.3 Neural Network (NN)

The results of Neural Network are presented in Table 06. The confusion matrix is presented in Figure 11 and Figure 15. The best fitted NN model is shown in Figure 12. In Table 6, NN full model with all 30 predictor variables are used, the overall accuracy is 88.46% . In optimized model we use the important variables, as shown in Figure 13. The top five selected variables are: ROC, CCI, EMACROSS, CLV and SAR.

Table 6: Prediction using Neural Networks

Measures	NN Full Model	NN Optimized (Selected Features)
Specificity (TNR)	0.8079	0.9113
Sensitivity (TPR) / Recall	0.9814	0.9130
Precision (PPV)	0.8020	0.8909
NPV	0.9820	0.9296
Fallout / FPR (1-Specificity)	0.1921	0.0887
MISS RATE / FNR (1-Specificity)	0.0186	0.0870
Accuracy	0.8846	0.9121
F1 Score	0.8827	0.9018

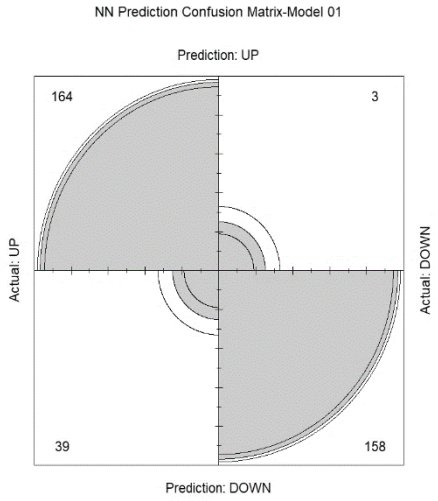


Fig 11: NN Prediction using all Features

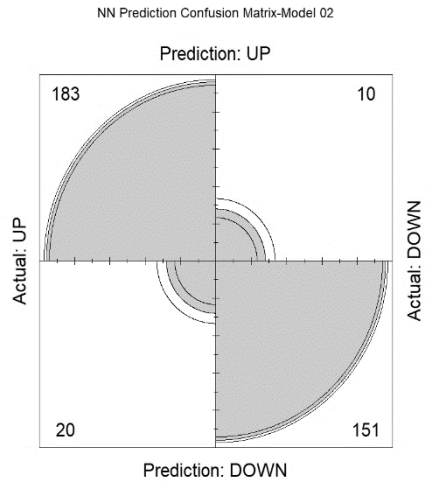


Fig 15: Tuned NN prediction using Selected Features

The optimized model shows significant increase in overall accuracy to 91.21% from 88.46% in full model which is comparable with the previous study (Kara et al.2011). TNR and TPR shows 91% of down and up directions are correctly predicted by optimized NN while FPR and FNR shows approximately 8.70% of up and down directions were incorrectly predicted by NN model. Figure 12 shows the structure of neural network used in this research work. This neural network consists of input layer with five nodes representing five selected features, single hidden layer with number of nodes equal to number of input variables and the output layer representing the predicted variable.

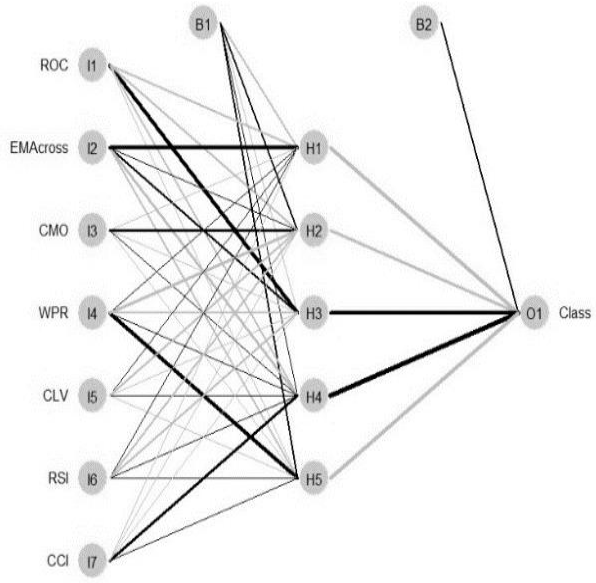


Fig 12: NN result: best fitted Model

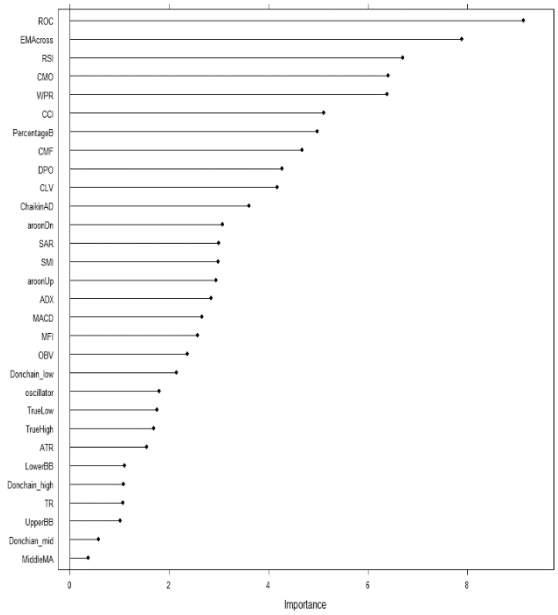


Fig 14: Feature Selection NN

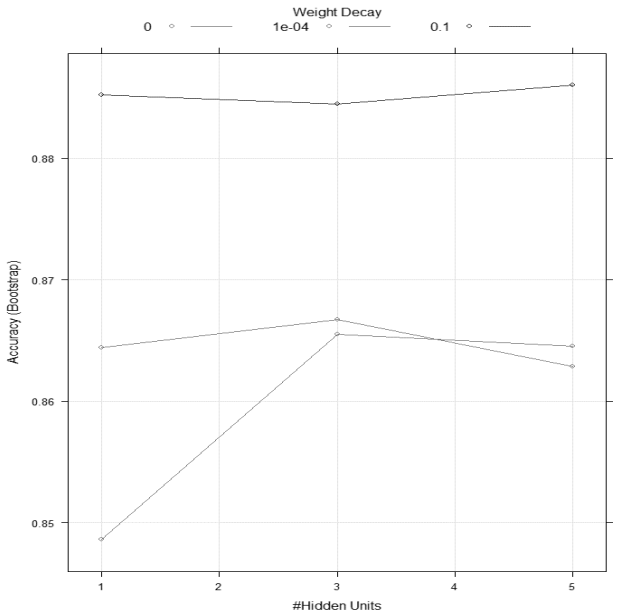


Fig 13: Hyperparameter Optimization - NN

5. DISCUSSION

Models accuracy comparison is reported in Table 7 using all 30 input indicator variables along with target variable Class. It shows that in full model Random forest gives accuracy of 92.58% and out of remaining 7 criteria Random forest performs better in 4 criteria (Alavi et al. 2015, Han et al. 2018). In Table 8 a comparison of optimized models is carried out which clearly shows Neural Network as a winner in most of the model selection criteria with the

highest accuracy of 91% (Kara et al. 2011, Raczko and Zagajewski, 2017). The top five input indicators are selected by using feature selection with respect to Random Forest, Neural Network and Support Vector Machines classifiers.

Table 7: Comparison of Full Models

Measures	SVM	RF	NN	Best Model
Specificity (TNR)	0.6158	0.9163	0.8079	RF
Sensitivity (TPR) / Recall	0.9752	0.9379	0.9814	NN
Precision (PPV)	0.6681	0.8988	0.8020	RF
NPV	0.9690	0.9490	0.9820	NN
Fallout / FPR (1-Specificity)	0.3842	0.0837	0.1921	RF
MISS RATE / FNR (1-Specificity)	0.0248	0.0621	0.0186	NN
Accuracy	0.7747	0.9258	0.8846	RF
F1 Score	0.7929	0.9179	0.8827	RF

Table 8: Comparison of optimized Models

Measures	SVM	RF	NN	Best Model
Specificity (TNR)	0.9212	0.8966	0.9113	SVM
Sensitivity (TPR) / Recall	0.8385	0.9006	0.9130	NN
Precision (PPV)	0.8940	0.8735	0.8909	SVM
NPV	0.8779	0.9192	0.9296	NN
Fallout / FPR (1-Specificity)	0.0788	0.1034	0.0887	SVM
MISS RATE/FNR (1-Specificity)	0.1615	0.0994	0.0870	NN
Accuracy	0.8846	0.8984	0.9121	NN
F1 Score	0.8654	0.8869	0.9018	NN

6. CONCLUSIONS AND RECOMMENDATIONS

Due to nonlinear and chaotic nature of times series; its complex to predict its direction. However, recent development in machine learning techniques support stock market forecasting. By using the daily frequency of stock market data, three classifiers namely ANN, SVM and RF are employed for predicting the direction. The robustness of the three models is evaluated on eight parameters such as accuracy, precision, sensitivity and specificity. It is quite interesting to see that the results in SVM and NN optimized algorithms show improvement in accuracy and precision except in case of RF where both accuracy and precision decreased in the optimized model. Only in case of RF, addition of input variables increased the prediction accuracy. Neural network stands to be the best prediction algorithm that can be used for market direction.

This study testifies that ML algorithms can be effectively used to forecast the market direction, refuting the EMH. In this research work neural network outperform other two predictive models, provides a guideline for model selection in case of emerging market price movements. Moreover, parameter tuning completely changes the model performance, augmenting the importance of parameter optimization. The findings of this study are useful of investment decision, particularly for the institutional investors who have flexibility in cash flows. In the light of these findings, several profit-making strategies can be designed by using index-based products. The simple one is to accelerate or postpone the buying and selling decision according to the market direction. Other strategies can be used by using put, reverse put options etc. These findings can be further utilized for predicting the direction of individual stocks which in-turn help the investors to build risk management models, developing new trading strategies, performing stock portfolio management and making profitable stock selection for the growth of the investment. The

analysis is limited to technical indicators as input vectors. In future macro-economic variables may also be added as input vectors to study their effect on market direction.

REFERENCES

Adhikari, R., & Agrawal, R. K. (2014). A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Computing and Applications*, *24*(6), 1441-1449.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–Part II: Soft computing methods. *Expert Systems with Applications*, *36*(3), 5932-5941.

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, *42*(20), 7046-7056.

Barak, S., & Modarres, M. (2015). Developing an approach to evaluate stocks by forecasting effective features with data mining methods. *Expert Systems with Applications*, *42*(3), 1325-1339.

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552-567.

Choudhury, S., Ghosh, S., Bhattacharya, A., Fernandes, K. J., & Tiwari, M. K. (2014). A real time clustering and SVM based price-volatility prediction for optimal trading strategy. *Neurocomputing*, *131*, 419-426.

Dixon, M., Klabjan, D., & Bang, J. H. (2015, November). Implementing deep neural networks for financial market prediction on the Intel Xeon Phi. In *Proceedings of the 8th Workshop on High Performance Computational Finance* (p. 6). ACM.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654-669.

Gerlein, E. A., McGinnity, M., Belatreche, A., & Coleman, S. (2016). Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, *54*, 193-207.

Han, T., Jiang, D., Zhao, Q., Wang, L., & Yin, K. (2018). Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery. *Transactions of the Institute of Measurement and Control*, *40*(8), 2681-2693.

Hsu, M. W., Lessmann, S., Sung, M. C., Ma, T., & Johnson, J. E. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications*, *61*, 215-234.

Huang, C. L., & Tsai, C. Y. (2009). A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Systems with Applications*, *36*(2), 1529-1539.

Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, *32*(10), 2513-2522.

Huck, N. (2009). Pairs selection and outranking: An application to the S&P 100 index. *European Journal of Operational Research*, *196*(2), 819-825.

Huck, N. (2010). Pairs trading and outranking: The multi-step-ahead forecasting case. *European Journal of Operational Research*, *207*(3), 1702-1716.

Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert systems with Applications*, *38*(5), 5311-5319.

Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003*.

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1-2), 307-319.

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, *259*(2), 689-702.

Kumar, D., Meghwani, S. S., & Thakur, M. (2016). Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets. *Journal of Computational Science*, *17*, 1-13.

Lee, E. J. (2015). High frequency trading in the Korean index futures market. *Journal of Futures Markets*, *35*(1), 31-51.

Lu, C. J., Lee, T. S., & Chiu, C. C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, *47*(2), 115-125.

Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, *25*(2), 383-417.

Miró-Julià, M., Fiol-Roig, G., & Isern-Deyà, A. P. (2010, June). Decision trees in stock market analysis: construction and validation. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 185-194). Springer, Berlin, Heidelberg.

Moritz, B., & Zimmermann, T. (2014). Deep conditional portfolio sorts: The relation between past and future stock returns. In *LMU Munich and Harvard University Working paper*.

Nayak, R. K., Mishra, D., & Rath, A. K. (2015). A Naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices. *Applied Soft Computing*, *35*, 670-680.

Niaki, S. T. A., & Hoseinzade, S. (2013). Forecasting S&P 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International*, *9*(1), 1.

Oztekin, A., Kizilaslan, R., Freund, S., & Iseri, A. (2016). A data analytic approach to forecasting daily stock returns in an emerging market. *European Journal of Operational Research*, *253*(3), 697-710.

Pai, P. F., & Lin, C. S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, *33*(6), 497-505.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259-268.

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.

Qu, H., & Zhang, Y. (2016). A New Kernel of Support Vector Regression for Forecasting High-Frequency Stock Returns. *Mathematical Problems in Engineering*, *2016*.

Raczko, E., & Zagajewski, B. (2017). Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral APEX images. *European Journal of Remote Sensing*, *50*(1), 144-154.

Takeuchi, L., & Lee, Y. Y. A. (2013). Applying deep learning to enhance momentum trading strategies in stocks. In *Technical Report*. Stanford University.

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *omega*, *29*(4), 309-317.

Tiwari, S., Pandit, R., & Richhariya, V. (2010). Predicting future trends in stock market by decision tree rough-set based hybrid system with HHMM. *International Journal of Electronics and Computer Science Engineering*, *1*(3).

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Wu, M. C., Lin, S. Y., & Lin, C. H. (2006). An effective application of decision tree to stock trading. *Expert Systems with Applications*, *31*(2), 270-274.

Yeh, C. Y., Huang, C. W., & Lee, S. J. (2011). A multiple-kernel support vector regression approach for stock market price forecasting. *Expert Systems with Applications*, *38*(3), 2177-2186.

Yu, H. Y., Niu, X. Y., Lin, H. J., Ying, Y. B., Li, B. B., & Pan, X. X. (2009). A feasibility study on on-line determination of rice wine composition by Vis–NIR spectroscopy and least-squares support vector machines. *Food Chemistry*, *113*(1), 291-296.

Zbikowski, K. (2015). Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. *Expert Systems with Applications*, *42*(4), 1797-1805.