International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies

http://TuEngr.com

TuEngr Group

PAPER ID: 11A04J

COMPARATIVE ANALYSIS OF TRADITIONAL AND SOFT COMPUTING MODELS FOR TRADING SIGNALS PREDICTION

Beenish Bashir [a], **Faheem Aslam** [a*]

[a] *Department of Management Sciences, COMSATS University Islamabad, PAKISTAN.*

ARTICLEINFO

Article history:
Received 11 July 2019
Received in revised form 20 November 2019
Accepted 29 November 2019
Available online 09 December 2019

Keywords:
Moving averages; Autoregressive models; Artificial neural network; Support vector machine; Financial marketing; Extreme learning machine.

ABSTRACT

Financial market forecasting always remains a challenging issue due to the nature of the time series data as well the information stock prices reflect. Economist takes this data as a linear process whereas the soft computing models assume that time-series data is nonlinear, complex and dynamic in nature and can be better analyzed through nonlinear models. This research paper settles the debate by conducting a comparative analysis of traditional forecasting models and soft computing models for the trading signals prediction of Pakistan stock exchange covering the daily data from 1997 to 2018. Our study is unique regarding the target variable it predicts which reflects the overall dynamics of the market not just the next period future price. In traditional models, we include moving averages as well as autoregressive models whereas in soft computing models we take artificial neural networks, support vector machines and extreme learning machines to have a comprehensive analysis. This comparison shows that soft computing models perform better than traditional models in trading signals prediction showing non-linear behavior of financial time series data. However, amongst all soft computing models, the artificial neural network becomes the best predictive model. Our results are more convincing as compare to existing literature regarding the careful selection of market features, which can help investors to better understand market behavior and improve the predictive ability of the model.

Disciplinary: Management Sciences (Financial Science).

1. INTRODUCTION

Financial markets are the barometer of any economy. It facilitates international trade, foreign investment, institutional investment acting as a part of mutual funds and portfolio management. Given their importance, much attention has been given by researchers and academia regarding its prediction. These predictions are then incorporated into financial risk models and developing

*Corresponding author (Faheem Aslam). Tel: +92-321-5063253. Email: fahimparacha@gmail.com

trading strategies. (Moradi & Rafiei, 2019; Zhong & Enke, 2019).

As the importance and predictability of financial markets are supported by previous studies (Fama, 1965; Malkiel & Fama, 1970) but when it comes to the modeling techniques for forecasting there is no consensus. The dispute is much related to the nature of the data financial markets generate. The long-term investors are more concerned about the fundamentals of companies like price-earnings ratio, revenue, expenses, assets, liabilities, management policy and financial ratio (Lam, 2004). Whereas, short-term investors rely on price movements of stock, understanding market behavior through different market features (Murphy, 1986). Technicians avoid the analysis of all economic factors by focusing on pattern recognition of price considering this information enough for future price determination (Hu et al., 2015; Patel et al., 2015; Teixeira & De Oliveira, 2010; Żbikowski, 2015). Tsinaslanidis and Kugiumtzis (2014) use numerous technical indicators for market analysis which is time-series data in nature. Huang et al. (2019) also use different momentum and volatility indicators for bitcoin predictive analysis.

For short-term prediction, traditional forecasting models consider time series data as a linear process and apply the smoothening and autoregressive process to predict future price movement (Kumar & Murugan, 2013; Lin et al., 2012; Wang et al., 2012). Financial time series are complex, nonlinear, dynamic and chaotic. Soft computing models can capture this nonlinear behavior (Cheng & Wei, 2014; Huang & Tsai, 2009; Lee, 2009). Tay and Cao (2001) compare artificial neural networks (ANN) and support vector machines (SVM) and confirm their suitability for prediction of the stock market. Huang, et al. (2005) use SVM for predicting the directional movement of the NIKKIE 225 index. Kara et al. (2011) also employ SVM and ANN for predicting the Istanbul Stock Exchange.

Our work is an addition to existing literature to settle the debate amongst traditional forecasting models and soft computing models for trading signals prediction. All the existing literature is related to price prediction, completely ignoring the trading signals forecasting that can enlighten investors about entry and exist decisions. Secondly, we also focused on the relevant feature selection to improve the predictive ability of models and to better understand market behavior. Lastly, our analysis confirms the nonlinear and dynamic nature of financial time series data negating the assumption of traditional forecasting models.

2. LITERATURE REVIEW

In the past ten years, various time series methodologies have been developed for financial market forecasting that helps improve investment decisions (Teixeira & De Oliveira, 2010). Traditional forecast models and soft computing models are two major approaches (Wang et al., 2011). In both modeling approaches, financial data is considered as a time series data which is actually numerical observations accumulated in sequential order over a period of time. (Brockwell & Davis, 2009; B. Wang et al., 2012). This organization of financial data enables model developers to utilize time series tools to understand market behavior (Oliveira & Meira, 2006). Further, these tools pave a way to do data mining tasks, such as classification, trend analysis, seasonal effect, cycles and extreme event detection (Cryer & Chan, 2008).

The traditional models involve averages, regression and autoregressive models based on the linearity of normally distributed variables (Lin et al., 2012; Wang et al., 2012). In reality, financial time series data is dynamic, chaotic, nonlinear and highly volatile which makes its prediction complex as compare to other non-financial time-series data (Vanstone & Finnie, 2009). Financial time series inherit these characteristics from economic factors, investor's sentiments, political events and movement of other stock markets (Kara et al., 2011).

The nature of financial time series data calls for flexible and adaptive models for future price prediction (Huang & Tsai, 2009) and soft computing models serve the purpose (Lin et al., 2012). Soft computing models, such as artificial neural networks, support vector machine and extreme learning machine give better predictions with high accuracy (Lee, 2009). Most of the soft computing models can handle nonlinear relations through relevant market features with less statistical assumptions (Atsalakis & Valavanis, 2009). Weng et al. (2017) find that soft computing models are appropriate for developing rules when using with a rich knowledge database. Zhong and Enke (2017) conclude a trading strategy based on classification models yield high returns than the benchmark T-bill strategy. Liang et al. (2009) find that the non-parametric model outperforms parametric models in financial market forecasting. As soft computing models are self-adaptive and are more tolerant of imprecision (Cheng & Wei, 2014). Hence, we investigate whether these findings hold for trading signals prediction of Pakistan stock exchange or not.

3. METHOD

3.1 DATA

We use the Pakistan Stock Exchange (KSE-100 index) daily quotes from 7/02/1997 to 18/07/2018 with the total observations 5168. The data set is bifurcated into a training dataset and testing dataset. The training dataset has 3764 observations cover 70% of total observations and the testing set has 1404 observations include the remaining 30% of total observation.

3.2 TARGET VARIABLE "T"

The target variable T (Equation 1) gives a holistic picture of stock prices by incorporating overall dynamics in the following days. This cannot be merely done through price movements, therefore, the target variable is defined as the sum of all variation above an absolute value of target margin $x\%$ (Torgo, 2011)

$$T = \sum (v \in V_i : v > x\% \vee v < -x\%) \tag{1}$$

where V_i (Equation 2) is k percentage variation of current close price and the following k days price average.

$$V_i = \left\{ \frac{\bar{P}_{i+j} - C_i}{C_i} \right\}_{j=1}^{k} \tag{2}$$

and the daily average price is computed as Equation 3.

$$\bar{P}_i = \frac{C_i + H_i + L_i}{3} \tag{3}$$

$C_i, H_i \; and \; L_i$ are close, high and low prices for the day i respectively.

The target variable is T value and develops a model that predicts this value by using the feature's information that gives three trading actions (sell, hold and buy), this transformation is carried out using the values in Equation 4.

$$signal = \begin{cases} sell\ if\ T < -0.1 \\ hold\ if\ -0.1 \le T \le 0.1 \\ buy\ if\ T > 0.1 \end{cases} \tag{4}$$

The threshold 0.1 and -0.1 are heuristic and any other value can be used. However, the value 0.1 is the average of last four days average price that is 2.5% higher than the current close (4x0.025=0.1) The only care should be taken is that very high values will generate fewer signals and very small values let an investor trade on minor variation, incurring higher risk (Torgo, 2014).

3.3 INPUT FEATURES

We use fifteen features (Table 1) as the input of the soft computing model. These features are selected from the list of features available in the Technical trading rule package (Ulrich et al., 2018) through a random forest technique (Figure 1).

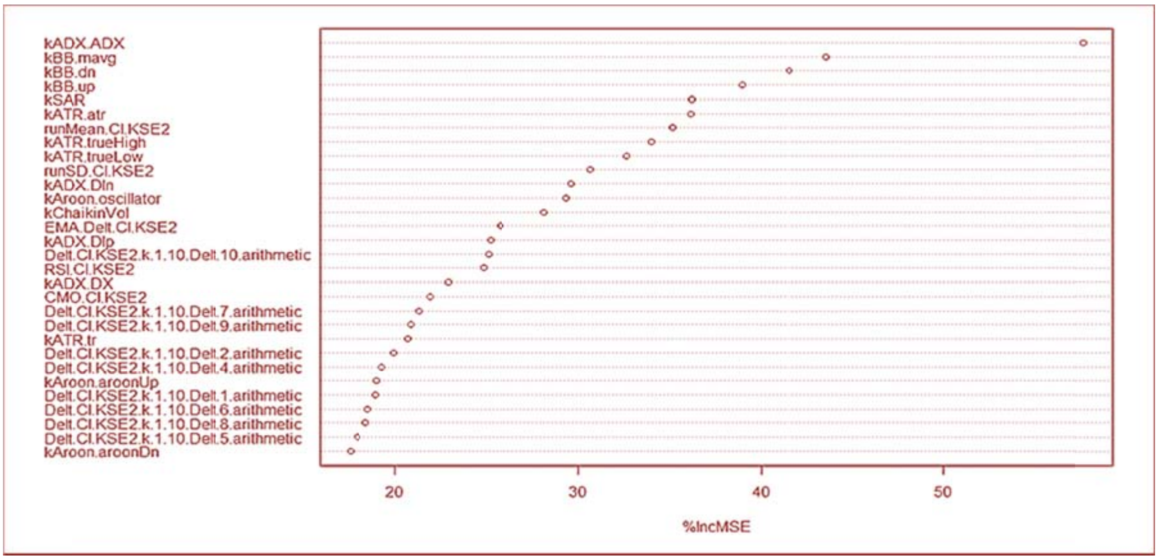

Figure 1: Variable importance according to the random forest.

Table 1: List of Selected Technical Indicators

Sr. No.	Selected Technical Indicators
1	Average True Range
2	Average True Range High
3	Average True Range Low
4	ATR tr.
5	ADX (Welles Wilder's Directional Movement Index)
6	ADX Dip
7	ADX Din
8	ADX DX
9	Bollinger Bands Down
10	Bollinger Bands Moving Average
11	Bollinger Band
12	Bollinger Band Up
13	SAR (Parabolic Stop and Reverse)
14	Run Mean
15	Run Standard Deviation

3.4 METHODOLOGY

This section explains both the traditional and soft computing models used for prediction analysis.

3.4.1 TRADITIONAL MODELS

We first explain the simple moving average, centered moving average, right-aligned moving average, exponential weighted moving average then move to autoregressive moving average model.

Simple Moving Average (SMA) is the average of *n* prices (Equation 5) where each observation is given equal weight. If we take difference of the two of SMA values at times *t* and *t-1*, it gives Equation 6 and recursive form become as Equation 7 which improves the computational speed of SMA in real-life scenarios (Zakamulin, 2017) by reducing total *n* operations involve in Equation 5 to just three mathematical operations (Equation 8) irrespective of the length of averaging window length.

$$SMA_t(n) = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i} \tag{5}$$

$$SMA_t(n) - SMA_{t-1}(n) = \frac{P_t - P_{t-n}}{n} \tag{6}$$

$$SMA_t(n) = SMA_{t-1}(n) + \frac{P_t - P_{t-n}}{n} \tag{7}$$

The Herfindahl index of SMA (Rhoades, 1993) equals $\frac{1}{n}$; hence defining the smoothness of SMA(n) as $(\frac{1}{n})^{-1} = n$. The increase in the average window length not only increases its smoothness but also increases the average lag time of SMA which is a linear function of smoothness (Equation 9).

$$Lag\ time(SMA_n) = \frac{\sum_{i=1}^{n-1} i}{\sum_{i=0}^{n-1} i} = \frac{n-1}{2} \tag{8}$$

$$Lag\ time(SMA_n) = \frac{1}{2} \times Smoothness(SMA_n) - \frac{1}{2} \tag{9}$$

It is quite easy to find a trend and breakthroughs in a trend by looking on historical data considering time series data of P_t as a combination of trend T_t and an irregular component called as noise I_t. Then, an additive model can be written as Equation 10. Noise is a short-lived variation around the trend eliminated through centered moving average or other smoothening tools. Any average of time series data is computed using a fixed window length that rolls through time. This window length is called an averaging period. In case of centered moving average if *n* is the window length then it consists of a center and two halves of length *k* such that $n = 2k + 1$. Since centered moving average at time *t* (equation11) removes noise so the value of CMA is the value of trend in the given time series data. The window length *n* is selected keeping in mind the elimination of noise (Equation 12)

$$P_t = T_t + I_t \tag{10}$$

$$CMA_t = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i} \tag{11}$$

*Corresponding author (Faheem Aslam). Tel: +92-321-5063253. Email: fahimparacha@gmail.com

$$\mathrm{Tt} = CMA_t \text{ (n)} \tag{12}$$

In real life, an analyst is interested in the forecasting of time series *t* + 1 given the historical data series *t* (Equation 13).

$$RMA_t = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i} \tag{13}$$

Mathematical comparison of centered moving average and right-aligned moving average (RMA) at a given time *t* leads to the conclusion that RMA_t is equal to CMA_{t-k} (Equation 11). In fact, RMA is a lagged version of CMA given the same window length and shares the same properties of CMA. Particularly, RMA with longer window length is better at removing noise from the data series but this comes with longer lag time. The lag time is given as Equation 15

$$RMA_t(n) = CMA_{t-k}(n) \tag{14}$$

$$lagtime = k - \frac{n-1}{2} \tag{15}$$

However, the linearly weighted moving average has the drawback that it assigns the same weight to all observations ignoring the importance of recent observation in future prediction. This problem is addressed by exponential moving average (EMA), using the concept of exponential factor λ (Equation 16). The value of λ can be greater than zero and less than or equal to 1.

$$EMA_t(\lambda, n) = \frac{\sum_{i=0}^{n-1} \lambda^i P_{t-i}}{\sum_{i=0}^{n-1} \lambda^i} \tag{16}$$

Autoregressive Integrated Moving Average (ARIMA) is the most commonly used forecasting model amongst traditional forecasting models. It is a generalized version of ARMA (autoregressive moving average) when used for differenced data rather than original data series. Orders of AR part *(p)*, the difference *(d)* and MA *(q)* part specify the ARMA model and the model is said to be of order (*p,d,q*). However, AR and MA are different models for stationary time series and ARMA (and ARIMA) is a hybrid form of these two models for a better fit. The steps of building ARIMA models are explained as follows.

Auto Regression (AR) is a class of linear models where the dependent variable is regressed against its own lagged values. If y_t is modeled via the AR process, it is written as Equation 17 similar to simple linear regression. It has an intercept like term (δ), regressors y_{t-i}, and parameters $\emptyset_{t-i}$ an error term ε_t. The only special thing is that regressors are the dependent variable's own lagged terms. If lag up to *p* is included in the model, the AR process is said to be of order *p*.

$$y_t = \delta + \emptyset y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + \varepsilon_t \tag{17}$$

Moving Average (MA) is another class of linear models. In MA, the output or the variable of interest is modeled via its own imperfectly predicted values of current and previous times. It can be written in terms of error terms as Equation 18.

$$y_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{q-1} + \varepsilon_t \tag{18}$$

Again, it has a form similar to classic linear regression. The regressors are the imperfections (errors) in predicting previous terms. Here the model is specified with a positive sign for the parameters. It is not uncommon where we have a negative sign for the parameters. The model above include errors for q lags and said to have an order of q. In the case of differenced data ARIMA *(p,d,q)* become ARMA*(p,q)* having the mathematical form as Equation 19.

$$y_t = \delta + \sum_{i=1}^{p} \emptyset_i y_t + \sum_{j=1}^{q} \emptyset_j \varepsilon_{t-j} + \varepsilon_t \tag{19}$$

3.4.2 SOFT COMPUTING MODELS

Artificial Neural Network (ANN) is a nonlinear regression technique and popularly used for stock market prediction (Zhiqiang et al., 2013). This field is tracked back to (McCulloch & Pitts, 1943) mathematical function perceived as a model of biological neural network. This model of the neural network comprises three components: weighted inputs that are like synapses in the human brain. Adder- The summation of all input signals corresponds to the neuron membrane which assembles all electrical charges. Activation function- determines if a neuron has an action potential for a specific set of inputs (Algorithm 1).

ALGORITHM 1 Artificial Neural Network

1. Start with small random weights
2. Input the data set
3. For forwarding phase: hidden layer compute activation function of each neuron then calculate the activation function of output
4. For the backward phase: calculate error at the output and at the hidden layers then update the weights of both layers
5. For recall apply the forward phase described in step 3

When it comes to the interpretation of ANN it's more like a black box. Therefore, Support vector machines (SVMs) apply the concept of margin to solve the problem of ANN. SVM easily separates the data by mapping it into high dimensions (Boser et al., 1992). By aiming to maximize the size of margin that classifies the objects without any point lying inside.

ALGORITHM 2 Support Vector Machine

1. Begin with an input data set
2. Classify the data set
3. Apply SVM with a different kernel function
4. Specify hyperplane
5. Repeat step 3 if obtained accuracy is not obtained

The learning speed of ANN and SVM is slower than what is required because of slow gradient-based learning algorithms. Whereas the Extreme Learning Machine (ELM) randomly selects hidden nodes and analytically finds the output weights (Algorithm 3).

ALGORITHM 3 Extreme Learning Machine
1. With the training set and activation function and hidden neuron number 2. Assign random input weights and bias 3. Compute hidden layer output matrix 4. Compute the output weights

4. RESULT AND DISCUSSION

SMA is computed using sma function () in R package smooth (Tukey) and forecast function () in R package forecast (Hyndman et al., 2019). By default, SMA order selection *(h)* is based on AIC and returns the model with the lowest value. Here the order of SMA is 12 giving a good fit shown in purple colored line with AIC value of -3151.602. The forecast is done for the next 10 days represented after a red cutting line (Figure 2). The forecast trajectory of SMA (12) is not just a straight line. This is because the actual values are used in the construction of point forecasts up to h=12 with a 95% prediction interval.

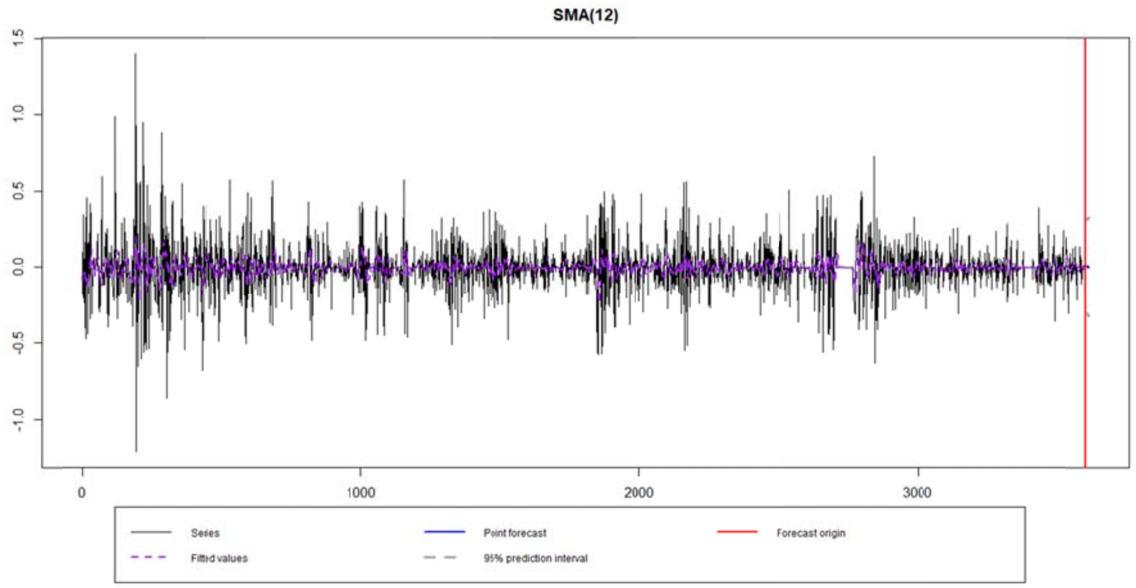

Figure 2: Simple Moving Average.

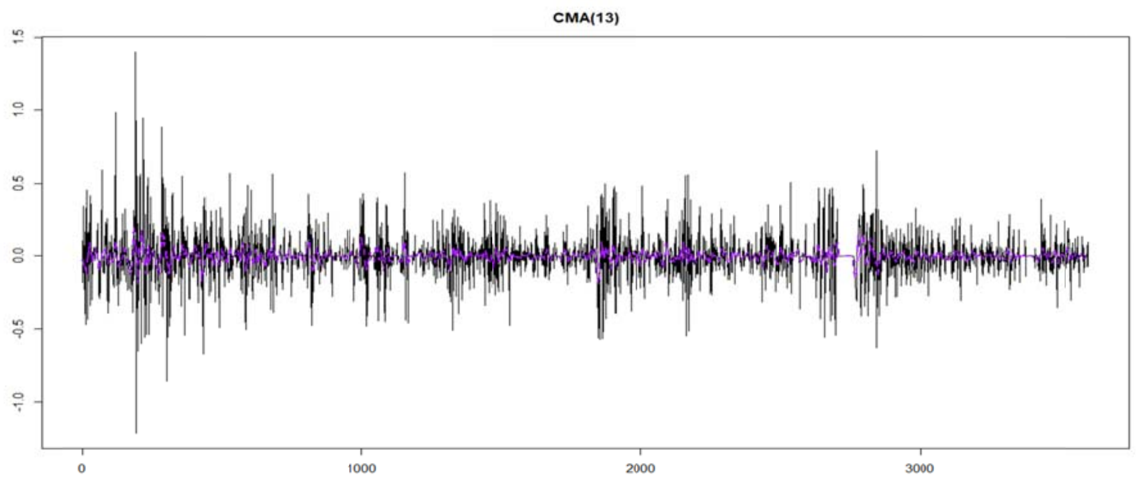

Figure 3: Centered Moving Average.

Centered moving average is computed in R deploying function cma () in package Smooth (Svetunkov, 2017) with the argument order equals to "null" to find order based on AIC. In this analysis CMA at order 13 with AIC value, -3896.784 is computed. Our result reveals the basic property of a centered moving average that is irrespective of window length, CMA closely follows real data pattern (Figure 3).

The right-aligned moving average is computed by using function rollmean () with argument align "right" in R package zoo (Shah, Zeileis, & Grothendieck, 2005). The actual data series is in black line and the right aligned moving average is shown by the orange line and the gray line is the prediction interval (Figure 4). Overall, the RMA fits well with the pattern of real data. Figure 5 shows a magnified version of the forecasted region where the dark shaded area is 80% prediction interval and the light shaded area is 95% prediction interval and the predicted blue line lies within the bounds of the prediction interval.

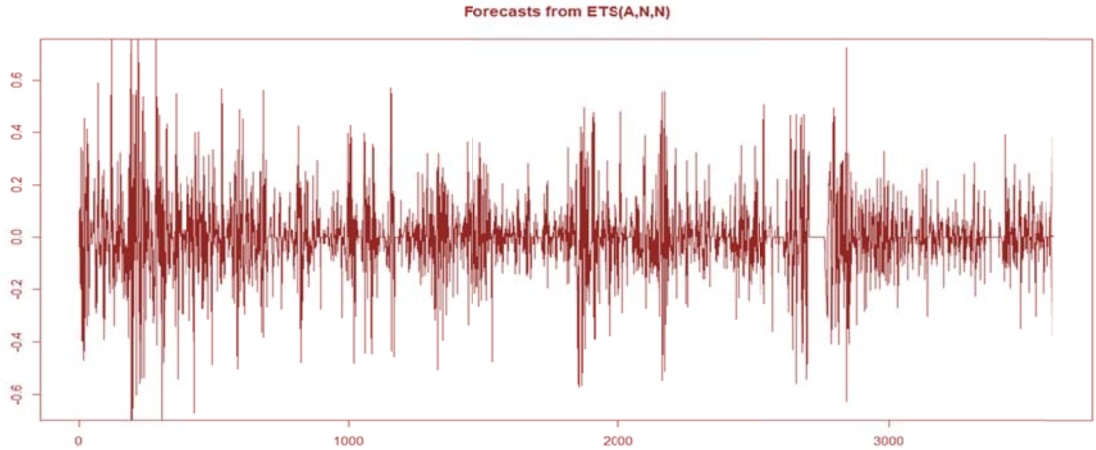

Figure 4: RMA and Trading Signals Data.

We estimate exponential moving average by own code with the lambda optimized at 0.1. The method of least square is used to find the optimal value of λ for which the sum of squared errors (SSE) is minimized (Čisar & Čisar, 2011). Table 2, there is an increasing trend in SSE values with the higher values of lambda. Whereas most of the packages use lambda value as 0.2 (Lucas & Saccucci, 1990). Figure 6 illustrates EMA and actual dataset, the black line is the actual data and the orange line is EMA whereas the gray line is the predicted interval and EMA is following the actual data set.

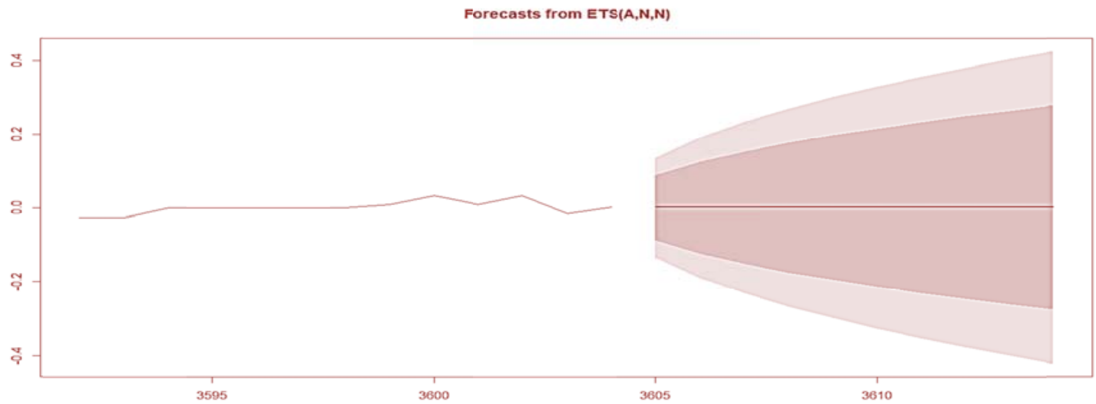

Figure 5: Magnified Graph of RMA Forecasted Region.

Table 2: The optimized value of Lambda

λ	SSE
0.1	0.04432
0.2	0.05035
0.5	0.05146
0.7	0.05157
0.9	0.05157

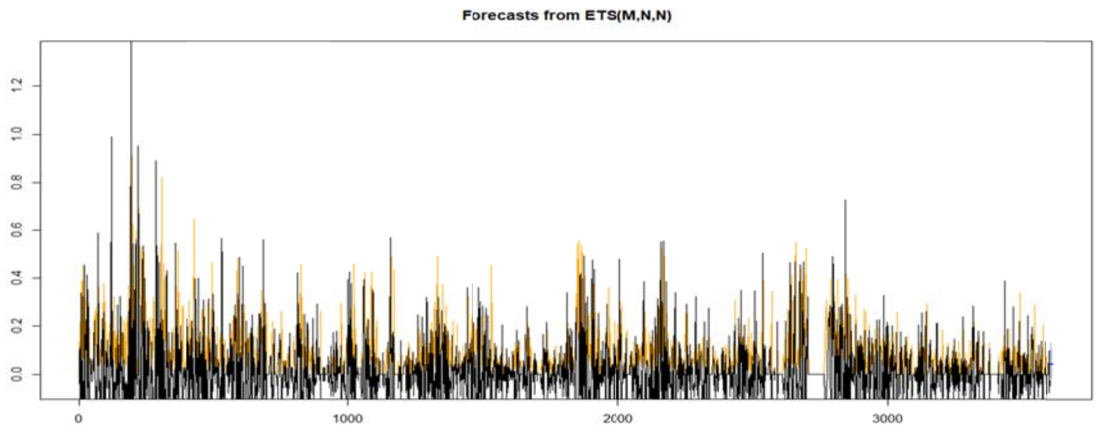

Figure 6: EMA and Actual Data Set.

Three major steps are followed to make an ARIMA forecast model: (1) Model specification (2) Parameter estimation (3) Diagnostics and potential improvement. We first check the stationarity of the data by taking differences and keep taking the difference until the series becomes stationary that number is *d.* There are different tools to detect stationarity. Like visual observation of the data plot and autocorrelation. After determining *d*, we can utilize sample PACF (partial autocorrelation) to get the AR order *p*. The lag at which PACF cuts off is the order of AR and the lag at which ACF (autocorrelation) cut off is the order *q* of MA. The trading signal data was not level stationary but its first difference stationary as shown in Figures 7 and 8. Next Figures 9 and 10 show the ACF and PACF plot of a simulated time series. For the series, the ACF cuts off after lag 1 and PACF also cuts off after lag 1. So three potential *(p, q)* specification would be (1, 0), (0, 1) and (1,1).

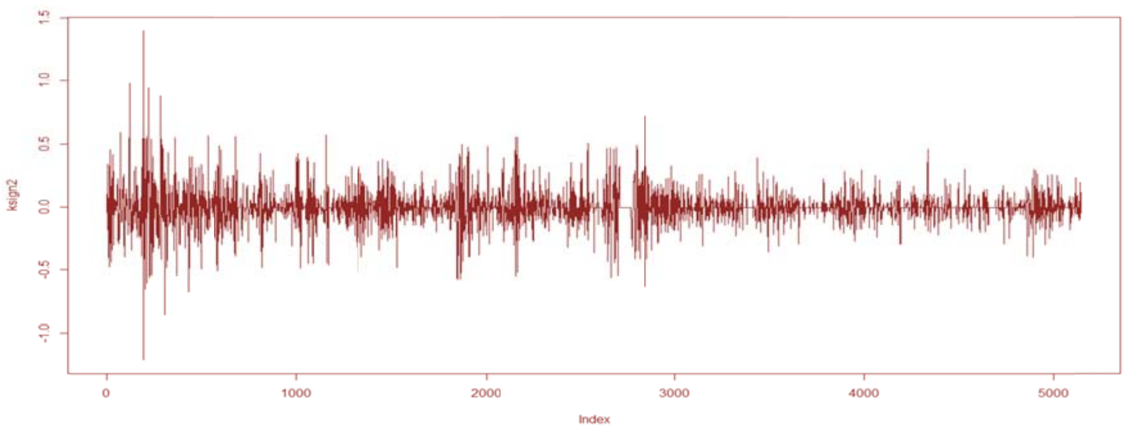

Figure 7: Graphical Representation of First Differenced series of Trading Signals.

Since the ACF plot of first difference is better than level stationary graph so the series is stationary at first difference. We further investigate it with unit root test KPSS in R package urca (Pfaff et al., 2016). The value of the test statistic is 0.0039 so the series is first differenced stationary. Once we determine the orders, the next step is to find the parameters. Commonly used techniques are

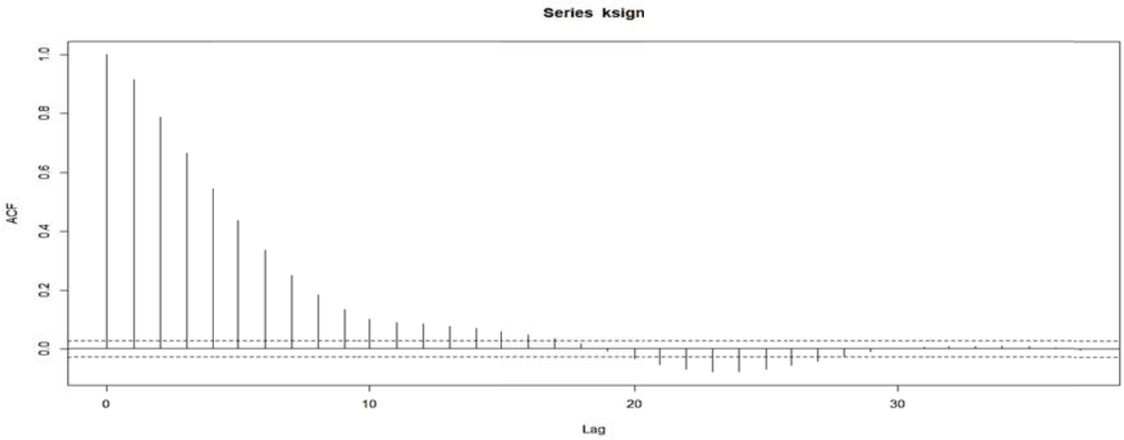

Figure 8: Autocorrelation Plot of Original Trading Signals Series.

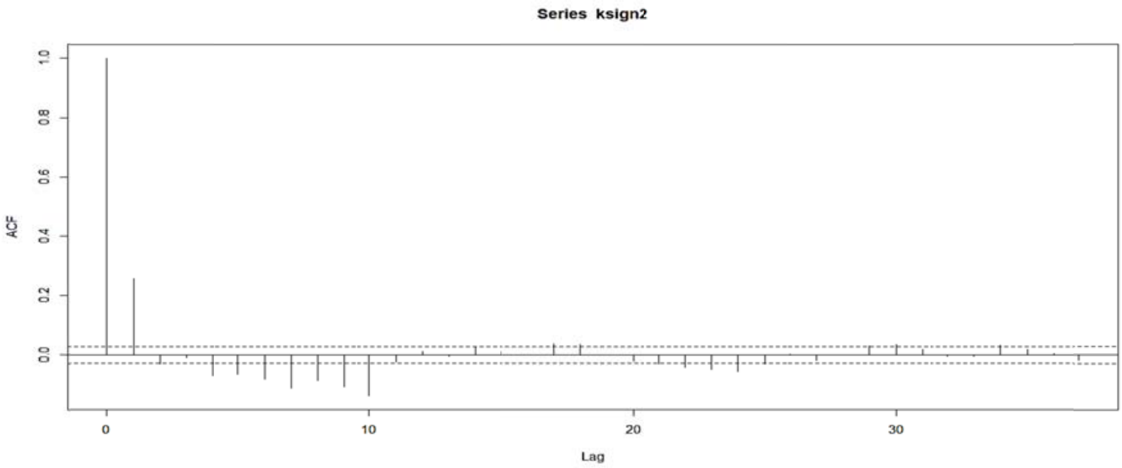

Figure 9: Autocorrelation Plot of First Differenced Trading Signals Series.

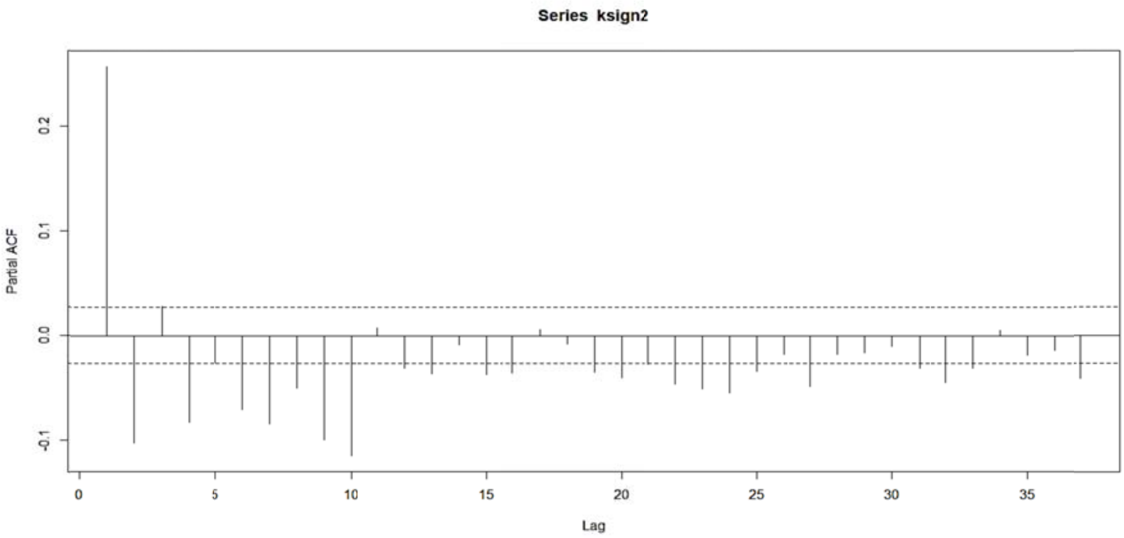

Figure 10: Partial Autocorrelation Plot of First Differenced Trading Signals Series

least-square, maximum likelihood, method of moments. R package forecast (Hyndman & Khandakar, 2007) use the maximum likelihood method. After developing the model, we check the model adequacy through diagnostics.

The original plot shows clear nonstationarity. The ACF does not cuts off, rather it shows a slow decrease. On the other hand, the first difference data looks much stationery. The ACF cuts off after some lags. We can perform another formal test to test the stationarity of the differenced data. But here from the plot of the original data and the ACF plot provide a good indication that the differenced data is stationary. Since it took one differencing to get stationarity, here $d=1$. The values of p and q, which is the order of AR and MA part is determined through the plots of ACF and PACF that makes three models, ARIMA(0,1,1), ARIMA(1,1,0) and ARIMA(1,1,1).

4.1 MODEL BUILDING AND DIAGNOSTIC

We develop three ARIMA models with orders as proposed earlier. We bifurcate the data into a 70:30 ratio. The total number of observations is 5151, the training data set has 3606 observations and the testing dataset has the remaining 1545 observations. For estimation purposes, R package Forecast (Hyndman & Khandakar, 2007) and package Metrics (Hamner et al., 2018) are deployed.

In Figure 11, the forecasts for the next 10 days are plotted as a blue line, where 80% prediction interval is shown in dark shaded area and 95% prediction interval as a light shaded area. If there is no autocorrelation between forecast errors then the model cannot be improved. In Figure 12, spike 2, 4, 7 and then 10 is out of the significance bounds. We carry out a unit root test where KPSS has the value of the test statistic 0.0287 confirming the evidence of non-zero correlation. The forecast errors in Figure 13 represents a more or less normal distribution.

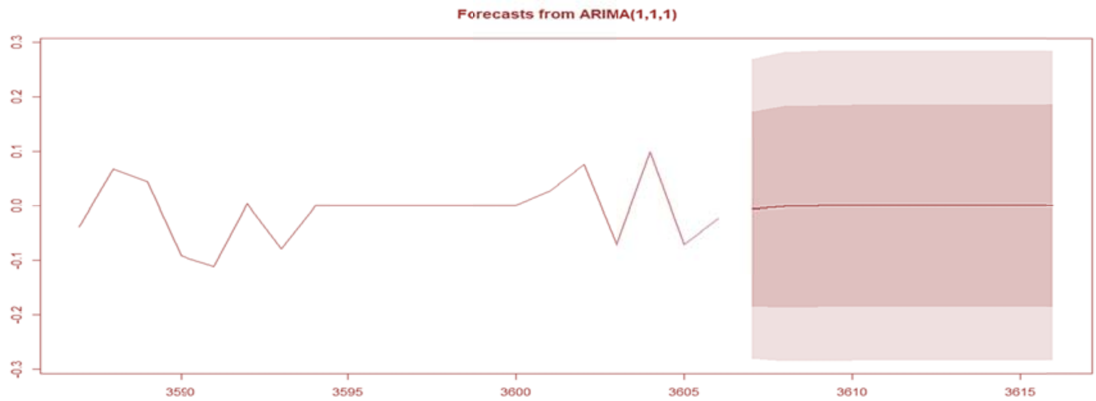

Figure 11: Ten days Forecast.

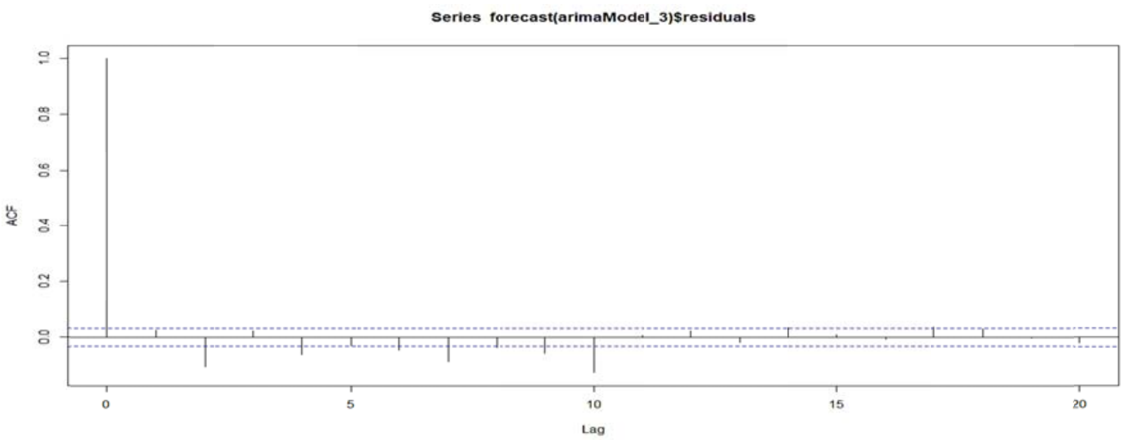

Figure 12: ACF Plot of Residual ARIMA (1, 1, 1).

The model selection among the above tested traditional models is based on AIC values. Table 3 shows that all AIC values are in negative means that the likelihood at the maximum was > 1 and algebraically lower AIC is selected rather than the absolute value of AIC which in our case is ARMA (1,1,1) with AIC value -3923.65 and precision value 0.1333. Right-aligned averages and the exponentially weighted average has no reported value of AIC.

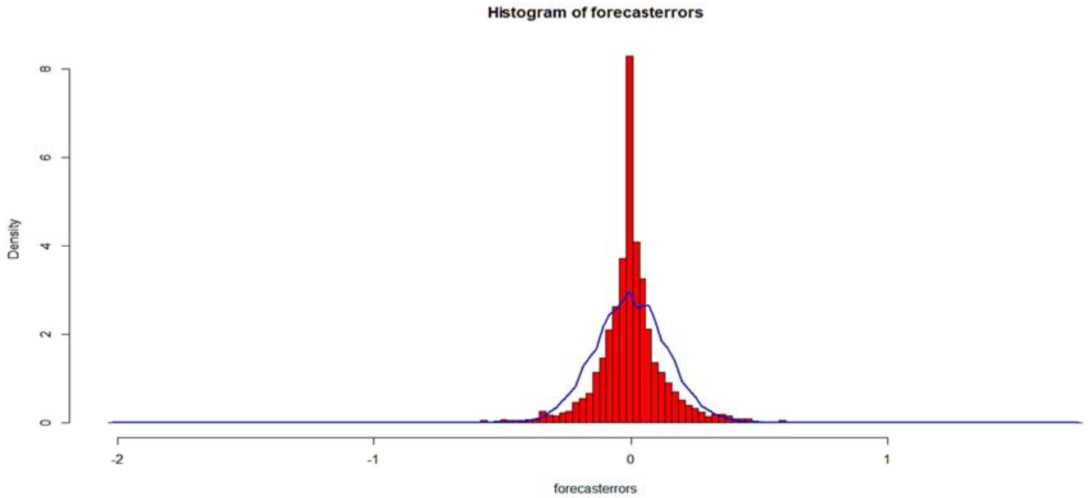

Figure 13: Histogram of Residual ARIMA (1, 1, 1).

Table 3: Comparison of Traditional Model

Model	AIC
SMA	-3151.602
CMA	-3896.784
RMA	Null
EMA	Null
ARMA(0,1,1)	-3680.29
ARMA(1,1,0)	-2611.42
ARMA(1,1,1)	-3923.65

Table 4: Comparison of Soft Computing Model.

Technique	Activation function	Precision Score
ANN	ReLU	0.6311
	Tanh	0.5836
	Sigmoid	0.5945
SVM	Polynomial	0.5265
	Radial Basis	0.4830
	Tanh	0.4950
ELM	Sigmoid	0.5000
	ReLU	0.4285
	Tanh	0.4410

The ANN is applied in package nnet of R. The feed-forward ANN model has three layers: input, hidden and output. ANN model utilized in this study employs the sigmoid, tanh and ReLU activation function. The network has 15 input neurons that correspondents to the 15 selected input technical indicators through the random forest (Table 1). The output layer has a predicted signal. All parameters are fixed for this ANN. The network is trained on 70% data and tested on the remaining 30% of the data set. The highest precision score is 0.6311 with activation function ReLU. SVM is estimated against three activation functions which are, polynomial, tanh and radial basis with the highest precision value 0.5265 in polynomial activation function. ELM is applied using function elm_train() and elm_predict(). The number of hidden nodes is 10, three activation functions ReLU, sigmoid and tanh are used and the input layer has 15 selected input technical indicators. The highest precision value is 0.5000 with activation function sigmoid. The soft computing models ANN gives the highest precisions score with activation function ReLU for KSE-100 trading signal data as shown in Table 4. The analysis of traditional econometric techniques only ARMA (1,1,1) is the best model but the precision score is quite low (Table 5). These results are in line with previous studies (Hsu, Lessmann, Sung, Ma, & Johnson, 2016; Rojas et al., 2008) which also assert the better predictive performance of soft computing models over traditional models.

Table 5: Comparison of Best Traditional Model and Best Soft Computing Model

Model Type	Model	Precision Score
Traditional Model	ARMA(1,1,1)	0.1333
Soft Computing Model	ANN	0.6311

5. CONCLUSION

This study brings forward the unsettled debate in the literature about modeling techniques for trading signals prediction. We cover a vast range of models from moving averages to autoregressive models and then soft computing models. Our detailed analysis finds the best of the traditional models and best of soft computing models but soft computing models perform better than traditional forecasting models for trading signals prediction. These findings are important for traders who can forecast trading signals on the basis of the soft computing model rather than the traditional model. Our analysis also confirms the non-linear behavior of time series financial data which can be better handled through the soft computing model.

6. AVAILABILITY OF DATA AND MATERIAL

Data can be made available by contacting the corresponding authors

7. REFERENCES

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.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). *A training algorithm for optimal margin classifiers.* Paper presented at the Proceedings of the fifth annual workshop on Computational learning theory.

Brockwell, P. J., & Davis, R. A. (2009). Time Series: Theory and Methods, (Springer Series in Statistics).

Cheng, C.-H., & Wei, L.-Y. (2014). A novel time-series model based on empirical mode decomposition for forecasting TAIEX. *Economic Modelling, 36*, 136-141.

Čisar, P., & Čisar, S. M. (2011). Optimization methods of EWMA statistics. *Acta Polytechnica Hungarica, 8*(5), 73-87.

Cryer, J. D., & Chan, K.-S. (2008). Time series regression models. *Time series analysis: with applications in R*, 249-276.

Fama, E. F. (1965). The behavior of stock-market prices. *The journal of Business, 38*(1), 34-105.

Hamner, B., Frasco, M., & LeDell, E. (2018). Package 'Metrics'.

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.

Hu, Y., Feng, B., Zhang, X., Ngai, E., & Liu, M. (2015). Stock trading rule discovery with an evolutionary trend following model. *Expert systems with Applications, 42*(1), 212-222.

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, J.Z., Huang, W., & Ni, J. (2019). Predicting bitcoin returns using high-dimensional technical indicators. *The Journal of Finance and Data Science, 5*(3), 140-155.

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

Hyndman, R. J., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., . . . Razbash, S. (2019). Package 'forecast'. *Online] https://cran. r-project. org/web/packages/forecast/forecast. pdf.*

Hyndman, R. J., & Khandakar, Y. (2007). *Automatic time series for forecasting: the forecast package for R*: Monash University, Department of Econometrics and Business Statistics ….

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.

Kumar, D. A., & Murugan, S. (2013). *Performance analysis of Indian stock market index using neural network time series model.* Paper presented at the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering.

Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems, 37*(4), 567-581.

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert systems with Applications, 36*(8), 10896-10904.

Lin, C.-S., Chiu, S.-H., & Lin, T.-Y. (2012). Empirical mode decomposition–based least squares support vector regression for foreign exchange rate forecasting. *Economic Modelling, 29*(6), 2583-2590.

Lucas, J. M., & Saccucci, M. S. (1990). Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics, 32*(1), 1-12.

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.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics, 5*(4), 115-133.

Moradi, S., & Rafiei, F. M. (2019). A dynamic credit risk assessment model with data mining techniques: evidence from Iranian banks. *Financial Innovation, 5*(1), 15.

Murphy, J. J. (1986). Technical Analysis of the Futures Markets, New York Institute of Finance. *Englewood Cliffs, NJ.*

Oliveira, A. L., & Meira, S. R. (2006). Detecting novelties in time series through neural networks forecasting with robust confidence intervals. *Neurocomputing, 70*(1-3), 79-92.

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.

Pfaff, B., Zivot, E., Stigler, M., & Pfaff, M. B. (2016). Package 'urca'. *Unit root and cointegration tests for time series data. R package version*, 1.2-6.

Rhoades, S. A. (1993). The herfindahl-hirschman index. *Fed. Res. Bull., 79*, 188.

Rojas, I., Valenzuela, O., Rojas, F., Guillén, A., Herrera, L. J., Pomares, H., . . .Pasadas, M. (2008). Soft-computing techniques and ARMA model for time series prediction. *Neurocomputing, 71*(4-6), 519-537.

Shah, A., Zeileis, A., & Grothendieck, G. (2005). zoo Quick Reference. *Package vignette*.

Svetunkov, I. (2017). Statistical models underlying functions of'smooth package for R.

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

Teixeira, L. A., & De Oliveira, A. L. I. (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert systems with Applications, 37*(10), 6885-6890.

Torgo, L. (2011). *Data mining with R: learning with case studies*: Chapman and Hall/CRC.

Torgo, L. (2014). An infrastructure for performance estimation and experimental comparison of predictive models in r. *arXiv preprint arXiv:1412.0436*.

Tsinaslanidis, P. E., & Kugiumtzis, D. (2014). A prediction scheme using perceptually important points and dynamic time warping. *Expert systems with Applications, 41*(15), 6848-6860.

Tukey, J. Exploratory data analysis 1977 Reading. *MA Addison-Wesley*.

Ulrich, J., Ulrich, M. J., & RUnit, S. (2018). Package 'TTR'.

Vanstone, B., & Finnie, G. (2009). An empirical methodology for developing stockmarket trading systems using artificial neural networks. *Expert systems with Applications, 36*(3), 6668-6680.

Wang, B., Huang, H., & Wang, X. (2012). A novel text mining approach to financial time series forecasting. *Neurocomputing, 83*, 136-145.

Wang, J.-Z., Wang, J.-J., Zhang, Z.-G., & Guo, S.-P. (2011). Forecasting stock indices with back propagation neural network. *Expert systems with Applications, 38*(11), 14346-14355.

Weng, B., Ahmed, M. A., & Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. *Expert systems with Applications, 79*, 153-163.

Zakamulin, V. (2017). *Market Timing with Moving Averages: The Anatomy and Performance of Trading Rules*: Springer.

Żbikowski, 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.

Zhiqiang, G., Huaiqing, W., & Quan, L. (2013). Financial time series forecasting using LPP and SVM optimized by PSO. *Soft Computing, 17*(5), 805-818.

Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert systems with Applications, 67*, 126-139.

Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation, 5*(1), 4.

Beenish Bashir is a PhD scholar at Department of Management Sciences, COMSATS University Islamabad. She was a visiting research student at University of Côte d'Azur, France. Her research interest are Financial Markets, Financial Risk Management, Machine Learning and Data Science.

Dr. Faheem Aslam is an Assistant Professor at Department of Management Sciences, COMSATS University Islamabad. He earned his Master's and Ph.D. Degrees from Hanyang University Business School, Seoul, South Korea. His research interests are Financial Analytics, Data Mining, Artificial Intelligence and Data Science.