

COMPARING ARIMA, MLP AND HYBRID TIME SERIES MODELS WITH APPLICATION ON CEMENT STOCK PRICES: A CASE STUDY OF SAUDI ARABIA

¹Abdullah M. Almarashi, ¹Rana M. Alqurashi, ¹Shouq M. Aldawood, ¹Shatha M. Alnajjar, ¹Fatimah A. Zahrani, ¹Reham S. Jamil, ¹Rafal A. Fira, ¹Alhanouf A. Almalki, ¹Fatima H. Almalki, ¹Omaymah F. Alyamani, ¹Ghadeer F. Naseeb, ¹Hibah I. Alnashri, ¹Khulud A. Algarni, ^{2*}Khushnoor Khan.

¹Department of Statistics. Faculty of Science, King Abdulaziz University, 21589, Jeddah, Saudi Arabia.

^{2*}Research Scholar, Department of Business Studies, Faculty of Management Sciences, Muslim Youth University,44000, Islamabad, Pakistan

Email:Khushnoorkhan64@gmail.com

ABSTRACT: *The present study compares the predictive ability of the Auto-Regressive Integrated Moving Average (ARIMA) model, Artificial Neural Network technique –Multi-Layer Perception (MLP) and Hybrid Models technique using stock prices of Yamama Saudi Cement Company. Data for the study was downloaded from Yahoo Finance Website for the period 20th September 2022 to 20th September 2023 (249 days). Root Mean Squared Errors (RMSE) is used to assess the accuracy of the forecasting measures. R software ver 4.3.2 coupled with three libraries **nnfor**, **forecast**, **thief** and **forecastHybrid** were used for coding and graphing the results. For ease of use the relevant R codes used in the present study are attached in Appendix ‘A’ to be used by academia and practitioners. Outcomes of the study revealed that Neural network auto-regression (NNAR) outperformed the other competing models and it hence can be used effectively to predict Cement Stock Prices within the Saudian Context. Limitations of the studied need to be looked into prior to application of the appropriate prediction technique.*

Keywords: Forecasting, ARIMA, MLP, Cement, Root Mean Square, Hybrid.

1. INTRODUCTION:

Time series forecasting plays a crucial role in various domains, from finance to weather prediction. Two widely used approaches for time series forecasting are the Auto-Regressive Integrated Moving Average (ARIMA) model and the Multi-Layer Perceptron (MLP) neural network. The ARIMA model, developed by Box and Jenkins [1], has been a cornerstone in time series forecasting for decades. It combines auto-regression, differencing, and moving average components to capture temporal patterns. Notable studies, such as Granger and Newbold [2], have highlighted the effectiveness of ARIMA in capturing linear dependencies in time series data. In recent years, neural networks, particularly Multi-Layer Perceptrons (MLPs), have gained popularity in time series forecasting due to their ability to capture complex nonlinear relationships. Zhang and Leung [3] demonstrated the potential of MLPs in forecasting financial time series, showcasing their adaptability to diverse patterns.

Several studies have compared the performance of ARIMA and MLP models. While ARIMA is effective in capturing linear trends, Yu and Wang [4] demonstrated that MLPs excel in handling nonlinear patterns, making them suitable for more complex time series data. Zhang [5] explores the combination of ARIMA and neural networks, emphasizing the potential of MLPs in capturing complex time series patterns. Hornik et al [6] establishes the universal approximation property of MLPs, highlighting their capability to approximate any continuous function. The universal approximation theorem supports the use of MLPs for modeling complex relationships in time series data. For in depth study of Neural networks see Ord K., Fildes R. and Kourentzes N. [7].

A hybrid model in time series analysis combines multiple forecasting techniques to enhance prediction accuracy. Typically, it integrates both statistical and machine learning approaches to leverage the strengths of each method. For instance, a common hybrid model might fuse autoregressive integrated moving average (ARIMA) or

exponential smoothing methods with machine learning algorithms like neural networks or decision trees. By blending the strengths of these diverse models, hybrid approaches aim to capture complex patterns and trends present in time series data, offering improved forecasting performance compared to individual methods alone. This flexibility allows the model to adapt to different types of time series patterns, making it a valuable tool in diverse forecasting scenarios. Hyndman & Athanasopoulos, [8] explored various combinations of traditional time series methods, such as ARIMA or exponential smoothing, with advanced machine learning algorithms like neural networks and ensemble techniques to create more robust and accurate hybrid models.

Present study will compare ARIMA, MLP and Hybrid models using easy to follow approach by using libraries of R language, without delving into complex mathematics of neural networks. For illustrative purposes and application of the aforementioned techniques data we will use data of stock prices for Yamama Cement Company, Saudi Arabia. Yamama Saudi Cement Company was established in 1956 by Prince Mohammed Bin Saud Al-Kabeer, mainly to manufacture and trade in cement in Riyadh, the capital of K.S.A. Yamama which is a shareholding company, is regarded as the oldest cement company in the central region and the third of its kind in the kingdom providing credence for using its data for time series analysis.

1.2. Format of the Paper: The rest of the paper is organized as follows: Section 2 presents the methodologies for the traditional and machine learning models to be used in the current study, while Section 3 presents the results of data analysis. Section 4 provides discussion/conclusion/limitations and future directions on the topic.

2. Methods

Box-Jenkins methodology. The Box-Jenkins method (ARIMA) is one of the most widely used time series forecasting methods in practice. Box and Jenkins (1970 & 1976)[9] developed the BJ methodology. There are three

primary stages in building a Box-Jenkins time series model: *model identification*, *model estimation* and *model validation*. For the present study, we will follow BJ methodology.

ARIMA. Forecasting non-stationary time series is a challenging task that requires specialized techniques and models to handle the dynamic nature of the data. Before delving into non-stationary time series, it is crucial to grasp the concept of stationarity. A stationary time series is one whose statistical properties do not change over time. These properties include the mean, variance, and autocorrelation. In essence, a stationary time series has a constant mean and variance and exhibits consistent patterns and trends throughout its observations. Stationary data is easier to model and forecast since it follows stable statistical properties. The method of maximum likelihood is used in estimating the parameters. To select the most appropriate predictive model, Akaike information criterion (AIC) and Schwarz information criterion (SIC) will be compared. All analyses are carried out using R software

Terminology in Neural Network and Regression.

Analogous terminology commonly used in neural network and regression is shown in table

Table: Analogous Terminology used in NN and Regression

Neural Networks	Regression
Input	Predictor or Explanatory Variable
Output	Response Variable
Weights	Coefficients or Parameters
Bias	Intercept
Training	Fit a model or estimate the parameters
Backpropagation	Minimize the error

MLP: In time series analysis, Multilayer Perceptron (MLP) is utilized as a predictive model to forecast future values based on historical time-ordered data. The input layer of the MLP corresponds to the temporal features, while one or more hidden layers' capture complex temporal dependencies. The output layer produces the predicted values for future time steps. Training the MLP for time series involves using historical sequences to learn the weights that optimize predictive performance. The backpropagation algorithm is commonly employed, adjusting weights to minimize the error between predicted and actual values. Time series-specific considerations, such as windowing or lagged observations, may be incorporated to ensure the model captures temporal patterns effectively. Regularization techniques like dropout or early stopping can be applied to prevent overfitting in the context of time series data. MLPs are versatile for time series forecasting tasks, accommodating various patterns and trends present in sequential data. For MLP we use R software ver 4.3.2 using two libraries *nnfor* and *forecast* and the relevant codes are attached as per appendix 'A'. For the present study we use a 20% validation set (randomly) sampled to find the best number of hidden nodes. Automatic hidden layers option was used to evaluate from 1 to 10 hidden nodes and picks the best on validation set using MSE.

The key idea behind MLP is to use the historical values of a time series to predict its future values. Basic steps and mathematical concepts involved in using MLPs for time series forecasting are enumerated as follows:

a. **Data Preparation:** Organize the time series data into sequences. Each input sequence contains past values of

the time series, and the corresponding output is the value to be predicted

b. Neural Network Architecture:

- **Input Layer:** The number of neurons in the input layer is determined by the length of the input sequences.
- **Hidden Layer:** One or more hidden layers can be used to capture complex patterns in the time series data. The number of neurons in each hidden layer is a hyper-parameter that can be tuned
- **Output Layer:** The output layer typically has one neuron for regression tasks, providing the predicted value for the next time step.

c. **Activation Functions:** Common activation functions include Sigmoid, Hyperbolic Tangent and ReLU (Rectified Linear Unit) for hidden layers and linear activation, softmax or speaky ReLU for the output layer in regression/classification tasks.

d. **Loss Function:** Mean Squared Error (MSE) is commonly used as the loss function for regression tasks. It measures the average squared difference between the predicted and actual values.

e. **Optimization Algorithms:** Gradient Descent or one of its variants (e.g., Adam, RMSprop) is used to minimize the loss function and update the weights of the network during training.

f. **Training:** The network is trained using historical data. The input sequences and corresponding target values (future values of the time series) are used to update the weights of the network through backpropagation

g. **Validation and Testing:** After training, the model is validated on a separate dataset to ensure it generalizes well to unseen data. Testing is then performed on a different set of data to evaluate the model's performance.

h. **Hyper-parameter Tuning:** Parameters such as the learning rate, the number of hidden layers, and the number of neurons in each layer need to be tuned to optimize the model's performance.

With the availability of fast computing tools most of the functions are automatically handled by the R libraries instantly, but comprehending the underlying mathematics is of utmost importance for academia and practitioners.

In mathematical terms, the forward pass of an MLP can be expressed as:

$$\text{Output} = \sigma(\mathbf{W}_{out} * \sigma(\mathbf{W}_{hidden} * \sigma(\mathbf{W}_{in} * \text{Input} + \mathbf{b}_{in}) + \mathbf{b}_{hidden}) + \mathbf{b}_{out})$$

Where:

- Input is the input time series sequence
- \mathbf{W}_{out} , \mathbf{W}_{hidden} , \mathbf{W}_{in} are the weight matrices for the input, hidden, and output layers
- \mathbf{b}_{in} , \mathbf{b}_{hidden} , \mathbf{b}_{out} are the bias vectors,
- σ represents the activation function

The model is trained to minimize the Mean Squared Error between the predicted output and the actual target values. Training involves adjusting the weights and biases through backpropagation and gradient descent

Data. Data for the present study was downloaded from Yahoo.finance.com for Yamama Cement Company for a period of one year (20th September 2022 to 20th September 2023) for a total of 249 days. The Box-Jenkins models often require a slightly lengthy series to fit well.

Hybrid Models: Hybrid models for time series methodology represent a powerful approach in forecasting

that combines the strengths of multiple forecasting techniques to improve accuracy and reliability. These models integrate different methodologies to address the limitations of individual models and enhance overall predictive performance. Key features of hybrid time series models include:

- a. **Combination of Models:** Integrating diverse forecasting techniques, such as autoregressive integrated moving average (ARIMA), exponential smoothing, and machine learning algorithms, to leverage their respective strengths.
- b. **Ensemble Methods:** Utilizing ensemble techniques like model averaging or stacking to aggregate predictions from multiple models, resulting in a more robust and accurate forecast.
- c. **Adaptability:** Hybrid models can adapt to various types of time series data, incorporating both linear and non-linear patterns, seasonality, and external factors.
- d. **Optimization Strategies:** Employing optimization algorithms to dynamically adjust the weights assigned to different models, ensuring the adaptive nature of the hybrid approach.
- e. **Improved Forecasting Accuracy:** By combining complementary forecasting methods, hybrid models aim

to provide more accurate predictions, particularly in complex and dynamic time series datasets.

These hybrid models offer a flexible and sophisticated methodology for time series forecasting, contributing to better decision-making in various fields such as finance, economics, and operations [10]. For in-depth study of Hybrid models see [11, 12, 13, 14],

3. RESULTS:

Empirical results for the present study will be presented in the following sequence:

- ARIMA
- MLP
- Hybrid Model

3.1. ARIMA: Three stages of ARIMA are studied sequentially as under:

Stage 1: Model Identification: ARIMA (p, d, q) model. The first step in BJ-methodology is identification of the order for p, d, q in ARIMA model. Prerequisite for applying BJ-methodology is to see whether the series is stationary or not. But before looking into the stationarity of the time series we will study the general characteristics of the data

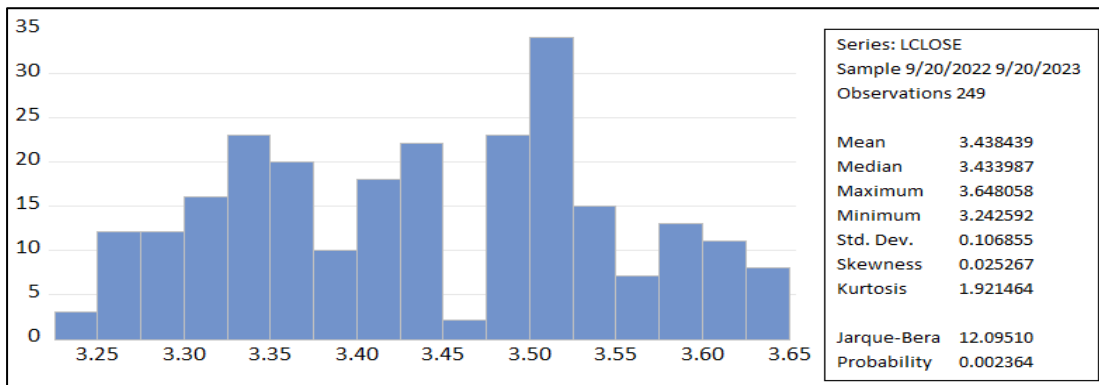
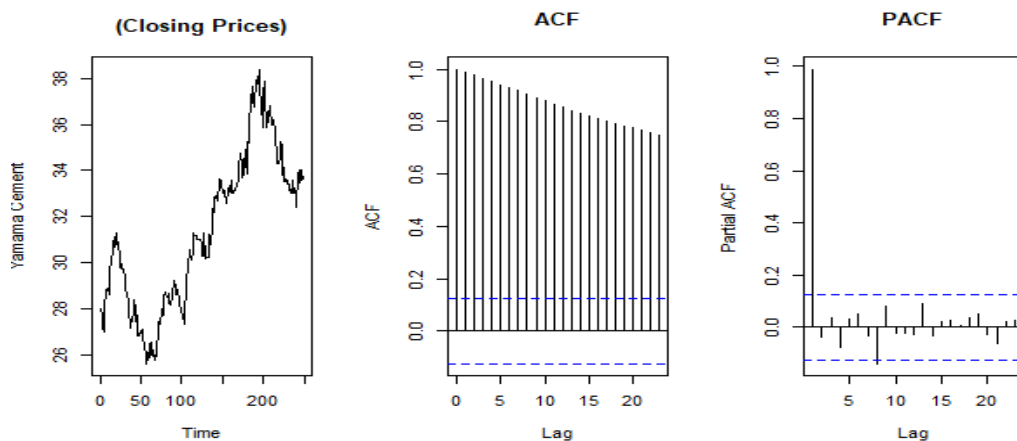


Figure 1: Time Series Graph (original data) for Ln (Closing stock prices) of Yamama Saudi Cement Company

There is some information about the original series than we can extract from right panel of Figure 1, for instance there are 250 observations from 20/9/2022 to 20/9/2023 the peak

is at 38.40 which occurs on 7/6/2023 and the trough is equal to 25.60 which occurs in 12/11/2022

Figure 2: Histogram and Descriptive statistics for Ln (Closing stock prices) of Yamama Saudi Cement Company



For Figure 2 we used log natural of closing prices and the values of Mean, Standard Deviation, Skewness,

Kurtosis, Range, and Jarque-Bera test are discussed as under:

The mean is equal to 3.438439 and skewness is 0.025267 slightly greater than zero, which is positively skewed distribution indicating that an investor may expect frequent (more) small losses and a (few) large gains from the investment. The excess kurtosis is equal to $(1.921464 - 3 = -1.078536)$ and is negative, then the distribution is platykurtic and has short tails, which means that the tail values have less probability of occurrence. Jarque-Bera test statistic is equal to 12.0951, and the p-value is $0.002364 < 0.01$ meaning that the distribution is non-normal. The standard deviation is equal to 0.106855, close to zero indicating that the data points tend to be very close to the mean of the set and the range lies between 3.242592 and 3.648058. Now reverting again to the to To address problem of stationarity we will apply a formal stationarity test as under:

Formal Stationarity Test: We will study the stationary of the time series by using Augmented Dickey-Fuller (ADF) test and Correlogram:

- Hypotheses for the Augmented Dickey-Fuller (ADF) test are:
 $H_0: \theta = 0$ (i.e. the data needs to be differenced to make it stationary).
 $H_1: \theta < 0$ (i.e. the data is stationary and doesn't need to be differenced).

Since $p\text{-value} = 0.7121$ is > 0.05 so H_0 is not rejected which means the time series has a unit root and it is non-stationary. Visually from the ACF and PACF graphs shown in the middle and right panel of Figure 1 it is evident that the series is not stationary. To stationarize the series we difference the series and again check the ADF.

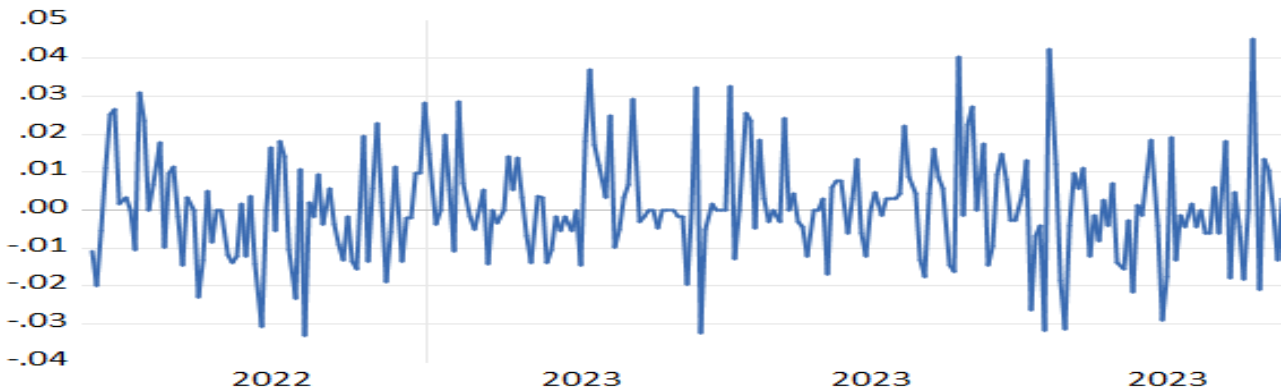


Figure 3: Graph of the Differenced Time Series

Figure 3 shows the series after taking first differences becomes stationary and the formal statistical test of Augmented Dickey-Fuller (ADF) test had $p\text{-value} = .0000 < 0.05$ so, H_0 , will be rejected which indicates the time series does not contain a unit root and it is stationary.

Stage 2: Model Estimation: After attaining stationarity we estimate the parameters of the ARIMA model (p, d, q). Using auto arima function in R software with forecast library ARIMA (1,1,1) is selected as the best model with

minimum model selection criteria AIC and minimum RMSE. The parameters of the models are estimated using Maximum Likelihood Estimation. The AR (1) estimate is 0.4201 and MA (1) estimate is -0.4220. ARIMA (1,1,1) model can be expressed as:

$$y_t - y_{t-1} = a + 0.4201y_{t-1} - 0.4220\varepsilon_{t-1} + \varepsilon_t$$

$$y_t = a + 0.420y_{t-1} - 0.4220\varepsilon_{t-1} + \varepsilon_t + y_{t-1}$$

$$y_t = a + 1.420y_{t-1} - 0.4220\varepsilon_{t-1} + \varepsilon_t$$

RMSE is used to measure forecasting accuracy and for the selected ARIMA (1,1,1) model $RMSE = 0.4522$

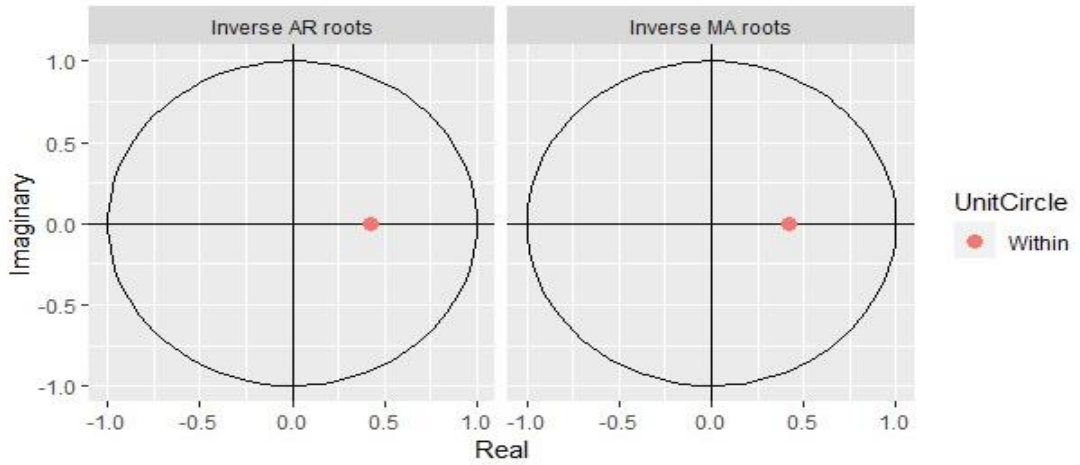


Figure 4: Showing Inverse Characteristic Roots of AR and MA

A causal invertible model should have all the roots outside the unit circle. Equivalently, the inverse roots should lie inside the unit circle. If all roots have modulus less than one and lie inside the unit circle, then the estimated ARMA is stable (stationary) and invertible and therefore will give good estimates. Figure 4 shows that the inverse roots of AR and MA for the current study lie within the unit circle.

Stage 3: Model Validation and Forecasting (Diagnostics): Forecasts for 20 steps ahead are exhibited

along with the original data for ARIMA (1,1,1) are shown in Figure 5. The blue band is 90% confidence band. The last step in the time-series analysis is to look at the diagnostics of the proposed model ARIMA (1, 1, 1). The top panel of Figure 6 shows the residual vs fit plot and the series is quite mean-reverting (stationary) and in the bottom panel, ACF and PACF show that all spikes lie within the 5% band pointing towards the presence of no autocorrelation in the proposed ARIMA (1, 1, 1) model.

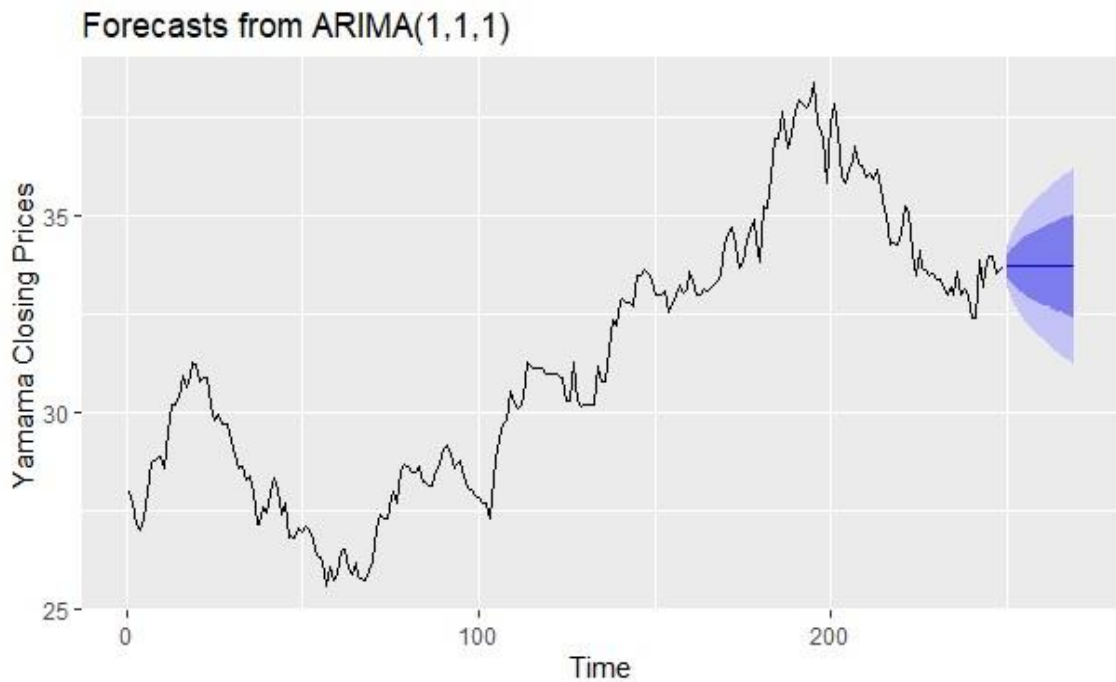


Figure 5: Original Series with 20 Steps a head Forecasts

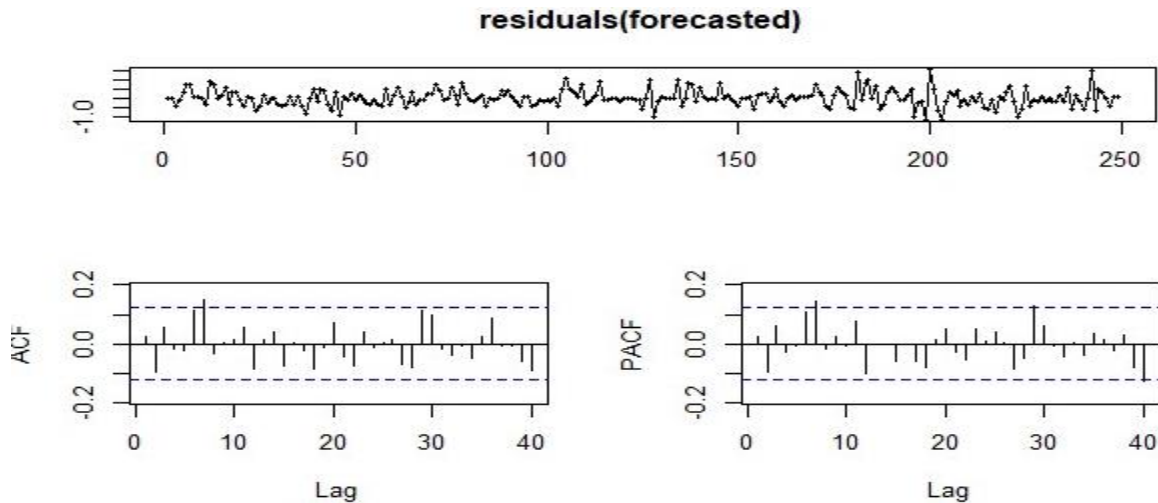


Figure 6: Residuals of the Forecasted Value with ACF and PACF Graphs.

3.2. MLP: Using automatic hidden layers option 1 to 10 hidden nodes are evaluated and the best one based on validation set using MSE is selected Table exhibits the number of hidden layers along with the MSE. Based

on the RMSE values, for the present study 5 hidden layers are selected to be used in MLP.

Table1: Showing Hidden Layers and related MSE's

Hidden nodes	MSE
1	0.0036
2	0.0036
3	0.0036
4	0.0036
5	0.0034
6	0.0034
7	0.0034
8	0.0034
9	0.0034
10	0.0034

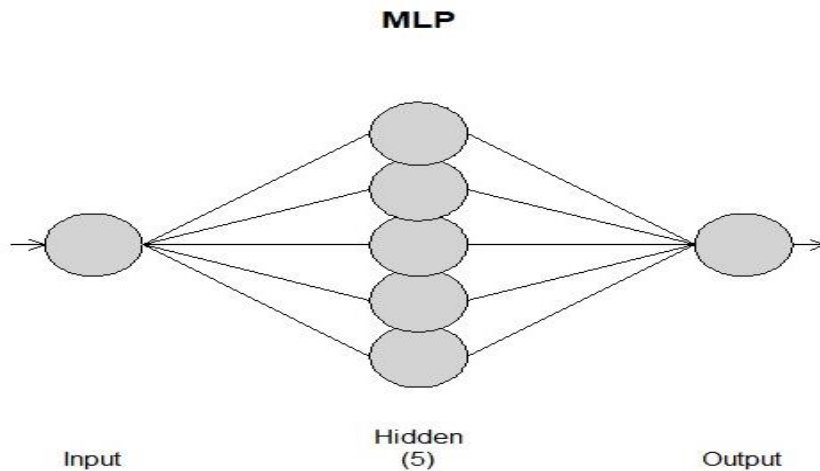


Figure 7: Showing MLP Architecture for Closing Prices of Yamama Cement

Figure 7 show the neural network architecture for the current series with one input mode depicting a lag of one-time period with five hidden nodes as suggested by automatic hidden layers option and one output node. The whole procedure was iterated/repeated 20 times using

gradient descent algorithm to get minimum value for the loss function (RMSE). RMSE is used to measure forecasting accuracy and for the MLP neural network and the value of RMSE = 0.4300.

Forecasts from MLP

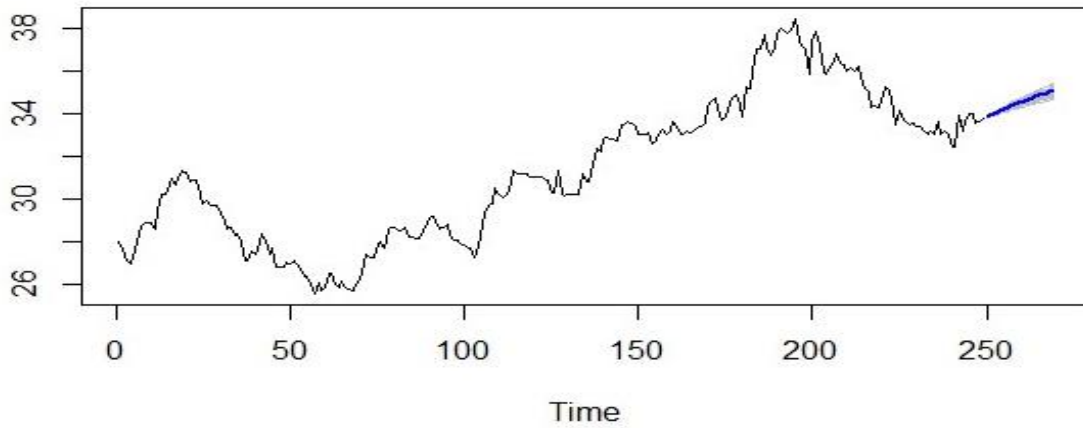


Figure 8: Forecasts using MLP for Closing Prices of Yamama Cement

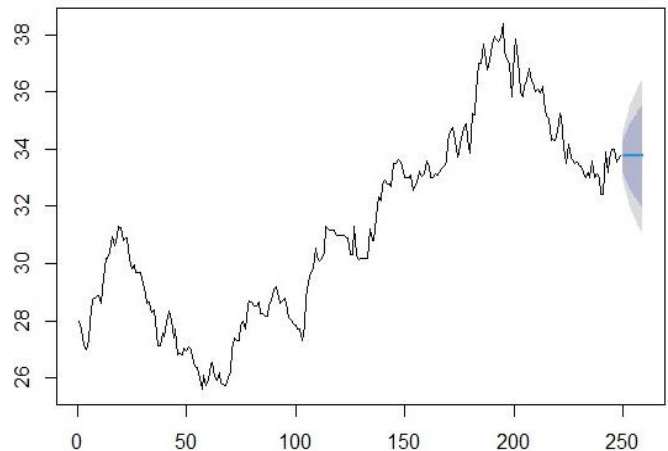
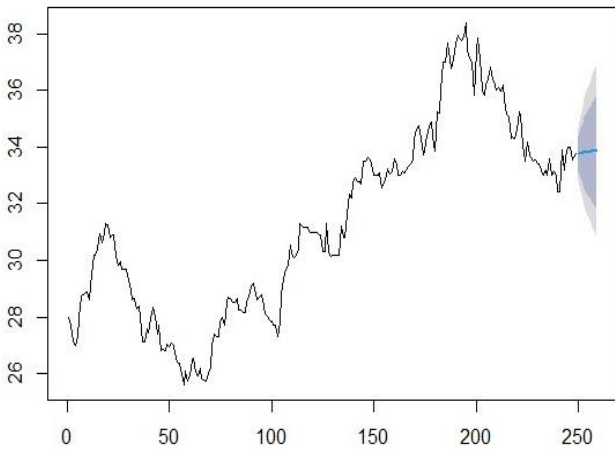
Forecasts for 20 step ahead using MLP along with the original data are depicted in Figure 8 with blue line.

3.3. Hybrid Models: The `hybridModel` function in the library (`hybridModel`) fits multiple individual model specifications to allow easy creation of ensemble forecasts.

For estimating Hybrid Models Box-Cox transformation is automatically estimated for the data. Seven models are estimated simultaneously using `forecastHybrid` library but for the present study only four models were estimated (Hybrid mode, ARIMA, NNAR and ETS).

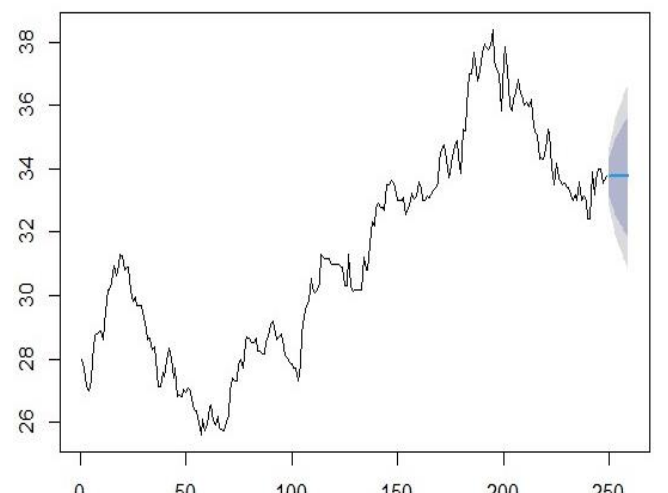
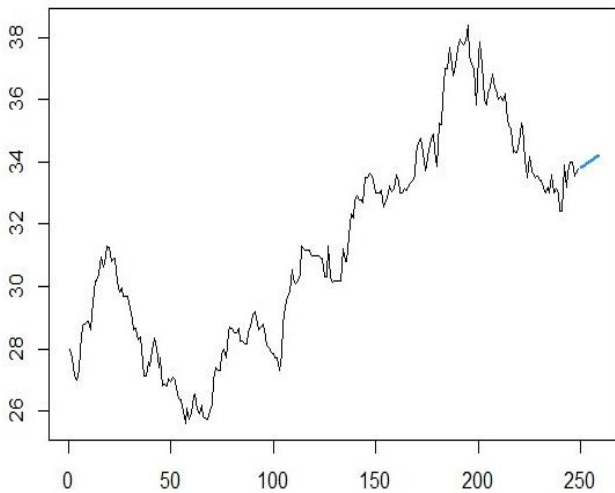
HybridModel

ARIMA(1,1,1)



NNAR(1,1)

ETS



January-February

Figure 9: Forecasts using Arima, MLP Neural Network and Hybrid Models for Closing Prices of Yamama Cement**Table 2: Comparison of Traditional and Machine Learning Models**

Models	ARIMA	Exponential Smoothing	MLP	NNAR	Hybrid Model
RMSE	0.4331	0.4331	0.4307	0.4302	0.4325

Of all the models Neural Network Auto-regressive models (NNAR) out performed all the other competing models based on the RMSE forecasting accuracy method, followed by MLP model.

4. Discussion/Conclusion/Limitations/Future Directions:

The present study was conducted to obtain a suitable forecasting model for the stock prices of Yamama Saudi Cement Company. ARIMA (1,1,1) model has been selected for forecasting stock prices which has the minimum AIC value. Comparing ARIMA and MLP model we see that the forecasting accuracy for MLP is better. But looking at the results of Hybrid Models we see than Neural Network Auto-regressive model outperforms the MLP model. Comparing ARIMA, MLP (Multi-Layer Perceptron), and hybrid models in time series forecasting reveals the strengths and weaknesses of each approach. ARIMA, a classical linear model, is effective for capturing linear trends and seasonality but may struggle with non-linear patterns. MLP, a type of artificial neural network, excels in capturing complex, non-linear relationships but requires careful tuning and is susceptible to overfitting. In conclusion, we see that ARIMA, MLP and Hybrid models have proven their efficacy in time series forecasting, each excelling in different scenarios. The hybrid models aim to provide a more versatile and accurate forecasting tool by leveraging the complementary strengths both traditional and machine learning models. The choice between these models depends on the nature of the data and the underlying patterns. A comprehensive interpretation of results from hybrid time series models allows practitioners to gain insights into the intricacies of the forecasting process, facilitating informed decision-making and improving the overall reliability of predictions in dynamic environments. In future researchers may explore Recurrent Neural Network, LSTM, Convolutional Neural Network and more novel architectures to improve predicting/forecasting accuracy.

Acknowledgement:

The authors are highly indebted to Dr. Farooq M. Alam, Head of Statistics Department, King Abdulaziz University for providing the much needed impetus for the completion of the current study.

REFERENCES:

- [1]. Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1970). *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day
- [2]. G. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, San Francisco, Holden-Day, 1976.
- [3]. Zhang, G. P., & Leung, W. K. (1997). Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. *Neurocomputing*, 10(2), 159-175.
- [4]. Yu, L., & Wang, S. (2000). A Comparison of Forecasting Performance of Neural Networks, Autoregression, and ARIMA Models on Time Series Data. *International Journal of Forecasting*, 16(3), 391-407.
- [5]. Zhang, G. P., & Leung, W. K. (1997). Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model.
- [6]. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359-366.
- [7]. Ord K., Fildes R., Kourentzes N. (2017) *Principles of Business Forecasting 2e*. Wessex Press Publishing Co., Chapter 10
- [8]. Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- [9]. Yahoo.finance.com
- [10]. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE*, 13(3), e0194889
- [11]. Aladag, C. H., Egrioglu, E., & Kadilar, C. (2009). Forecasting nonlinear time series with a hybrid methodology. *Applied Mathematics Letters*, 22(9), 1467-1470.
- [12]. Khashei, M., & Bijari, M. (2012). A new class of hybrid models for time series forecasting. *Expert Systems with Applications*, 39(4), 4344-4357.
- [13]. Hajirahimi, Z., & Khashei, M. (2019). Hybrid structures in time series modeling and forecasting: A review. *Engineering Applications of Artificial Intelligence*, 86, 83-106.
- [14] Hajirahimi, Z., & Khashei, M. (2023). Hybridization of hybrid structures for time series forecasting: A review. *Artificial Intelligence Review*, 56(2), 1201-1261.

R codes for Hybrid Models and MLP

```
library(forecast)
library(thief)
library(forecastHybrid)
library(nnfor)
YCC=data.frame(YCC1);
YCC=data.frame(YCC1$Cclose);
YCC=ts(YCC);YCC
```



```
mod1 <- hybridModel(YCC)
par(mfrow=c(1,1))
plot(forecast(mod1),main="HybridModel")
plot(forecast(mod1$auto.arima),main="ARIMA(1,1,1)")
plot(forecast(mod1$nnetar),main="NNAR(1,1)")
plot(forecast(mod1$ets),main="ETS")
accuracy(mod1)
accuracy(mod1$auto.arima)
accuracy(mod1$nnetar)
accuracy(mod1$ets)
#####MLP#####
library(forecast)
library(thief)
library(forecastHybrid)
library(nnfor)
mlp1.fit <- mlp(YCC,hd.auto.type="valid",hd.max=10)
print(round(mlp1.fit$MSEH,4))
mlp.fit=mlp(YCC)
plot(mlp.fit);
print(mlp.fit)
mlp.frc=forecast(mlp.fit,h=20,include=20)
mlp.frc
plot(mlp.frc)
accuracy(mlp.frc)
```