Time Series Forecasting using LSTM and ARIMA

Khulood Albeladi, Bassam Zafar, Ahmed Mueen
Department of Information Systems-Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, KSA

Abstract—**Time series analysis is the process of evaluating sequential data to extract meaningful statistics. In the current era, organizations rely greatly on data analysis to solve and predict possible answers to a specific problem. These predictions help greatly in decision-making. In time series problems, the data is used to train the different machine and deep learning models. The models train on provided data displays particular outcomes. These outcomes anticipate possible solutions. In this paper, the two most effective Python models LSTM (Long Short Term Memory Loss) and ARIMA (Autoregressive integrated moving average) are used. These are the two most recommended models while dealing with Time series forecasting. The selected dataset is from Mulkia Gulf Real Estate available at MarketWatch. The main objective of this research paper is to study and compare the results of the two models used and determine which one is the best-suited model for that particular type of prediction. However, these are widely used models but the focus point of this research is determining the performance variance between these two models. LSTM became famous in 1997 as a training model that can remember patterns based on previous data while ARIMA is famous for forecasting variables of interest using a linear combination of previous values of the variable. The findings state that ARIMA is better for time series forecasting than LSTM based on the mean average of the basic evaluation parameters.**

Keywords—Stock analysis; machine learning; deep learning; time series; ARIMA; LSTM

I. Introduction

Time series is a time-dependent dataset, which means that the values are obtained in specific intervals of time. Usually, the values are taken at regular intervals, but the sampling could be irregular [4]. If a time series has a definite pattern, then any value of the series should be a function of previous values. Time series models differ from others in the way it predicts. With the advancement of information technology, now there are many more ways to collect time series data. A time series model uses a lag value of the target variable and uses it as a predictor variable, whereas traditional models use other variables as predictors. Time series analysis is the process of extracting output from time series data using different techniques. One of the famous types of time series analysis is Time series forecasting. In time series forecasting, the results are the predicted outputs from the trained models. There are many forecasting models available. In this research, LSTM and ARIMA models are used.

A general LSTM model has a cell; each cell has three parts, i.e. Forget gate, Input Gate and Output Gate. The states are affected by both past states and current input with feedback connections. LSTM models are able to learn long-term dependencies with feedback connections. LSTM can be single or multilayered models, the functionality differs for both. They are preferred because they hold information for a long time by default. ARIMA forecasts temporal dependencies using only historical values. These models help to gain better insights into the data and predict future trends. It works by stationarising the series, means studying the correlation of the values and check its residual diagnosis ACF and PACF plots.

A number of forecasting problems can be solved using LSTM and ARIMA models. Although, the models have been used for prediction in the past (21) 19 18 12, but the goal of this research is to elaborate the differences between the outcomes that these models provides. The key point is understanding how well one model performs than the other when trained on same datasets. The evaluation of these models is done based standard evaluation parameters (MAE, MSE, R recall, etc.). The graphs included will also help in determining which model is best for doing time series for stock price prediction.

II. Review of Related Literature

The time series is the set of quantitative observations arranged in chronological order. Time series analysis has attracted a lot of attention in the past three decades [1]. In the past, it is generally believed that time can be a continuous or discontinuous variable and no comparison exists between the dependent variables [2]. Time series have always been used in the field of econometrics. Jan Tinbergen (1939) devised the first time series econometric model. He also started the scientific research program on the basis of experienced econometrics. At that time, it was hardly considered that chronologically ordered observations might depend on each other. The dominating assumption was that, according to the classical linear regression model, the surplus of the estimated equations is randomly determined and is independent of each other [3].

The fundamental goal of building a time series model is the same as building a precision model that provides a value nearly equal to the values present in the series. From a statistical point of view, time series are the recordings of aleatory processes which vary over time. The most distinguishing feature of time series is that the distribution of observation at a specific point conditional on the previous value of the series depends on the outcome of those previous observations, simply making the outcomes independent. The recordings can be a continuous pattern or a set of discrete values. There are two main types of time series patterns that exist, stationary and non-stationary. Stationary time series values have statistical properties and moments while non-stationary values are simple recordings with changing statistical properties. Both types of series can be used in time series forecasting models [5] [6].

Time series forecasting model building is done based on the type of dataset used to train models. Stationary datasets are easier to train for prediction than non-stationary datasets. In fact, it is a necessity to convert a non-stationary dataset into a stationary dataset [7]. Models easily understood stationary datasets and extract information more efficiently from them. There are certain methods devised to convert non-stationary datasets into stationary datasets [8]. The popular methods are the Hilbert-Hhuang transform, Fourier transform, Dicky Fuller, etc. Hilbert Hhuang method is specially developed for analyzing non-linear and non-stationary data [9].

Naturally, people's habit of forecasting and making predictions is immemorial. Simply forecasting is the name of predicting outcomes of a plan, but in python, the forecasting is done using datasets available and training them using in-built functions. Using time series forecasting is the process of finding possible values for anything using a known data set. Time series is a popular technique in the current era to solve all types of problems, predicting directly affects the decisions and escort towards clearer imagination. Time series forecasting can be done using both machine learning and deep learning models. The models are different based on their working specifications [10].

There are numerous models in python that serve the purpose of predicting values. Supervised machine learning models and deep neural networks are used for prognostications. Supervised machine learning empowers systems with the ability to learn automatically from the data and get better outcomes with experience without being explicitly programmed [11]. Similarly, deep networks have applications in many areas of life including prediction, detection, creation, etc. Deep neural networks model provide better forecasts as compared to machine learning models. When dealing with real world's problems shallow neural networks need to be sufficiently expressive to predict the task optimally while deep neural networks (DNNs) have been proposed as a way of producing more predictive models [12]. These models are sequences of layers in which each layer uses a linear transformation function. When the layers are combined, it constructs a deep neural model. As many functions are included, that automatically enhances the models' prediction ability.

These models have complex backgrounds and seem difficult to build. With the passage of time, there came a great evolution and now these models are readily present in python libraries. Python has libraries with inbuilt forecasting models that can be used to do predictions. LSTM and ARIMA are the two most influential and long-established models. Models are used considering the characteristics of the encountered problem. A traditional LSTM is a sub-type of RNN model that saves previous sequential data as temporal pieces of information [13], [14], [15].

A general LSTM model comprises the cell; each cell has three parts, i.e. Forget gate, Input Gate and Output Gate. These states are affected by both past states and current input with feedback connections. A standard recurrent cell consists of Sigma and Tanh cells. Each memory cell in LSTM has recurrently self-connected linear unit LSTM. These linear units are called CEC (Constant Error Council). The architecture of LSTM permits it to bridge huge time lags between relevant input events of almost 1000 steps and more. This method is used in the processing of time-series data, in prediction, as well as in the classification of data.

LSTM models work efficiently and have been widely used in various kinds of tasks. The activation function in the output layer determines in which direction the training will lead here [16], [17].

ARIMA (autoregressive integrated moving average) is also a sub-type of RNN networks and has proven immensely useful for prediction [18], [19], and [20]. They have been studied thoroughly and remodeling has been done on ARIMA processes. These models were popularized by George Box and Gwilym Jenkins [21]. That's why ARIMA processes are sometimes known as Box-Jenkins models. They effectively put together everything in a comprehensive manner and the relevant information required to understand and use univariate ARIMA processes. ARIMA models have gained popularity because they can very accurately do short-term forecasts. Sometimes there is some information that cannot be extracted through regression, so ARIMA is used to capture this additional information. In these models, autocorrelation and partial autocorrelation functions are used, as basic instruments, to identify the stationarity of time series [22]. Both LSTM and ARIMA models have their own specific functionalities and their own advantages and disadvantages.

The previous studies showcase the use of ARIMA and LSTM models in different industries. The use of these models is common in finance and commerce markets but they cannot be seen used in trading markets. The literature gives insights that use of LSTM and ARIMA models in prediction problems is very limited. The research is done to discuss the results of implementation of these two models and evaluating the performance models when they are trained on same dataset. The results helps estimate which model is more useful for stock exchange prediction.

III. METHODOLOGY

Time series forecasting is used to deal with a surprisingly vast set of problems. There are different techniques available to implement time series forecasting, each differs slightly but affects greatly. The process consists of five major steps. As in this paper, High Price prediction is being done using two different models (i.e. LSTM and ARIMA). This five-step methodology is used to implement these two models. Although the general working of all five models is identical, it varies greatly according to the technicalities used in each model.

Fig. 1 shows the general methodology steps which are explained, respectively.

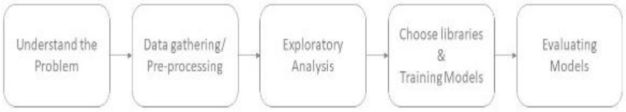

Understand the Problem
Data gathering/ Pre-processing
Exploratory Analysis
Choose libraries & Training Models
Evaluating Models

Fig. 1. General methodology flowchart.

A. Understand the Problem

As mentioned earlier, time series data is used to deal with lots of problems. Time series forecasting is the type where certain solutions are speculated by training previous data. The solutions are; predicting stock market reach at a certain time, estimating survey results based on previous data, weather prediction at a certain date, etc. it is important to build a model keeping in view the nature of the problem you are dealing with. Explore the models that can be used according to the given problem. In addition to understanding models, thoroughly understand the techniques used to implement these models in order to choose the best according to the problem.

The purpose of this research is to build two different time series forecasting models in order to proclaim which is the best model for predicting high prices. To do so, there are two best-performing models (LSTM & ARIMA) that have been working exceptionally well while dealing with prediction problems.

B. Data Gathering / Pre-processing

After understanding the nature of the problem, the next step is to determine the type of data needed in order to train the models. The data should be the cross-sectional type of data. This type has one dependable variable that confides in numerous independent variables. In such data types, the small-scale or aggregate entity is observed at different points in time. In a nutshell, this is the data of different entities collected at the same time.

In this paper, data from Mulkia Gulf Real Estate from Saudi Exchange datasets are used. The data is extensive to cause models to overfit, so certain sheets (i.e. Fig. 2; sheet 1 & Fig. 3; sheet 7) are chosen to train models. It is a cross-section of data perfectly meeting the criteria of the needed data. The purpose of using this data is because the data possess similarities that are consonant with the prediction problem. After training models on this data the trained model can be used to predict the high price value. This data accurately meets the criteria of needed data to predict the highest price value.

After data has been selected, the next step is to pre-process this data. As mentioned earlier, the data type used is non-stationary and cross-sectional data. The models used can train on stationary data more efficiently, so it is necessary to convert this data into stationary data. There are certain methods available to convert non-stationary data into stationary. ADF (Argumented Dicky Fuller) method is used to transform the data into non-stationary. In python, ADF can be imported from the Stats model library as statsmodels.tsa.stattools.adfullers.

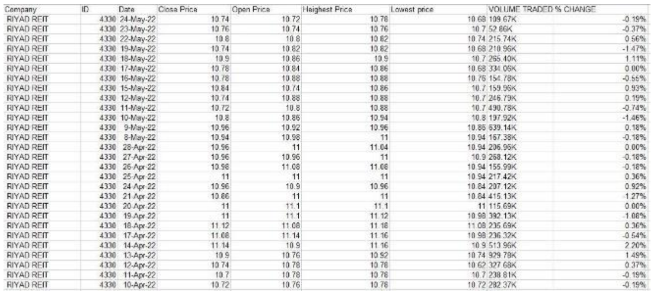
C. Exploratory Analysis

Exploratory analysis is the process of analyzing the relationships between variables that exist in raw data. These initial relationships help to understand the nature of the data and how accurately the desired information will be extracted from it. It's mandatory to do exploratory analysis independent of the type of problem, it's mandatory to explore the data first. It's also called preliminary analysis of data, where you plot the data in its original form to find certain structures. When preliminary analysis is done, check the validity of measures, and point out any outliners. It also helps in examining the weightage of different variables to evaluate the effectiveness of certain manipulative variables, etc. The major thing that should be implemented while doing preliminary analysis is cleaning disrupted data, checking the data for null values, etc.

Fig. 4 shows that the data has been scaled properly and is ready to train for the prediction models. It also shows the regression patterns of various variables alongside dates.

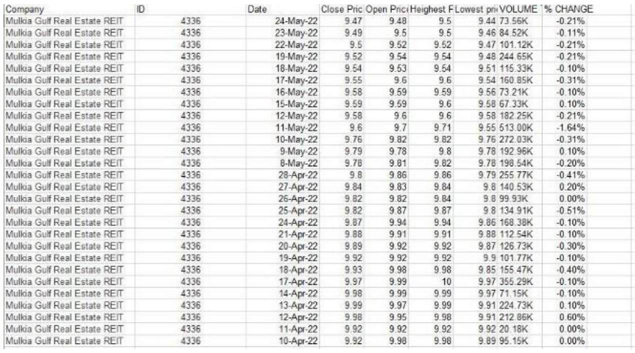
D. Choosing Libraries and Training Models

In Python, there are thousands of libraries and in-build models available that deal with machine learning and deep learning problems. Python libraries are a set of specific functions put together in a single file. The purpose to make these libraries is to assist coders in doing obvious steps. There is a whole set of libraries available for different domains of artificial intelligence. The problem explained in this research paper is time series forecasting, which is a type of machine learning and deep learning problem. The libraries which come in handy while dealing with such problems are sklearn, tensorflow, numpy, etc.



Company	ID	Date	Close Price	Open Price	Heighest Price	Lowest price	VOLUME TRADED	% CHANGE
RIYAD REIT	4330	24-May-22	10.74	10.72	10.78	10.68	109.67K	-0.19%
RIYAD REIT	4330	23-May-22	10.76	10.74	10.76	10.7	52.66K	-0.37%
RIYAD REIT	4330	22-May-22	10.8	10.8	10.82	10.74	215.74K	0.56%
RIYAD REIT	4330	19-May-22	10.74	10.82	10.82	10.68	210.96K	-1.47%
RIYAD REIT	4330	18-May-22	10.9	10.86	10.9	10.7	265.40K	1.11%
RIYAD REIT	4330	17-May-22	10.78	10.84	10.86	10.68	334.06K	0.00%
RIYAD REIT	4330	16-May-22	10.78	10.88	10.88	10.76	154.78K	-0.55%
RIYAD REIT	4330	15-May-22	10.84	10.74	10.86	10.7	159.96K	0.93%
RIYAD REIT	4330	12-May-22	10.74	10.88	10.88	10.7	246.79K	0.19%
RIYAD REIT	4330	11-May-22	10.72	10.8	10.88	10.7	490.78K	-0.74%
RIYAD REIT	4330	10-May-22	10.8	10.86	10.94	10.8	197.92K	-1.46%
RIYAD REIT	4330	9-May-22	10.96	10.92	10.96	10.86	639.14K	0.18%
RIYAD REIT	4330	8-May-22	10.94	10.98	11	10.94	167.38K	-0.18%
RIYAD REIT	4330	28-Apr-22	10.96	11	11.04	10.94	206.95K	0.00%
RIYAD REIT	4330	27-Apr-22	10.96	10.96	11	10.9	268.12K	-0.18%
RIYAD REIT	4330	26-Apr-22	10.98	11.08	11.08	10.94	155.99K	-0.18%
RIYAD REIT	4330	25-Apr-22	11	11	11	10.94	217.42K	0.36%
RIYAD REIT	4330	24-Apr-22	10.96	10.9	10.96	10.84	207.12K	0.92%
RIYAD REIT	4330	21-Apr-22	10.86	11	11	10.84	415.13K	-1.27%
RIYAD REIT	4330	20-Apr-22	11	11.1	11.1	11	115.69K	0.00%
RIYAD REIT	4330	19-Apr-22	11	11.1	11.12	10.98	392.13K	-1.08%
RIYAD REIT	4330	18-Apr-22	11.12	11.08	11.18	11.08	235.69K	0.36%
RIYAD REIT	4330	17-Apr-22	11.08	11.14	11.16	10.98	236.32K	-0.54%
RIYAD REIT	4330	14-Apr-22	11.14	10.9	11.16	10.9	513.96K	2.20%
RIYAD REIT	4330	13-Apr-22	10.9	10.76	10.92	10.74	929.78K	1.49%
RIYAD REIT	4330	12-Apr-22	10.74	10.78	10.78	10.62	327.68K	0.37%
RIYAD REIT	4330	11-Apr-22	10.7	10.78	10.78	10.7	238.81K	-0.19%
RIYAD REIT	4330	10-Apr-22	10.72	10.76	10.78	10.72	282.37K	-0.19%

Fig. 2. Mulkia gulf real estate from Saudi exchange sheet 1.



Company	ID	Date	Close Pric	Open Pric	Heighest F	Lowest pri	VOLUME	% CHANGE
Mulkia Gulf Real Estate REIT	4336	24-May-22	9.47	9.48	9.5	9.44	73.56K	-0.21%
Mulkia Gulf Real Estate REIT	4336	23-May-22	9.49	9.5	9.5	9.46	84.52K	-0.11%
Mulkia Gulf Real Estate REIT	4336	22-May-22	9.5	9.52	9.52	9.47	101.12K	-0.21%
Mulkia Gulf Real Estate REIT	4336	19-May-22	9.52	9.54	9.54	9.48	244.65K	-0.21%
Mulkia Gulf Real Estate REIT	4336	18-May-22	9.54	9.53	9.54	9.51	115.33K	-0.10%
Mulkia Gulf Real Estate REIT	4336	17-May-22	9.55	9.6	9.6	9.54	160.85K	-0.31%
Mulkia Gulf Real Estate REIT	4336	16-May-22	9.58	9.59	9.59	9.56	73.21K	-0.10%
Mulkia Gulf Real Estate REIT	4336	15-May-22	9.59	9.59	9.6	9.58	67.33K	0.10%
Mulkia Gulf Real Estate REIT	4336	12-May-22	9.58	9.6	9.6	9.58	182.25K	-0.21%
Mulkia Gulf Real Estate REIT	4336	11-May-22	9.6	9.7	9.71	9.55	513.00K	-1.64%
Mulkia Gulf Real Estate REIT	4336	10-May-22	9.76	9.82	9.82	9.76	272.03K	-0.31%
Mulkia Gulf Real Estate REIT	4336	9-May-22	9.79	9.78	9.8	9.78	192.96K	0.10%
Mulkia Gulf Real Estate REIT	4336	8-May-22	9.78	9.81	9.82	9.78	198.54K	-0.20%
Mulkia Gulf Real Estate REIT	4336	28-Apr-22	9.8	9.86	9.86	9.79	255.77K	-0.41%
Mulkia Gulf Real Estate REIT	4336	27-Apr-22	9.84	9.83	9.84	9.8	140.53K	0.20%
Mulkia Gulf Real Estate REIT	4336	26-Apr-22	9.82	9.82	9.84	9.8	99.93K	0.00%
Mulkia Gulf Real Estate REIT	4336	25-Apr-22	9.82	9.87	9.87	9.8	134.91K	-0.51%
Mulkia Gulf Real Estate REIT	4336	24-Apr-22	9.87	9.94	9.94	9.86	168.38K	-0.10%
Mulkia Gulf Real Estate REIT	4336	21-Apr-22	9.88	9.91	9.91	9.88	112.54K	-0.10%
Mulkia Gulf Real Estate REIT	4336	20-Apr-22	9.89	9.92	9.92	9.87	126.73K	-0.30%
Mulkia Gulf Real Estate REIT	4336	19-Apr-22	9.92	9.92	9.92	9.9	101.77K	-0.10%
Mulkia Gulf Real Estate REIT	4336	18-Apr-22	9.93	9.98	9.98	9.85	155.47K	-0.40%
Mulkia Gulf Real Estate REIT	4336	17-Apr-22	9.97	9.99	10	9.97	355.29K	-0.10%
Mulkia Gulf Real Estate REIT	4336	14-Apr-22	9.98	9.99	9.99	9.97	71.15K	-0.10%
Mulkia Gulf Real Estate REIT	4336	13-Apr-22	9.99	9.97	9.99	9.91	224.73K	0.10%
Mulkia Gulf Real Estate REIT	4336	12-Apr-22	9.98	9.95	9.98	9.91	212.86K	0.60%
Mulkia Gulf Real Estate REIT	4336	11-Apr-22	9.92	9.92	9.92	9.92	20.18K	0.00%
Mulkia Gulf Real Estate REIT	4336	10-Apr-22	9.92	9.98	9.98	9.89	95.15K	0.00%

Fig. 3. Mulkia gulf real estate from Saudi exchange sheet 7.

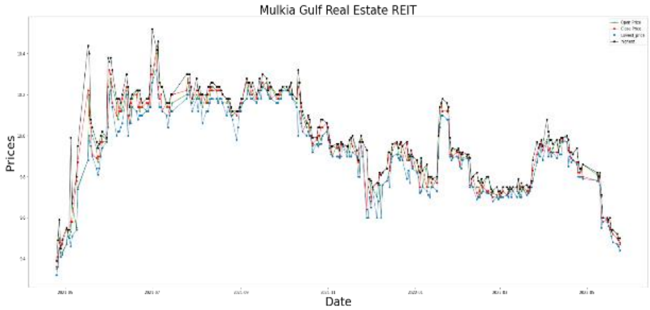


Fig. 4. Exploratory analysis.

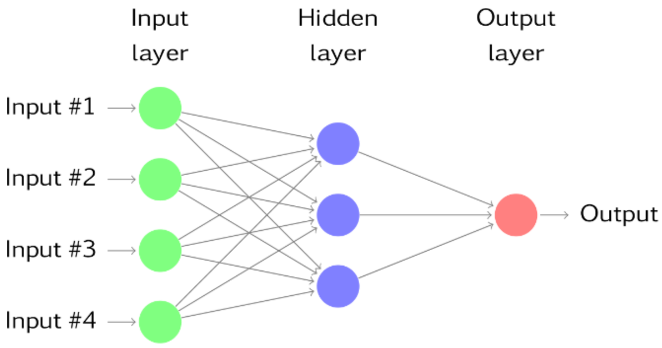


Fig. 5. Basic deep learning neural network.

These libraries are designed according to a specific set of rules that fit their receptive type of problem. For preprocessing of this data; numpy, sklearn, tensorflow, and keras are used. Similarly, there are some in-built forecasting models that are available in python. Use these models suitably for respective problems. The models serve specific purposes; they have been created according to some mathematical ordinances. According to the problem type choose these rules and embed them into deep learning models. A typical deep learning model consists of an input layer, multiple hidden layers (where most of the work is done), and the output layer. Fig. 5 shows the general representation of a deep learning neural network model.

Now keeping in view the nature of the problem, use LSTM and Auto-Regressive models.

E. *Evaluating Models*

Evaluation is done by measuring evaluation parameters. The three basic parameters that are used in measuring a neural model's reliability in time series forecasting problems are Accuracy, Precision, and Recall. In addition to these parameters, some auxiliary parameters are available to evaluate regression results. The added parameters are MSE (Mean Square Error), MAE (Mean Absolute error), MAPE (Mean Absolute percentage), and MDAPE (Median Absolute Error Average and Percentage). These metrics collectively help provide an explanation regarding the mistakes made unknowingly.

IV. Implementing Time Series Models

In this research paper, the two best time series forecasting models are used that are already available in python. These models will be trained on two datasets from Mulkia Gulf Real Estate from Saudi Exchange. Sheet 1 & Sheet 7 of these datasets will be used to train the models. The aim is to determine which model is the best model for respective targeted industry. The problem is predicting the highest profits values. The trained models will be attested, and finding will be evaluates based on evaluation parameters to get exquisite results. The general methodology explained above will be followed. The models are embedded one by one into the neural networks. After embedding the models, the pre-processed datasets will be provided and will run in the code for the training of models. The general method proposed to train the models is nearly identical. They only differentiate based on the functionality attributes, and the technical differences, i.e. number of hidden layers used, the evaluation parameters used, and the dataset preparations, the working background. Detailed information on these python models is provided below.

A. *Long Short-Term Memory Loss (LSTM)*

As the prediction model should remember long-lasting events from part, LSTM is the first choice to use in this paper. LSTM is an inbuilt python function that can be imported from TensorFlow. LSTM is the branch of recurring neural networks. It was seen that there weren't any RNN structures that can do backpropagation of long intervals, so to solve such difficulties LSTM was proposed. LSTM has two gate unit cells that open and close to the information flow within each memory cell, i.e. Packs of information between time lags [23]. It requires the data used to be in a definite shape. The dimension should be equivalent, and the data should be properly cleaned, integrated, and scaled. The commonly used activation functions for LSTM-based regression problems are Sigmoid and Tanh. Tanh has proven to be very effective in dealing with vanishing gradients.

The functioning of the LSTM model is explained in Fig. 6.

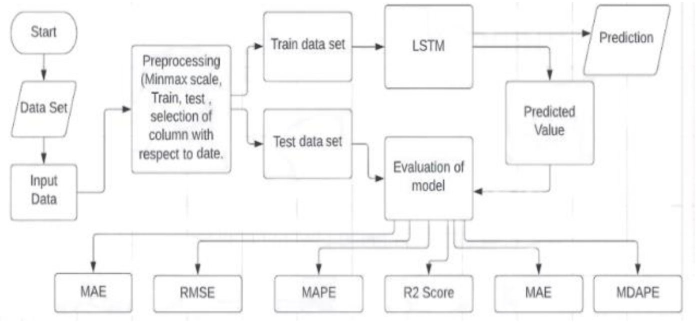


Fig. 6. Flowchart for LSTM working.

As said earlier, LSTM has the ability to remember information for the long term. This is done by remembering the previous output and combining it with the current one. To understand the working of LSTMs better, there is a need to fathom the mathematics behind them.

Divide the working of LSTM neural networks into four stages. In the first stage, the model decides whether to forget or remember information in the previous cell. This value is decided by doing the following calculation:

$$\text{Forgetting value } f_t = \sigma \, [\, W_f \, (h_t - 1, X_t) + b_f \,]$$

Where σ is the activation used in the input layer, W_f and B_f are the weight and bias vectors, h_t-1 is the output at time t-1, and X_t is the input vector at time t. If the forgetting value is equal to zero then this means that the previous value has been forgotten.

In the second stage, the model decides which value will get stored in the next cell state. To do so, sigmoid and tanh layers are used. The sigmoid layer chooses the pieces of information which need to be updated, and the layers decide on a second option value. By combing these two values, the models create new values to update the cell state. The formula to calculate sigmoid and tanh values are:

Input gate value $i_t = \sigma\ (W_i \cdot [h_t{-}1, X_t] + bi)$

Updated value $C_t = \tanh\ (WC \cdot [h_t{-}1, X_t] + bC)$

σ represents the sigmoid-shaped function, it is the input gate value and C is the updated value at time t.

In the third stage, the new cell state NC_t is obtained using ft and its value. The equation below shows how to calculate this NC_t value,

$C_t = f_t\ [C_t{-}1 + i_t]e$

In the final stage of LSTM, it is determined which value will get considers as output. Use the sigmoid equation to determine which cell state will be the output; the value is processed through tanh to get a value between 1 and -1.

The inbuild model present in python work is according to the above-explained structure. The models can be improved further by choosing the correct values of weights and bias values, activation functions, and the number of hidden layers.

B. Autoregressive Integrated Moving Average (ARIMA)

ARIMA forecasts temporal dependencies using only historical values. The data used for Autoregressive models are prepared differently from LSTM. In addition to necessary preprocessing steps, the AR model's data needs to be stationary. Simply put, data is stationary when its numeric properties do not change over time. From a mathematical perspective, it refers to the data whose Mean and Variance will not depend on time. If the data doesn't meet the properties of the stationary dataset, you can do a series transformation to make it stationary.

ARIMA model is a combination of two models Autoregressive (AR) and Moving Average (MA), integration (I) is applied at least once to make the data set stationary.

In the AR part of the model future values are predicted using the lags from the data values. The general equation AR model is:

$$\text{AR(p)}: x_t = \alpha + \sum_{i=1}^{p} \beta i x_{t-i} + \epsilon$$

The Moving Average is the part of the model where value is forecasted using the forecasting error differences is calculated while making predictions. The general form of MA equations is:

$$x_t = \mu + \sum_{i=1}^{q} \Phi_i \epsilon_{t-1}$$

Prediction is done by combining all three orders and getting an estimation of how to quickly fit the model. Some standard denotations are used to represent the above three, i.e. p, d, and q; "p" is the number of observations included in the model, "d" is the number of times differentiating the raw observations, "q" is the number of moving average size. To find these parameters, first fetch a Correlation and Partial Correlational graph from the dataset.

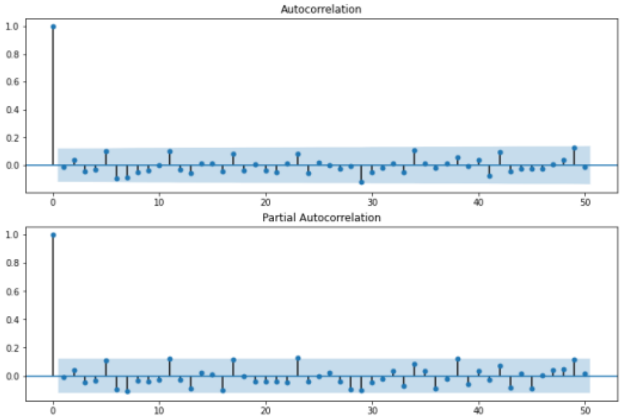


Fig. 7. Autocorrelation graph of sheet 1 data.

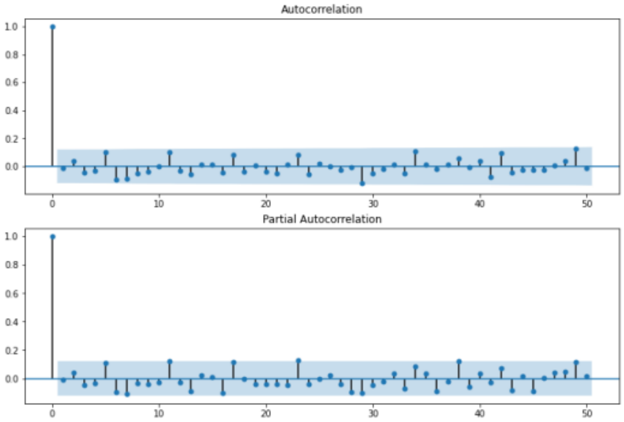


Fig. 8. Autocorrelation graph of sheet 7 data.

Fig. 7 and 8 can be used to roughly estimate the value of p, d, and q. The integer value of p can be obtained from the cut-off edge points of PAC graphs. Similarly, the value of q can be obtained using an AC graph. If the graph does not represent steady cut-off points, use in-built PACF and CAF functions.

Fig. 9 shows how the ARIMA works and on which parameters its working can be evaluated.

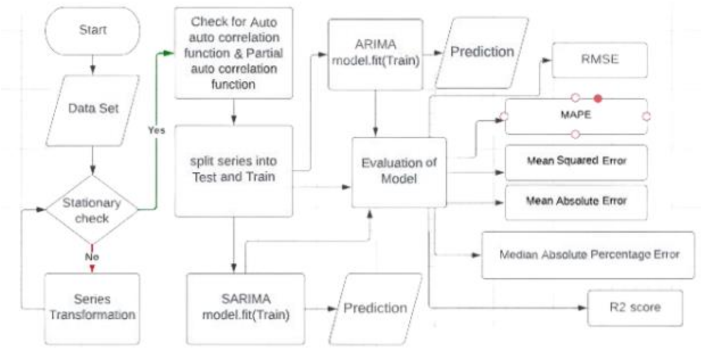


Fig. 9. Flowchart for ARIMA working.

V. Evaluation Parameters for Model's Performance

Taking note of the significance of time series models, it is very important to examine the results of these models. For specific industries, the targeted values of forecasting models are the milestones for making decisions [9]. Artificial intelligence learning models are always trained using real-world data. They are prepared to use for actual problems, so they are usually tested using the same type of data.

Keeping in view the importance of time series models, scientists have devised a number of ways to test the audacity of the models used. In this part, the performance parameters the will be used to evaluate the models will be discussed, and how best model among them will be picked. Following are the parameters used for testing model's prediction validation.

A. R Square Score - R2 Score

It's a statistical value that is used in a prediction model to find the extent of difference between dependent variables that can be made sense of by the independent factor. In simple words, R2 is how well the testing data fits the trained regression model. The general formula used to calculate the R2 score of the training models is.

R2 = 1- sum of squares due to regressions total sum of squares

The R2 score can only be calculated for both. The R2 values for both models are displayed in Table I.

TABLE I. R2 Values

R2 Score	LSTM	ARIMA
Sheet 1	-0.2142	0.826684
Sheet 7	-0.3214	0.7389812

B. Mean Square Error - MSE

It is the mean square of differences between the trained model values and tested values. It squares the values of the differences in order to remove the negative sign and increase the weight of the larger values. The formula used to calculate the MSE is

$$MSE = \sum (Xt - X't)\ 2n$$

n = Number of items forecasted

t = Time period values

X = actual values of the dependable variable (In the case, actual values of the highest prices)

X' = Prediction values of the dependable Variable

In order to estimate the performance of the model, calculate MSE, the smaller a value is the better the model prediction. The MSE value for the ARIMA model (Sheet 1 & Sheet 2; 0.033151 and 0.015704).

C. Root Mean Square Error - RMSE

RMSE is obtained by taking the square root of MSE equations. Root square is added in order to return the MSE value to become consistent, not compromising the ability to penalize error. There are certain models that contain inconsistent but larger errors. To calculate the possibility of occurring such an error, compare RMSE with MAE to see if such errors are present. The RMSE value for LSTM is (Sheet 1 & Sheet 2; 0.088/0.474 and 0.130/0.232).

D. Mean Absolute Error - MAE

It is the mean of the absolute difference between the observed value, and the real observation value. In other words, it tells how much larger the difference between actual and predicted values which can be expected from the model. The smaller the value of MAE, the better the model will work. If the value is zero, the model can predict future values accurately. Models are compared on the basis of MAE values such as; a model with a smaller MAE will be considered best among all. MAE simply calculates the error; it can't identify the weightage of individual values. The general equation used to measure MAE is:

$$MAE = \sum (X - X')\ n$$

The MAE value for ARIMA is (Sheet 1 & Sheet 2; 0.111602 and 0.071160).

E. Mean Absolute Percentage Error - MAPE

MAPE is obtained by measuring the percentage of MAE to the real values, (In this case it's X - Real values of Prices. Mainly find MAPE, when data is without Zeroes and Extreme values. Read this error the same as MAE which means the lower MAPE value model is the best model. The MAPE value for ARIMA is (Sheet 1 & Sheet 2; 0.011073 and 0.007107).

F. Median Absolute Percentage Error - MDAPE

A little modification is done in MAPE - Mean Absolute percentage error in order to get MdAPE. The median of the MAE is found by arranging the values from the smallest to the largest; then the middle value only if it's even is picked as MdAPE.

MdAPE is recommended for evaluation when you are dealing with ARIMA models. MDAPE evaluation is done for ARIMA models. MdAE Percentage is interpreted as good when it's between 10% and 20%. The RMSE value for ARIMA is (Sheet 1 & Sheet 2; 0.630533 and 0.333124).

In addition to all the parameters mentioned above, also study the avg_loss and val_loss graphs to evaluate models like LSTM. These additional parameters help to evaluate and compare the two models better.

VI. Results

The purpose of the research is to know which model is the best-performing model for the Highest Price value prediction among the two that have trained using different Python libraries. The results will comprehend using graphs combined with the evaluation parameters explained above.

As mentioned, two sets of Data have been used; Sheet 1 & Sheet 7 from the dataset of Mulkia Gulf Real Estate available at Saudi Exchanges. The first model trained on this data is Long Short Term Memory (LSTM). The evaluation graph shows the regression between actual trained and prediction train values from sheet 1 to sheet 7 as follows:

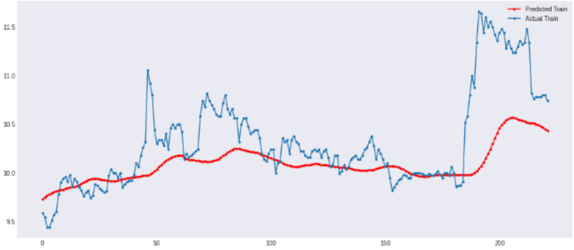


Fig. 10. Sheet 1 LSTM prediction.

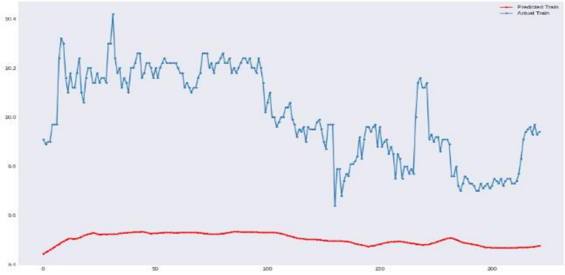


Fig. 11. Sheet 7 LSTM prediction.

Fig. 10 and 11 shows the regression differences present between actual value and predicted values. The LSTM model is evaluated using RMSE, avg_loss, and val_loss. The RMSE value for sheet 1 is 0.088 & for sheet 7 is 0.130. The average and validation losses are between 0.08 and 0.17.

The next model trained was the Autoregressive Integrated Moving Average (ARIMA) model made by importing ARIMA from statsmodels.tsa library from python. The graphs indicating the model results for real and predicted values are:

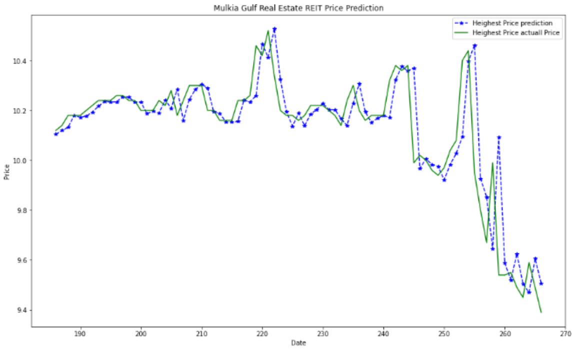


Fig. 12. Sheet 1 ARIMA prediction.

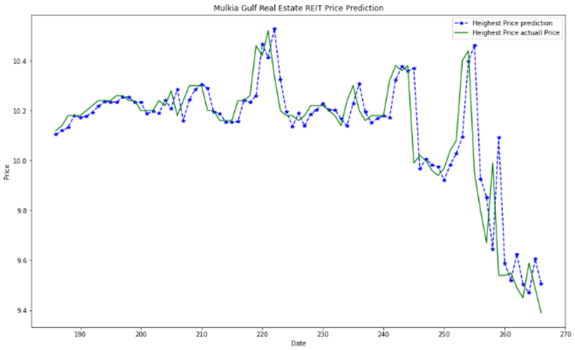


Fig. 13. Sheet 1 ARIMA prediction.

The visualization shows (Fig. 12 & 13) shows very little difference between the actual value and the predicted values from ARIMA.

The mathematical evaluation of ARIMA was done using MAE, MAPE, MdAPE, and MSE. You can read those values from tables. The overall values gave an idea that ARIMA can predict the future highest price more accurately than LSTM.

VII. Conclusions

In this paper, the time series forecasting problem is explained. In this paper, the highest price prediction using two python in-build models (i.e. LSTM & ARIMA) is done. The data is trained on the Mulkia Gulf Real State dataset. The dataset is pre-processed and trained on deep-learning models. After training evaluation of models is done using evaluation parameters; MAE, MSE, RMSE, val_loss, accuracy, R2 score, etc. The values of these parameters collectively determine which model performed the best. These parameters indicate that the ARIMA model can predict the highest price more accurately. When viewed from the graphical point of view, there is very little difference between actual values and predicted values in Fig. 12 and 13, this shows that ARIMA can predict more precisely as compared to LSTMs. Overall, it was concluded that for stock price prediction, ARIMA models can perform better than LSTM Models.

References

[1] Kam, Kin Ming. *Stationary and non-stationary time series prediction using state space model and pattern-based approach*. The University of Texas at Arlington, 2014.

[2] Hyndman, Rob J., and George Athanasopoulos. Forecasting: principles and practice. texts, 2018.

[3] Kirchgässner, Gebhard, Jürgen Wolters, and Uwe Hassler. *Introduction to modern time series analysis*. Springer Science & Business Media, 2012.

[4] Charlton, M. & Caimo, A. Time Series Analysis. European Spatial Planning Observation Network (ESPON). 2012.

[5] Wei, William WS. "Time series analysis." (2013).

[6] Parzen, Emanuel. "An approach to time series analysis." *The Annals of Mathematical Statistics* 32.4 (1961): 951-989.

[7] Manuca, Radu, and Robert Savit. "Stationarity and nonstationarity in time series analysis." *Physica D: Nonlinear Phenomena* 99.2-3 (1996): 134-161.

[8] Manual, Solution. "Data Mining: Concepts and Techniques".

[9] Huang, Norden E., et al. "Applications of Hilbert–Huang transform to non-stationary financial time series analysis." *Applied stochastic models in business and industry* 19.3 (2003): 245-268.

[10] Yaffee, Robert A., and Monnie McGee. *An introduction to time series analysis and forecasting: with applications of SAS® and SPSS®*. Elsevier, 2000.

[11] Choudhary, Rishabh, and Hemant Kumar Gianey. "Comprehensive review on supervised machine learning algorithms." *2017 International Conference on Machine Learning and Data Science (MLDS)*. IEEE, 2017.

[12] Samek, Wojciech, et al. "Explaining deep neural networks and beyond: A review of methods and applications." *Proceedings of the IEEE* 109.3 (2021): 247-278.

[13] Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." *Neural computation* 31.7 (2019): 1235-1270.

[14] Gers, Felix A., Nicol N. Schraudolph, and Jürgen Schmidhuber. "Learning precise timing with LSTM recurrent networks." *Journal of machine learning research* 3.Aug (2002): 115-143.

[15] Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." *International conference on machine learning*. PMLR, 2013.

[16] Elsworth, Steven, and Stefan Güttel. "Time series forecasting using LSTM networks: A symbolic approach." *arXiv preprint arXiv:2003.05672* (2020).

[17] Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." *Neural computation* 31.7 (2019): 1235-1270.

[18] Jakaša, Tina, Ivan Andročec, and Petar Sprčić. "Electricity price forecasting—ARIMA model approach." *2011 8th International Conference on the European Energy Market (EEM)*. IEEE, 2011.

[19] Contreras, Javier, et al. "ARIMA models to predict next-day electricity prices." *IEEE transactions on power systems* 18.3 (2003): 1014-1020.

[20] Meyler, Aidan, Geoff Kenny, and Terry Quinn. "Forecasting Irish inflation using ARIMA models." (1998): 1-48.

[21] Box, GEORGE EP, Gwilym M. Jenkins, and G. Reinsel. "Time series analysis: forecasting and control Holden-day San Francisco." *BoxTime Series Analysis: Forecasting and Control Holden Day1970* (1970).

[22] Contreras, Javier, et al. "ARIMA models to predict next-day electricity prices." *IEEE transactions on power systems* 18.3 (2003): 1014-1020.

[23] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997; 9 (8):1735–1780.doi: https://doi.org/10.1162/neco.1997.9.8.1735.