IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 October 2022, accepted 21 November 2022, date of publication 28 November 2022,
date of current version 1 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3224938

RESEARCH ARTICLE

Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison

UPPALA MEENA SIRISHA, MANJULA C. BELAVAGI, AND GIRIJA ATTIGERI
Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India
Corresponding author: Manjula C. Belavagi (manjula.cb@manipal.edu)

ABSTRACT Time series forecasting using historical data is significantly important nowadays. Many fields such as finance, industries, healthcare, and meteorology use it. Profit analysis using financial data is crucial for any online or offline businesses and companies. It helps understand the sales and the profits and losses made and predict values for the future. For this effective analysis, the statistical methods- Autoregressive Integrated Moving Average (ARIMA) and Seasonal ARIMA models (SARIMA), and deep learning method- Long Short- Term Memory (LSTM) Neural Network model in time series forecasting have been chosen. It has been converted into a stationary dataset for ARIMA, not for SARIMA and LSTM. The fitted models have been built and used to predict profit on test data. After obtaining good accuracies of 93.84% (ARIMA), 94.378% (SARIMA) and 97.01% (LSTM) approximately, forecasts for the next 5 years have been done. Results show that LSTM surpasses both the statistical models in constructing the best model.

INDEX TERMS Statistical methods, time series forecasting, deep learning, profit prediction, ARIMA, SARIMA, LSTM.

I. INTRODUCTION

A Sales forecast is essential for any business for decision making. It helps manage overall business. It allows efficient allocation of resources for future growth and manage its cash flow. It also helps to identify early warning signs of failure/loss before its too late for managing. It is also essential for estimating cost and revenue and predict short and long-term performances. In order to do this several Machine Learning techniques have been used. In the current work we are using time series models and comparing their performances and build an application for forecasting sales and help in decision making for any business.

A Time series is considered as a group of data points enumerated in time sequence [1]. Time series data is a group of quantities which are assembled over uniform intervals in time and ordered in a chronological fashion [2], [3]

Auto Regressive Integrated Moving Average (ARIMA) [4] explains the time series under consideration on the basis of its previous values, that is, its lags and the lagged prediction errors. It can be useful for the future forecast for a non stationary time series exhibiting patterns and is not irregular white noise. The 3 characteristic terms of ARIMA model are the parameters (p, d, q) wherein, each of the terms are the orders of the AR term, the differencing needed to change the time series into a stationary one and the MA term respectively. The term AR in ARIMA signifies that it is a linear regression model that makes use of its lags in order to predict. Linear regression models give the finest results when there is no correlation between the predictors, and they are not dependent on each other. A time series whose properties do not change over time is called stationary. For Example temperatures of specific month plotted over years. Temperatures of all the months plotted for a year is non stationary as temperatures show variation with respect to the season. For building prediction model we need stationary time series. To eliminate non stationarity from a series, commonly differencing is done.

The associate editor coordinating the review of this manuscript and approving it for publication was Sajid Ali.

Sometimes, if the time series is more complex, more than one difference operation may be necessary. Hence, the difference value "d" value is the minimum number of differencing required to turn the time series into a stationary one. The d value would be 0, if the series is already stationary. If a time series is univariate and contains trend and/or seasonal components, then Seasonal ARIMA (SARIMA) model is used. If an external predictor, known as, 'exogenous variable' is added to the SARIMA model then, it is known as the SARIMAX model [5]. In order to use an exogenous variable, the requirement is to know the variable's value during the period of forecast also.

Since time series has sequence dependence among the input variables, a great way of analysis would be to use Neural Network(NN)s that can handle the dependent properties. Recurrent NN (RNN) would be a perfect choice for the same. Long Short Term Memory (LSTM) network is one kind of RNN that is used in DL as huge datasets can be trained to obtain huge accuracies. This model has a learning mechanism to memorize and understand mapping from input variables to output variables and figures out what context deriving out of the input data is helpful to do the mapping, and could dynamically alter the context as per the necessity.

The gross profit obtained will be predicted using ARIMA, SARIMA and LSTM in Time Series Forecasting and a comparative study of the outcomes of these models is performed. These methods help in understanding the underlying context of the data points, thereby make predictions about the future values of those data points [6], [7]. The paper focuses on the following:

- To perform data collection and explore the intrinsic structure of the series
- To analyze the dataset and extract required variables
- Develop models for profit prediction
- Perform comparative analysis of ARIMA, SARIMA and LSTM models
- Forecasting for the next 5 years using the models

The paper is organized as follows: In section II literature related to time series analysis and deep learning models are discussed. In section III model building using ARIMA, SARIMA and LSTM are discussed. Subsequently in section IV result analysis is performed. Finally, Section V concludes by highlighting the work carried out.

II. LITERATURE REVIEW

In this section papers related to ARIMA, SARIMA and LSTM models have been discussed. In [2] Auto-correlation-functions(ACF) and cross-correlation functions (CCF) are used to show the relationship between lags which occurs between the time series. Authors mentioned ordinary, weighted and classical correlated least squares regression techniques. Mishra et. al. [8] presented a literature review regarding the usefulness of data science and stochastic approaches for time series forecasting. They mentioned how various researchers used ARIMA, SARIMA, vector ARIMA for variable time series and ARIMAX models to analyse the rainfall pattern. They also mentioned the use of neural networks and hybrid methods for weather forecasting. Luo et al [9] have discussed the identification of correlations between time series and event traces. ARIMA model is also used in road safety research [10]. To do this authors have integrated moving average with explanatory variables (ARIMAX). Sangare et al. [11] used analytical measures and hybrid machine learning to predict the road-traffic accidents. Almeida et al. [12] used SARIMA model to understand the traffic flow characteristics. Artificial neural network algorithms have also been proposed in [13] and [14] for the forecasting approach. The most commonly used algorithms are the Feed-Forward Neural Network (FFNN), the Long Short-Term Memory (LSTM), the Convolutional Neural Network (CNN) and a hybrid LSTM-CNN.

Ruchir et al. [15] and [16] performed stock market prediction using 10 years Bombay Stock Exchange data. They have used ARIMA, Simple Moving Average(SMA) and Holt-Winters models. The parameters considered for the evaluation are RMSE, Mean Absolute(MA) Error and MA Presentation Error. They concluded that SMA shows best performance whereas ARIMA model's performance was poor.

Permanasari et al. [17] analyzed and implemented SARIMA on time series to predict the malaria occurrences in the United States of America, based on the monthly data. Disease forecasting is important to help the stakeholders make better policies.

Owing to the increasing market and importance in the field of green energy, Alsharif et al. [18] used ARIMA and SARIMA to predict daily and month-wise mean solar radiation in the city of Seoul, respectively. This study is carried out to help the government to make changes in government policies for advancements in the fields of renewable energy.

Chen et al. [19] used the ARIMA model for the predictions of crimes related to property which includes robbery, theft, and burglary in the city of China. A period of fifty weeks of recordings of property crime was selected as the dataset. The model was trained and the predicted outcomes have been analysed and compared with the Single Exponential Smoothing (SES) and Hyperbolic Exponential Smoothing (HES). It was found out that the SES and HES gave a lesser accuracy than the ARIMA model. Dattatrayet et al. [20] conducted survey on stock market prediction techniques based on year publication, methodology and datasets used and performance metrics. They concluded that NN based and fuzzy based techniques can be effectively used for the stock market prediction. Omer Berat Sezer et al. [21] conducted a systematic review using the concept of deep learning for financial time series prediction. Various types of DL models which include DNN, RNN, DBN and CNN have been used for predicting the prices of products. They observed that CNN works better for classification when compared with deep learning models which are dependent on RNN and is suitable for static data representation. They further observed that LSTM was the best method for financial time series forecasting problems.

Siami et al. [22] investigated ARIMA and LSTM in calculating the forecasts for data belonging to financial time series and compared their error percentages. They have split up their datasets into train (70%) and test (30%) data for the accuracy of their models and observed that the prediction was improved by 85% on an average using LSTM algorithm and hence indicated that LSTM performed better compared to ARIMA.

Chniti et al. [23] used LSTM and SVR models for forecasting mobile phone costs in the Europe market. A comparison of the mentioned models has been done on uni and multivariate data and it has been found out that SVR worked better on univariate data while LSTM performed better on multivariate data, producing RSME of 35.43 and 24.718 respectively.

Srihari et al. [24] performed comparative analysis of forcasting algorithms namely ARIMA, MVFTS, CNN, LSTM and CBLSTM. They have tested the performance of these algorithms by considering various domains time series data. They concluded that performance of weighted MVFTS, ARIMA, CNN (Convolution Neural Network)s, CBLSTM was good for data considered in periods more than a couple of years.

Neural Network based method for stock market is proposed by Pang et al. [25], [26] and Jiang et al. [27]. Machine learning based stock market prediction is carried out in [16]. In [25] authors used advanced LSTM to perform real time data analysis on Internet data. They concluded that performance of the model was satisfactory for real time data. However, the model performance is poor on historical data due to limited use of text information. Comparative analysis of ARIMA and NNs models for stock market prediction is carried out in [26]. They have analysed the results based on forecast error. Based on this parameter working of both models was good. They found that performance of ARIMA model is better with respect forecast accuracy. Financial time series forecasting is carried by Sezer et al. [28]. They used image dta and extracted the technical indicators which were necessary for processing.

III. METHODOLOGY

The Sales dataset [29] has been obtained from the downloads section of efor excel.com website, and consists of around 1 million sales records, ranging over a period of 46 years (1972 – 2017). The dataset comprises of multiple variables, which are the item type, order date, shipment date, order ID, order priority (high, medium, low, cancelled), the region and the country where the orders belong to, the sales channel (online or offline), the unit price and the cost of each item type, the number of units sold per item type, the total revenue, the total cost and the gross profit made, after taxes.

Figure 1 shows the different steps involved in the process of time series forecasting. The first step is to collect the data over the period of time. The data collected may contain erroneous, incomplete or repeated data. Hence, in the next step data preprocesing is carried out by handling missing values. Once the data is ready exploratory data analysis is carried out to have better understanding of the data. Subsequently ARIMA, SARIMA and LSTM models are built for profit prediction. The models built are evaluated and visualized.

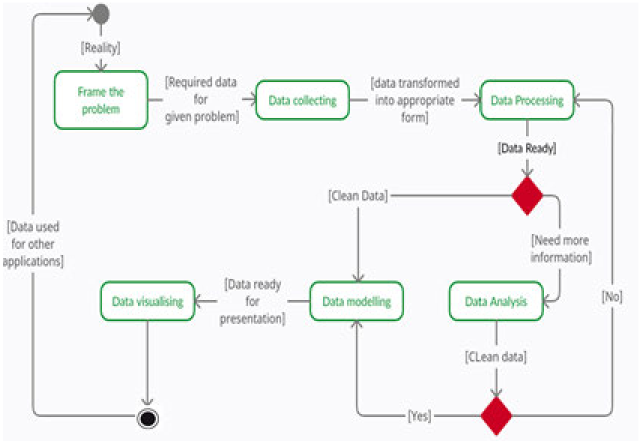


FIGURE 1. State Machine Diagram of General Flow of Time Series Prediction.

1) Data Preprocessing

The preprocessed data set is cereated by handling missing values and grouping the data. The same occurrences of the order date have been grouped together from the branches of the company in different regions and countries in the world and the profits on these order dates have been added together using the sum aggregate function. The necessary fields required for the time series analysis are also extracted and represented in specific format. The attribute "Order-Date" is converted to a datetime object and the year, month and days are extracted to perform exploratory analysis.

Year wise profit is analyzed using a scatter plot, the order dates belonging to the same year have been grouped and the mean of the profit of all the orders of the respective years are computed. A bar graph is also used to analyze yearwise mean profit. Similarly, for the next graph the order dates belonging to the same month have been grouped and the mean of the profit of all the orders of the respective months has been taken and plotted. A bar graph is plotted to analyze the monthly mean profit.

The scatter plot between the Year of order on x-axis and Profit on y-axis in Figure 2 indicates that there has been a steady increase in profit from 1972 to 2000. The profit remained almost the same till 2005 then there is a sudden fall from 2005 to 2010. After that, there has been a gradual increase from 2010 to 2017.

The bar graph shown in Figure 2 has been displayed to record the mean profit of each of the years. It is seen that there have been considerable dips in the profit in 1975, 1980, 1983, 1992, 1999, and 2009. The general trend of the dataset has been noted to increasing gradually and reaching a maxima value between 2000 and 2005, and thereafter, a fall in profit has been noticed till 2009, post which the increasing trend continued. The bar graph in Figure 3 shows the mean profit with respect to order month. The order dates belonging to

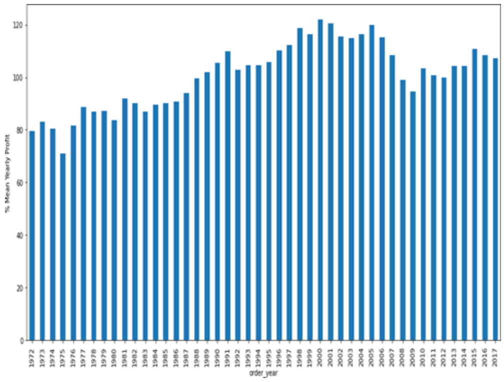


FIGURE 2. **Mean Yearly Profit vs Order Year.**

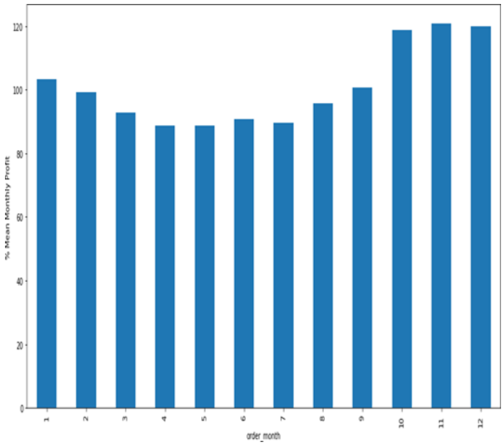


FIGURE 3. **Mean Monthly Profit vs Order Month.**

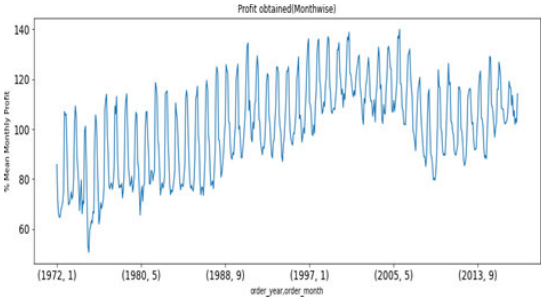


FIGURE 4. **Profit (monthwise) vs (Order Year, Order Month) plot.**

the same year are grouped month wise. The mean profit of all the orders of each month is analyzed as shown in Figure 4.

A. MODEL BUILDING USING ARIMA

In line with the process mentioned ARIMA model is built. Figure 5 shows the flow of ARIMA model. The series is checked for stationarity and if it is not several transformations are applied to make it stationary. Subsequently Auto Correlation Function (ACF), Partial ACF (PACF) graphs are plotted and the values for the terms p, d and q (model parameters) are obtained. ACF fetches auto correlation values belonging to a series with the lagged values. These values will be graphed with the confidence band to obtain the ACF plot which tells the strength of the relationship between the current value of the series with respect to its previous values. The ACF function finds the correlation depending on all of the four components of a time series, namely, trend, seasonality, cyclic and residual. PACF fetches correlation of the residuals with the next lagged value instead of finding present lagged values like the ACF. Further the model fit is performed in three stages, building AR model, the MA model and lastly combining them to obtain ARIMA. The model is used to make predictions on validation data. After this, the error and accuracy of the models are checked and evaluated.

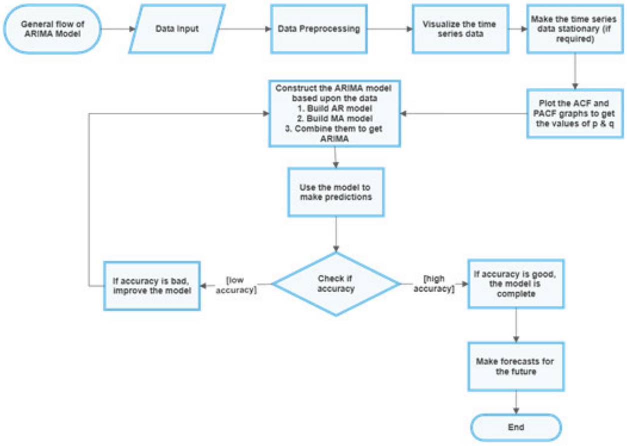

General flow of ARIMA Model
Data Input
Data Preprocessing
Visualize the time series data
Make the time series data stationary (if required)
Plot the ACF and PACF graphs to get the values of p & q
Construct the ARIMA model based upon the data
1. Build AR model
2. Build MA model
3. Combine them to get ARIMA
Use the model to make predictions
Check if accuracy
[low accuracy]
If accuracy is bad, improve the model
[high accuracy]
If accuracy is good, the model is complete
Make forecasts for the future
End

FIGURE 5. **FLow Chart of ARIMA Model.**

1) TRAINING AND VALIDATION - ARIMA MODEL

The dataframe has been split into training and validation datasets in the ratio of 4:1 (80% train and 20% valid datasets). The model was built on train data and the validation data was used in prediction to check for the accuracy. Window functions have been used to perform statistical operations on data subsets. Over every row in the DataFrame, new value can be calculated with rolling functions. A window consists of a subset of rows from the dataset and a desired calculation can be performed on these rows. A required amount of window can be specified. Window rolling mean or moving average calculation leads to an updated average value for each row in the specified window. Similarly, Window rolling standard deviation is used. The window has been chosen to be 24. The Window rolling mean and Window rolling standard deviation calculated with a window of 24 is plotted.

2) TESTS FOR STATIONARITY - ADF AND KPSS TESTS

Stationarity of the time series is analyzed by considering different statistical methods of forecasting. If the time series show it is stationary, its statistical characteristics remain constant over a time period (example: mean variance standard deviation). Hence, there would be no visible trend or seasonality. While a time series exhibiting non-stationarity is quite the opposite and these properties are time dependent.

The ADF test is a classic stochastic test for determining if the time series being used is stationary or not. According to mathematics, unit roots cause non-stationarity in a time series. This test determines the presence or non existence of a unit root. The ADF test uses two types of hypotheses namely null and alternate. The first one assumes the existence of a unit root. This implies the non-stationarity of the time series. The second one assumes the non-existence of a unit root. This implies the stationarity of the time series. Mathematically, ADF test states that: Null Hypothesis (H0): $\alpha = 1$ in the below equation, as in root is existing.

$$y_t = c + \beta t + \alpha y_{t-1} + \phi \Delta y_{t-1} + e_t \tag{1}$$

where, y_{t-1} is first lag of time series and Δy_{t-1} first difference of the series at time $t - 1$.

From this test ADF Test Statistic, p-value, the number of lags that have been made use of, the number of observations that have been made use of ADF regression and the critical values are obtained. In case the p-value is lesser than or equal to the defined Significance level of 5% (0.05) then it is concluded that the null hypothesis has been declined and stationarity of the series has been established. On the other hand, if the p-value is higher than the defined significance level and if the ADF test statistic is higher than any of the critical values, then it is weak evidence against the null hypothesis and the series is concluded to be non-stationary. The p-value obtained from performing ADF for the first time on pre-processed data is 0.338 approximately. Hence the data has no unit root and is considered to be non-stationary.

The KPSS test is also performed in order to find out if the time series is stationary around a mean or a linear trend or is not stationary because of the presence of any unit root(s).

This test is different from ADF test as its null hypothesis is exactly opposite to that of ADF's. The ADF test uses two types of hypothesis which are null and alternate, which assume that the time series under consideration is either stationary or not, respectively.

The results of the test contain KPSS Test Statistic, p-value, the number of lags that have been used and critical values. The p-value here is the probability score that helps decline the null hypothesis if it is less than 0.05, making the series non-stationary and vice versa. To decline the null hypothesis, the test statistic should also be higher than the critical values. The p value obtained from performing KPSS for the first time on preprocessed data is 0.010. Hence the data is non-stationary.

3) TRANSFORMATIONS FOR ACHIEVING STATIONARITY

ARIMA model requires stationarity. As the time series in consideration is non-stationary, differencing has to be applied to reduce trend and seasonality. The transformations are done as follows.

There is a general increasing trend in the series, except between 2005 and 2010, so a transformation that penalizes higher values more than the smaller ones has been chosen. The Logarithmic Transformation has been performed to reduce the trend, i.e., by taking the natural logarithm of the dependent variable namely Gross Profit (in thousands) from the train data.

After this, the Moving Average has been subtracted from the above. This subtraction is known to be Differencing. Since the average of 24 values has been taken by specifying a window of 24, the rolling mean has not been defined for the first 23 values. Therefore, these 23 null values have been dropped. Subsequently, after removing the moving average, it has been observed that the rolling mean and standard deviation are approximately horizontal. This has been done to remove the remaining trend and get a stationary series.

Shift transformation is also carried out where the previous value is subtracted from the current value. This also helps ensure stationarity. Thus two different transformations have been tried out namely log and time shift. For the sake of simplicity, only log scale is used because reverting back to the original scale during forecasting would be easier.

Residual is the variability left in the series after eliminating the trend and seasonality, and it cannot be explained by the model. Residual is used to build the ARIMA model, so its stationarity has been ensured.

4) AR MODEL

This model declares that the output variable depends linearly on its own previous values. The order for this model has been taken as (2,1,0), by considering q=0 as it is just AR. According to the estimated AR terms from the PACF plot, p's value was supposed to be 1 (RSME of ARIMA = 13.0421), but it resulted in RSME of the combined model ARIMA to be greater as opposed to when p's value is 2 (RSME of ARIMA = 12.5764). Using ARIMA.fit() function from statsmodels.tsa.arima model, the AR model has been fitted by maximum likelihood, i.e. building of model is done using the transformed train data.

Later the AR model fitting is carried out by maximum likelihood, i.e. building of model is done using the transformed train data.

The ARIMA.predict() function is used, which takes the fitted results of the AR model and the start and end parameters as the datetimes of the beginning and ending of the valid data and the gross profits from the valid dataset. The gross profit of valid and the predicted gross profit values vs order year has been plotted and the accuracy metrics have been displayed after scaling back.

Now, the model has to be scaled back to its original scale. So, to deal with the rolling mean transformation done earlier, cumulative sum has been performed on the predicted data, using cumsum() function. To counter the effect of log transformation, log scaling and exponential have been performed using numpy.ones()*numpy.log() (for the given indexes) and numpy.exp() respectively. To nullify the effect of differencing, the numpy.add() function has been used. The AR prediction graph and the accuracy metrics have been displayed.

5) MA MODEL

This model declares that the output variable depends linearly on the present and numerous previous values of a statistic (imperfectly predictable) term. The order for this model has been taken as (0,1,2), by considering p = 0 as it is just MA. According to the estimated MA terms from the ACF plot, q's value is taken as 2. Model is built using the transformed train data.

6) COMBINED MODEL – ARIMA MODEL

The order for this model has been taken as (2,1,2), by considering p=2,d=1,and q=2 as per the insights gathered from AR and MA models. The model has been built using the ARIMA.fit() function. Now, the model has to be scaled back to its original scale, similar to AR model.

B. MODEL BUILDING USING SARIMA

SARIMA which is an improved version of the ARIMA model which incorporates seasonal effects as well. The flow of SARIMA model is shown in the Figure 6. The series is checked for non-stationarity data as SARIMA works for such data. This model takes two kinds of orders namely order and seasonal order (p,d,q) and (P,D,Q,s). Similar to ARIMA, the order of this model consists of number of parameters of AR, order of differencing, and parameters of MA as p, d, q terms. The seasonal order consists of the seasonal element of this model for the AR units, differences, MA units, and periodicity as P, D, Q, s terms. D here has to be the integer that tells about the order of integration of the process being performed. P and Q should be integral values that indicate the orders of the AR, MA units which help in including all the lags up until that point or they can either be iterating values that give specific AR/MA lags that need to be included.

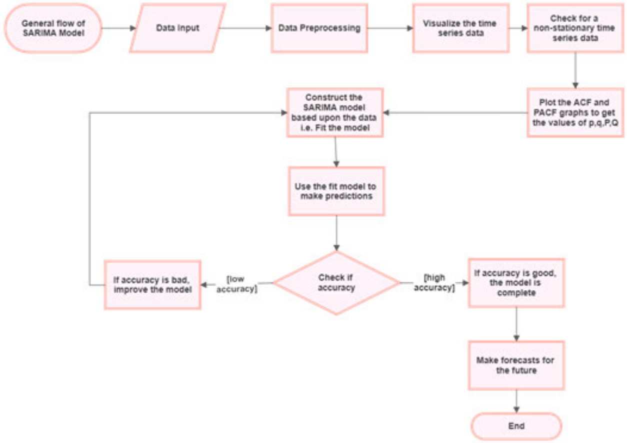

General flow of SARIMA Model
Data Input
Data Preprocessing
Visualize the time series data
Check for a non-stationary time series data
Plot the ACF and PACF graphs to get the values of p,q,P,Q
Construct the SARIMA model based upon the data i.e. Fit the model
Use the fit model to make predictions
Check if accuracy
[low accuracy]
If accuracy is bad, improve the model
[high accuracy]
If accuracy is good, the model is complete
Make forecasts for the future
End

FIGURE 6. FLow Chart of ARIMA Model.

The data splitting and other steps remains the same as ARIMA model.

C. MODEL BUILDING USING LSTM

Figure 7 shows the activity swimlane diagram of LSTM model, having three lanes. The first lane depicts taking data input subsequently data cleaning, feature extraction, performing EDA and MinMax Scaling to fit in the range of (0,1).

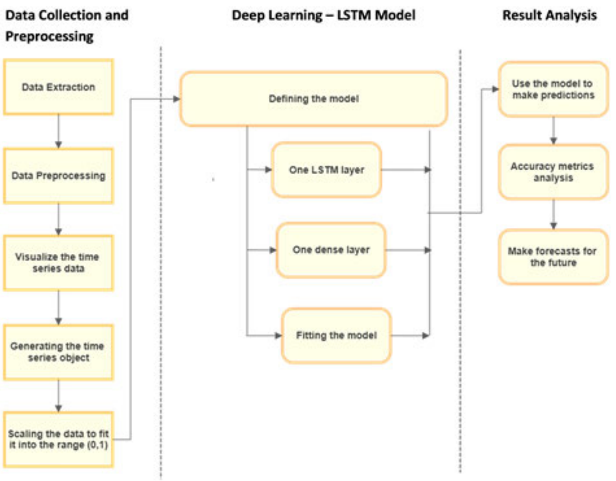

Data Collection and Preprocessing
Deep Learning – LSTM Model
Result Analysis
Data Extraction
Data Preprocessing
Visualize the time series data
Generating the time series object
Scaling the data to fit it into the range (0,1)
Defining the model
One LSTM layer
One dense layer
Fitting the model
Use the model to make predictions
Accuracy metrics analysis
Make forecasts for the future

FIGURE 7. Swimlane Diagram of LSTM Model.

The second lane, depicts implementation of LSTM model by defining the model, which consists of one layer of LSTM and a Dense each.

The third lane depicts predictions on validation data using the fitted model and evaluation. Subsequently forecasting for next 5 years.

1) DATA PREPROCESSING FOR LSTM

LSTM requires additional data preprocessing compared to stochastic models. It has been incorporated as follows.

Splicing of Data The dataset being considered here has only Order Date and Gross Profit in Thousands columns. The total number of rows of the dataset is 548. Out of these 548, Then the training and testing sets are observed and it is found that they consist of two columns or attributes which are order date and the gross profit in thousands. These parameters are normalized using MinMax scaler transform.

Each and every feature/attribute is translated completely individually so as to ensure every single value lies within the range of the training data set. This scaler is used in place of the mean and variance stabilization transformations.

The training set is fitted so that the new model is able to adapt to unknown data. Data transformations on training and testing data is performed to obtain the data in the specific range.

For the time series, LSTM model to be used on a dataset, the data has to be reorganized into sample structures containing both input and output constituents prior to fitting the data into the LSTM model. It is challenging for all of these tasks to be finished in a proper manner. TImeseiesgenerator() will embed the dataset that is being used into an object of the class Time Series Generator. This object will then be inputted straight into the NN as the dataframe that should be worked upon. Timeseriesgenerator function takes many parameters which will be discussed in detail below. First two parameters are the input and output dataset. Length parameter gives the sample's length that is to be fed into the NN to fit the model. The sampling rate is the time period that occurs between the two outputs that the model predicts with the given input values. Since the length of parameter is 12, it indicates that the generator takes in the previous twelve months values to predict the next one month's profit, as the batch size is 1. The values generated by this function are stored in an object named as generator, having two columns which consists of an array of the lags and an array of the predicted value.

2) TRAINING AND TESTING OF LSTM

This includes defining, and fitting the model, making predictions on test data, and finally, forecasts.

LSTM is specifically designed for removing the long term dependency problem which means they find it easy to remember the information that is given to them for a very long period of time. The sequential model can help in easily stacking up of layers of the NN on top of each other and does not depend on the exact shape of the tensors or the layers in each model. Next the Sequential constructor is created for this model. This model consists of the one single visible layer, a hidden layer consisting of 100 LSTM neurons and the output layer which is used to predict the future profits. A batch size of 1 and 20 epochs are used in the training of this model. Verbose is set to 1 which means that the progress of the training of every epoch with an animated progress bar is shown. The LSTM neurons require a sigmoid activation function. Here the batch size means the number of samples from the training dataset that are used for a single iteration. Epochs tells us the number of iterations of the training data set that the LSTM model has completed. Since the amount of data in the data set is usually very huge the data is divided into batches for easier processing. The loss function used here is mean squared error loss which comes under the category of regression loss functions. This loss specifically calculates the differences between the squares of the profit attribute in training dataset and the profit attribute in the predicted datasets. The lower the mean squared error value the more accurate the model is because the predicted values are very close to the actual or training values.

The optimizer used here is Adam. Which is very fast in computation and it optimizes the weights in every level. The metrics used for evaluating the model fit is accuracy. This metric comes under the accuracy class. It computes the number of times the predictions equal to the existing values. Rectifiedlinear(Relu) activation function will activate the node and give the output directly if the output is positive and directly output 0 if the immediate output is otherwise. The benefits of this function are that the model is very easy to train on this model and more often than not attain great performance. For training the network's gradient descent functions need to be used which allow for the feedback of the errors. A nonlinear function which can permit the establishment of complicated relations between the neurons. For giving more sensitivity to the added input activation and to evade from saturating the neuron, relu is used. Subsequently model fitting is done.

Subsequently predictions on test data is carried out.

IV. RESULT ANALYSIS

In this section sales forecasting using of ARIMA, SARIMA and LSTM models is discussed.

A. RESULT ANALYSIS OF ARIMA MODEL

Window size for MA Forecast is chosen to be 24 because it gives the least possible error for RSME and MAPE and best possible accuracy of 91.671%, approximately as shown in the Table 1. The Window rolling mean and Window rolling standard deviation calculated with a window of 24 have been plotted as shown in Figure 8 and they seem to be varying a lot. *Rolling Window Size (RWS) The p-value obtained from performing ADF for the first time on pre-processed data as shown in Figure 9 is 0.338 approximately. Hence the data has no unit root and is considered to be non-stationary. While the p value obtained from performing KPSS for the first time on pre-processed data is 0.010. Hence the data is non-stationary. So, in the next step transformations are applied. By taking the natural logarithm of the dependent variable, i.e., Gross Profit in thousands in train data, the Logarithmic transformation has

TABLE 1. Rolling window size comparison.

RWS	RSME	MAPE	MAPA (Accuracy %)
5	14.4374196	0.107453364	89.25466363
12	11.84809209	0.086463782	91.35362177
24	11.40494076	0.083290455	91.67095453
50	12.27557348	0.089923931	91.00760693

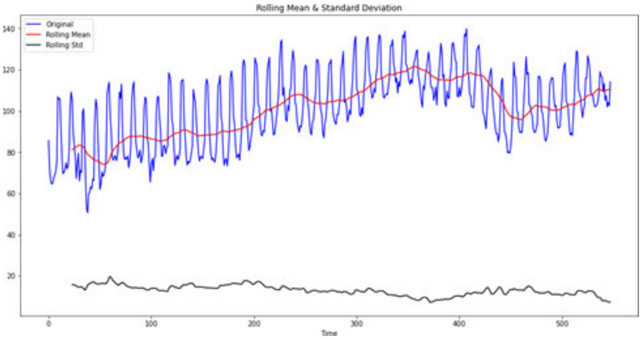
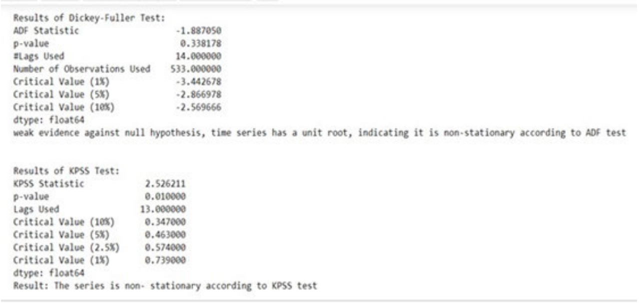


FIGURE 8. Window (24) Rolling Mean and Standard Deviation vs Time plot.



```
Results of Dickey-Fuller Test:
ADF Statistic                  -1.887050
p-value                         0.338178
#Lags Used                     14.000000
Number of Observations Used   533.000000
Critical Value (1%)            -3.442678
Critical Value (5%)            -2.866978
Critical Value (10%)           -2.569666
dtype: float64
weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary according to ADF test

Results of KPSS Test:
KPSS Statistic           2.526211
p-value                  0.010000
Lags Used               13.000000
Critical Value (10%)     0.347000
Critical Value (5%)      0.463000
Critical Value (2.5%)    0.574000
Critical Value (1%)      0.739000
dtype: float64
Result: The series is non- stationary according to KPSS test
```

FIGURE 9. ADF, KPSS test results for preprocessed data.

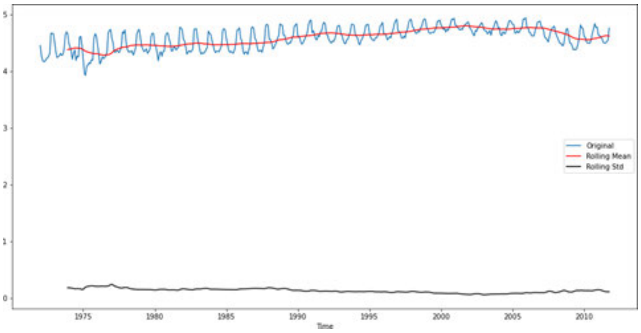


FIGURE 10. Log transformations.

been performed as shown in Figure 10. It can be understood that the rate at which the rolling mean is increasing has been lowered and the variance has been stabilized.

Now, after removing the moving average, it has been observed in Figure 11 that the rolling mean and standard deviation are approximately horizontal. That is the mean of the series has been stabilized by Differencing the series. This stabilization has been done to ensure stationarity. From the ADF test results, the p-value obtained was 0.05, thus the series is rendered as stationary as shown in Figure12.

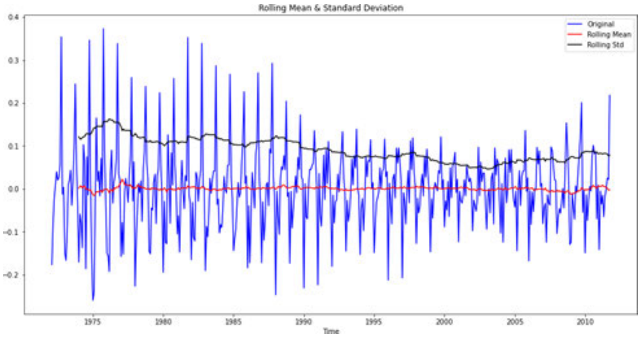
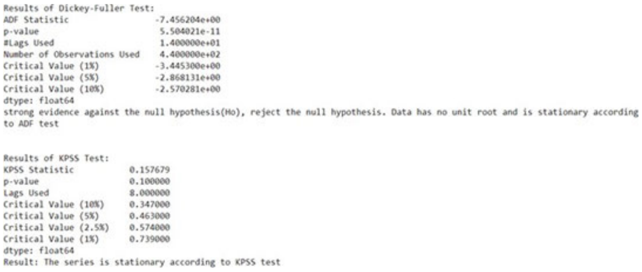


FIGURE 11. Differencing.



```
Results of Dickey-Fuller Test:
ADF Statistic                  -7.456204e+00
p-value                         5.504021e-11
#Lags Used                      1.400000e+01
Number of Observations Used     4.400000e+02
Critical Value (1%)            -3.445300e+00
Critical Value (5%)            -2.868131e+00
Critical Value (10%)           -2.570281e+00
dtype: float64
strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data has no unit root and is stationary according
to ADF test

Results of KPSS Test:
KPSS Statistic            0.157679
p-value                   0.100000
Lags Used                 8.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64
Result: The series is stationary according to KPSS test
```

FIGURE 12. Stationarity after Transformations.

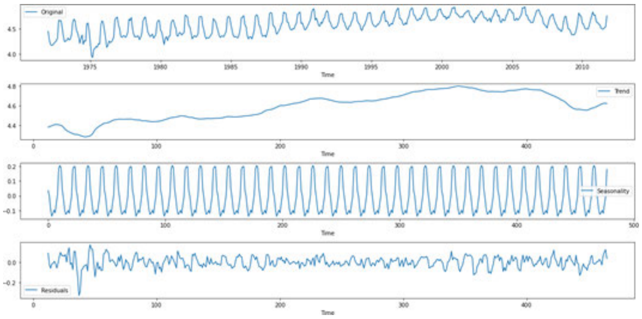


FIGURE 13. Seasonality Decomposition.

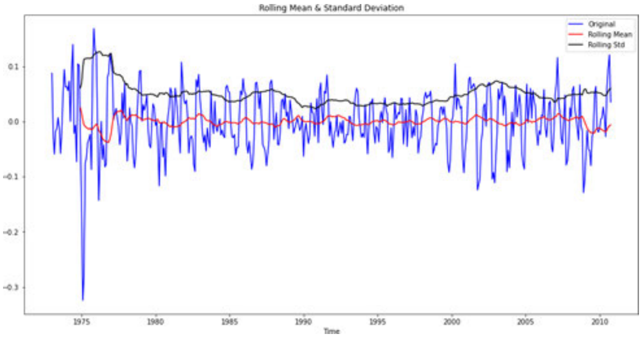
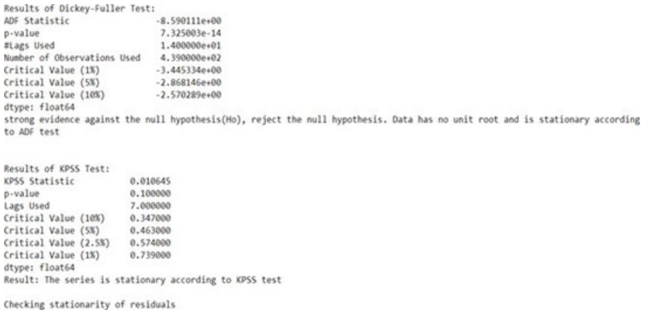


FIGURE 14. Rolling Mean and Std. Deviation of Residuals.



```
Results of Dickey-Fuller Test:
ADF Statistic                  -8.590111e+00
p-value                         7.325003e-14
#Lags Used                      1.400000e+01
Number of Observations Used     4.390000e+02
Critical Value (1%)            -3.445334e+00
Critical Value (5%)            -2.868146e+00
Critical Value (10%)           -2.570289e+00
dtype: float64
strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data has no unit root and is stationary according
to ADF test

Results of KPSS Test:
KPSS Statistic            0.010645
p-value                   0.100000
Lags Used                 7.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64
Result: The series is stationary according to KPSS test

Checking stationarity of residuals
```

FIGURE 15. Rolling Mean and Std. Deviation of Residuals.

Seasonal decompose is used to break up the time series data into trend component, seasonality part, level and the residual as shown in Figure 13. Residual is used to build the ARIMA model, so its stationarity is examined. Its rolling mean and standard deviation have been checked and shown in Figure 14. Figure 15 shows ADF and KPSS test results as stationary. Figure 16 shows ACF and PACF plots. ‘‘p’’ term is estimated from the PACF plot, there is only 1 lollipop above the confidence interval (blue region) at lag=1, before the next one at lag=2 falls into the confidence interval. The value at lag 0 is ignored as it always shows perfect correlation by default. Hence, p should be 1. ‘‘q’’ term is estimated from the ACF plot, there are 2 values above the confidence interval (blue region) at lags 1 and 2 that are quite significant, before the next one falls below the confidence interval. The value at lag 0 is ignored as it always shows perfect correlation by default. Hence, q should be 2.

I term, or d value is the order of differencing. Only log difference is performed. Hence, the d value is 1.

In Figure17, the AR model has been fitted by maximum likelihood, i.e. building of model is done using the transformed train data. The actual vs predicted results on validation data is shown in Figure 18, after it has been scaled

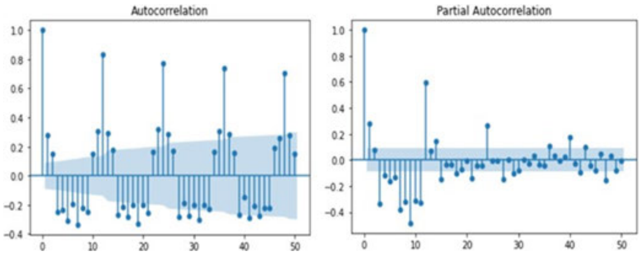


FIGURE 16. ACF and PACF plots of ARIMA.

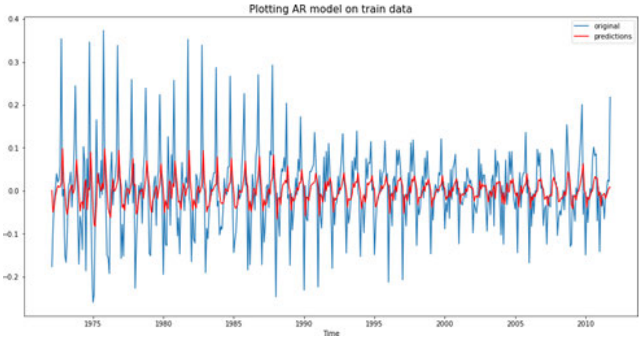


FIGURE 17. Plotting AR Model on Train Data.

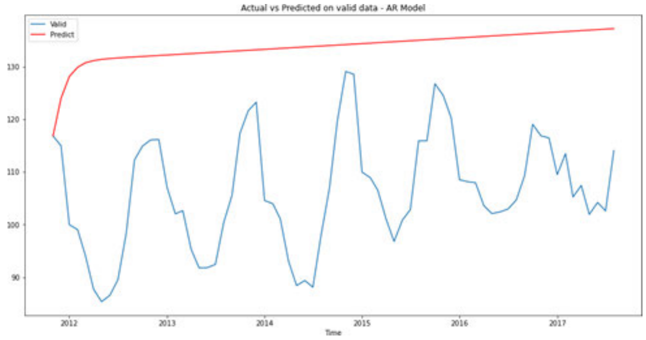


FIGURE 18. Actual vs Predicted on Validation Data – AR Model.

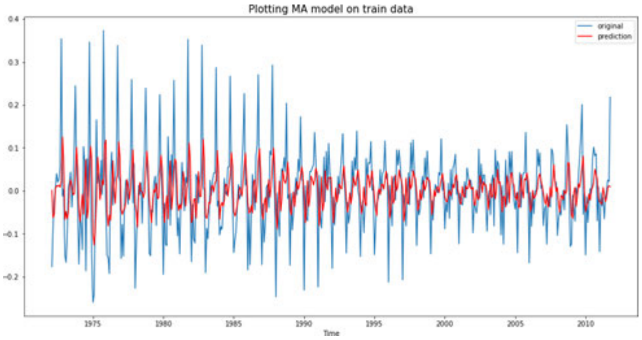


FIGURE 19. Plotting MA Model on Train Data.

back to original scale. In Figure 19, the gross profit of valid and the predicted gross profit values vs order year is plotted. Subsequently, the model is scaled back to its original scale as shown in Figure 20, similar to AR model.

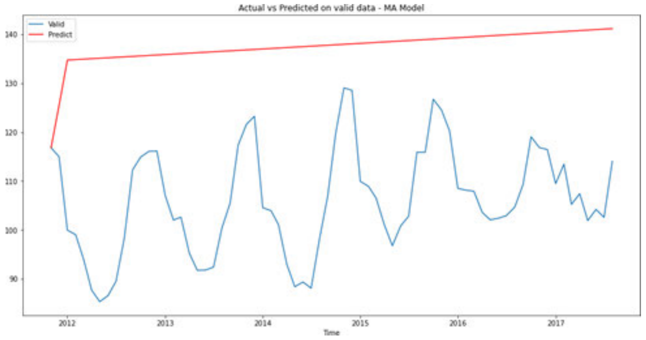


FIGURE 20. Actual vs Predicted on Validation Data – MA Model.

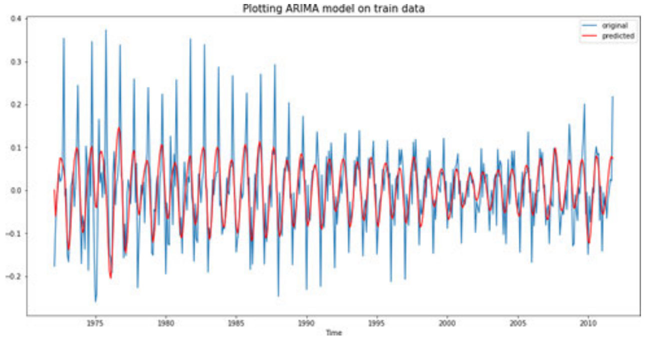


FIGURE 21. Plotting ARIMA Model on Train Data.

The order for the combined ARIMA model has been taken as (2,1,2), by considering p=2,d=1,and q=2 as per the insights gathered from AR and MA models. The model has been built using the ARIMA.fit() function and can be seen in Figure 21) Now, the model has to be scaled back to its original scale in Figure 22, similar to AR model.

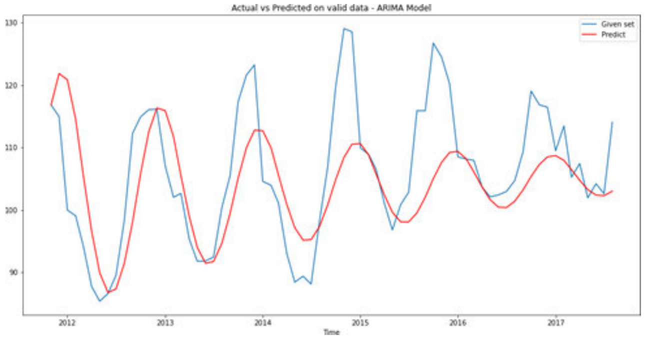


FIGURE 22. Actual vs Predicted on Validation Data – ARIMA Model.

B. RESULT ANALYSIS OF SARIMA MODEL

Inferences from the ACF and PACF lollipop charts shown in Figures 23. p term is estimated from the PACF plot, there are 2 lollipops below the confidence interval (blue region) at lags 3 and 4, before the next lag at 2 falls above the confidence interval. The value at lag 0 is ignored as it always shows perfect correlation by default. Hence, p should be 2. q term is estimated fom the ACF plot, there are 4 values above the

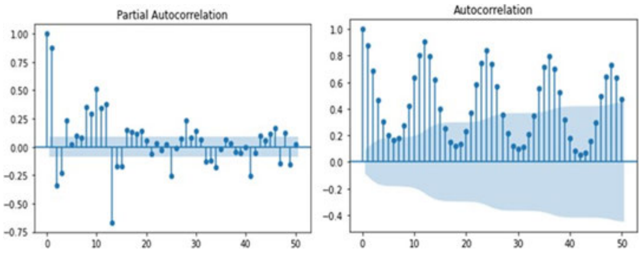


FIGURE 23. ACF and PACF plots for SARIMA.

confidence interval (blue region) at lags 1, 2, 3 and 4 that are quite significant, before the next one falls into the confidence interval. The value at lag 0 is ignored as it always shows perfect correlation by default. Hence, q should be 4. *d* value is the order of integration. Hence, d value is 1. With respect to the seasonal terms, the plots show expected behaviour's with unexpected spikes at certain lags. So, it has been hypothesized that P = 0 and Q = 1, owing to the tapering auto correlation function. These values have been checked when they are applied to the model.

The ARIMA.predict() function is used, which takes the fitted results of the MA model and the start and end parameters as the datetimes of the beginning and ending of the valid data and the gross profits from the valid dataset. The gross profit of valid and the predicted gross profit values vs order year has been plotted as shown in Figure 23 and the accuracy metrics have been displayed. Now, the model has to be scaled back to its original scale, similar to AR model. The order and seasonal order have been taken as (2,1,4) and (0,1,1,7). The Sarimax.fit() function has been used on the training dataset to build the model. The predict() function has been used on the fitted model to make the SARIMAX prediction for validation dataset. These have been displayed below in Figure 24.

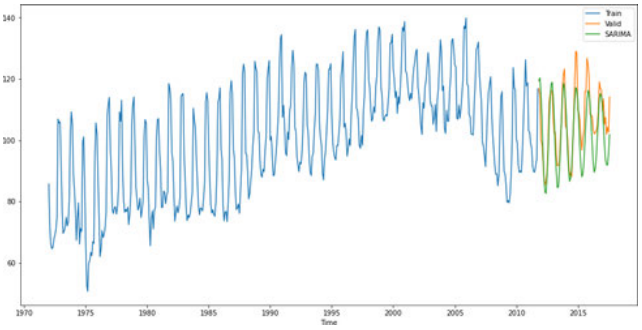


FIGURE 24. SARIMA Model fit() and predictions on Valid Data.

The plot diagnostics have been displayed in Figure 25. To determine the validity of fit of the model, its residuals errors should have almost constant mean and variance. From the Standardized Residual graph, the residual errors appear to vary around a mean of zero and have a uniform variance. This indicates an unbiased forecast. The Histogram plus estimated density graph, known as the density plot, suggests a normal distribution having a mean of zero. The Normal Q-Q plot shows almost all the dots falling in line with the red line, which means that the distribution is proper and not skewed. The Correlogram, also known as the Auto correlation Function (ACF) plot or lag plot, indicates that the residuals are not auto-correlated at lag 1. If correlations exit among residuals, it means there is unexplored data left in the residuals that must be considered for the purpose of forecasting. Then, a need arises to search for more exogenous variables for SARIMA. Hence, these plots indicates that the fit is good and can be used for forecasting.

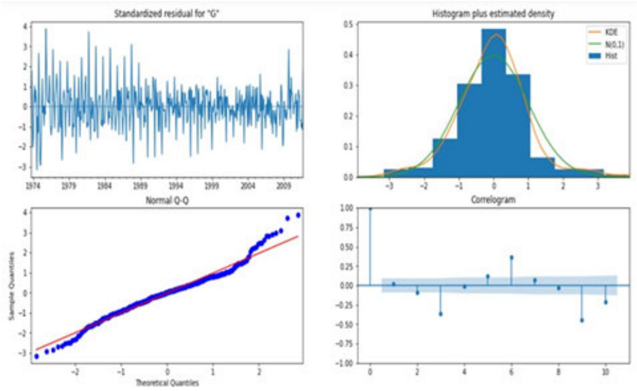
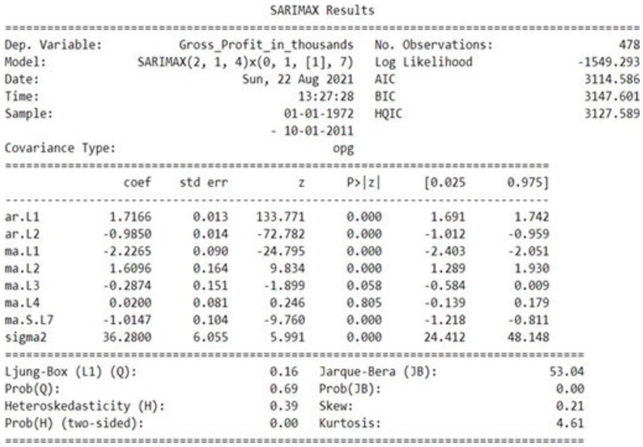


FIGURE 25. Plot Diagnostics of SARIMA.

The summary() function displayed the SARIMAX Results in Figure 26. It is evident that the value of AIC, as well as the P values belonging to the coefficients estimated by the model looks significant.



```
                                      SARIMAX Results
==========================================================================================
Dep. Variable:                 Gross_Profit_in_thousands   No. Observations:           478
Model:              SARIMAX(2, 1, 4)x(0, 1, [1], 7)   Log Likelihood           -1549.293
Date:                             Sun, 22 Aug 2021   AIC                       3114.586
Time:                                     13:27:28   BIC                       3147.601
Sample:                                 01-01-1972   HQIC                      3127.589
                                      - 10-01-2011
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.7166      0.013    133.771      0.000       1.691       1.742
ar.L2         -0.9850      0.014    -72.782      0.000      -1.012      -0.959
ma.L1         -2.2265      0.090    -24.795      0.000      -2.403      -2.051
ma.L2          1.6096      0.164      9.834      0.000       1.289       1.930
ma.L3         -0.2874      0.151     -1.899      0.058      -0.584       0.009
ma.L4          0.0200      0.081      0.246      0.805      -0.139       0.179
ma.S.L7       -1.0147      0.104     -9.760      0.000      -1.218      -0.811
sigma2        36.2800      6.055      5.991      0.000      24.412      48.148
===================================================================================
Ljung-Box (L1) (Q):                   0.16   Jarque-Bera (JB):                 53.04
Prob(Q):                              0.69   Prob(JB):                          0.00
Heteroskedasticity (H):               0.39   Skew:                              0.21
Prob(H) (two-sided):                  0.00   Kurtosis:                          4.61
===================================================================================
```

FIGURE 26. SARIMAX Results by summary() function.

C. RESULT ANALYSIS OF LSTM MODEL

The losses and accuracies of the train and test datas have been plotted using plot() and the model.history.history. This is useful to know how the model has converged. It is seen from the plot that the losses for train and val loss have reached their minimum at epoch=2, as seen in Figure 27. The predictions made after all the inverse transformations have been printed for the span of 2016-17 in Figure 28. It shows

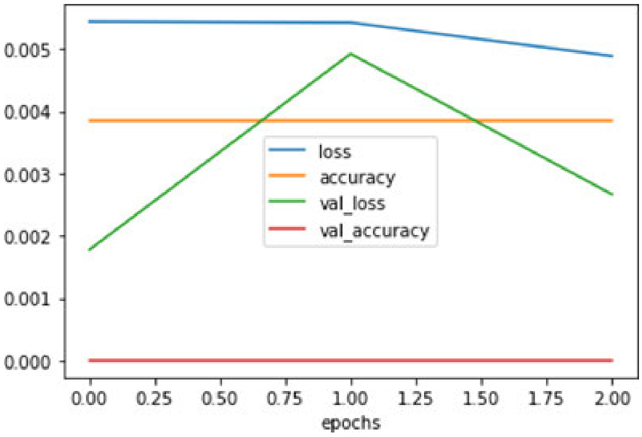


FIGURE 27. LSTM Model losses and accuracies of train and test data.

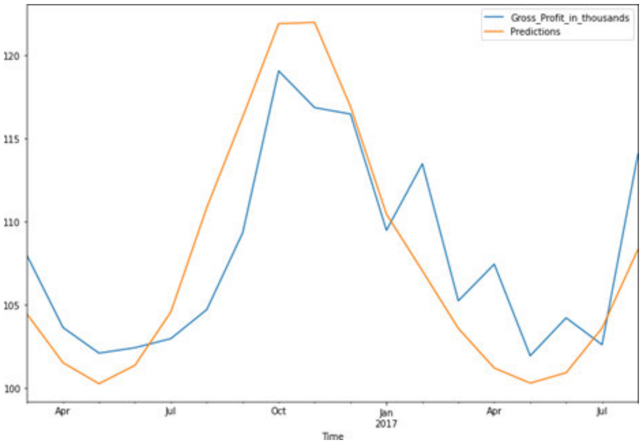


FIGURE 28. Predictions screenshot of 2016-17.

that the predictions are almost in line with the actual data and the model built has understood the dataset well.

1) COMPARISON OF RESULTS

From Table 2, it has been understood that the AR and MA models when combined to get the ARIMA model produce an accuracy of 93.840% approximately.

TABLE 2. Accuracy metrics for AR, MA model predictions.

Model	AR	MA
RSME	29.65521352	33.26912075
ME	-27.60785087	-31.38192957
MPE	-0.272996613	-0.309043979
MAE	27.60785087	31.38192957
MAPE	0.206260488	0.227875589
Corr	0.101832397	0.059874417
MinMax Error	0.377903407	0.395349002
Accuracy % (MAPA)	79.37 %	77.21 %

Following observations are made from the Table3. and their Significance:

- It is observed that SARIMA has higher accuracy than that of ARIMA because the seasonal constituents (trends and seasonality), that were removed in ARIMA, have been taken into consideration to make a more realistic prediction on valid data. On the other hand, LSTM surpasses both the stochastic models, as expected.
- Additionally, a positive corr value above 0.6 can be seen in all the three cases, and it indicates a rather good positive relation between profit and time, and this in turn explains the increasing trend of the data considered, as time progresses.
- The accuracies have been computed based on MAPA and MAPE (MAPA % = (1 – MAPE) * 100) because it is a percentage metric, hence enhances easier interpretation compared to RSME.

TABLE 3. Comparison of accuracy metrics of all 3 models.

Model	ARIMA	SARIMA	LSTM
RSME	8.681204196	7.274401885	3.917264522
ME	2.135885345	3.93202283	0.470382847
MPE	0.030512648	0.036229118	0.004556761
MAE	6.481059891	6.010790505	3.257199791
MAPE	0.061593781	0.056216104	0.029891568
Corr	0.622249004	0.84321785	0.840131571
MinMax Error	0.338681671	0.35963917	0.178130191
Accuracy % (MAPA)	93.84%	94.38%	97.01%

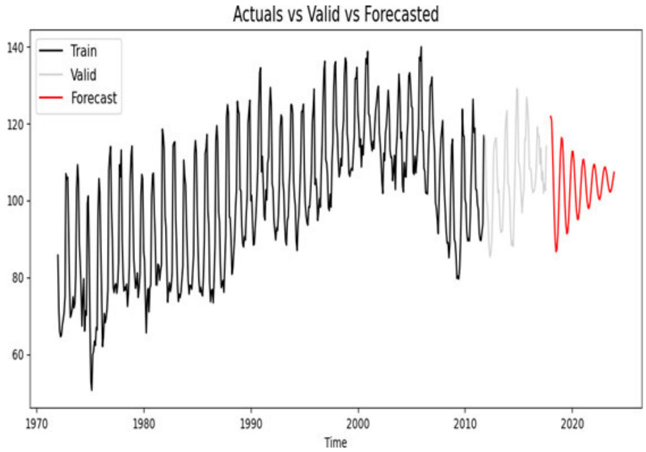


FIGURE 29. Future Forecast – ARIMA Model.

As the models are giving good accuracy, the forecast for the next 5 years has been made for each of them as follows. Observations from the below figures:

Figures 29 and 30: ARIMA, SARIMA forecast the profit with a gradual decreasing trend over time.

Figure 31: LSTM forecasts profit with a sudden, but gradual decreasing trend over time.

V. CONCLUSION AND FUTURE SCOPE

Profit analysis helps to understand the sales and the profits and losses made and predict values for the future In the current work this is carried out on sales data using the statistical method-Autoregressive Integrated Moving Average (ARIMA) and Seasonal ARIMA models (SARIMA), and deep learning method- Long Short- Term Memory (LSTM) Neural Network model in time series forecasting. It has been converted into a stationary dataset for ARIMA, not

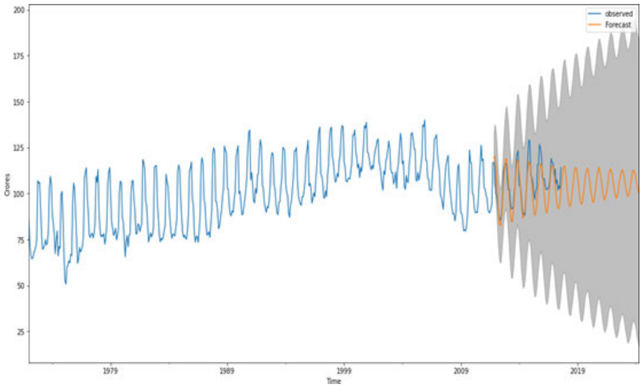


FIGURE 30. Future Forecast – SARIMA Model.

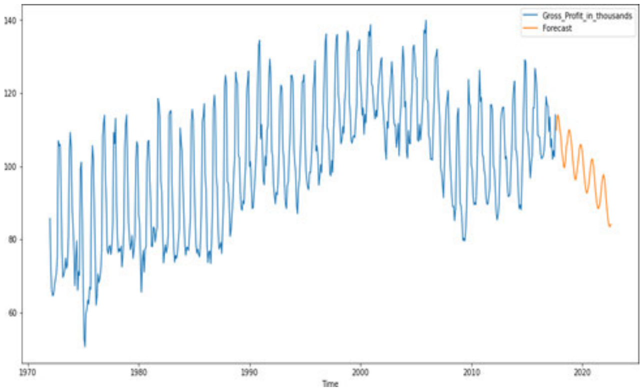


FIGURE 31. Future Forecast – LSTM Model.

for SARIMA and LSTM. The fitted models have been built and used to predict profit on test data. Accuracies of 93.84% (ARIMA), 94.378% (SARIMA) and 97.01% (LSTM) approximately are observed. Using the models built forecasts for the next 5 years have been done. Results show that LSTM surpasses both the statistical models in constructing the best model.

LSTM surpasses both the stochastic models in constructing the best model, but it is expensive in terms of runtime and computational capability if the data used, and the number of iterations required are huge. As it provides only around 3% betterment in accuracy, it can be replaced by SARIMA for dataset that is larger and not very complex but contains seasonality. It has been uncovered that the number of epochs used do not influence the accuracy of LSTM, as it increases or decreases randomly with epochs. Hence it is best to stop at minimum epochs once a decent accuracy is achieved. The accuracy of the future forecasts decreases as more time elapses from the last known data point. Various new DL models can be tried in the future. Also, combinations of stochastic and DL models can be implemented to obtain more benefits, depending on the data. Also, we can develop full fledged web application or mobile application for sales forecast which can help business decision making as a whole.

REFERENCES

[1] C. Chatfield, *Time-Series Forecasting*. Boca Raton, FL, USA: CRC Press, 2000.

[2] R. H. Shumway, D. S. Stoffer, and D. S. Stoffer, *Time Series Analysis and Its Applications*, vol. 3. New York, NY, USA: Springer, 2000.

[3] H. Li, "Time-series analysis," in *Numerical Methods Using Java: For Data Science, Analysis, and Engineering*. Hong Kong: O'Reilly, 2022, pp. 979–1172.

[4] Y. Takahashi, H. Aida, and T. Saito, "ARIMA model's superiority over f-ARIMA model," in *Proc. Int. Conf. Commun. Technol. (WCC-ICCT)*, vol. 1, 2000, pp. 66–69.

[5] N. Deretić, D. Stanimirović, M. A. Awadh, N. Vujanović, and A. Djukić, "SARIMA modelling approach for forecasting of traffic accidents," *Sustainability*, vol. 14, no. 8, p. 4403, Apr. 2022.

[6] K. Mokhtar, S. M. M. Ruslan, A. A. Bakar, J. Jeevan, and M. R. Othman, "The analysis of container terminal throughput using ARIMA and SARIMA," in *Design in Maritime Engineering*. Cham, Switzerland: Springer, 2022, pp. 229–243.

[7] T. Falatouri, F. Darbanian, P. Brandtner, and C. Udokwu, "Predictive analytics for demand forecasting—A comparison of SARIMA and LSTM in retail SCM," *Proc. Comput. Sci.*, vol. 200, pp. 993–1003, Jan. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922003076

[8] N. Mishra and A. Jain, "Time series data analysis for forecasting—A literature review," *Int. J. Mod. Eng. Res.*, vol. 4, no. 7, pp. 1–5, 2014.

[9] C. Luo, J.-G. Lou, Q. Lin, Q. Fu, R. Ding, D. Zhang, and Z. Wang, "Correlating events with time series for incident diagnosis," in *Proc. KDD*. New York, NY, USA: Association for Computing Machinery, Aug. 2014, pp. 1583–1592.

[10] C. C. Ihueze and U. O. Onwurah, "Road traffic accidents prediction modelling: An analysis of Anambra State, Nigeria," *Accident Anal. Prevention*, vol. 112, pp. 21–29, Mar. 2018.

[11] M. Sangare, S. Gupta, S. Bouzefrane, S. Banerjee, and P. Mühlethaler, "Exploring the forecasting approach for road accidents: Analytical measures with hybrid machine learning," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 113855. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03119076

[12] A. Almeida, S. Brás, I. Oliveira, and S. Sargento, "Vehicular traffic flow prediction using deployed traffic counters in a city," *Future Gener. Comput. Syst.*, vol. 128, pp. 429–442, Mar. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21004180

[13] I. O. Olayode, L. K. Tartibu, and M. O. Okwu, "Prediction and modeling of traffic flow of human-driven vehicles at a signalized road intersection using artificial neural network model: A South African road transportation system scenario," *Transp. Eng.*, vol. 6, Dec. 2021, Art. no. 100095. [Online]. Available: https://www.sciencedirect.com/science/article/pii/

[14] M. A. Rahim and H. M. Hassan, "A deep learning based traffic crash severity prediction framework," *Accident Anal. Prevention*, vol. 154, May 2021, Art. no. 106090. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457521001214

[15] M. Kulkarni, A. Jadha, and D. Dhingra, "Time series data analysis for stock market prediction," in *Proc. Int. Conf. Innov. Comput. Commun. (ICICC)*, 2020, pp. 1–6.

[16] G. V. Attigeri, M. M. M. Pai, R. M. Pai, and A. Nayak, "Stock market prediction: A big data approach," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2015, pp. 1–5.

[17] A. E. Permanasari, I. Hidayah, and I. A. Bustoni, "SARIMA (seasonal ARIMA) implementation on time series to forecast the number of malaria incidence," in *Proc. Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2013, pp. 203–207.

[18] M. Y. M. Alsharif and J. Kim, "Time series ARIMA model for prediction of daily and monthly average global solar radiation: The case study of Seoul, South Korea," *Symmetry*, vol. 11, no. 2, pp. 1–17, 2019.

[19] P. Chen, H. Yuan, and X. Shu, "Forecasting crime using the ARIMA model," in *Proc. 5th Int. Conf. Fuzzy Syst. Knowl. Discovery*, Oct. 2008, pp. 627–630.

[20] D. P. Gandhmal and K. Kumar, "Systematic analysis and review of stock market prediction techniques," *Comput. Sci. Rev.*, vol. 34, Nov. 2019, Art. no. 100190. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157401371930084X

[21] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Appl. Soft Comput.*, vol. 90, May 2020, Art. no. 106181. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620301216

[22] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 1394–1401.

[23] G. Chniti, H. Bakir, and H. Zaher, "E-commerce time series forecasting using LSTM neural network and support vector regression," in *Proc. Int. Conf. Big Data Internet Thing (BDIOT)*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 80–84, doi: 10.1145/3175684.3175695.

[24] S. Athiyarath, M. Paul, and S. Krishnaswamy, "A comparative study and analysis of time series forecasting techniques," *Social Netw. Comput. Sci.*, vol. 1, no. 3, pp. 1–7, May 2020.

[25] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *J. Supercomput.*, vol. 76, no. 3, pp. 2098–2118, Mar. 2020.

[26] A. A. Ariyo, A. A. Oluyinka, and A. C. Korede, "Applications of deep learning in stock market prediction: Recent progress," *J. Appl. Math.*, vol. 1, no. 1, pp. 1–22, 2014.

[27] W. Jiang, "Applications of deep learning in stock market prediction: Recent progress," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115537, doi: 10.1016/j.eswa.2021.115537.

[28] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach," *Appl. Soft Comput.*, vol. 70, pp. 525–538, Sep. 2018.

[29] *Data Sets for Testing (Till 5 Million Records)—Sales*. Accessed: Feb. 5, 2021. [Online]. Available: https://excelbianalytics.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/

UPPALA MEENA SIRISHA received the B.Tech. degree in computer and communication engineering from the Manipal Institute of Technology, Manipal, India, in 2021. She has finished internships as a Junior Front End Developer at Modulus Motors Pvt. Ltd., as a SDE Intern at Vizag Steel Plant, BlackRock Services India Pvt. Ltd., and as a Research Analyst at Mitti (NGO). She is the Co-Founder and Current Board Member of Mudra-Imprint—a social service organization. Her research interests include machine learning, database and management systems, and computer networking.

MANJULA C. BELAVAGI received the B.E. degree in computer science and engineering from Karnatak University, Dharwad, India, the master's degree in network and internet engineering from JNNCE, Shivamogga, VTU, Belgaum, India, and the Ph.D. degree from the Manipal Academy of Higher Education, Manipal, India. She is currently working as an Assistant Professor-Selection Grade with the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal. She has published research papers in national and international conference proceedings and journals. Her research interests include machine learning, game theory, and wireless sensor networks security.

GIRIJA ATTIGERI received the B.E. and M.Tech. degrees from the Visvesvaraya Technological University, Karnataka, India, and the Ph.D. degree from the Manipal Institute of Technology, Karnataka. She is currently an Associate Professor with the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. She has more than 15 years of experience in teaching and research. She has around ten publications in reputed international conferences and journals. Her research interests include big data analytics, machine learning, and data science.