A NOVEL DEEP-LEARNING BASED APPROACH FOR TIME SERIES FORECASTING USING SARIMA, NEURAL PROPHET AND FB PROPHET

Khulood Albeladi[1]*, Bassam Zafar[2], Ahmed Mueen[3]

[1]* Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
Email: kabdulghanialbeladi@stu.kau.edu.sa; ORCID: https://orcid.org/0009-0002-4651-5539

[2]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
Email: bzafar@kau.edu.sa; ORCID: https://orcid.org/0000-0001-5083-1548

[3]Department of Computer and Information Technology, Faculty of Applied Studies, King Abdulaziz University, Jeddah, Saudi Arabia
Email: mueen@kau.edu.sa; ORCID: https://orcid.org/0000-0003-3579-8107

*Corresponding Author: Khulood Albeladi, Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
Email: kabdulghanialbeladi@stu.kau.edu.sa; ORCID: https://orcid.org/0009-0002-4651-5539

ABSTRACT

Objective: The article aims to explore and evaluate a novel deep-learning approach for time series forecasting using three specific models: SARIMA (Seasonal Auto-Regressive Integrated Moving Average), Neural Prophet, and Facebook Prophet. The primary goal is to assess the effectiveness of these models in predicting stock market values in the Gulf region, providing insights into the best-suited models for forecasting tasks.

Methods: The study employs Python libraries and frameworks to implement the SARIMA, Neural Prophet, and Facebook Prophet models. The models are trained using stock market data from the Mulkia Gulf Real Estate dataset. The methodology includes data preprocessing, model training, evaluation, and comparison using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

Results: The evaluation results show that SARIMA performs well in general prediction tasks, especially when datasets contain seasonality trends. Facebook Prophet excels with smaller datasets containing seasonal data, while Neural Prophet demonstrates its ability to capture complex, non-linear patterns. However, Neural Prophet requires more intricate data and fine-tuning for optimal results.

Contribution: This study provides a comparative analysis of deep-learning models for time series forecasting, offering valuable insights into their strengths and weaknesses. The findings contribute to the understanding of which models are most suitable for stock market prediction and how they can be adapted to different data types and scenarios.

Conclusion: The research concludes that each model—SARIMA, Facebook Prophet, and Neural Prophet—has its unique strengths in time series forecasting. SARIMA is reliable for handling seasonal data, Facebook Prophet is efficient for smaller datasets with clear trends, and Neural Prophet is best for more complex datasets. The study highlights the importance of selecting the appropriate model based on the specific requirements of the forecasting task.

UMA NOVA ABORDAGEM BASEADA EM APRENDIZADO PROFUNDO PARA PREVISÃO DE SÉRIES TEMPORAIS USANDO SARIMA, NEURAL PROPHET E FB PROPHET

RESUMO

Objetivo: O artigo tem como objetivo explorar e avaliar uma nova abordagem baseada em aprendizado profundo para previsão de séries temporais, utilizando três modelos específicos: SARIMA (Média Móvel Integrada Autoregressiva Sazonal), Neural Prophet e Facebook Prophet. O principal objetivo é avaliar a eficácia desses modelos na previsão de valores do mercado de ações na região do Golfo, fornecendo insights sobre os modelos mais adequados para tarefas de previsão.

Métodos: O estudo utiliza bibliotecas e frameworks do Python para implementar os modelos SARIMA, Neural Prophet e Facebook Prophet. Os modelos são treinados com dados do mercado de ações do conjunto de dados do Mulkia Gulf Real Estate. A metodologia inclui pré-processamento de dados, treinamento de modelos, avaliação e comparação usando métricas como Erro Quadrático Médio (MSE), Raiz do Erro Quadrático Médio (RMSE) e Erro Percentual Médio Absoluto (MAPE).

Resultados: Os resultados da avaliação mostram que o SARIMA apresenta um bom desempenho em tarefas de previsão gerais, especialmente quando os conjuntos de dados contêm tendências sazonais. O Facebook Prophet se destaca com conjuntos de dados menores que possuem dados sazonais, enquanto o Neural Prophet demonstra sua capacidade de capturar padrões complexos e não lineares. No entanto, o Neural Prophet requer dados mais detalhados e ajustes finos para resultados ótimos.

Contribuição: Este estudo fornece uma análise comparativa de modelos de aprendizado profundo para previsão de séries temporais, oferecendo insights valiosos sobre seus pontos fortes e fracos. Os resultados contribuem para a compreensão de quais modelos são mais adequados para previsão de mercado de ações e como podem ser adaptados a diferentes tipos de dados e cenários.

Conclusão: A pesquisa conclui que cada modelo—SARIMA, Facebook Prophet e Neural Prophet—possui suas forças únicas na previsão de séries temporais. O SARIMA é confiável para lidar com dados sazonais, o Facebook Prophet é eficiente para conjuntos de dados menores com tendências claras e o Neural Prophet é ideal para conjuntos de dados mais complexos. O estudo destaca a importância de selecionar o modelo adequado com base nos requisitos específicos da tarefa de previsão.

Palavras-Chave: Análise de séries temporais. Previsão de séries temporais. Inteligência artificial. SARIMA. Modelo profeta neural. Modelo profeta Facebook. Parâmetros de avaliação.

1. INTRODUCTION

A time series is a type of dataset that offers valuable data about an entity based on different factors over different time intervals. Most importantly, it gives valuable insights regarding factors that impact a certain entity from time to time. It includes sequential observations that can be used to monitor processes or track performance metrics in many other fields. A time series is merely a series of points that are calculated at different times but can also be used to identify market trends and predict possible future occurrences. Time series datasets could contain hourly, daily, weekly, monthly, or yearly information but are differentiated based on the components. There are four components in a time series: trends, seasonal variations, cyclic variations, and irregular movements. Trends include the data representing movement along the term. Seasonal data provides information about seasonal trends, which represent seasonal changes. Cyclic variations mean continuous variation. Random variations showcase random sources of variation that can be seen in a dataset.

However, time series analysis is a statistical technique that uses collected data to devise solutions, make predictions, and classify things. For time series analysis, time is the most important factor. Trends, changes, and predictions are made by analyzing the data outcomes. Time series analysis not only provides surfacing results but also uses the data values to help us understand the underlying causes of the trends happening.

Time series forecasting is a subdomain of time series analysis that is used to have a complete future prediction. There are not any specific predictions it could be able to done anything. All of these predictions help us in decision-making, analyzing future requirements, making adjustments, accordingly, preparing for the unseen, and helping us in imagining what comes ahead. Time series forecasting could be done through modern models leveraging artificial intelligence. Firstly, we have to train these models, and then it will provide us with predictions on the basis of variables and the type of data that we are targeting. These models are available in different coding frameworks, like Python and R. However, Python is the most significant framework used to deploy many AI-based models. In this paper, we are using Python libraries to make the prediction model. Python has plenty of models for regression problems, but we are using the three most dominant models for building the model.

The three models used in this paper are SARIMA, the neural prophet model, and the FB prophet model. SARIMA (Seasonal autoregressive integrated moving average) is a prediction-based AI model that is capable of integrating both exogenous variables and endogenous variables. Exogenous variables are analyzed as independent time series; they include external factors that can help make our models more reliable. Although endogenous are the dependent factors, in this paper, the precision value itself is an endogenous factor dependent on many factors. On the other hand, neural prophet and Facebook prophet models are developed by the Facebook team to handle larger datasets and use them for training. These models are available to use and are present in the Pytorch library. It is a powerful tool to manage complex datasets and get insights from complex sets of information. There

is only one difference between these two models: their methods of learning through information. By prosecuting interconnected layers, the neural prophet analyzes the data, and the FB prophet utilizes a traditional statistical approach.

This methodology used for model building follows a five-step model-building methodology that includes data preprocessing, model structure defining, Model training, Model Evaluation, and Model tuning. When the model is created, we use evaluation metrics such as loss functions, accuracy, precision, recall, or F1 scoring to evaluate the functioning of the model. Subsequently, once the results are obtained, model parameters can be adjusted to optimize the functioning of the prediction model. The evaluation parameters indicated that each has its own specifications. SARIMA provides reliable results when the datasets contain various seasonality trends. In our dataset, we have obvious dataset information; that's why SARIMA works perfectly in terms of reliability. The composed model performs exceptionally well when it's trained on Prophet Models. Overall, SARIMA is reliable, given the prediction values, while pointing out inconsistent errors. Although the Prophets Model evaluation parameters showcase good results, they are not as reliable as the SARIMA models. The observation from this paper helps us identify suitable models with the help of different scenarios and utilizing time series forecasting for different domains of stock market prediction and beyond.

2. LITERATURE REVIEW

Form the past two decades, time series forecasting has experienced more significant advancements. With the advent of big data analysis, time series forecasting is the best possibility whenever there is a problem related to data (Madsen, 2007). Deep neural networks (DNNs) are versatile approaches that are helpful for deep neural networks (DNNs) to adopt versatile approaches and find applications in various fields such as prediction, detection, and creation. With the comparison of traditional machine learning models, DNNs can provide more accurate forecasts. In real-world problem-solving, shallow neural networks may require significant complexity to achieve optimal predictions, whereas DNNs have been suggested as a means of developing more effective predictive models (Samek et al., 2021).

Time series data consist of a sequence of values measured over time intervals. The method used to get results from such data is called time series analysis (Chatfield, 2000). The main objective is to extract valuable analysis from the available data and then utilize these insights to analyze an AI model. Time series forecasting is a popular technique in which certain predictions are made by a model when a trained model is used to extract information. Mostly, this popular technique is used to solve prediction-based problems (Torres et al., 2001). A deep neural network or in a machine learning model implemented by time series forecasting. However, these two methods exhibit outstanding performance and precision.

Deep learning-based forecasting models have gained popularity due to their power and ability to achieve high accuracy across a diverse range of application fields. Deep learning has been proven to be the most useful method to derive results from big data nowadays (Torres et al., 2001). There was research conducted in April 2021 to analyze the efficiency of deep learning models, in which 1, 3, and 5 layer-based neural network models were evaluated and the outcomes that were calculated based on the MSE score, with the help of the three models' MSE avg values are 0.98 for the 1 and 3-layered models and 0.96 for the 5-layered models. These values indicate the efficiency of the neural networks (Masini et al., 2021).

Deep learning has also emerged and gained popularity for raising stock price prediction models. Using the stock price for the growth of a company plays a crucial role in increasing the interest of speculators in investing their money in the company. Due to this, stock price prediction has become a targeted problem for researchers. That's why they have built several machine learning and deep learning models to predict the stock price (Khare et al., 2019). Several models were attested, and deep learning-based models were able to produce more accurate results than machine learning models (Nikou et al., 2019). The most commonly attested deep learning models are the LSTM, ARIMAS, and prophet models.

The most recommended model for time series forecasting is SARIMA. It is derived from the LSTM model, which supports univariate entries that come with seasonal components (Nontapa et al., 2021). We used seasonality information in our data, and using this model was quite a great decision. In the past, SARIMA models have been used in many time series forecasting problems and consistently provide exceptionally good results in this domain (Kritharas, 2021). It is perfectly fitting when the features of the data have seasonal behaviors. Although, by separating all of the seasonal behaviors, SARIMA has separate polynomials to get each value. With the use of external feature values, SARIMA models also come with exogenous values that help to improve forecasting (Jha and Pande, 2021).

There is an open-source library available on the internet that is the Facebook Prophet Model, devised by Facebook. This model is very popular for accuracy showcasing and the seasonality trends that are present in datasets (Sivaramakrishnan et al., 2022). However, Facebook models are used for an analyst-in-the-loop approach, which means that analysis could apply to their domain knowledge even if they could not easily understand the statistical methods. This approach is a blend of statistical forecasting and traditional human forecasting (González Mata, 2020). The Facebook Prophet model is used to forecast sales using real-world data, and this model was able to generate an accurate forecast regarding sales (Zunic et al., 2020). Although the Facebook prophet model has also been used in COVID case forecasting (Jha, B. K., & Pande, 2021; Mahmud, 2020).

The following model is utilized: the neural prophet model, which builds upon the progress of the Facebook prophet model. It is basically the same as a hybrid framework model, which depends on

Pytorch. It likewise gives full automation and permits users to change and tune the given dataset appropriately (Triebe et al., 2021). The neural prophet model is very popular for its hybrid framework, and it is also used to solve several problems. In 2023, the neural prophet was utilized (alongside different models) to assess the outbreak of the Monkeypox virus in the US. The Neural Prophet model performed well in the wide range of various models as far as prediction of the data (Long et al., 2023). The neural prophet gives exceptionally exact outcomes when it is utilized in the LSTM hybrid model. By utilizing this model, the forecasting of the electrical burden was done effectively, and it gave high accuracy values (Shohan et al., 2022). And afterward, each model works appropriately in its specific domains.

Evaluating a model is a crucial stage to estimating the future values needed to build a model, and choosing the right parameters for evaluation depends on the type of dataset used. However, general terms like MSE, F1 score, and accuracy remain the same (Cerqueira et al., 2020). There are multiple parameters that help us estimate the performance of a particular model. One of the best ways to evaluate a model is by using a holdout procedure, meaning testing the model using out-of-sample data. However, another approach that is highly recommended for testing the performance of the forecasting model is cross-validation (Bergmeir et al., 2018).

3. METHODOLOGY

In this paper, three distinct Python models are utilized to precut the stock value. These models are independently prepared on accessible datasets, yet they are utilized in the same general approach. An overall strategy for preparing a model incorporates five stages: understanding the problem, data pre-processing, exploratory analysis, training the models (three models are trained separately), and evaluation of models using evaluating parameters like MSE, RMSE, MAE, MdAPE, etc. There is Figure 1 that describe general methodology that will be used:
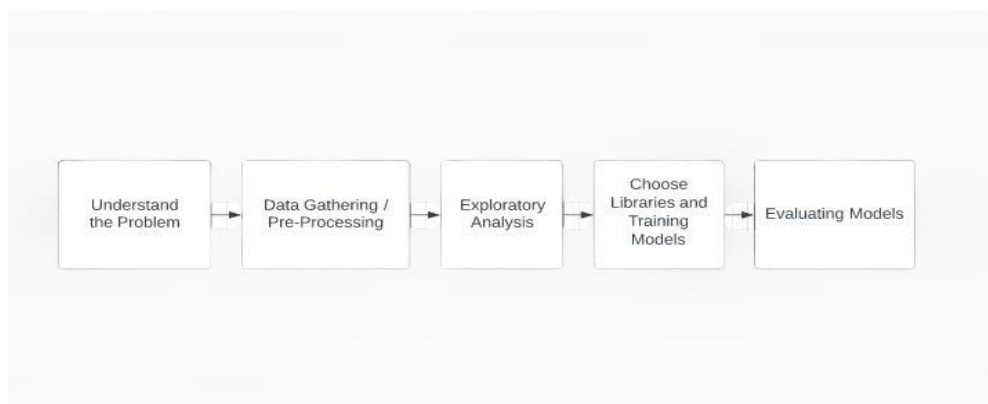

Figure 1: Flowchart for the General Methodology of Training Model

3.1. Understanding the problem statement

In the real world, there are a lot of problems in the life of every human, and they are constantly trying to resolve them and find great solutions for them. Having a solution to every problem exists, but grasping the true concept behind every problem and comprehending a solution while keeping in view the broader prospect is important. By doing this, it is necessary to understand the nature of the problem. In this paper, we address the solution to the time series forecasting problem in which proper prediction is required. A proposed solution is a deep learning model that can predict future stock market values for the Saudi Exchange.

3.2. Data gathering/ Data Pre-Processing

The data is in the form of Mulkia Gulf Real Estate, which contains information about the stock prophets. However, the data is extremely large, so there are two sheets of data, Sheet 1 and Sheet 7, which are selected to be used in building the models. This model has cross-functional data that is used before the training and is also pre-processed for splitting and encoding. There are data sheets 1 and 7 shown in

Figure 2.

	Company	ID	Date	Close Price	Open Price	Heighest Price	Lowest price	VOLUME TRADED	% CHANGE
0	Mulkia Gulf Real Estate REIT	4336	2022-05-24	9.47	9.48	9.50	9.44	73.56K	-0.0021
1	Mulkia Gulf Real Estate REIT	4336	2022-05-23	9.49	9.50	9.50	9.46	84.52K	-0.0011
2	Mulkia Gulf Real Estate REIT	4336	2022-05-22	9.50	9.52	9.52	9.47	101.12K	-0.0021
3	Mulkia Gulf Real Estate REIT	4336	2022-05-19	9.52	9.54	9.54	9.48	244.85K	-0.0021
4	Mulkia Gulf Real Estate REIT	4336	2022-05-18	9.54	9.53	9.54	9.51	115.33K	-0.0010
...	...		...	...	...	...	...	...	...
262	Mulkia Gulf Real Estate REIT	4336	2021-04-29	9.47	9.43	9.49	9.42	149.20K	0.0021
263	Mulkia Gulf Real Estate REIT	4336	2021-04-28	9.45	9.43	9.45	9.41	137.41K	-0.0032
264	Mulkia Gulf Real Estate REIT	4336	2021-04-27	9.48	9.50	9.59	9.45	88.31K	0.0074
265	Mulkia Gulf Real Estate REIT	4336	2021-04-26	9.41	9.36	9.49	9.36	180.85K	0.0053
266	Mulkia Gulf Real Estate REIT	4336	2021-04-25	9.36	9.35	9.39	9.32	117.45K	0.0043

	Company	ID	Date	Close Price	Open Price	Heighest Price	Lowest price	VOLUME TRADED	% CHANGE
0	RIYAD REIT	4330.0	2022-05-24	10.74	10.72	10.78	10.68	109.67K	-0.0019
1	RIYAD REIT	4330.0	2022-05-23	10.76	10.74	10.76	10.70	52.86K	-0.0037
2	RIYAD REIT	4330.0	2022-05-22	10.80	10.80	10.82	10.74	215.74K	0.0056
3	RIYAD REIT	4330.0	2022-05-19	10.74	10.82	10.82	10.68	210.96K	-0.0147
4	RIYAD REIT	4330.0	2022-05-18	10.90	10.86	10.90	10.70	265.40K	0.0111
...	...		...	...	...	...	...	...	...
262	RIYAD REIT	4330.0	2021-04-29	9.20	9.14	9.22	9.14	548.29K	0.0066
263	RIYAD REIT	4330.0	2021-04-28	9.14	9.35	9.35	9.10	645.97K	-0.0193
264	RIYAD REIT	4330.0	2021-04-27	9.32	9.14	9.40	9.14	1.52M	0.0298
265	RIYAD REIT	4330.0	2021-04-26	9.05	8.96	9.15	8.96	414.99K	0.0044
266	RIYAD REIT	4330.0	2021-04-25	9.01	9.00	9.15	8.97	589.64K	0.0045

Figure 2: Mulkia GULF Real Estate Dataset Sheet 1 and Sheet 7

3.3. Exploratory Analysis

While building a model, firstly, it is mandatory to explore the data, which is called a preliminary analysis of data, where you can plot the data in its original form to find certain structures. With the help of this, we can check the validity of measures, point out any outliers, examine the weightage of different variables, evaluate the effectiveness of certain manipulative variables, etc. there is Adfuller is an in-built Python library to draw the best conclusions from the data when it's in raw form. The plot indicates an obvious trend that is happening because of certain variables shown in

Figure **3** that provide data and are structured.
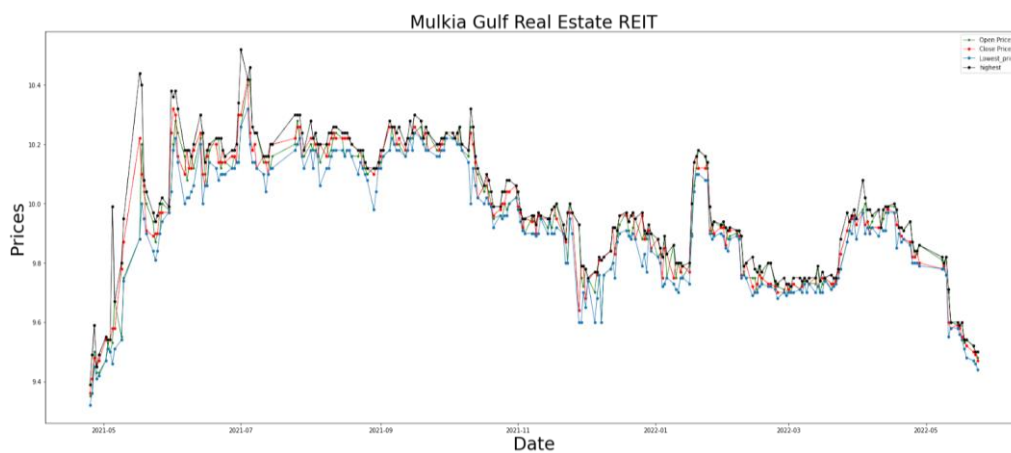
Figure 3: Exploratory Analysis of the Data

3.4. Training Models

As depicted before, Python contains many in-built training models for estimating issues. In this paper, the recommended models to be utilized are SARIMA, Brain Prophet, and FbProphet. These models are now present in Python and are applied to the two arrangements of information (Sheets 1 and 7). The training dataset is utilized to prepare these models independently on both datasets. After training, the result values given by the prepared models are assessed utilizing the score values.

3.4.1. SARIMA (Seasonal Auto-regressive Integrated Moving Average)

The primary model utilized is SARIMAX. Just like ARIMA, SARIMAX additionally estimates fleeting conditions, but it includes seasonality in the forecast. The estimation of SARIMA boundaries is equivalent to ARIMA. Just an added substance variable *m* is added to address the seasonal lags, it very well may be yearly or six months. We compose write the AR, I, and MA parameters for SARIMA as (p,d,q)*m*. The mathematics behind is especially like ARIMA which works out AR and MA values for integration I. All of these variables are determined by utilizing Eq. (1) and (2):

$$AR_t = \alpha_+ \Sigma_{i=1}^{\rho} \beta_{ixt=i} + \epsilon \qquad (1)$$

$$MA_t = \mu_+ \Sigma_{i=1}^{\rho} \Phi_i \epsilon_{t-1} \qquad (2)$$

Forecast is finished by these orders alongside m. A few standard significations are utilized to address the above three, i.e., p, d, and q; "p" is the quantity of perceptions remembered for the model, "d" is the times separating the raw observations, and "q" is the quantity of moving average sizes.

To find these parameters Connection and partial correlational graphs Figure 4 displays autocorrelation and partial correlational graphs are gotten from our dataset.
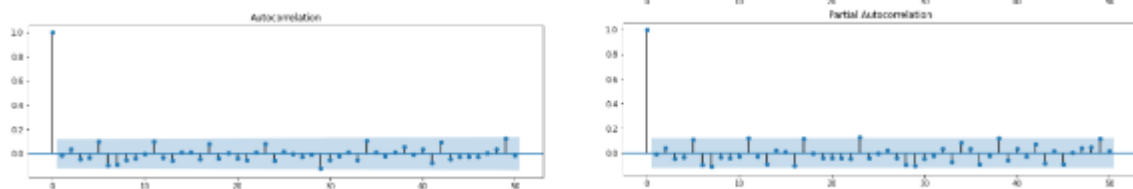

Figure 4: Auto Correlation and Partial Correlation Graphs

These graphs can be utilized to generally estimate the values of p, d, and q. After getting these underlying solvents, we continue on toward the following stage of dividing the dataset into preparing and testing data. The preparation and fitting of the models are finished utilizing the SARIMAX library. Figure 5 is utilized to show the method involved with preparing models utilizing SARIMAX. Subsequent to preparing, RMSE, MAPE, MSE, and MAE values are determined for evaluation.
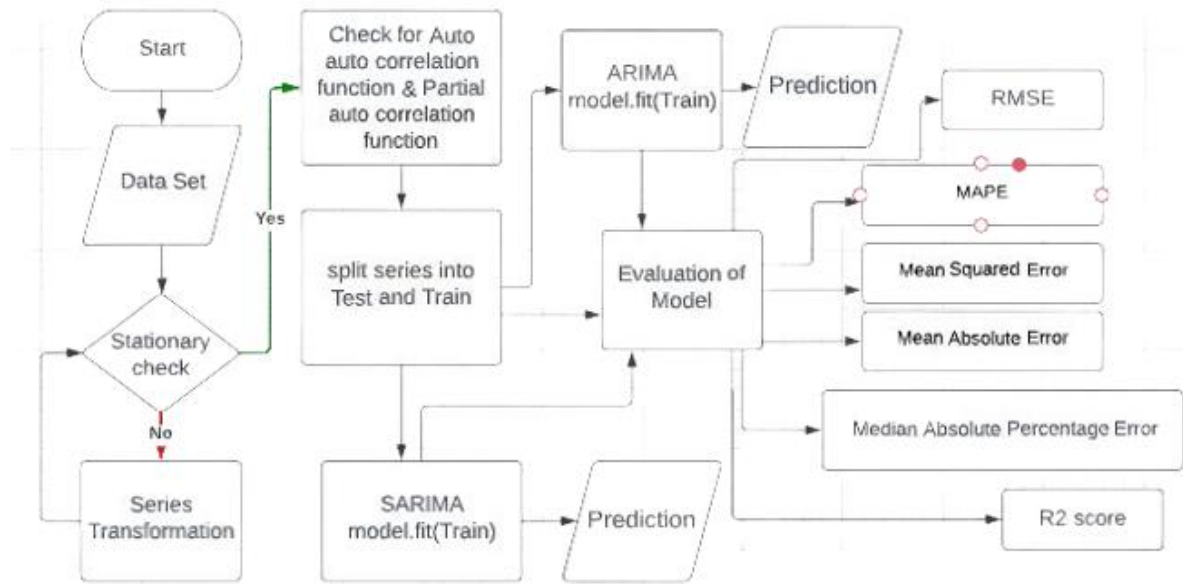
Figure 5: Flowchart for SARIMA Models

3.4.2. Prophet Models

There is an open-source Python library called Prophet () intended to be utilized for forecasting called an additive regression model, which is made to fit and understand yearly, monthly, and weekly trends that can effectively tackle non-linear regression problems.

This dataset has univariate data, in data there is one variable known as "High Pice" from the dataset of Mulkia Gulf Real that have Date column, predict future values with highest profits. This Prophet model work easily with its main features:

Importing the data: Perform preprocessing (optional step); it's encouraged to do preprocessing to clean data, or probably there is possibility of strange results.

Splitting into test and training data: Fitting the model - Getting prediction data from in- and out-sample data: we get out-sample data by producing a type of dummy data from trained datasets; this dummy data consists of predicted values of high price.

Assess by contrasting the results of the in-sample and out-sample datasets: There is only a Facebook Prophet model and the Neural Model libraries that have been utilized to assess the price prediction. There is no change in the process of using these while coding, yet the background technicalities may differ. They are utilized to recognize based on AR Net, and the autoregressive neural networks are neurons designed for forecasting. These are the main differences that give us reason to independently fit and evaluate these two models. Although the evaluation of the Facebook prophet is done with the assistance of simple error differences, as shown in Figure 6, the functioning of the Facebook prophet model.
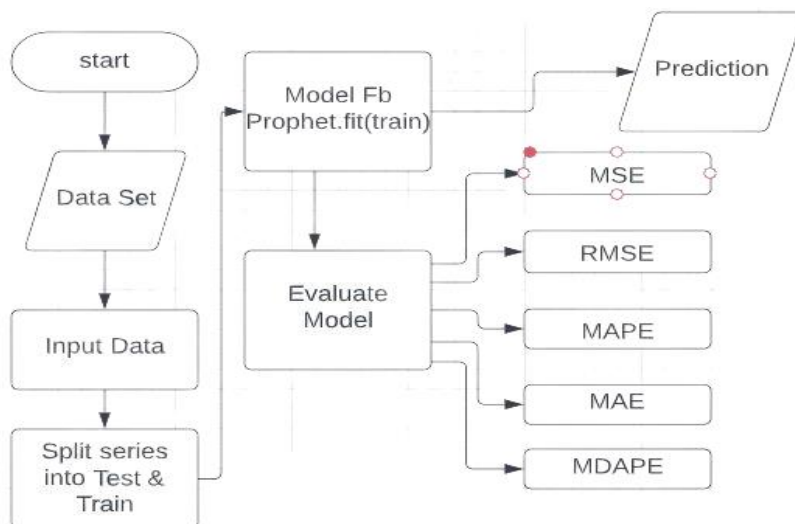
Figure 6: Flowchart for Facebook prophet Models

While a Facebook prophet is a simple prophet () model that works according to the given general Eq. (3):

$$y_t = g(t)+s(t)+h(t)+\varepsilon_t \qquad (3)$$

In Eq. (3), g (t) represents the prediction trend pricewise, s(t) represents the seasonality pattern, h(t) represents the holiday's effects on prices, and εt represents the white noise data.

In a Neural Prophet model that inclusion a simple prophet model is included for a neural network. It has pre-built functionality that doesn't require any additional coding. User preferences need to have a configuration of seasonality modes; aside from data processing, we will just zero in on preparing the data and evaluating its performance based on the in- and out-sample models. There is a workflow of neural prophet models shown in Figure 7.
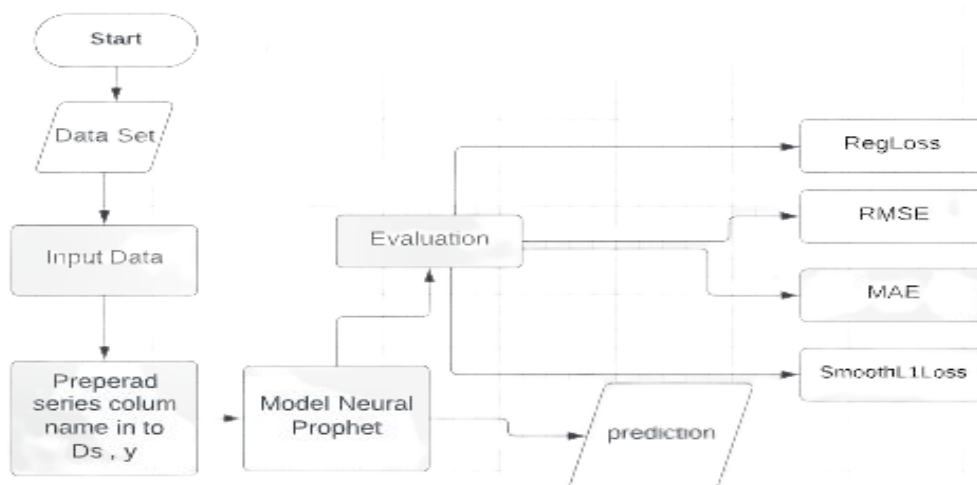

Figure 7: Flowchart for Neural Prophet Models

The assessment of neural prophet is finished by calculating RegLoss, SmoothL1Loss, and other error evaluation parameters.

4. RESULTS AND DISCUSSION

The fundamental reason for doing this research is to analyze the proficiency and use of three models, and the information utilized for this research is Mulkia GULF real state data to predict the future values of the Saudi stock market. The first model that is used is SARIMA model, and it was prepared using SARIMA and SARIMAX, then imported from statsmodels. TSA in seasonal orders. Figure 8 is used to show a graphical representation of SARIMA models, and these graphs provide an estimate to predict the performance of SARIMAX models.
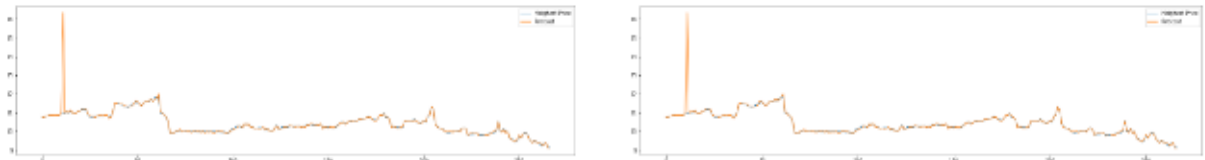

Figure 8: SARIMA model results from Sheet 1 and Sheet 7

The evaluation of the SARIMAX model is assessed by calculating MAPE, MdAPE, and MSE. After the complete evaluation, the R2 score for Sheet 1 is 0.6330271 and -0.7617077 for Sheet 7. Then the MSE value for sheet 1 is 0.1098026 and 0.0851622 for sheet 7, the MAPE value for sheet 1 is 0.0042032 and 0.0035787 for sheet 7, and the values for MdAPE for sheet 1 are 0.1880252 and 0.1204309 for sheet 7. The values that indicate the presence of non-consistent errors have a larger value than SSMSE, i.e., 0.085 and 0.109.

Now there is a next model that is trained with the help of the Facebook prophet model from prohet() from fbprophet. Here are Figure 9 and Figure 10 that show the differences between actual and predicted values.
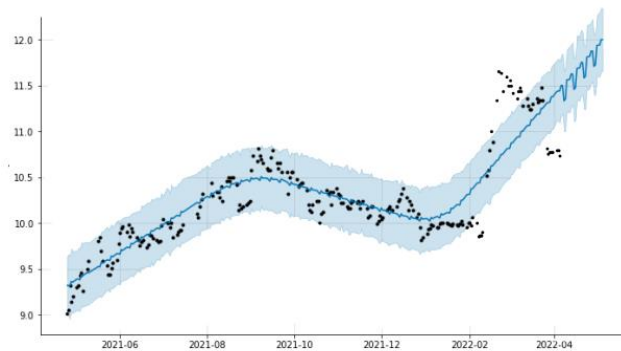

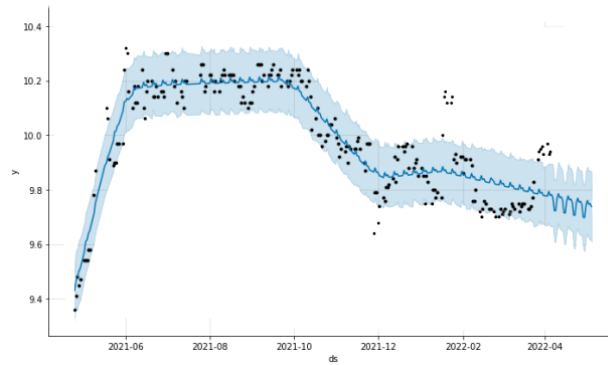**Figure 9:** Fb Prophet Model results for Sheet 1

Figure 10: Fb Prophet Model results for Sheet 7

The most important thing to note is that the graphs and dates are scaled along the x-axis labeled as 'ds', while the highest price values are scaled with the y-axis labeled as 'y'.

The next evaluation uses the Fb prophet model that is used to evaluate MSE, MAE, MAPE, and MdAPE. The value of R2 for sheet 1 is 0.050958 and for sheet 7 is 050958, then the value for MSE for sheet 1 is 0.735285 and for sheet 7 is 0.028332, then the value for MAE for sheet 1 is 0.823271 and for sheet 7 is 0.154208, now the MAPE value for sheet 1 is 0.075926 and for sheet 7 is 0.015819, and the values for MdAPE in sheet 1 are 7.253010 and 1.656689 for sheet 7. All of these results indicate satisfactory results for the FB Prophet model. Model that trained previously is also a prophet model that is also known as Neural Prophet model.

In this prophet model, the presented model in Python is used as Neural Prophet. Everyone can easily install it and then use it to train their datasets. Here is a graph representation in Figure 11 and Figure 12 that is used to display the efficiency of the predicted values without adding seasonality trends. It can be used to compare it with the actual values that are given below:
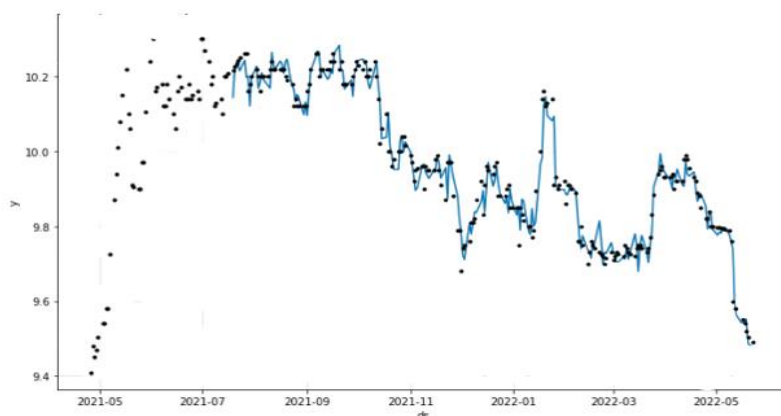

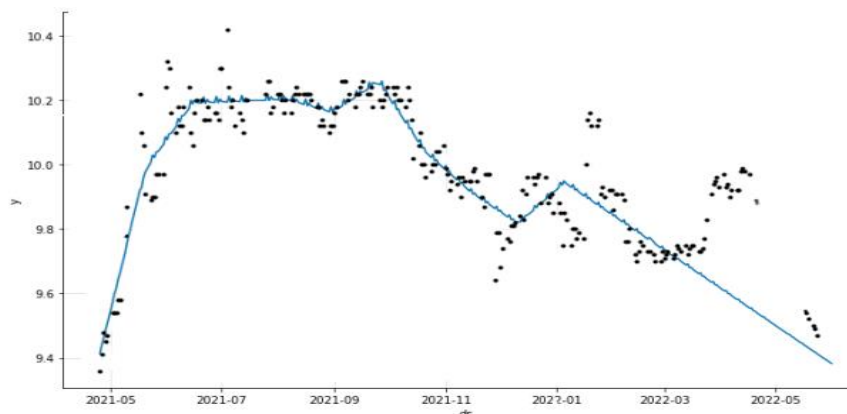**Figure 11:** Neural Prophet model results for Sheet 1

Figure 12: Neural Prophet model results for Sheet 7.

The wide range of various graphs that are plotted while including the weekly and monthly seasonality modes show similar outcomes. In the previous graphs above, regression demonstrates the dummy values that are utilized to predict the values, and the dark dots that are seen there are actual prices. with the evaluation of the neural prophet model using MAE and RMSE, with the assistance of two additional parameters, Smooth1 loss and RegLoss, the RMSE an incentive for sheet 1 is 0.0954/0.2555 and 0.0945/0.269 for sheet 7, and the values for MAE in sheet 1 are 0.077 and 0.0766 for sheet 7. However, the Koss values that are utilized to give positive outcomes to the proficiency of the model that was utilized, as well as the neural model, seems to work well with smaller datasets, however for this situation, the larger values found in MAE and RMSE.

CONCLUSION

In this paper, we have seen that time series forecasting can easily be done for predicting future market values for the Saudi Exchange, and the data that is used to train this model take from Mulkia GULF real estate. There are three unique Python models that are utilized to train the dataset: SARIMAX, neural Prophet, and Facebook Prophet Model. The evaluation for each model is done by utilizing evaluation parameters like MSE, RMSE, F1 score, precision, and so on. These evaluation parameters are utilized to give the conclusion for each model. The SARIMA model is an evaluation parameter that was utilized here to give the best outcomes (avg. MSE score is – 0.04, avg. MAPE esteem is 0.0038), however it gives best outcomes with context of the values for MSE, i.e., 0.085 and 0.109, which show the presence of non-consistent errors. Although the Facebook prophet model furnishes a satisfactory prediction with an R2 score of 0.50958, an average MSE value of 0.3818, and an average MAPE value of 0.0458725, it likewise offers a typical benefit for MdAPE with a result of 4.454. While there is a neural prophet model that is very well known for giving great accuracy, in our dataset, it doesn't give the best accuracy. Results demonstrate that the neural prophet recognizes complex patterns and irregularities, while the Facebook prophet model can get a handle on occasional impacts and long-term trends. However, these three models are great in their own domain and work completely great. The SARIMA and Facebook Prophet models are utilized to show seasonality trends,

and neural Prophet is more qualified for smaller datasets that have the accuracy to capture the complex nonlinear patterns that are present in time series data. To conclude all of this, the literature suggests that the performance of SARIMAX, Facebook Prophet, and neural Prophet Models to predict stock prices relies upon many factors, including the quality and amount of data that is accessible, the complexity of the relationships between the variables, and the capacity of the model to capture its patterns and trends. While SARIMAX is utilized to perform improved results to capture complex patterns and seasonal examples, the Facebook prophet model may have more efficiency and be easy to utilize, and neural prophets might be more effective in catching complex relationships in the data, however it requires more ability and implementation to carry out the data. It is essential to take note of that the effectiveness of each model may vary depending on the specific problem at hand, and it is recommended to attempt to try different experiments with various models to figure out which one performs best.

REFERENCES

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120, 70-83.

Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. Machine Learning, 109, 1997-2028.

Chatfield, C (2000). Time-series forecasting. CRC press.

González Mata, A. (2020). A comparison between LSTM and Facebook Prophet models: a financial forecasting case study.

Jha, B. K., & Pande, S. (2021). Time series forecasting model for supermarket sales using FB-prophet. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 547-554). IEEE.

Khare, K., Darekar, O., Gupta, P., & Attar, V. Z. (2017, May). Short term stock price prediction using deep learning. In *2017 2nd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)* (pp. 482-486). IEEE.

Kritharas, P. (2014). Developing a SARIMAX model for monthly wind speed forecasting in the uk (Doctoral dissertation, Loughborough University).

Long, B., Tan, F., & Newman, M. (2023). Forecasting the Monkeypox Outbreak Using ARIMA, Prophet, NeuralProphet, and LSTM Models in the United States. Forecasting, 5(1), 127-137.

Madsen, H. (2007). *Time series analysis*. CRC Press.

Mahmud, S. (2020). Bangladesh COVID-19 daily cases time series analysis using Facebook prophet model. *Available at SSRN 3660368*.

Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2023). Machine learning advances for time series forecasting. Journal of economic surveys, 37(1), 76-111.Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. Intelligent Systems in Accounting, Finance and Management, 26(4), 164-174.

Nontapa, C., Kesamoon, C., Kaewhawong, N., & Intrapaiboon, P. (2021). A New Hybrid Forecasting Using Decomposition Method with SARIMAX Model and Artificial Neural Network. Computer Science, 16(4), 1341-1354.

Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K. R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, *109*(3), 247-278.

Shohan, M. J. A., Faruque, M. O., & Foo, S. Y. (2022). Forecasting of electric load using a hybrid LSTM-neural prophet model. Energies, 15(6), 2158.

Sivaramakrishnan, S., Fernandez, T. F., Babukarthik, R. G., & Premalatha, S. (2022). Forecasting time series data using arima and facebook prophet models. In *Big data management in Sensing* (pp. 47-59). River Publishers.

Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., & Troncoso, A. (2021). Deep learning for time series forecasting: a survey. *Big Data*, *9*(1), 3-21.

Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., & Rajagopal, R. (2021). Neuralprophet: Explainable forecasting at scale. *arXiv preprint arXiv:2111.15397*.

Zunic, E., Korjenic, K., Hodzic, K., & Donko, D. (2020). Application of facebook's prophet algorithm for successful sales forecasting based on real-world data. *arXiv preprint arXiv:2005.07575*.