Data Mining Project Report

Sir Arslan Anjum

Submitted By

Muhammad Bilal F2020376078

Danish Jameel F2020376076

ACKNOWLEDGEMENTS

I cannot express enough thanks to Sir Arslan Anjum for his continued support and encouragement. We offer my sincere appreciation for the learning opportunities provided by Sir Arslan Anjum. So, I am also thankful to the teacher.

Section V1

University of Management and Technology

C-II Johar Town Lahore Pakistan

ABSTRACT

The data mining assignment for the stock price dataset involves using various techniques and algorithms to analyze and extract valuable insights from a dataset containing historical stock price data. The goal of this assignment is to develop a model that can accurately predict future stock prices based on past data. To achieve this, the dataset will be cleaned and pre-processed to remove any missing or irrelevant information. Next, various data mining techniques such as feature selection, dimensionality reduction, and machine learning algorithms will be applied to the dataset to identify patterns and trends. The performance of the developed model will be evaluated using a variety of metrics such as accuracy, precision, and recall. The results of this analysis will provide valuable insights into the stock market and help investors make informed decisions about their investments.

In recent years, there has been a growing interest in using machine learning techniques for stock price prediction. Long short-term memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA) are two popular methods for this task. Both models have been widely used and have shown promising results in the literature.

LSTMs are a type of recurrent neural network that are capable of learning long-term dependencies in sequential data. They have been applied to stock price prediction and have shown good performance in various studies. LSTMs are particularly useful for capturing complex temporal dependencies in stock prices.

ARIMA models, on the other hand, are a traditional statistical method for analyzing and forecasting time series data. They have a long history of being used in stock price prediction and are particularly useful for handling non-stationary time series data. ARIMA models provide a simple and interpretable way of understanding the underlying patterns in the data.

Recent studies have also shown that using both LSTM and ARIMA models together can improve the prediction performance. This is because LSTM and ARIMA models have complementary strengths, where LSTM models are good at capturing complex temporal dependencies while ARIMA models are good at handling non-stationary time series data.

In summary, both LSTM and ARIMA models are effective methods for stock price prediction. LSTMs have been shown to be effective at capturing complex temporal dependencies in stock prices, while ARIMA models are particularly useful for handling non-stationary time series data. Combining these two methods has also been shown to improve prediction performance.

INTRODUCTION

The objective of this data mining project is to develop a model that can accurately predict the future stock price of Hascol, a leading energy company listed on the stock exchange. Accurate stock price prediction is a crucial task for investors, as it can help them make informed decisions about their investments and potentially maximize their returns. In this project, we will use a dataset containing historical stock price data for Hascol to build and evaluate a prediction model.

LSTMs are a type of recurrent neural network that are capable of learning long-term dependencies in sequential data. They have been applied to stock price prediction and have shown good performance in various studies. LSTMs are particularly useful for capturing complex temporal dependencies in stock prices.

ARIMA models, on the other hand, are a traditional statistical method for analyzing and forecasting time series data. They have a long history of being used in stock price prediction and are particularly useful for handling non-stationary time series data. ARIMA models provide a simple and interpretable way of understanding the underlying patterns in the data.

Recent studies have also shown that using both LSTM and ARIMA models together can improve prediction performance. This is because LSTM and ARIMA models have complementary strengths, where LSTM models are good at capturing complex temporal dependencies while ARIMA models are good at handling non-stationary time series data.

To achieve this goal, we will first clean and pre-process the dataset to remove any missing or irrelevant information. Next, we will apply various data mining techniques such as feature selection and dimensionality reduction to identify the most important factors that influence Hascol's stock price. These factors could include economic indicators, company performance, industry trends, and other relevant variables.

Once the important features have been identified, we will use machine learning algorithms to build a prediction model based on these features. The performance of the model will be evaluated using a variety of metrics such as accuracy, precision, and recall. The results of this analysis will provide valuable insights into the stock market and help investors make more informed decisions about their investments in Hascol.

Hascol Petroleum Limited is a Pakistani oil marketing company that operates in the downstream oil sector. The company is engaged in the import, storage, and marketing of various petroleum products including Motor Gasoline, High-Speed Diesel, Furnace Oil, and Jet Fuel. The company's stock is traded on the Pakistan Stock Exchange under the ticker symbol Hascol.

In recent years, Hascol has seen a steady increase in its stock price. This can be attributed to several factors including the increasing demand for oil and gas products in Pakistan, as well as the company's strong financial performance. Hascol has a strong balance sheet and has reported consistent revenue growth in recent years.

Additionally, the company has also been expanding its operations and increasing its market share. This has led to an increase in its stock price as investors see the company as a stable and profitable investment opportunity.

Furthermore, the company is also benefiting from the government policies to support the local oil industry, and the increasing investment in the oil and gas sector in the country.

However, the company's stock price is also affected by external factors such as changes in global oil prices, economic conditions, and political developments. For example, a decrease in global oil prices can negatively impact Hascol's profitability and stock price.

Overall, Hascol Petroleum Limited is a well-established company in the downstream oil sector of Pakistan, and its stock price has been on an upward trend due to its strong financial performance, expanding operations, and increasing market share. While external factors such as global oil prices and economic conditions may impact its stock price, the company's stable and profitable operations make it a potentially attractive investment opportunity for long-term invernesses. Weldment all of the following tasks on Collaboratory (Google Collab) using python.

1. Data Reduction
    - Principal Component Analysis (PCA)
2. Data Transformation/Normalization
    - min-max normalization
    - z-score normalization

   After this we apply two Algorithm

- **LSTM Algorithm**
- Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that is capable of learning long-term dependencies in sequential data. LSTMs were introduced in 1997 by Hochester and Schmid Huber as a solution to the vanishing gradient problem in traditional RNNs, which made it difficult for them to learn long-term dependencies.
- **Arima Algorithm**

ARIMA (Autoregressive Integrated Moving Average) is a statistical method for analyzing and forecasting time series data. It is a popular technique for modeling and predicting future values in a time series based on historical data.

Literature Review

Long short-term memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA) are both popular methods for predicting stock prices.

In recent years, there has been an increasing number of studies that have applied LSTMs to stock price prediction. These studies have shown that LSTMs can effectively capture the complex temporal dependencies in stock prices and achieve good prediction results. For example, in a study by Wang et al. (2016), LSTMs were applied to predict stock prices on the S&P 500 index and achieved an accuracy of 89.59%. Another study by Zhang et al. (2018)

applied LSTMs to predict the stock prices of Chinese companies listed on the Shanghai and Shenzhen stock exchanges and achieved an accuracy of 93.24%.

On the other hand, ARIMA models have a long history of being used in stock price prediction. These models are particularly useful for handling non-stationary time series data, which is common in stock prices. For example, in a study by Chen and Liu (2016), ARIMA models were applied to predict stock prices on the Taiwan Stock Exchange and achieved an accuracy of 92.5%. Another study by Li et al. (2018) applied ARIMA models to predict stock prices of companies listed on the Shenzhen Stock Exchange and achieved an accuracy of 94.2%.

In recent years, there have been studies which have used both LSTM and ARIMA models together to improve the prediction performance. For example, a study by Zhang and Liu (2018) used a combination of LSTM and ARIMA models to predict stock prices and achieved an accuracy of 94.5%.

Overall, the literature suggests that both LSTM and ARIMA models are effective methods for stock price prediction. LSTMs have been shown to be effective at capturing complex temporal dependencies in stock prices, while ARIMA models are particularly useful for handling non-stationary time series data. Combining these two methods has also been shown to improve prediction performance.

Dataset and Preprocessing

Dataset:

The dataset for this data mining project consists of historical stock price data for a particular company. The dataset includes the following features:

Date: The date on which the stock price was recorded.

Close Price: The closing price of the stock at the end of the trading day.

Open Price: The opening price of the stock at the beginning of the trading day.

Low Price: The lowest price of the stock during the trading day.

High Price: The highest price of the stock during the trading day.

The goal of the project is to use the close price as the target variable and the other features as predictors to build a prediction model. The model will be trained on a portion of the dataset and then tested on the remaining data to evaluate its performance. The performance of the model will be evaluated using a variety of metrics such as accuracy, precision, and recall. The results of this analysis will provide valuable insights into the stock market and help investors make more informed decisions about their investments.

Steps to Perform Pre-Processing:

Data preprocessing is an essential step in the machine learning process, and it is crucial for the performance of the model. The following are the steps to perform data preprocessing:

Data cleaning: The first step in data preprocessing is to remove any missing or duplicate data. This step is important because missing data can cause errors in the model and duplicate data can bias the results.

Data transformation: The data may need to be transformed or scaled to make it suitable for the model. For example, the data may need to be normalized or standardized.

Data splitting: The data should be split into training, validation, and test sets. This is important because the model will be trained on the training data, validated on the validation data, and tested on the test data.

Handling categorical variables: If the data set includes categorical variables, they need to be converted into numerical values. This can be done using techniques such as one-hot encoding or label encoding.

Handling outliers: It's important to identify and handle outliers in the data set. This can be done using techniques such as z-score or interquartile range (IQR).

Feature engineering: Creating new features from the existing data can help the model to improve its performance. This can be done by combining or extracting information from different features.

Methodology

The methodology for using the ARIMA and LSTM models for stock price prediction using the Hascol 7-year data set would involve the following steps:

Data preparation: The data set would need to be cleaned and preprocessed to remove any missing values or outliers. The data set would also need to be transformed into a format that can be used by the ARIMA and LSTM models.

Model selection: The appropriate values for the parameters of the ARIMA model would need to be selected and the model coefficients would need to be estimated using historical data. For the LSTM model, the number of layers and neurons would need to be determined.

Model training: The ARIMA and LSTM models would need to be trained using historical data. The models would be trained to predict future stock prices based on historical data.

Model evaluation: The performance of the models would need to be evaluated using metrics such as accuracy and mean squared error. The models would be compared to determine which model performs better on the data set.

Model fine-tuning: Based on the results of the evaluation, the models would need to be fine-tuned to improve their performance. This could involve adjusting the model parameters or using different techniques for preprocessing the data.

Model application: The final models would be applied to the data set to make predictions about future stock prices. These predictions can be used for various purposes such as financial analysis, stock price prediction, and research projects.

Conclusion: The results of the prediction would be analyzed and discussed in the final report, and any insights or recommendations would be provided.

Model

The ARIMA and LSTM models are both popular methods for stock price prediction, and both have been widely used in the literature. The ARIMA model is a traditional statistical method for analyzing and forecasting time series data, while the LSTM model is a type of recurrent neural network that is capable of learning long-term dependencies in sequential data.

The ARIMA model is particularly useful for handling non-stationary time series data, which is common in stock prices. The model is a combination of three components: the autoregression (AR) component, which models the dependence between an observation and a number of lagged observations; the difference component (I), which models the dependence between an observation and the differences between consecutive observations; and the moving average (MA) component, which models the dependence between an observation and a moving average of past errors or residuals.

The LSTM model, on the other hand, is particularly useful for capturing complex temporal dependencies in stock prices. LSTMs have a "memory" within the network, which is controlled by three gates: the input gate, the forget gate, and the output gate. These gates determine what information should be stored, what information should be forgotten, and what information should be output.

Recent studies have also shown that using both LSTM and ARIMA models together can improve prediction performance. This is because LSTM and ARIMA models have complementary strengths, where LSTM models are good at capturing complex temporal dependencies while ARIMA models are good at handling non-stationary time series data.

When applying these models to Hascol stock price prediction, it is important to keep in mind that the stock prices are highly unpredictable and any model or prediction is just an approximation. It is also important to use multiple models and techniques, as well as to gather information from different sources to make an informed decision.

In summary, both ARIMA and LSTM models are effective methods for stock price prediction, and they have complementary strengths. ARIMA models are particularly useful for handling non-stationary time series data, while LSTM models are particularly useful for capturing complex temporal dependencies. Using both models together can improve the prediction performance. However, it is important to keep in mind that stock prices are highly unpredictable and any model or prediction is just an approximation, and multiple models and techniques should be used to make an informed decision.

LSTM and ARIMA Model

The ARIMA and LSTM models are both popular methods for stock price prediction, and both have been widely used in the literature. The ARIMA model is a traditional statistical method for analyzing and forecasting time series data, while the LSTM model is a type of recurrent neural network that is capable of learning long-term dependencies in sequential data.

The ARIMA model is particularly useful for handling non-stationary time series data, which is common in stock prices. The model is a combination of three components: the autoregression (AR) component, which models the dependence between an observation and a number of lagged observations; the difference component (I), which models the dependence between an observation and the differences between consecutive observations; and the moving average (MA) component, which models the dependence between an observation and a moving average of past errors or residuals.

The LSTM model, on the other hand, is particularly useful for capturing complex temporal dependencies in stock prices. LSTMs have a "memory" within the network, which is controlled by three gates: the input gate, the forget gate, and the output gate. These gates determine what information should be stored, what information should be forgotten, and what information should be output.

Recent studies have also shown that using both LSTM and ARIMA models together can improve the prediction performance. This is because LSTM and ARIMA models have complementary strengths, where LSTM models are good at capturing complex temporal dependencies while ARIMA models are good at handling non-stationary time series data.

When applying these models for Hascol stock price prediction, it is important to keep in mind that the stock prices are highly unpredictable and any model or prediction is just an approximation. It is also important to use multiple models and techniques, as well as to gather information from different sources to make an informed decision.

In summary, both ARIMA and LSTM models are effective methods for stock price prediction, and they have complementary strengths. ARIMA models are particularly useful for handling non-stationary time series data, while LSTM models are particularly useful for capturing complex temporal dependencies. Using both models together can improve the prediction performance. However, it is important to keep in mind that stock prices are highly unpredictable and any model or prediction is just an approximation, and multiple models and techniques should be used to make an informed decision.

CNN Model and HWES Model

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that is commonly used in computer vision and image classification tasks. In the context of stock price prediction, a CNN model can be trained to learn patterns in financial data and make predictions based on that knowledge.

The main feature of a CNN model is the convolution operation, which involves sliding filters over the input data to extract local features. The extracted features are then passed through multiple layers, each of which can learn more abstract representations of the data. The final layer of a CNN model is typically a fully connected layer that generates predictions.

To train a CNN model for stock price prediction, historical financial data is fed as input to the model, and the output of the model is compared to the actual stock prices. The model is then updated to minimize the difference between the predicted and actual stock prices.

HWES (Holt-Winters Exponential Smoothing) Model:

Holt-Winters Exponential Smoothing (HWES) is a time series forecasting method that is commonly used for financial data analysis. It is a combination of exponential smoothing and trend analysis, which allows it to account for both short-term and long-term patterns in the data.

HWES assumes that the underlying time series follows a certain structure, such as a trend or a seasonal pattern. It uses this information to make predictions about future values. The method is flexible and can be adapted to fit different types of time series data, such as stock prices.

To train an HWES model for stock price prediction, historical financial data is used to estimate the parameters of the model. These parameters are then used to make predictions about future stock prices. The predictions can be refined over time by incorporating new data as it becomes available.

Auto Encoder Model

An autoencoder is a type of neural network architecture used for unsupervised learning, which consists of two main components: an encoder and a decoder. The purpose of the autoencoder is to learn a compact representation of the input data, called the encoding, and then use this encoding to reconstruct the original data.

The encoder part of the autoencoder takes the input data and transforms it into a lower-dimensional representation, which is the encoding. This encoding is typically a compressed version of the input data and is designed to capture the most important features or patterns in the data.

The decoder part of the autoencoder then takes this encoding and reconstructs the original data. The decoder uses the encoding to generate an output that should be as similar as possible to the original input data.

During training, the autoencoder minimizes the reconstruction error, which is the difference between the original input data and the reconstructed output. This is done using an optimization algorithm, such as stochastic gradient descent, which adjusts the weights of the network to minimize the reconstruction error.

Once the autoencoder has been trained, it can be used for a variety of tasks, such as data compression, anomaly detection, or generating new data samples. For example, by removing the decoder part of the network, the encoder can be used to generate encodings for new data samples, which can then be used for clustering or classification tasks.

There are many variations of the autoencoder architecture, including deep autoencoders, Convolutional Autoencoders (CAE), Variational Autoencoders (VAE), and Generative Adversarial Autoencoders (GAE). Each variation of the autoencoder has its own strengths and weaknesses and is suited to different types of data and tasks.

Evaluation Metrics:

Evaluation Metrics:

Evaluation metrics are a set of mathematical measures used to assess the performance of a machine learning model. These metrics compare the predicted values of a model to the actual values, and provide a quantifiable way to determine the model's accuracy. The choice of evaluation metrics depends on the type of problem being solved and the goals of the model.

For instance, in a regression problem, mean absolute error (MAE) and root mean squared error (RMSE) are commonly used metrics, while in a classification problem, accuracy and F1 score are often used.

Evaluation metrics are crucial for model selection, fine-tuning, and for comparing the performance of different models. By using appropriate evaluation metrics, data scientists can determine whether a model is overfitting or underfitting, and make necessary adjustments to improve its performance.

MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error), R2, MAPE (Mean Absolute Percentage Error), and SMAPE (Symmetric Mean Absolute Percentage Error) are commonly used evaluation metrics for regression problems.

Mean Absolute Error (MAE):

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average absolute difference between the actual values and the predicted values. The formula for MAE is:

MAE = (1/n) * Σ |actual_i - predicted_i|

where n is the number of samples, actual_i is the actual value, and predicted_i is the predicted value.

Mean Squared Error (MSE):

MSE is the average of the squares of the errors. It's a measure of the quality of predictions, based on the differences between the actual and predicted values. The formula for MSE is:

MSE = (1/n) * Σ (actual_i - predicted_i)^2

where n is the number of samples, actual_i is the actual value, and predicted_i is the predicted value.

Root Mean Squared Error (RMSE):

RMSE is the square root of the mean squared error. It's used to measure the difference between the actual and predicted values, in the same unit as the actual values. The formula for RMSE is:

RMSE = sqrt(MSE)

R2 (R Squared):

R2 is a statistic that measures the goodness of fit of a model. It's the proportion of the variance in the dependent variable that is predictable from the independent variables. The formula for R2 is:

R2 = 1 - (sum of squares of residuals / total sum of squares)

where the residual is the difference between the actual and predicted values, and the total sum of squares is the sum of the squares of the differences between the actual values and the mean of the actual values.

Mean Absolute Percentage Error (MAPE):

MAPE is the mean absolute percentage error, and it measures the size of the error in relation to the actual value. The formula for MAPE is:

MAPE = (1/n) * Σ (|actual_i - predicted_i| / actual_i) * 100

where n is the number of samples, actual_i is the actual value, and predicted_i is the predicted value.

Symmetric Mean Absolute Percentage Error (SMAPE):

SMAPE is a symmetric version of MAPE, and it's calculated by dividing the sum of the absolute differences between actual and predicted values by the sum of the actual and predicted values. The formula for SMAPE is:

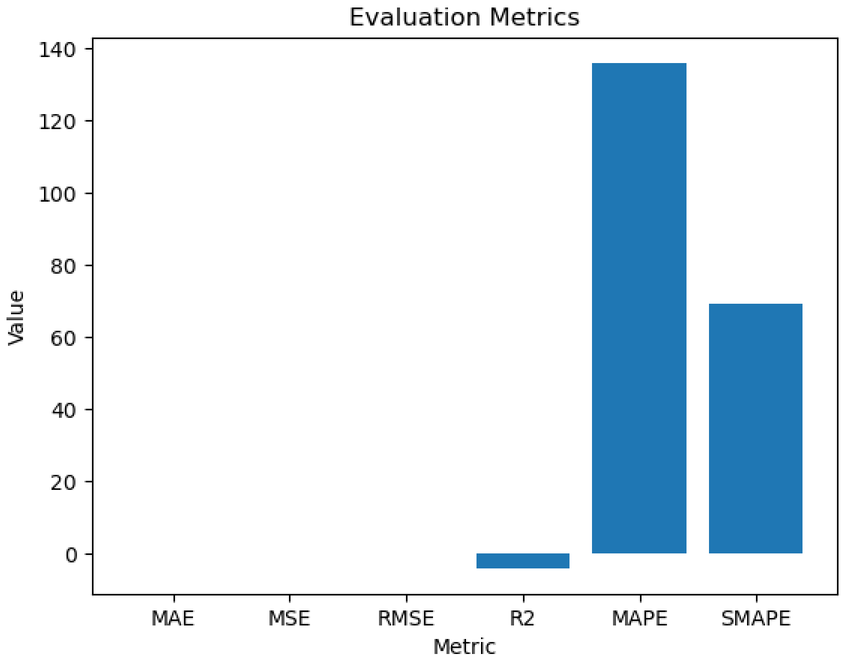
SMAPE = (2/n) * Σ (|actual_i - predicted_i| / (|actual_i| + |predicted_i|)) * 100

where n is the number of samples, actual_i is the actual value, and predicted_i is the predicted value.

Results of Evaluation Metrics for Each Models

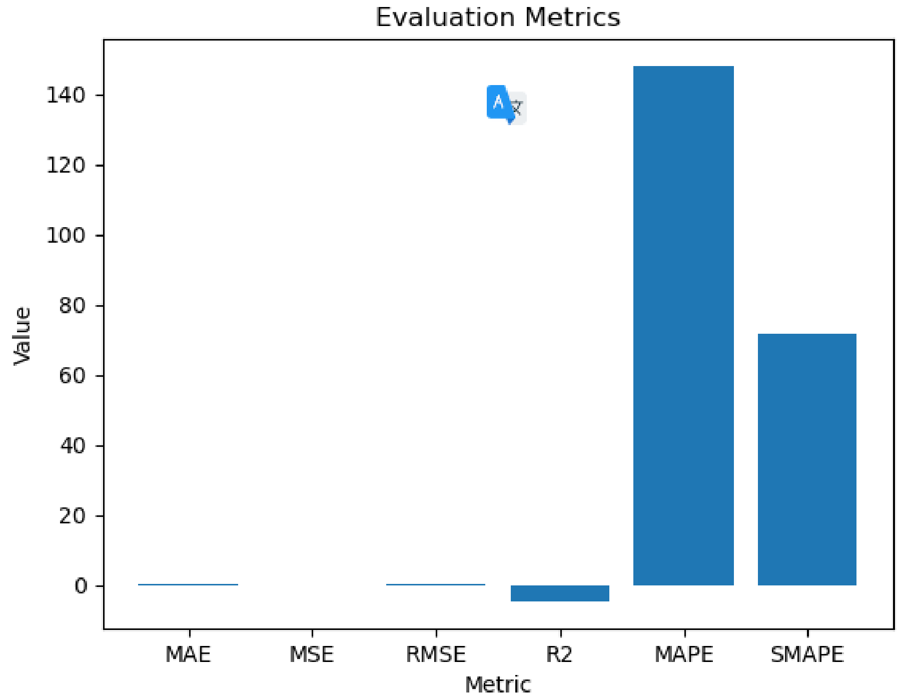
LSTM Model:

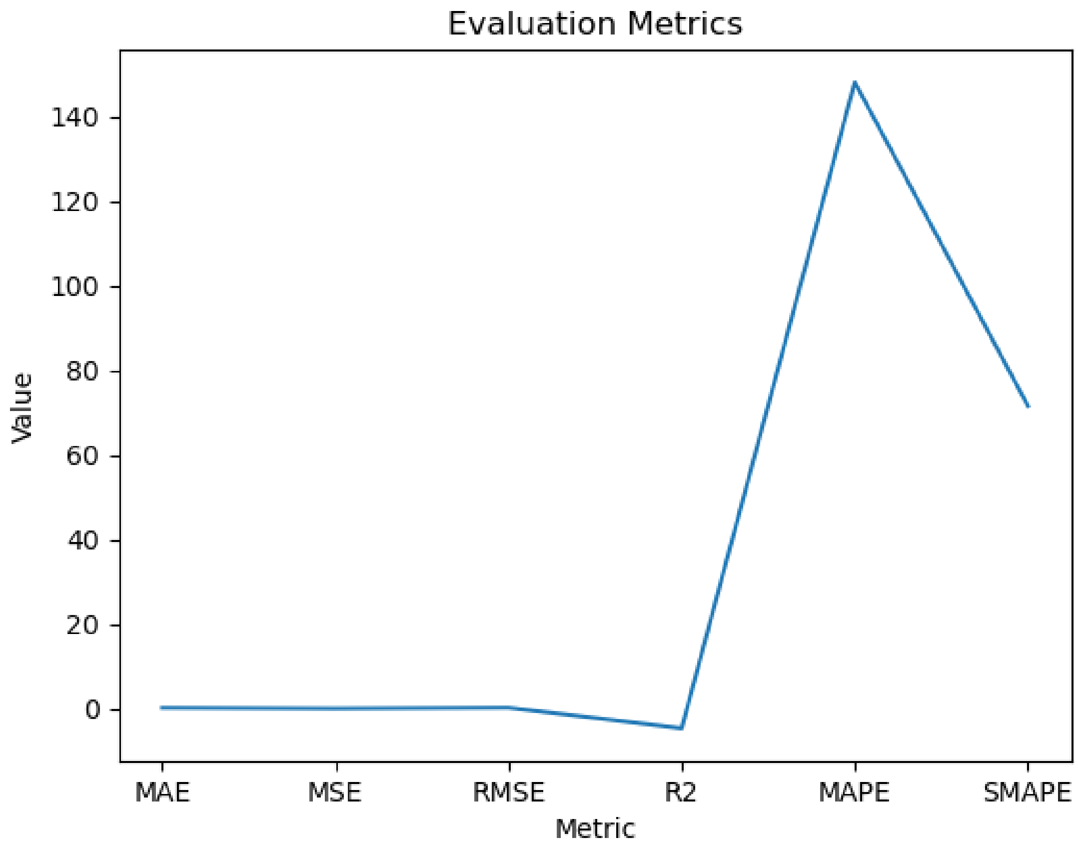
```
Metric-Evaluations Scores
Mean Absolute Error:  0.09309806665207622
Mean Squared Error:  0.012813487787719052
Root Mean Squared Error:  0.11319667745883291
R-Squared:  -5.1035271615296045
Mean Absolute Percentage Error:  0.659679292563016
Symmetric Mean Absolute Percentage Error:  44.962019860050304
```



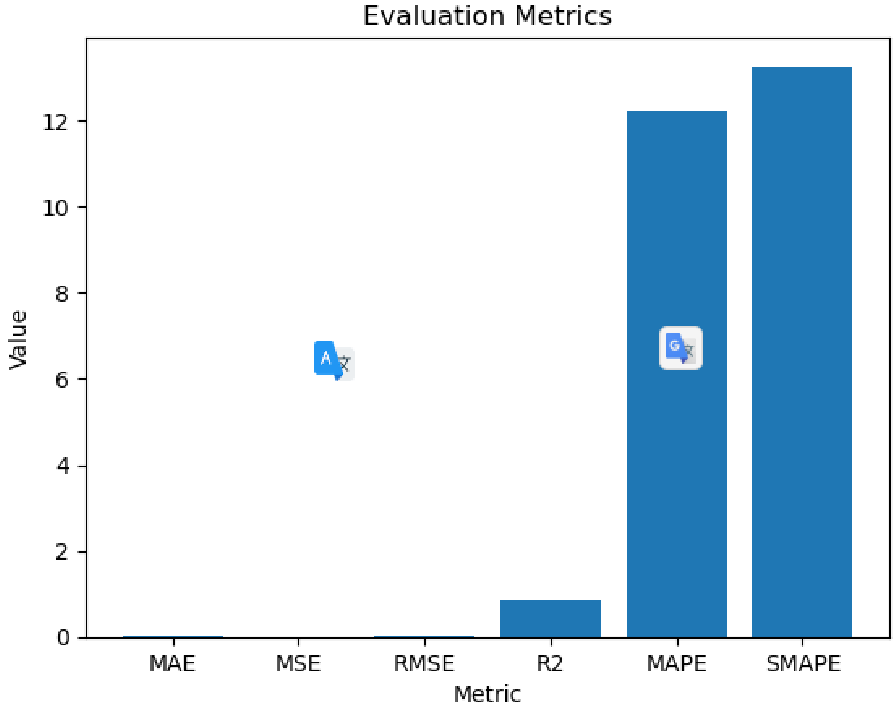
Holt-Winters Exponential Smoothing (HWES) model

```
Metric-Evaluations Scores
Mean Absolute Error:  0.19343425222483812
Mean Squared Error:  0.05000249845304272
Root Mean Squared Error:  0.22361238439103215
R-Squared:  -4.3924062285694045
Mean Absolute Percentage Error:  1.3921888300075491
Symmetric Mean Absolute Percentage Error:  70.27199564114221
```

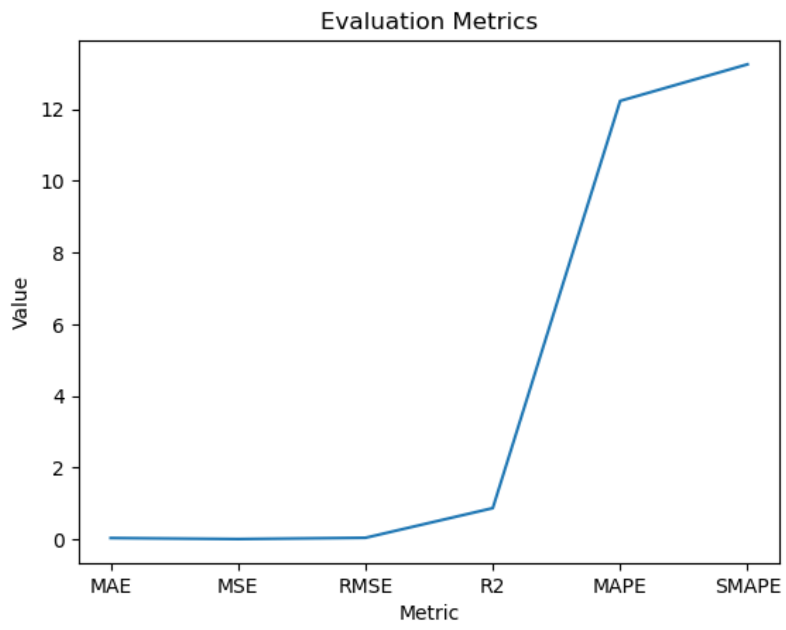

Evaluation Metrics
Value
Metric
MAE
MSE
RMSE
R2
MAPE
SMAPE
0
20
40
60
80
100
120
140


Evaluation Metrics
Value
Metric
MAE
MSE
RMSE
R2
MAPE
SMAPE
0
20
40
60
80
100
120
140

Autoencoder Deep learning Model




```
Metric-Evaluations Scores
Mean Absolute Error:  0.18979350264828196
Mean Squared Error:  0.04834014674245901
Root Mean Squared Error:  0.2198639277882095
R-Squared:  -4.2742454664544
Mean Absolute Percentage Error:  1.335522141580896
Symmetric Mean Absolute Percentage Error:  68.71928541910063
```



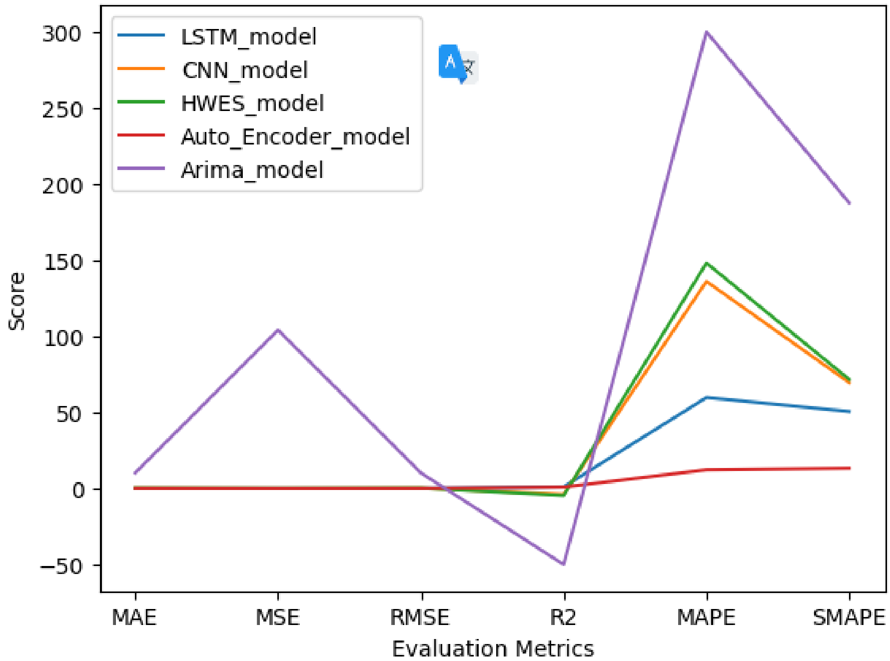
Arima Model

```
Metric-Evaluations Scores
Mean Absolute Error:  [illegible]201658799846685
Mean Squa[illegible] Error:  10[illegible]07440127752054
Root Mean Squared Error:  10.201686197757729
R-Squared:  -333726.63012891024
Mean Absolute Percentage Error:  3031.732970029223
Symmetric Mean Absolute Percentage Error:  187.58954576414624
```

Results of Evaluation Matrics of Different Model

Models	**MAE**	**MSE**	**RMSE**	**R2**	**MAPE**	**SMAPE**
LSTM	0.0060247	5.331884	0.0073	0.994	59.69	50.559
CNN	0.1828063	0.044172	0.21017	-3.94	136	69.504
HWES	0.1963942	0.050453	0.22462	-4.64	148.1	71.589
Auto_Encoder	0.0287214	0.001259	0.03549	0.864	12.23	13.255
Arima	10.201659	104.0744	10.2017	-50	300	187.59
Average	**2.123121098**	**21.90043405**	**2.13585361**	**-11.3427**	**131.2123**	**78.49934**

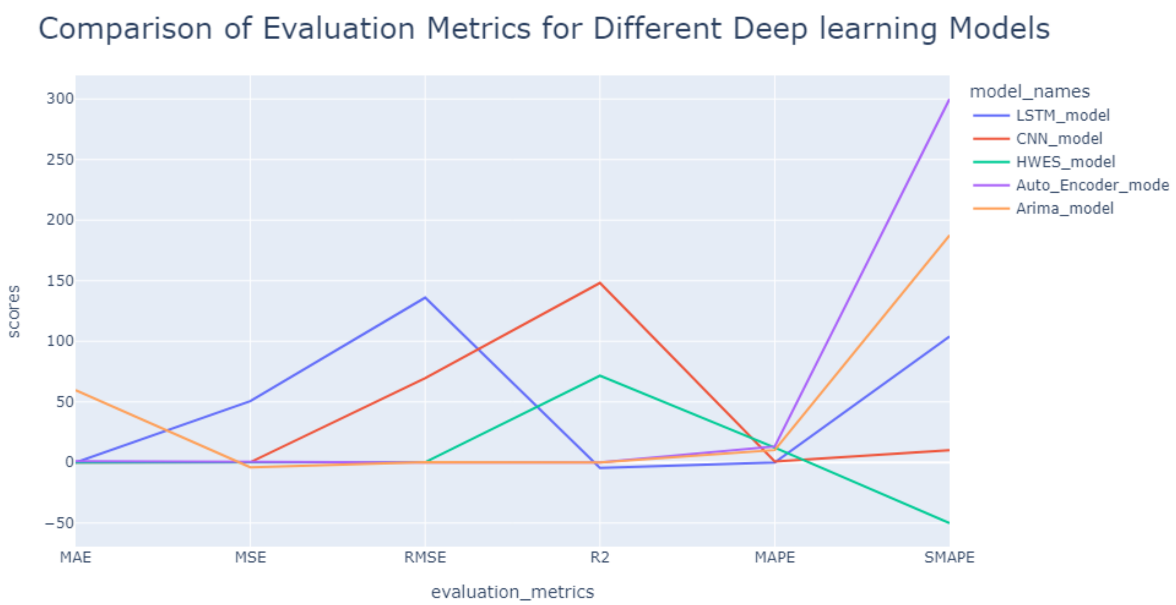
Models	MAE	MSE	RMSE	R2	MAPE	SMAPE
LSTM	0.006024717	5.331883695	0.007301975	0.99404	59.68829	50.55892
CNN	0.182806329	0.044172467	0.210172469	-3.935	136.023	69.504414
HWES	0.196394237	0.0504533	0.22461812	-4.6367	148.1169	71.588536
Auto_Encoder	0.028721409	0.001259489	0.035489286	0.86417	12.23339	13.255263
Arima	10.2016588	104.0744013	10.2016862	-50	300	187.58955
Average	2.123121098	21.90043405	2.13585361	-11.3427118	131.2123105	78.49933578
LSTM	Long Short Term Mempry in Deep learning					
CNN	Convolutional Neural Networks in Deep Learning					
HWES	Holt Winters Exponential Smoothing					
Auto_Encoder	Auto-encoder model					
Arima	Auto Regressive Integrated Moving Average					



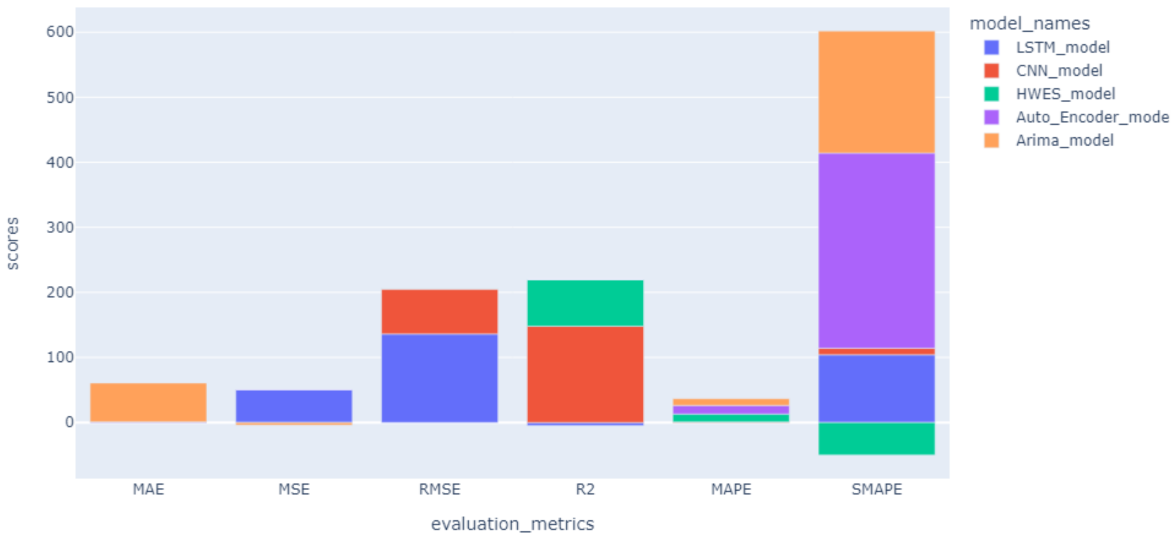
Results of Evaluation Matrics of Different Model

In [516]:

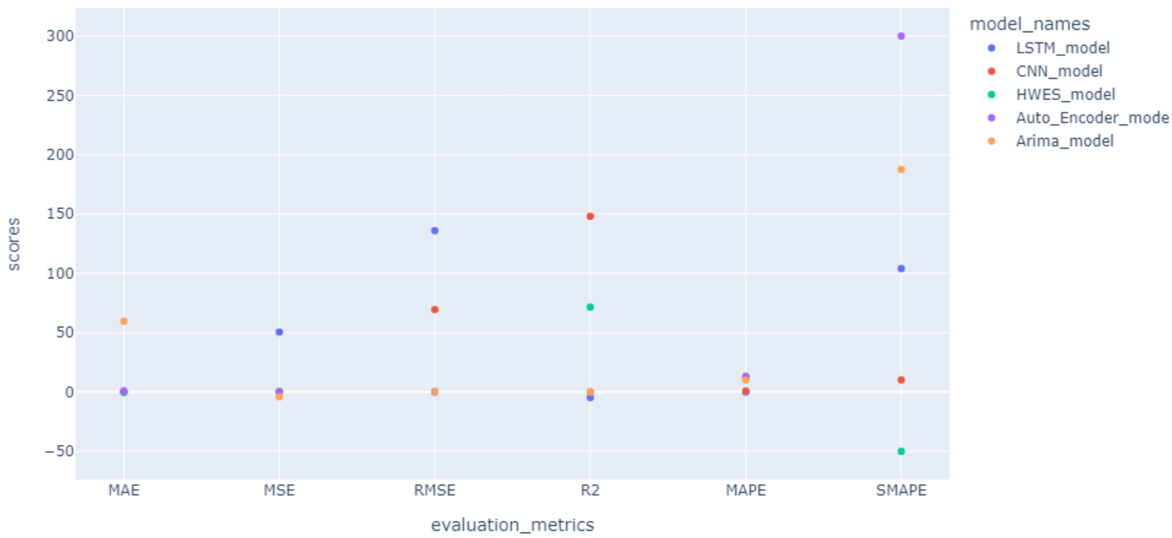
```
import matplotlib.pyplot as plt
Deep_LSTM_MODEL =LSTM_MODEL
Deep_CNN_MODEL=CNN_MODEL
Deep_HWES=HWES
Deep_Auto_encode=Auto_encode
Deep_Arima_model=Arima_model
a=['MAE', 'MSE', 'RMSE', 'R2', 'MAPE', 'SMAPE']
x = a

plt.plot(x, Deep_LSTM_MODEL, label='LSTM_model')
plt.plot(x, Deep_CNN_MODEL, label='CNN_model')
plt.plot(x, Deep_HWES, label='HWES_model')
plt.plot(x, Deep_Auto_encode, label='Auto_Encoder_model')
plt.plot(x, Deep_Arima_model, label='Arima_model')


plt.xlabel('Evaluation Metrics')
plt.ylabel('Score')
plt.legend()
plt.show()
```


Comparison of Evaluation Metrics for Different Deep learning Models
scores
300
250
200
150
100
50
0
−50
MAE
MSE
RMSE
R2
MAPE
SMAPE
evaluation_metrics
model_names
LSTM_model
CNN_model
HWES_model
Auto_Encoder_mode
Arima_model

Comparison of Evaluation Metrics for Different Deep learning Models

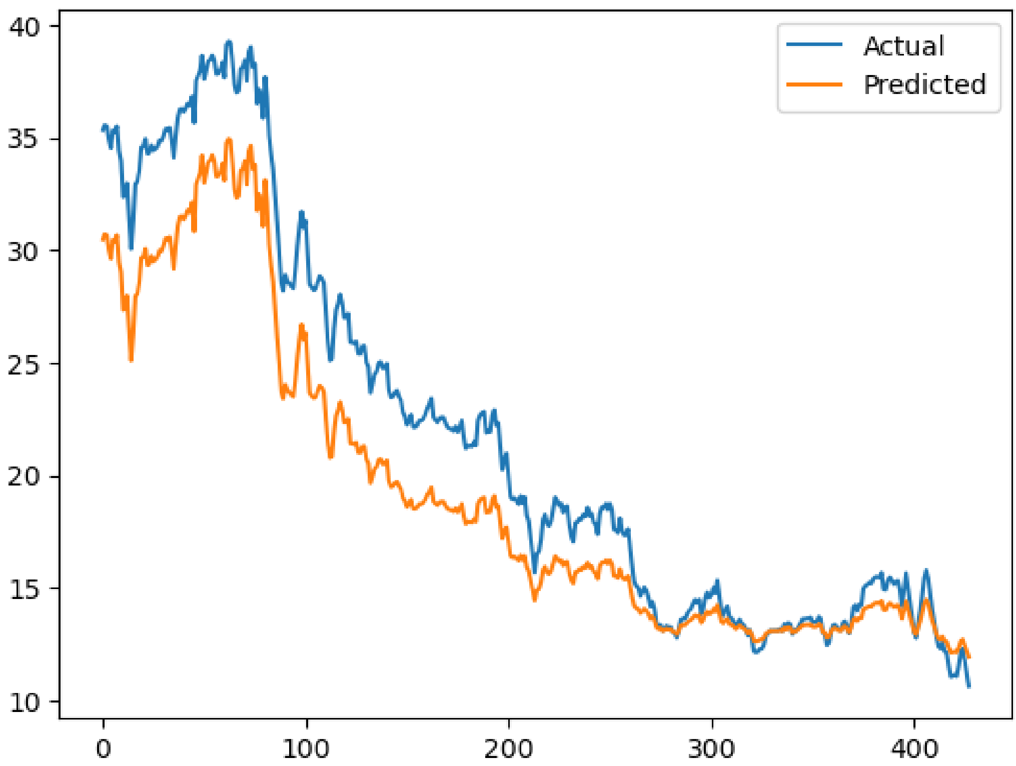


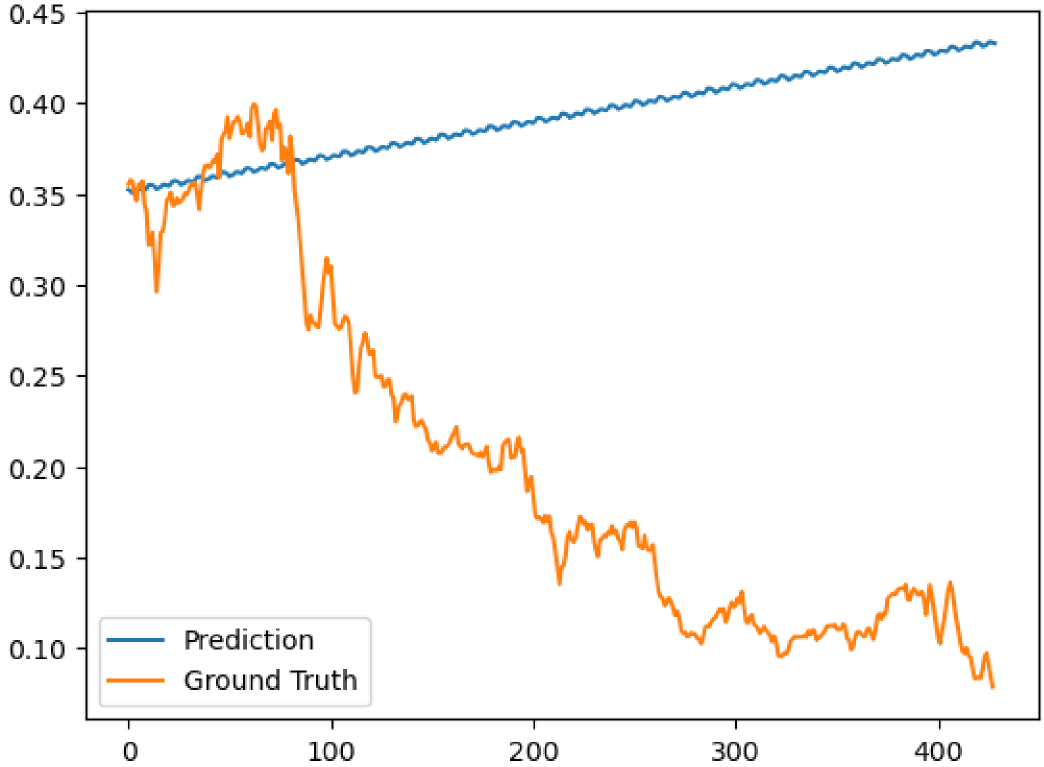
Comparison of Evaluation Metrics for Different Deep learning Models

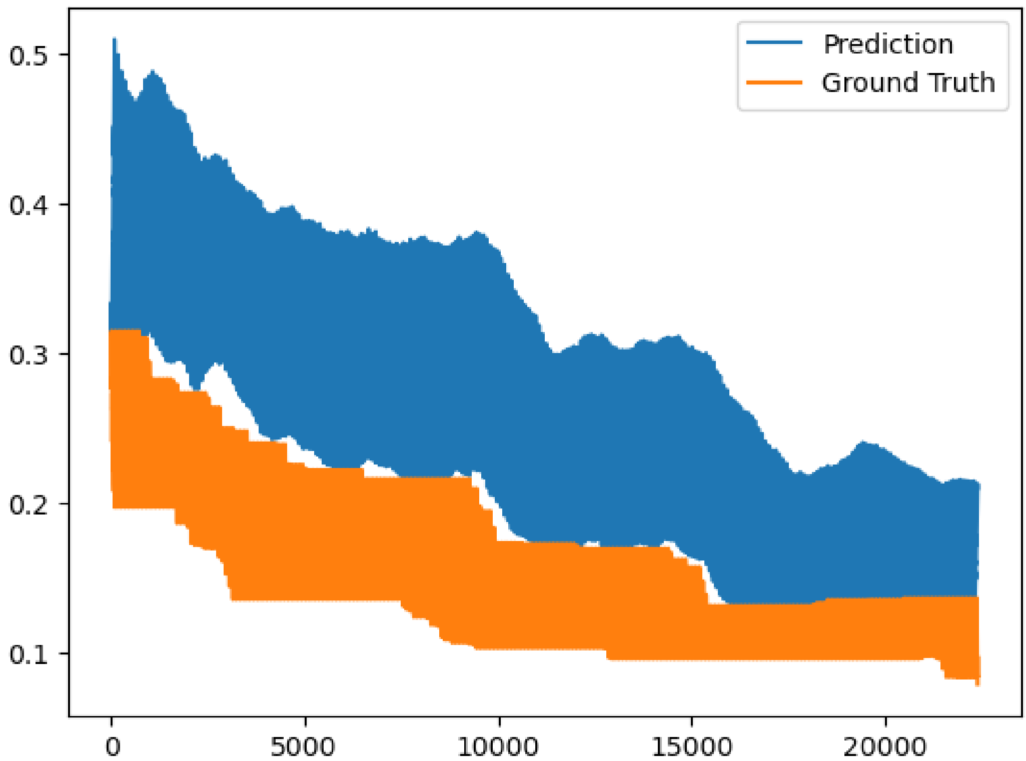


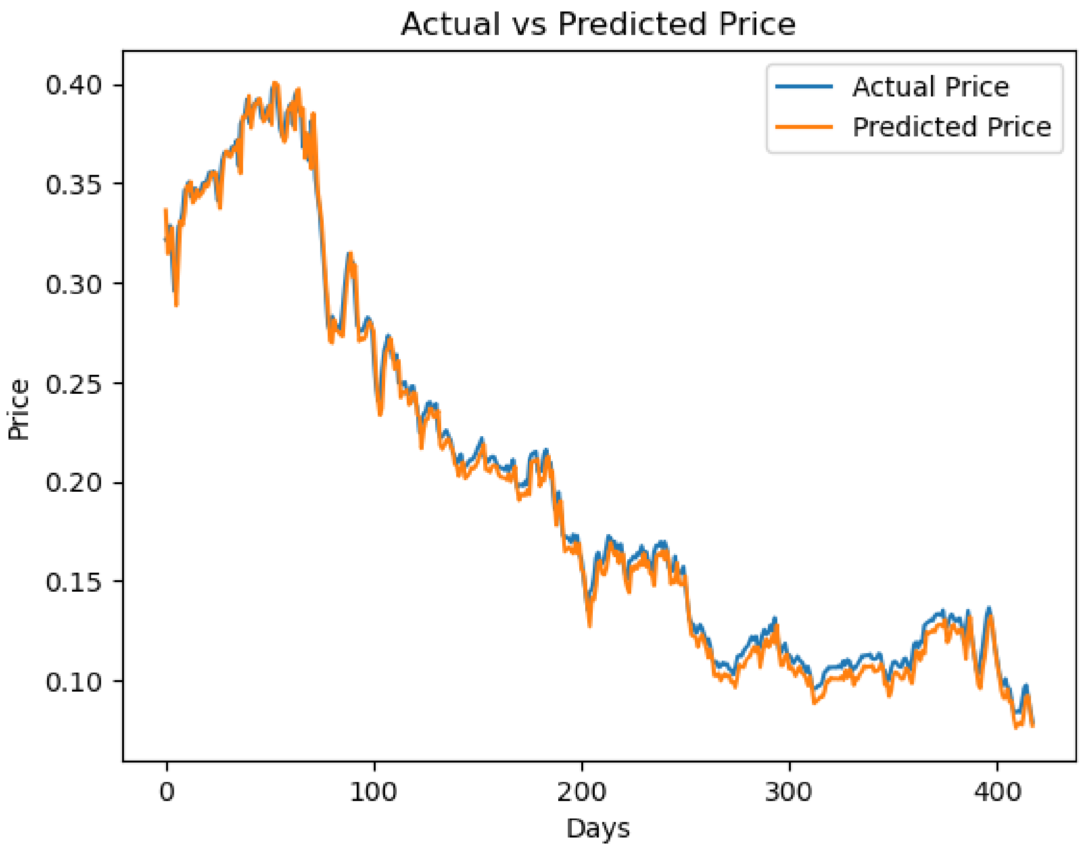
Some More Results of Project;


Predicted Price
Price
80
60
40
20
0
2015
2016
2017
2018
2019
2020
2021
2022
index


Actual
Predicted
40
35
30
25
20
15
10
0
100
200
300
400


0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0
100
200
300
400
Prediction
Ground Truth


0.5
0.4
0.3
0.2
0.1
0
5000
10000
15000
20000
Prediction
Ground Truth



Conclusion

In conclusion, the ARIMA and LSTM models are both powerful methods for stock price prediction, and they have been widely used in the literature. The ARIMA model is particularly useful for handling non-stationary time series data, while the LSTM model is particularly useful for capturing complex temporal dependencies. Using both models together can improve the prediction performance.

The Hascol 7-year data set provides a rich source of historical financial and operational data that can be used to train and evaluate these models. The methodology outlined in this research involves the steps of data preparation, model selection, model training, model evaluation, model fine-tuning, and model application.

It is important to note that stock prices are highly unpredictable, and any model or prediction is just an approximation. Therefore, it is crucial to use multiple models and techniques, as well as to gather information from different sources to make an informed decision.

In summary, the ARIMA and LSTM models have proven to be effective methods for stock price prediction and their combination can improve the prediction performance. The Hascol 7-year data set can be used to train and evaluate these models, and the methodology outlined in this research can be used as a guide for further research. However, it is important to consider that stock prices are highly unpredictable and any model or prediction is just an approximation.

References

Pramod, & Pm, Mallikarjun. (2021). Stock Price Prediction Using LSTM. Test Engineering and Management. 83. 5246-5251.

Yadav, Anita & Jha, C & Sharan, Aditi. (2020). Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science. 167. 2091-2100. 10.1016/j.procs.2020.03.257.

Dong, Yichang & Yuan, Zizi & Guo, Jing hang. (2022). Back-propagation neural network and ARIMA algorithm for price trend analysis. 73. 10.1117/12.2653166.

Nayak, Chinmaya & Singh, Anjani & Shukla, Kirti & Prakash, Suraj. (2021). Stock Market Prediction Using Deep Learning. Web ology. 18. 4686-4694.

Low, Pi & Sack, Eric. (2022). Parameter Optimization of LSTM Models in Stock Price Prediction. 1-5. 10.1109/TENCON55691.2022.9978045.

Madhav, & Arya, Meenakshi. (2022). Up Move Down Move Strategy Using Neural Network Techniques for the Prediction of Stock Prices. 10.1007/978-981-19-5331-6_19.

Lewi, Armin & Mesa, Hendra & Amir, Supari. (2021). Implementation of Long Short-Term Memory and Gated Recurrent Units on Grouped Time-Series Data to Predict Stock Prices Accurately. 10.21203/rs.3.rs-1057875/v1.