CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Chandubhai S Patel Institute of Technology

Research Project

Stock Price Prediction

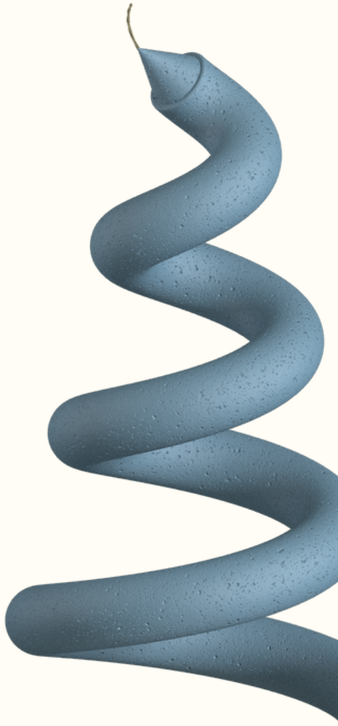
Guided by - Prof. Amit Thakkar
Co-Guided by - Prof. Hemang Thakar

21CS012 - Dhairya Gohil
21CS021 - Jay Kanjariya
21CS023 - Meet Kothari

Content

Problem Statement

Develop a deep learning-based stock price prediction model that can help people make better decisions about stocks by looking at things like past price.



Technology

Machine Learning and Deep Learning Algorithms

Tiingo API

Library

numpy

pandas , pandas_datareader

matplotlib

sklearn

tensorflow

keras

Purpose...

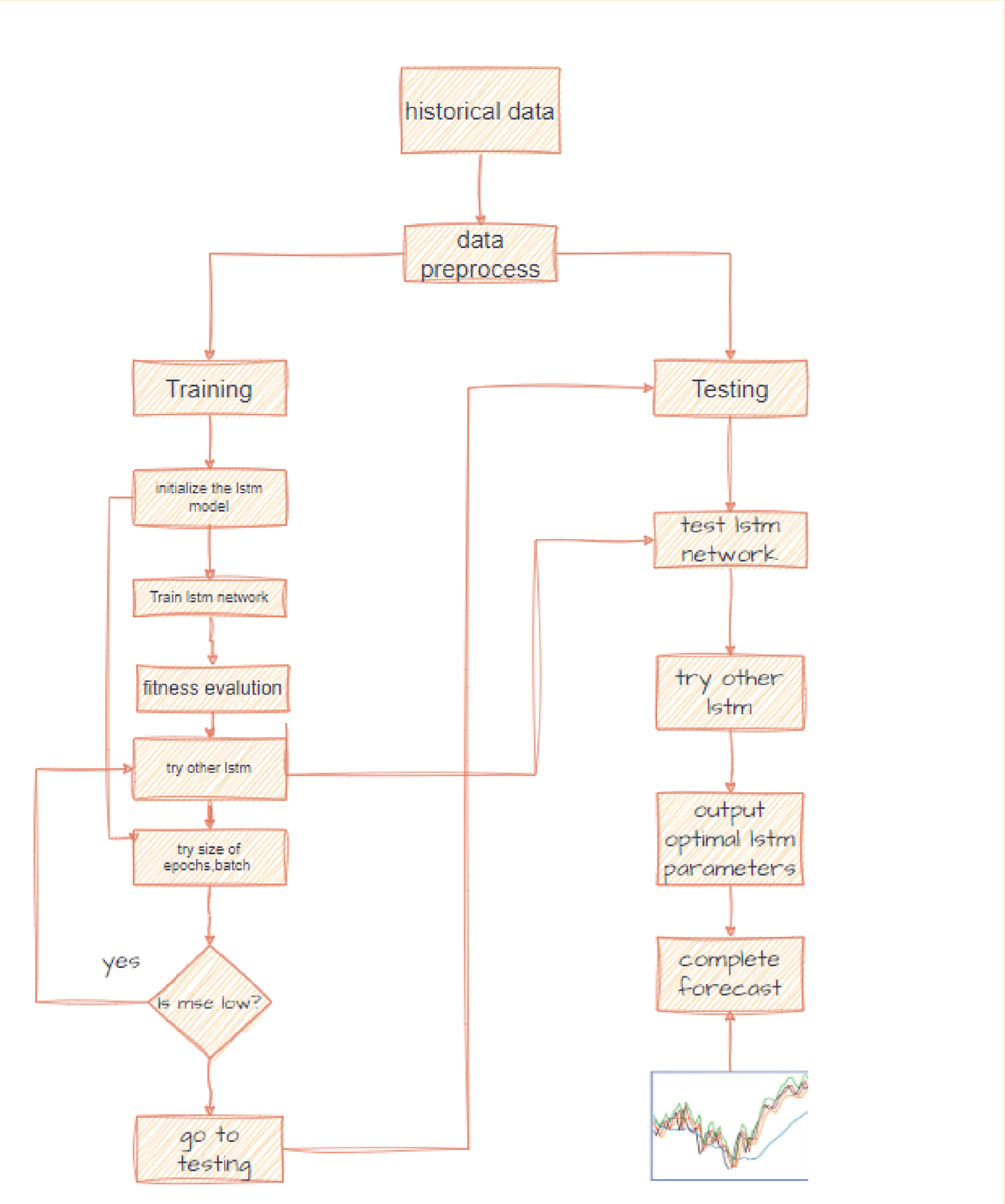
The primary purpose of using machine learning and deep learning algorithms for stock price prediction is to provide investors with a valuable tool for making informed and data-driven decisions, ultimately aiming to enhance their chances of achieving profitable outcomes in the stock market while also reducing risk and saving time for traders.

Literature Review.....

A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing

In this paper, they presented several approaches to stock price and movement prediction on a weekly forecast horizon using eight regression and classification methods. These models are based on machine learning and deep learning approaches. but we have done lot more variation of lstm to predict more accurate.

Workflow....


historical data
data preprocess
Training
initialize the lstm model
Train lstm network
fitness evalution
try other lstm
try size of epochs,batch
yes
Is mse low?
go to testing
Testing
test lstm network
try other lstm
output optimal lstm parameters
complete forecast

Data Gathering.....

we have use tiingo api to get data for our model

Data preprocess...

MIN-MAX scaler

- Min-Max scaling is beneficial when features have different scales, as it helps models converge faster and prevents features with larger scales from dominating the learning process. It's commonly used in machine learning to prepare data for algorithms like neural networks and support vector machines.

Data preprocess...

Train - testing split

- train-test splitting divides a dataset into two subsets: the training set used to train the model and the test set used to evaluate its performance. This ensures that the model's effectiveness can be assessed on unseen data.

- Typically, data is split into a larger portion for training (e.g., 70-80%) and a smaller portion for testing (e.g., 20-30%). The exact ratio depends on the dataset size and specific requirements, with larger training sets usually leading to more robust models.

Models

LSTM

LSTM + Dense

LSTM with Regularization

LSTM – CNN Hybrid

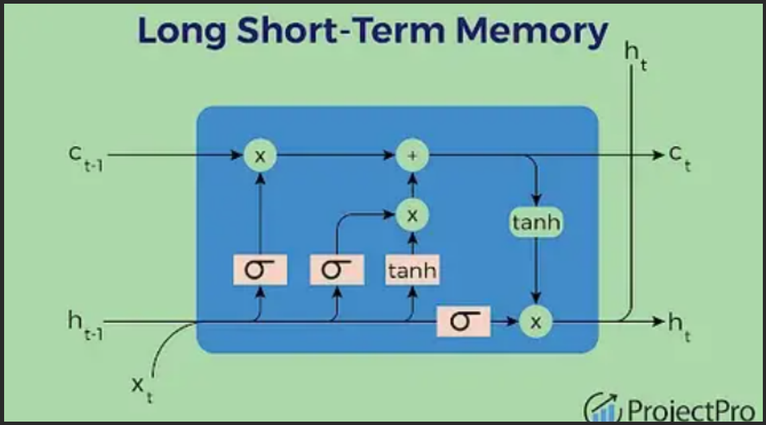
GRU

LSTM + GRU

Models brief....

LSTM

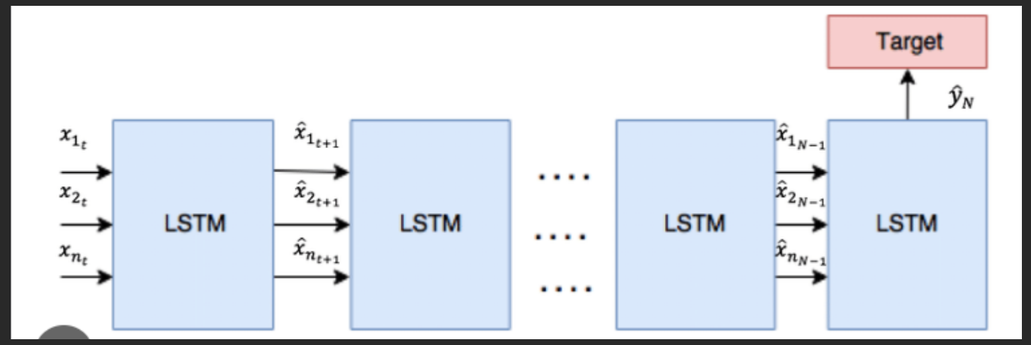
LSTM (Long Short-Term Memory): LSTM is a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data, making it suitable for tasks like time series prediction.




```
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

LSTM+Dense

This model combines LSTM layers with dense (fully connected) layers to learn complex patterns and make predictions. It's often used in sequential data tasks with additional processing in the dense layers.




```
model_combined = Sequential()
model_combined.add(LSTM(50, return_sequences=True, input_shape=(100, 1)))
model_combined.add(LSTM(50, return_sequences=True))
model_combined.add(LSTM(50))
model_combined.add(Dense(50, activation='relu'))
model_combined.add(Dense(1))
model_combined.compile(loss='mean_squared_error', optimizer='adam')
```

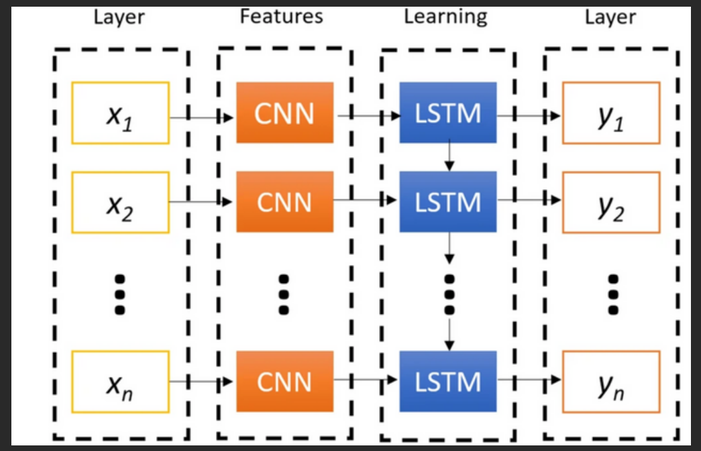
LSTM with regularization

LSTM with regularization works by incorporating techniques like dropout, L1/L2 regularization, or gradient clipping during training to prevent overfitting and improve the model's ability to generalize to unseen data. This helps the LSTM network become more robust and perform better on real-world tasks

```
model_lstm_dropout = Sequential()
model_lstm_dropout.add(LSTM(50, return_sequences=True, input_shape=(100, 1)))
model_lstm_dropout.add(Dropout(0.2))  # Dropout layer to prevent overfitting
model_lstm_dropout.add(LSTM(50, return_sequences=True))
model_lstm_dropout.add(Dropout(0.2))
model_lstm_dropout.add(LSTM(50))
model_lstm_dropout.add(Dense(1))
model_lstm_dropout.compile(loss='mean_squared_error', optimizer='adam')
```

LSTM+CNN

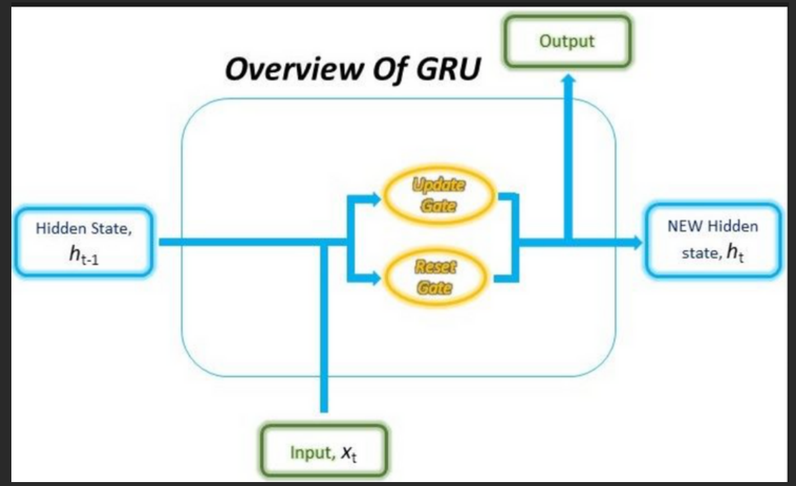
LSTM - CNN Hybrid: This hybrid model combines LSTM and Convolutional Neural Networks (CNNs) to capture both sequential and spatial patterns in data, often used in tasks involving sequences with spatial features (e.g., text with word embeddings).




```
model_cnn_lstm = Sequential()
model_cnn_lstm.add(Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(100, 1)))
model_cnn_lstm.add(MaxPooling1D(pool_size=2))
model_cnn_lstm.add(LSTM(50, return_sequences=True))
model_cnn_lstm.add(LSTM(50))
model_cnn_lstm.add(Dense(1))
model_cnn_lstm.compile(loss='mean_squared_error', optimizer='adam')
```

GRU

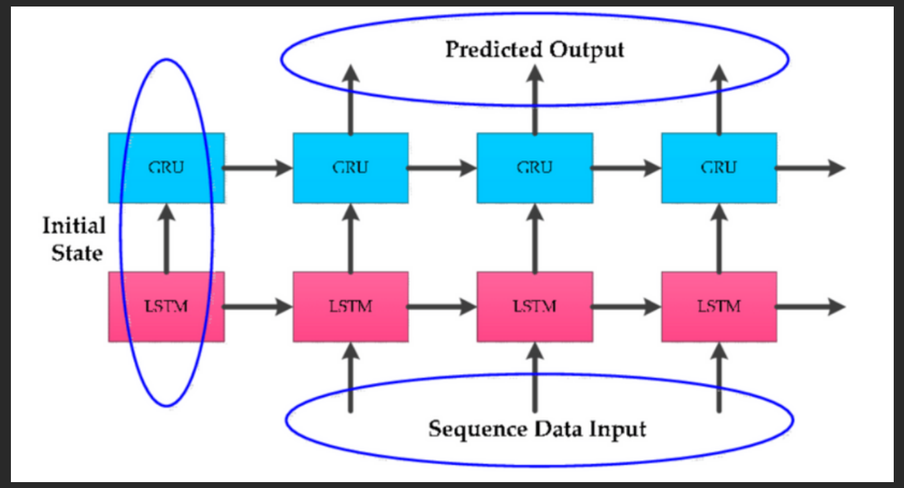
GRU is a simplified recurrent neural network architecture that uses gating mechanisms (update and reset gates) to control the flow of information through the network. It can capture Long-range dependencies in sequential data while being computationally more efficient than LSTMs due to its reduced number of gates. GRUs are particularly useful for tasks involving sequential data, such as natural language processing and time series analysis.




```
model_gru = Sequential()
model_gru.add(GRU(50, return_sequences=True, input_shape=(100, 1)))
model_gru.add(GRU(50, return_sequences=True))
model_gru.add(GRU(50))
model_gru.add(Dense(1))
model_gru.compile(loss='mean_squared_error', optimizer='adam')
```

LSTM+GRU

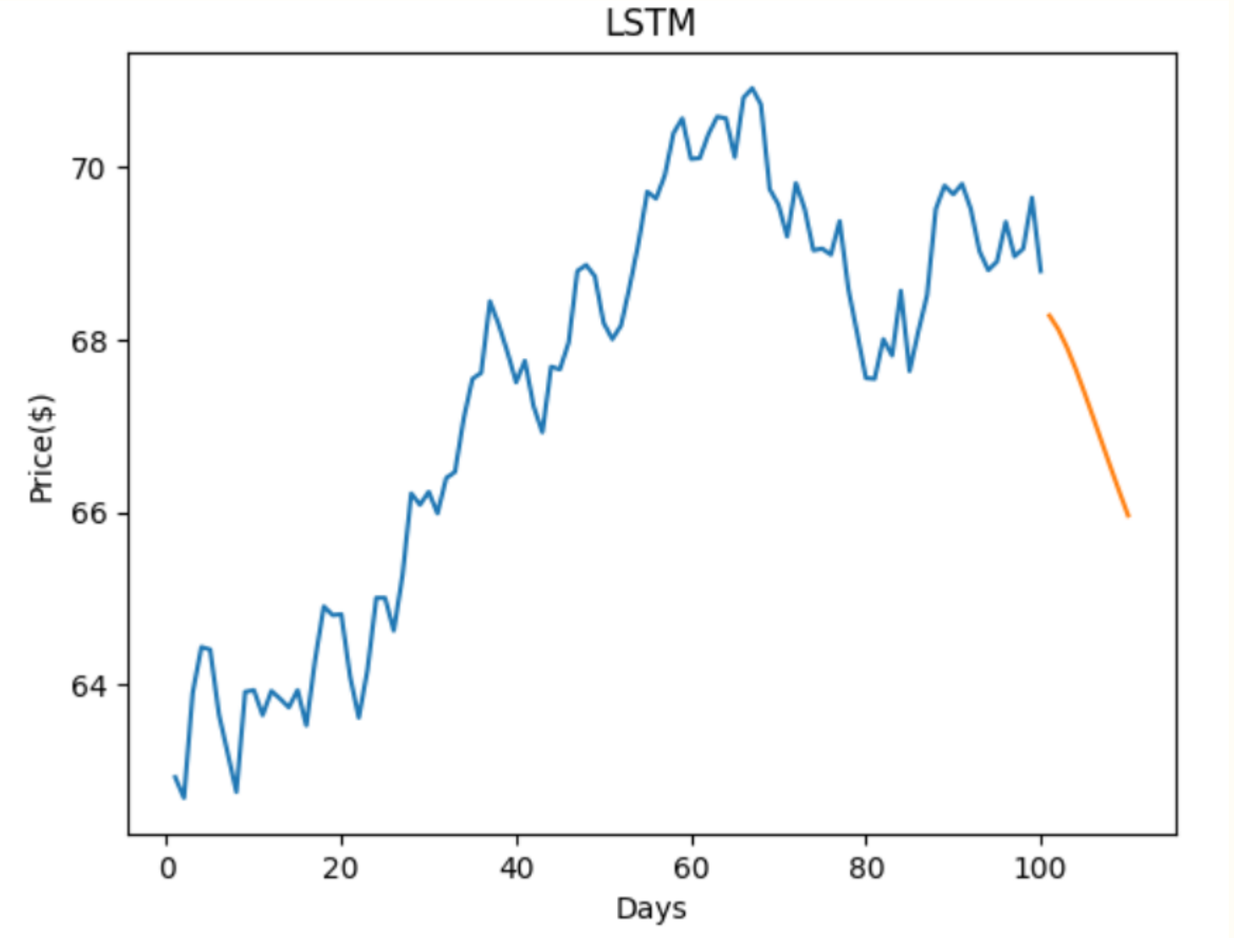
combining LSTM and GRU layers in a neural network architecture leverages the strengths of both architectures, allowing the model to capture a wider range of sequential dependencies. This hybrid approach can lead to improved performance in tasks involving sequential data



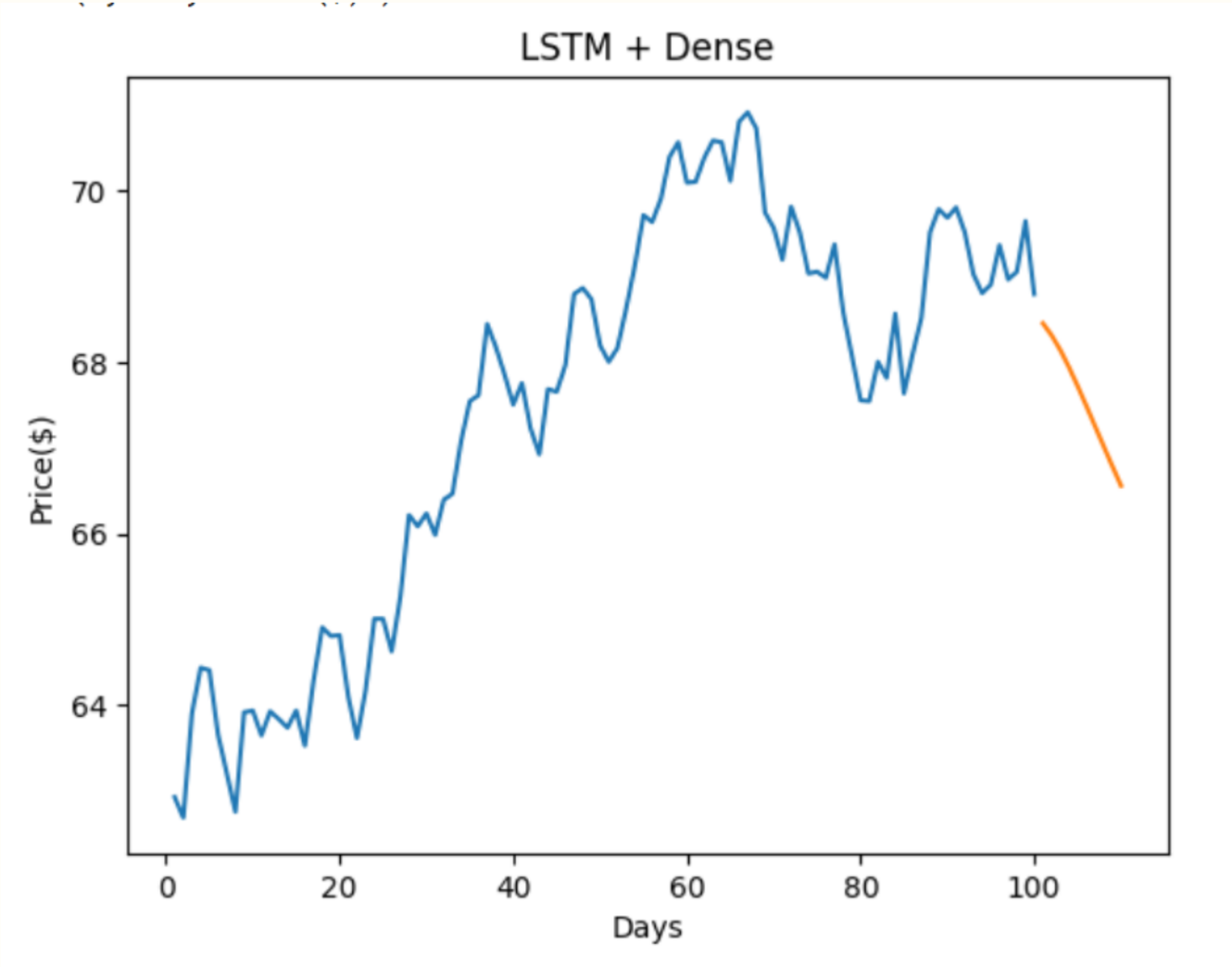

```
model_lstm_gru = Sequential()
model_lstm_gru.add(LSTM(50, return_sequences=True, input_shape=(100, 1)))
model_lstm_gru.add(LSTM(50, return_sequences=True))
model_lstm_gru.add(GRU(50, return_sequences=True))  # Adding a GRU layer
model_lstm_gru.add(GRU(50))
model_lstm_gru.add(Dense(1))
model_lstm_gru.compile(loss='mean_squared_error', optimizer='adam')
```

Results...

LSTM

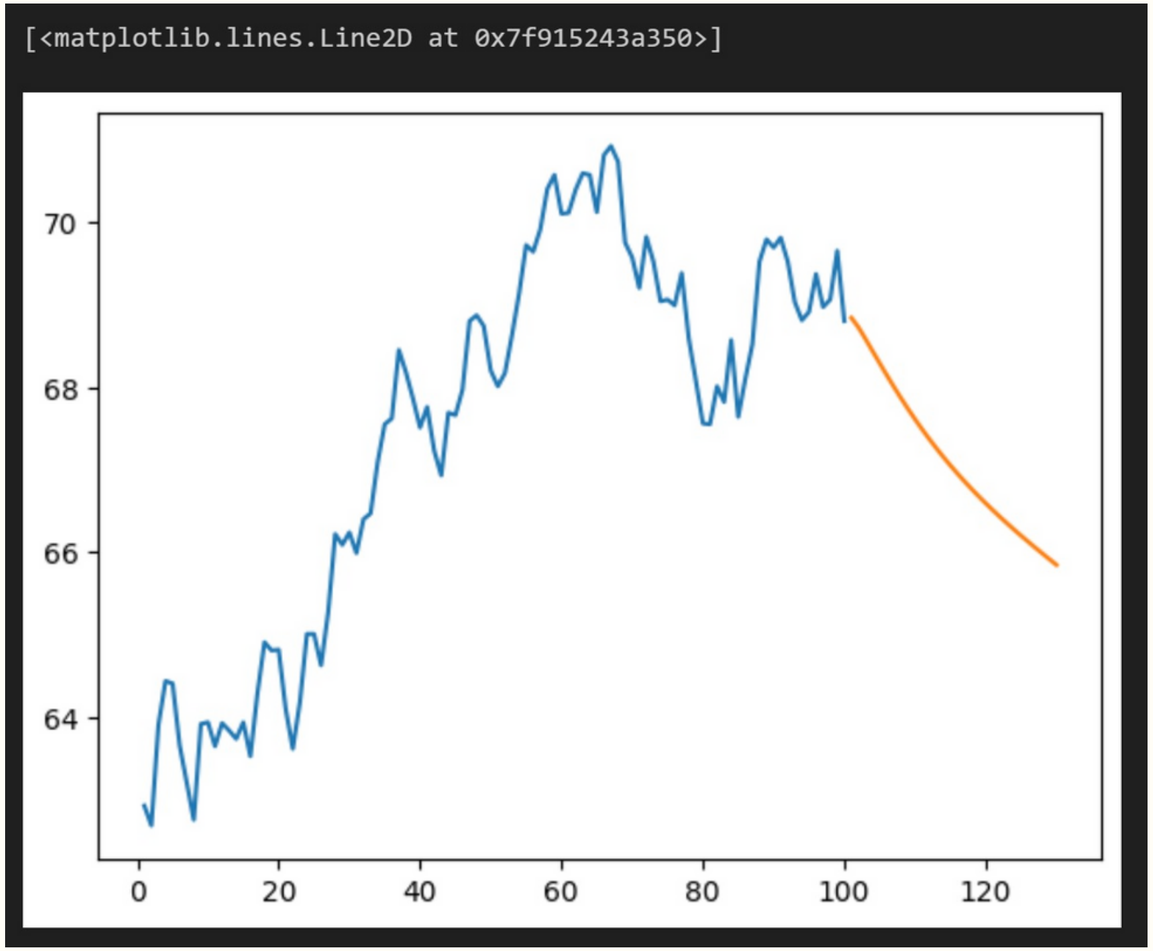


LSTM+Dense



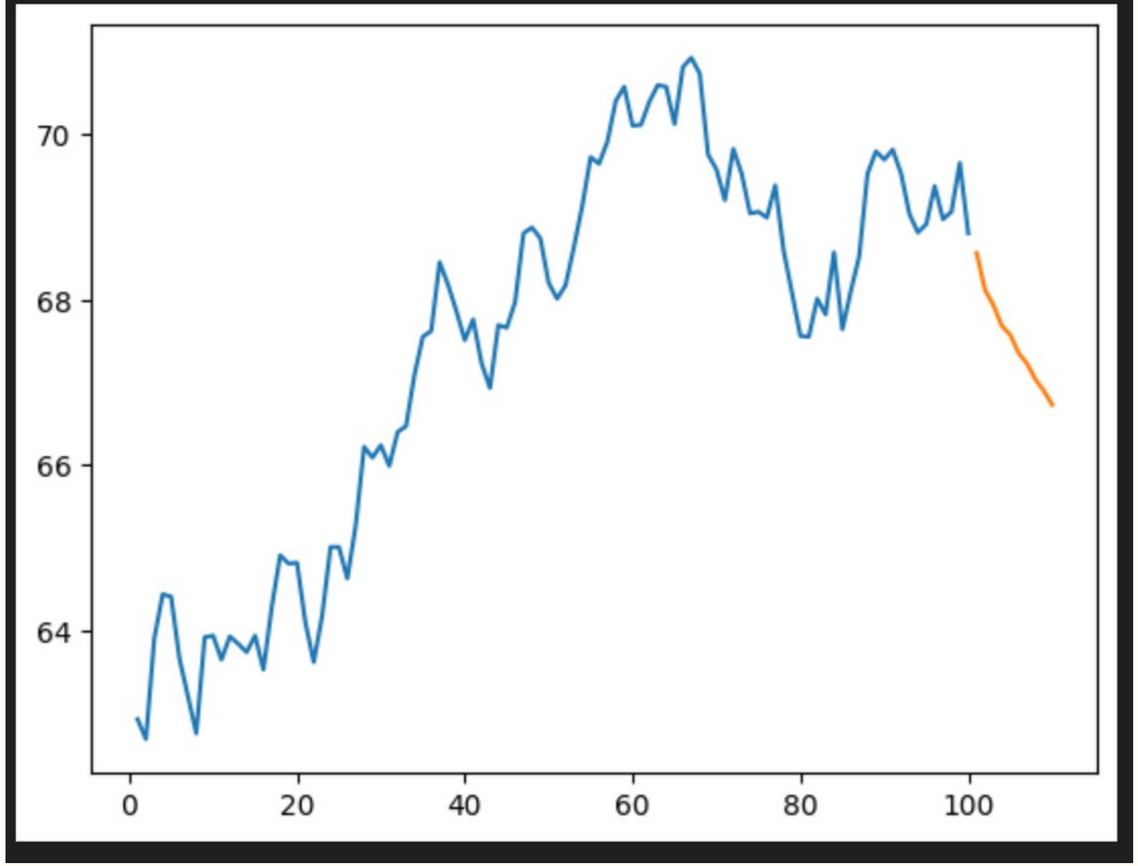
LSTM+Regularization

```
[<matplotlib.lines.Line2D at 0x7f915243a350>]
```



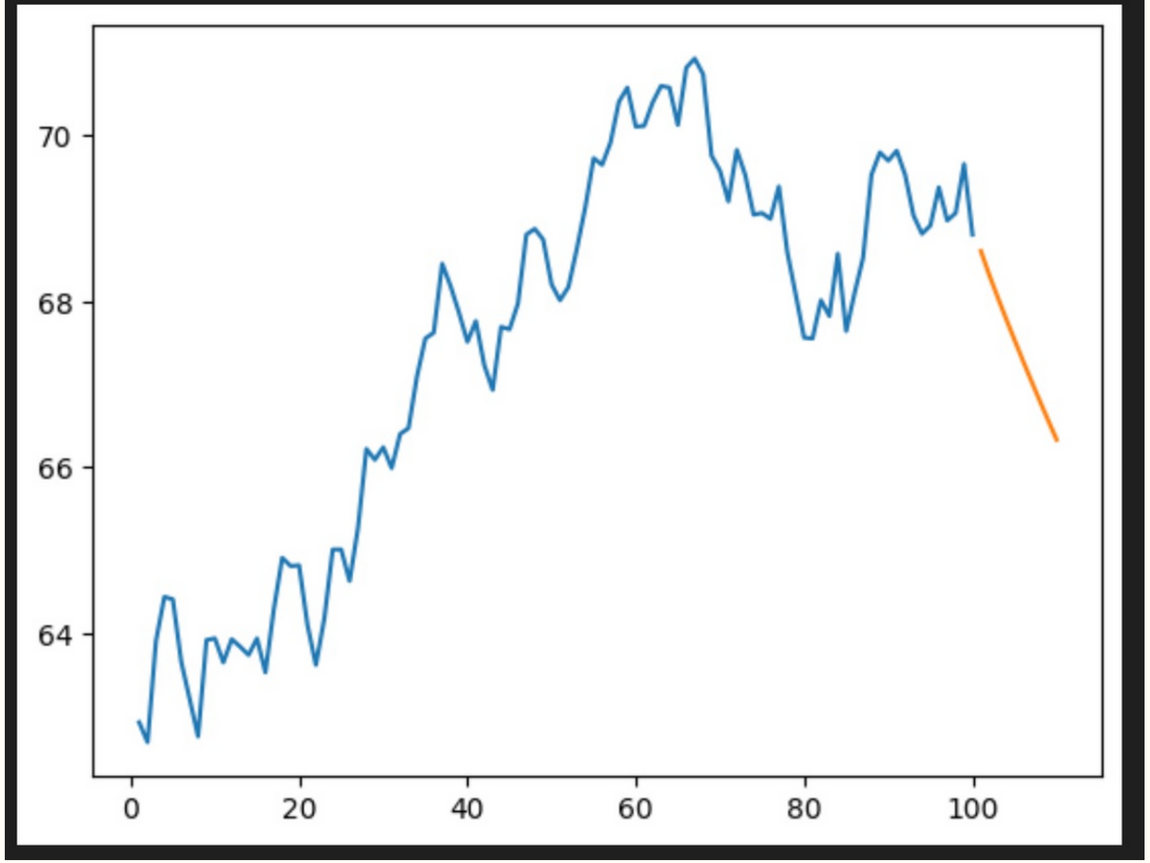
LSTM+CNN

```
[<matplotlib.lines.Line2D at 0x7fa1efcb1de0>]
```



LSTM+GRU

```
[<matplotlib.lines.Line2D at 0x7c8f19c599f0>]
```



Real market analysis

Comparison (RMSE)

```
LSTM :  64.5013189261503

LSTM + Dense nureal :  39.35552701324161

LSTM + Regularization :  39.30781271156046

LSTM-CNN Hybride :  39.93382418025153

GRU :  0.026886604778902017

LSTM + GRU :  0.017287859211553253
```

References

Mehtab, Sidra, and Jaydip Sen. "A robust predictive model for stock price prediction using deep learning and natural language processing." arXiv preprint arXiv:1912.07700 (2019).

Moghar, Adil, and Mhamed Hamiche. "Stock market prediction using LSTM recurrent neural network." Procedia Computer Science 170 (2020): 1168-1173.

Mehtab, Sidra, and Jaydip Sen. "Stock price prediction using CNN and LSTM-based deep learning models." 2020 International Conference on Decision Aid Sciences and Application (DASA). IEEE, 2020.

Gao, Ya, Rong Wang, and Enmin Zhou. "Stock prediction based on optimized LSTM and GRU models." Scientific Programming 2021 (2021): 1-8.

Conclusion

we have implemented 6 types of deep learning model and from that LSTM+GRU have Less Mean square error.

we have concluded that how the market trends goes upward or downwards.

Future Improvement

We will do Sentiment analysis of News headline / Social Media data to predict stock price more accurately and try more variations of deep learning algorithms for more accuracy.

Thank you