Soc assignment 3 (report)

22b3901- Hardik jangir

Overview:

In this report, we will be exploring several advanced models to understand their basics and evaluate their usefulness. Our focus will be on the **Auto Regressor (ARIA)**, **Time GAN**, **CNN-LSTM**, and **Transformer** models. Analyzing their **Hyperparameters** , their **Predictions** and their **Usefulness.**

Auto Regressor (ARIMA)

- **How It Works**: This model predicts future stock prices based on past prices. It assumes that past price trends and patterns will continue into the future. Essentially, it uses historical data to estimate what the next price will be, following the same pattern.

  The ARIMA model is configured with an order of (5, 1, 0). This means it uses 5 lagged observations (p), 1 difference (d) to make the data stationary, and no moving average component (q). The model is fitted to the historical data up to a specified split point using the ARIMA class, and then the model is trained with the fit() method.

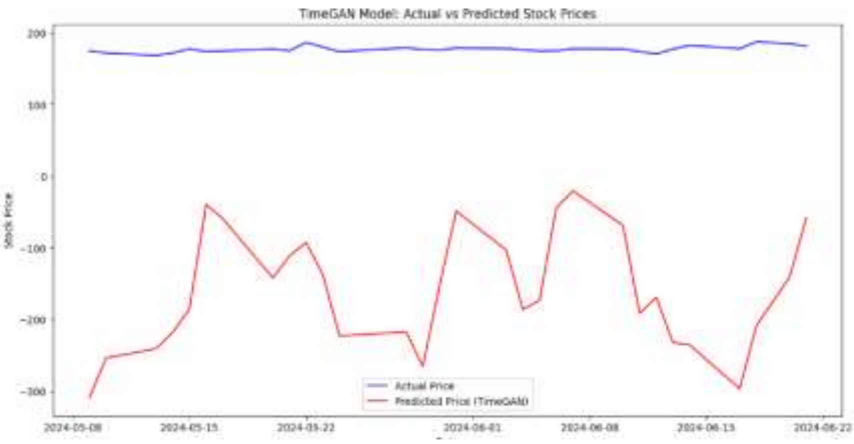
ARIMA Model MSE: 141.2007429859356

Why It's Useful:
The Auto Regressor is good for predicting stock prices because it relies on historical data to make forecasts. Its key characteristic is its simplicity and direct approach. By analyzing past prices, it can provide straightforward predictions based on the assumption that historical patterns will continue.

Time GAN

- **How It Works**: Time GAN generates new data based on existing time-series data. It learns the patterns from historical stock prices and creates synthetic data that mimics these patterns. This can be useful for improving predictions by providing more data or filling in gaps.

  For the Time GAN model, the key hyperparameters are as follows: **seq_length** is set to 60, defining the length of the input sequences. The **n_features** is 1, since we're working with a single stock price feature. The model is trained for **epochs** of 10,000 iterations, with a **batch_size** of 128 samples per batch. The **learning_rate** for the Adam optimizer is 0.0002, and the **beta_1** parameter is 0.5, which controls the decay rate for the 1st moment estimates in the optimizer.



TimeGAN Model MSE: 120670.088237232

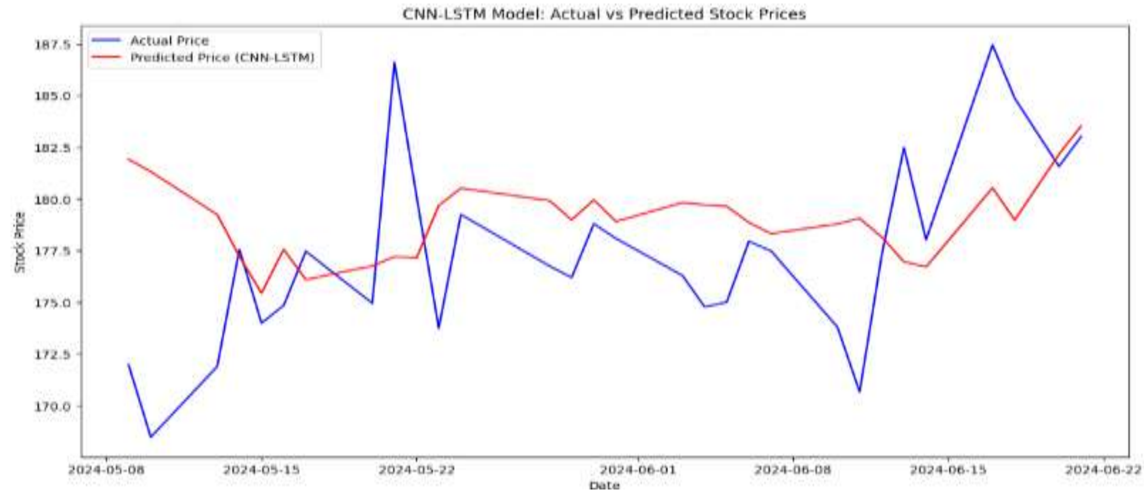
Why It's Useful:

Time GAN is useful for generating synthetic time-series data that mimics real data patterns. This is beneficial for training models when data is limited or incomplete. Its key feature is its ability to create realistic data that can help improve model performance and robustness by providing more diverse training examples.

CNN-LSTM

- **How It Works**: This model combines two approaches: Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs). CNNs are used to extract important features from stock price data, while LSTMs handle the sequence and time aspect, capturing trends over time. Together, they can analyze complex patterns and make more accurate predictions.

  The CNN-LSTM model is defined using a **Sequential** architecture. It starts with a **Conv1D** layer with 64 filters, a kernel size of 2, and ReLU activation. This is followed by a **MaxPooling1D** layer with a pool size of 2. The model then includes an **LSTM** layer with 50 units and ReLU activation. Finally, a **Dense** layer with 1 unit outputs the prediction. The model is compiled with the **Adam** optimizer and **mean squared error (MSE)** loss function. It is trained for 20 epochs with a **batch size** of 32, using 20% of the data for validation and an **EarlyStopping** callback to monitor validation loss with a patience of 10 epochs.



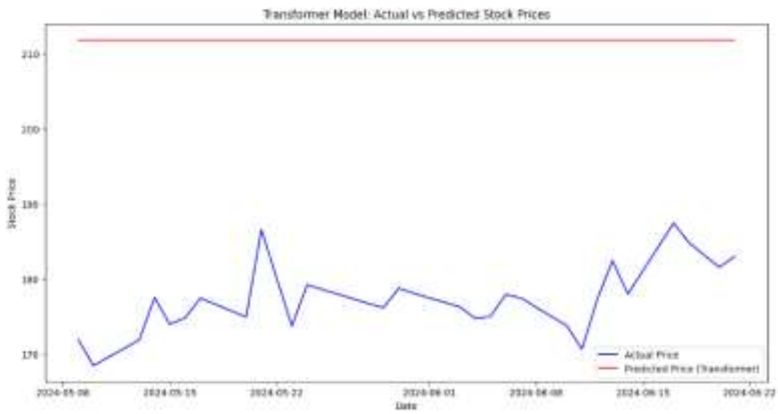
CNN-LSTM Model MSE: 25.242830563730834

Why It's Useful:
The CNN-LSTM model combines the strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. CNNs are effective at extracting features from data, while LSTMs capture temporal dependencies and trends over time. This combination allows the model to understand both detailed features and long-term patterns, making it powerful for sequential data like stock prices.

Transformer Model

- **How It Works**: Transformers use an attention mechanism to focus on different parts of the data, allowing them to handle long sequences effectively. They analyze the relationships between different time points in the stock price data and consider long-term dependencies, which helps in understanding complex patterns and making predictions.

The Transformer model is configured with **8 attention heads** and a **model dimension (d_model)** of 512. It has **2 encoder layers** and **2 decoder layers**, with a feedforward dimension of 2048. The dropout rate is set to 0.1 to prevent overfitting. The model is trained for **1 epoch** with a **batch size** of 32. The Transformer is instantiated with these hyperparameters and compiled using the **Adam** optimizer and **mean squared error (MSE)** loss function.



Transformer Model MSE: 1208.0743044622154

Why It's Useful: The Transformer model excels at handling long sequences of data due to its attention mechanism, which focuses on different parts of the data to understand context and dependencies. This makes it particularly good for capturing complex patterns and relationships over long time periods, which is crucial for accurate stock price predictions