CS337 : Artificial Intelligence and Machine Learning **July-Nov 2023**

End-term Project Report : Stock Market Indices Prediction

Team Name: Stock Prophets *Team Members: 210050043,210050120,210050051,210050153*

Abstract

In this project we aim to explore LSTM and GRU architectures on our project which aims to predict stock indices with data of the market from the time of its inception. We describe the architectures involved in our project, the pre-processing. Also the process of identifying the relevent quantities for predicting the closing price is presented here with graphs(with and without)

1 Introduction

Predicting stock market movements is a complex task that requires a nuanced understanding of financial data and market trends. In the realm of machine learning, the choice between discriminative and generative models plays a crucial role in modeling the underlying data distributions. Our project focuses on the discriminative approach, employing LSTM and GRU architectures to forecast stock indices.

In discriminative models, we eschew imposing explicit priors on the data, concentrating solely on optimizing the likelihood. The **LSTM** (Long Short-Term Memory) and **GRU** (Gated Recurrent Unit) architectures, both types of recurrent neural networks (RNNs), excel in capturing sequential dependencies and temporal patterns present in time-series data. Unlike traditional models such as logistic regression, these networks are adept at learning long-range dependencies, making them well-suited for predicting the dynamic and evolving nature of stock prices.

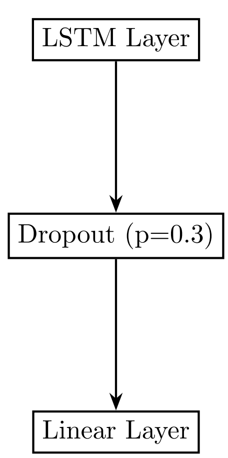
LSTMs and **GRUs** stand out in their ability to mitigate the vanishing gradient problem, a common challenge in training deep neural networks on sequential data. The LSTM's memory cell and the GRU's gating mechanism enable the models to retain and selectively update information over time, enhancing their capacity to capture intricate patterns in financial time series.

In our project, we construct a predictive model that leverages the power of LSTM and GRU architectures to analyze historical stock data and make informed predictions about future market movements. By focusing on the discriminative aspect of modeling, we aim to optimize the likelihood and enhance the accuracy of our predictions. Through this work, we explore the potential of advanced recurrent neural networks in the domain of financial forecasting, with a particular emphasis on stock indices.

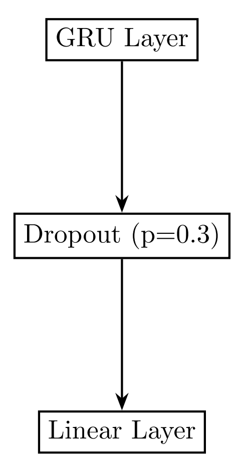
2 MODELS

In this section, we discuss all the models that we will be using in our project

A. LSTM Model



B. GRU Model



Input taking

The inputs are imported from yfinance library which uses the Ticker module, allowing us to access ticker data in a more Pythonic way. A particular company data has the following format

date	open	high	low	close	volume	dividends	stock splits
——	——	——	——	——	——	——	——

In all these quantities, for predicting the closing price, we felt including dividends in the feature set will be more relevant because dividends are often seen as a sign of financial health and stability. The announcement of dividends or changes in dividend policies can influence market expectations. For example, if a company increases its dividend, it might be interpreted as a positive signal about the company's future prospects, and investors may respond by bidding up the stock price.

Only relevant data like closing price$[p_1,p_2....p_n]$ and dividends$[d_1,d_2....d_n]$ are extracted and this data after passing through Timeseriesgenerator produces data (data) in the following format .When taking dividends also into consideration, in data[0] each p_i, it is coupled with corresponding d_i.

index	data[0]	data[1]
0	$[p_0,p_1...p_k]$	p_{k+1}
1	$[p_1,p_2...p_{k+1}]$	p_{k+2}
.	.	.
.	.	.
i	$[p_i,p_{i+1}...p_{i+k-1}]$	p_{i+k}
.	.	.
.	.	.
n-k-1	$[p_{n-k},p_{n-k+1}...p_{n-1}]$	p_n

index	data[0]	data[1]
0	$[[p_0,d_0],[p_1,d_1]...[p_k,d_k]]$	p_{k+1}
1	$[[p_1,d_1],[p_2,d_2]...p_{k+1}]$	p_{k+2}
.	.	.
.	.	.
i	$[[p_i,d_i],[p_{i+1},d_{i+1}]...[p_{i+k-1},d_{i+k-1}]]$	p_{i+k}
.	.	.
.	.	.
n-k-1	$[[p_{n-k},d_{n-k}],[p_{n-k+1},d_{n-k+1}]...[p_{n-1},d_{n-1}]]$	p_n

Training

The LSTM serially takes the lists in data[0] as X_{input} and and data[1] as Y_{input} and updates the weights conforminf with error from its prediction. We have used an 80-10-10 split for the train-validation-test split. The results of the model on test set are plotted in RESULTS section .

3 Experimental Settings

Hyperparameters

The Stock Predictor has the following hyperparameters to be tuned: while validating the models using the validation set

1. *n_layers* signifying number of layers in LSTM/GRU
2. *num_epochs* signifying number of epochs
3. *hidden_dim* signifying hidden dimension in LSTM/GRU. This is the shape of the output provided by LSTM.
4. *learning_rate* signifying Learning Rate

Tuning

Our choices of n_layers are 1,2,3 and num_epochs are 50, 100, 150. We use the validation set to get the best model out of all possibilities and use it for the test sets. So, the set of hyperparameters varies based on the company. Here are the list of companies we observed and the values we got for LSTM(left) and GRU(right)

Company	num_epochs	layers
GOOGLE	100	1
APPLE	100	1
S& G	100	1

Company	num_epochs	layers
GOOGLE	50	2
APPLE	50	1
S& G	100	1

4 RESULTS

Generated Samples

- Table 1 has the graphs of stock predicted for google including dividends in feature set, and also testing for a sample of 10 days in third row if we can predict for more than 1 day accurately, i.e if we have 100 days of data, we predict 101^{th} day price, now using 2 to 100 and our predicted value, if we predict the 102^{th} day and so on till 110^{th} day price (orange line)
- Table 2 has the graphs of stocks predicted using LSTM model and GRU model for S&P with and without including dividends in feature set.
- Table 3 has the graphs of stock predicted using LSTM model and GRU model of NIFTY and APPLE (in that order) while including dividends in feature set.

GOOGLE WITH DIVIDENDS

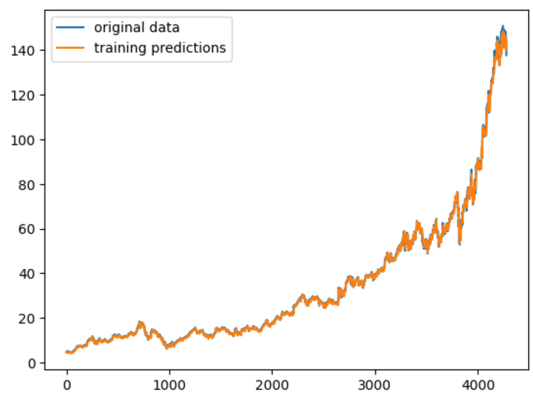
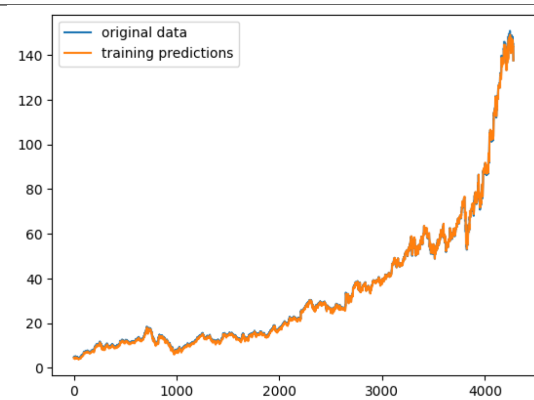
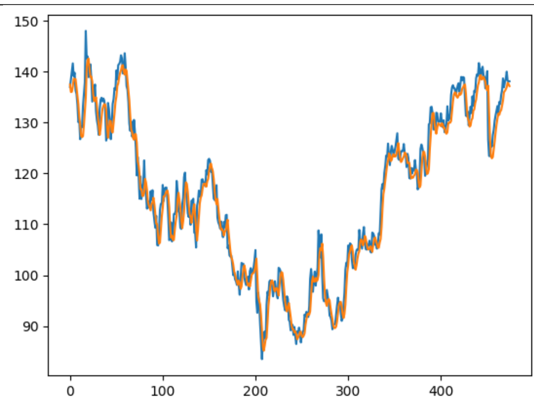
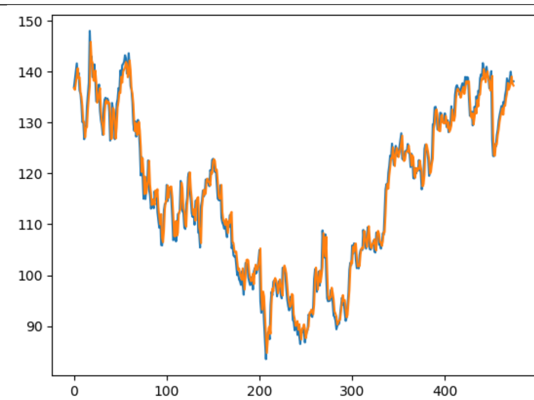
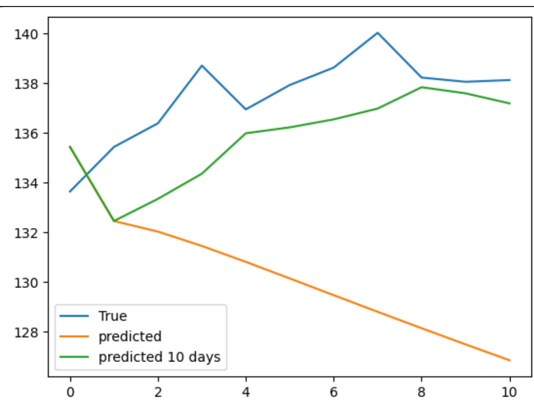
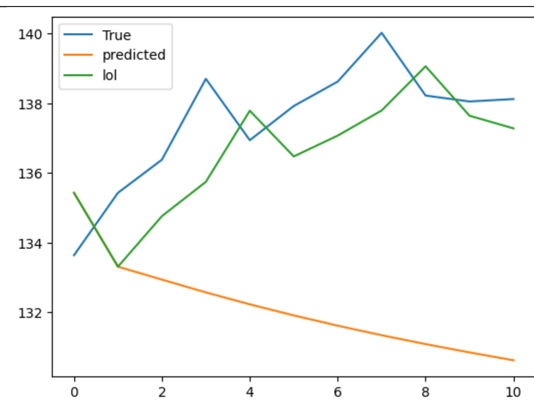
<table>
<tr><th>Type</th><th>LSTM</th><th>GRU</th></tr>
<tr><td>Training</td><td>
</td><td>
</td></tr>
<tr><td>Testing</td><td>
</td><td>
</td></tr>
<tr><td>Future Prediction</td><td>
</td><td>
</td></tr>
</table>

Table 1: Google with dividends

S&P

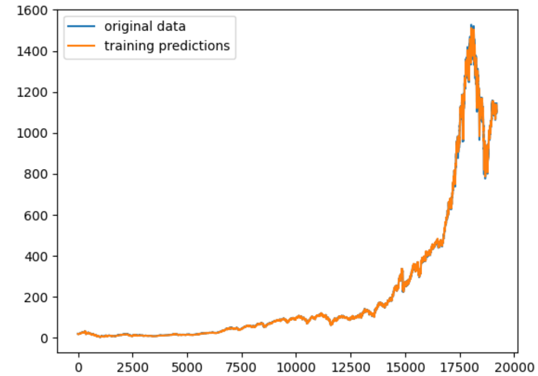
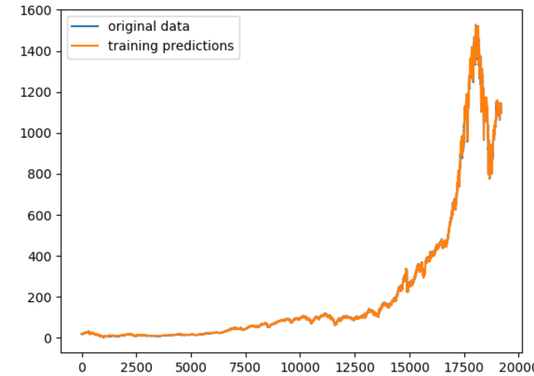
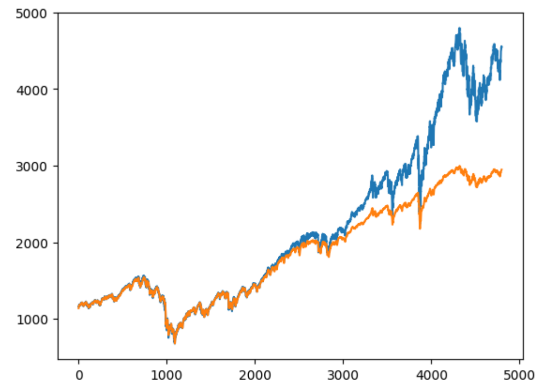
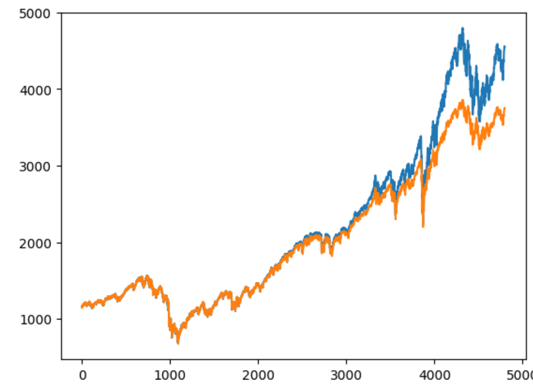
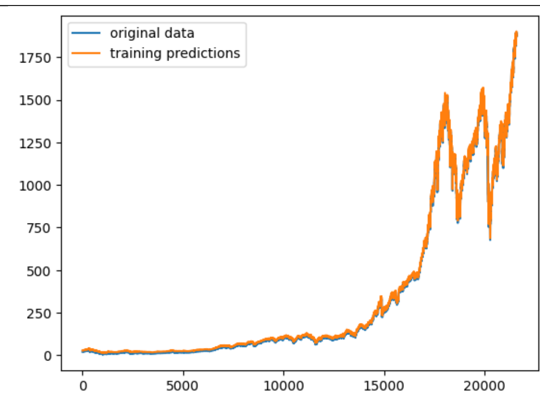
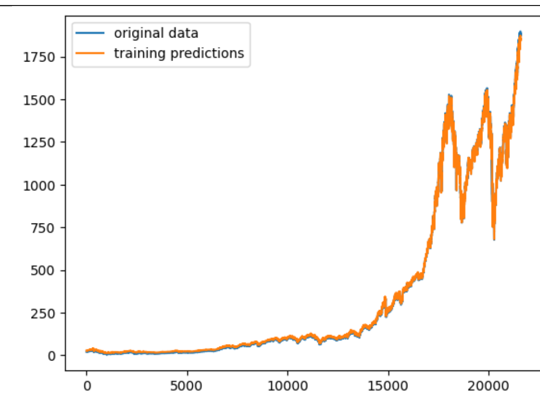
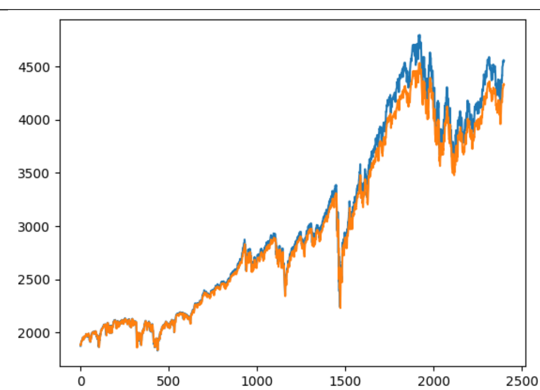
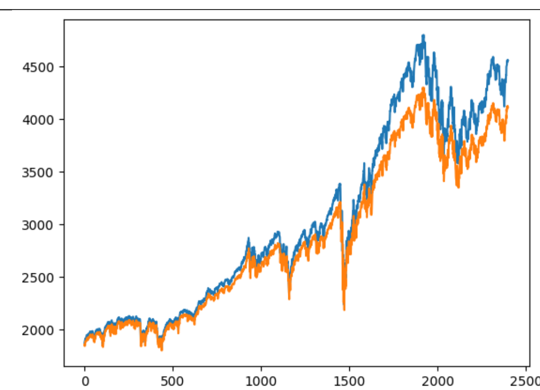
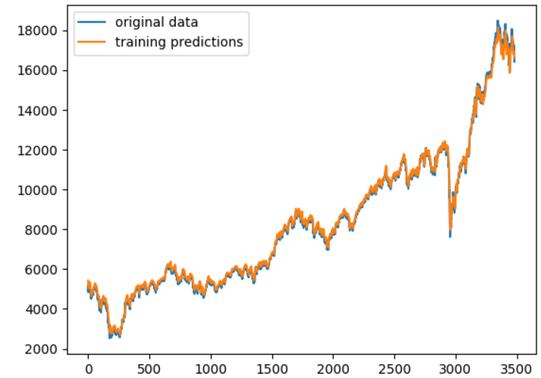
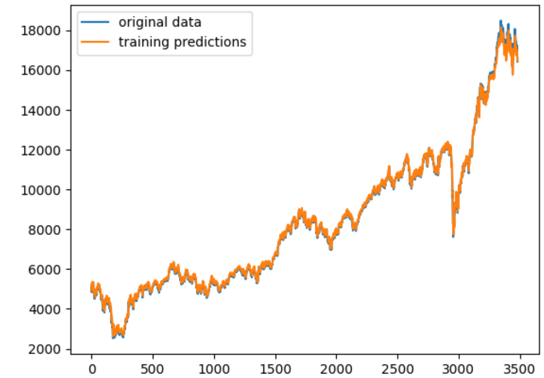
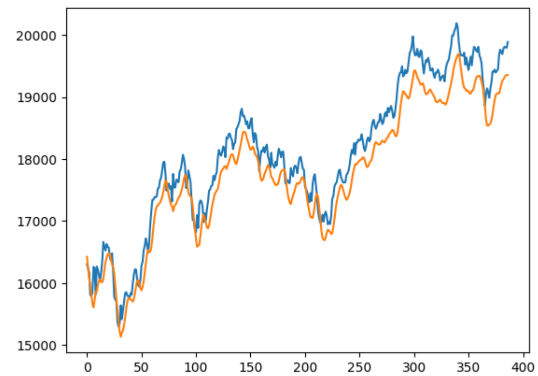
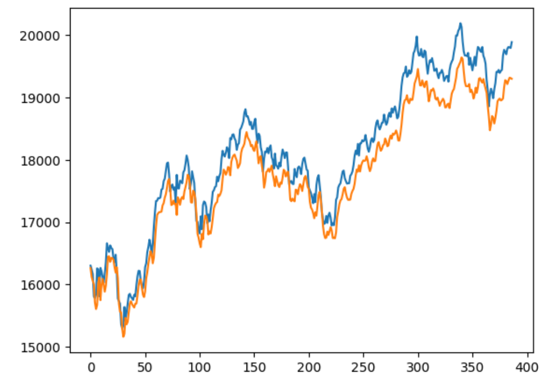
<table>
<tr><th>Without dividends</th><th>LSTM</th><th>GRU</th></tr>
<tr><td>Training</td><td>
</td><td>
</td></tr>
<tr><td>Testing</td><td>
</td><td>
</td></tr>
<tr><th>With dividends</th><th>LSTM</th><th>GRU</th></tr>
<tr><td>Training</td><td>
</td><td>
</td></tr>
<tr><td>Testing</td><td>
</td><td>
</td></tr>
</table>

Table 2: S&P with dividends and without dividends

NIFTY

<table>
<tr><th>NIFTY</th><th>LSTM</th><th>GRU</th></tr>
<tr><td>Training</td><td>
</td><td>
</td></tr>
<tr><td>Testing</td><td></td><td></td></tr>
</table>

APPLE

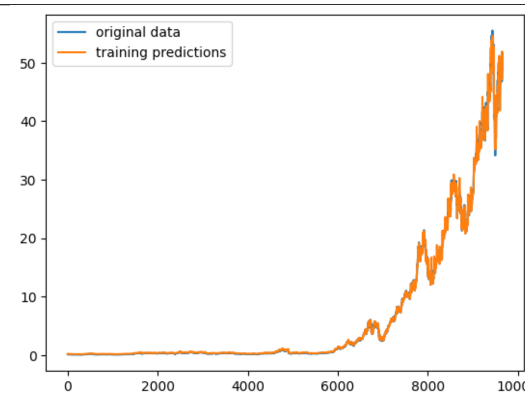
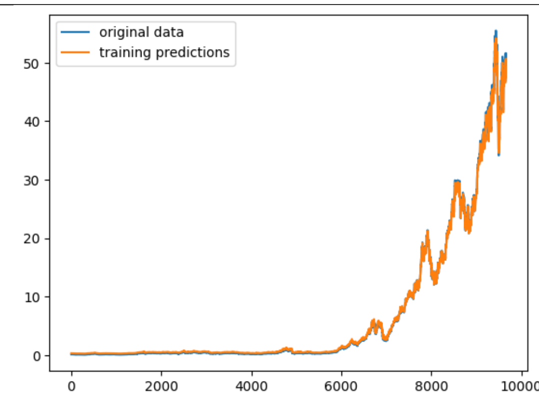
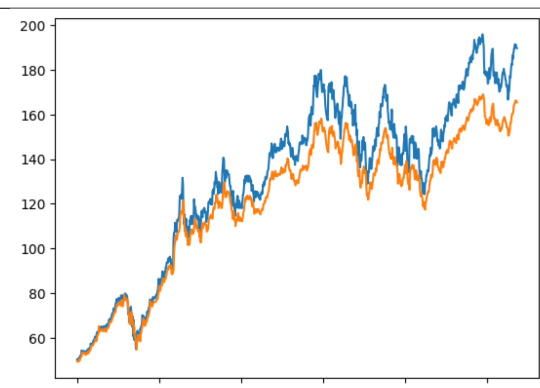
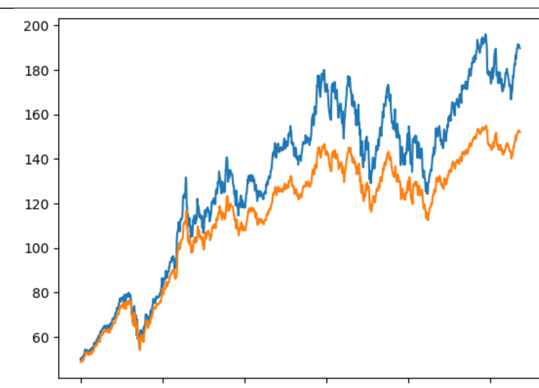
<table>
<tr><th>APPLE</th><th>LSTM</th><th>GRU</th></tr>
<tr><td>Training</td><td>
</td><td>
</td></tr>
<tr><td>Testing</td><td></td><td></td></tr>
</table>

Table 3: NIFTY and APPLE with dividends in feature set

Metrics

We used **MAE** (Mean Absolute Error) for its simplicity and easy interpretability. It calculates the average absolute differences between predicted and actual values, providing a straightforward measure of prediction accuracy

We employed **RMSE** (Root Mean Squared Error) to complement MAE by penalizing larger errors more significantly. This makes it suitable for capturing the impact of outliers.
For a percentage-based evaluation, we utilized **MAPE** (Mean Absolute Percentage Error), useful for interpreting errors relative to the scale of stock prices. It helps assess the accuracy of predictions while considering the proportion of errors in relation to actual values. The combination of MAE, RMSE, and MAPE allows for a comprehensive evaluation of stock prediction models, considering different aspects of accuracy and scale.
While comparing any 2 companies or 2 architectures it is apt to do so using MAPE
Here are the values of the aforementioned errors for some companies

Table 4: errors with LSTM and GRU architectures of S&P left-without dividends(left), with dividends(right)

Architecture	RMSE	MAE	MAPE
LSTM	568.434	309.256	0.08907
GRU	517.252	409.159	0.0281299

Architecture	RMSE	MAE	MAPE
LSTM	107.873	75.647	0.020700
GRU	268.172	140.936	0.040890

Table 5: error values with LSTM model and GRU model of GOOGLE

Architecture	RMSE	MAE	MAPE
LSTM	3.034	2.403	0.020845
GRU	2.564	1.969	0.017278

Table 6: error values with LSTM model and GRU model of NIFTY(left) and APPLE(right)

Architecture	RMSE	MAE	MAPE
LSTM	377.210	337.689	0.018450
GRU	367.210	332.679	0.018115

Architecture	RMSE	MAE	MAPE
LSTM	12.809	10.709	0.072204
GRU	20.022	16.994	0.11451

INFERENCES

- In some of the cases we can observe that GRU error is less than LSTM. That maybe because of the fact that GRU is designed to handle short-term dependencies more effectively compared to LSTM. If the relevant patterns for stock prediction are characterized by short-term dependencies, GRU may outperform LSTM
- If we have a look at Metrics from Table 4(MAPE)4 corresponding to plots from Table 22, our idea of including dividends in the feature set has been validated conclusively because there is lot of difference in error while testing.
- If we observe plots in third column of Table 11, when trying to predict for the next 1 day (green line) the graph imitates the True values(blue) closely, but when tried to predict the next 10 days(orange), it performs poorly. It is probably due to the following reasons
  1. **Error Accumulation:** Predicted values can introduce small errors that accumulate over time, impacting the model's accuracy as it propagates through subsequent predictions.
  2. **Lack of Ground Truth Feedback:** Using predicted values breaks the feedback loop from true values during training, hindering the model's ability to learn from its mistakes and refine its predictions.
  3. **Violation of Autoregressive Assumption:** Many time series models, including LSTMs, assume future values depend on past values. Using predicted values disrupts this assumption, leading to a potential loss of temporal patterns crucial for accurate predictions.

References

[1] Aishwarya Singh. Stock prices prediction using machine learning and deep learning, 2023.

[2] Jason C. Sullivan. Stock price volatility prediction with long shortterm memory neural networks, 2023.