MAJOR PROJECT

Reporting the approach, process and results of the deep learning project

Gohel Mruganshi Jatinbhai (B20CS014),

Ghelani Shubham Bhaveshbhai (B20EE019),

Vedasamhita Challapalli (B20CS078)

Deep Learning

Financial Time Series Forecasting

PROBLEM STATEMENT

The application of Deep Learning has acquired a significant amount of attention and interest leading to extensive research into novel methodologies. The project delves into the real, examining various techniques alongside the well-established LSTM methods to forecast future values in time series data. Furthermore, it explores innovative approaches such as encoding time series data as images and utilizing Convolution Neural Networks (CNNs), ResNets and LSTM for prediction. The premise lies in CNN's adeptness at learning from 2-dimensional image data potentially enhancing forecasting accuracy by capturing intricate spatial patterns, while retaining comprehensive information from the financial data.

SOLUTION STRATEGY

Following models (or sequential combination of models) have been implemented to compare the results and draw conclusions:

1. Using Simple LSTM on the time series data
2. Using CNN on the generated GADF images
3. Using LSTM on the generated GADF images (LSTM Image Model)
4. Using ResNet-18 to encode images and passing embeddings to LSTM With Encoded Input

The above steps of strategy have been further explained in detail in the section: Approach/Major innovations.

DATASET

The project utilizes the hourly AAPL stock price data from Yahoo Finance (Uploaded on Kaggle). This dataset is a stock prediction on Apple Inc. The start date is taken as 2022-05-11 and the end date as 2024-04-10 . It was collected from Yahoo Finance. The dataset contains a csv file named AAPL.csv which has the opening, high, low and closing prices for each hour.

[LINK TO DATASET]

The pandas dataframe of the dataset:

```
data
```

```
[*********************100%%**********************]  1 of 1 completed
```

Datetime	Open	High	Low	Close	Adj Close	Volume
2023-04-11 09:30:00-04:00	162.350006	162.360001	160.559998	161.149994	161.149994	11628279
2023-04-11 10:30:00-04:00	161.160004	161.199997	160.589996	160.909897	160.909897	6083197
2023-04-11 11:30:00-04:00	160.899994	161.230103	160.509995	161.119904	161.119904	4322884
2023-04-11 12:30:00-04:00	161.119995	161.520004	160.839996	161.411697	161.411697	4141413
2023-04-11 13:30:00-04:00	161.415604	161.830002	161.309998	161.710693	161.710693	4299346

APPROACH/MAJOR INNOVATIONS

We have used several approaches and experimented on them in order to find the best approach and architecture to produce accurate results.

1) **Implementing a simple LSTM:** The initial approach involved implementing a simple model of LSTM on the time series data which showed decent fitting of data but failed near some of the troughs and was not very smooth
2) **GADF Image Representations and CNN-based Prediction:** The approach

involved generating GADF images from the time series data and training a CNN model to predict the 'close price' of the stock directly. However, this method did not yield satisfactory results, as CNNs alone were unable to capture the sequential properties of time series data effectively.

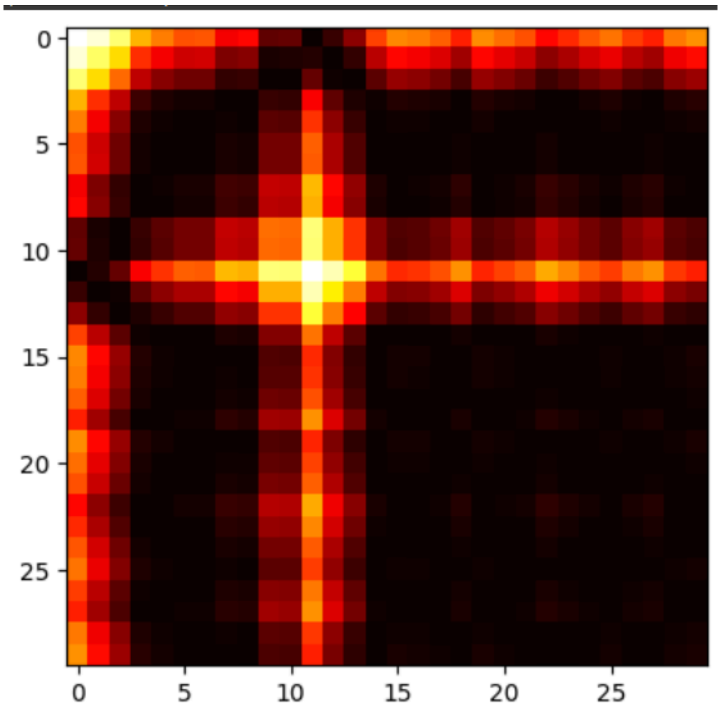
3) **LSTM -based Prediction with GADF Image Sequences:** Building upon the limitations of the first approach, the next innovation was to utilize a sequence of GADF images as input to a recurrent neural network (LSTM) model, with the 'close_price' as the target variable. This approach demonstrated more promising results, as LSTMs are well-suited for modeling sequential data, and the GADF images provided a suitable representation of the temporal patterns present in the time series.
4) **CNN Feature Extraction and LSTM Forecasting:** One of the key innovations was the development of a two-stage model, where a pre-trained CNN, such as ResNet-50, was employed as a feature extractor to obtain embeddings from the GADF-encoded images. These embeddings were then fed into a Long Short-Term Memory (LSTM) network, which leveraged the sequential nature of the data to make predictions of future stock prices using a sliding window approach.

The aim of our project is to devise new approach/architecture that can lead to better accuracy on prediction for time series data. We used GADF images instead of time series data columns directly and then feed it to our Neural Networks such as LSTM and ResNet. This improved our accuracy significantly. Also, using the embeddings obtained from ResNet and then feeding them to LSTM gave the best results.

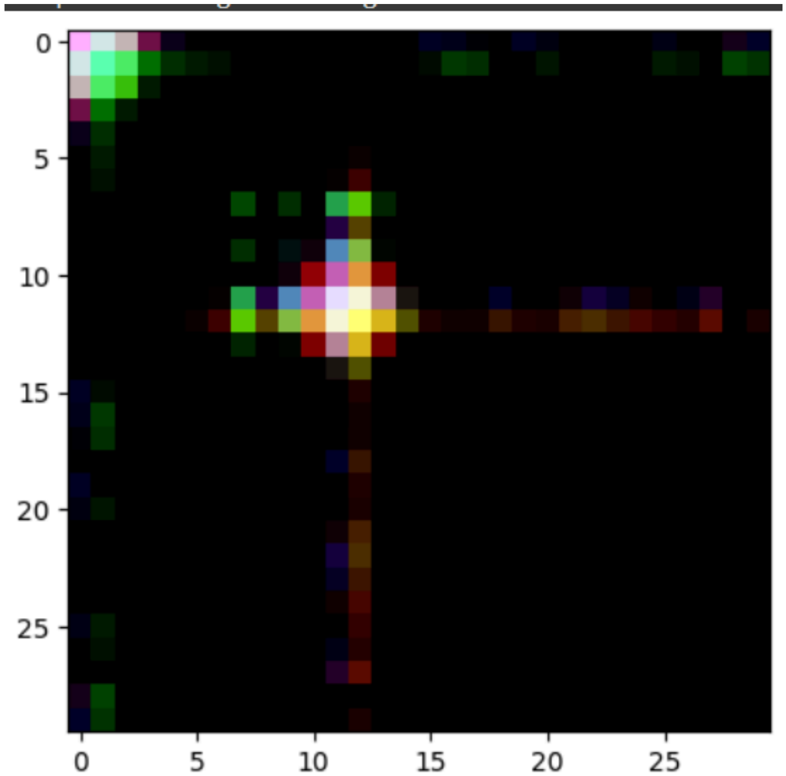
RESULTS

The entire time series data was first converted into single channel images of size 32*32 and number of images 3500 using Gramian Angular Fields (summation mode). The dataset was also divided into regions of 80% and 20% accuracy.

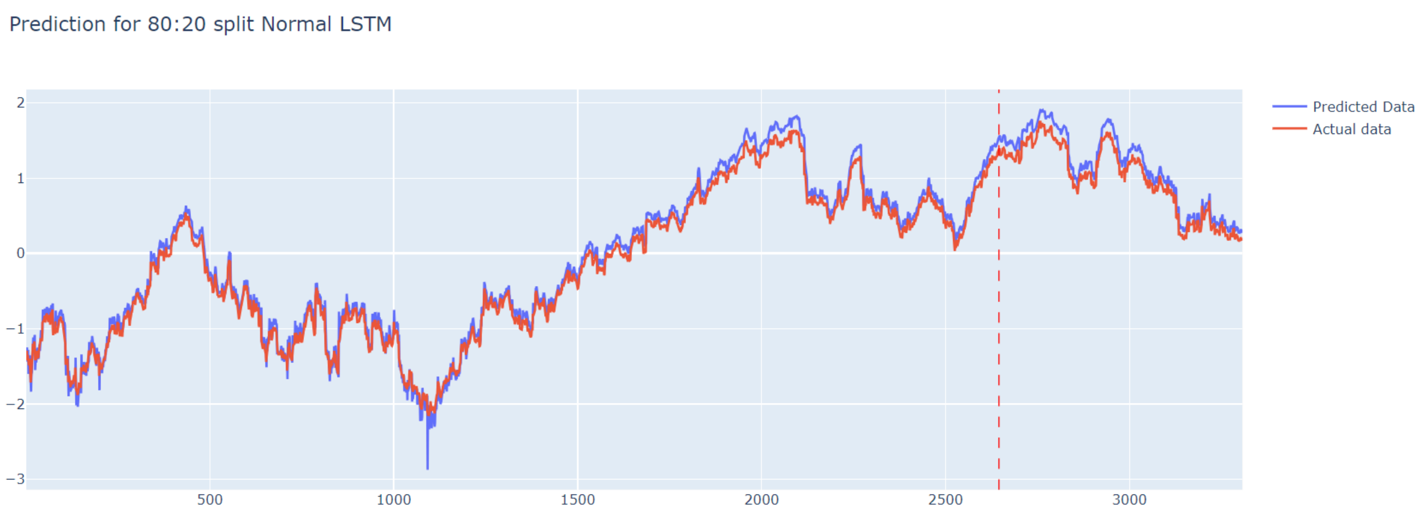
- The generated GADF image for the column “close”:



- The generated first GADF image for all the columns:



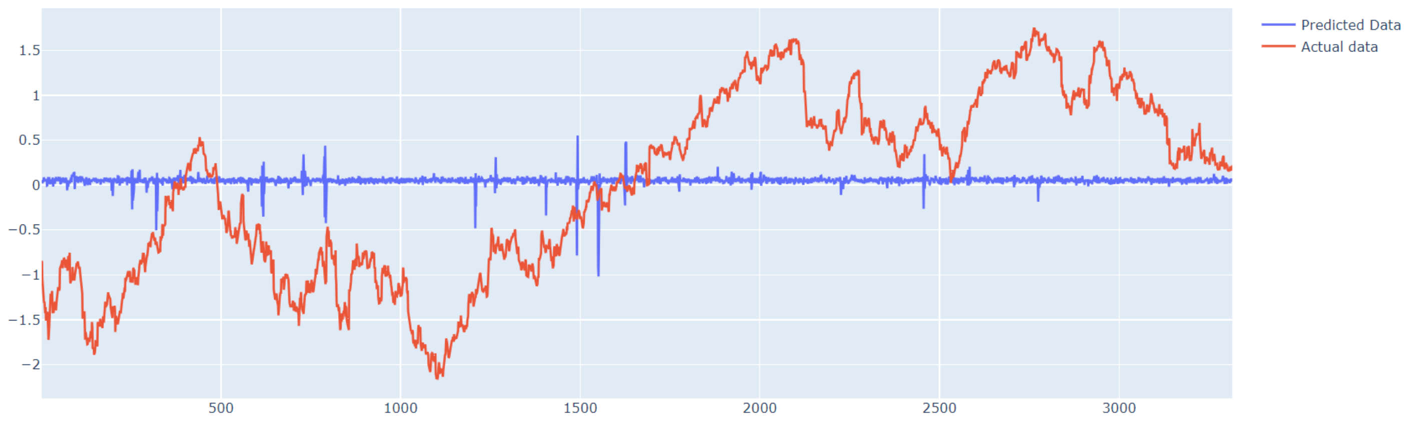
1. **Using Simple LSTM on the time series data**:



Even though simple LSTMs seem to work fine, As we see near the dashed line, near troughs, the predicted seem to deviate from the actual.

2. **Using CNN on the GADF generated images:**

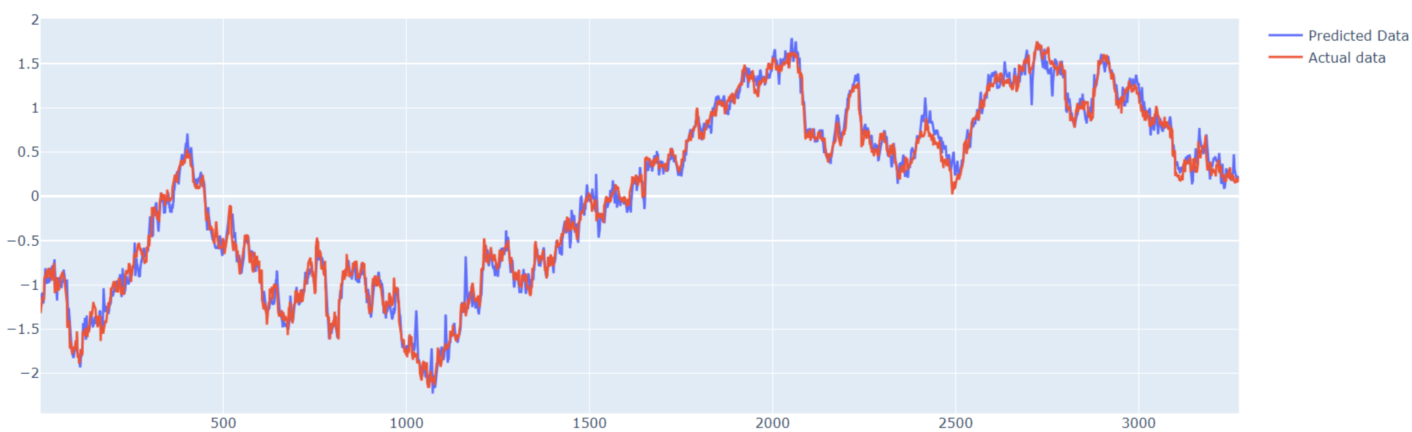
Prediction for 80:20 split Using CNN



CNNs are performing very poorly compared to LSTMs, so simple CNNs are discarded for further process. The reason for this is theLSTM can then focus on modeling the sequential dependencies and long-term relationships in the extracted features, which is a task that CNNs alone struggle with for time series data.

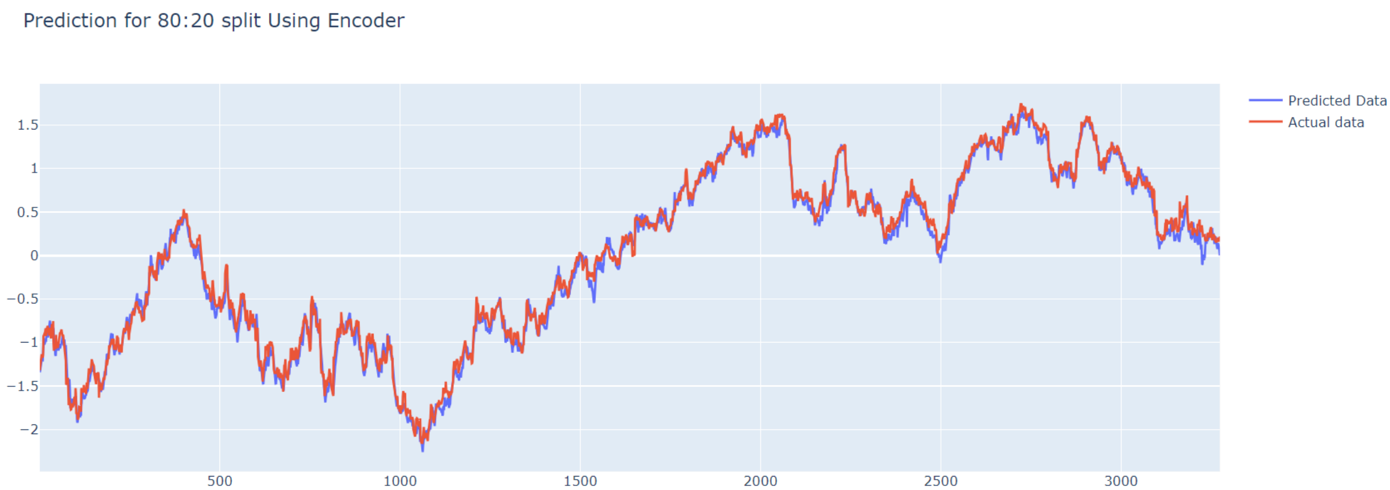
3. **Using LSTM on the GADF generated images (LSTM Image Model)**

Prediction for 80:20 split



LSTMs on the GADF generated images fit better than simple LSTMs, hence we can observe that converting time series data into images using GADF and then fitting LSTMs is the most optimal approach till now.

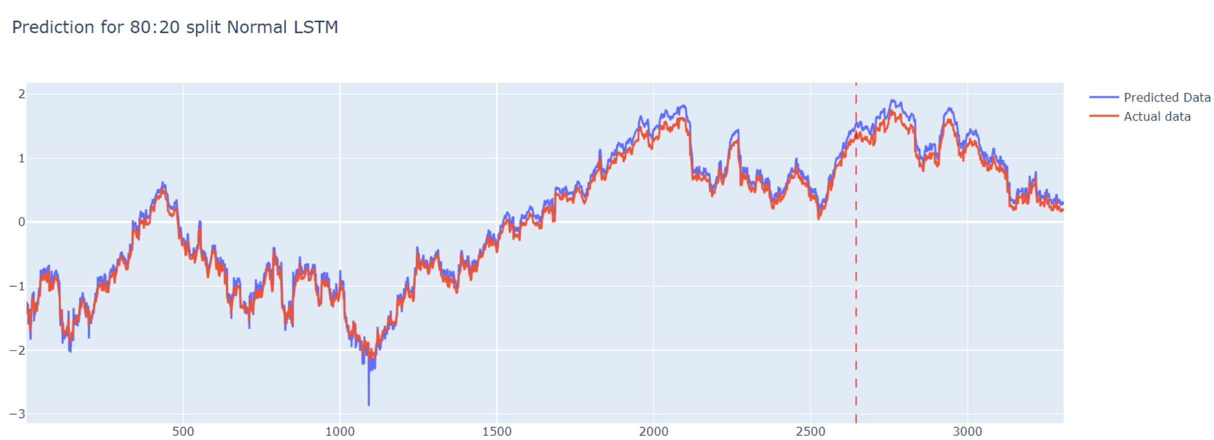
4. **Using ResNet-18 to encode images and passing embeddings to LSTM With Encoded Input**



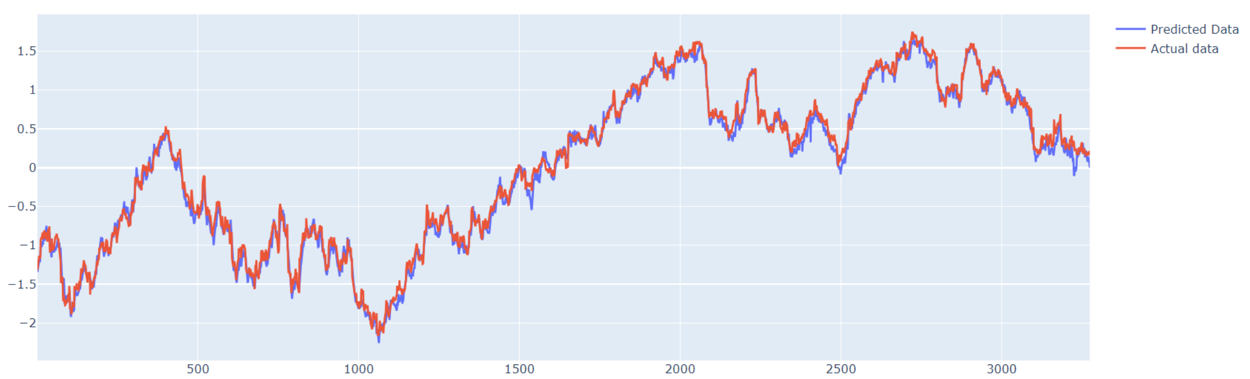
As we can see, using a ResNet to encode the images and then passing these embeddings to LSTM works the best.

COMPARISONS OF THE BEST MODEL WITH OTHERS

1. Simple LSTM on time series data vs LSTM on encoded GADF images:

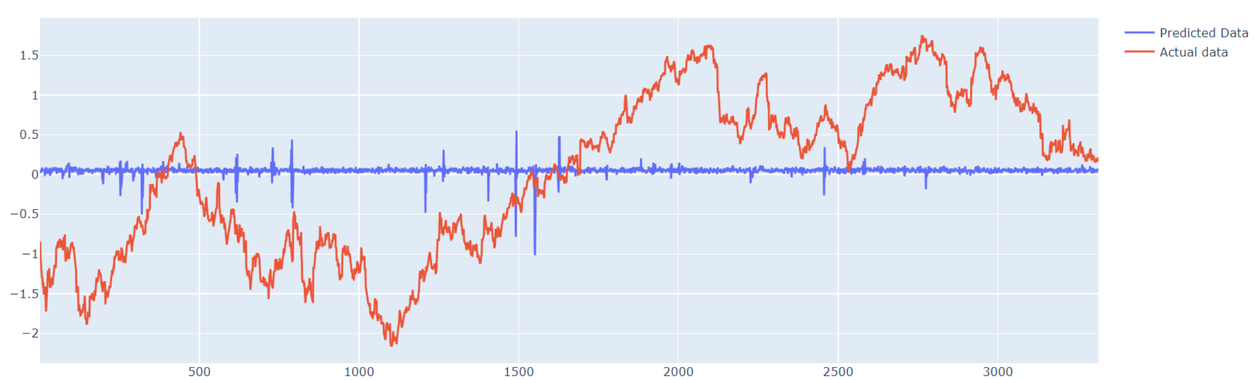


Prediction for 80:20 split Using Encoder

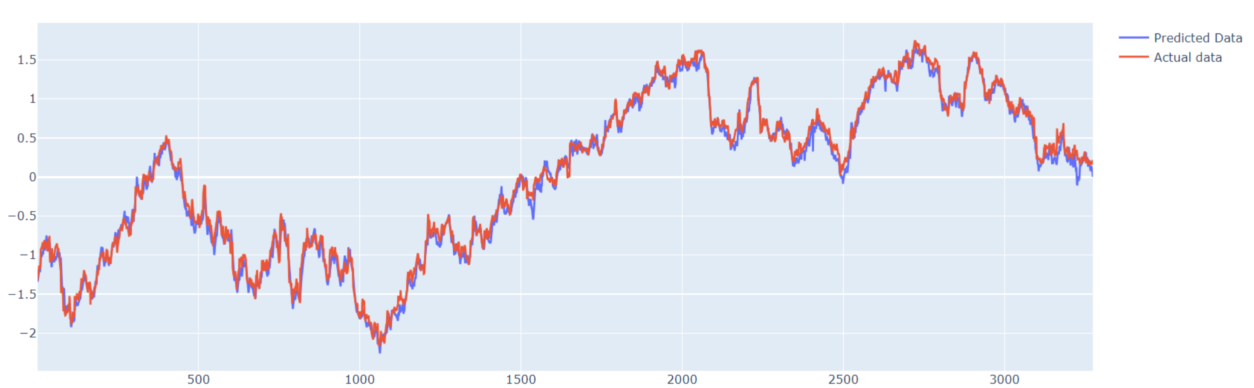


2. CNN on GADF images vs LSTM on encoded GADF images:

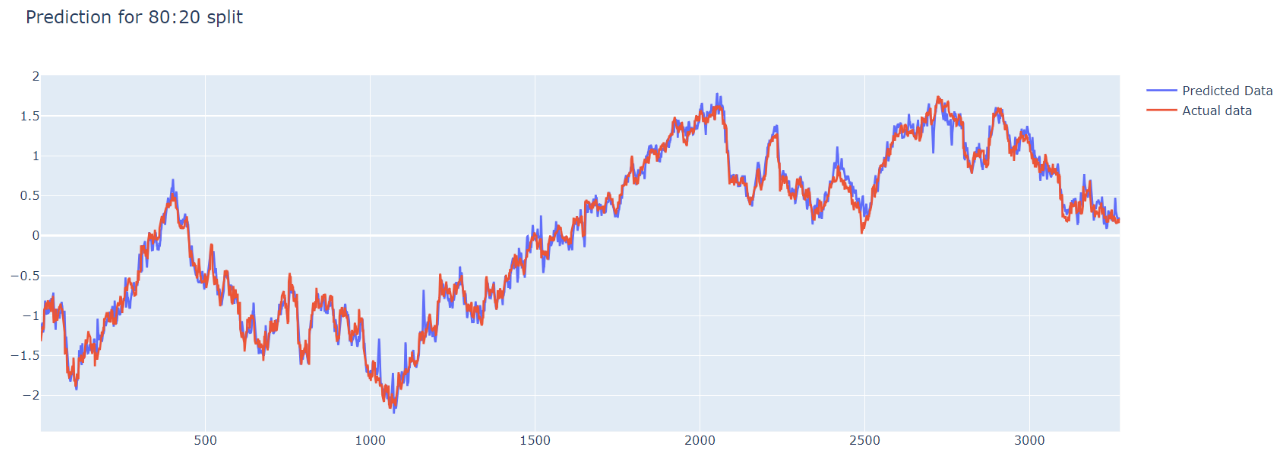
Prediction for 80:20 split Using CNN

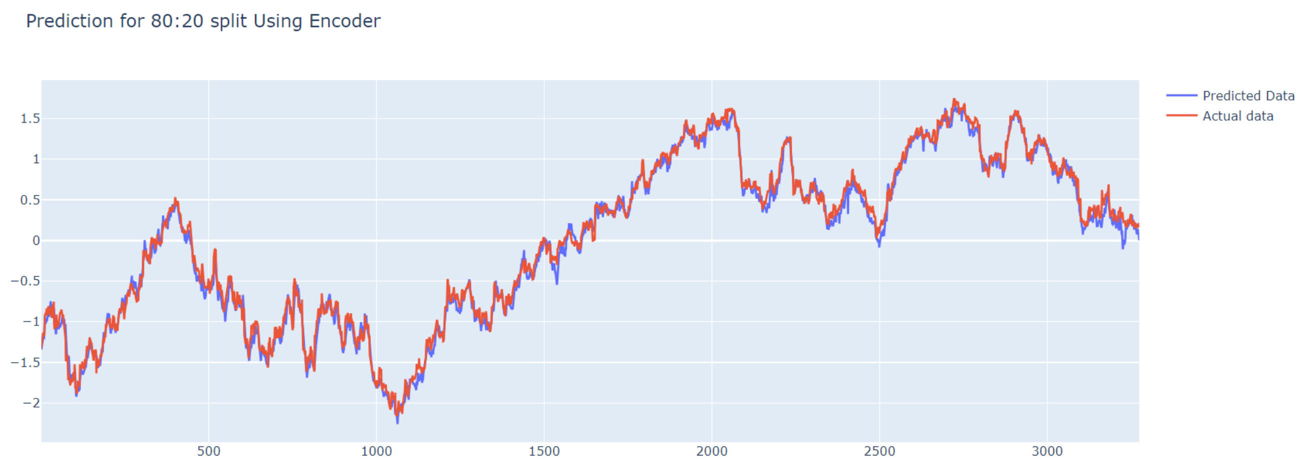


Prediction for 80:20 split Using Encoder



3. LSTM on GADF images vs LSTM on encoded GADF images:





As we can see LSTM on encoded (using ResNet) GADF images has the best fit graphs.

CONCLUSION AND FUTURE SCOPE

We have developed approaches which are better than LSTM and can be used if far more accurate time series prediction is needed. Eg: High Frequency Trading, Medical Time Series Prediction. The project explores deep learning approaches for stock price prediction by combining GADF image representations, pre-trained CNNs for feature extraction, and LSTM networks for sequential modeling. Through extensive experimentation, the approach of using a pre-trained ResNet as a feature extractor, followed by an LSTM for forecasting, yields the most promising results, effectively capturing both spatial patterns in GADF images and temporal dependencies in time series data. Future Scope of this project is to improve the performance of this implementation, as it has potential for better time-series specific image generation

REFERENCES

1. Barra, S., Carta, S., Corriga, A., Podda, A. S., & Recupero, D. R. (2020, May 1). *Deep learning and time series-to-image encoding for financial forecasting*. IEEE/CAA Journal of Automatica Sinica. https://doi.org/10.1109/jas.2020.1003132
2. Sezer, M. B., Gudelek, M. U., & Özbayoğlu, A. M. (2020, May 1). *Financial time series forecasting with deep learning : A systematic literature review: 2005–2019*. Applied Soft Computing (Print). https://doi.org/10.1016/j.asoc.2020.106181
3. M. K. (2021, April 2). *Leveraging Computer Vision to Encode Time-Series Data as Images for Visual Recognition Algorithms*. YouTube. https://www.youtube.com/watch?v=cWzo29wTCII

LINK TO COLAB

The zip file contains the Report and the Colab file, the Colab file can be independently run as an individual file.