Foreign Exchange Prediction

Presented By:

Nikhil Naik

Content

- Introduction
- Problem Statement
- Data Description
- Data Preprocessing
- Exploratory Data Analytics
- DL models
- DL analysis
- Conclusion

Introduction

- Foreign exchange markets are dynamic and challenging. Predicting foreign exchange rates has been a arduous task for researchers.
- Innovation in Deep Learning Technologies has proven valuable in enhancing forecasting capabilities. In this project, we attempt to use these technologies to predict Foreign Exchange Rate of Indian Rupees
- Economic indicators (GDP, Interest Rate, Inflation Rate) integrated for a comprehensive view in the training process.
- Global perspective considered, leading to a generalized model for predicting Forex rates for many countries.
- Transfer learning techniques employed for fine-tuning pretrained models, ensuring adaptability to diverse economic scenarios.
- India chosen as the target country, showcasing the model's applicability and effectiveness in a significant global economic context.

Objective

- **Develop Deep Learning Models:** Create and implement RNN and LSTM models that harness historical foreign exchange data to predict future currency exchange rates accurately.
- **Enhance Prediction Accuracy:** Explore techniques to improve the precision of foreign exchange rate forecasts using advanced DL techniques such as Transfer Learning and Hyper-Parameter Tuning.
- **Provide Insightful Analysis:** Gain insights into market trends and patterns to aid investors, financial institutions, and businesses in making informed decisions within the global currency exchange landscape.
- **Compare Transfer Learning Methodologies**: Compare different type of training a Pre-trained model for further fine tuning. Seeing which type gives best results.

Tools/Technologies used

- **Platform for Python Code:** Utilized Google Colab as the coding environment to execute Python scripts.
- **GPU for Model Training:** Employed the GTX 1650 GPU to facilitate the training process for our models, optimizing computational performance.
- **Pandas Library Usage:** Leveraged functions like ‘merge_asof’ and interpolation within the Pandas library to manage data merging and interpolation tasks effectively.
- **TensorFlow/Keras Implementation:** Utilized the Sequential method along with layers like LSTM, RNN, and Dense within TensorFlow/Keras for model architecture and development.
- **Matplotlib for Data Visualization:** Utilized Matplotlib extensively for data visualization, employing its graphical capabilities to present insights and results effectively.
- **Datetime Standardization:** Incorporated the datetime library to standardize datetime formats, ensuring consistency and uniformity across the dataset.
- **Seaborn for Visual Summaries:** Utilized Seaborn to create informative boxplots, visually summarizing the variability of dataset values, aiding in data exploration.
- **Scikit-learn for Data Scaling and Metrics:** Employed Scikit-learn to scale the data using MinMaxScaler and access various metrics available in the library, facilitating data preprocessing and evaluation of model performance.
- **Hyper-Parameter Tuning:** Used Keras_tuner library for hyper-parameter tuning.

Exploratory Data Analysis

- Data Collection
- Data Pre-Processing
- Data Cleaning & Formatting
- Data Visualization

Data Description

Forex Data

- Forex data is used from the Federal Reserves here and the dataset consists of daily rates from 1 Jan 1996 to 31 Oct 2023.

Consumer Price Index (CPI)

- It measures change over time in the prices paid by consumers for goods and services.
- We've used data from Government Website here. The dataset consists of daily rates from 1 Jan 1996 to 31 Oct 2023.

Interest Rates

- Interest rates increases the demand for and value of the home country's currency.
- The dataset was used from Government Website here. The dataset consists of daily rates from 1 Jan 1996 to 31 Oct 2023.

Normalized GDP

- Normalized GDP is used because it removes the distortions caused by population growth, and inflation.
- The dataset is open-sourced and available Government Website here. The dataset consists of daily rated from 1 Jan 1996 to 31 Oct 2023.

*All the above datasets have been taken for the following countries: Australia, India, Russia, China, South Africa, Poland, and USA

Data Preprocessing

- **Diverse Date Formatting Across Countries:** The datasets contained varying date formats based on different country standards, leading to inconsistencies in the representation of dates.
- **Presence of Missing Values:** Within the datasets, there were instances of missing data, which could affect the completeness and accuracy of the information.
- **Integration of Four Datasets:** The datasets were consolidated and merged, creating a unified and structured dataset. This merging process aimed to organize the information cohesively, potentially addressing inconsistencies and facilitating a more systematic analysis
- **Pre-Processing Techniques:**
  1. **Interpolation:** Implemented interpolation methods to estimate missing values, enhancing dataset completeness while preserving patterns within the data.
  2. **Join Method:** Utilized the merge_asof function from the pandas library to robustly merge datasets, ensuring all records from each dataset were included in the merged result, maintaining data integrity.

Data Preprocessing

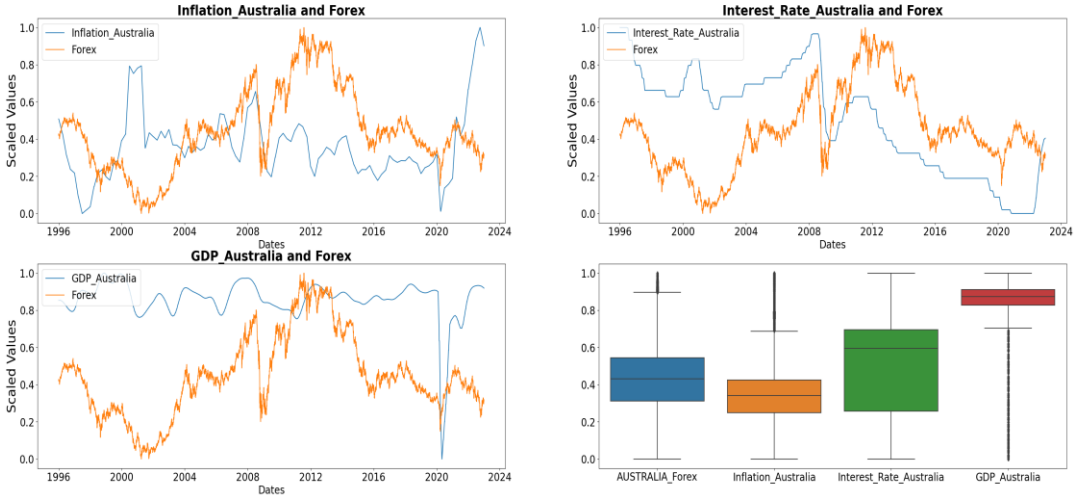
The resulting dataset after all processing steps is as follows:

Series Description	AUSTRALIA_Forex	BRAZIL_Forex	CHINA_Forex	INDIA_Forex	SOUTHAFRICA_Forex	POLAND_Forex	USA_Forex	Inflation_India	Inflation_USA	Infl
1996-01-02	0.7433	0.97230	8.3380	35.209999	3.642500	2.4705	1.0	8.996540	0.586319	
1996-01-03	0.7485	0.97300	8.3391	35.220001	3.630000	2.4770	1.0	8.996540	0.586319	
1996-01-04	0.7453	0.97220	8.3403	35.320000	3.632000	2.4940	1.0	8.981523	0.576598	
1996-01-05	0.7475	0.97210	8.3404	35.439999	3.628000	2.4813	1.0	8.966505	0.566876	
1996-01-08	0.7473	0.97210	8.3415	35.849998	3.634500	2.4832	1.0	8.921453	0.537711	
...	...	...	...	...	...	...	...	...	...	
2022-12-26	0.6734	5.24345	6.9670	82.875000	17.220375	4.3626	1.0	6.061835	0.641458	
2022-12-27	0.6742	5.27900	6.9600	82.879997	17.302999	4.3974	1.0	6.085145	0.680978	
2022-12-28	0.6752	5.26750	6.9774	82.739998	17.115000	4.4144	1.0	6.108456	0.720497	
2022-12-29	0.6779	5.25470	6.9625	82.830002	16.889999	4.3833	1.0	6.131766	0.760017	
2022-12-30	0.6805	5.28600	6.8972	82.720001	16.995001	4.3775	1.0	6.155076	0.799536	

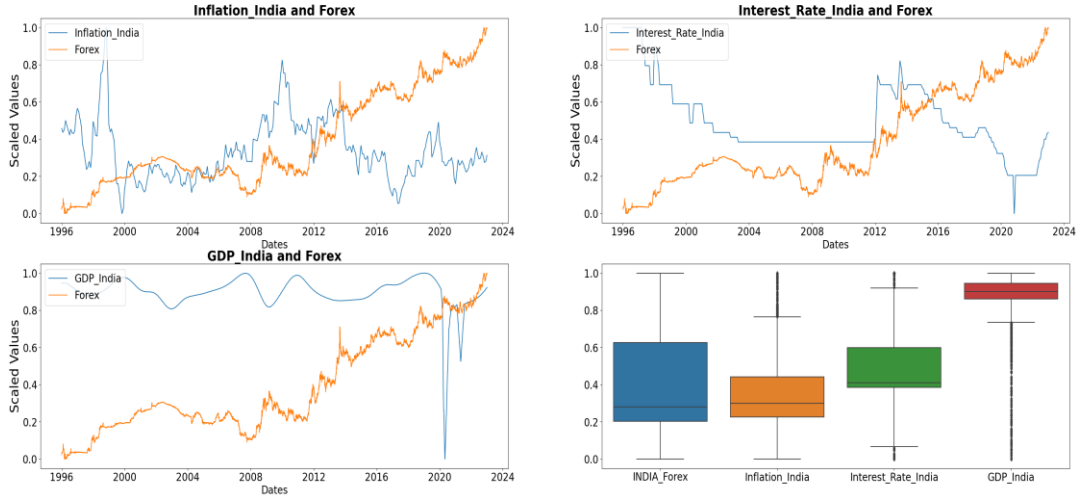
7044 rows × 28 columns

Data Visualization

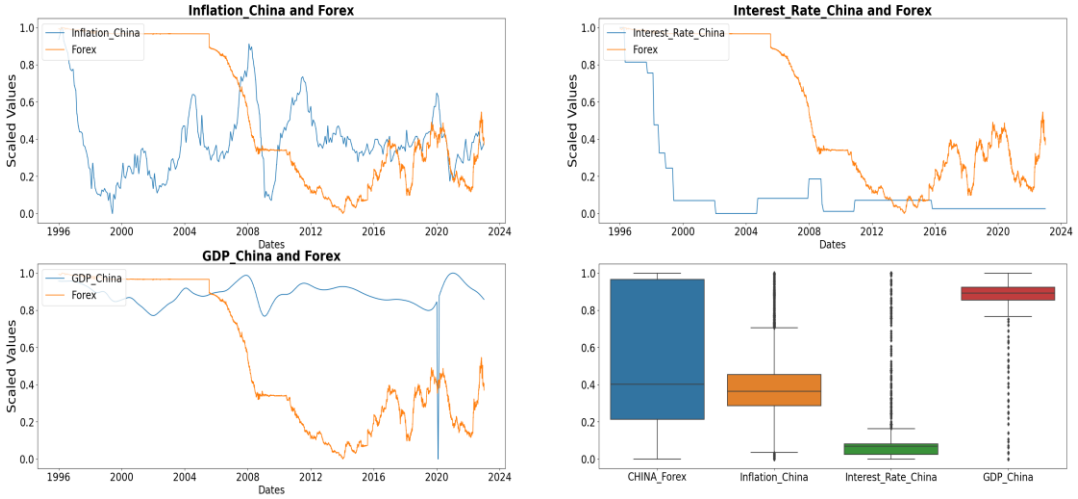
Plots for Australia



Plots for India



Plots for China



Insights

- No apparent linear correlation between the countries features and their Foreign Exchange Rate.
- Australia and China showing the best and the worst correlations visually.
- Box plot reveals there are many outliers in the data indicating huge fluctuations in feature values.

Deep Learning Technologies

- Model Pipeline
- Model Creation & Training
- Hyper-Parameter Tuning Details
- Model Performance Evaluations

ML Models – Pipeline

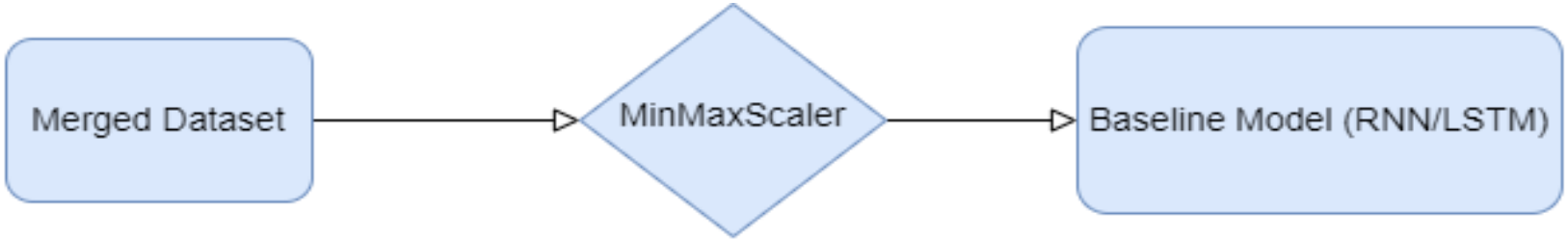


Fig 1: Baseline Model

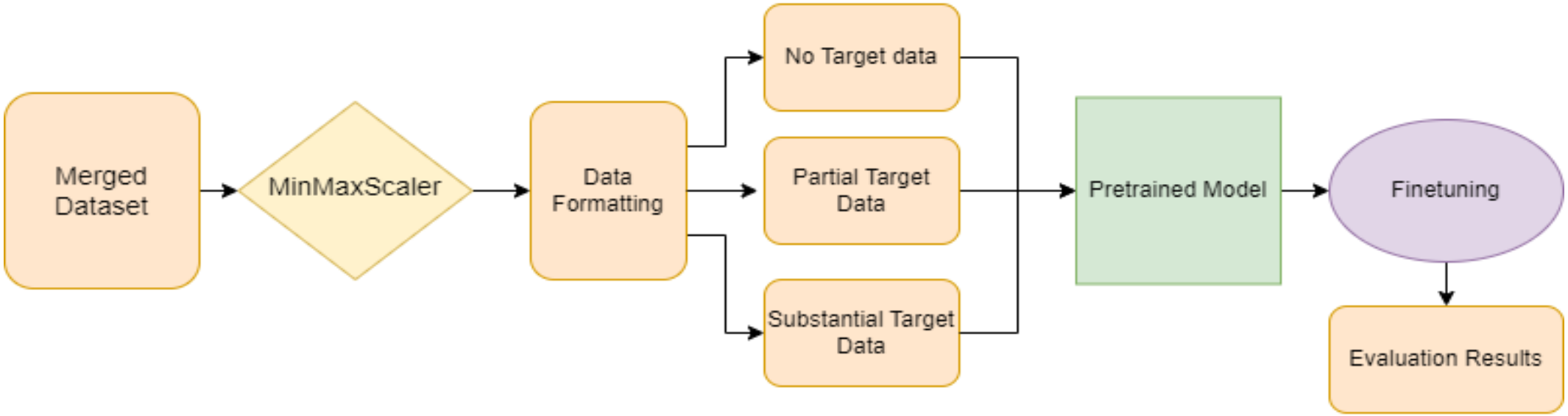
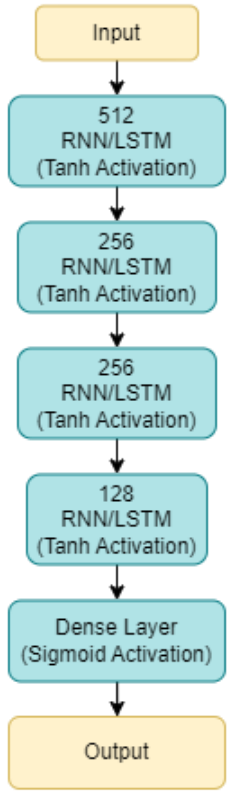


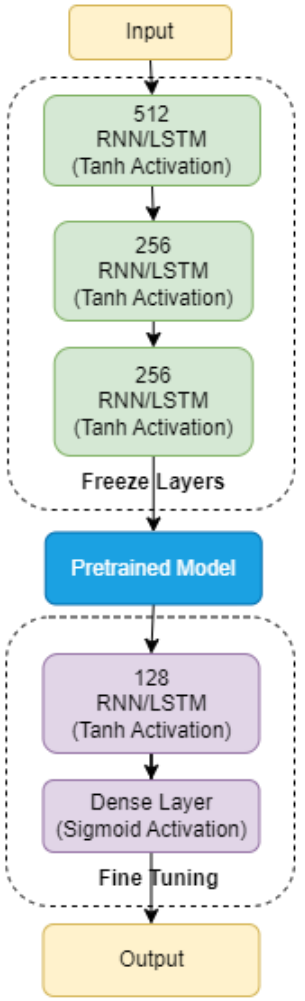
Fig 2: Final Model

Model Creation & Training

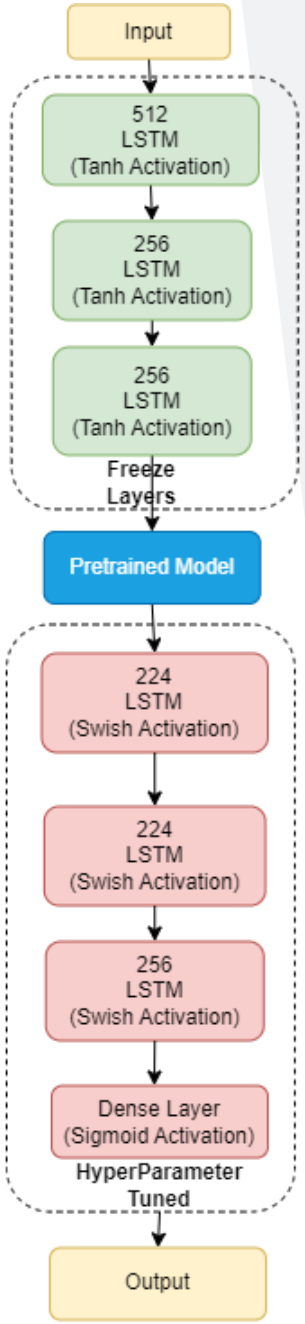
- For all models 30 days sequential data was considered and target was considered as 31st days Forex value
- Baseline model was created by using features from all countries as training features and India's forex data as target feature
- Baseline Model
  - Target = India's Forex
  - Features = All columns of all countries
- Transfer Learning Model
  - 3 Types of Data formats
  - Features = Each countries details
  - Target = Each countries own forex
  - Fine Tuning = India's Data
- Hyper-Parameter Tuned Model
  - Best pretrained model of Transfer Learning
  - Fine Tuning done with Random Search Keras Tuner.



Baseline Model
Optimizer= SGD



Transfer Learning Model
Optimizer = SGD



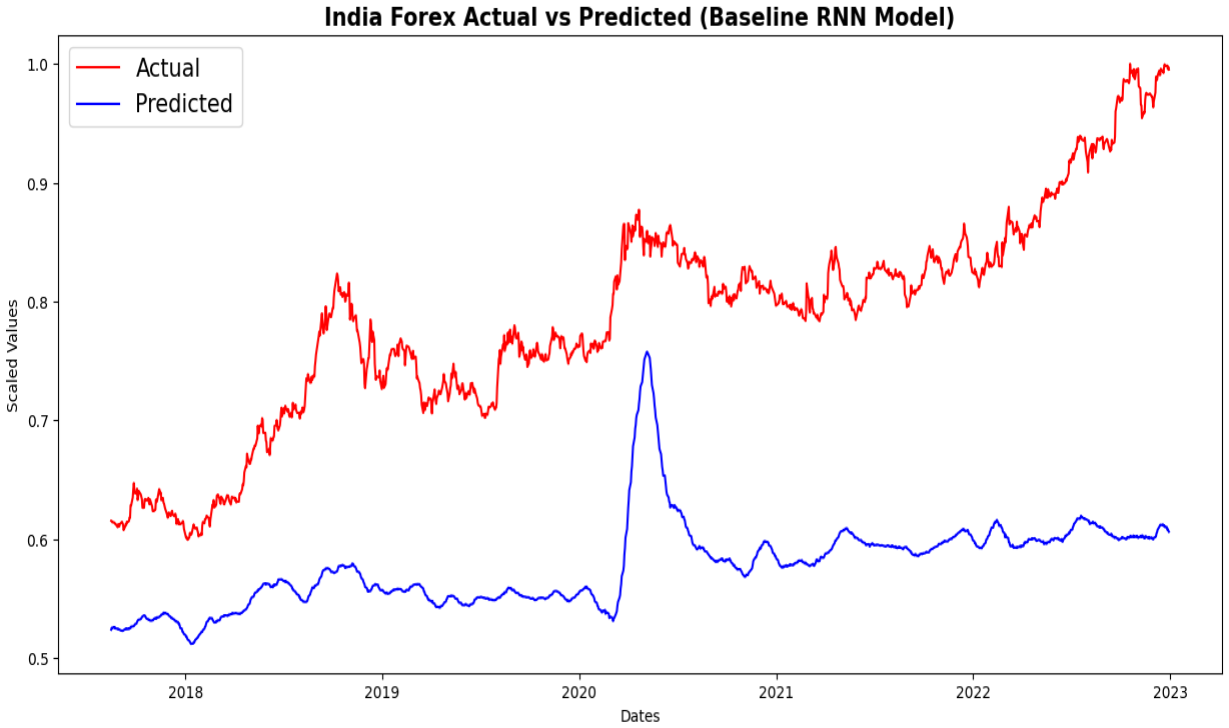
Pretrained Model
Optimizer = Adam

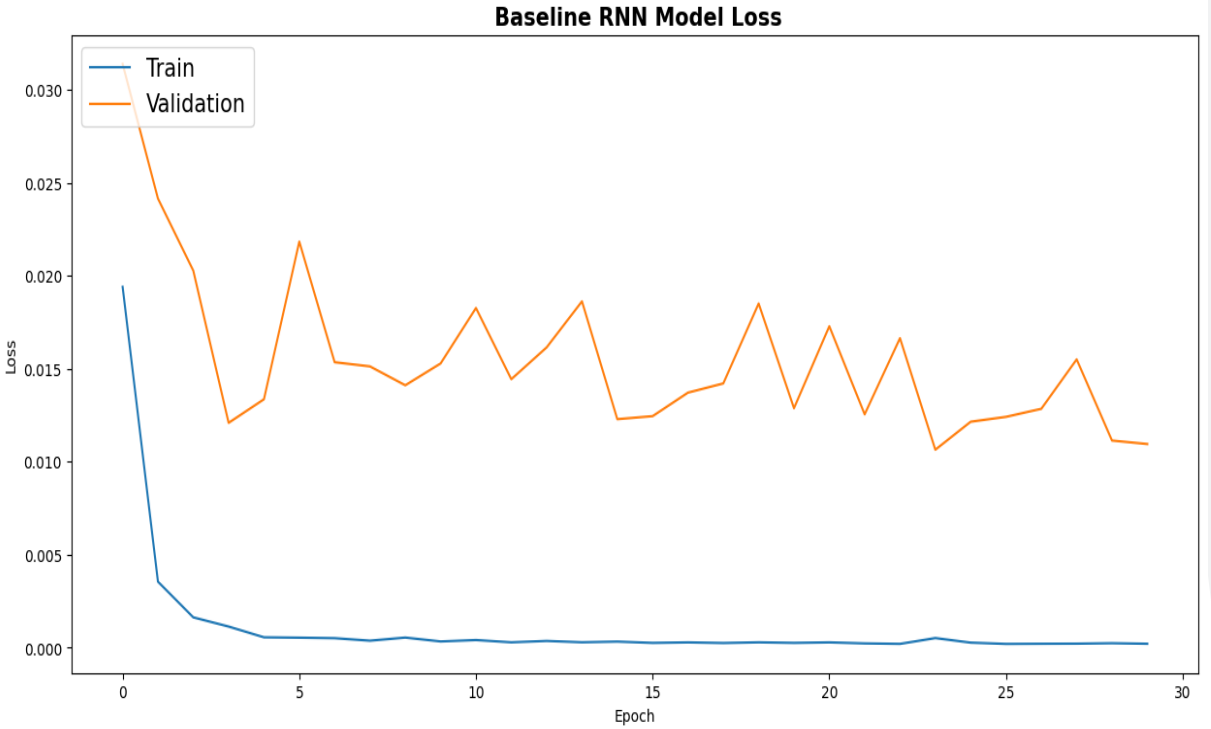
Hyper Parameter Tuning Details:

- The best pretrained model was selected for Hyper-Parameter Tuning
- Parameters for the Fine-Tuning Layers were selected using Random Search Tuner from Keras_Tuner Library
- Parameter selected using HP Tuning and their options:
  - LSTM Layers Activation = ['relu', 'sigmoid', **'swish'**]
  - Total LSTM Layers = [0,1,2,**3**]
  - Dense Layer Activation = ['relu', **'sigmoid']**
  - Optimizer = ['sgd', 'rmsprop', '**adam**', 'Adadelta']
- 25 max trials were used with 2 execution per trial
- Early stopping was also used to improve computation

```
For Layer 1
Layer =  lstm   Is Trainable =  False   Activation =  tanh

For Layer 2
Layer =  lstm_1         Is Trainable =  False   Activation =  tanh

For Layer 3
Layer =  lstm_2         Is Trainable =  False   Activation =  tanh

For Layer 4
Layer =  lstm_0_layer   Is Trainable =  True    Activation =  swish

For Layer 5
Layer =  lstm_1_layer   Is Trainable =  True    Activation =  swish

For Layer 6
Layer =  lstm_last      Is Trainable =  True    Activation =  swish

For Layer 7
Layer =  dense  Is Trainable =  True    Activation =  sigmoid
```

Model Evaluation: Baseline Model (RNN)

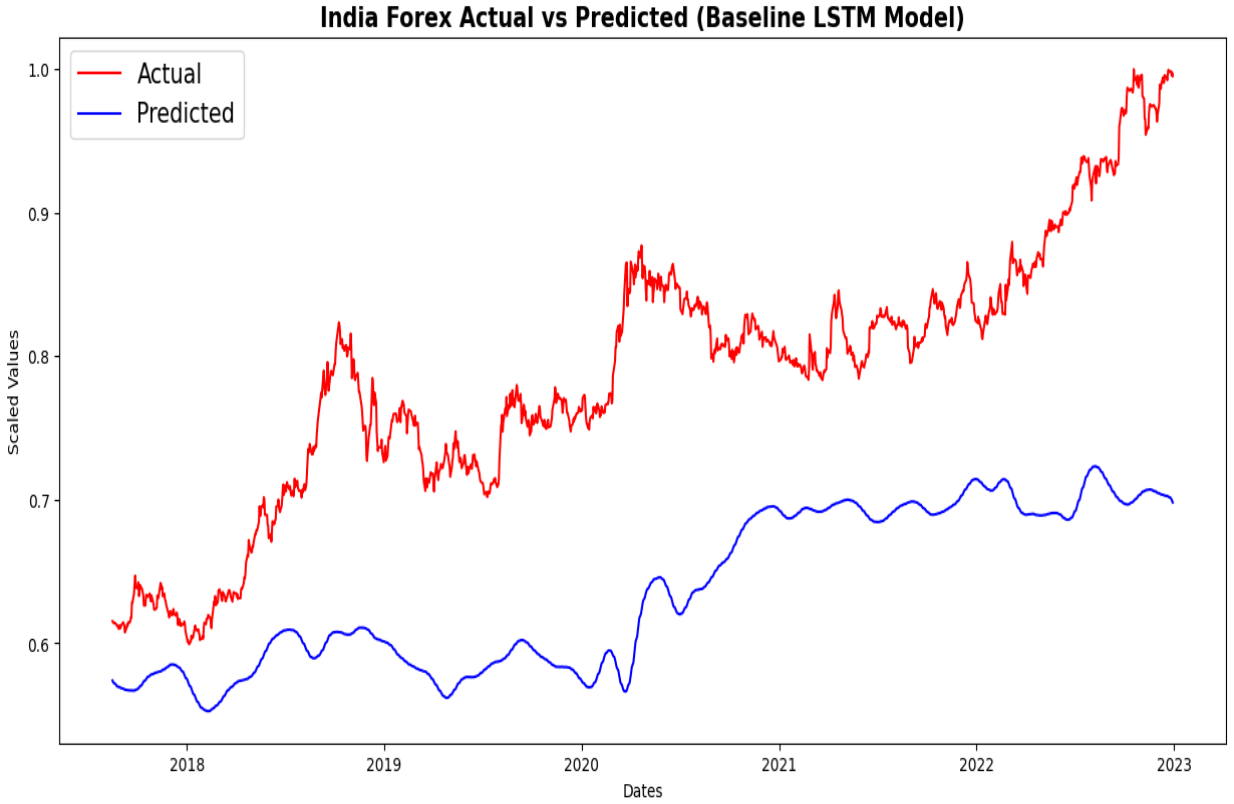


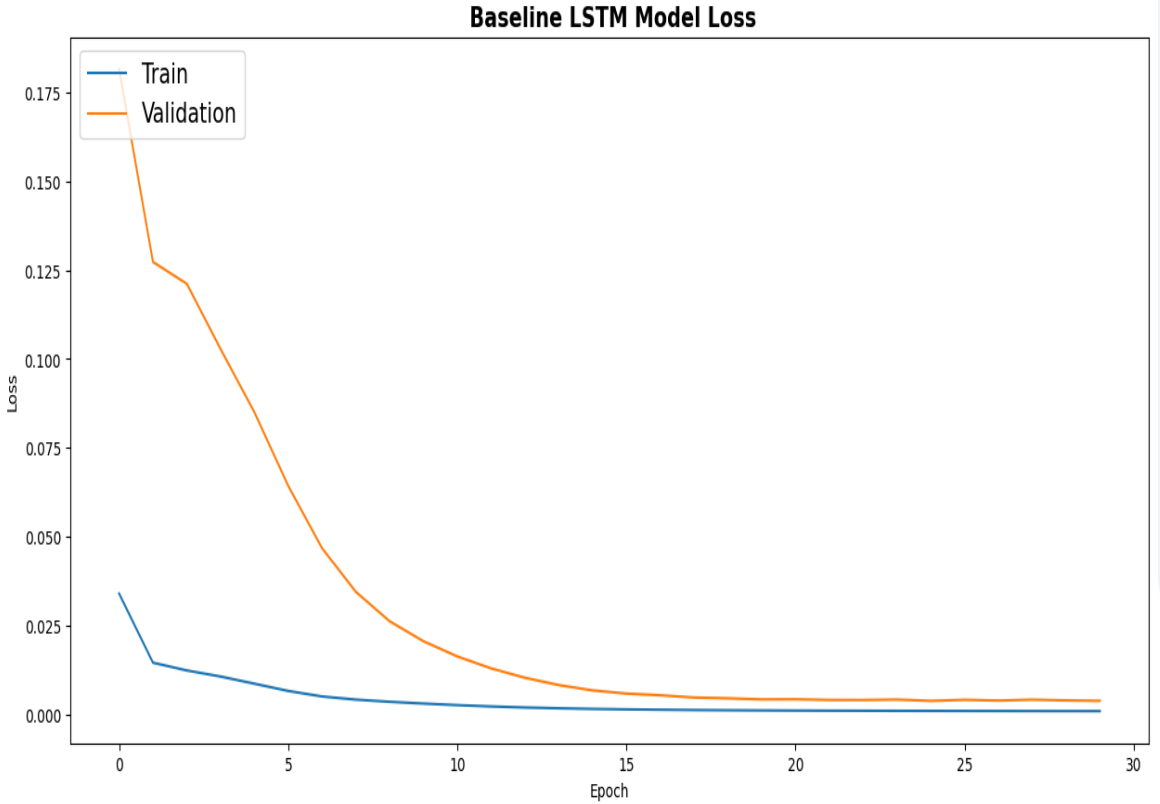


- Baseline Model using RNN created poor results as we can see from the graph above.
- It consists of 4 RNN Layers and a Dense Layer.

- The loss gradually decreased with some abnormalities.

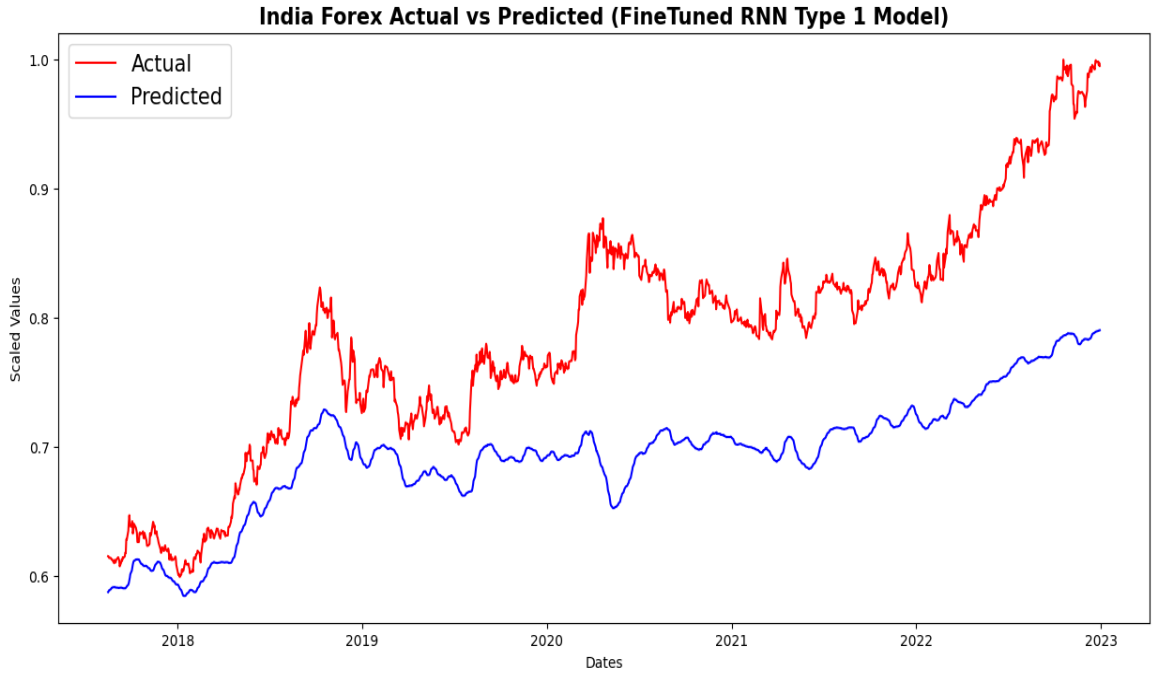
Model Evaluation : Baseline Model (LSTM)

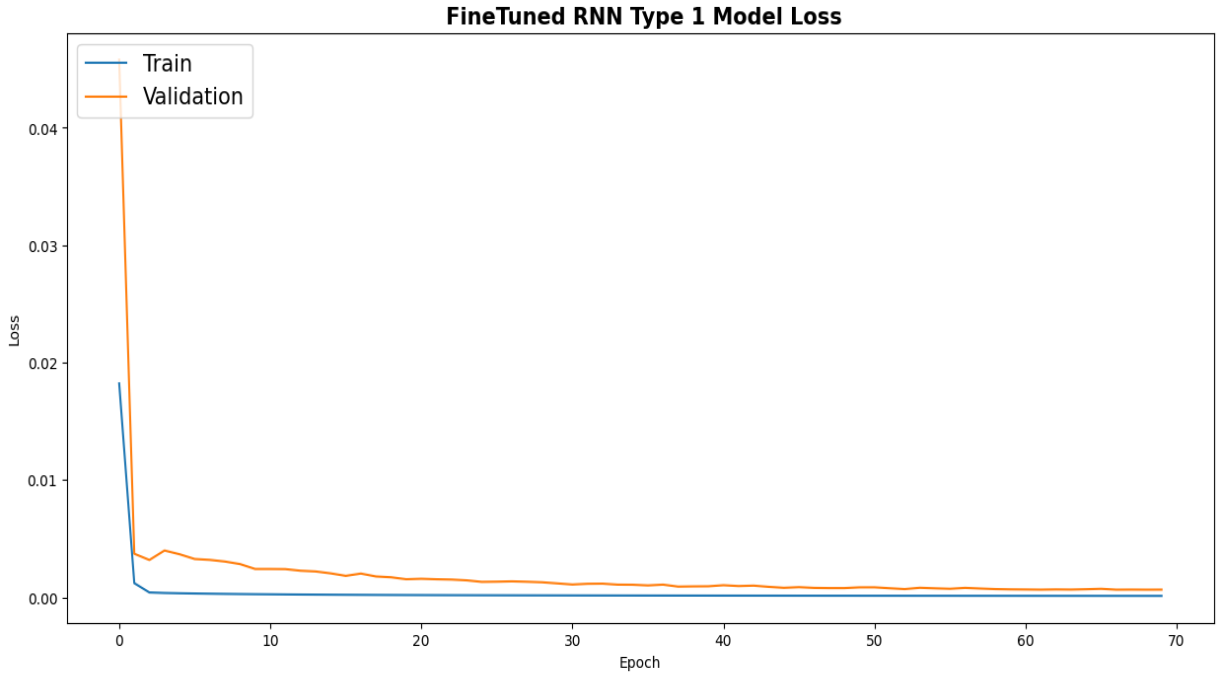




- LSTM Baseline Model also produced poor results.
- It consists of 4 LSTM Layers and a Dense Layer.
- The reduction in loss on the validation set was not substantial.

Model Evaluation : Transfer Learn RNN (Type 1: Substantial)

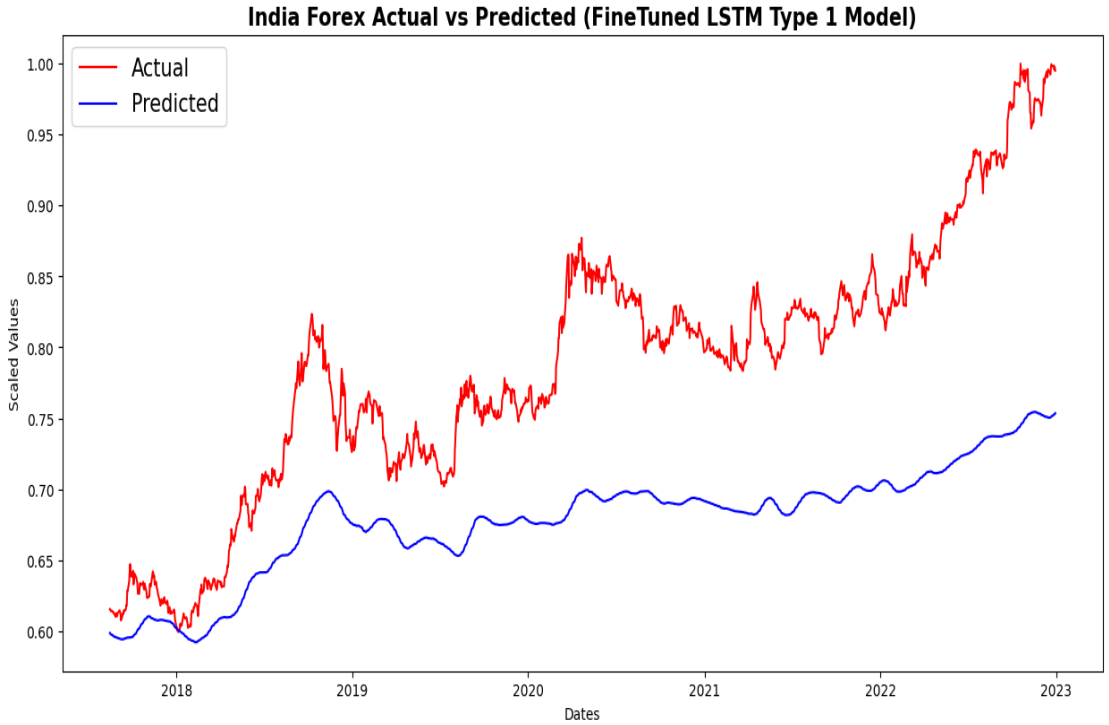


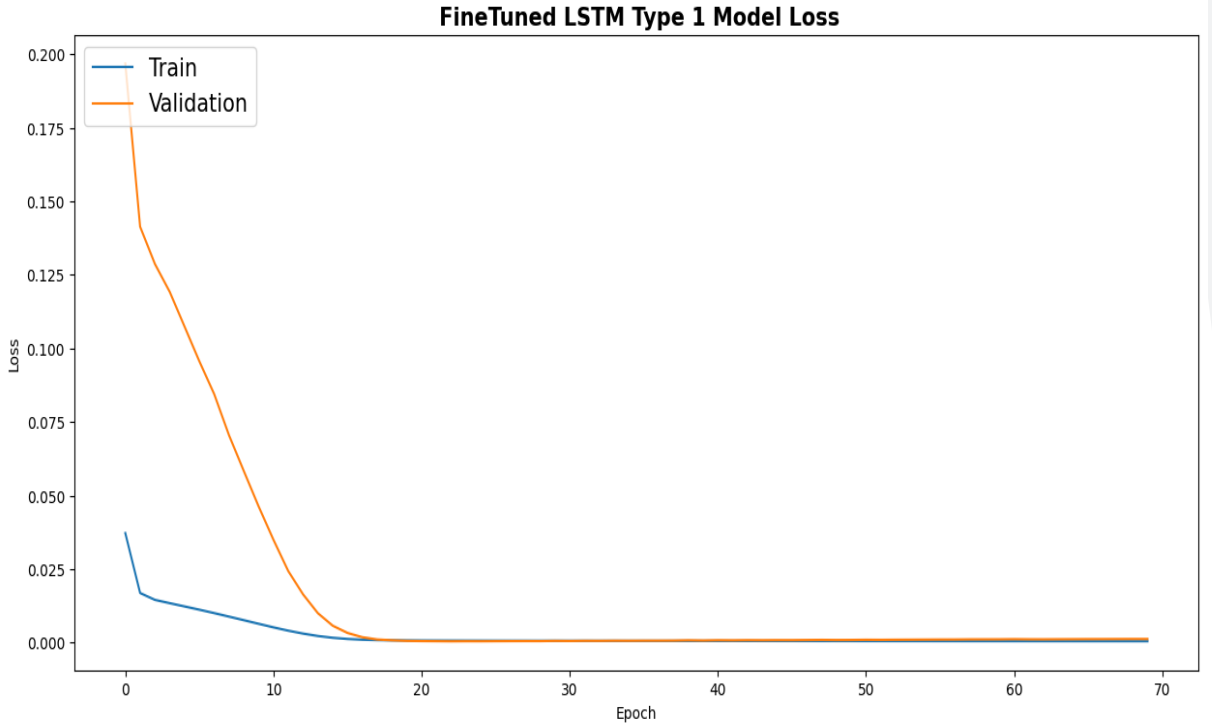


- Pretrained RNN with substantial data of India used while training had significant improvements.
- Fine-tuned model on a new RNN Layer with only India dataset.

- Graph shows loss function during Fine tuning
- The loss decreased quickly both in the training as well as the validation.

Model Evaluation : Transfer Learning LSTM (Type 1)

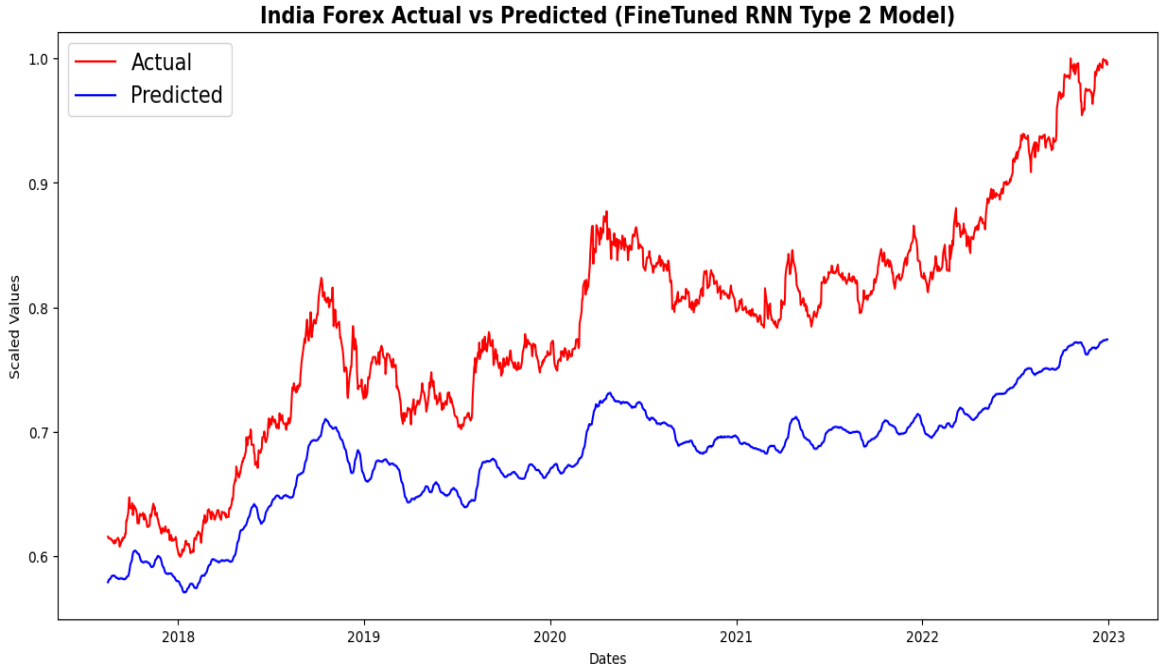


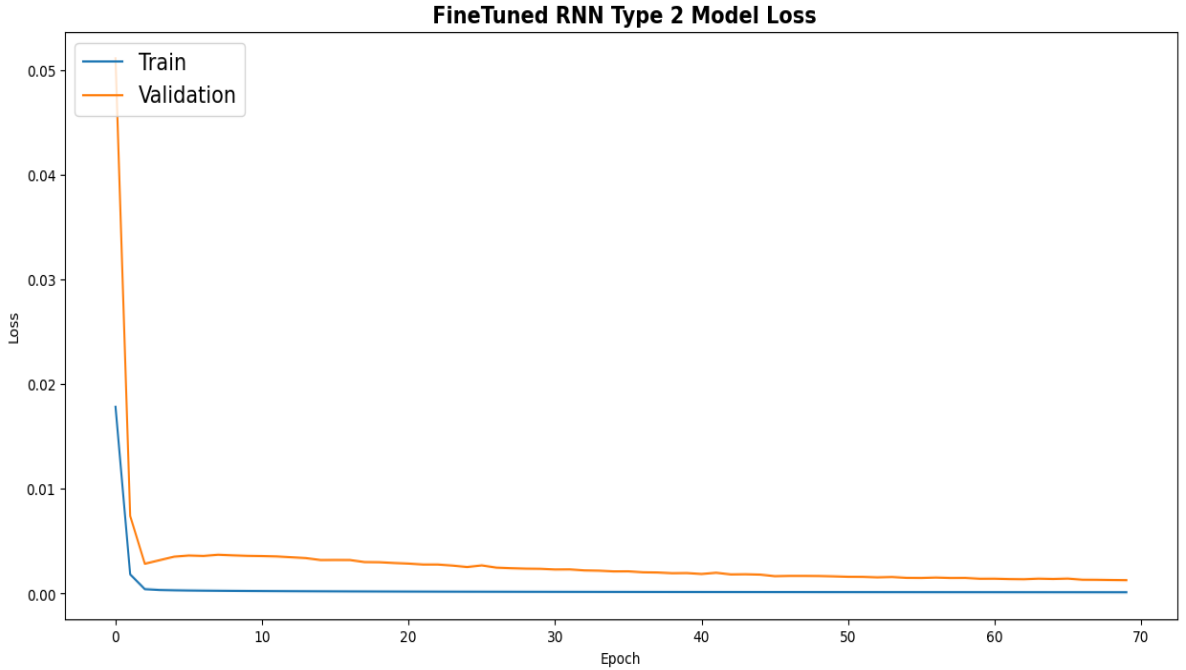


- Pretrained LSTM with substantial data of India used while training had significant improvements.
- Fine-tuned model on a new LSTM Layer on India dataset.

- Graph shows loss function during Fine tuning
- The loss decreased quickly both in the training as well as the validation.

Model Evaluation : Transfer Learn RNN (Type 2: Partial)

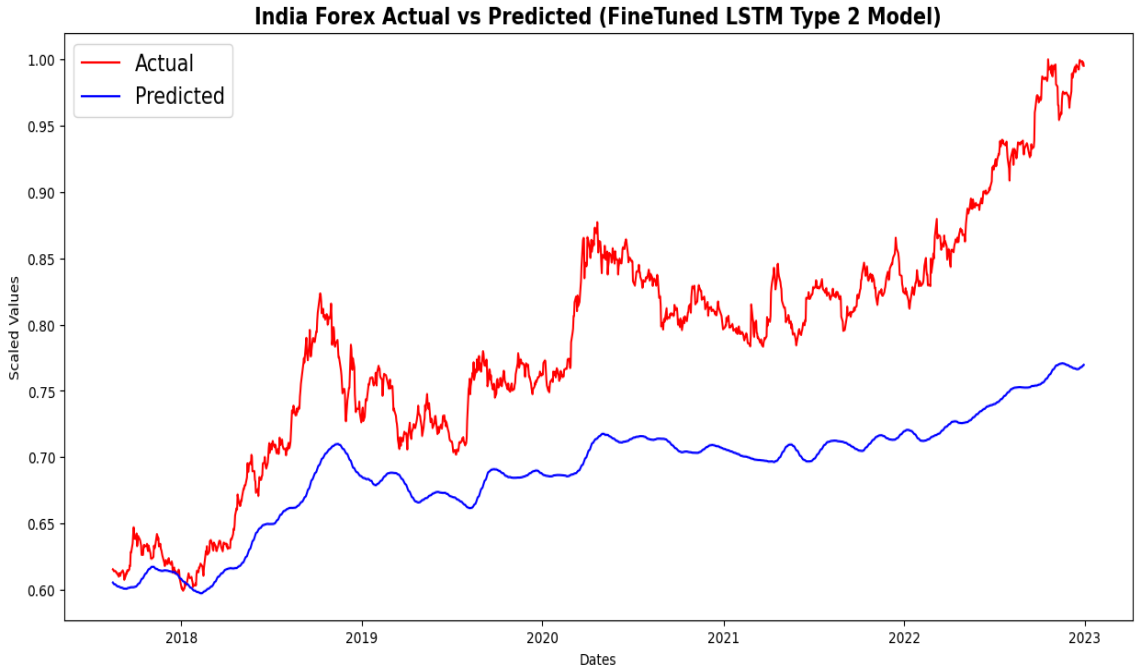


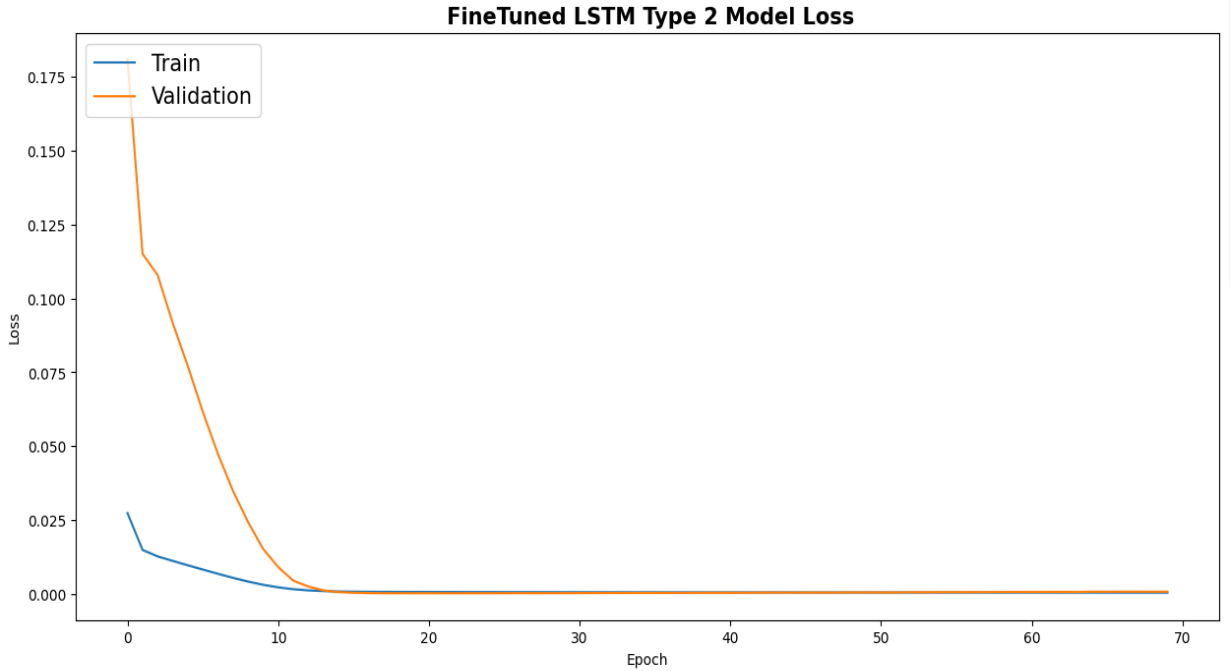


- Pretrained RNN with partial data of India used while training showed slightly poor results compared to substantial target data.
- Fine-tuned model on a new RNN Layer with only India dataset.

- Graph shows loss function during Fine tuning
- The loss decreased quickly both in the training as well as the validation.

Model Evaluation : Transfer Learn LSTM (Type 2: Partial)

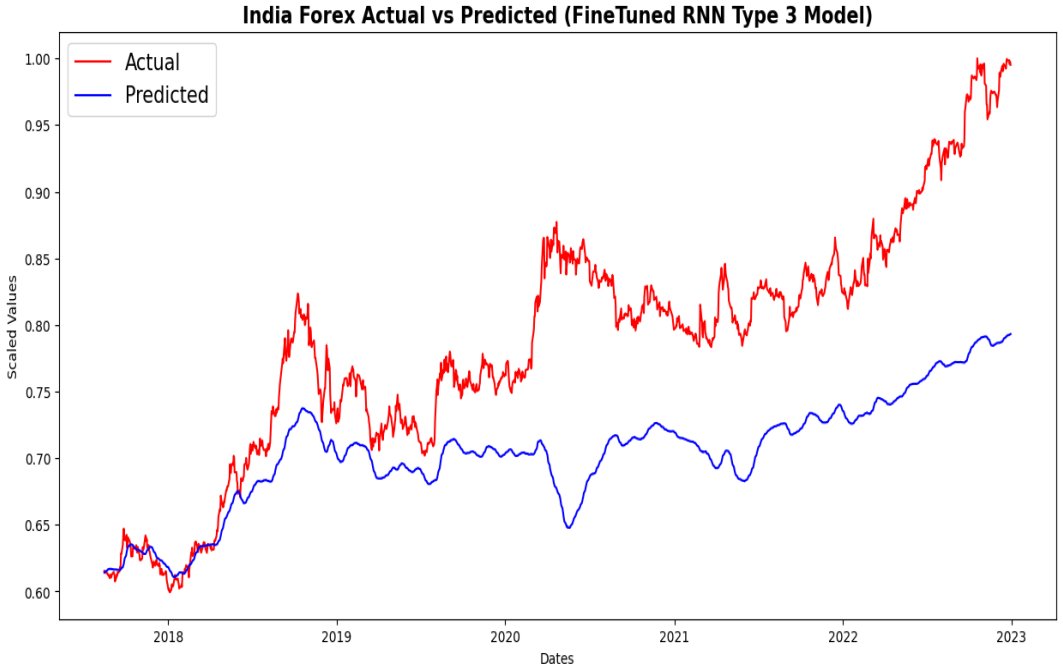


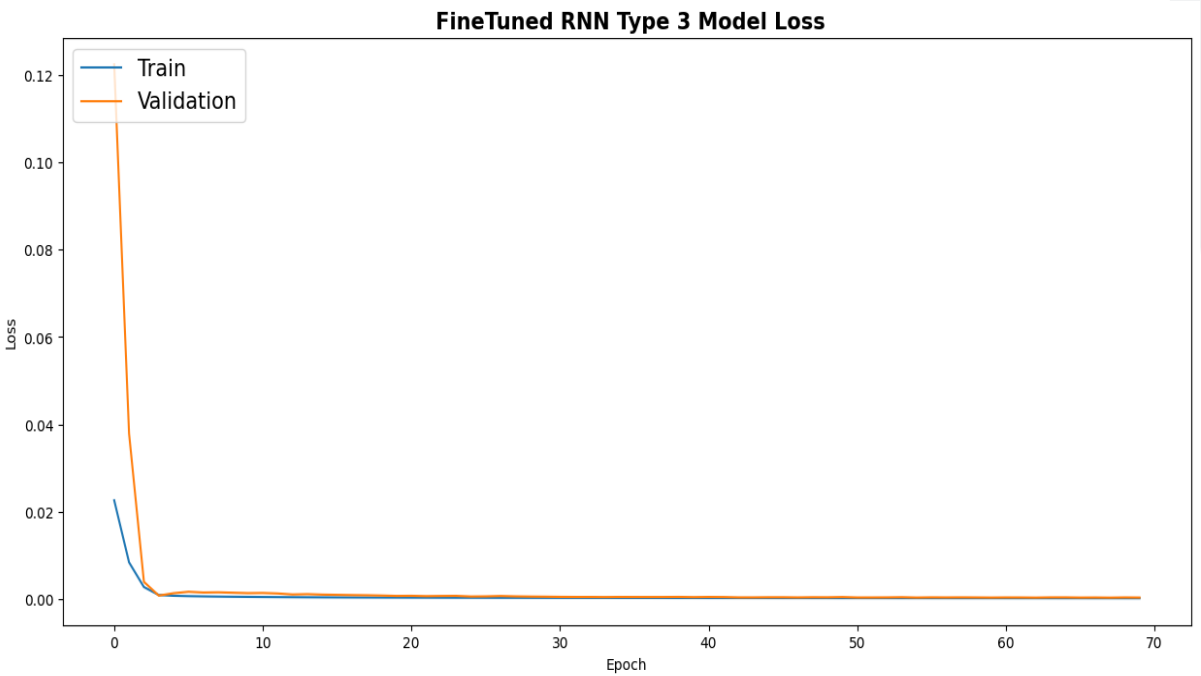


- Pretrained LSTM with partial data of India used while training showed slightly better results compared to substantial target data.
- Fine-tuned model on a new LSTM Layer with only India dataset.

- Graph shows loss function during Fine tuning
- The loss decreased quickly both in the training as well as the validation.

Model Evaluation : Transfer Learn RNN (Type 3: No Data)

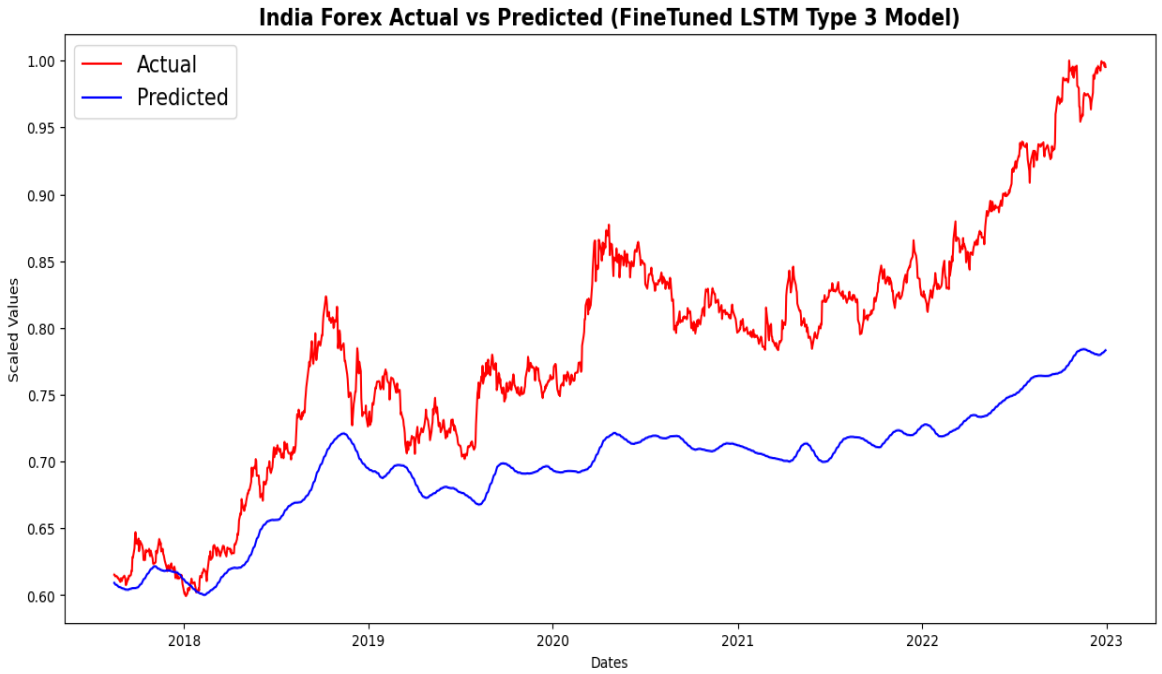


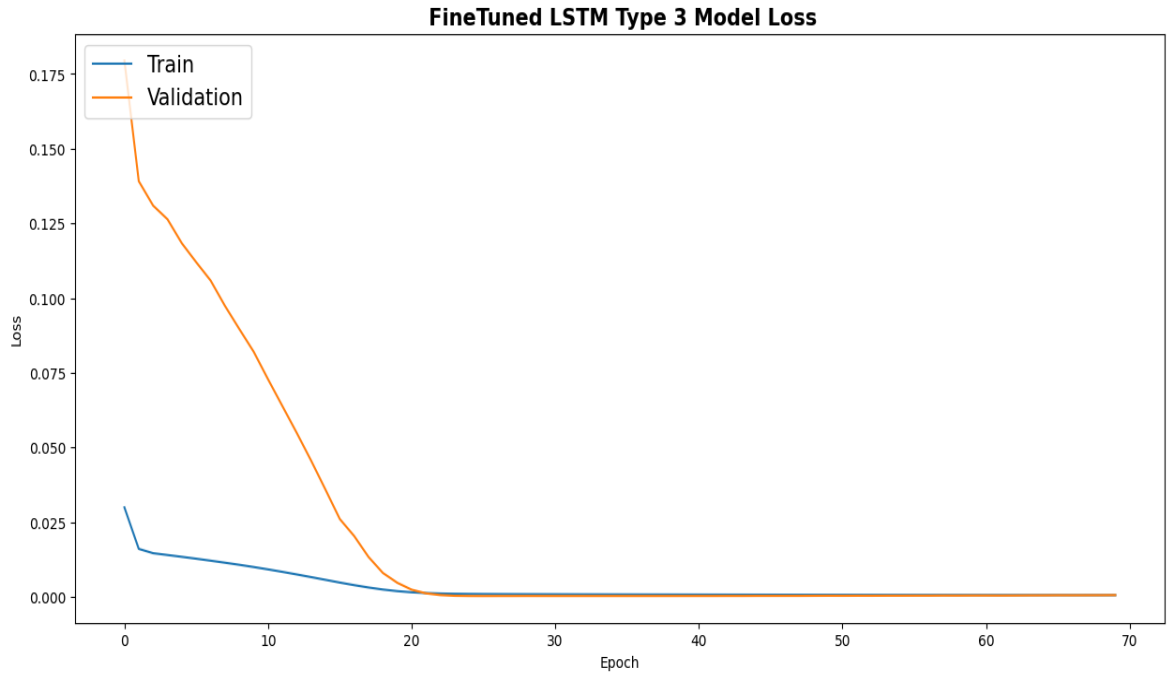


- Performance increased from earlier methods of RNN as the target data was not given during pretraining.
- Fine-tuned model on a new RNN Layer. Using only India Dataset

- Graph shows loss function during Fine tuning
- The loss decreased quickly both in the training as well as the validation.

Model Evaluation : Transfer Learn LSTM (Type 3: No Data)

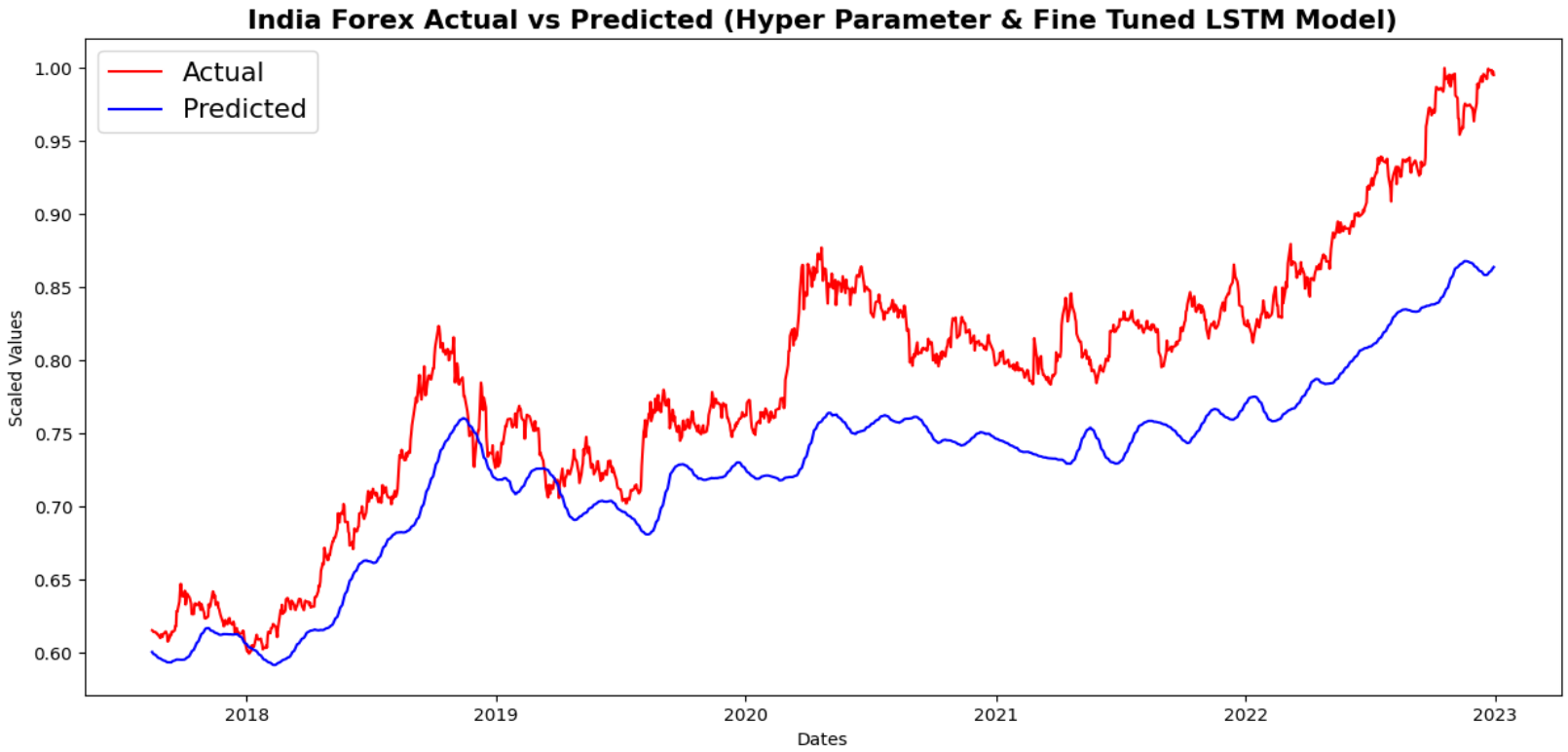




- The model performed much better compared to the previous models.
- Fine-tuned model on a new LSTM Layer.

- Graph shows loss function during Fine tuning
- The loss decreased quickly both in the training as well as the validation.

Model Evaluation : HyperParameter Tuned LSTM



- Hyper-Parameter tuned model outperformed all other models
- The parameter chosen indicates that previous models were not complex enough to capture the trends of Forex Data

Model Comparison

- The table shows the performance of two models (RNN and LSTM) on the following metric scores RMSE, MAE, and R^2.
- As we can see LSTM consistently out-performed RNN in all Transfer Learning formats and in Base model
- Results show Hyperparameter Tuning is a significant step to improve model performance

Sr.No.	Model	MSE	MAE	R^2
1	Baseline RNN	0.048	0.208	-4.73
2	Baseline LSTM	0.026	0.149	-2.081
3	RNN (Substantial Target Data)	0.011	0.092	-0.325
4	LSTM (Substantial Target Data)	0.014	0.107	-0.748
5	RNN (Partial Target Data)	0.013	0.105	-0.585
6	LSTM (Partial Target Data)	0.011	0.095	-0.420
7	RNN (No Target Data)	0.010	0.083	-0.196
8	LSTM (No Target Data)	0.010	0.089	-0.260
9	HP Tuned LSTM	0.004	0.059	0.446

Project Insights

- Project Limitations
- Conclusion

Project Limitations

- **Unaccounted Parameters Impacting Forex Exchange Rates:** Several parameters beyond the scope of our analysis, such as Inflation, Government Debts, Political Stability, and Economic Recession, influence Forex Exchange Rates. Regrettably, due to unavailable data, we couldn't incorporate these crucial variables alongside Normalized GDP, CPI, and Interest Rates.
- **Constraints in Model Complexity due to Computational Limitations:** The computational resources at our disposal limited the complexity of the models constructed for analysis. Given access to higher computational power, we could explore and develop more intricate models, enhancing our capacity to effectively train and analyze the available data.

Conclusion

- Using Deep Learning, it is possible to capture trends in Foreign Exchange rate with decent accuracy.
- Using information about other countries as features to predict the Forex Rate of some country did not perform well
- Creating a generalized model which can take input parameter of a county and predict the forex rate of that country only gave better result.
- Using transfer learning to create a fine-tuned model for some country gave much better result.
- Overall, LSTMs performed well compared to RNN models.
- Hyperparameter tuning's result suggests that the initial models tested were not complex enough.

STEVENS
INSTITUTE OF TECHNOLOGY
1870

Thank You !