A Comparative Study on Deep Learning Models for Time Series Forecasting on infectious diseases

Lokendra Kumar, Abani Singha, Abhinav TK,
Gharib Mohamed Saleh, Muhammed Dilshah U
Under the Supervision of
Dr. Priyanka Shukla

Department of Mathematics
Indian Institute of Technology Madras

April 29, 2024

CONTENT

1. INTRODUCTION
2. EXTENDED BACKGROUND
    a. TIME SERIES FORECASTING
    b. SIR MODELS
3. EXPLORING THE MODELS
    a. RNN
    b. LSTM
    c. Prophet
4. METHODOLOGY
    a. Data Analysis & Visualization
    b. Experiment Setup
    c. Evaluation Metrics
5. RESULTS & CONCLUSION

Introduction

❑ Time series forecasting is a valuable tool for predicting future trends based on historical data.

❑ Commonly used techniques: ARIMA, exponential smoothing, and neural networks.

❑ Applications in various industries like finance, marketing, and healthcare.

❑ Understanding the underlying patterns in time series data is crucial.

❑ **Steps involved:** Data Collection, Data Preprocessing, EDA, Model Selection, Model Training, Model Evaluation, Forecasting, Monitoring and Updating

Motivation

- ❑ **Historical Lessons:** Historical pandemics like the 1918 Spanish flu highlight the global threat posed by infectious diseases.

- ❑ **Global Connectivity:** With modern air travel, emerging infectious diseases can swiftly traverse borders, necessitating proactive measures to prevent and manage outbreaks.

- ❑ **COVID-19 Impact:** The COVID-19 pandemic has left an indelible mark, with over 703 million infections and nearly 7 million deaths worldwide, underscoring the ongoing threat posed by infectious diseases.

- ❑ **Effective Interventions:** Identifying and implementing effective strategies to contain and eliminate endemic diseases remains a pivotal goal, emphasizing the importance of infectious disease modeling for targeted interventions.

Problem Statement

“Identify optimal deep learning models for accurate infectious disease forecasting to improve disease control strategies”

OBJECTIVES

❑ Conduct a comprehensive literature review to evaluate the effectiveness of conventional statistical models and deep learning techniques utilized for time series forecasting.

❑ Assess the accuracy and reliability of various disease forecasting methodologies by conducting a thorough review of past studies in the field.

❑ Apply deep learning models to epidemiological datasets to explore their effectiveness in forecasting disease outbreaks and patterns.

❑ Perform a comparative study to assess the predictive performance of various forecasting models

Literature Review

- Time Series Analysis and Forecasting of Coronavirus Disease in Indonesia Using ARIMA Model and Prophet
- Deep Learning Methods for Forecasting COVID-19 Time-Series Data: A Comparative Study
- A Comparison of ARIMA and LSTM in Forecasting Time Series
- Time Series Forecasting of COVID-19 Using Deep Learning Models:India-USA Comparative Case Study
- Inefficiency of SIR models in forecasting COVID-19 epidemic: a case study of Isfahan

Model Selection

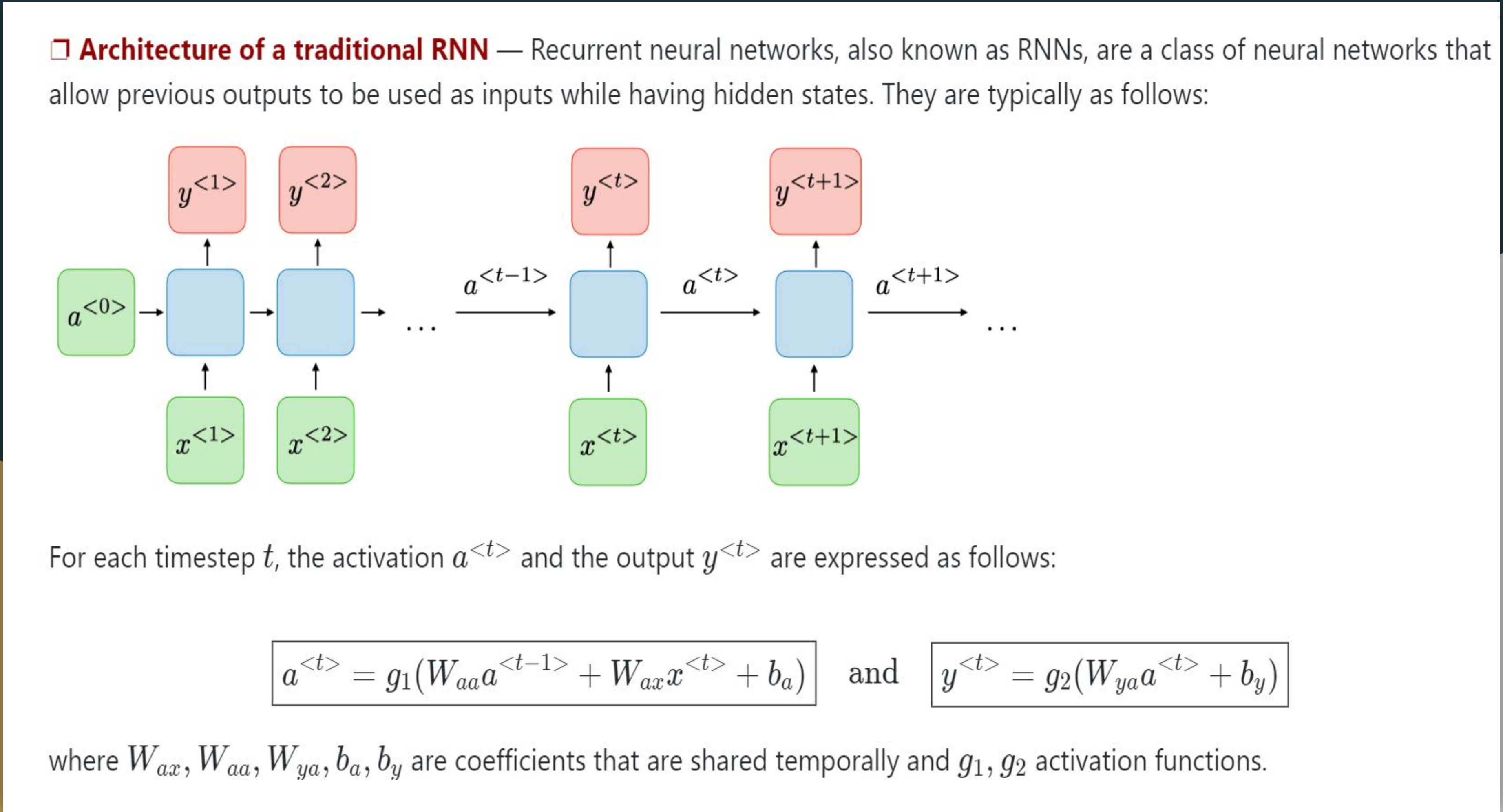
- ❑ We can utilize various mathematical/statistical models such as SIR, SEIR and ARIMA.
- ❑ Limitations:
  - ❑ Simplistic assumption
  - ❑ Limited capacity for complexity
- ❑ Deep learning models are capable of capturing nonlinear relationships and temporal dependencies that traditional statistical models might miss.
- ❑ Our selections include:
  - ❑ Recurrent Neural Networks(RNN)
  - ❑ Long Short-Term Memory(LSTM)
  - ❑ PROPHET(Developed By Facebook)

What are Recurrent Neural Networks?

❑ A recurrent neural network (RNN) is a type of artificial neural network that uses sequential data or time series data.

❑ RNN is essential in deep learning for analyzing temporal correlations in time series prediction due to their sequential memory handling, retaining information from past time steps.

❑ RNN is commonly used for ordinal problems, such as language translation, natural language processing (nlp), speech recognition and etc.

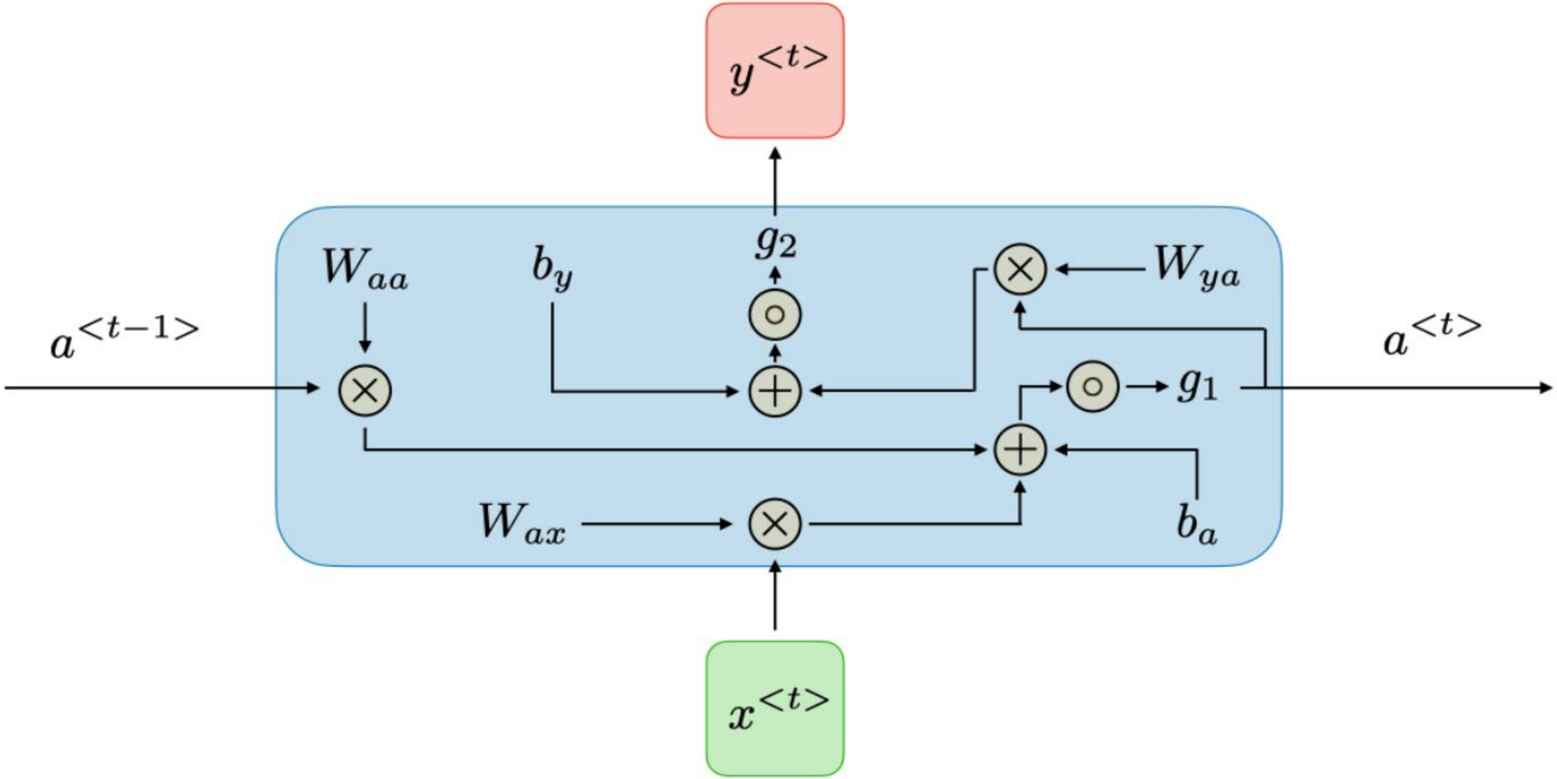


❒ **Architecture of a traditional RNN** — Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. They are typically as follows:

For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)} \quad \text{and} \quad \boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.



- Mathematical calculation inside one Hidden layer

Vanishing and Exploding Gradients Problem

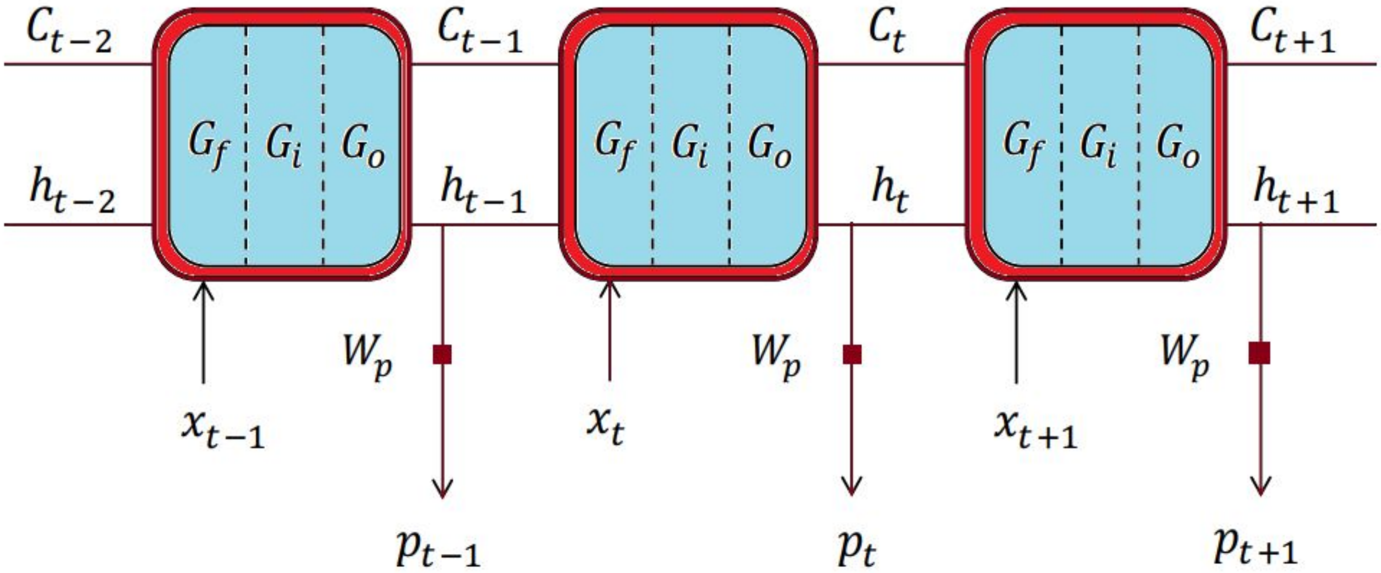
❏ The vanishing and exploding gradient phenomena are often encountered in the context of RNNs.

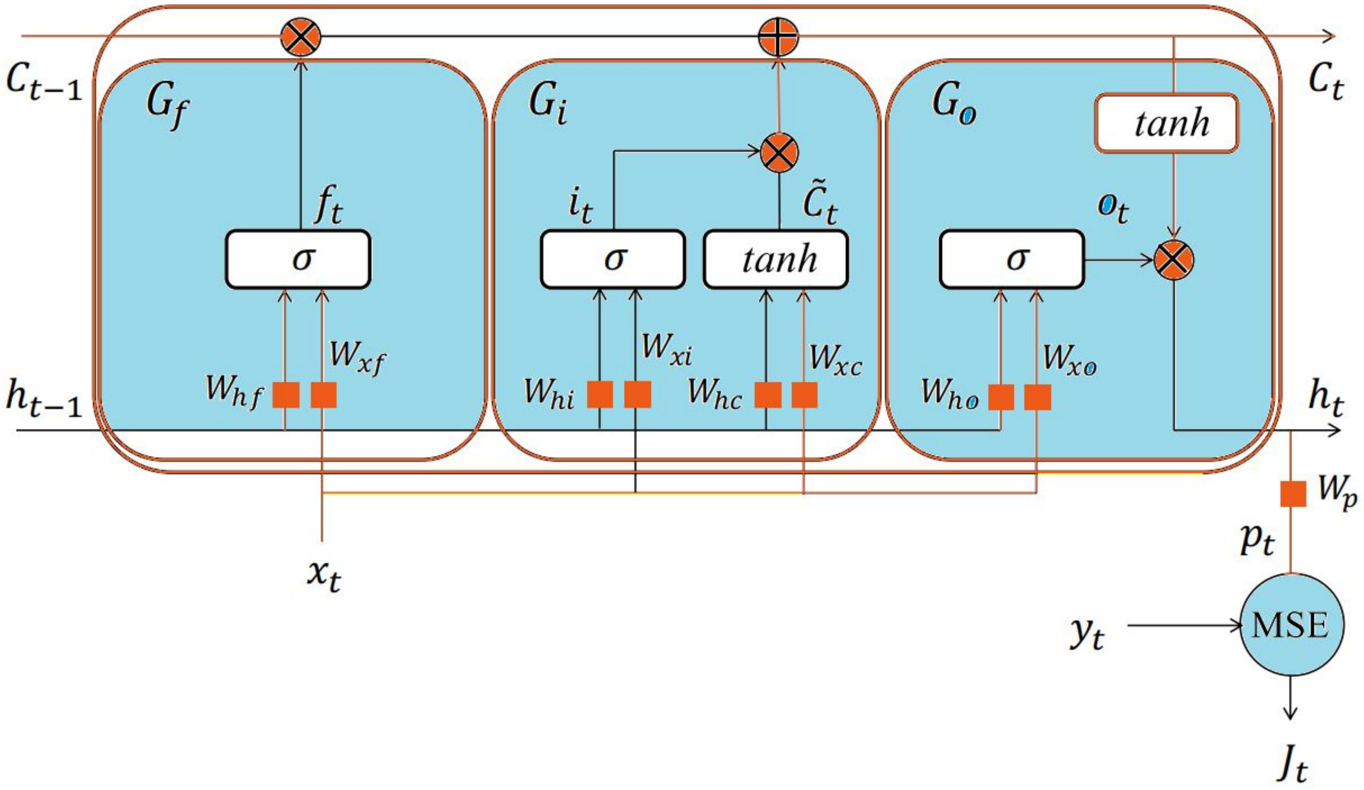
❏ The reason why they happen is that it is difficult to capture long term dependencies because of multiplicative gradient that can be exponentially decreasing/increasing with respect to the number of layers.

What are LSTM Models?

- ❑ LSTM stands for Long Short Term Memory, and it is a variant of Recurrent Neural Network (RNN).
- ❑ It is designed to address the limitations of traditional RNNs in learning long-term dependencies.
- ❑ In LSTM, the hidden layers of RNN are replaced with memory cells, allowing better retention of information over time.
- ❑ The architecture of LSTM includes different gate units, namely input gate (it), output gate (ot), and forget gate (ft).

❑ Architecture of a traditional LSTM Model:





- Mathematical calculation and all the Gates inside one Hidden layer

Mathematical Formulation of LSTM

$$f_t = \sigma(W_{hf} * h_{t-1} + W_{xf} * x_t + b_f)$$

$$i_t = \sigma(W_{hi} * h_{t-1} + W_{xi} * x_t + b_i).$$

$$\overline{C}_t = \tanh(W_{hc} * h_{t-1} + W_{xc} * x_t + b_c).$$

$$C_t = f_t \oplus C_{t-1} + i_t \oplus \overline{C_t}$$

$$O_t = \sigma(W_{ho} * h_{t-1} + W_{xo} * x_t + b_o).$$

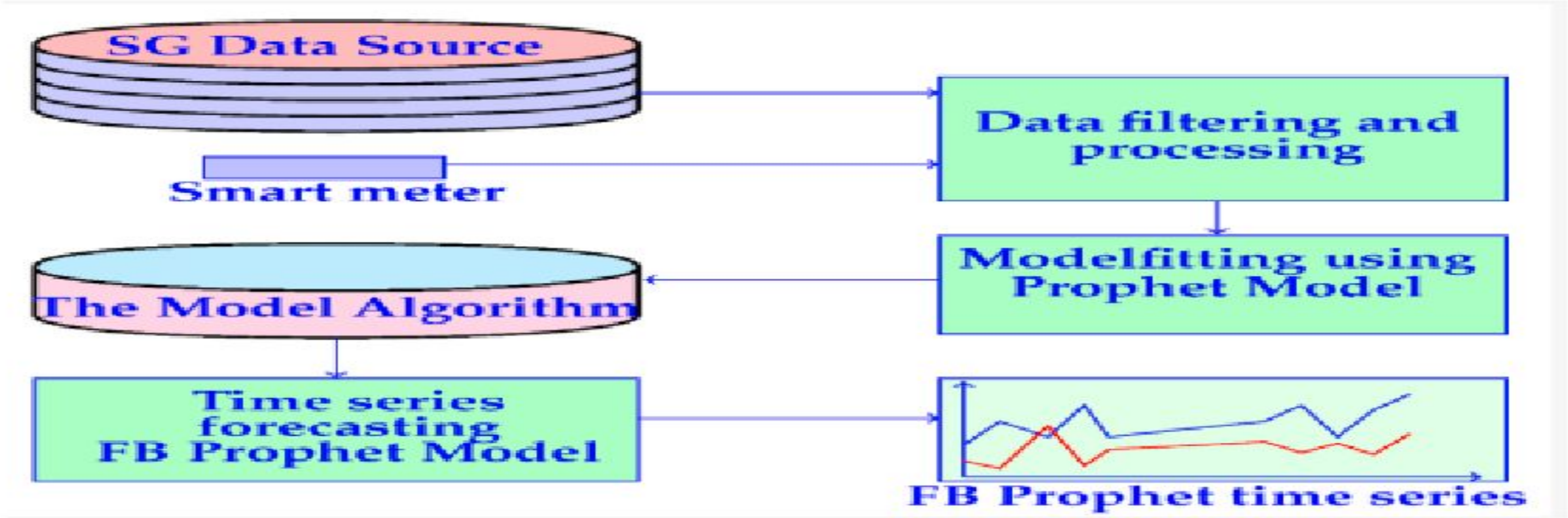
$$h_t = O_t \odot \tanh(C_t).$$

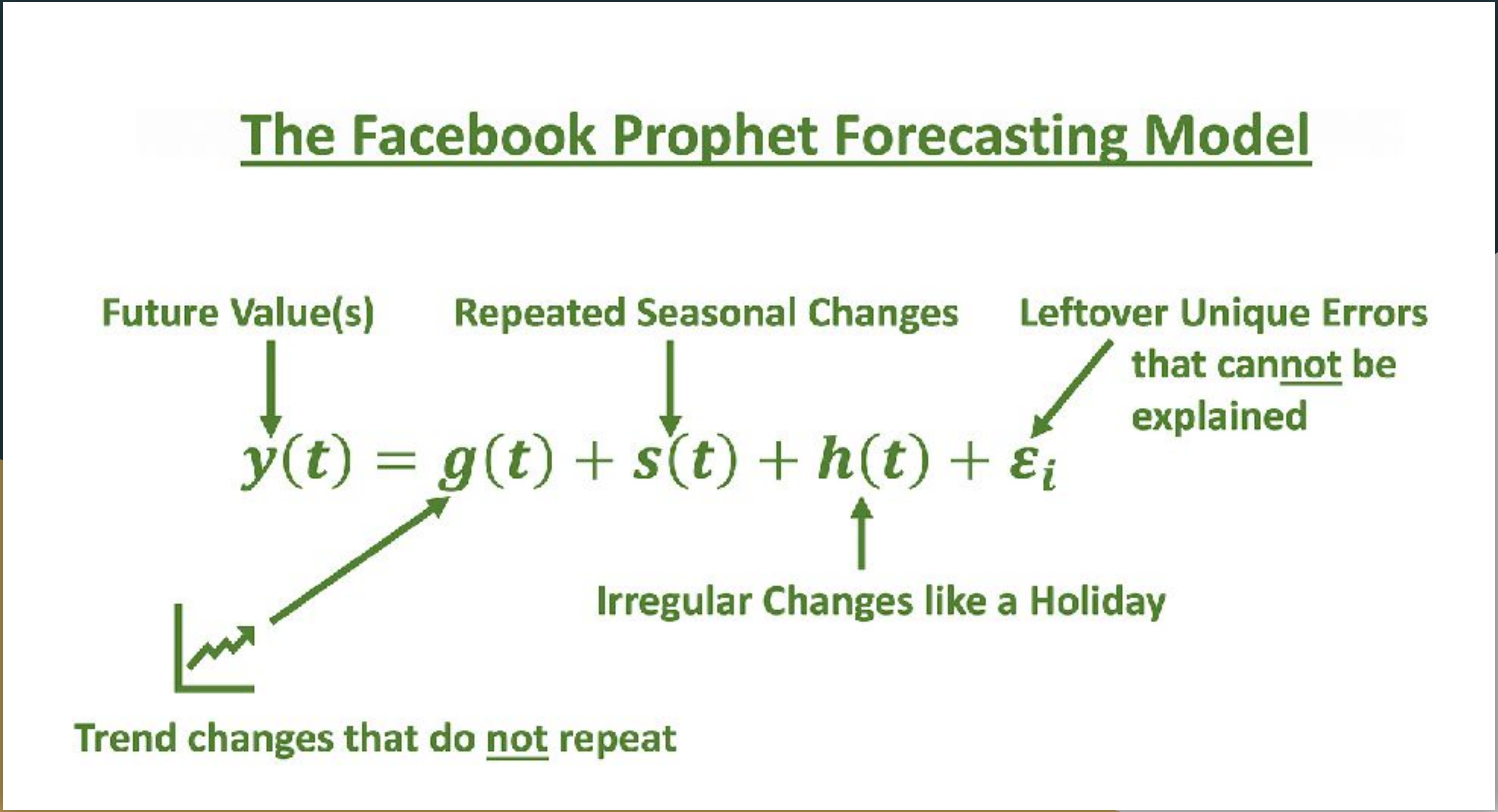
Where σ is logistic sigmoid function and tanh is tanh function used as activation and i, f, c, o are input gate, forget gate, memory cell and output gate respectively.

PROPHET MODEL

- ❑ Prophet is an open source software available in Python and R for forecasting time series data.
- ❑ Prophet is published by Facebook’s Core Data Science team.
- ❑ Prophet is always a time series with two input features: date dt and value x.

Prophet Model Workflow


SG Data Source
Smart meter
Data filtering and processing
Modelfitting using Prophet Model
The Model Algorithm
Time series forecasting FB Prophet Model
FB Prophet time series



The Facebook Prophet Forecasting Model

$$\boldsymbol{y(t) = g(t) + s(t) + h(t) + \varepsilon_i}$$

Future Value(s)

Repeated Seasonal Changes

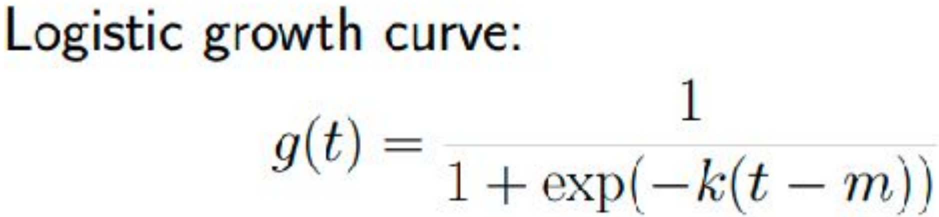
Leftover Unique Errors that cannot be explained

Irregular Changes like a Holiday

Trend changes that do not repeat

GROWTH OR TREND

❑ Prophet is divided into Logistic Growth Model and PieceWise Linear Model.

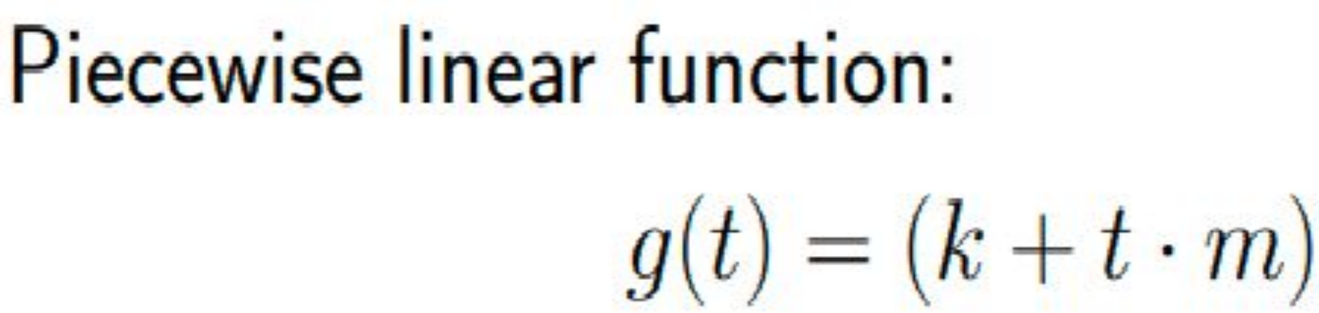


Logistic growth curve:

$$g(t) = \frac{1}{1 + \exp(-k(t - m))}$$

Where k is the growth rate factor, and m is the offset parameter.

- The piecewise linear model is fit using the following equation.



Piecewise linear function:

$$g(t) = (k + t \cdot m)$$

- Where k is the growth rate factor and m is the offset parameter.

HOLIDAYS AND EVENTS: h(t)

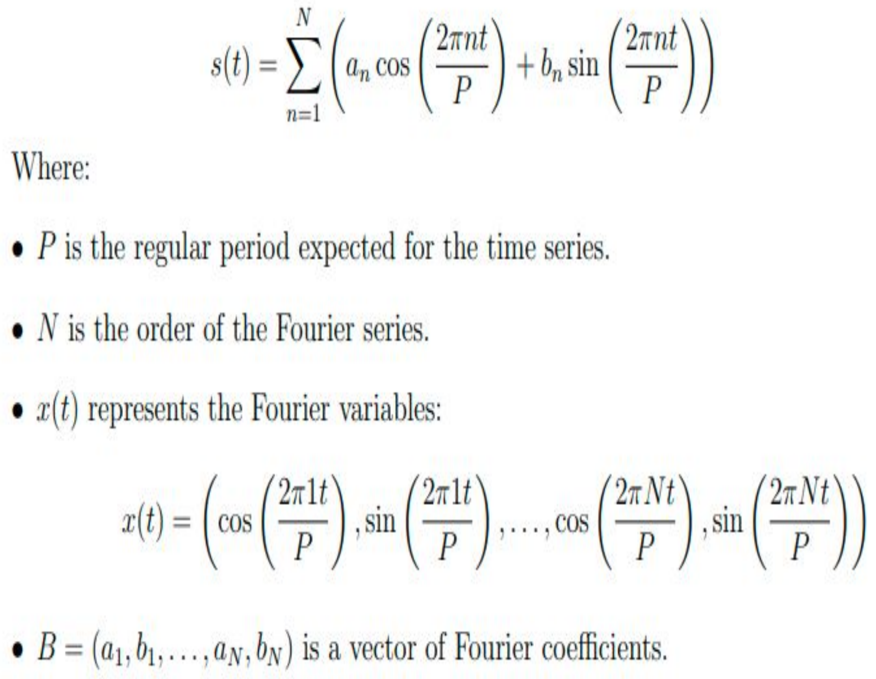
Holidays provides large predictable shocks to many time series and often do not follow a periodic pattern.

❑ Holidays show characteristics of specific day.

SEASONALITY: s(t)

❑ It means periodic or short term changes.

❑ It can be weekly(7 days) or yearly(365 days) seasonal effects.



$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$$

Where:

- P is the regular period expected for the time series.
- N is the order of the Fourier series.
- $x(t)$ represents the Fourier variables:

$$x(t) = \left(\cos\left(\frac{2\pi 1t}{P}\right), \sin\left(\frac{2\pi 1t}{P}\right), \ldots, \cos\left(\frac{2\pi Nt}{P}\right), \sin\left(\frac{2\pi Nt}{P}\right) \right)$$

- $B = (a_1, b_1, \ldots, a_N, b_N)$ is a vector of Fourier coefficients.

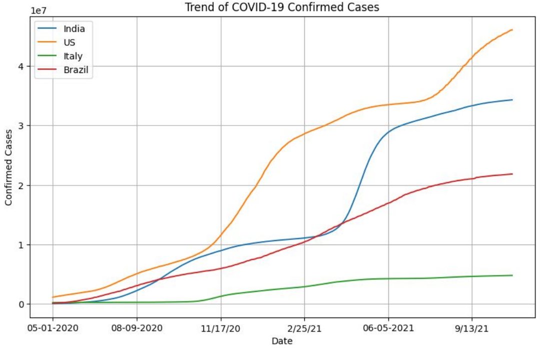
METHODOLOGY

Data Collection

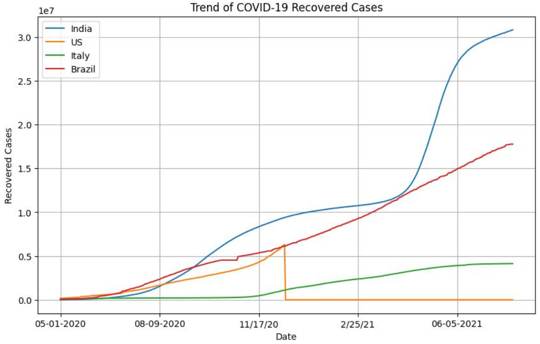
- Data collected from *https://github.com/CSSEGISandData/COVID-19/tree/master*
- COVID-19 time series data from *22 Jan 2020* to *31 Oct 2021* of more than 200 countries.
- The dataset consists of confirmed cases, recovered cases and deaths of these countries.
- We focused on countries like the US, India, Italy and Brazil since they showed more consistent data.

Data Analysis & Visualization

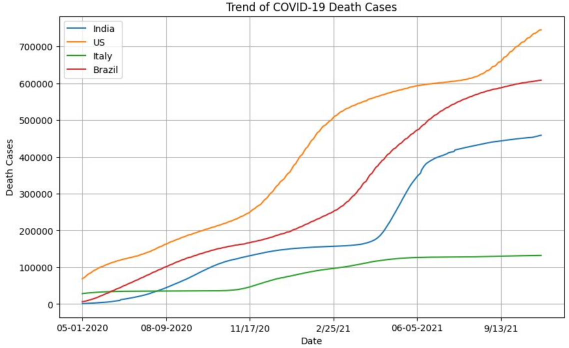
1. **Trend Analysis:**
   Plot the daily cases for each country over time to visualize the trend and identify any significant changes or patterns.



Trend of COVID-19 Confirmed Cases



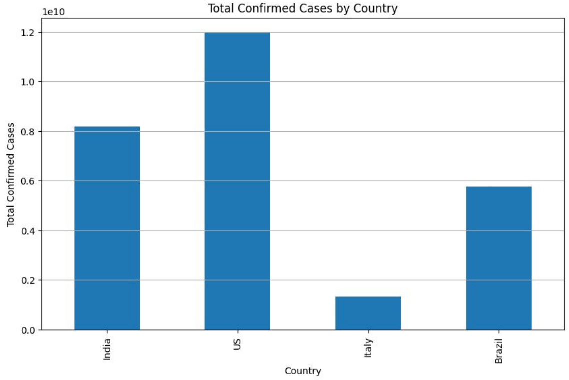
Trend of COVID-19 Recovered Cases

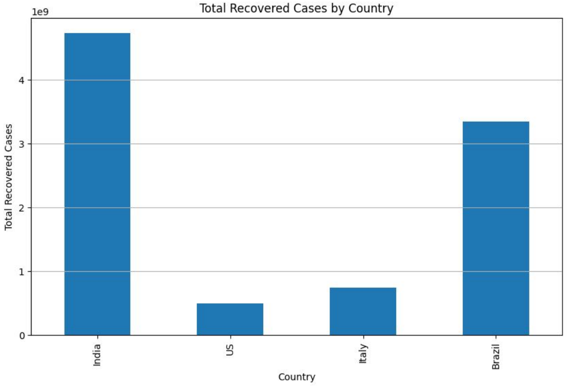


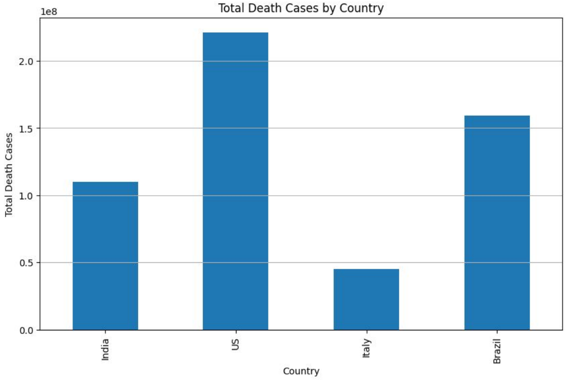
Trend of COVID-19 Death Cases

Data Analysis & Visualization

2. Comparative Analysis:
Compare the total cases across different countries using bar charts or line graphs to see how the pandemic has unfolded in different regions.



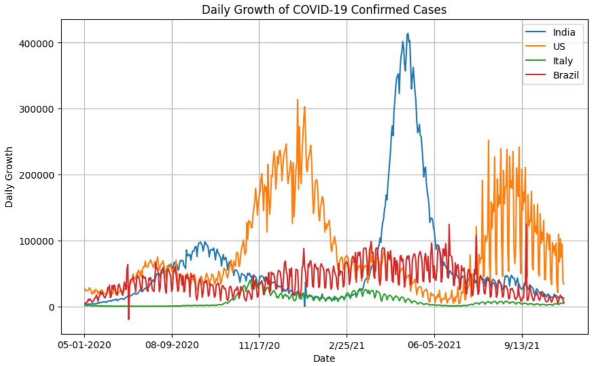




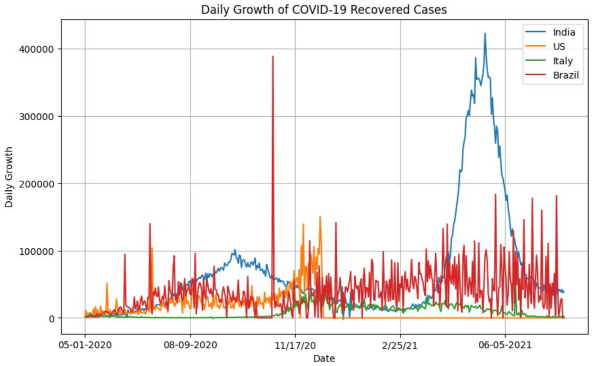
Data Analysis & Visualization

3. Growth Rate Analysis:

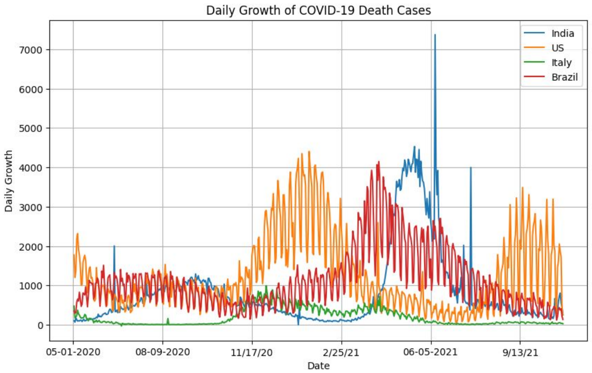
Calculate and visualize the daily or weekly growth rate of cases for each country to understand the rate of spread.



Daily Growth of COVID-19 Confirmed Cases



Daily Growth of COVID-19 Recovered Cases

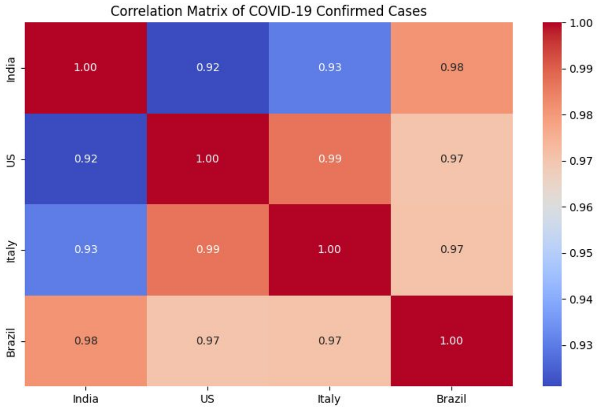


Daily Growth of COVID-19 Death Cases

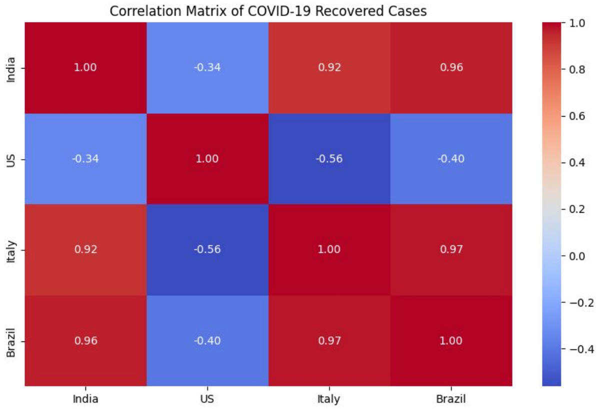
Data Analysis & Visualization

4. Correlation Analysis:

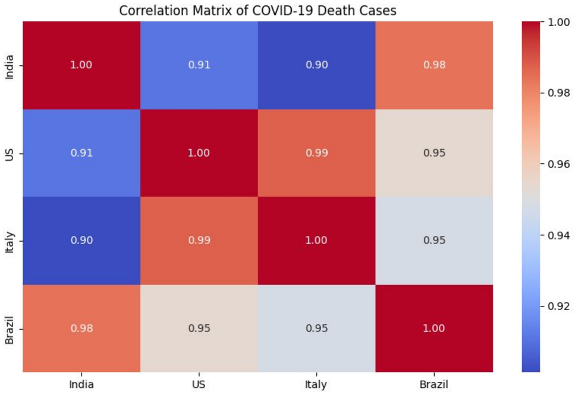
Explore potential correlations between the number of cases in different countries over time to understand how the pandemic has spread globally.



Correlation Matrix of COVID-19 Confirmed Cases



Correlation Matrix of COVID-19 Recovered Cases



Correlation Matrix of COVID-19 Death Cases

Experiment Setup

Hyperparameter tuning using wandb

Wandb sweeps: We define a search space for the hyperparameters, and wandb automatically explores different combinations, tracking the performance of each configuration. This helps to identify the best set of hyperparameters for our model without manual trial and error.

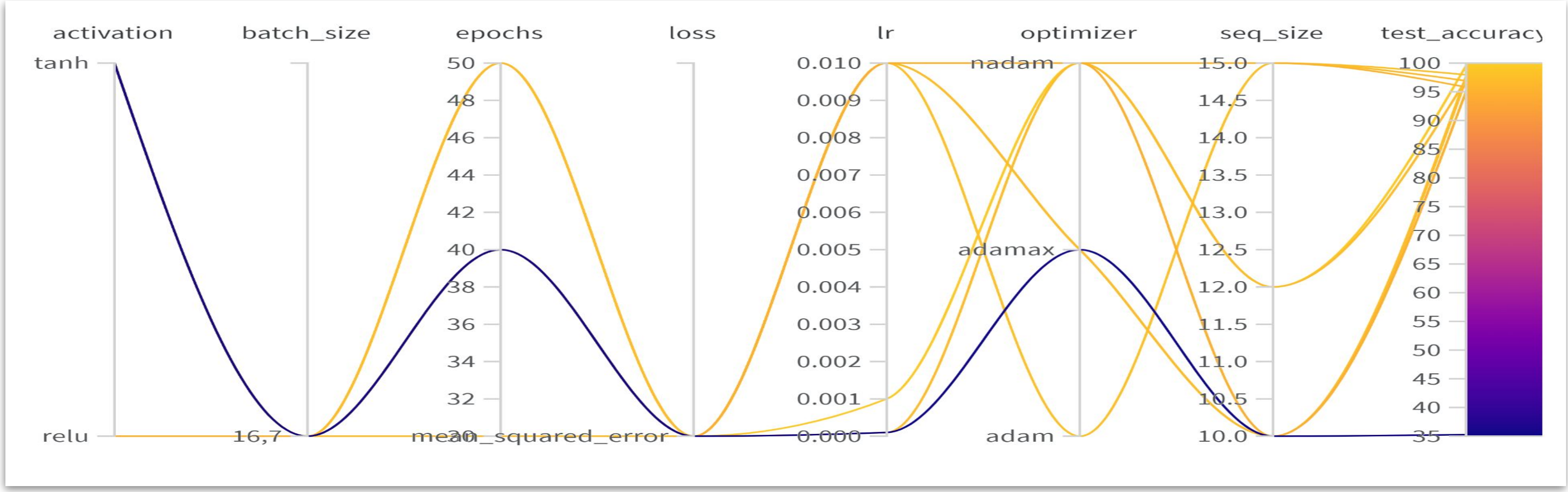
Hyperparameter combinations for RNN and LSTM

1. epochs - [30, **40**, 50],
2. learning rate - [**0.0001**, 0.001, 0.01],
3. batch size - [**[16, 8]**, [32, 8], [64, 16]]
4. activation function - ['relu', **'tanh'**]
5. sequence size - [10, 12, **15**]
6. optimizer - ['adam', 'adamax', **'nadam'**]
7. loss - [**'mean_squared_error'**]

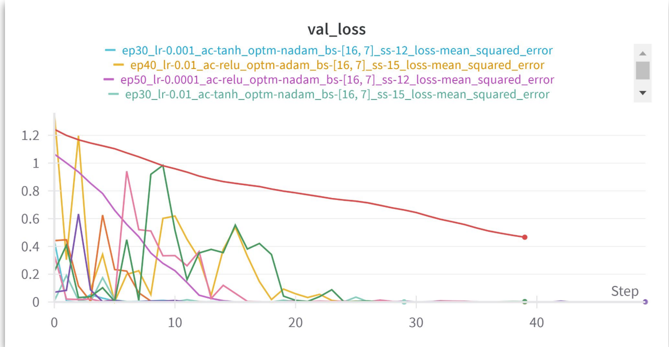
Hyperparameter combinations for Prophet

1. seasonality_mode: [**'additive'**, 'multiplicative']
2. changepoint_prior_scale: [0.01, 0.1, **0.5**]
3. seasonality_prior_scale: [1, 10, **30**]

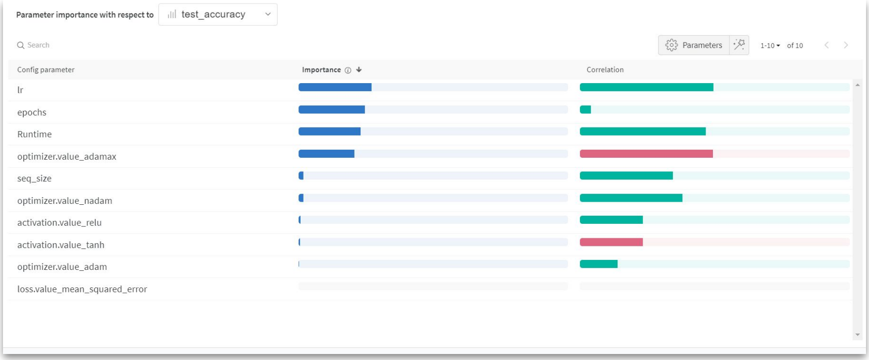
WANDB Parallel Coordinates plot for LSTM hyperparameter tuning



Loss for Validation data



Correlation matrix for LSTM sweep



Parameter importance with respect to test_accuracy

Search

Config parameter	Importance ⓘ ↓	Correlation
lr		
epochs		
Runtime		
optimizer.value_adamax		
seq_size		
optimizer.value_nadam		
activation.value_relu		
activation.value_tanh		
optimizer.value_adam		
loss.value_mean_squared_error		

Evaluation Metrics

1. Accuracy:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

2. Precision:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

3. Recall:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

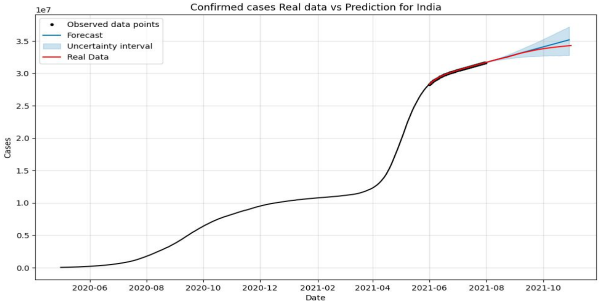
4. F1-Score:

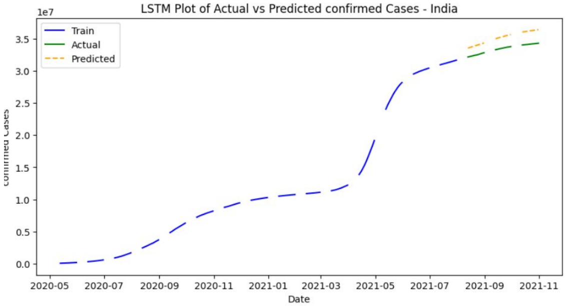
$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

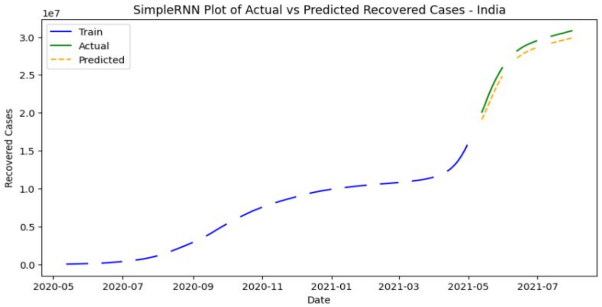
5. MAPE (Mean Absolute Percentage Error):

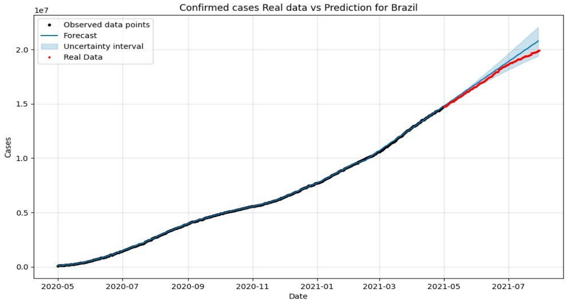
$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| \times 100\%$$

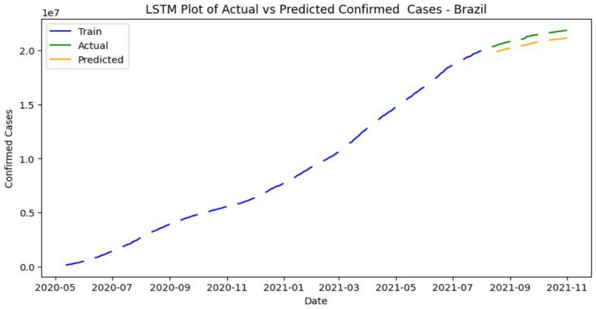
Prediction for Covid-19 Confirmed Cases

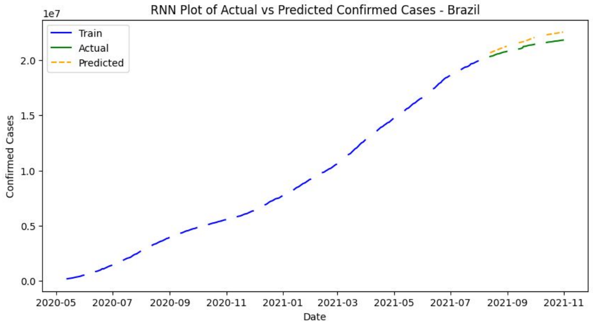




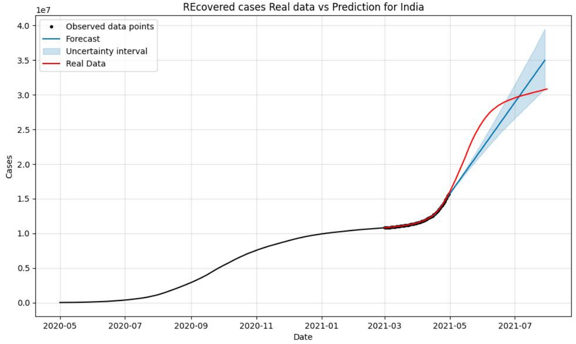


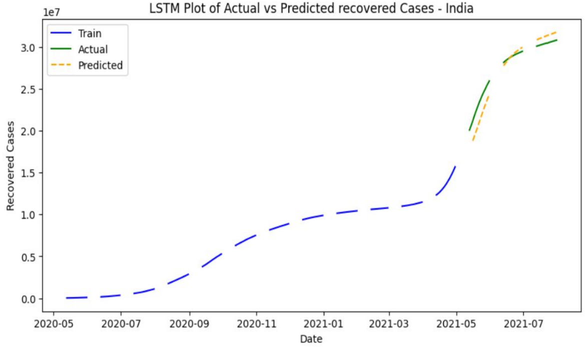


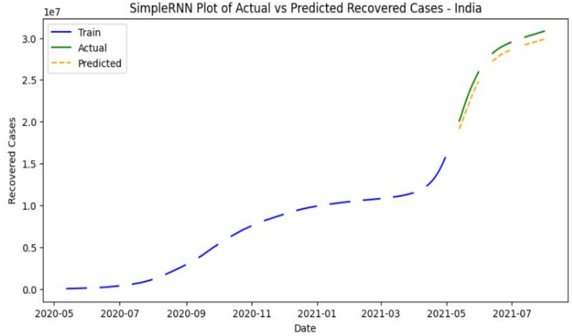


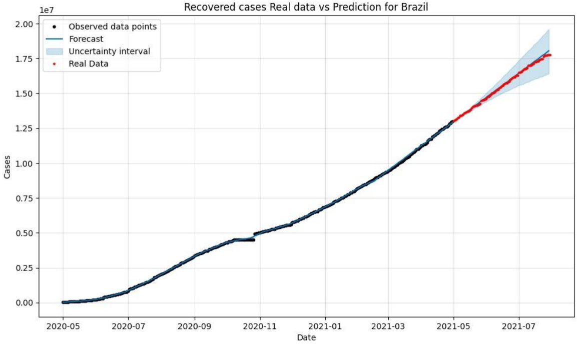


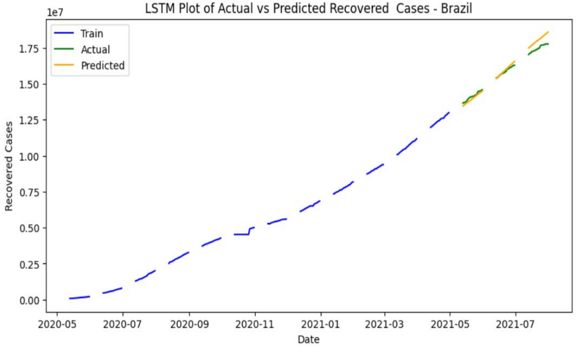
Prediction for Covid-19 Recovered Cases

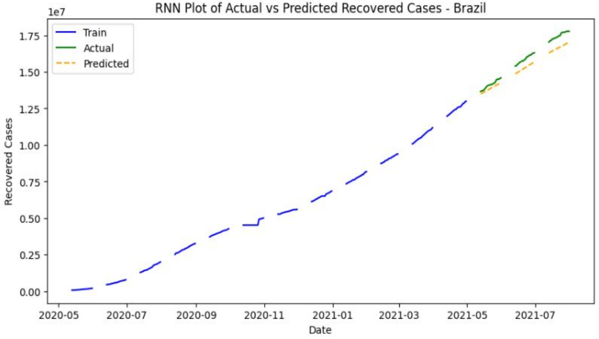

REcovered cases Real data vs Prediction for India


LSTM Plot of Actual vs Predicted recovered Cases - India


SimpleRNN Plot of Actual vs Predicted Recovered Cases - India

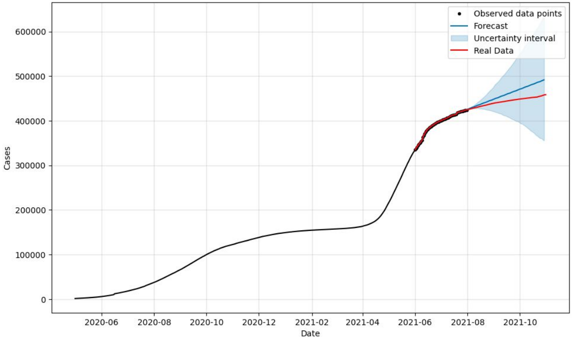

Recovered cases Real data vs Prediction for Brazil


LSTM Plot of Actual vs Predicted Recovered Cases - Brazil

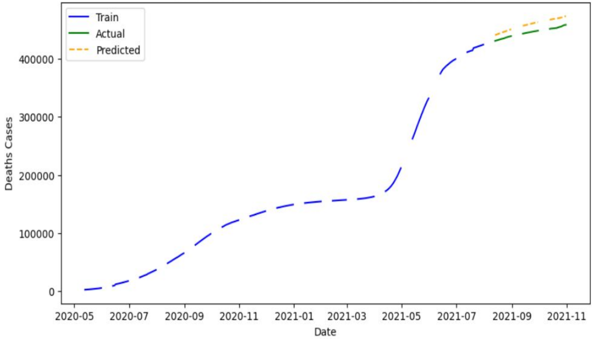

RNN Plot of Actual vs Predicted Recovered Cases - Brazil

Prediction for Covid-19 Death Cases

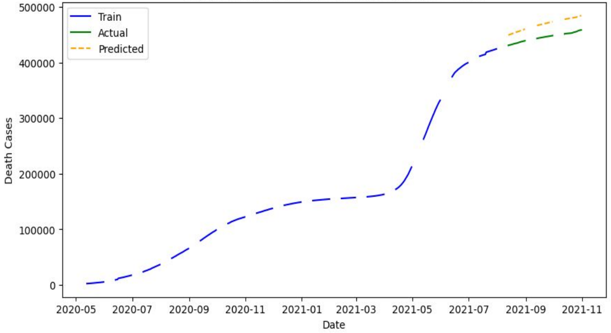
Death cases Real data vs Prediction for India



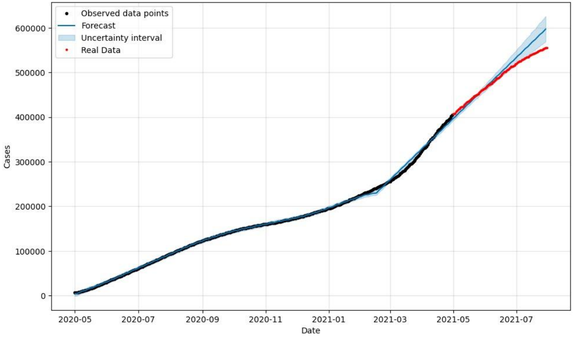
LSTM Plot of Actual vs Predicted deaths Cases - India



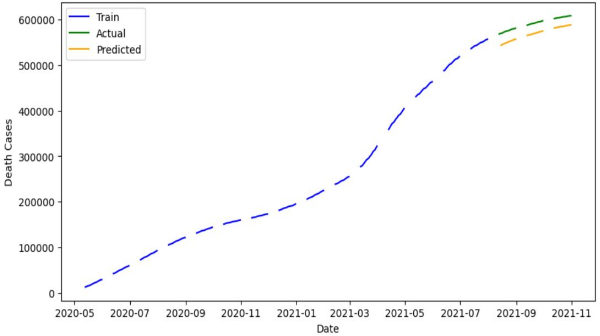
RNN Plot of Actual vs Predicted Death Cases - India



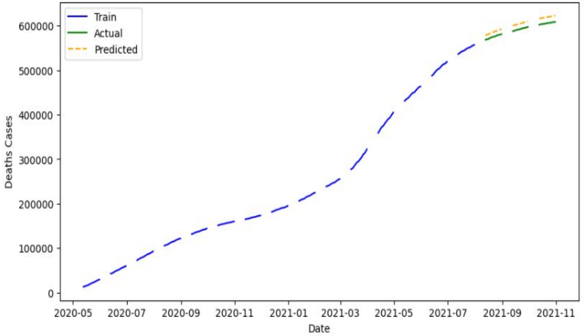
Death cases Real data vs Prediction for Brazil



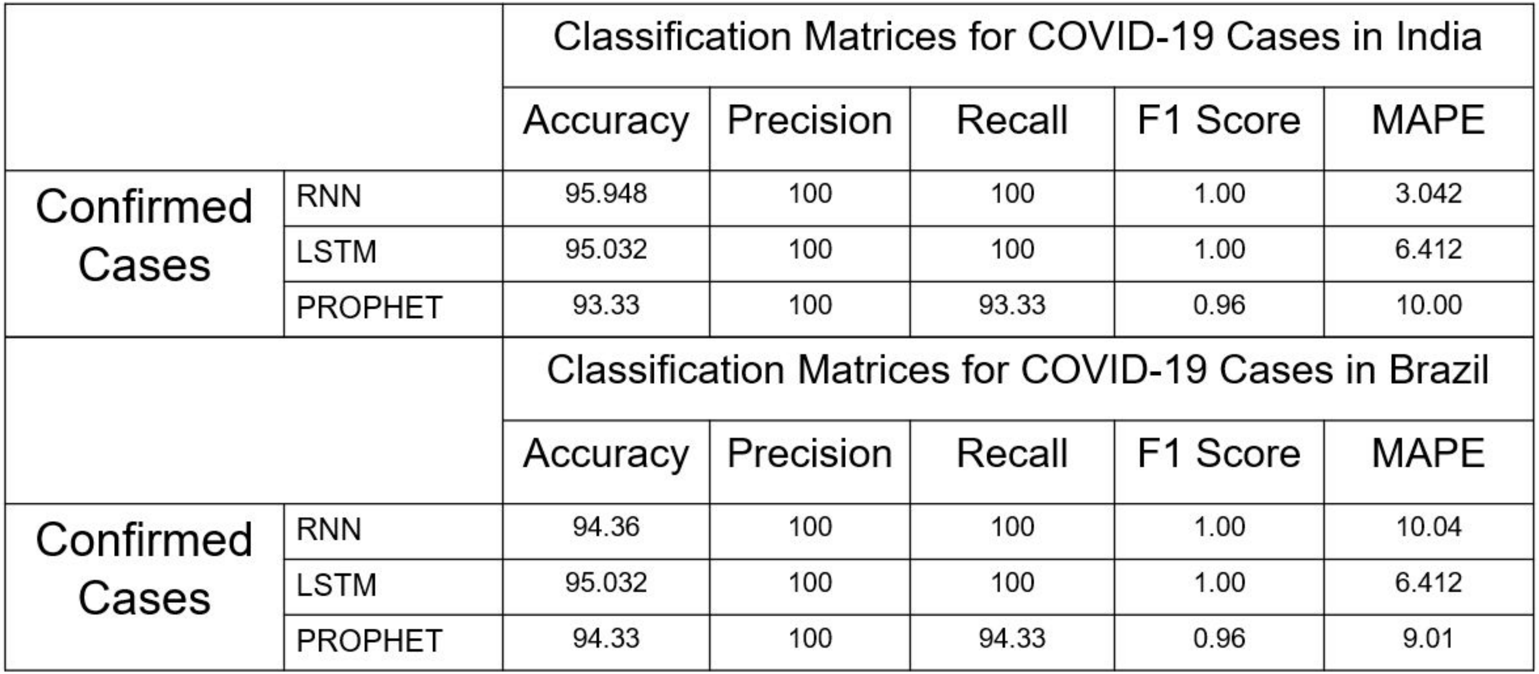
LSTM Plot of Actual vs Predicted Death Cases - Brazil



RNN Plot of Actual vs Predicted Deaths Cases - Brazil

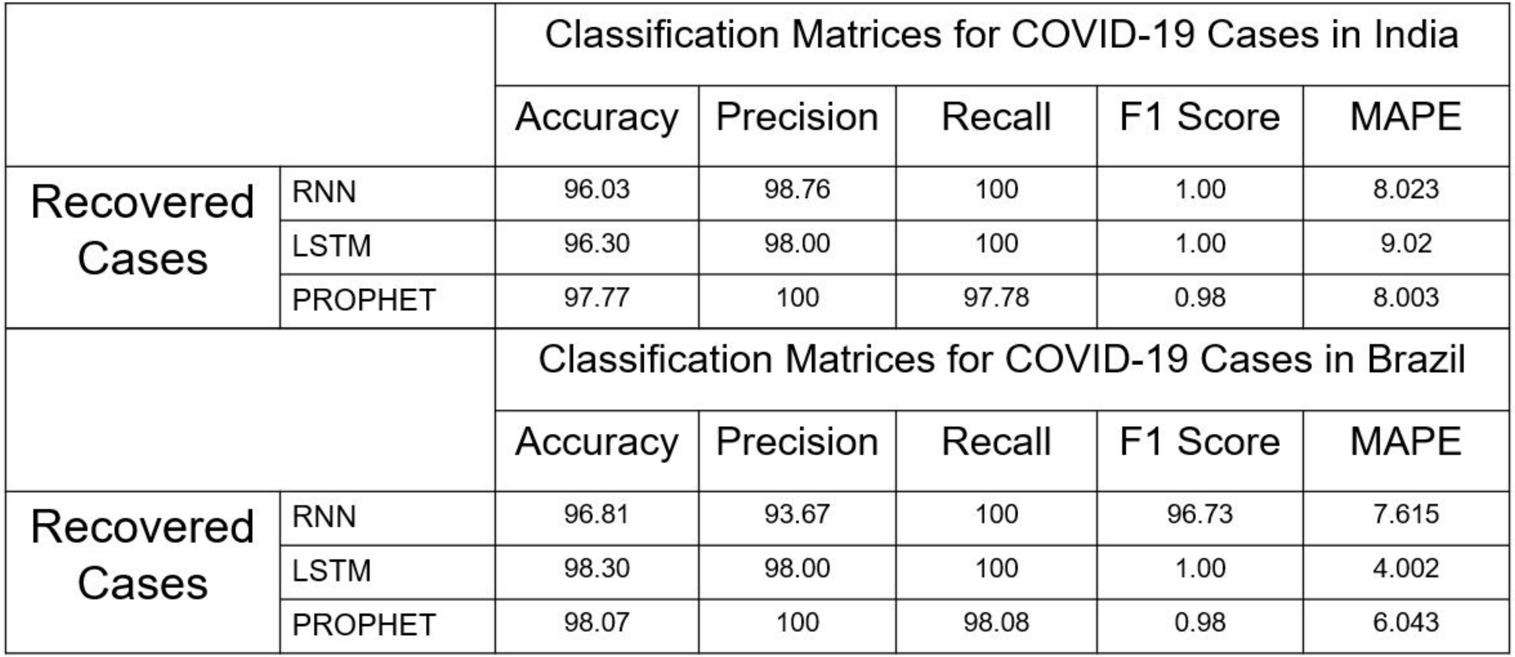


Classification matrices for Confirmed Cases



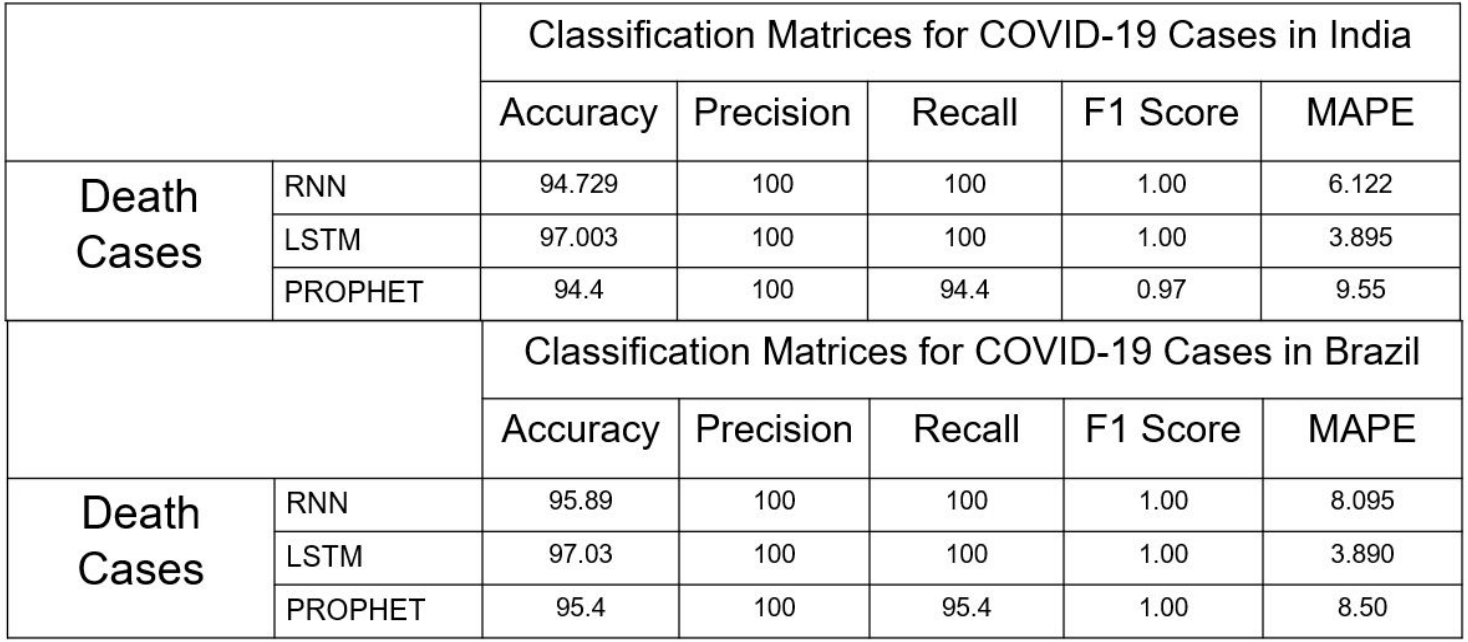
		Classification Matrices for COVID-19 Cases in India				
		Accuracy	Precision	Recall	F1 Score	MAPE
Confirmed Cases	RNN	95.948	100	100	1.00	3.042
	LSTM	95.032	100	100	1.00	6.412
	PROPHET	93.33	100	93.33	0.96	10.00
		Classification Matrices for COVID-19 Cases in Brazil				
		Accuracy	Precision	Recall	F1 Score	MAPE
Confirmed Cases	RNN	94.36	100	100	1.00	10.04
	LSTM	95.032	100	100	1.00	6.412
	PROPHET	94.33	100	94.33	0.96	9.01

Classification matrices for Recovered Cases



		Classification Matrices for COVID-19 Cases in India				
		Accuracy	Precision	Recall	F1 Score	MAPE
Recovered Cases	RNN	96.03	98.76	100	1.00	8.023
	LSTM	96.30	98.00	100	1.00	9.02
	PROPHET	97.77	100	97.78	0.98	8.003
		Classification Matrices for COVID-19 Cases in Brazil				
		Accuracy	Precision	Recall	F1 Score	MAPE
Recovered Cases	RNN	96.81	93.67	100	96.73	7.615
	LSTM	98.30	98.00	100	1.00	4.002
	PROPHET	98.07	100	98.08	0.98	6.043

Classification matrices for Death Cases



		Classification Matrices for COVID-19 Cases in India				
		Accuracy	Precision	Recall	F1 Score	MAPE
Death Cases	RNN	94.729	100	100	1.00	6.122
	LSTM	97.003	100	100	1.00	3.895
	PROPHET	94.4	100	94.4	0.97	9.55

		Classification Matrices for COVID-19 Cases in Brazil				
		Accuracy	Precision	Recall	F1 Score	MAPE
Death Cases	RNN	95.89	100	100	1.00	8.095
	LSTM	97.03	100	100	1.00	3.890
	PROPHET	95.4	100	95.4	1.00	8.50

Conclusion

- To forecast Covid-19 cases in India and Brazil, we used LSTM, RNN, and Prophet models.
- For this study we have considered confirmed, recovered and death cases for both countries.
- For this study sufficient Covid-19 data was available for accurate forecasting.
- Here we have chosen best model based on prediction accuracies and MAPE.
- Reliability of best model is satisfactory with real-time data.
- First comparative study for India-Brazil highlighting Covid-19 factors.
- This study aids in preemptive measures against Covid-19.
- Our code and data is flexible to adapt forecasting to other countries.

Main References

- ❑ Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena 404 (2020): 132306.
- ❑ Forecasting at Scale – Prophet by Facebook (https://peerj.com/preprints/3190.pdf)
- ❑ Chhabra, A., Singh, S. K., Sharma, A., Kumar, S., Gupta, B. B., Arya, V., & Chui, K. T. (2024). Sustainable and intelligent time-series models for epidemic disease forecasting and analysis. Sustainable Technology and Entrepreneurship, 3(2), 100064.
- ❑ Keshavamurthy, R., Dixon, S., Pazdernik, K. T., & Charles, L. E. (2022). Predicting infectious disease for biopreparedness and response: A systematic review of machine learning and deep learning approaches. One Health, 100439.
- ❑ Shastri, Sourabh, Kuljeet Singh, Sachin Kumar, Paramjit Kour, and Vibhakar Mansotra. "Time series forecasting of Covid-19 using deep learning models: India-USA comparative case study." Chaos, Solitons & Fractals 140 (2020): 110227.

- Zeroual, A., Harrou, F., Dairi, A., & Sun, Y. (2020). Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. Chaos, solitons & fractals, 140, 110121.

- Satrio, C. B. A., Darmawan, W., Nadia, B. U., & Hanafiah, N. (2021). Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. Procedia Computer Science, 179, 524-532.

- Abdelkader Dairi, Fouzi Harrou, Abdelhafid Zeroual, Mohamad Mazen Hittawe, Ying Sun,Comparative study of machine learning methods for COVID-19 transmission forecasting,Journal of Biomedical Informatics,Volume 118, 2021, 103791, ISSN 1532-0464

TEAM MEMBERS

ABANI SINGHA - MA23M001

ABHINAV T K - MA23M002

$\varphi_{DSB-SC}(t) \neq \Delta\varphi_R \neq \Delta\theta_R$

LOKENDRA KUMAR - MA23M008

MUHAMMED DILSHAH U - MA23M014

GHARIB MOHAMED SALEH - MA23M801

THANK YOU