SAMHAR-COVID19 HACKATHON 2020

Forecasting Pandemics and Understanding the Post Lockdown scenario

Participant Details

Prateek Jain , IIT Dharwad, jprateek2606@gmail.com

S V Praveen, IIT Dharwad, svprav999@gmail.com

Anudeep Tubati, IIT Dharwad, anudeep.tumati99@gmail.com

Ganesh Samarth C.A. , IIT Dharwad, ganesam8804@gmail.com

Mandeep Bawa , IIT Dharwad, bawa.mandeep0001@gmail.com

Indian Institute of Technology [IIT] Dharwad

PRATEEK JAIN

3rd year B.Tech, Computer Science and Engineering, IIT Dharwad. Data Analyst Intern **at National Research and Innovation Agency of Uruguay** working on AI in healthcare. **Interned at RCI , DRDO**. Published the paper titled **“Semi Automated Detection of Airbases in satellite images using Deep Convolutional Neural Networks”** at the IACC conference.

S V PRAVEEN

3rd Year B. Tech, Computer Science and Engineering, IIT Dharwad. Experienced Developer and Deep Learning Enthusiast. Mainly worked on sequential data and **text analysis using BERT**. Prior experience building end-to-end deep learning pipelines. **Google Cloud Program ,Big Data and ML certified.**

ANUDEEP TUBATI

3rd Year B.Tech, Computer Science and Engineering, IIT Dharwad. NLP and Cybersecurity Enthusiast. Interned at **NUS Singapore on summarizing reviews.** Mainly worked on text analysis and classification. **2nd Runners-up, CSAW Embedded Security Challenge India Region**, 2019.

GANESH SAMARTH

3rd Year B.Tech, Electrical Engineering, IIT Dharwad. “**Experimental Exploration of Compact Convolutional Neural Network Architectures for Non-temporal real time fire detection**” published at the ICMLA conference. Interned **at Durham University, UK** and worked on exploring state of the art techniques such as EfficientNet and Neural Architecture Search

MANDEEP BAWA

3rd Year B.Tech, Computer Science and Engineering, IIT Dharwad. Built a cloud based movie **recommender system for Movie** users. Worked on projects **Credit Card Fraud Detection** and **Subjective Feedbacks using Deep Learning** Modules. Extensively worked with RNNs and LSTMs

Problem Statement Description

- Since the advent of the COVID19 pandemic, entire countries have gone into lockdowns to prevent community spread of the virus.

- However, one of the major questions which needs to be addressed by the government is to determine when would it be appropriate to lift the lockdown to ensure minimal probability of a relapse and how the cases may rise again in different regions.

- To answer these we first understand how the cases would spread with the lockdown enforced. Our model is intended to raise awareness of the spread of the virus among the general public.

Dataset Description and Feature Extraction

We have compiled a dataset from various resources, to ensure we have the best features for our models.

1. **Covid 19 India**

It contains the covid 19 dataset for india and is regularly updated.

- Dataset per district is available
- Confirmed, Active, Recovered cases are present from the day of origin.
- Hospitals, testing labs, number of tests per day conducted.
- Zone division of states

Source: https://api.covid19india.org/

Dataset Description and Feature Extraction

2. Covid 19 Global Dataset

This dataset was used to capture the dynamics of spread in countries with unchecked lockdown and possibly use this scenario to extrapolate the nature of curve in India.

- Confirmed, Fatalities count present for all countries from the day of origin.
- Population weight for each country.

Source: https://www.kaggle.com/c/covid19-global-forecasting-week-5/data

3. Census 2011 Data -

The census data had valuable information about the districts with over 131 features. Some of them are listed below:

```
Index(['state', 'district', 'active', 'confirmed', 'deceased', 'recovered',
       'date', 'District_x', 'Classification', 'District code',
       ...
       'Power_Parity_Rs_150000_330000', 'Power_Parity_Rs_330000_425000',
       'Power_Parity_Rs_425000_545000', 'Power_Parity_Rs_330000_545000',
       'Power_Parity_Above_Rs_545000', 'Total_Power_Parity', 'Headquarters',
       'Population(2011)', 'Area', 'Density'],
      dtype='object', length=131)
```

https://censusindia.gov.in/DigitalLibrary/Archive_home.aspx

4. News Dataset

We collected the news data for various districts in india for a span of 30 days. Various websites were used to scrap the news dataset

Manual Labelling of the news as follows

- 0 -> Not related to Coronavirus.
- 1 -> Spread of Virus / Gathering of Crowds / Lockdown Protocol not followed
- 2 -> More Recovered than Confirmed / Awareness / More Ventilators

https://github.com/Mandeep3838/Covid-19-Dataset

5. List of Green, Orange and Red Zones

A good indicator of the spread of cases

RED, ORANGE, GREEN ZONES IN STATES

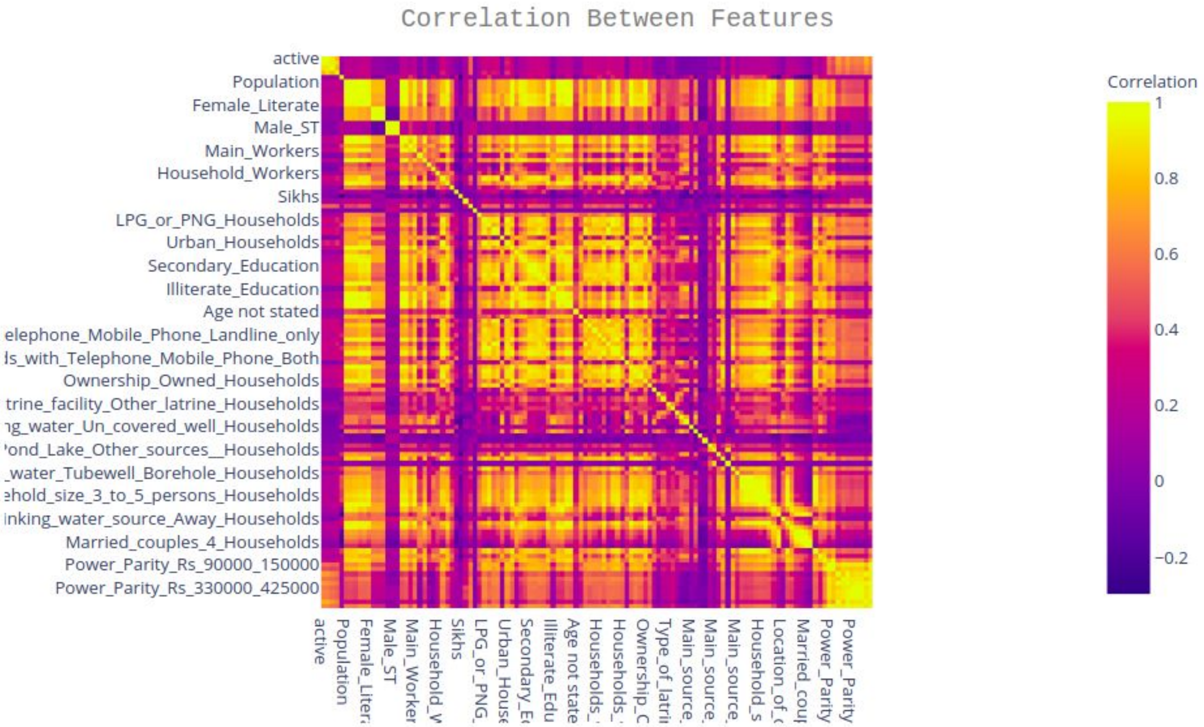
Search..

State	Red Zone	Orange Zone	Green Zone	Total
Andaman And Nicobar Islands	1	0	2	3
Andhra Pradesh	5	7	1	13
Arunachal Pradesh	0	0	25	25
Assam	0	3	30	33

Source:https://www.ndtv.com/india-news/coronavirus-full-list-of-red-orange-green-districts-in-india-2221473

Feature Selection Using Spearman Coefficient

(Some feature labels are not seen as there are too many)



Top 32 Highly Correlated Features Were Selected

Why Spearman? Non-linearity

Description of Stationary Features

Of the **32** selected features, below are the stationary ones -

1) Socio-Economic Features

    1. Purchasing Power Parity(PPP) of the district
        - Ranges of PPP are used as separate features

    2. Variety of households
        - Urban
        - having LPG Connections
        - Technically Equipped(eg:- Internet Connection, Computer, etc.)
        - Source of drinking water
        - Personal Vehicle Available

2) Demographic features

1. Area of district
2. Density of People
3. Population of district
4. Classification into COVID-19 zones:-
    - Green Zone
    - Orange Zone
    - Red Zone
5. Literacy Rate

3) Medical Related Features:-

1. Hospital Beds

(We wished more medical data for Indian Hospitals was available)

"Icon made by Freepik from www.flaticon.com"

Description of Non- Stationary Features

Generally, all features directly related to COVID-19 cases in the district fall in this category, including -

1) Confirmed Cases
2) Deceased Cases
3) Recovered Cases
4) Active Cases

Source:https://www.moneycontrol.com/news/trends/health-trends/coronavirus-cases-death-count-state-wise-tally-may-15-latest-news-today-maharashtra-most-affected-5269411.html

Challenges faced during implementation

- Availability of Less Data
  - Manually compiling data from various sources and cleaning it was a real challenge.
  - We had really hoped to train crowd interaction models, although access to real CCTV footage, proved to be a challenge.
- Lack of clean data
  - Most of the COVID-19 news data we were planning to use had news from other domains as well which made it difficult to be used
- Predictions on Non-Stationary Data
  - Time series data are known to perform poorly on non-stationary data.

Models Considered

- **SIR Model**

  This mathematical model divide population into three categories:

  - S -> Susceptibles
  - I -> Infectives
  - R -> Removed

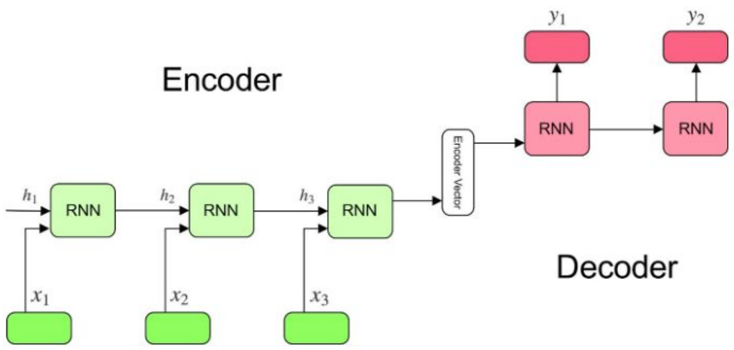
There is currently lot of literature based on SIR models and while most of them have proved to show good results none of them have managed to provide exceptional results as simple S-I-R model is not complex enough to capture the subtleties of many infectious disease outbreaks. In this project, we explore an alternative model that has proved its worth over the years at predicting time series data, ie, **sequence to sequence models.**

Source: https://theprint.in/science/how-experts-are-using-maths-to-stay-ahead-of-the-coronavirus/388745/

Models Considered

- **Sequence to Sequence Model**

A sequence to sequence model makes use of an encoder - decoder architecture to map an input sequence with an output sequence. A key advantage is its ability to map inputs and outputs of different lengths.



Encoder-decoder sequence to sequence model

https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346

Existing literature shows the growing trend of using sequence to sequence models for time series forecasting. Refer links below -

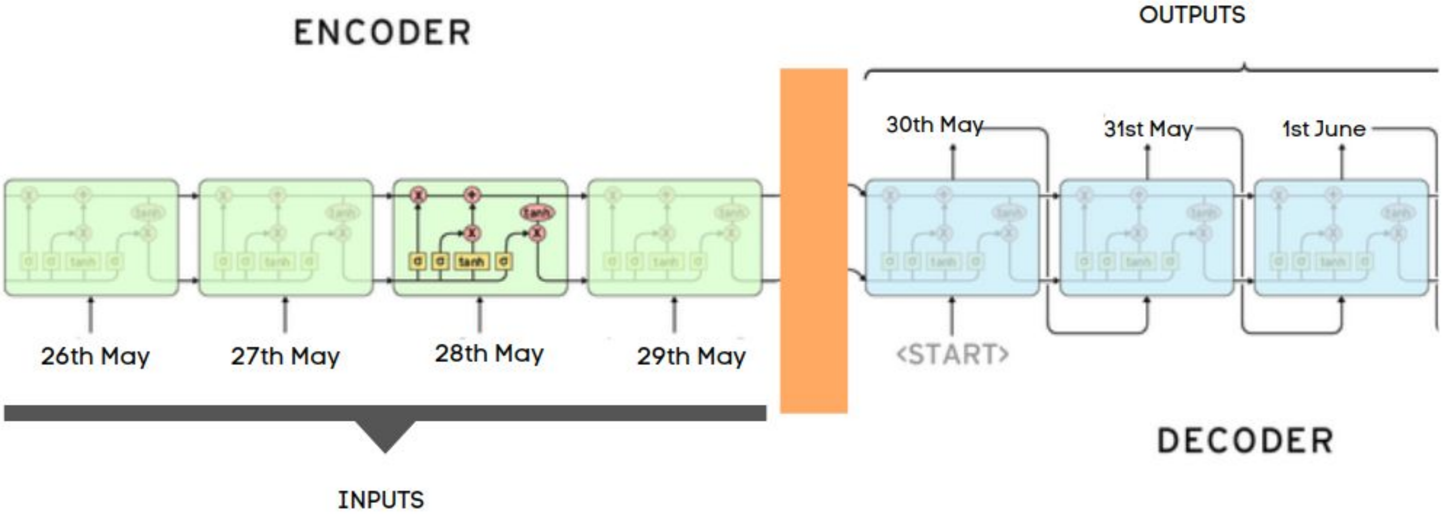
http://proceedings.mlr.press/v89/mariet19a/mariet19a.pdf

https://www.researchgate.net/publication/325075449_Foundations_of_Sequence-to-Sequence_Modeling_for_Time_Series

SOLUTION DESCRIPTION

Solution Approach

Seq 2 Seq Model



Modified: https://github.com/pranoyr/seq-to-seq

Seq 2 Seq Model Detailed

This model consists of 3 parts

Encoder

- A stack of several recurrent units (LSTM or GRU cells for better performance) where each accepts a single element of the input sequence, collects information for that element and propagates it forward.
- In our model, encoder encodes the hidden trend of the past few days.

Intermediate (encoder) vector

- This vector aims to encapsulate the information for all input elements in order to help the decoder make accurate predictions.
- It acts as the initial hidden state of the decoder part of the model.

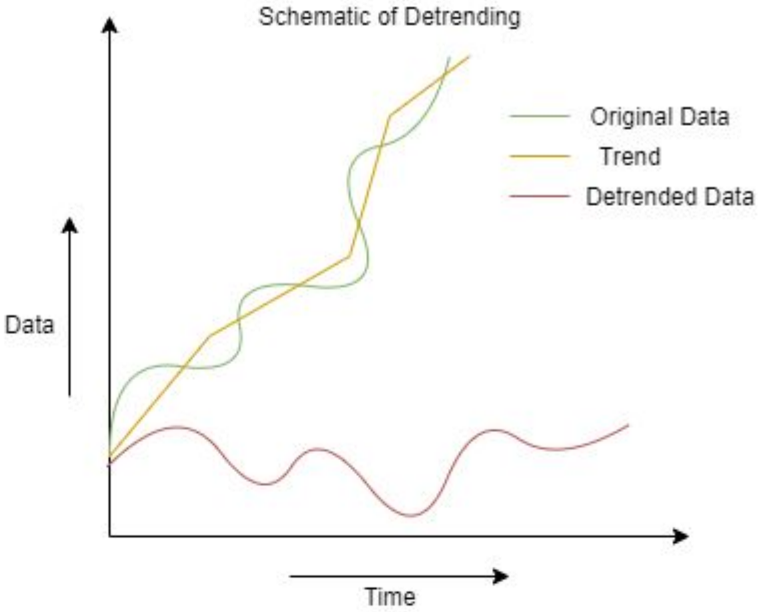
Decoder

- A stack of several recurrent units where each predicts an output *y_t* at a time step *t*.
- Each recurrent unit accepts a hidden state from the previous unit and produces and output as well as its own hidden state.

Source: https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346

Detrending to Improve Model Accuracy

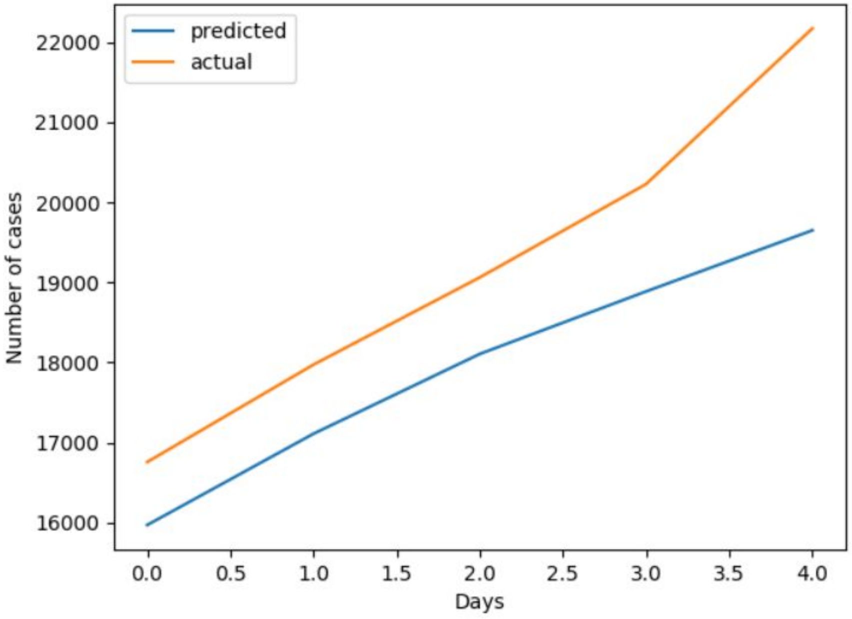
- Major reason for difficulty in predicting was non-stationarity of data.
- We performed windowed detrending using piecewise linear functions.
- Detrending converts Non Stationary Series to approx. Stationary Series
- Literature suggests that time series model learns stationary data much better than non-stationary data [1].
- Results on following slides will prove our point



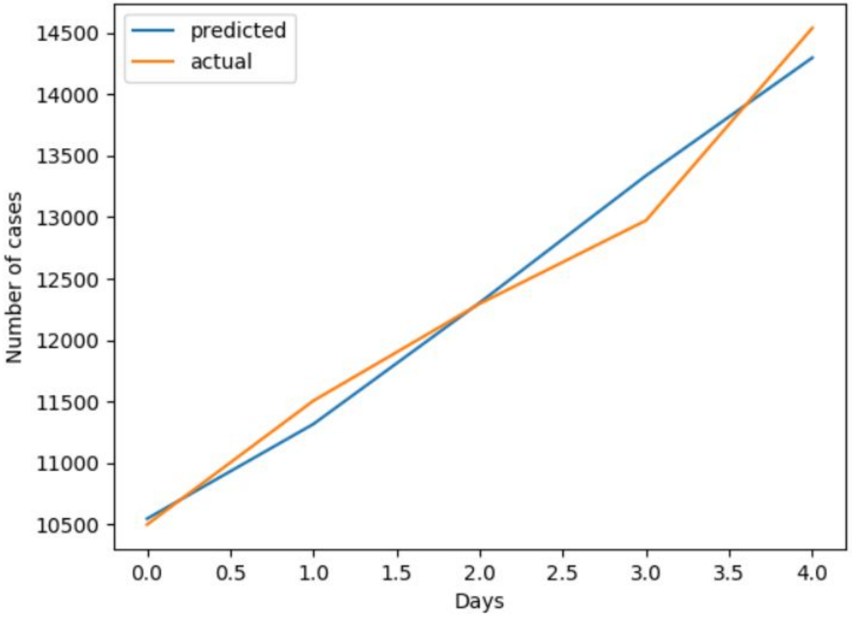
1. arXiv:1302.6613

Detrending Results Comparison (5 days)

Without Detrending

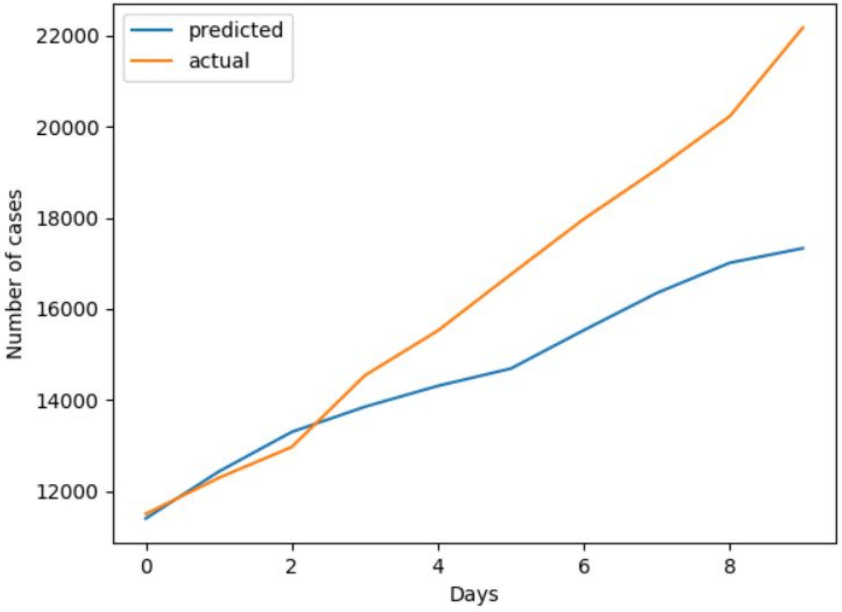


With Detrending

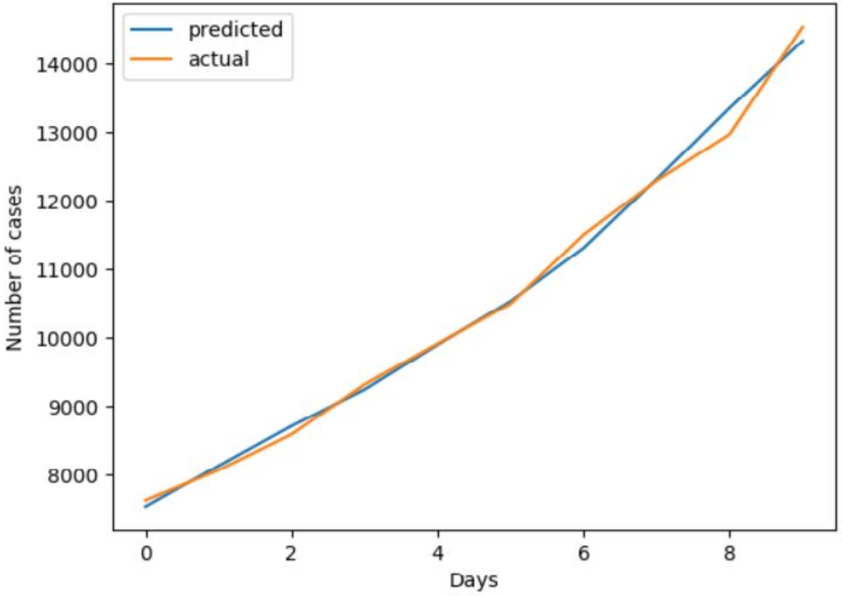


Detrending Results Comparison (10 days)

Without Detrending



With Detrending



Current Approach VS Alternatives (Accuracies)

* Our Model

Sl No.	Model	RMSE(Days)
1.	*** Sequence to sequence model**	22.857
2.	Bi-LSTMs	33.112
3.	Stacked LSTMS	43.70
4.	Transformers	102.37

Results!

Demo Link here: http://cov19-predictor-delta.herokuapp.com/

CoVID-19 Predictor India

CDAC SAMHAR CoVID-19 Hackathon, Team Delta

Showing Predictions For Chennai

S. NO.	DATE	# CASES
1	15-05-2020	4758
2	16-05-2020	5244
3	17-05-2020	5706

Performance Numbers

CPU (Min Requirements)

7th Generation Intel® Core™ i5 Processors, i5-7200U, 2.5GHz

GPU

On Tesla V100-SXM2 (32 GB)

Training time on **India** dataset per district **22 sec**
Training time on **India** dataset unified model **171 minutes**
Training time on **USA** dataset **180 minutes**

Inference Time

Prediction time for any model trained **less than 2 sec**

Practical and Implementable Nature of Solution

Refrained from using features not available easily

Scalable, can be customized for a single district to a whole state

Compact Network, can be used on any machine

Heavily Inspired from Results available in Literature

Evolves Continuously to adapt to latest trend through online learning

Less Number of Parameters, can be easily trained on embedded systems

Future Work

- Integrate BERT Model - Manually label the news dataset, and use it to model the seeming unpredictability of the spread of the disease.

- Generate Crowd Interaction Scores - With access to CCTV Footage, we would be able to generate crowd interaction features that would increase our model accuracy.