Hack4Resilience: Nowcasting Covid Hotspots in an Indian city

Team: *The Hotspotters* **(India)**

Life around the world has been disrupted by the Covid-19 pandemic. The situation is no different in India, where Government authorities, hospitals, businesses and citizens struggle to deal with the fallout of a public health emergency, the magnitude of which has not been seen in recent memory.

With the objective of not only recovering from the crisis, but rebuilding more resilient societies, the World Bank and partners have launched the Hack4Resilience challenge in which our team (*The Hotspotters*) participated. Under this program, teams from around the world work on the unique challenges faced by Government and other stakeholders in managing situations arising out of the Covid crisis.

The Challenge

Hyderabad, the capital of the Indian State of Telangana, covers an area of 625 square kilometers and is the fourth most populous city in the country. With almost 10 million people in the greater Metropolitan region, the area is seeing a rising number of Covid19 cases which the authorities have been trying to control by locking down selected "containment zones".

Our team's challenge, which was posed by the Emerging Technologies Wing of the Government of Telangana, was "**forecasting new hotspots and rate of transmission.. using mobility data or other proxies.. before the cases are confirmed by tests**". How may we use computer models and datasets including mobility data from Facebook, network providers etc. to predict emerging hotspots in Hyderabad that can then be confirmed by targeted testing?

The Design Process

We were provided excellent online design tools and mentorship from Hack4Resilience partner WorldStartup, based on a synthesis of design thinking, the lean startup approach and others. The design process enabled a detailed examination of the problem-space, from which we realized that it is not just the Department of Health that would be under pressure from the Covid situation. Other entities, including the Department of Administration, the Department of Labour, the Department of Police, and various stakeholders like private businesses, factories, schools and others would all need coordination and support to adapt to the new normal. In other words, in building resilience in response to Covid, we must avoid siloed strategies, and consider the systems perspective.

The "pain points" common to many of these stakeholders were found to include incomplete information, and difficulties in coordination (due to the structure of institutions, for example).

These manifested in various ways depending on the stakeholder. For example, the police might be required to check and enforce lockdown in various containment zones, but might lack the specific information or procedures on how to execute this effectively. Interestingly, we found that value propositions can be crafted across stakeholders, in a sort of platform model fulfilling these and other needs, and using a technological approach underpinned by computer modelling and deployed through mobile and web tools.

We then completed the design process, narrowing down to the exact constraints of the challenge statement. We decided to create a **lo-fi prototype** (mockup below) augmented by a functional backend computer model to demonstrate proof-of-concept. These would be used to get feedback from potential users to drive further design stages, determine data requirements etc. The **intended users** of the solution are **officials from the Department of Health** in Hyderabad city.
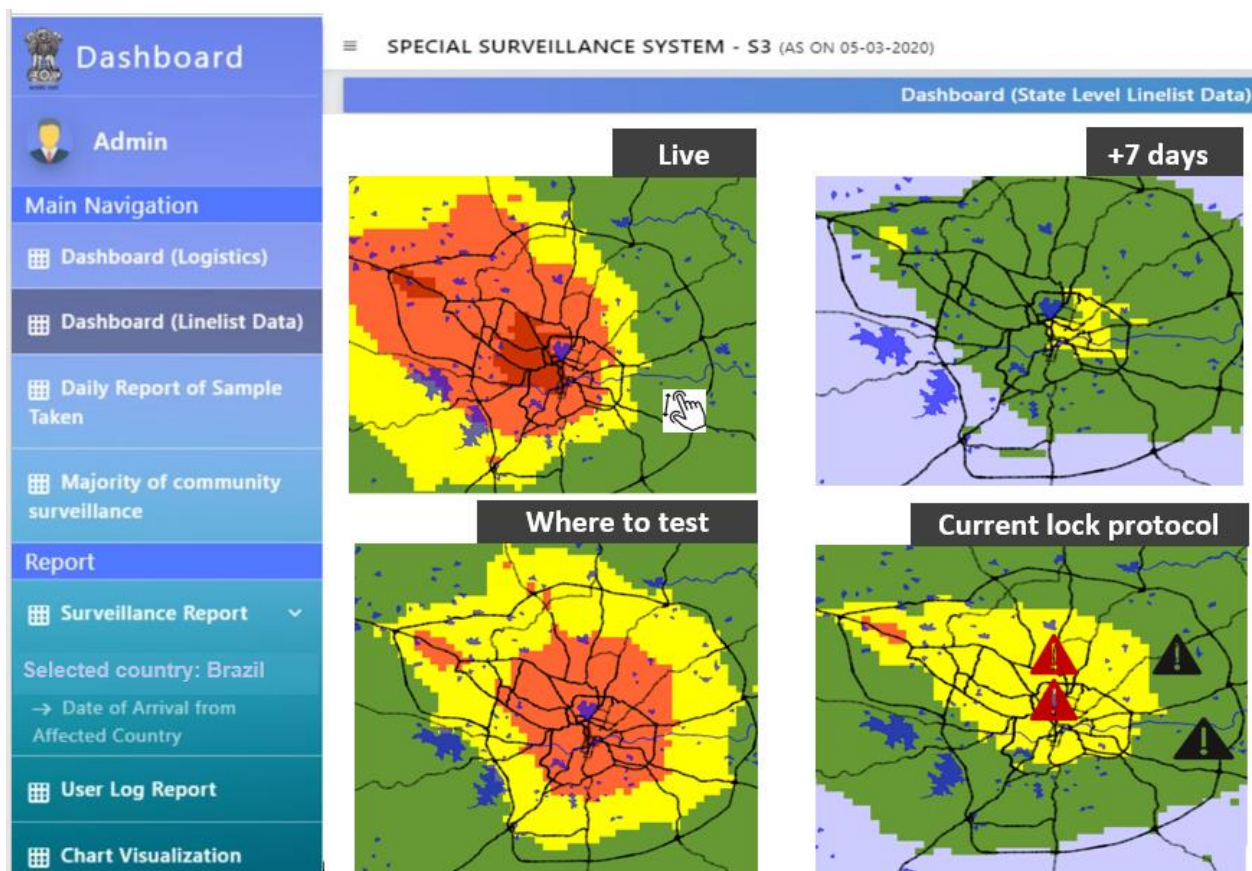

Figure 1: lo-fi prototype – mockup of main dashboard for Department of Health officials

In this dashboard, 4 "heatmaps" of the city's hotspots are available.

- **"Live"** shows where the model predicts Covid hotspots to have emerged, on the basis of the latest available data in conjunction with the underlying computer model.

- "**Where to test**" helps prioritize testing based on the new hotspots and available resources. Support will also be provided in terms of "how many" people to test, using representative random sample sizes calculated for the cluster.

- "**Current lock protocol**" is a display where officials can *input* their preferred protocol for lockdown in a granular fashion across zones in a city.

- "**+7 days**" shows the computer model's estimation of cases in the city 7 days later, if the currently selected lockdown protocol is applied from now onwards. Of course the actual time delay could be selectable by the user.

(It was decided to leave other ideas, such as area-wise estimation of resources required to deal with cases, etc, out of the current proposal as it is out of scope of the challenge definition.)

The Heart of the Solution: Building a Network SIR Model

In order to generate the displays in *Figure 1*, it is obvious that some type of computer modelling is required. One of the basic building blocks available is the SIR Model. Here S is the *susceptible* population; I is the *Infected* population, and R is the *Removed* population (either through death or immunity, they no longer contribute to the immediate spread of infection.)

One of our team members has developed and published a simple *network* SIR model[1]– where several individual SIR nodes are connected together, say by transport linkage such as air travel or roads. This allows us to model the case of a city like Hyderabad, where each cluster in the city can be represented by an SIR node. If the "connectivity" (i.e. travel) between the nodes is known or can be estimated, the model can be used to predict the spread of Covid in the city.
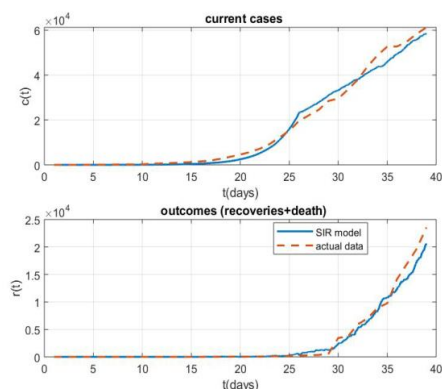
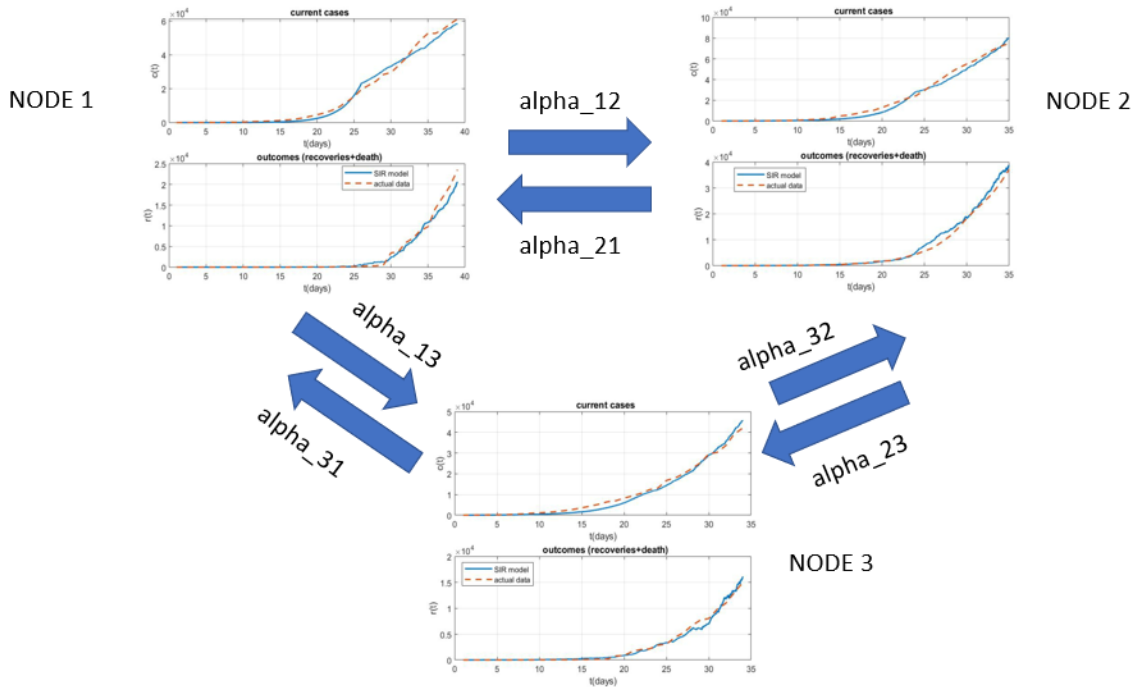*Figure 2*: A single-node SIR model's predictions. This is an example run from German data[1]

Figure 3: Linking SIR nodes to form a simple network. The connectivity factors are marked by *alpha_ij*

Figure 3 shows a *network* that is created by combining SIR nodes 1, 2 & 3. Now, the **time-evolution** of Susceptible, Infected and Removed populations in each node is **no longer independent** of the other nodes. There is a **"connectivity factor"** *alpha* that has been included between each pair of nodes in the system. This factor is defined to vary between 0 & 1 and is meant to represent the "mobility between nodes". For example, *alpha_23*=0 means that there is *no mobility* from node 2 to node 3 – and hence no conversion of node 3's susceptible by contact with node 2's infected. Setting *alpha_23=1* would imply complete transport of population of node 2 to node 3.

Deriving the Connectivity Factors alpha_ij

The factor *alpha_ij* represents the possibility for the susceptible population in node *i* to become infected by travelling to node *j* and coming into contact with the infected population there. This factor varies between 0 and 1. If *alpha_ij* is 0, it means that there is no infection of node 1's (susceptible) population by mixing through mobility with node 2's (infected) population. If *alpha_ij* is 1, it means that there is complete mixing between node 1 and node 2's population – the susceptible population of node 1 will be completely exposed to node 2's infected

population as a result of mobility. Of course, the actual disease spread will depend on the dynamics of Covid19 (i.e. the *beta* factor in the node(s) – see technical note).

Figure 4: Adjacency matrix computed by our approach. The boxes are colour-coded in proportion to mobility.

There is no way of knowing the *alpha_ij*'s for sure. However, we can form a daily estimate by using mobility data (such as that provided by mobile operators or from platforms like Google or Facebook). For example, the Quadrant Asia-Pacific Data Alliance provides a dataset where 3 fields are of interest – *Latitude-Longitude; Unique ID for each mobile device; Timestamp*. For a particular mobile device, an activity is determined id, within a timeframe, the change in *Latitude-Longitude* is significant.

We then sum up the number of all such smart-phone devices which have commuted through different areas in order to compute the *adjacency matrix* of the model, which can be be updated in real time as new data comes in. This information will be utilized to construct the *alpha_ij* which our model uses to predict hotspots, but it has other potential uses as well – such as controlling traffic to/from certain areas.

There are various possible approaches to making these *alpha_ij* more accurate. The basic method, as mentioned, is to compute these factors from mobility data. However, a more refined model could also incorporate the geographical connectedness of different wards; factoring in the presence of roads; incorporate multiple estimates from different datasets, etc. Finally, we aim to use machine learning in our full solution in order to correctly ground-truth these factors using a range of datasets.

Building the Full Solution for an Indian City

We have built a proof-of-concept of the solution for Hack4Resilience hackathon, with various components implemented to the degree possible. These include a multinode network SIR model (coded in MATLAB), code for computing connectivity factors *alpha_ij* from mobility data, and client-side code for rendering the main components of the main dashboard.

The challenge definition was to predict emerging hotspots in Hyderabad city which could then be confirmed by testing. In order for us to move from our current prototype to a system which is capable of doing this, a more complete model reflecting actual areas/clusters in Hyderabad needs to be built.

This requires more spatially disaggregated Covid19 data, along with time history to be provided. Ideally, the data – on number cases, with timestamps – would be provided disaggregated at the level of ward or less. (There are upto 200 wards in the Greater Hyderabad Metropolitan region). This would allow a model with up to 200 or more nodes (population clusters) to be set up. The displays in *Figure 1* would be granular to the same level as the model structure.

Machine Learning to Ground-truth the Model

With such a complex model with hundreds of nodes running, it is natural to enquire after the accuracy or validity of the model. This can be framed as a Machine Learning (ML) challenge as outlined in *Figure 5*.
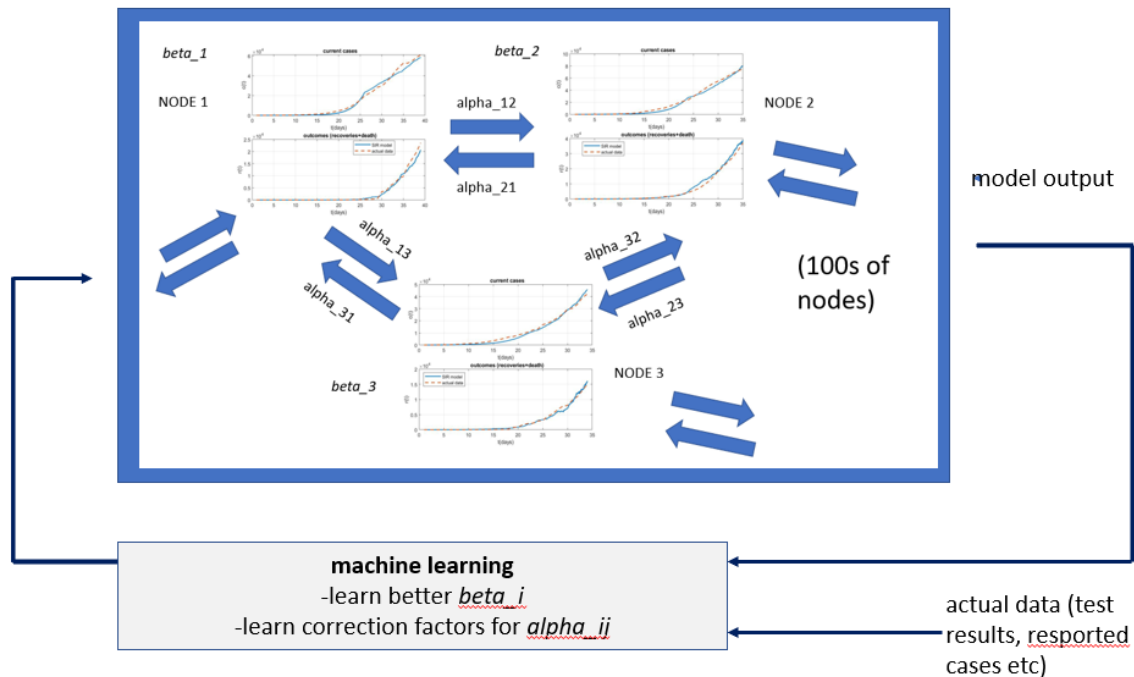
Figure 5 – Using machine learning to improve the model as time progresses

In the setup of *Figure 5,* there are now hundreds of nodes representing different areas/clusters comprising Hyderabad city. (There can also be special nodes to represent migrant populations). The connectivity factors, *alpha*, as before are derived from mobility data. Further, in *Figure 5*, a more fine-grained approach has been used, with each node having its own *beta_i*. (For example, in a densely populated residential area, the *beta* might be different than in an industrial zone).

The system is faced with the following challenge - as new data comes in regarding occurrence of cases, results of testing, etc – can the model "compare" it's predictions with the actual data, and then adjust itself by learning closer approximations to the true, underlying parameters so that there is a better fit with reality?

There are a number of ways to approach this problem, including using techniques of feedback learning (from computer science) and techniques derived from the theory of optimal control. One of the team members (Ashutosh) has already demonstrated **"learning of parameters" in the single-node model**[1]

***If* supplied appropriate data (including time-series Covid data with appropriate spatial resolution), then we can extend this Machine Learning approach to cover the whole network of connected clusters representing Hyderabad city.** The machine learning algorithm must also be provided data high-resolution historical data of when, and where, lockdown was applied precisely.

Technical notes

1. Commented **MATLAB Code** is present in our team's GitHub for the single node, two-node and multi-node (network) SIR models. A stub has been created for "trivial" machine learning. This can be expanded in the next version of the model provided adequate data is provided.

2. Each SIR node in the network model also possesses an intrinsic parameter *beta_i*, representing the growth rate of the infection. It is a "lumped parameter" which depends on the "exposure factor" in the node (cluster) among other things. Different clusters can certainly have different *beta* values, and this is illustrated in *Figure 5*. For the sake of convenience, we have currently modelled a homogenous *beta* into our simulations, based on India data.

3. In order for the adjacency matrix computation to return a non-trivial (i.e. non-Identity Matrix) result, city-level mobility data of sufficient spatial disaggregation (e.g. at the ward level or lower) needs to be provided. We were not able to find this in the provided datasets or on Quadrant. Therefore, we have currently implemented only rendering of a fictitious adjacency matrix for proof-of-concept. Of course, this could be upgraded given appropriate data availability.

References

[1] Simha, Ashutosh, R. Venkatesha Prasad, and Sujay Narayana. "A simple stochastic SIR model for covid 19 infection dynamics for karnataka: Learning from europe." *arXiv preprint arXiv:2003.11920* (2020).

Available online here: https://arxiv.org/abs/2003.11920

About the Team

Team Hotspotters is –

Hari Dilip Kumar – Team Lead & Problem Solver in Sustainable Development
Shailaja Sampath – PhD student at Arizona State University & Hacker
Ashutosh Simha –Researcher in Control theory @ Talinn University of Technology, Estonia
Sachin Gattu – MBA Student at Politecnico di Milano- Designer & Researcher