Analyzing the effectiveness of using Early Biological Threat Detection on preventing the spread of contagious diseases using Multi-agent Based Modelling

Ramos Peter, Sarmiento Joel

October 2022

1 Overview

- Diseases which have the ability to transfer from one person to another can be destructive if not immediately contained - especially for diseases wherein a cure is not yet immediately available. To be able to lessen the spread of such diseases, people have come to use Non-pharmaceutical Interventions (NPI) such as lockdowns. We aim to study the effect of using Early Biological Threat Detection as an additional Non-pharmaceutical Intervention when preventing the spread of such diseases using Multi-agent Based Modelling.

2 The Importance of Non-Pharmaceutical Interventions (NPIs)

- A study conducted by Spinelli et al., analyses the importance of non-pharmaceutical interventions on preventing the transmission of severe acute respiratory syndrome coronavirus 2 or SARS-CoV-2, by reducing the the dose of viral particles from an infected source over time, known as the viral inoculum, when there is exposure to an infectious source. According to Spinelli et al., data from physical sciences research suggest that several non-pharmaceutical interventions such as masks protect the wearer by filtering virus from external sources, and others by reducing expulsion of virus by the wearer. Social distancing, handwashing, and improved ventilation also reduce the exposure amount of viral particles from an infectious source. [15]

NPIs are useful when suppressing the spread of such diseases especially in the case of new diseases where cures are still non-existent. In the case of SARS-Cov-2, Spinelli et al. also argued that even as safe and effective vaccines are being rolled out, non-pharmaceutical interventions will continue to play an essential role in suppressing SARS-CoV-2 transmission until equitable and widespread vaccine administration has been completed[15].

3 What is Lockdown and is it effective?

- Amidst the covid pandemic, countries have taken very strict restrictions such as vacation for schools, working from home, quarantine for regions with high number of cases, and most importantly, lockdown to slow down the COVID 19 outbreak [1].

Generally, lockdowns are non-pharmaceutical interventions encompassing stay-at-home orders, curfews, quarantines and similar societal restrictions, but

the implementation was different for some countries. Restrictions that were applied also lasted for different duration depending on the impact on the country's public. The lockdown has also created the ground for renewal of the environment, especially with the closure of factories and the reduction of both private and public transportation vehicles used. COVID-19 increased the air quality in many parts of the world with the lockdown imposed during the pandemic process. Additionally, due to the lockdown, economic activities have stopped, reducing the overall carbon emissions. [25]

3.1 Impacts of Lockdown in different countries

In response to the COVID-19 pandemic, various lockdown protocols have been implemented in most European countries. One such country is France, where a national lockdown was implemented on March 17, 2020. The implementation of National Lockdown protocols were disputed as some argued that cases were mostly concentrated in two out of 13 regions and some question the timing of the lockdown as the SARS-CoV-2 was naturally about to reach its epidemic peak. Nevertheless, the lockdown was implemented that resulted in 12 of the 13 regions experiencing a peak in hospital admissions on an average of 11 days, leading to a synchronized peak and shortening and alleviating the damage done by the disease to one period of time. Furthermore, enacting the lockdown protocols prevented over-admissions on some local ICUs, which could have meant that not everyone would get treatment or confined. The lockdown was also successful in preventing the unrestrained spread in other regions of France as 12 of the regions came to their epidemic peak at the same time. [3]

China, on the other hand, was the first to impose the lockdown on January 23, 2020. And since then, various researches have been made regarding its impact on containing COVID-19 and how it affects a country's economy. In one

statistical study [19], researchers made series models using the daily COVID-19 cases between January 12th and March 30th 2020. They found out that there is a significant decrease in the daily cases reported in China, following the establishment of lockdown protocols and concluded that it may be used to ease the burden of COVID-19. [19] Moreover, on January 23, many countries have restricted outbound flights to China while the country is on a lockdown. One study [12] investigated the changes in the spread of COVID-19 in China and internationally following these 2 protocol changes. They found out that there is a significant increase in the doubling time, from 2 days to 4 days, and that the correlation between air traffic and the COVID-19 spread were weaker after imposing the lockdown. [12] These simple observations support results from other studies which have estimated the impact of lockdown on SARS-CoV-2 spread to be strong. However, implementing lockdown protocols also has their drawbacks, especially in the country's economy. Since China started implementing the lockdown, it has had a dramatic effect on the country's flow of people such as low number of people going to the mall and office locations, cross-city travels were less, and the flow of goods were at a very low level due to the decline of daily truck flows, as is the consumption and sale of retails goods. [4] Overall, figuring out the ideal duration of lockdown to minimize the negative effects is of utmost importance.

3.1.1 How likely are people to adhere in self-isolation and/or quarantine protocols

Among the factors of the effectiveness of lockdown measures and quarantine protocols is the adherence of the community. The higher their adherence is with lockdown measures, the more effective it is. Adherence of the community may depend on a variety of factors. One study investigated adherence in UK through a cross-sectional survey involving 2240 participants aged 18 years or

older, conducted last May 2020. Participants who reported having symptoms or having symptoms within their households within the last 7 days consists of 9.7% of the overall sample. From those same participants, 75.1% were found to be non-adherent, which are participants that have left the home within the last 24h. Non-adherence was found to be associated with not having received helped from people outside their household - having to shop for essentials, 'decreased' perceived effectiveness of government lockdown measures, 'decreased' perceived severity of COVID-19, and decreased estimates of other people from their community that are following lockdown rules. Gender was also a factor in non-adherence as men were more likely to be non-adherent, possibly due to men being out for work or shopping for non-essential goods. Overall, the adherence to government lockdown measures can be improved with better communication of knowledge to the community, consistent reminder of the risks of COVID-19, and actions that can be taken should a person come in contact with an infected/symptomatic person. [28]

Another study was also made involving 14433 German respondents to see how they would react when lockdown scenarios are extended and/or intensified. The results reported that half of the population rejected the idea of further extension and intensification of lockdown and only 20% endorses long term strategies if necessary. [10]

4 Agent Based Modeling and Simulation

4.1 Models and Simulations

In this chapter we will be discussing the definition and the different types of modeling and simulation, together with a specific Modeling method called Agent-based Modeling which is the type of model that we're going to be using

in this paper. A *model* is an abstracted logical description/replication of an object, process or event. It may exaggerate or overstate certain aspects at the expense of others but it might be useful at times depending on the model's purpose [24]. There are many different types of models but for us to have an idea, we will define five main types of models: Physical, Mathematical, Virtual, Process and Computational models.

Any model that is constructed with the semblance of the object of interest's physical characteristics is called a Physical Model. Examples of these are statues and sculptures like wax or concrete or bronze, model toys, caricatures, and the blueprints of a building. Mathematical models on the other hand uses mathematical formulas, expressions and variables to represent, analyze, make predictions or otherwise provide insight into real-world phenomena [20]. Mathematical models can take many forms including, but not limited to graphs, differential equations, economical models, statistical models and they can usually solve or give insight to problems such as recycling options, rate of spread of disease, thrill-factor of theme park rides, etc. [8] Virtual models are mathematical representations of any surface of an object that are developed using specialized software. Models that are made virtually are called 3D or three-dimensional models, though they can also be displayed as 2D images through a process called 3D rendering. What's unique about Virtual Models is that with recent technology they can also be transformed into physical models using 3d printing, CNC or computer numerical control mills, and other technologies. The fourth type of model is called a Process Model. It describes the steps that we need to follow to perform different tasks, to-do lists, flowcharts and other examples. Process Models are important in the development of many applications and are usually used and revised repeatedly until optimized. One example of this is the

instruction manuals that we find in appliances because they include diagrams and numbered instructions that we need to follow to assemble the item [16]. Lastly, Computational Modeling is the use of computers to study complex systems using mathematics, physics and computer science. A computational model is composed of numerous variables that characterize the system being studied. Computational/Computer Modeling and recent technology allows thousands of repeated *simulations* that were not possible when the idea of models came out in the 1940's because of a lack of computational power, that lets scientists and researchers adjust variables and observe the patterns from the outcomes [21].

The next term we will define is called Simulation. Simulations are methods for implementing models over a period of time, which means that a model can be considered as the basis of a simulation. Some popular modelling and simulation tools include spreadsheets, flowcharts and specialized modeling hardware and softwares such as Maya, 3d studio max, Blender, Autodesk Inventor, and Google SketchUp. Keep in mind that the distinction between modeling and simulation can sometimes be vague as they are almost always related to each other. One difference that we can distinguish between the two is how they are made. Simulations usually are imitations of real-life events/actions while models are usually representations of static objects and events but it can also represent dynamic & fluid events/processes albeit inaccurately. To understand that further, we will look into the different types of Simulations [20].

Generally, there are three types of simulation - although you can use some combinations of them as well, namely Live, Virtual and Constructive simulations. Live simulations typically involve humans and/or equipment in a setting where they would operate for real. For example, a live simulation is when soldiers do combat practices & military maneuvering in the field or when basketball players play in a scrimmage. They are called live simulations because time is

concurrent as with the real world. While a Virtual simulation typically involves a user that is operating a simulated system or a computer-configured system, where time can be in discrete steps. Flight simulators, driving simulators and shooting simulators fall in to this category. Lastly, a Constructive simulation, also called Discrete-event Simulations, is when simulated people operate simulated programs. It models the operation of a complex system as a discrete sequence sequence of well-defined events. In other words, it only qualifies as an event if there is a change in the state of the system at a specific point in time. Examples of these are traffic simulations, explosion/accident simulations and weather, wherein they will simulate an accident and how the emergency team will respond such as the police, fire department, medical care and others. Discrete event simulations can help determine rules for building/maintaining many different types of system. Furthermore, Virtual and Constructive simulations are useful especially when live resources are hard to come by, just like in an operation for example, wherein real lives might be at stake if we do a live simulation in contrast to a virtual simulation. Tools used for simulations range from simple tools such as flipbooks and storyboards of movies, to powerful software like Unity 3D and Unreal Engine [16].

4.2 Methodologies for Creating Models and Simulations

Next, we will look into the many methods for creating models depending upon the specific challenges and purposes. These methodologies are evaluated on whether the simulation meets the minimum specification, requirements and/or parameters and whether the model is accurate compared to the real world. [17].

Given all of that, let's examine some examples of the Methodologies in creating Models and Simulations. The first methodology is called Stochastic Mod-

eling, and this model estimates probability distributions of potential outcomes by allowing for random variation in one or more inputs over time. Possible outcomes and the probability of them occurring are derived from a large number of simulations known as stochastic projections. One example of this is a dice game in a casino. The rules of the dice game says that you win 100$ if you roll a 1 or a 6. The first step is to determine all the probabilities of dice rolls, and since it's a six-sided dice, you can roll either 1, 2, 3, 4, 5 or 6. The next step is to determine the event of interests, and those are if we roll a 1 or a 6 since that will give us the win. Since rolling a 1 or a 6 both have a 1/6 probability of happening, then we have a 2/6 or 1/3 chance of winning 100$. Therefore, we used stochastic modeling to estimate that our chances of winning is 1/3 of the time, which is the event of interest, and 2/3 probability of losing when we gamble with the same rules. The second methodology that we will talk about is Structural Modeling. This type of modeling usually analyzes and describes things that are within a system and how these things are related to each other. The object of interest can be a class, a subsystem, an object or anything that is part of a system being developed. For example, a computer network can be considered as a Structural Model because of how its components are related and connected. Human behavior models, on the other hand, studies the relationships between the unpredictability and complexity of the human non-logical behavior to other humans and/or their environment. It is essential for studying different psychological and sociological scenarios such as how humans might react and should they evacuate in the time of a crisis/emergency or how a driver might react on various situations that might be deemed dangerous. Note that a driving simulator or a live simulation might be used in this example. The next methodology, Phsyics-based Modeling is more of a simple one but is also essential in the observation of many real-world objects, such as machines, before

mass-production is approved. It is done by constructing dynamic models of objects and computing their capabilities, and durability, via Physical/Dynamical Simulation - simulation of systems of objects that are free to move, usually in three dimensions and are subjected to the laws of dynamics, it can also be a live/virtual simulation. This type of modeling emphasizes the importance of computing power and their role in modeling and simulation, as with the next two examples of methods of modeling [16].

The Monte-Carlo Simulation is said to leverage the brute strength of modern computers as it could involve thousands or tens of thousands of re-calculations before it is complete. A Monte-Carlo simulation is a type of stochastic model and is also a tool for risk-analysis, it works by building models of possible results - its version of stochastic projection, by substituting a range of values—a probability distribution—for any factor that has inherent uncertainty. Basically, it determines the possible outcomes of every decision and estimates risks and the impact of that risk, which allows us to make better decisions in moments of uncertainty. As we can see, this method might take a lot of time especially for inadequate computers as it uses brute force because it calculates the probability that an outcome will occur for every single decision or course of action, but with a powerful computer it can be very effective and useful [22].

Finally, we have Agent-based Modeling, which is the method of modeling and simulation that we're going to be using in this paper. Agent-based models are computer or computational simulations used to study the interactions between people, things, places, and time, with the use of agents, often people, in epidemiology that model an intelligent and autonomous behaviour usually ascribed to a set of assigned attributes. An agent with autonomy means that it can decide for itself which actions they are going to perform at a time, based in part on external environmental conditions and in part on private internal

aspects such as beliefs, desires and uncertainty. Thus, in multi-agent systems, a potential source of uncertainty for each agent is not knowing for sure what other agents will do - called "behavioral" or "strategic" uncertainty [30]. The agents also usually have a local but imperfect information and often, there are random elements that exists either among the agents or in the space to represent realism. The agents are also programmed to behave and interact with other agents and the environment in certain ways. These interactions produces a ripple effect that may differ from the effects of individual agents, without interaction, and/or between other pair of interacting agents and so on [11].

Agent-based modeling differs from traditional regression-based models, that have defined relationships between agents or variables, in that, like systems dynamics modeling, it allows for the exploration of complex systems that display non-independence of individuals and feedback loops in causal mechanisms that can lead to the improvement of the simulation over time. It is also not limited to observed data and can be used to model the counterfactual or experiments that may be impossible or unethical to conduct in the real world. However, agent-based modeling is also limited in that: the data parameters are often difficult to find in the literature and some models can be difficult to assess in terms of validity, particularly when modeling unobserved or non-factual associations [11].

One famous example of an agent-based model is Thomas Schelling's Segregation Model. The Schelling model represents a version of the world as a grid, with each cell representing a house. The houses are then occupied by two kinds of agents but 10% of the houses are empty, labeled red and blue in roughly equal numbers. At any point in time, an agent can be happy or unhappy, depending on the surrounding agents. The neighborhood of each house is the set of eight adjacent cells. In one version of the model, agents are happy if they have at

least two neighbors like themselves, and unhappy if they have one similar neighbor or none at all. The simulation proceeds by choosing an agent at random and checking to see whether it is happy or not. If so, then nothing happens; if not, the agent chooses one of the unoccupied cells at random and moves. The results show that the model segregates the red and blue agents fairly quickly, with clusters of similar agents appearing and merging over time. By the end of the simulation, most agents live in homogeneous neighborhoods. Note that this model did not aim to solve racism as it is a complex human problem, but demonstrates a possible cause of segregation, which is not that the agents were racist, since all of them would be perfectly happy even in a mixed neighborhood, but they prefer not to be greatly outnumbered by different-colored agents, so they might be considered xenophobic at worst [6]. Another example is a traffic system which consists of roads and the vehicles on that road. Each vehicle in that road would be considered as an autonomous agent, and so with ABM, you can simulate the effect of that particular autonomous agent on the system and/or simultaneously simulate the effect of all the autonomous agents or all vehicles, on the system as a whole.

There are many programs that are used for the development and implementation of agent-based models such as Netlogo, Starlogo, which is a playful verson of Netlogo designed for grade school programmers, Anylogic, Swarm, REPAST, Python, Google Code, MASON, FLAME, and many others [26] [27]. NetLogo is the highest-level platform, providing a simple yet powerful programming language, built-in graphical interfaces, and comprehensive documentation. It is designed primarily for ABMs of mobile individuals with local interactions in a grid space, but not necessarily clumsy for others. NetLogo is highly recommended, even for prototyping complex models [23].

Overall, agent-based models provide an additional tool for assessing the im-

pacts of exposures on outcomes. It is particularly useful when interrelatedness, reciprocity, such as mutual dependence and mutual influence and feedback loops are known or suspected to exist in the environment or when real world experiments are not possible [11].

5 An Agent Based Model for Rapid Biological Threat Detection

This section presents an agent-based model for rapid biological threat detection from the study of Al-Zinati et al. 2018 [18], starting with the general overview of the model, followed by the main components of the model and their interaction with each other.

However, before proceeding to discuss the model, this section first introduces the technologies that the proposed model is dependent to.

5.1 Telemonitoring and Mobile Edge Computing

The proposed model is about real time analysis of gathered patient information. There are two key objectives that this model has to meet, that is:

- To be able to gather and transmit patient information.
- To be able to analyze the data real time.

Telemonitoring patient information and mobile edge computing are technologies that would give the proposed model the ability to meet the previously enumerated objectives.

5.1.1 Telemonitoring Patient Information

Data gathering plays a vital role in surveillance systems. Traditional data gathering is impractical to use when dealing with real-time, large-scale monitoring.

In 2001, authors Negoslav D. et al proposed a method of telemonitoring in cardiology. They acknowledged the limitations of the traditional methods of using electrocardiography (ECG) machines and physical analysis and provided a more practical way to achieve such method. With the existence of smaller ECG machines that could be carried anywhere, and mobile phones which can send data wirelessly and remotely, monitoring remotely or telemonitoring with ECG became possible. A patients cardiology data is now able to be stored, sent, analyzed, and remotely monitored for diagnosis. Surveillance of a patient became more practical since data can already be accessed remotely, including location with the existence of GPS [35].

Telemonitoring will play a key role for this model, since monitored humans are expected to wear textiles with integrated sensors. These sensors are responsible for collecting different kinds of vital signs such as heart rate, temperature, etc. Each monitored human is also expected to carry a smart phone, PC, or tablet equipped with Bluetooth and GPS technology. [18]

5.1.2 Mobile Edge Computing

Cloud computing is the widely used method of data delivery and data processing of interconnected devices. As defined by the Oxford Dictionary – this is the practice of using a network of remote servers hosted on the internet to store, manage, or process data instead of a local server or a personal computing device. Figure 1 shows an image interpretation of the cloud.

While cloud computing has been effective, the Internet of Things, or IoT, is

Figure 1: The Cloud [5]

generating greater volume of data. Data transfer itself would already consume large amounts of time. "By the time the data makes its way to the cloud for analysis, the opportunity to act on it might already be gone" [5].

Imagine having a heat sensor for a certain industrial machine that helps to regulate the machine's temperature. This heat sensor generates and transfers data in real time, and once the temperature goes beyond the acceptable limit, it is required to almost instantly act on it. The time the data travels from the edge - near the heat sensor or the heat sensor itself, to the cloud for analysis is uncertain, and the opportunity to act on the problem might already be lost.

Connecting a great number of various things to the cloud is impractical. A solution for this problem is Fog computing. The fog extends the cloud to be closer to the things that produce IoT data as seen in Figure 2). Processing data near where it is generated - in the fog, can minimize gigabytes of data generated and reduce network traffic [5].

Another solution which can solve the problem on Cloud computing is with the use of Edge Computing. The difference of edge computing from fog computing is that, with fog computing, data is initially processed at the fog with the use of local area network before being sent to the cloud. Figure 3 shows

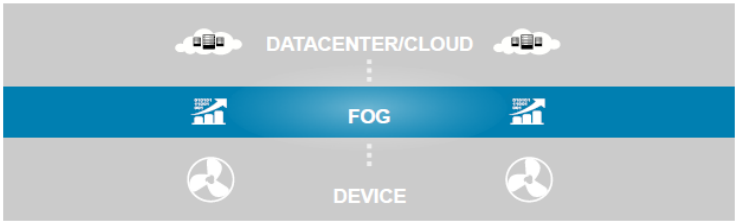


Figure 2: Fog Computing [5]

the difference between the cloud, the fog, and the edge. What edge computing features is that the computation occurs at the devices themselves, which in turn provides faster results since generated raw data is instantly processed [14].

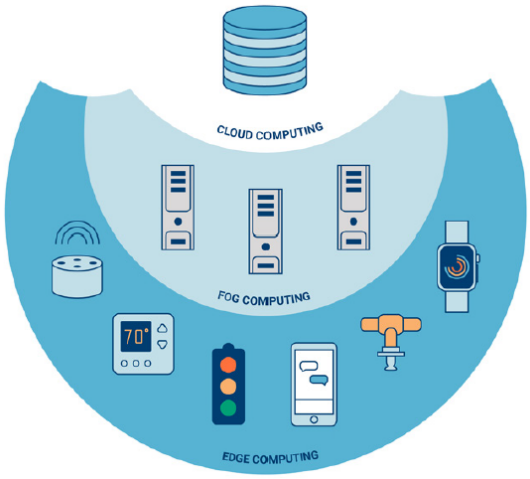


Figure 3: Cloud, Fog, and Edge [14]

In the mentioned model for this section, the researchers proposed a way to process data with Mobile Edge Computing or MEC. MEC is a type of edge computing which uses Radio Access Networks or RAN, which in turn, is more inclined to mobile subscribers. Simply put, it is an edge computing method used within a mobile network [33].

The researchers chose to use MEC in this paper since rapid biological detection needs real-time rapid responses [33]. The use of MEC provided qualities

that the researchers needed for this project such as low latency - faster data travel time and server responses, and highly efficient network operation.

5.2 General Overview

In the proposed model, monitored humans are supposed to wear equipment with a set of integrated sensors. The sensors are responsible for collecting various type of vital signals such as blood pressure, heart rate, and body temperature. Each monitored human is also expected to carry a cell phone or a tablet PC equipped with Bluetooth and GPS technology. A software agent is deployed on the mobile phone and is responsible for capturing the data provided by the sensors, performing basic data processing and analysis, and transmitting data to higher level agents for further processing.

In order to monitor large scale environment efficiently, a partitioning of the environment is proposed, each smaller parts are of equal sizes called 'cells.' Each cell is monitored by a specialized agent that is responsible for collecting and analyzing the data sent by the software agents embedded on each of the monitored humans within its vicinity. This agent is to be deployed on a MEC server rather than deploying it to the cloud due to the following reasons[33]:

- The MEC provides real-time response which is very critical in the system since the system promises rapid detection of biological threats.
- Managing local responses on the edge side reduces the overhead on the cloud layer.
- MEC agents are in close proximity with the monitored humans in their daily movements. In case a monitored human leaves the area of a cell monitored by a particular MEC agent, the monitored human should enter another area monitored by another MEC agent.

- As the collected data is processed and analyzed on the edge, then the location of the monitored human is characterized and identified.
- Privacy is maintained as the data is only processed and analyzed on the edge and only aggregated data is sent on the cloud for final analysis.

As the generated data within a cell represent a small portion of the monitored environment, it is not adequate to detect biological threats. Therefore, the environment is assigned with a meta-level macro - Super Manager agent, which is responsible for aggregating and analyzing data received from cells and responsible for detecting possible threats. The deployment of the Super Manager is suggested for this model to enhance the scalability and performance of the threat detection system.

5.3 Model Components

Figure 4 shows the model which consists of three agents of different types and responsibilities. The description of each component types is given below.

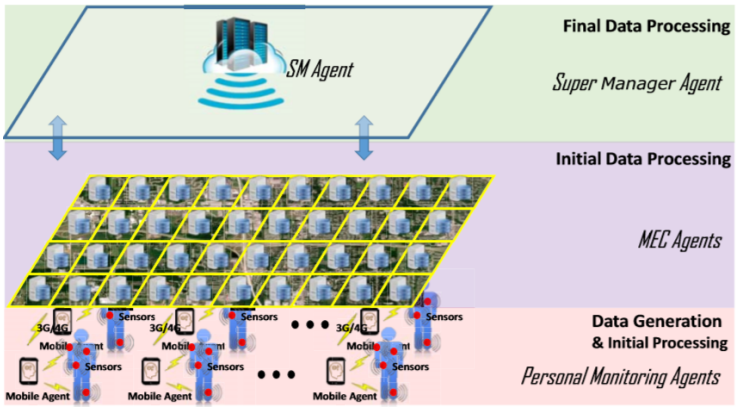


Figure 4: Model Components [18]

5.3.1 Personal Monitoring Agent

The monitored human is supposed to wear textiles with integrated sensors that measure the monitored human's vital signs. The personal monitoring agent is in a form of a software installed on the monitored human's smart phone. The wearable sensors periodically measure the human's vital signs and send them to the personal monitoring agent via Bluetooth. Upon the reception of the transmitted data, the personal monitoring agent performs an initial analysis to detect abnormal vital signs based on a predefined normal range stored on its knowledge base. The personal agent then communicates with the MEC agent supervising on its cell and provides it with its processed information. [18]

5.3.2 MEC Agent

A mobile edge computing MEC Agent is assigned to supervise a cell and is responsible for collecting and analyzing health monitoring data received from personal agents within its cell. It is also responsible for aggregating the analyzed data at the cell's level and sending them to the Super MEC for further processing. [18]

5.3.3 Super Manager Agent

The Super Manager Agent is responsible for gathering and aggregating the partially processed data received from the MEC agents. It is also responsible for performing the final data processing needed to detect the possible threats. [18]

6 Technique for Threat Detection in Rapid Biological Threat Detection Model

In order to detect the source of a biological attack, the early biological threat detection model by Al-Zinati et al. [18] ranks different cells based on their suspicion scores - a higher suspicion score yields to a higher rank, given that 1 is the highest rank. The more humans that traversed a specific cell and ends up being infected, the more suspicious that specific cell becomes. The concept of taxonomy [9] is used in this study to classify the different states of the observed organisms, in this case, humans.

6.1 Similarity Coefficient Concept Definitions

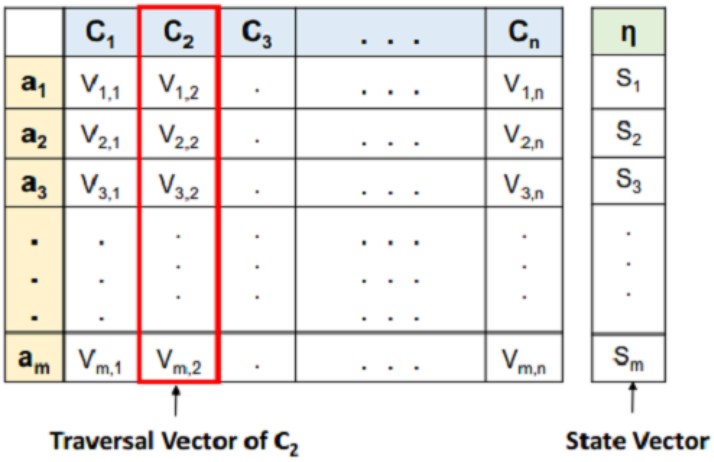


Figure 5: Data Model for Cell Traversal and Agent States [18]

Take Figure 5 for example; Let a_j be different agents, C_k be different cells, $V_{j,k}$ be traversal vectors, and S_j be the states of agent a_j. $V_{j,k}$ represents the history of traversal of agent a_j within cell C_k. S_j represents the state of agent a_j, if infected or not.

In order to identify the level of suspicion score of a certain cell, the State Vector is compared to each of the Traversal Vectors (see the highlighted box in Figure 5). The more similar the traversal vector of a cell to the state vector, the more likely the cell is infected, hence, the more suspicious a cell is.

To measure the similarity between two vectors, a statistical method called Similarity Coefficient is used. One similarity coefficient method used in numerical taxonomy is called Jaccard's Similarity Coefficient or the Jaccard Index. Jaccard's coefficient outputs a number between 0 and 1 after a comparison, which can also be interpreted as 0% to 100%. It indicates that the higher value means higher similarity between the compared samples [9]. An example of using Jaccard's coefficient to compare two sets is presented in Figure 6.

$$A = \{0,1,2,5,6\}$$
$$B = \{0,2,3,4,5,7,9\}$$
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|\{0,2,5\}|}{|\{0,1,2,3,4,5,6,7,9\}|} = \frac{3}{9}$$

Figure 6: Jaccard's Similarity Example [29]

As the values used in the State and Traversal vectors in Figure 5 only consists of 0s and 1s, we are using a different option to effectively compare two vectors with Jaccard's coefficient. Since the vectors only contain two states (0 and 1), they fall in the category of Binary characters, commonly used by taxonomists [9]. To properly know how can we compare the two vectors, we first define our OTUs (Operational Taxonomic Units/classification of organisms to be studied), and present them to the contingency table in Figure 7.

Taking the Traversal Vector and State Vector as OTUs, we can define their characters as follows:

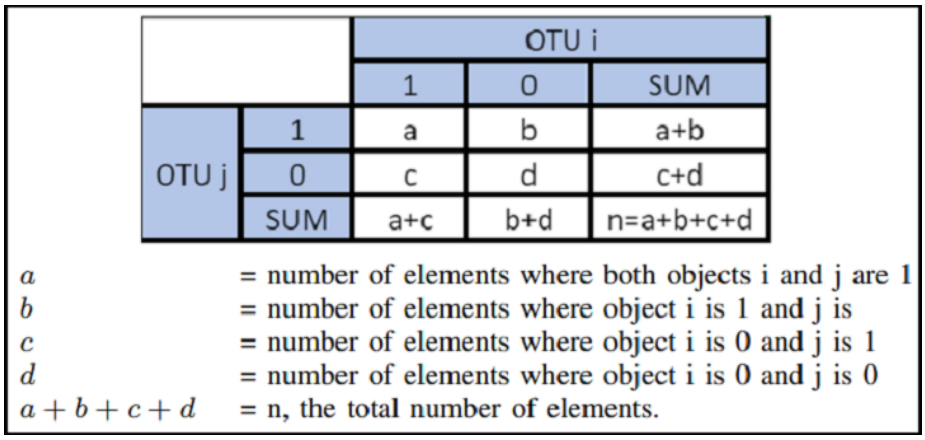


Figure 7: Operational Taxonomic Units Contingency table

- For the values of the Traversal Vector; 1 if cell "C" is traversed, 0 otherwise.
- For the values of the State Vector; 1 if agent "a" is infected, 0 otherwise.

These provided data generates the following contingency table in Figure 8 along with the replacement values of a, b, d, and d which are useful in the similarity coefficient measurement.

6.2 Similarity Coefficient Construction

Given the previous contingency table in Figure 8, we should apply the following observations to our similarity coefficients:

- If a human got infected, then it must have traversed one or more infected cells. the suspicion score of a cell rises as more humans that traversed it are infected. The more humans infected after traversing a certain cell, the more likely that certain cell is infected.
- If a human is not infected after traversing a cell, then the cell most probably is not infected. The suspicion score of a cell must be inversely propor-

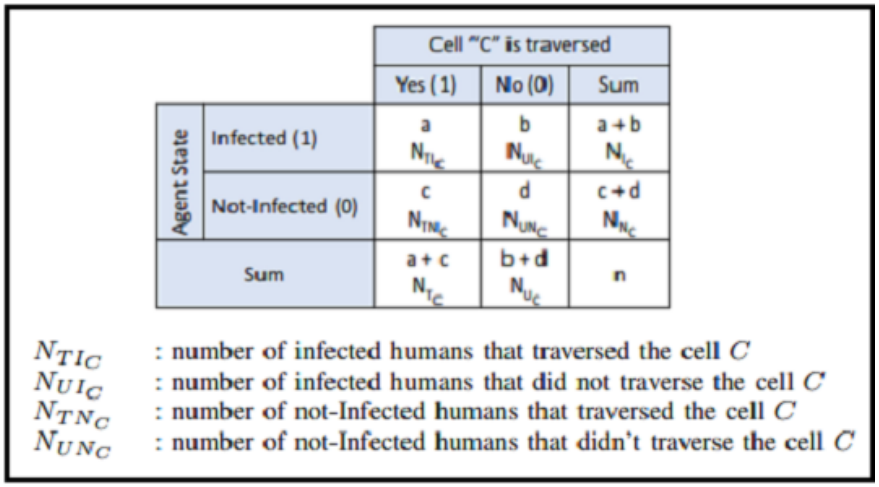


Figure 8: Agent State and Cell Traversal as OTUs [18]

tional to the number of not infected humans that traversed it. The larger the number of non-infected humans that traversed it, the less likely that cell is infected.

- If a human got infected without traversing a particular cell, then the less likely that cell is infected. The suspicion score of a cell must be inversely proportional to the number of infected humans that did not traverse it.

There have been many similarity techniques created that satisfies the given observation. In the study "On the Equivalence of Certain Fault Localization Techniques," the researchers found that there is mutual equivalence in similarity coefficient techniques since the techniques studied produce identical rankings, even if producing different values. [14]

Since the purpose of this study is to detect a cell with the highest suspicion score, the ranking is more important than the output value of the similarity coefficient technique used. Jaccard's coefficient is chosen in this paper for the following reasons:

- It meets the given observations from the generated contingency table.
- It is a simple technique which is highly effective for a real-time, rapid computing system.
- It does not produce a division by zero mathematical error for its computation.

Figure 9 shows the formula for Jaccard's similarity index as labled original and the formula for generating the suspicion score of a cell based on the previous given observations as applied.

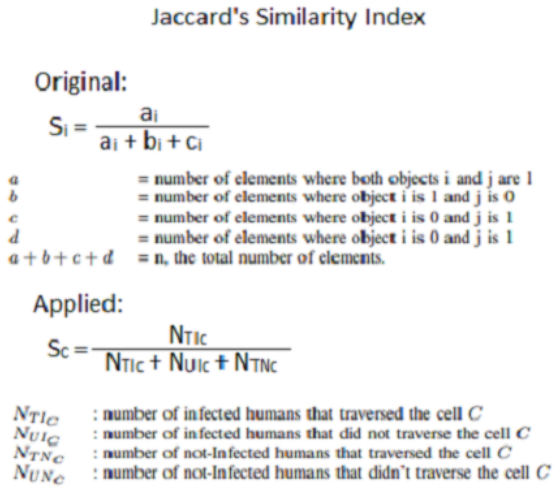


Figure 9: Suspicion score formula

7 Algorithms for Threat Detection in Rapid Biological Threat Detection Model

The following subsections introduces the algorithms for the personal monitoring agent, MEC agent, and supermanager agent - and will also show some

data generated for each of the algorithms.

At the initialization time, the environment is partiotioned into a set of equal-dimension fix-sized cells. There are a total of three levels of data processing in the proposed model.

At the micro level, personal monitoring agents or PMA continiously gather data generated by the integrated sensors and analyze if abnormalities exist. PMAs are also responsible for associating the collected data with the information on traversed cells of each human. Periodically, the PMAs process the data and send them to the MEC agents for further processing.

At the cell level, MEC agents combine the processed data from the PMAs within their cell and partially apply the similarity coefficient based technique using their local knowledge - one such technique is summation of terms. After that, the MEC agents send the information to the Super Manager.

Lastly, the Super Manager computes the suspicion of each cell and also ranks each cell according to their suspicion in order to detect potential threats.

7.1 Personal Monitoring Agent (PMA)

The personal monitoring agent is responsible for gathering the data generated by sensors and gathering the traces of movement around the environment. Algorithm 1 detects and process information related to 1) leaving the current cell boundaries and 2) abnormal vital signs.

In case the personal agent detects entering a new cell, it executes Algorithm 1 Section(A) to add the newly traversed cell (C_w) to its set of traversed cells (l). Then, the PMA applies the similarity coefficient technique to its local knowledge - i.e. setting up the values for each characteristics of a_n. Given this, the PMA executes Algorithm 1 Section(B) to test if the human has entered a new cell with normal vital signs ($IsNormal(\rho(t))$). It sets $a_n.N_{TN_{C_w}}$ to 1 to indicate

that human a_n traversed the cell C_w without being infected (with normal vital signs), then other values $a_n.N_{TI_{C_w}}$ and $a_n.N_{UI_{C_w}}$ are set to 0. These values are sent to the corresponding MEC agent M_k that governs the cell C_w for further processing.

Algorithm 1: a_n Personal Monitoring Agent Algorithm [18]

input : $\rho(t)$: vital signs

output: $a_n.N_{TI_C}, a_n.N_{TN_C}, a_n.N_{UI_C} \mid C \in ENVIRONMENTCELLS$

begin

/* ******* A ******* */

if *CellHasChanged* **then**

$C_w \leftarrow CurrentCell(t)$

$l.add(C_w)$

/* ******* B ******* */

if $CellHasChanged$ **and** $isNormal(\rho(t))$ **then**

$a_n.N_{TI_{C_w}} \leftarrow 0$

$a_n.N_{TN_{C_w}} \leftarrow 1$

$a_n.N_{UI_{C_w}} \leftarrow 0$

/* ******* C ******* */

else if $\neg isNormal(\rho(t))$ **and** $abnormalFirstDetected = \phi$ **then**

$abnormalFirstDetected = t$

$a_n.N_{TI_{C_w}} \leftarrow 1$

$a_n.N_{TN_{C_w}} \leftarrow 0$

$a_n.N_{UI_{C_w}} \leftarrow 0$

foreach $C_n \notin l$ **do**

$a_n.N_{UI_{C_w}} \leftarrow 1$

foreach $C_t \in l$ **do**

if $C_t \neq C_w$ **then**

$a_n.N_{TI_{C_w}} \leftarrow 1$

$a_n.N_{TN_{C_w}} \leftarrow -1$

if $\neg CellHasChanged$ **then**

$a_n.N_{TN_{C_w}} \leftarrow -1$

$SEND(M, a_n.N_{TI_{\{C\}}}, a_n.N_{TN_{\{C\}}}, a_n.N_{UI_{\{C\}}})$

end

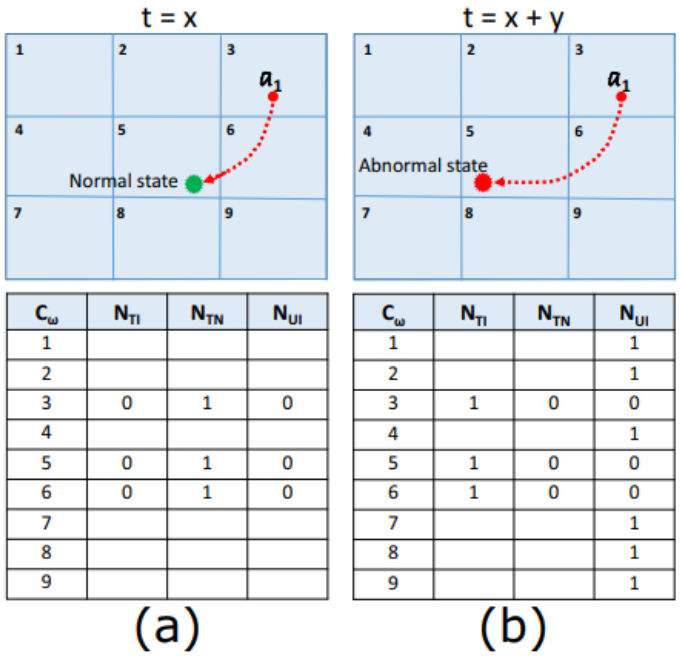
On the other hand, if the PMA detects abnormal vital signs for the first time $abnormalFirstDetected = \phi$, it executes Algorithm 1 Section C. In this case, it sets $a_n.N_{TI_{C_w}}$ to 1 to indicate that the human traversed cell C_w with abnormal vital signs (infected). At the same time, the values $a_n.N_{TN_{C_w}}$ and $a_n.N_{UI_{C_w}}$ are set to 0. Additionally, the PMA iterates on all cells not traversed by the monitored human $C_n \notin l$ and sets $a_n.N_{UI_{C_n}}$ to 1 to indicate that the monitored human is infected without traversing these cells.

As the infected state in this case is detected after there were previous records sent to the MEC indicating that the monitored human traversed cell/s C_w with normal state $a_n.N_{TN_{C_w}} \leftarrow 1$, the PMA iterates over all the traversed cells $C_t \in l$ and sets $a_n.N_{TI_{C_t}}$ to 1 to indicate that the monitored human became infected after traversing these cells. Moreover, for each of these cells, the PMA sets $a_n.N_{TN_{C_t}}$ to -1 to reverse the effect of the previously sent values to M $M_k \in M$.

In case the monitored human got infected while traversing the same cell $\neg CELLHASCHANGED$, the PMA sets $a_n.N_{TN_{C_w}}$ to -1 to reverse the effect of setting the value $a_n.N_{TI_{C_w}}$ to 1 in Algorithm 1 Section B.

Finally, the PMA sends the processed data to the corresponding MEC agent for further processing.

Note that after the first detection of abnormal vital signs, the PMA does not change values of $a_n.N_{TI_{C_w}}$, $a_n.N_{TN_{C_w}}$ and $a_n.N_{UI_{C_w}}$. This is necessary to filter out irrelevant movement traces and enhances the accuracy and performance of detecting threats.



C_ω	N_{TI}	N_{TN}	N_{UI}
1			
2			
3	0	1	0
4			
5	0	1	0
6	0	1	0
7			
8			
9			

C_ω	N_{TI}	N_{TN}	N_{UI}
1			1
2			1
3	1	0	0
4			1
5	1	0	0
6	1	0	0
7			1
8			1
9			1

Figure 10: Example of PMA's movement traces with a change in vital signs [18]

To illustrate the execution of Algorithm 1, Take a look at the example illustration in Figure 10. At time $t = x$, a_1 has traversed cells 3, 6 and 5 with normal vital signs. Hence, the values of $a_1.N_{TNC_3}$, $a_1.N_{TNC_5}$ and $a_1.N_{TNC_6}$ are set to 1. However, at time $t = x + y$, a_1's vital signs becomes abnormal while traversing cell 5. Therefore, the values of $a_1.N_{TNC_3}$, $a_1.N_{TNC_5}$ and $a_1.N_{TNC_6}$ are set to 0 while the values of $a_1.N_{TIC_3}$, $a_1.N_{TIC_5}$ and $a_1.N_{TIC_6}$ are set to 1. Additionally, the values of $a_1.N_{UIC_w}$ where $w \notin (3, 6, 5)$ are set to 1.

7.2 MEC Agent

The MEC Agent asynchronously receives data sent by the PMAs within its cell and periodically executes algorithm 2 to aggregate the data and send it to the Super Manager Agent for final processing.

Algorithm 2: M_k Agent Algorithm [18]

input : $a_n.N_{TI_{\{C\}}}, a_n.N_{TN_{\{C\}}}, a_n.N_{UI_{\{C\}}} \,|\, a_n \in$
$M_k.CELLAGENTS, C \in ENVIRONMENTCELLS$

output: $M_k.N_{TI_C}, M_k.N_{TN_C}, M_k.N_{UI_C} \,|\, C \in$
$ENVIRONMENTCELLS$

begin

/* ******* A ******* */

foreach

$a_n.N_{TI_C}, a_n.N_{TN_C}, a_n.N_{UI_C} \in \{a_n.N_{TI_{\{C\}}}, a_n.N_{TN_{\{C\}}}, a_n.N_{UI_{\{C\}}}\}$

do

$M_k.N_{TI_C}.add(a_n.N_{TI_C})$

$M_k.N_{TN_C}.add(a_n.N_{TN_C})$

$M_k.N_{UI_C}.add(a_n.N_{UI_C})$

/* ******* B ******* */

$SEND(SM, \{M_k.N_{TI_C}, M_k.N_{TN_C}, M_k.N_{UI_C} \,|\, C \in$
$ENVIRONMENTCELLS\})$

end

The MEC agent in a cell (M_k) starts executing the algorithm by aggregating the received values $a_n.N_{TI_C}$, $a_n.N_{TN_C}$, and $a_n.N_{UI_C}$ for each agent a_k within its cell k to find their totals at the cell level. See Algorithm 2 Section A.

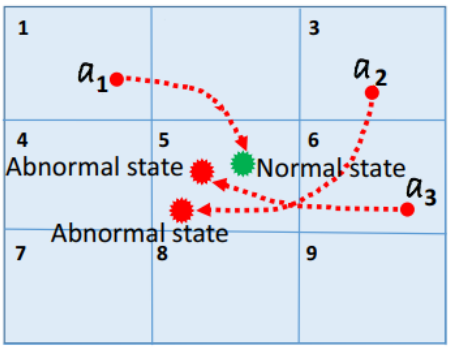
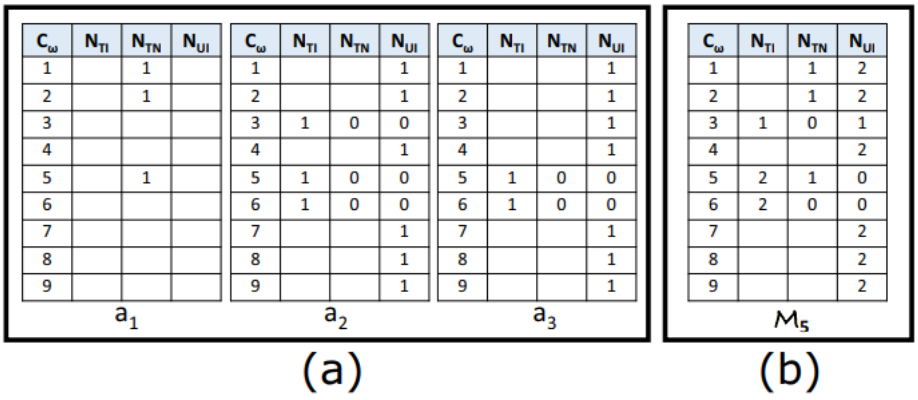


Figure 11: Example of a 9-cell environment with 3 monitored humans [18]



C_ω	N_{TI}	N_{TN}	N_{UI}
1		1	
2		1	
3			
4			
5		1	
6			
7			
8			
9			

a_1

C_ω	N_{TI}	N_{TN}	N_{UI}
1			1
2			1
3	1	0	0
4			1
5	1	0	0
6	1	0	0
7			1
8			1
9			1

a_2

C_ω	N_{TI}	N_{TN}	N_{UI}
1			1
2			1
3			1
4			1
5	1	0	0
6	1	0	0
7			1
8			1
9			1

a_3

(a)

C_ω	N_{TI}	N_{TN}	N_{UI}
1		1	2
2		1	2
3	1	0	1
4			2
5	2	1	0
6	2	0	0
7			2
8			2
9			2

M_5

(b)

Figure 12: The values of $N_{TI_{C_w}}$, $N_{TN_{C_w}}$, and $N_{UI_{C_w}}$: *a*) generated by agents a_1, a_2, a_3. *b*) aggregated by M_5. [18]

Figure 11 shows the movement traces of 3 monitored humans in a 9-cell environment. The current cell (C_w) of each humans is cell 5 (C_5). Agents a_1 and a_3 are infected while agent a_2 has a normal state. The values generated by the personal agents (a_1,a_2,a_3) is given in Figure 12 (a). M_k aggregates the values received in the current cell 5; the result of M_5 is in Figure 12 (b).

M_k then sends the set of aggregated data to the supermanager agent (SM) for further processing and analysis (Alorithm 2 Section B).

7.3 Super Manager Agent

The Super Manager Agent (SM) combines the processed data sent by the MEC agents and performs the final processing and analysis for threat detection. SM starts by combining the intermediary data or the previous aggregated data sent by the MEC agents, see Algorithm 3 Section A. For each cell in the environment, $C \in ENVIRONMENTCELLS$, SM computes the following:

- $SM.N_{TI_C}$: the final number of infected agents that traversed the cell C.
- $SM.N_{TN_C}$: the final number of not-infected agents that traversed the cell C.
- $SM.N_{UI_C}$: the final number of infected agents that did not traverse the cell C.

Algorithm 3: SM Agent Algorithm [18]

input : $M_k.N_{TI_{\{C\}}}, M_k.N_{TN_{\{C\}}}, M_k.N_{UI_{\{C\}}} \mid C \in ENVIRONMENTCELLS$

output: $SUSPICIOUSCELLS(t)$

begin

/* ******* A ******* */

foreach $M_k.N_{TI_C}, M_k.N_{TN_C}, M_k.N_{UI_C} \in \{M_k.N_{TI_{\{C\}}}, M_k.N_{TN_{\{C\}}}, M_k.N_{UI_{\{C\}}}\}$ **do**

$SM.N_{TI_C}.add(M_k.N_{TI_C})$

$SM.N_{TN_C}.add(M_k.N_{TN_C})$

$SM.N_{UI_C}.add(M_k.N_{UI_C})$

/* ******* B ******* */

foreach $C \in ENVIRONMENT$ **do**

$SM.S_c \leftarrow \frac{SM.N_{TI_c}}{SM.N_{TI_c}+SM.N_{TN_c}+SM.N_{UI_c}}$

$SUSPICIOUSCELLS_x(t) \leftarrow RANK(SM.S_{\{c\}}(t))$

end

After the computations in Algorithm 3 Section A, SM proceeds to Algorithm 3 Section B for computing the suspicion score for each environment cells (S_C). The suspicion score of each cell are ranked in descending order to identify the cells with highest probability of containing a threat. The higher the suspicion score, the more likely the cell contains a threat.

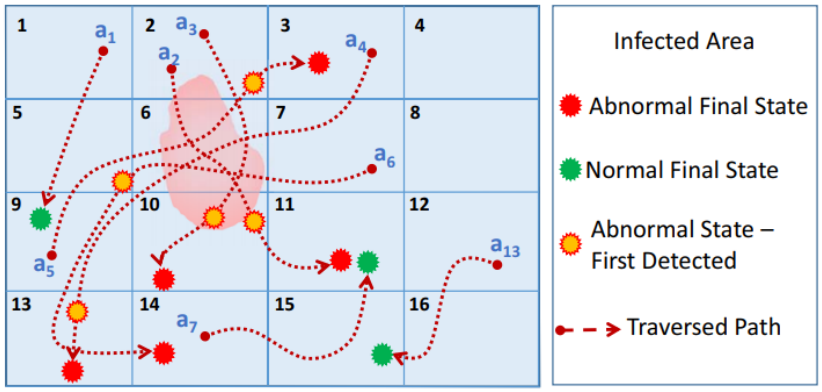


Figure 13: Example of a monitored health environment with agent movement traces [18]

Figure 13 helps to illustrate the execution of Algorithm 3. In this example, the environment contains 16 equally sized cells. The infected area spans inside $cell_2$, $cell_6$, and $cell_10$. The dashed arrows represents the agents' movements.

Cell	N_{TI}	N_{UI}	N_{TN}	Suspicion Score	Suspicion Rank
1	0	5	1	0	
2	3	2	0	0.6	2
3	2	3	0	0.4	4
4	0	5	0	0	
5	0	5	1	0	
6	5	0	0	1	1
7	2	3	0	0.4	4
8	0	5	0	0	
9	3	2	1	0.5	3
10	3	2	0	0.6	2
11	1	4	1	0.166667	5
12	0	5	1	0	
13	2	3	0	0.4	4
14	1	4	1	0.166667	5
15	0	5	1	0	
16	0	5	1	0	

Figure 14: Final suspicion scores computation and ranking [18]

Figure 14 shows the data generated by Algorithm 3, with the ranking of each cells. Take $cell_6$ for example; the final computations of SM shows that the number of infected humans that traversed this cell N_{TI_6} is 5, the number of infected humans that did not traverse this cell N_{UI_6} is 0, and the number of not-infected humans that traversed this cell N_{TN_6} is 0. By applying the Jaccard similarity coefficient applied formula presented in figure 9, the computed suspicion score of $cell_6$ is 1, which is the maximum value in terms of probability. Hence, there is a high probability that $cell_6$ contains a threat. Additionally, $cell_2$ and $cell_{10}$ are ranked 2^{nd} with the suspicion score of 0.6.

The computed results are compatible with the scenario given in figure 13 since the threat is mainly contained in $cell_6$ and parts of $cell_2$ and $cell_{10}$.

Take note that although in Figure 14, the cells $C \in [3, 7, 9, 11, 13, 14]$ contains suspicion scores, that does not mean that these cells should contain threats - they do not, as shown in Figure 13. Since the suspicion score is a probability, it just means that these cells likely contains a threat but since their suspicion scores are low, it means that the likeliness of these cells to contain any threat is low. Additionally, this study only focuses on the suspicion rank not on suspicion score as mention in section 6, since the cells that contain any threat should have higher suspicion score, and thus, ranked higher.

8 An example of a spreading disease

The model of early biological threat detection of Al-Zinati et al. [18] is only effective in such scenarios where humans are only infected by infections present in the environment. We aim to identify the effectiveness of this model in such scenarios where infected humans also have the ability to infect other humans. To be able to do so, we will introduce a certain disease and aim to replicate this disease in our simulation.

This section introduces a scenario where a disease with the ability to spread among humans is present. This section also provides an analysis as to which variables are needed to be able to replicate a scenario where such disease is present to be able to make a simulation for testing.

8.1 Coronavirus disease 2019

Coronavirus disease 2019 or $COVID-19$, is a highly contagious viral illness caused by severe acute respiratory syndrome coronavirus 2, or $SARS-CoV-2$. After the first case was reported in Wuhan, Hubei Province, China in late December 2019, SARS-CoV-2 rapidly disseminated across the world in a short span of time, compelling the World Health Organization WHO to declare it as a global pandemic on March 11, 2020. [2]

In the paper 'Features, Evaluation, and Treatment of Coronavirus (COVID-19)', Cascella et al. stated that; "Like other RNA viruses, SARS-CoV-2, while adapting to their new human hosts, is prone to genetic evolution with the development of mutations over time, resulting in mutant variants that may have different characteristics than its ancestral strains. Several variants of SARS-CoV-2 have been described during the course of this pandemic." [2]

Despite the speed of vaccine development and global mass vaccination effort against the Coronavirus disease 2019, the emergence of new variants threatens the progress made to preventing the spread of the disease [2]. The importance of Non-Pharmaceutical Interventions is clearly highlighted on such scenarios where humans cannot fully rely on medical treatments to prevent contracting the spread of such diseases.

The following subsections will introduce variables which are needed to be able to simulate a disease.

8.2 Incubation Period

A study by Lauer, S. et al., analyzes the incubation period - the period between the time of contact of a person to the disease to the time the person shows symptom, of Coronavirus disease 2019 from pooled reports between 4 January 2020 and 24 February 2020. Their results are the following: 5.1 days median incubation period and for conservative assumptions, people monitored will develop symptoms after 14 days. [13]

To be able to replicate a realistic scenario into our simulation, we will be using the incubation period as an additional variable.

8.3 Transmission Rate

Zhang, L. et al. studied the transmission process of Coronavirus disease 2019 from close contact observations using 9120 reported cases in mainland China from January 15 to February 29, 2020. They defined the transmission rate as the number of people infected in one close contact event over the number of people in that event. From their empirical distribution of the transmission rate, the mean and median transmission rate is 0.2 and 0.13 respectively.[34]

8.4 Recovery Time

Tolossa et al. aimed to study the median recovery time for patients showing symptoms of Coronavirus Disease 2019. Their result found that the median for the recovery time of patients was 18 days from showing symptoms, within the range of 10-27 days. [31]

8.5 Variable values for the experiment

To be able to replicate a more realistic scenario for this experiment, since we will be attempting to use rapid biological threat detection to improve the

NPI lockdown, we will be integrating incubation period, transmission rate, and recovery time to our simulation. More specifically, the values to be used in this experiment are the following; the median for the incubation period 7 days and the mean for the transmission rate 0.2 from the experiment of Zhang, L. et al [34], and using 10 days from the study of Tolossa et al. [31].

Although we aim to make our simulation more realistic, it could be seen that we are using fixed values for the incubation period, transmission rate and recovery time. In our end, we are only making the experiment not too complicated, as long as we can have the feature of humans infecting each other in our experiment, which can be seen as we chose a higher value of transmission rate.

9 Introducing the additional variables and agents for the experiment

This section introduces variables that would be used for the simulation of this experiment. The variables included would only be used to determine the effectiveness of adding rapid biological threat detection as a non-pharmaceutical intervention for lockdown and does not include other prevention such as social distancing, wearing face masks, and other forms of non-pharmaceutical interventions.

9.1 Incubation period, transmission rate and recovery time

The values to be used are 7 days for the incubation period, 0.2 or 20% for the transmission rate and 10 days for the recovery time as indicated in section 8.5. These fixed values are to be used for the simulation of this experiment to maintain simplicity of the simulation, avoiding over-definition of terms that are outside the scope of this experiment.

9.2 Health facilities and the Growth Factor

To be able to have an indicator for the spread of a disease we are using "growth factor", as also used by the Australian government to gauge their success on containing the Coronavirus disease 2019. Contrary to the growth rate or reproduction rate R and $R0$ which estimates how many people are to be infected, growth factor measures the change from new cases being reported. The growth factor gives a minimum value of 0; if the growth factor is above 1, then new cases are increasing, if below 1, it means the spread is under control. [7]

The Health facility is another crucial tool we will use, not for helping the treatment of infected humans but for helping the prevention of the spread against threats. For this experiment we will be adding health facilities for their function of detecting possible disease carriers among humans that are traversing inside their area of observation. The records of health facilities are crucial for initiating lockdowns in regions such as how they can monitor the number of cases in real world scenarios. We will be replicating this particular function of health facilities for this experiment. Specifically, health facilities will be a key for computing growth factor within their areas.

9.3 The SIR model

The research model is based on the susceptible-infectious-recovered or SIR model for infectious diseases [32]. We have $S(t)$ as people vulnerable to infection at time t, $I(t)$ as infected people at time t, and $R(t)$ as recovered people at time t. We have the following equation $P = S(t) + I(t) + R(t)$, where P is the total population. This model shows that the population only have three states, S, I,and R, where after some time $t > 0$, every infected human in the population approaches R, with a small chance that some humans stay in the S state if they are not infected by the disease. Having an R or $P = R(t)$ population means

having an immune population, with each person being recovered or have died.

The aim of this study is to discover whether the model "an agent based model for rapid biological threat detection" [18], is effective as a non-pharmaceutical intervention for preventing the spread of infectious diseases.

10 The proposed model for using Early Biological Threat detection as an effective Non-Pharmaceutical Intervention

This section presents a model for making use of the Early Biological Threat detection model from Al-Zinati et al.[18], to improve lockdowns on containing the spread of contagious or viral diseases. Same as the early biological threat detection model, the simulation for this experiment will also partition the environment into equally divided areas called as cells.

10.1 The population

In this simulation, The population consists of three states; the susceptible, infected, and recovered.

The simulation will start with an N number of susceptible population. The susceptible population can be infected in two ways - by traversing areas that are infected and by staying on the same area with other infected humans.

Infected humans may infect other humans depending on the provided transmission rate of the disease. Infected humans may appear normal until they start showing symptoms based on the provided incubation period of the disease.

After a certain period, infected humans transition to being recovered. The time of the recovery of these humans depend on the provided recovery time. Recovered humans will cease to be monitored by the health facilities and per-

sonal monitoring agents as recovered humans would be considered immune to the disease or in the worse case, dead.

10.2 Area of Infection

A certain area will be marked as an infected area, which will be the source of infection, similar to the area of infection in the experiment conducted by Al-Zinati et al.[18] Humans traversing this area will be infected. The rapid biological threat detection model aims to detect cells that contain this infected area.

10.3 Health Facilities and Lockdown

Each cell would contain one health facility. Health facilities are responsible for recording infected humans that traverse its area of observation, see algorithm 4 section A. Health facilities are also responsible for calculating the local growth factor within its cell. When the value of local growth factor exceeds 1, the health facility will initiate lockdown on its cell, where humans inside will not be able to go out, and humans outside will not be able to go in.

Health facilities are added to our experiment and are given the ability to initiate lockdowns. In real life scenarios such as on the recent outbreak of the Coronavirus 2019 disease, health facilities not only functions as treatment and isolation sites of infected patients, they also function as tools for counting and recording infected humans within their scope of observations. In this experiment, we will use their function of counting and recording infected humans traversing within their area of observation and give them the authority of initiating lockdowns.

Algorithm 4: Health Facilities

input : infectedAgents, lastWeekInfectedTotal, weekday

output: none

begin

```
/* ******* A ******* */
if weekday ≠ 7 then
    thisWeekInfectedTotal.add(count infectedAgents);
/* ******* B ******* */
else
    growthFactor ←
     thisWeekInfectedTotal/lastWeekInfectedTotal
    lastWeekInfectedTotal ← thisWeekInfectedTotal
    if growthFactor ≥ 1 then
        LockDownOnCell(thisCell)
```

end

The calculation of the growth factor is done every 7 days, or a week, where the growth factor is equal to the number of infected population this week divided by the number of infected population last week as shown in algorithm 4 section B.

To be able to track the time for the computation of the growth factor, a variable weekday is introduced, where it has values 1-7 - 1 being the first day of the week and 7 being the last day of the week. After each week cycle, the value of the weekday variable will transition from 7 to 1 repetitively. See algorithm 5.

Algorithm 5: Weekday Computation

input : ticks : environment time tick

output: weekday : integer with values 1-7

begin

days ←ticks / (ticks per day);

weeks ←days / 7;

weekday ←floor(weeks modulo 7) + 1;

end

10.4 Personal Monitoring Agent

The personal monitoring agent is responsible for recording traversed cells of each human and detecting if a human has abnormal symptoms or infected. For this model, we will be limiting what cells to record by removing recorded cells that exceeds the given incubation period of the disease. This change makes sense as infected humans are only labeled as infected by the personal monitoring agents if they start to show symptoms, meaning, they have only traversed areas within the incubation time of the disease. If the incubation period of a certain disease is one day, it is not likely that the infected human has traversed an infected area 2 days ago.

The algorithm of the personal monitoring agent will then be modified as presented in algorithm 6.

Algorithm 6: a_n Modified Personal Monitoring Agent Algorithm [18]

input : $\rho(t)$: vital signs

output: $a_n.N_{TI_C}, a_n.N_{TN_C}, a_n.N_{UI_C} \mid C \in ENVIRONMENTCELLS$

begin

/* ******* A ******* */

if *CellHasChanged* **then**

$C_w \leftarrow CurrentCell(t)$

$l.add(C_w)$

/* ******* A.2 ******* */

foreach $C_w \in l$ **do**

if C_w *exceeds incubation period* **then**

$l.remove(C_w)$

/* ******* B ******* */

if $CellHasChanged$ **and** $isNormal(\rho(t))$ **then**

$a_n.N_{TI_{C_w}} \leftarrow 0$

$a_n.N_{TN_{C_w}} \leftarrow 1$

$a_n.N_{UI_{C_w}} \leftarrow 0$

/* ******* C ******* */

else if $\neg isNormal(\rho(t))$ **and** $abnormalFirstDetected = \phi$ **then**

$abnormalFirstDetected = t$

$a_n.N_{TI_{C_w}} \leftarrow 1$

$a_n.N_{TN_{C_w}} \leftarrow 0$

$a_n.N_{UI_{C_w}} \leftarrow 0$

foreach $C_n \notin l$ **do**

$a_n.N_{UI_{C_w}} \leftarrow 1$

foreach $C_t \in l$ **do**

if $C_t \neq C_w$ **then**

$a_n.N_{TI_{C_w}} \leftarrow 1$

$a_n.N_{TN_{C_w}} \leftarrow -1$

if $\neg CellHasChanged$ **then**

$a_n.N_{TN_{C_w}} \leftarrow -1$

$SEND(M, a_n.N_{TI_{\{C\}}}, a_n.N_{TN_{\{C\}}}, a_n.N_{UI_{\{C\}}})$

Algorithm 6 section A.2 removes the record of cells that exceeds the incubation period of the disease. This helps in making the Rapid Biological Threat detection model more accurate in identifying possible suspicious cells as it removes cells traversed beyond the incubation period of the disease, meaning, a human showing symptoms only traversed an infected area within the incubation period of the disease.

10.5 MEC agent

The MEC agent is used for aggregating data received from personal monitoring agents received in its cell, aggregated data will be sent to the supermanager agent. There are no modification made for the MEC agent.

10.6 Supermanager agent

The supermanager agent aggregates data received from mec agents, then ranks each cell according to their suspicious score. Another method for marking cells for lockdown is then introduced to the supermanager agent, this method lets the rapid biological threat detection model to choose the most suspicious cells and mark them for lockdown.

Algorithm 7: a_n Marking suspicious cells for lockdown

```
input  : supermanager, topCellCount
output: none
begin
    topList ←
     supermanager.getTopRankSuspiciousCells(topCellCount)
    flagList ← []
    foreach c ∈ topList do
        if flagList = [] then
            flagList.add(c)
        else if flagList.containsAdjacentCellTo(c) then
            flagList.add(c)
end
```

Algorithm 7 first initialize *topList* where it gets the top 1 suspicious cell to top *topCellCount* as a list. It then creates a $flagList$ which should contain flagged or marked cells for lockdown, and initializes it to an empty list. For the first iteration, since $flagList$ is empty, it adds the top 1 suspicious cell to the list. For the next iterations, it checks if the current cell in variable c is adjacent to a cell contained in the $flagList$.

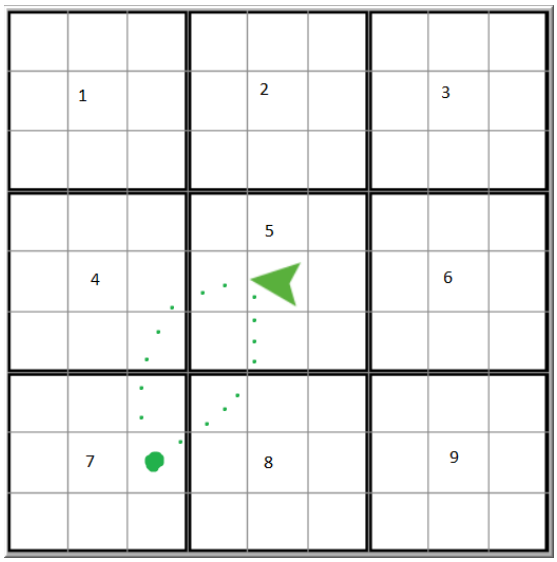


Figure 15: Turtles and Patches

The adjacency is true if a cell is above, below, or beside another cell. The diagonal adjacency is not considered since it is assumed that when humans traverse a diagonally adjacent cell, it first traverses the cells which are above, below, or beside the origin cell before arriving to the diagonally adjacent cell. See figure 15 for example. This is especially true for larger environments.

10.7 The duration of lockdowns

Lockdown duration differ with accordance to the implementations of different countries as mentioned in chapter 3. For our simulation, we will be using a time of seven days, or one week for the duration of each lockdown. We aim to minimize the duration of every lockdown on our simulation to take into consideration the negative effects of prolonged lockdown such as the adherence of the population as mentioned in chapter 3.1.1.

11 How the modelling environment works

Netlogo will be used for this experiment, a free agent-based modelling software. Netlogo, by default has three main agents - turtles, patches, and the observer. Patches are stationary agents that act as tiles for the environment, turtles are moving agents that can interact with each other, and the observer, as defined by Netlogo is the person/user that oversees and controls the whole experiment.

11.1 Netlogo tools and agents used

In this section, we will be discussing the tools and agents that were used in all three simulations (No Lockdown Simulation of an Epidemic, Epidemic with a Lockdown NPI, Epidemic with an RBTD-Lockdown NPI) of the experiment.

11.1.1 Turtles and Patches

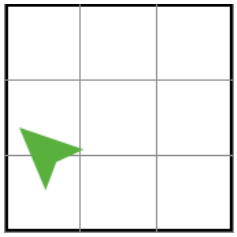


Figure 16: Turtles and Patches

For the particular example in figure 16, the environment consists of nine patches, with their boundaries shown in grey borders, and a turtle, which is shown as a cursor pointer image with a color green.

All turtles created will have a default movement speed of 1, meaning that they can travel one patch distance regardless of where they are facing. For the environment in this experiment, turtles can face 0-360 degrees randomly for

each movement, and will always land on a different patch for each step.

11.1.2 Cells

The experiments use a multi-cellular environment. To be able to create cells, the environment is partitioned into equally sized cells.

For this example, figure 17 has a 9x9 patch environment, and is partitioned into 9 equally sized cells. Thick black markers define the border of each cell. Turtles may also know which cells they are in since patches residing in a certain cell receive the id of that cell.

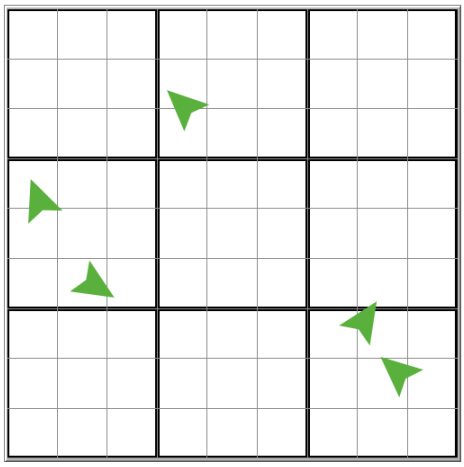


Figure 17: Environment with 3x3 cells

11.2 Turtle types

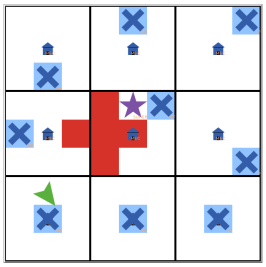


Figure 18: Setup of an RBTD-lockdown simulation with one Susceptible Human (green turtle)

- Susceptible Human/ Susceptible Population - These types of turtles seen in figure 18 are of the susceptible population of humans, given the color green.
- Infected Human/ Infected Population - when susceptible humans get infected, they are still green but after they show symptoms, they turn yellow.
- Recovered Human/ Recovered Population - After the recovery time passes, infected turtles transition from color yellow to grey. The turtles also stop moving as they are no longer part of the simulation.
- Healthcare Facilities - are shown as X in figure 18. They have their area of observation, which is also colored blue, and they are randomly placed inside a cell.
- Personal Monitoring Agent - these agents constantly monitor the humans, in the simulation, it is run whenever changes happen to humans.
- Mobile Edge Computing Agent - These agents are represented as houses located at the center of each cell as seen in figure 18, they are responsible

for aggregating data received from personal monitoring agents within their corresponding cells.

- Supermanager - the supermanager is representad as a star in the center of the environment as shown in figure 18. They are responsible for ranking suspicious cells and marking those cells for lockdown from the data gathered from all of the MEC agents.

11.3 Infection behavior

In all three experiments, the researchers have simulated the behaviours of COVID-19 and how it infects humans by introducing the ff:

- Area of Infection or Origin of the Epidemic - where all the infection comes from. The area of infection is an area with a set size that infects all the susceptible humans that step on it. In the real world, the area of infection is the area where the disease is first detected to have been contracted by a human.

- Infections between agents - infected humans are designed to be able to infect susceptible humans when they come into contact, stepping into the same patch. All susceptible agents that are infected show symptoms, turning yellow after 7 days and it also follows that they can infect other susceptible agents as well after showing symptoms.

- Transmission Rate and immune system - Depending on the given transmission rate, there is a chance that an agent might not be infected. On the other hand, the immune system determines how fast an infected human recovers from the disease. Higher immunity means the agent takes fewer days to recover, with a minimum of 5 days and lower immunity means they will take longer, up to 10 days, increasing the chances that

they might pass the disease onto others. The immunity of a human is set randomly as they are created in the simulation.

11.4 Area of Infection

The red patches in the middle of the environment located in the 4th and 5th cell of Fig 18 determines the area of infection in the environment. This means that if any susceptible human step into a red patch, they are already infected and thus they will show symptoms and turn into an infected human after 7 days. This can be adjusted in the netlogo interface (19) so that the area of infection would still be significant for larger environments. An example of this can be found in Figure 20.

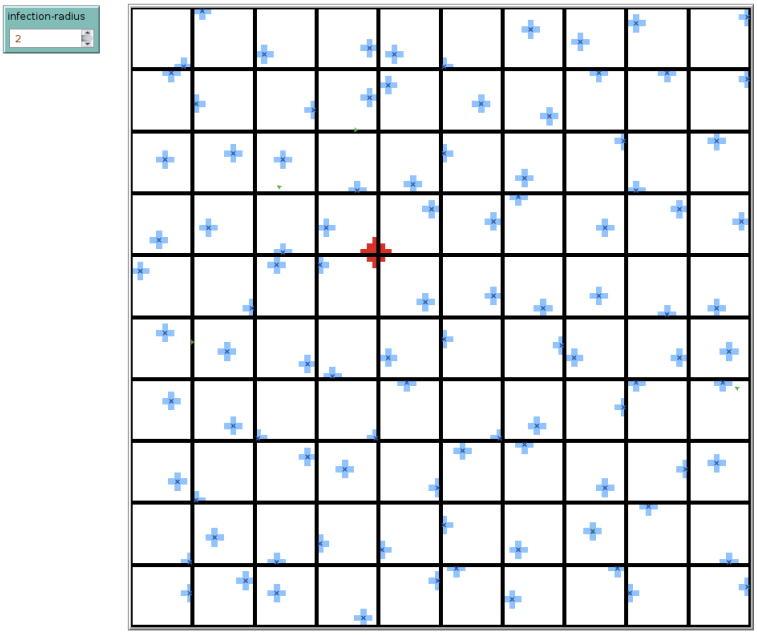


Figure 19: Netlogo 10 x 10 lockdown simulation with infection radius 2

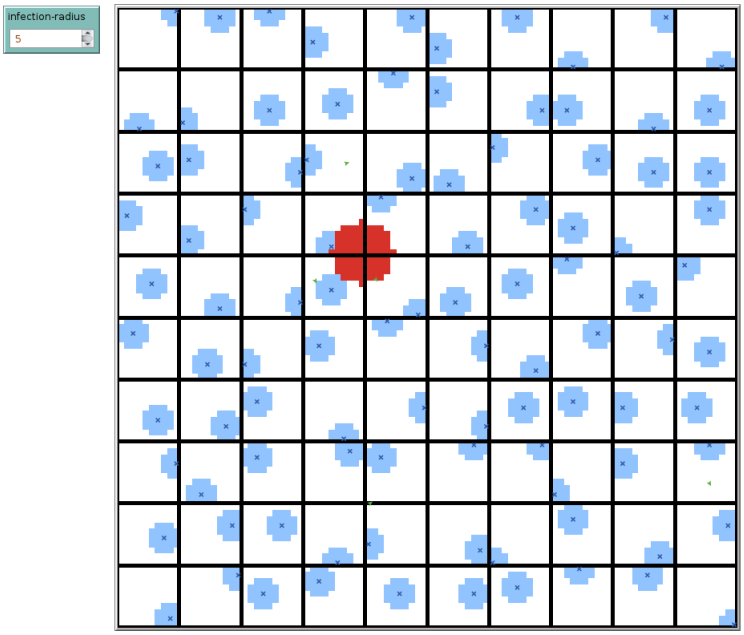


Figure 20: Netlogo 10 x 10 lockdown simulation with infection radius 5

12 Experiment

To be able to see if the proposed model is effective in containing contagious or viral diseases, the experiment will consist of three parts - no intervention, with lockdown intervention, and with lockdown intervention added with rapid biological threat detection model.

All experiment will consist of 1000 by 1000 patches and partitioned into 100 equally sized cells. The infection area is positioned in the middle of the environment with a radius of 50 patches.

The incubation period for the disease is set as 7 days, whereby an agent that become infected will show symptoms after 7 days. The transmission rate of the disease is set to 20%, where humans can infect each other when being on the same patch. The recovery period of an infected human after showing symptoms

is set to 5 days, plus an additional 0 to 5 days depending on their generated immunity.

The days are computed based on 1 cell dimension. In this case, each cell has 100x100 patches dimension, therefore, 1 day for a human in the simulation can take it to 100 patches. Assuming that 1 patch is equivalent to 20x20 meters in the real world, the humans in the simulation travels 2000 meters per day on foot. The size of the patches is limited due to the ability of the computing devices used.

12.1 Experiment 1: No Lockdown

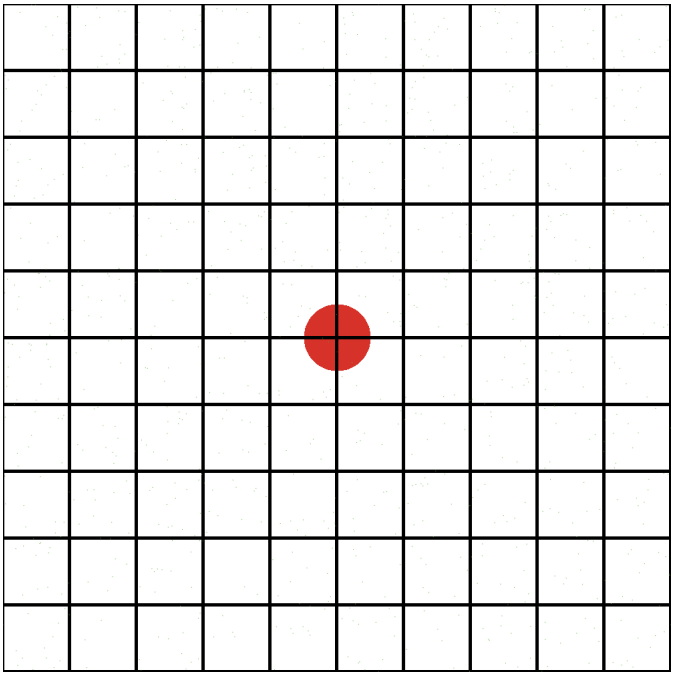


Figure 21: The simulation environment without lockdown

The environment generated in netlogo is shown on figure 21.

The first part of the experiment is to simulate 1000 humans in this environ-

ment. The simulation is done five times and the average result is displayed in figure 22.

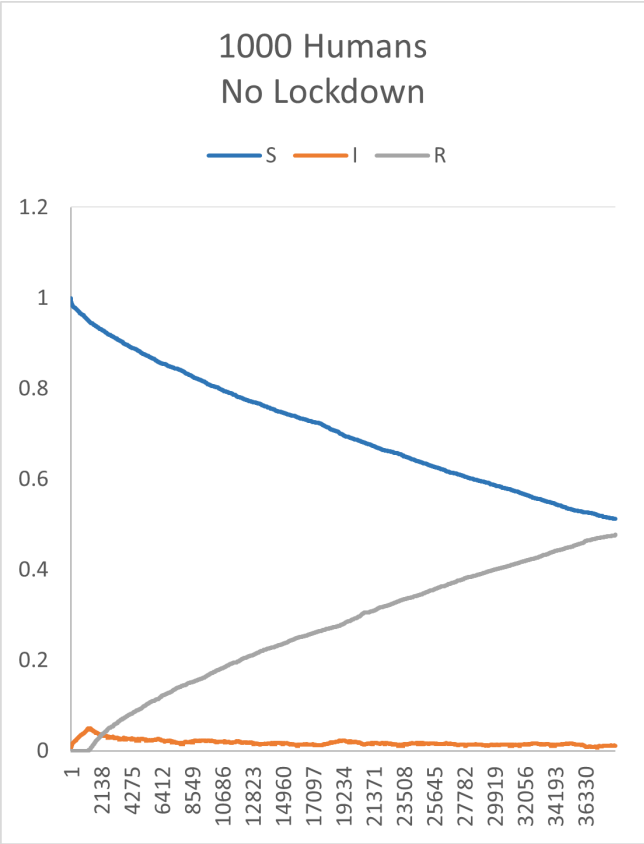


Figure 22: Average result for the simulation of 1000 humans without lockdown

The five simulations where executed until the recovered population reached 50 percent of the total population. Since each simulation took different amount of time to be able to reach the condition for termination, we acquired the fastest time and made it the basis for the five simulation when getting their averages. In this case we have 38450 ticks or 54.9 weeks.

Within 54.9 weeks, the susceptible population is left with 51.2 percent, the infected population is 1.08 percent and the recovered population is 47.92 percent.

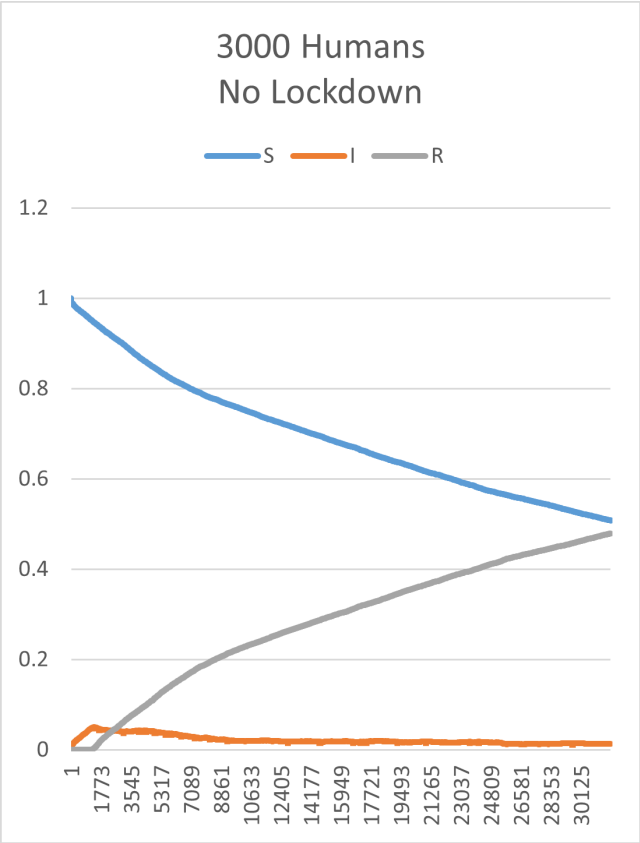


Figure 23: Average result for the simulation of 3000 humans without lockdown

In figure 23, the graph is shown for SIR values on the simulation of 3000 humans without lockdown. The time frame for this simulation is 31880 ticks or 45.5 weeks, resulting in 50.79 percent of susceptible, 1.29 percent of infected, and 47.92 percent of recovered population.

The result of both experiment is seen in figure 24

	1000	3000
Ticks	38450	31880
Susceptible	51.20%	50.79%
Infected	1.08%	1.29%
Recovered	47.72%	47.92%

Figure 24: The resulting SIR values for 1000 and 3000 humans in their respective timeframes

12.2 Experiment 2: Lockdown

For this experiment, health facilities are added to the environment. Health facilities record infected humans that traverses within their area. See figure 25.

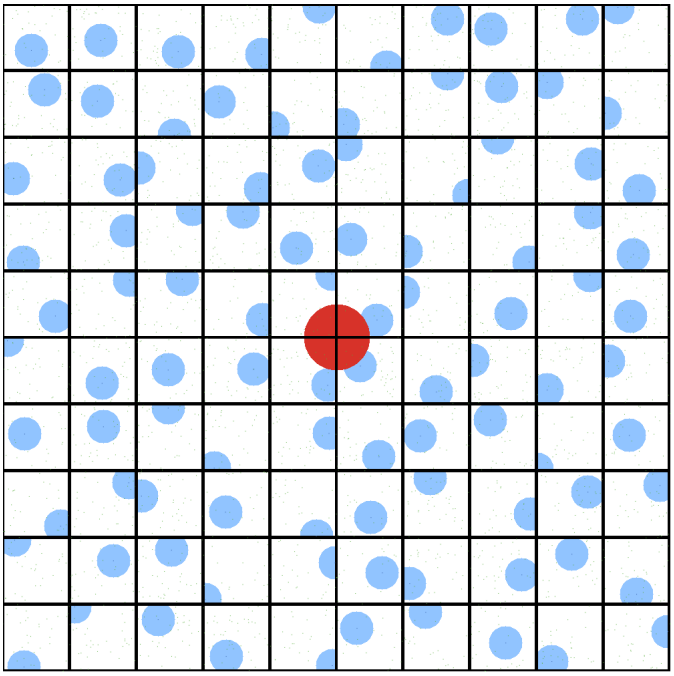


Figure 25: The simulation environment with lockdown

Each cell would have 1 health facility, recording every infected human that

traverses its area within that cell. The health facility is responsible for computing the growth factor within its cell. When growth factor reaches 1 or above as its value, the health facility will initiate a lockdown of its cell, preventing every human within from leaving and preventing humans outside from entering. This experiment sets the radius of observation of health facilities to 25 patches.

The experiment is simulated with 1000 humans and 3000 humans and the timeframe observed is based on the timeframe for 1000 and 3000 humans in the experiment on section 12.1.

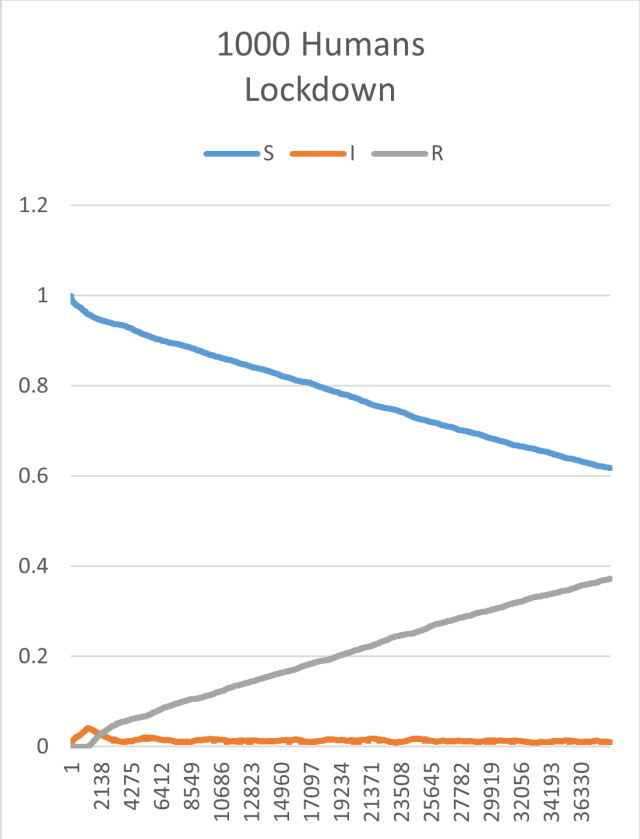


Figure 26: Average result for the simulation of 1000 humans with lockdown

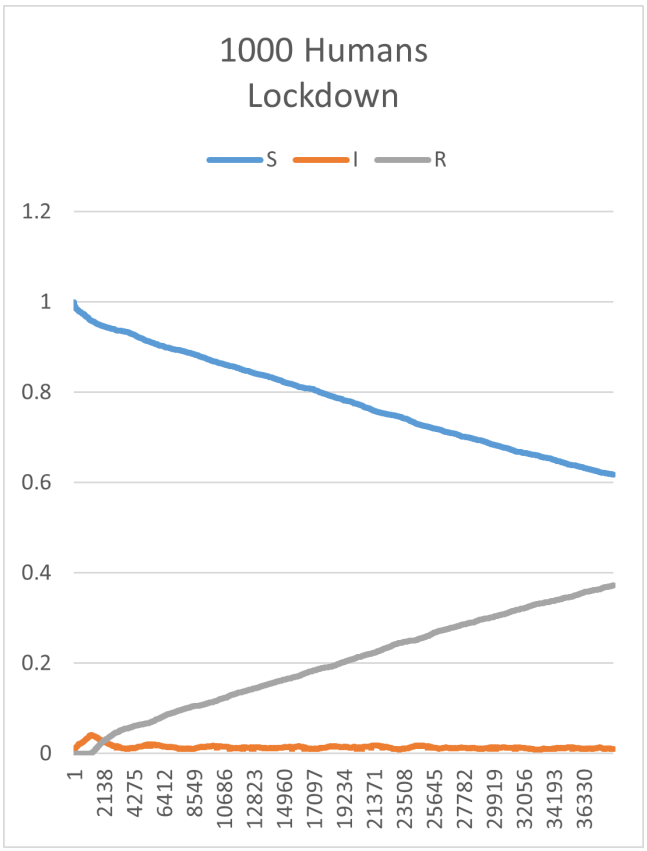


Figure 27: Average result for the simulation of 3000 humans with lockdown

	1000	3000
Ticks	38450	31880
S	61.72%	66.76%
I	1.06%	0.95%
R	37.22%	32.29%

Figure 28: The resulting SIR values for 1000 and 3000 humans in their respective timeframes for the environment with lockdown

Figure 26 and figure 27 shows the graph for the SIR of 1000 and 3000 humans respecively and figure 28 shows the resulting SIR at the end of the observed

timeframe for 1000 and 3000 humans.

12.3 Experiment 3: RBTD Lockdown

This experiment is done to observe whether the agent-based model for early biological threat detection is effective when used as an additional non-pharmaceutical intervention for lockdown. The simulation environment is presented in figure 29.

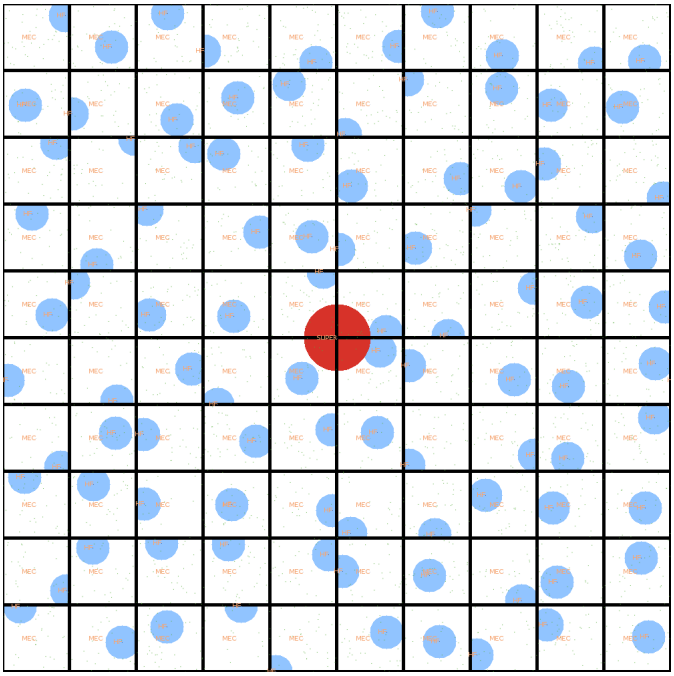


Figure 29: The simulation environment with rbtd enhanced lockdown

In this experiment, the rapid biological threat detection (RBTD) model is used with the normal lockdown (NL) model. RBTD is used for the computation of the suspicion scores of each cell, with cells having the highest suspicion score assumed as cells carrying more infected humans.

Although health facilities can determine whether humans are infected even if

they are not showing symptoms, they have a limited area of observation within their cell, amounting only for up to 20 percent of the total cell area. The RBTD further helps to analyze the situation of infection on cells since it can detect symptoms of humans within the simulation. Whenever the RBTD detects a symptomatic human, it already counts that human as an infected.

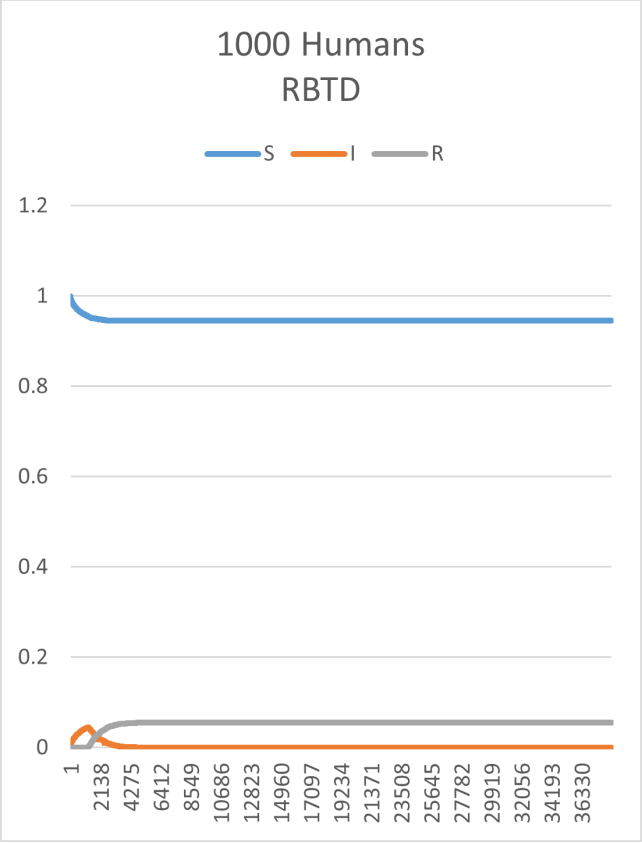


Figure 30: Average result for the simulation of 1000 humans with RBTD enhanced lockdown

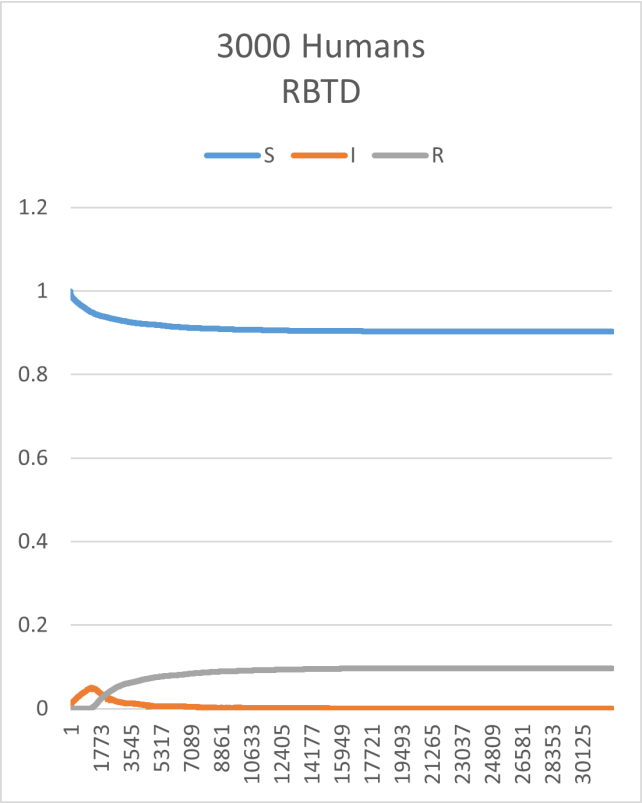


Figure 31: Average result for the simulation of 3000 humans with RBTD enhanced lockdown

Figure 30 and 31 shows the graph of the SIR of the 1000 humans and 3000 humans respectively. We may notice the flatness of the curves as time progresses, this indicates that the method is effective in preventing the spread of the disease.

	1000	3000
Ticks	38450	31880
S	94.52%	90.37%
I	0.00%	0.00%
R	5.48%	9.63%
Infection Purged (ticks)	5463	19519

Figure 32: The resulting SIR values for 1000 and 3000 humans in their respective timeframes for the environment with RBTD lockdown

In figure 32, it shows a section of infection purged. This is the specific time tick where it has successfully isolated the area of infection in the environment. Since no humans are being infected at this time, the suspicion scores of the cells in the environment shall remain constant and the last flagged cells by this method will be constantly flagged every week, hence preventing humans outside from entering these most suspicious cells.

13 Results Comparison

1000 humans at 38450 ticks (54.9 weeks)							
	S	I	R	Effectiveness compared to No Lockdown S	I	R	Disease Contained
No Lockdown	51.20%	1.08%	47.72%	0.00%	0.00%	0.00%	No
Lockdown	61.72%	1.06%	37.22%	20.55%	-1.85%	-22.00%	No
RBTD Lockdown	94.52%	0.00%	5.48%	84.61%	-100.00%	-88.52%	5463 ticks (7.8 weeks)
3000 humans at 31880 ticks (45.5 weeks)							
	S	I	R	Effectiveness compared to No Lockdown S	I	R	Disease Contained
No Lockdown	50.79%	1.29%	47.92%	0.00%	0.00%	0.00%	No
Lockdown	66.76%	0.95%	32.29%	31.43%	-26.42%	-32.61%	No
RBTD Lockdown	90.37%	0.00%	9.63%	77.92%	-100.00%	-79.91%	19519 ticks (27.9 weeks)

Figure 33: The comparison of the three experiments

Interpreting the results found in Figure 33, we can observe that for 1000 humans: after a little more than a year (54.9 weeks), the no lockdown simulation did not contain the disease with only 51.2% of the starting population left, 1.08 % are still infected and 47.72% have either recovered or died by the disease. In the second row, we can see the effects of using NPIs in slowing down the spread of the disease, as 61.72% of the population are still healthy (10.5% more than the no lockdown simulation), 1.06% still infected and 37.22% have either died or recovered from the disease.

On the other hand, the results of the RBTD lockdown were significantly more effective in slowing down the spread of the disease (and successfully containing it) because in just 7.8 weeks, the disease was already contained and 0% of the humans are still infected by the disease in a short amount of time. This emanated in a 94.52% healthy population and only 5.48% of the population that were either dead or recovered from the disease.

However, as we can see from the second table (3000 humans) of Figure33, the effectiveness of the RBTD Lockdown decreases by a small margin as the population increases. One of the reasons for this result is that the simulation took longer to contain the disease (27.9 weeks) as more humans mean that it might take longer for a pattern to emerge (and therefore locate the origin of the disease) as more suspicious cells will arise, taking more time for the origin of disease to come out on top as the most suspicious cells. This means that the origin of the disease would not be lockdown just as fast compared to the 1000-human simulations. Furthermore, as the population increased in a same-sized environment, the number of infections transmitted to other healthy humans would also increase, as there is just less space for the agents to move and therefore there is a higher chance of infecting others. This led to a 6.69% decrease in the healthy population with only 90% of the original population left

(2711 humans out of 3000) compared to 94.52% of the original 1000 humans for the first table (945 humans).

14 Improvements

The incubation period of the disease plays a vital role for determining the accuracy of the rapid biological threat detection model on ranking suspicious cells. The experiment made use of a fixed value, hence, making the RBTD model more accurate. However, also take note that the original RBTD model did not use this incubation period and achieved high results of accuracy.

The value for the day is also fixed to 1 cell dimension. Further experiment may also be made to different day values such as 2 cell dimension or other values other than 1 to see if results may vary.

Additionally, this experiment may have more potential uses if incubation period and recovery rates are not manually set, and are instead generated dynamically. This modification is useful, specifically for new threats since data regarding them are still non-existent.

References

[1] A. Atalan. Is the lockdown important to prevent the covid-19 pandemic? effects on psychology, environment and economy-perspective. *Annals of Medicine and Surgery*, 2020.

[2] M.; Aleem A.; Dulebohn S.C.; Di Napoli R. Cascella, M.; Rajnik. *Features, Evaluation, and Treatment of Coronavirus (COVID-19)*. 2022.

[3] CT.; Pairea J.; Rolland P.; Fontanet A.; Cauchemez, S.; Kiem. Lockdown

impact on covid-19 epidemics in regions across metropolitan france. *Lancet*, 2020.

[4] He.; Chang-Tai H.; Chen, Q.; Zhiguo and Z.; Song. Economic effects of lockdown in china. *mpact of COVID-19 on Asian Economies and Policy Responses*, pages 3–10, 2020.

[5] CISCO. *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are*. 2015.

[6] A. Downey. *Think Complexity, 2nd Edition*. 2018.

[7] Spraggon B. Martino-M. Elvery, S. *The one Covid-19 number to watch*. 2020.

[8] Society for Industrial and Applied Mathematics. What is math modeling? video series part 1: What is math modeling?, *Youtube*. https://www.youtube.com/watch?v=xHtsuOB-TPw, 2016.

[9] B.S. Everitt G. Dunn. *An Introduction to Mathematical Taxonomy*. 1982.

[10] et. al Gollwitzer M. Public acceptance of covid-19 lockdown scenarios. *International Journal of Psychology*, 2020.

[11] Columbia Public Health. Agent-based modeling. https://www.publichealth.columbia.edu/research/population-health-methods/agent-based-modeling, 2021.

[12] V.; Mikolajczyk A.; Schubert J.; Bania J.; Khosrawipour T.; Lau, H.;Khosrawipour. The positive impact of lockdown in wuhan on containing the covid-19 outbreak in china. *Journal of Travel Medicine*, 27:800–801, 2020.

[13] K.; Bi Q.; Jones F.; Zheng Q.; Meredith H.; Azman A.; Reich N.; Lessler J. Lauer, S.; Grantz. *The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application*. 2020.

[14] InfoSys Limited. *Edge Computing*. 2019.

[15] E.D. Gennatas M. Bielecki C. Beyrer G. Rutherford H. Chambers E. Goosby Monica Gandhi M.A. Spinelli, D.V. Glidden. *Importance of non-pharmaceutical interventions in lowering the viral inoculum to reduce susceptibility to infection by SARS-CoV-2 and potentially disease severity*. 2021.

[16] MegaInstructorX. Introduction to modeling simulation, *Youtube*. https://www.youtube.com/watch?v=od7fWoKjV6E, 2017.

[17] MITRE. Verification and validation of simulation models. https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/other-se-lifecycle-building-blocks-articles/verification-and-validation-of-simulation-models.

[18] Yaser Jararweh Mohammad Al-Zinati, Qutaibah Al-Thebyan. *An Agent Based Model for Health Surveillance Systems and Early Biological Threat Detection*. 2018.

[19] JT.; Phologolo-T.; Hamda SG.; Masupe T.; Tsima B.; Setlhare V.; Mashalla Y.; Wiebe DJ.; Molefi, M.; Tlhakanelo. The impact of china's lockdown policy on the incidence of covid-19: An interrupted time series analysis. *Biomed Res Int*, 2021.

[20] David NatlCtrSim. Modeling and simulation 101, *Youtube*. https://www.youtube.com/watch?v=M0iZ52kUOiQ, 2010.

[21] National Institute of Biomedical Imaging and Bioengineering. Computational modeling. https://www.nibib.nih.gov/science-education/science-topics/computational-modeling.

[22] Palisade. Monte-carlo simulation. https://www.palisade.com/risk/monte_carlo_simulation.asp.

[23] Jackson S. Railsback S., Lytinen S. Agent-based simulation platforms: Review and development recommendations. *SIMULATION*, 2006.

[24] William Rand. Agent-based modeling: What is agent-based modeling?, *Youtube*. https://www.youtube.com/watch?v=FVmQbfsOkGc, 2018.

[25] D.; Saadat, S.; Rawtani and C.M.; Hussain. Environmental perspective of covid-19. *Science of the Total Environment*, 728, 2020.

[26] Jeff Schank. Agent-based models. http://www.agent-based-models.com/blog/resources/agent-based-models/, 2010.

[27] Jeff Schank. Agent-based models - simulators. http://www.agent-based-models.com/blog/resources/simulators/, 2010.

[28] R.; Lambert H.; Oliver I.; Robin C.; Yardley L.; Rubin G.J.; Smith, L.E.; Amlt. *Factors associated with adherence to self-isolation and lockdown measures in the UK: a cross-sectional survey*. 2020.

[29] Statisticshowto. Jaccard's index / similarity coefficient. https://www.statisticshowto.com/jaccard-index/#:~:text=Count%20the%20number%20of%20members,in%20(3)%20by%20100.

[30] Leigh Tesfatsion. Agent-oriented programming: Intro. http://www2.econ.iastate.edu/tesfatsi/AOPRePast.pdf, 2021.

[31] B.; Gebre D.S.; Atomssa E.M.; Getachew M.; Fetensa G.; Ayala D.; Turi E. Tolossa, T.; Wakuma. *Time to recovery from COVID-19 and its predictors among patients admitted to treatment center of Wollega University Referral Hospital (WURH), Western Ethiopia: Survival analysis of retrospective cohort study.* 2021.

[32] H. Weiss. *The SIR model and the Foundations of Public Health.*

[33] Dario Sabella Nurit Sprecher Valerie Young Yun Chao Hu, Milan Patel. *Mobile Edge Computing A key technology towards 5G.* 2015.

[34] J.; Wang X.; Yang J.; Liu X.F.; Xu X.K. Zhang, L.; Zhu. *Characterizing COVID-19 Transmission: Incubation Period, Reproduction Rate, and Multiple-Generation Spreading.* 2021.

[35] RELJIN I. RELJIN B. AJA, N. *Telemonitoring in Cardiology - ECG Transmission by Mobile Phone.* 2001.