DISSERTATION

Surrogate Assisted Strategies for the Parameterisation of Infectious Disease Agent–Based Models

Author:
Rylan PERUMAL (1396469)

Supervisor:
Prof. Terence VAN ZYL

UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

submitted to
the Faculty of Science, in fulfilment of the requirements for the degree of
Master of Science

in the

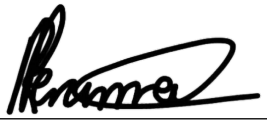
School of Computer Science and Applied Mathematics

December 7, 2021

Declaration of Authorship

I, Rylan PERUMAL (1396469), declare that this dissertation titled, "Surrogate Assisted Strategies for the Parameterisation of Infectious Disease Agent–Based Models" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this dissertation has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the dissertation is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 08/12/2021

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

Abstract

Faculty of Science
School of Computer Science and Applied Mathematics

Master of Science

Surrogate Assisted Strategies for the Parameterisation of Infectious Disease Agent–Based Models

by Rylan PERUMAL (1396469)

Agent-based modelling and simulation (ABMS) is a viable solution for real-time decision analysis and policy-making towards preventing an epidemic. This is due to its ability to model the real world by incorporating more complexity than previously used compartmental models. Model validation, parameter calibration and long simulation time are significant limitations of ABMS. Surrogate models (SMs) can overcome these limitations. However, there is a lack of comprehensive comparison between surrogate assisted parameterisation strategies. In addition, there is a lack of research on using an SM to tackle problems outside of parameterisation. We provide a comparison of some state-of-the-art and classical intelligent sampling, optimisation and evolutionary methods for parameter calibration along with a framework for evaluation of these methods. The extensive experimental results show that the Dynamic Coordinate Search Using Response Surface Models paired with the XGBoost SM outperforms competing methods regarding accuracy and speedup achieved on synthetic epidemic data. Parameterising an ABM taking an average similarity score across each output distribution allows the parameterisation approach to more closely match real-world data. Lastly, we have shown that a Long Short-Term Memory network SM can replicate the transmission dynamics of a complex ABM, significantly reducing simulation time whilst maintaining accuracy.

Acknowledgements

I want to thank my supervisor, Professor van Zyl, for his help, time and insightful weekly meetings. He is a fantastic supervisor, and his guidance and knowledge have been invaluable.

I thank my family and friends for all of their support during the past year and a half. A special thanks goes out to my partner for all of her support, including proofreading my work to ensure it is grammatically sound.

Lastly, I thank the National Research Foundation (NRF) for funding my research. The financial assistance of the NRF towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the NRF.

Publications

1. R. Perumal and T. L. van Zyl, "Surrogate Assisted Methods for the Parameterisation of Agent-Based Models," 2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI), 2020, pp. 78-82, doi: 10.1109/ISCMI51676.2020.9311568.

2. R. Perumal and T. L. van Zyl, "Surrogate Assisted Strategies (The Parameterisation of an Infectious Disease Agent–Based Model)", 2021. (**Under Review**)

Contents

Declaration of Authorship i

Abstract ii

Acknowledgements iii

Publications iv

1 Introduction 1
- 1.1 Background . . . 2
  - 1.1.1 Introduction . . . 2
  - 1.1.2 Agent-Based Modelling and Simulation in Epidemiology . . . 2
  - 1.1.3 Surrogate Models in Agent-Based Modelling and Simulation . . . 3
  - 1.1.4 Surrogate Assisted Optimisation . . . 4
  - 1.1.5 Evolutionary Algorithms . . . 5
  - 1.1.6 Recurrent Neural Networks . . . 5
  - 1.1.7 Conclusion . . . 6
- 1.2 Problem Statement . . . 8
- 1.3 Significance and Motivation . . . 8
- 1.4 Research Aims and Objectives . . . 8
- 1.5 Research Questions . . . 9
- 1.6 Delineations, Limitations and Assumptions . . . 10
- 1.7 Outline . . . 10

2 Surrogate Assisted Sampling Strategies 11
- 2.1 Introduction . . . 11
- 2.2 SIRD Agent-Based Model . . . 11
- 2.3 Approximate Two-Sample Kolmogorov-Smirnov Test . . . 13
- 2.4 Agent-Based Modelling and Simulation Framework . . . 13
- 2.5 Sampling Methods . . . 14
  - 2.5.1 Random Sampler . . . 14
  - 2.5.2 Surrogate Assisted Random Sampler . . . 14
  - 2.5.3 Quasi-Random Sobol Sampler . . . 14
  - 2.5.4 Surrogate Assisted Quasi-Random Sobol Sampler . . . 16
- 2.6 Surrogate Models . . . 16
- 2.7 Sanity Check . . . 16
- 2.8 Experiment Setup . . . 17
  - 2.8.1 Hardware Specifications . . . 17
- 2.9 Results and Discussion . . . 17
- 2.10 Conclusion . . . 20

3 Comparison of Surrogate Assisted Parameterisation Strategies **21**
3.1 Introduction . . . 21
3.2 Improved Agent-Based Modelling and Simulation Framework . . . 21
3.3 Surrogate Assisted Optimisation Strategies . . . 24
3.3.1 Metric Stochastic Response Surface . . . 24
3.3.2 Dynamic Coordinate Search Using Response Surface Models . 24
3.4 Surrogate Assisted Sampling Strategies . . . 25
3.4.1 Surrogate Assisted Random Sampler . . . 25
3.4.2 Surrogate Assisted Quasi-Random Sobol Sampler . . . 25
3.5 Covariance Matrix Adaptation Evolutionary Strategy . . . 25
3.6 Surrogate Models . . . 26
3.7 Experiment Setup . . . 26
3.7.1 Hardware Specifications . . . 26
3.8 Results and Discussion . . . 27
3.9 Conclusion . . . 30

4 SIRD vs Complex Agent-Based Model **31**
4.1 Introduction . . . 31
4.2 Data . . . 31
4.2.1 Oxford COVID-19 Government Response Tracker . . . 31
4.2.2 COVID-19 Data Repository . . . 33
4.2.3 Google Trends Dataset . . . 35
4.3 Complex Agent-Based Model . . . 35
4.4 Average Kolmogorov-Smirnov Test Statistic . . . 36
4.5 Experiment Setup . . . 37
4.5.1 Hardware Specifications . . . 38
4.6 Results and Discussion . . . 38
4.7 Conclusion . . . 42

5 Surrogate Agent-Based Model **45**
5.1 Introduction . . . 45
5.2 Data . . . 45
5.3 Long Short-Term Memory Recurrent Neural Network . . . 45
5.4 Experiment Setup . . . 46
5.4.1 Hardware Specifications . . . 47
5.4.1.1 Experiment 1 . . . 47
5.4.1.2 Experiment 2 . . . 47
5.5 Results and Discussion . . . 48
5.6 Conclusion . . . 51

6 Conclusion **52**

Bibliography **54**

List of Figures

1.1 Diagram representing the Long Short-Term Memory (LSTM) unit [43]. 7

2.1 Susceptible–Infected–Recovered–Dead Transmission Dynamics. 12
2.2 Agent-based modelling and simulation (ABMS) framework used to evaluate the different surrogate assisted sampling strategies. 15
2.3 Comparison between the KSTS values of the different surrogate assisted random sampling methods implemented and the number of parameters being estimated. 19
2.4 Comparison between the KSTS values of the different surrogate assisted sobol sampling methods implemented and the number of parameters being estimated. 19

3.1 Improved agent-based modelling and simulation (ABMS) framework used to evaluate the different parameterisation strategies implemented. 23

4.1 CSSE dataset containing the confirmed COVID-19 cases (infections), recoveries and deaths for South Africa between dates 22/01/2020–02/07/2021. 33
4.2 Time Period 1 – Pre-processed CSSE dataset containing the number of daily COVID-19 cases (infections), recoveries and deaths for South Africa between dates 05/10/2020–02/01/2021. 34
4.3 Time Period 2 – Pre-processed CSSE dataset containing the number of daily COVID-19 cases (infections), recoveries and deaths for South Africa between dates 03/01/2021–02/04/2021. 34
4.4 Time Period 3 – Pre-processed CSSE dataset containing the number of daily COVID-19 cases (infections), recoveries and deaths for South Africa between dates 03/04/2021–02/07/2021. 34
4.5 Google Trends Data using the search term *covid19*, across the time period 05/10/2020–02/01/2021 for South Africa. 35
4.6 Google Trends Data using the search term *covid19*, across the time period 03/01/2021–02/04/2021 for South Africa. 35
4.7 Google Trends Data using the search term *covid19*, across the time period 03/04/2021–02/07/2021 for South Africa. 36
4.8 Not Informed-Interested-Uninterested Transmission Dynamics. 37
4.9 Performance comparison plot between the single KS Test and the average KS Test for time period 05/10/2020–02/01/2021. 41
4.10 Performance comparison plot between the single KS Test and the average KS Test for time period 03/01/2021–02/04/2021. 42
4.11 Performance comparison plot between the single KS Test and the average KS Test for time period 03/04/2021–02/07/2021. 42
4.12 Comparison between the real world and the predicted data distributions for South Africa using the Average model during the third time period. 43

4.13 Comparison between the real world and the predicted cumulative distributions for South Africa using the Average model during the third time period. 43
4.14 Comparison between the real world and the predicted data distributions for Egypt using the Average model during the second time period. 44
4.15 Comparison between the real world and the predicted cumulative distributions for Egypt using the Average model during the second time period. 44

5.1 Long Short-Term Memory Network Architecture. 47
5.2 Comparison between the training RMSE obtained by the LSTM network in Figure 5.1 for each kth fold, using early stopping. 48
5.3 Comparison between the validation RMSE obtained by the LSTM network in Figure 5.1 for each kth fold, using early stopping. 48
5.4 Data distribution of parameterising the LSTM network to an epidemic simulation of the optimal prediction for Egypt during the time period 03/01/2021 – 02/04/2021. 49
5.5 Cumulative distribution of parameterising the LSTM network to an epidemic simulation of the optimal prediction for Egypt during the time period 03/01/2021 – 02/04/2021. 50

List of Tables

1.1 Agent-based epidemiological simulation models identified within the literature. . . . 3

2.1 Table of the ranges for each parameter value of the ABM that we have considered for parameterisation. Parameters $1, 2, 3$, and 6 are sampled between the range $(0, 1)$. Parameter 7 is sampled between the range $(0, 0.022)$ to mimic real world interaction as the space defined is bounded by $(1, 1)$. Parameters 4 and 5 are sampled between the range $(0, 41)$ days. . . . 12
2.2 Standardised L_2 Norm values for the optimal predicted parameter vectors using each of the strategies and SMs implemented. The optimal values are marked in bold. . . . 18
2.3 Kolmogorov-Smirnov Test Statistic (KSTS) values for the optimal predicted parameter vectors using each of the strategies and SMs implemented. The optimal values are marked in bold. . . . 18
2.4 Number of mini-batch evaluations on average to success at (97%, 97.5%, 98% and 98.5%) for estimating 7 ABM parameters. The optimal values are marked in bold. . . . 19

3.1 Table of the ranges for each parameter value of the infectious disease ABM, presented in Chapter 2, that we have considered for parameterisation. Parameters $1, 2, 3$, and 6 are sampled between the range $(0, 1)$. Parameter 7 is sampled between the range $(0, 0.022)$ to mimic real world interaction as the space defined is bounded by $(1, 1)$. Parameters 4 and 5 are sampled between the range $(0, 41)$ days. . . . 27
3.2 Standardised L_2 Norm values for the optimal predicted parameter vectors using each of the strategies and SMs implemented. The top three values have been highlighted varying in intensities of grey, the darkest grey represents the best value and the lightest grey represents the third best value. . . . 27
3.3 Kolmogorov-Smirnov Test Statistic (KSTS) values for the optimal predicted parameter vectors using each of the strategies and SMs implemented. The best KSTS values for each parameter are marked in bold. 28
3.4 Probability of reaching success and the speedup acquired within 98% and 99% of the optimal for seven parameters. The optimal stratey surrogate combination in terms of average accuracy and speedup are highlighted. In addition, the highest average accuracy and speedup are marked in bold. . . . 29

4.1 Table of containment and closure policy indicators used to calculate the speed restriction index. . . . 32
4.2 Table of health system policy indicators used to calculate the interaction radius restriction index. . . . 33

4.3 Table of the ranges for each parameter value of the ABM that we have considered for parameterisation. Parameters $1, 2, 4, 5, 9, 10$ and 11 are sampled between the range $(0, 1)$. Parameters 7 and 8 are sampled between the range $(0, 0.022)$ to mimic real world interaction as the space defined is bounded by $(1, 1)$. Parameters 3 and 6 are sampled between the range $(0, 90)$ days. Lastly, parameters 12 and 13 are sampled between $(1,\ N_{Agents})$, where N_{Agents} represents the number of agents within the modelled population. 37
4.4 Table of the countries we have used in our experiment along with their confirmed number of COVID-19 cases to date (02/07/2021). . . . 38
4.5 Comparison of KSTS values between the SIRD ABM and the Complex ABM, calibrating the models towards number of daily infections, between the time periods 05/10/2020 – 02/01/2021 (Time Period 1), 03/01/2021 – 02/04/2021 (Time Period 2) and 03/04/2021 – 02/07/2021 (Time Period 3). The time periods and corresponding countries where the SIRD model was optimal is highlighted. In addition, we have marked the optimal average for each time period in bold. 39
4.6 Comparison of KSTS values between the single KS Test and the average KS Test for the time period 05/10/2020 – 02/01/2021. The average optimal values are marked in bold. 40
4.7 Comparison of KSTS values between the single KS Test and the average KS Test for the time period 03/01/2021 – 02/04/2021. The average optimal values are marked in bold. Additionally, the optimal results are highlighted in grey. 40
4.8 Comparison of KSTS values between the single KS Test and the average KS Test for the time period 03/04/2021 – 02/07/2021. The average optimal values are marked in bold. Additionally, the best and worst results are highlighted in dark and light grey, respectively. 41

5.1 Table of the ranges for each parameter value of the ABM that we have considered for parameterisation. Parameters $1, 2, 4, 5, 9, 10$ and 11 are sampled between the range $(0, 1)$. Parameters 7 and 8 are sampled between the range $(0, 0.022)$ to mimic real world interaction as the space defined is bounded by $(1, 1)$. Parameters 3 and 6 are sampled between the range $(0, 90)$ days. Lastly, parameters 12 and 13 are sampled between $(1,\ N_{Agents})$, where N_{Agents} represents the number of agents within the modelled population. 46
5.2 Comparison of the KSTS values between the using the Complex ABM and the LSTM within the ABMS framework for Egypt during the time period 03/01/2021 – 02/04/2021. 50
5.3 Comparison between the Complex ABM and the LSTM network for the time taken in seconds for each part of the ABMS framework that deals with parameter evaluation. 50
5.4 Comparison between the speed up achieved by the LSTM as a replacement for the Complex ABM within the ABMS framework. 51

List of Abbreviations

ABMS	**A**gent–**B**ased **M**odelling (and) **S**imulation
ABM(s)	**A**gent–**B**ased **M**odel(s)
SM(s)	**S**urrogate **M**odel(s)
SOTA	**S**tate–**O**f–**T**he–**A**rt
ML	**M**achine **L**earning
SIRD	**S**usceptible–**I**nfected–**R**ecovered–**D**ead
EA(s)	**E**volutionary **A**lgorithm(s)
RMSE	**R**oot **M**ean **S**quare **E**rror
LSTM	**L**ong **S**hort–**T**erm **M**emory
KSTS	**K**olmogorov–**S**mirnov **T**est **S**tatistic

Chapter 1

Introduction

Epidemiological research helps us to understand the behavioural dynamics of an infectious disease. In particular, we can understand how many people have a specific infectious disease, how the numbers in the active infections are changing, the effect on society and the economy and what preventative strategies can be implemented [1]. For these reasons, it is imperative to model and understand infectious disease dynamics at an accelerated pace to help make informed decisions.

Previously, compartmental (mathematical) models have been used to analyse how an infectious disease affects a population. However, they suffer from a lack of complexity due to the assumption that the entire population is homogeneous. While these types of models do have the ability to model sub-populations, the additional complexity is exceedingly difficult to understand and solve [2]–[4].

Agent–based modelling and simulation (ABMS), on the other hand, models a population at an individual level, allowing for population heterogeneity. Agents are given different attributes and can make independent decisions within their environment based on rules. This allows for real-world level complexity to be incorporated into a model. ABMS aims at enriching our understanding of how unknown phenomena occurs in the real world. However, ABMS suffers from model validation, parameter calibration and long simulation time [2], [3], [5].

Surrogate models (SMs), learnt through machine learning (ML) algorithms, can overcome some of the limitations of ABMS. SMs can search the parameter space of an agent–based model (ABM) much more efficiently by ignoring areas where discontinuities and local optima occur. This reduces model development time, which is extremely useful when considering ABMS for policy-making and decision analysis during an epidemic [5]. In addition, SMs can also function as computational approximations to the complex simulation models [6].

This research develops an ABMS framework to evaluate how intelligent sampling methods can parameterise an infectious disease ABM. In order to make our analysis more robust, an improved ABMS framework is developed to evaluate different surrogate assisted parameterisation strategies in terms of accuracy and speedup. A Complex ABM is produced, which builds on the complexity of the Susceptible–Infected–Recovered–Dead (SIRD) ABM. The Complex ABM can incorporate sufficient complexity without increasing simulation time. The SIRD and Complex ABMs are evaluated to approximate the real-world data distributions for the COVID-19 virus. In addition, we compare the approximation accuracy of the similarity mechanism using a single output distribution versus all output distributions to understand whether incorporating more than one is necessary. Lastly, we analyse using an SM

to replace the simulation model within the ABMS framework. The purpose of using the SM is to reduce the long simulation time of the ABM whilst still maintaining accuracy.

1.1 Background

1.1.1 Introduction

In this section, we investigate the necessary background in order to conduct the research. Section 1.1.2 reviews agent-based modelling and simulation (ABMS) used in epidemiology. Section 1.1.3 considers surrogate models used in ABMS. Section 1.1.4 looks into surrogate assisted optimisation strategies. Section 1.1.5 briefly covers evolutionary algorithms. Lastly, Section 1.1.6 investigates recurrent neural networks (RNNs) and their potential use case within the research.

1.1.2 Agent-Based Modelling and Simulation in Epidemiology

Agent–based modelling and simulation (ABMS) is an effective and natural fit for modelling infectious diseases. Agent–based models (ABMs) are capable of modelling interactions between individuals and their environment. In addition, they are able to capture unexpected emergent patterns and trends during an epidemic that results from collective individual agent behaviours and interactions [2]. Each agent within an ABM can have different characteristics, more closely representing the variation in the human population. The agents act autonomously, governed by the combination of their set of pre-defined rules and distinctive characteristics. This autonomy allows ABMs to simulate many complex real-world scenarios with sufficient fidelity [7]. ABMs can also be used as a substitute for a real-world epidemiological study since it can often be infeasible or even impossible to run a real-world experiment [8], [9].

Compartmental (mathematical) models, like the Susceptible–Infected–Recovered (SIR) framework proposed by Kermack and McKendrick [10] and its variants, are not able to capture the various complexities of the real world. These models also assume that the population model is homogeneous. Population homogeneity is a significant modelling disadvantage as individuals behavioural patterns, demographics and health conditions are different. Additional complexity can be introduced into these models. However, the overall model becomes difficult to understand and solve [2].

Table 1.1 contains the epidemiological simulation models identified in the literature and the respective research focus. Aleman, Wibisono, and Schwartz [11] models the spread of influenza in the Greater Toronto Area in Ontario, Canada. The results from their model are output to a geographic information system, where they show it can be used as a policy-making tool for implementing preventative measures. Mao [12] models a triple-diffusion process in a metropolitan area of one million people. The triple-diffusion incorporates the transmission of the disease, the flow of information regarding the disease and the spread of human preventative behaviours against the disease. The results shown indicate that the model reasonably replicates the trends of influenza spread and information propagation. Cooley, Brown, Cajka, *et al.* [13] models the spread of influenza within New York City (NYC), focusing on the role that the subway plays in its spread. The model was calibrated using historical data

from the influenza pandemic during 1957–1958. Their results show that if influenza did occur in NYC, 4% of the transmission would occur on subways. Hackl and Dubernet [14] models the spread of seasonal influenza outbreaks in the metropolitan area of Zurich, Switzerland. Their results show that the ABM captures the trend of the data much better than a SIR model. In addition, they show that the understanding of disease spread dynamics can be improved, and better steps can be taken toward the prevention and control of an epidemic.

TABLE 1.1: Agent-based epidemiological simulation models identified within the literature.

Paper	Research Focus
Aleman, Wibisono, and Schwartz [11]	Epidemic Planning
Mao [12]	Disease Dynamics Research
Cooley, Brown, Cajka, *et al.* [13]	Epidemic Planning
Hackl and Dubernet [14]	Epidemic Planning

While there are significant benefits to using ABMs for infectious disease epidemiology, there are equally limitations. ABMs ordinarily require long run times due to the increased computational complexity resulting from agent interactions incorporated into the model [5], [6]. Additionally, model validation and parameterisation present significant challenges within the field of ABMS, precisely when matching real-world data [3]. Model validation is the process of checking that the model, combined with its assumptions, provides a sufficiently accurate representation of the real world. There is no general way of validating ABMs. However, researchers have been validating their models by comparing the outputs of ABMs to the corresponding real-world data. Due to data restrictions and privacy issues, it is not always possible to attain the data that was used to validate the model [2]. Of these challenges, we are particularly interested in addressing parameter calibration and the long simulation time of ABMs. Difficulty finding correct parameter combinations for ABMs lead to extensive calibration efforts resulting in increased model development time. As more complexity is added to the model, the parameter space expands, leading to the ABMS equivalent of the "curse of dimensionality" problem. The outcome is impractical memory and computational costs when searching for meaningful parameter combinations [4], [15].

1.1.3 Surrogate Models in Agent-Based Modelling and Simulation

Surrogate models (SMs) are generated using machine learning (ML) algorithms, and they can function as computational approximations to ABMs. SMs can reduce the dimensionality of an ABM's parameter vector, simplifying its form while still maintaining the original dynamical characteristics. In addition, SMs are capable of reducing the computational time required for parameter space exploration for high dimensional ABMs [5]. SMs provide computationally tractable solutions addressing parameter sensitivity analysis, robust analysis and empirical validation in ABMS. These properties make SMs appealing when dealing with significantly complex ABMs that are computationally expensive to validate and calibrate [6], [16].

The Kriging method, also known as Gaussian Process Regression, has been used as a surrogate modelling approach to facilitate the parameter space exploration and sensitivity analysis challenges in ABMS. This method estimates the model's response

over the entire parameter space from a finite set of model evaluations to generate the best unbiased linear predictor corresponding to the true variogram[1]. However, the spatial distribution of the data is rarely known when dealing with ABMs. This leads to a large number of model evaluations, increasing the size of the parameter space. Unfortunately, Kriging's performance is dependent on the model's ability to estimate the true variogram [5], [6], [17], [18].

Lamperti, Roventini, and Sani [5] presents a new approach to overcome the limitations of the Kriging method. An iterative algorithm is proposed for learning an SM to approximate the ABM effectively. This is achieved by combining ML and intelligent iterative sampling. Furthermore, it is shown that the model's parameter space can be effectively searched using fewer computational resources. The XGBoost[2] ML algorithm is used, where the SM is built in a stage-wise fashion, allowing optimisation of an arbitrary differentiable loss function. This method is applied to the Asset Pricing Model by Brock and Hommes [19] and the Island Growth model by Fagiolo and Dosi [20]. The results obtained show that the SM is an accurate proxy of the ABM. Furthermore, it dramatically reduces the computation time for large-scale parameter space calibration and exploration. Zhang, Li, and Zhang [6] improves on the work of Lamperti, Roventini, and Sani [5] using the CatBoost[3] ML algorithm, where it is shown that the surrogate is able to approximate the ABM and reduce parameter calibration and exploration time. We aim to enrich our understanding of the trade-off between increased speed and change in accuracy when using SMs in ABMS.

1.1.4 Surrogate Assisted Optimisation

Many real-world optimisation problems involve high-dimensional black-box functions that are outputs of computationally expensive simulations. Generally, finding the global optimum of these problems is unrealistic as it requires a significant amount of function evaluations [21]. It is often the case that the derivatives of these black-box functions are not available. Therefore, derivative-free strategies have been developed to allow for their optimisation. In particular, we are interested in derivative-free optimisation and derivative-free heuristic methods. A common approach to derivative-free optimisation is the use of surrogate models (SMs) or metamodels. Regis and Shoemaker [22] present a method for the global optimisation of expensive multimodal functions, where a response surface model (radial basis function and neural network) is used as an SM. The method iteratively uses the surrogate to approximate the output of the expensive multimodal function and then selects the best potential candidate. The candidate is selected based on two criteria: the estimated response from the SM and the minimum distance to the previously evaluated points. The results indicate that this method is a promising solution for the global optimisation of expensive high-dimensional multimodal functions. Regis and Shoemaker [21] combines a radial basis function SMs and dynamic coordinate search for the global optimisation of computationally expensive functions, which extends the previous work presented by Regis and Shoemaker [22]. Regis and Shoemaker [21] presents two algorithms, where it is shown that the combination of dynamic coordinate search and SMs improves classical approaches, especially for high dimensional problems. The surrogate assisted optimisation approaches mentioned

[1] A variogram is the description of the spatial continuity of the data.

[2] https://xgboost.readthedocs.io/en/latest/

[3] https://catboost.ai/docs/concepts/about.html

above are appealing as they provide an abstraction towards addressing the parameterisation challenges in ABMs. Specifically, in epidemiology, the accurate and efficient parameterisation of infectious disease models is imperative.

1.1.5 Evolutionary Algorithms

Evolutionary algorithms (EAs) are inspired by Darwinian evolutionary theory. These algorithms make up the field known as Evolutionary Computing, composed of the following subareas: evolutionary programming, evolutionary strategies, genetic algorithms and genetic programming. All EAs follow a common underlying idea. Given a population of individuals within an environment with limited resources, the process of natural selection occurs when individuals compete for these resources. This causes a rise in the fitness of a population. The fitness of a population corresponds to how well individuals survive and reproduce. Given an arbitrary function to be maximised, we can randomly create candidates, where candidates are solution elements within the function's domain. The quality of these candidates is measured using the arbitrary function as an abstract measure. Superior candidates are chosen to seed the next generation based on the fitness values through recombination or mutation. Recombination is an operator applied to two candidates producing one or more new candidates, and mutation is applied to one candidate producing one new candidate. Applying the recombination or mutation operators on a selected candidate (parent) leads to creating a set of new candidates (children). The fitness of the new candidates is evaluated, and then they compete with the old candidates for a place in the next generation. This process is repeated until sufficient candidates (solutions) are found or the computational limit is reached [23].

A major limitation in ABMS is parameter calibration [3], [6]. EAs, in particular evolutionary strategies, presents a plausible way to search for optimal parameters within simulation models [24]. The current state-of-the-art (SOTA) in evolutionary algorithms is the $(1 + (\lambda, \lambda))$ genetic algorithm (GA), where crossover is used to increase the speed in which the parameter space is explored [25]. Doerr and Doerr [26] presents a refined run-time analysis of the $(1, +(\lambda, \lambda))$ GA, where they show an improved upper bound.

1.1.6 Recurrent Neural Networks

Artificial neural networks (ANNs) are computational mechanisms that are designed based on the concept of how the human brain processes information [27]. ANNs have received much attention to date for tasks that include prediction, clustering and pattern recognition across multiple disciplines [28]. Recurrent neural networks (RNNs) are a particular type of ANNs with feedback connections. These connections allow information to persist in the network [29]. An RNN is made of up recurrent units, which contain a recurrent hidden state, where the activation for the given unit at each timestep is dependent on that of the previous timestep [30]. More formally, RNNs are able to handle a sequence of inputs that are of arbitrary length, where each input in the sequence has a relationship to the previous inputs within the sequence. It has been observed by Bengio, Simard, and Frasconi [31] that it is difficult to train RNNs to capture long term dependencies as gradients tend to vanish or explode. Therefore, gradient-based optimisation methods struggle. Thus, other approaches have been investigated to reduce the negative impacts of this problem. Two of the main approaches include the Long Short–Term Memory (LSTM) unit

proposed by Hochreiter and Schmidhuber [32] and, more recently, the Gated Recurrent Unit (GRU) proposed by Cho, Merrienboer, Bahdanau, *et al.* [33]. The LSTM and GRU are widespread in sequence-based problems like speech recognition, traffic flow prediction, text generation and wildfire spread modelling [34]–[37]. Of these two approaches, we are particularly interested in the LSTM unit.

LSTM architectures are effective at capturing long-term dependencies. They do not suffer from gradient optimisation problems that trouble simple RNNs. The main idea behind the LSTM architecture is a memory cell, which takes in a combination of the previous memory cell state and the current input as its input. The memory cell then determines the necessary information to retain and eliminate. In essence, the memory cell regulates the flow of information into and out of the LSTM unit [38], [39]. The LSTM architecture used in the literature, known as the vanilla LSTM, was originally described by Graves and Schmidhuber [40]. In Figure 1.1, we can see a diagrammatic representation of the vanilla LSTM unit. It has three gates, namely the forget gate f_t, update gate i_t and the output gate o_t. The LSTM unit, shown in Figure 1.1, is defined by the following equations:

$$\bar{f}_t = \sigma(W_f \cdot [\bar{h}_{t-1}, \bar{x}_t] + \bar{b}_f) \tag{1.1}$$

$$\bar{u}_t = \sigma(W_u \cdot [\bar{h}_{t-1}, \bar{x}_t] + \bar{b}_u) \tag{1.2}$$

$$\bar{o}_t = \sigma(W_o \cdot [\bar{h}_{t-1}, \bar{x}_t] + \bar{b}_o) \tag{1.3}$$

$$\bar{c}_t = \bar{f}_t \times \bar{c}_{t-1} + \bar{i}_t \times \bar{\tilde{c}}_t \tag{1.4}$$

$$\bar{\tilde{c}}_t = tanh(W_c \cdot [\bar{h}_{t-1}, \bar{x}_t] + \bar{b}_c) \tag{1.5}$$

$$\bar{h}_t = \bar{o}_t \times tanh(\bar{c}_t) \tag{1.6}$$

where W_f, W_u, W_0 and W_c and $\bar{b}_f$, $\bar{b}_u$, $\bar{b}_o$ and $\bar{b_c}$ are the weight matrices and bias vectors, respectively. In Equation 1.1, the forget gate calculates the amount of information required from the previous cell state, c_{t-1}, when computing the current cell state, c_t. In Equation 1.2, the update gates calculates how much of the candidate value, $\bar{\tilde{c}}_t$, is required in the current cell state calculation. In Equation 1.4, we can see the cell state at time t, which is dependent on the forget gate, the update gate, the previous cell state and the candidate cell state, $\bar{\tilde{c}}_t$, which is shown in Equation 1.5. Lastly, in Equation 1.6, we can see the output of the LSTM unit, $\bar{h}_t$, which is a filtered version of the cell state that is passed through a *tanh* activation multiplied by Equation 1.3 which is the output gate [41]. LSTM networks allow gradient based learning, which in turn allow gradients to be checked using finite difference methods making practical implementations of LSTM networks more reliable [37], [38], [42]. In addition, LSTM networks are capable of modelling long term dependencies, which make them a suitable proxy to replace a complex simulation model like an ABM.

1.1.7 Conclusion

Decision analysis and policy-making during an epidemic rely heavily on efficiently and accurately simulating complex simulations. Previously used compartmental models for simulating and understanding the spread of an epidemic do not include sufficient complexity to model the real world [2]. Adding more complexity to these models makes them difficult to understand and solve. Agent–based models (ABMs)

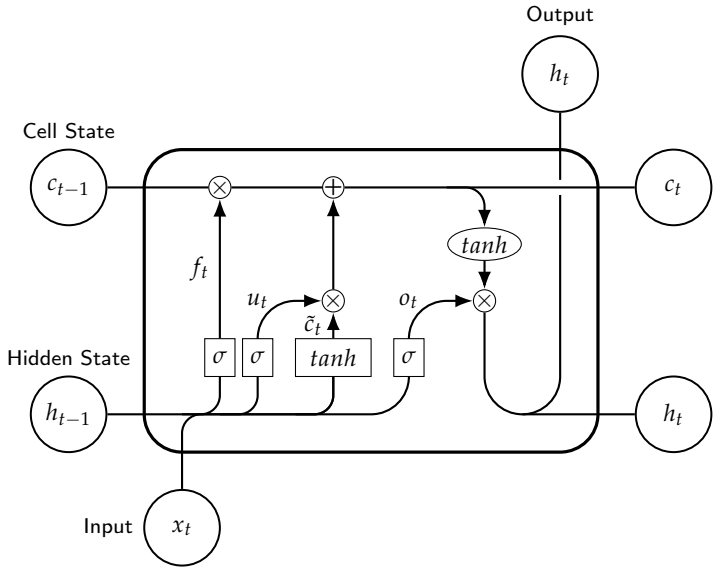


FIGURE 1.1: Diagram representing the Long Short-Term Memory (LSTM) unit [43].

are able to model an entire population at an individual level, making them significantly more complex than compartmental models [2], [3]. In this section, the relevant literature for agent–based modelling and simulation (ABMS) in epidemiology, surrogate models (SMs) used in ABMS, surrogate assisted optimisation, evolutionary algorithms and recurrent neural networks (RNNs) were presented. Due to the additional complexity, certain limitations are encountered in ABMs, such as model validation, parameter calibration, and long simulation time [5], [6]. These limitations can slow down the efforts of policy-makers to implement adequate preventative measures.

SMs have previously been used to reduce parameter space exploration time during the parameterisation of ABMs used in financial markets [5]. Derivative-free optimisation methods have been developed to find the global optimum of high-dimensional black-box functions, which are generally outputs of computationally expensive simulations. Derivative-free optimisation makes use of an SM to aid parameter exploration in conjunction with classical optimisation techniques [21], [22]. An ABM is an example of a high-dimensional black-box function, which can be parameterised more efficiently using derivative-free optimisation methods. Evolutionary algorithms present a viable approach towards parameter exploration tasks in that they are able to leverage ideas from nature to more efficiently search for optimal parameters [24]. Complex machine learning models like the Long Short–Term Memory (LSTM) network can act as an SM to combat the long simulation time inherent in ABMs. LSTM networks can model long-term temporal dependencies, making them a viable substitute to an ABM. This research aims to compare different surrogate assisted parameterisation strategies measuring increased speed and change in accuracy. In addition, we aim to develop a complex simulation model that can approximate the real world better than a simple simulation model. Finally, we aim to assess the trade-off in terms of efficiency using an LSTM network as a replacement to the ABM to tackle the parameterisation and long simulation challenges of ABMS.

1.2 Problem Statement

The reduction in development time for complex simulation models is vital for decision analysis and policy-making during an epidemic. Such models can be used to understand the dynamics of an infectious disease and how individuals will respond to preventative measures. Sufficient complexity needs to be added to these models in order to approximate the real world. With the increase in model complexity, the number of required parameters that need to be calibrated, to allow the model to match real-world data, grows. As a result, searching for meaningful parameter combinations can become computationally prohibitive. Machine learning (ML) models are capable of overcoming some of these computational challenges. To date, there is a lack of comprehensive comparison of ML assisted strategies applied to overcome these challenges. In response to this problem, we propose to evaluate ML models that can effectively search the parameter space of complex simulation models to reduce model development time. In addition, we aim to evaluate the benefit of replacing a complex simulation with an ML model in order to improve upon the efficiency of model simulation.

1.3 Significance and Motivation

The speed at which decision analysis and policy-making take place is vital to preventing the spread of an infectious disease. Agent–based modelling and simulation (ABMS) provide a more complex modelling approach than compartmental models, as they can model behaviours at an individual level [2]. However, ABMS has limitations regarding efficiency due to model validation, parameter calibration and long simulation time.

Surrogate models (SMs) are able to reduce parameter exploration time when parameterising an agent–based model (ABM) [5]. In addition, SMs are able to approximate an ABM and calibrate its parameters more efficiently [6]. However, they are potentially less accurate than the actual ABM. Derivative-free optimisation methods have previously been used to find the global optimum of high-dimensional black-box functions that commonly arise from computationally expensive simulation models. Derivative-free optimisation uses SMs in conjunction with classical optimisation techniques to solve the optimisation problem more efficiently [21], [22]. To date, there is a lack of comparison between surrogate assisted parameter calibration strategies for parameterising an infectious disease ABM. In addition, little work has gone into the exploration of replacing a complex simulation model with an SM for more efficient hyperparameter tuning of ABMs. We aim to compare different surrogate assisted parameterisation strategies to tackle the parameter calibration problem in ABMs. This comparison will provide insight into which strategies are best suited when modelling the spread of an infectious disease. Further, we aim to assess the trade-off between increased speed and change in accuracy when replacing the complex simulation model with an SM. The results obtained will provide better insight into the feasibility of using an SM in place of a complex simulation model.

1.4 Research Aims and Objectives

There is a societal need to rapidly simulate accurate complex simulation models for the spread of an epidemic within a population. This research aims at understanding

how one might reduce the model development time of a complex simulation model without reducing its accuracy. This understanding will be enriched by employing state-of-the-art research into surrogate assisted optimisation methods and recurrent neural networks.

The above aim of this research project will be achieved through the following objectives (goals):

- acquire or implement two epidemic simulation models;
- develop and validate a framework for the integration of surrogate assisted parameterisation strategies into the parameter calibration workflow of epidemiological ABMs;
- implement surrogate assisted sampling strategies for parameterising the implemented simulation models and record the results;
- implement the surrogate assisted optimisation and evolutionary strategies as parameterisation strategies for the implemented simulation models;
- conduct a comparative analysis between the implemented parameterisation strategies and record the results;
- compare the implemented simulation models in terms of approximating real-world data using the most efficient parameterisation strategy and record the results;
- implement an LSTM recurrent neural network to replace the simulation model within the developed framework;
- analyse and compare the results between the simulation model and the LSTM network within the developed framework using real-world data and
- write up and discuss all of the results observed.

1.5 Research Questions

1. How does the quality of the sampling method used affect the capability of finding optimal parameters for an epidemiological ABM?
2. What is the trade-off between increased speed and the change in accuracy when using surrogate parameterisation strategies for hyper-parameter tuning of ABMs?
3. What is the difference in overall accuracy between a complex and simple simulation model for modelling the spread of an infectious disease?
4. By how much can a complex machine learning model reduce the simulation time of an ABM during parameterisation without a significant reduction in accuracy?

1.6 Delineations, Limitations and Assumptions

The epidemiological ABMs identified from the literature focuses on the Influenza virus. We assume that the disease transmission dynamics for COVID-19 are similar in model structure to that used to model the Influenza virus [4].

A possible limitation of this research is that the real-world data may not accurately represent the disease transmission dynamics. The model parameterisation task significantly depends on the data used to compare the simulation output against the real world. Hence, the assumptions made regarding the accuracy in approximating real-world disease transmission dynamics are limited to the data used. Furthermore, we are not doing an exhaustive evaluation of all machine learning algorithms. Therefore, the conclusions we draw will be based on what has been implemented.

1.7 Outline

Having discussed the problem area in Chapter 1, the rest of the dissertation is structured as follows. Chapter 2 investigates surrogate assisted sampling strategies for the parameterisation of an infectious disease ABM. Chapter 3 provides a comparison between surrogate assisted strategies for the parameterisation of an infectious disease ABM. Chapter 4 explores the use of a more complex ABM in comparison to a simple infectious disease ABM when parameterising towards real-world data. Chapter 5 analyses the efficiency between a complex machine learning model in place of the complex simulation model. Lastly, Chapter 6 contains the conclusion to the dissertation.

[4] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-similarities-and-differences-with-influenza

Chapter 2

Surrogate Assisted Sampling Strategies

2.1 Introduction

This chapter presents an agent-based modelling and simulation (ABMS) framework to facilitate the parameterisation of an epidemic agent-based model. In addition, surrogate assisted sampling strategies are presented to address parameter calibration and exploration challenges present in agent-based models (ABMs). The surrogate assisted sampling strategies are compared using different surrogate models and sampling methods. The results show:

- XGBoost and DT SMs perform the best at assisting the parameterisation of ABMs.
- The surrogate assisted method XGBoost Random is able to get within 98.5% of the optimal distribution with the lowest number of mini-batch evaluations.
- Overall, we show that surrogate assisted methods are more likely to estimate the most optimal parameter vector, which generates a synthetic cumulative data distribution that matches the real cumulative data distribution.
- We note the difficulty of calibrating ABMs when considering that we are trying to estimate only seven of the possible nine parameters using synthetic data and raise the need for further investigation for real-world data.

The rest of this chapter is structured as follows. Section 2.2 describes the infectious disease agent-based model used. Section 2.3 unpacks the Approximate Two-Sample Kolmogorov-Smirnov Test. Section 2.4 formulates and describes the agent-based modelling and simulation (ABMS) framework developed. Section 2.5, defines the sampling methods and surrogate assisted sampling methods used. Section 2.6 lists the machine learning algorithms used as surrogate models. Section 2.7 defines a sanity check put in place to evaluate the quality of the ABMS framework. Section 2.8 explains the experiment setup and the initial configurations for the ABMS framework. Section 2.9 discusses the results we have observed from the experiments conducted. Lastly, Section 2.10 concludes the chapter with a conclusion.

2.2 SIRD Agent-Based Model

The agent-based model (ABM) used in this chapter is a pre-existing model implemented by the *Julia* library, *Agents.jl*[1]. The ABM is a continuous space virus spread

[1] https://juliadynamics.github.io/Agents.jl/stable/models/

model, where the disease transmission dynamics follows the Susceptible-Infected-Recovered-Dead (SIRD) framework. The SIRD framework is a variation of the Susceptible–Infected–Recovered (SIR) framework, proposed by Kermack and McKendrick [10]. The ABM takes in the input parameters presented in Table 2.1, where we can see the range for each parameter. The SIRD framework models the ratio of susceptible, infected, recovered and dead individuals within a population. In Figure 2.1, we can see a representation of the SIRD disease transmission dynamics, indicating how agents move between the disease states. Agents have position and velocity attributes that govern their movement within the environment. The number of days an agent is infected for at time t is tracked as one of the agent's attributes. Once the agent has passed the infection period, the agent will move into the recovered or dead state based on the probability of death. Agents who have recovered can be reinfected again based on the probability of reinfection. Agents within the environment are made immovable once they have passed the detection period, indicating they are aware of their infection. Contact between two agents is defined by the intersection of the agents' interaction radius. For simplicity, all agents are given the same value that the model predefines.

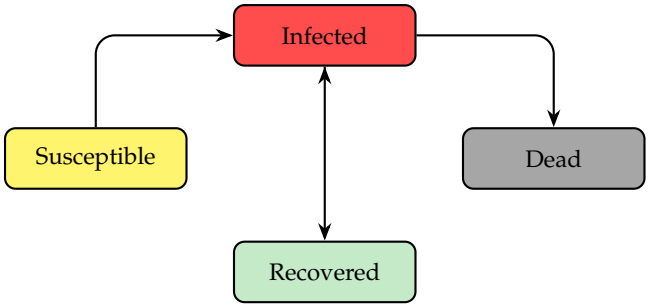


FIGURE 2.1: Susceptible–Infected–Recovered–Dead Transmission Dynamics.

TABLE 2.1: Table of the ranges for each parameter value of the ABM that we have considered for parameterisation. Parameters $1, 2, 3$, and 6 are sampled between the range $(0, 1)$. Parameter 7 is sampled between the range $(0, 0.022)$ to mimic real world interaction as the space defined is bounded by $(1, 1)$. Parameters 4 and 5 are sampled between the range $(0, 41)$ days.

Parameters	**Range**
1. Transmission Probability (β)	(0.0, 1.0)
2. Reinfection Probability	(0.0, 1.0)
3. Death Probability	(0.0, 1.0)
4. Infection Period	(0, 41) days
5. Detection Time	(0, 41) days
6. Speed	(0.0, 1.0)
7. Interaction Radius	(0.0, 0.022)

2.3 Approximate Two-Sample Kolmogorov-Smirnov Test

The approximate two-sample Kolmogorov-Smirnov (KS) test is used to compare the similarity between the distributions of the actual (real-world/synthetic) and simulated data as follows:

$$D_{A,S} = \sup_x |F_A(x) - F_S(x)|, \tag{2.1}$$

where x represents the feature we are measuring (number of infected individuals) and F_A and F_S are the distribution functions of the actual and simulated data respectively. The null hypothesis, H_0, states that the two distributions are not the same. The null hypothesis is rejected at a significance level $\alpha = 0.05$ if $D_{A,S} > D_{N,\alpha}$, where

$$D_{N,\alpha} = c(\alpha)\sqrt{\frac{2 \cdot N}{N^2}}, \tag{2.2}$$

$$c(\alpha) = \sqrt{-\ln(\alpha) \cdot \left(\frac{1}{2}\right)}, \tag{2.3}$$

and N = Population Size. The value $D_{A,S}$, calculated in Equation 2.1, is termed the Kolmogorov-Smirnov test statistic (KSTS). The KSTS value is between the range $[0, 1]$, where the distributions are more similar as the value tends to zero and less similar as it tends to one. The agent-based model (ABM) takes in a candidate parameter vector as its input, which generates simulated epidemic data as its output. The corresponding vector is labelled as negative if the actual and simulated distributions are not the same and positive if similar. The KS test compares the cumulative distributions of the samples, which must be calculated. The data distributions that we are comparing are time series. However, the empirical KS test is formulated to assess the distance between two independent and identically distributed samples. Applying the KS test on the time-series epidemic data requires the time-series to be converted to a cumulative distribution function. A cumulative sum is calculated along the time dimension, where it is then scaled to arrive at a cumulative distribution that maintains the integrity of the original time series. This normalisation makes our problem scale-invariant to the population size, allowing us to measure the similarity of the exact epidemic trends between the two time-series distributions that differ in scale.

2.4 Agent-Based Modelling and Simulation Framework

In order to evaluate different surrogate assisted sampling strategies more effectively, we have developed an agent-based modelling and simulation (ABMS) parameterisation framework. In Figure 2.2, we can see a diagrammatic representation of the developed framework, which is inspired by the algorithm of Lamperti, Roventini, and Sani [5]. An initialisation step is taken before we begin using the framework, where we set an initial configuration. The initial configuration includes:

- selection of the sampling method
- selection of the machine learning (ML) algorithm that will be used to construct the surrogate;
- setting the *MAX Budget* and *MIN Budget* which corresponds to maximum and minimum the number of agent-based model (ABM) evaluations, respectively;

- setting the mini-batch size (batch size);
- definition of a KSTS *Threshold* value;
- definition of a surrogate confidence criteria and
- a real-world/synthetic data distribution as input, to which the ABM will be parameterised.

The framework begins by generating a pool of candidate parameter vectors utilising the chosen sampling method. Subsequently, a mini-batch of parameter vectors are sampled at random, where each vector is passed as input to the ABM. The ABM simulates a synthetic epidemic based on the input vector it receives. The *KS Hypothesis Test*, explained in Section 2.3, calculates the similarity between the simulated epidemic and the real-world/synthetic cumulative data distributions. Accordingly, the candidate parameter vector used to produce the simulated epidemic is labelled as positive or negative. Once all of the sampled mini-batch has been evaluated by the ABM and the *KS Hypothesis Test*, it is then added to a database of labelled parameter vectors. The database of labelled parameter vectors is employed to construct a surrogate model (SM) using ML techniques. Then we evaluate if the SM meets the confidence criteria[2]. We either check if we have reached the *MIN Budget* or go to *Phase 2*. After examining the *MIN Budget*, we either resume at *Phase* 1 or predict the optimal parameter vector from our database of labelled vectors. The optimal parameter vector is predicted by selecting the vector within the labelled database with the lowest KSTS value. We then verify to see if the predicted optimal has a KSTS value less than or equal to the KSTS *Threshold*. Depending on the outcome of the check, we either evaluate the *MAX Budget* and go to *Phase* 2 or stop.

2.5 Sampling Methods

2.5.1 Random Sampler

This sampling technique is used as our baseline. We generate M candidate parameter vectors using a pseudo-random sampling method where the length of each vector is dependent on the number of parameters we parameterise.

2.5.2 Surrogate Assisted Random Sampler

To improve the accuracy and efficiency of the random sampler, we propose the surrogate assisted random sampler. Once we have a confident SM, we then re-generate a new set of random candidate parameter vectors. The new set is passed as input to the SM, classifying the candidates as positive or negative parameter calibrations. The parameter pool is then re-initialised using an ϵ-greedy algorithm. Positively predicted candidates are selected 90% of the time and negatively predicted candidates 10% of the time. This encourages exploration of the parameter space and drives the method to generate candidate parameter vectors classified as positive calibrations.

2.5.3 Quasi-Random Sobol Sampler

As an alternate approach to the pseudo-random sampler, a Sobol sampling mechanism is employed, which is based on the works of Bratley and Fox [44] and Joe

[2]The validation accuracy of the SM is compared to a predefined validation accuracy criteria.

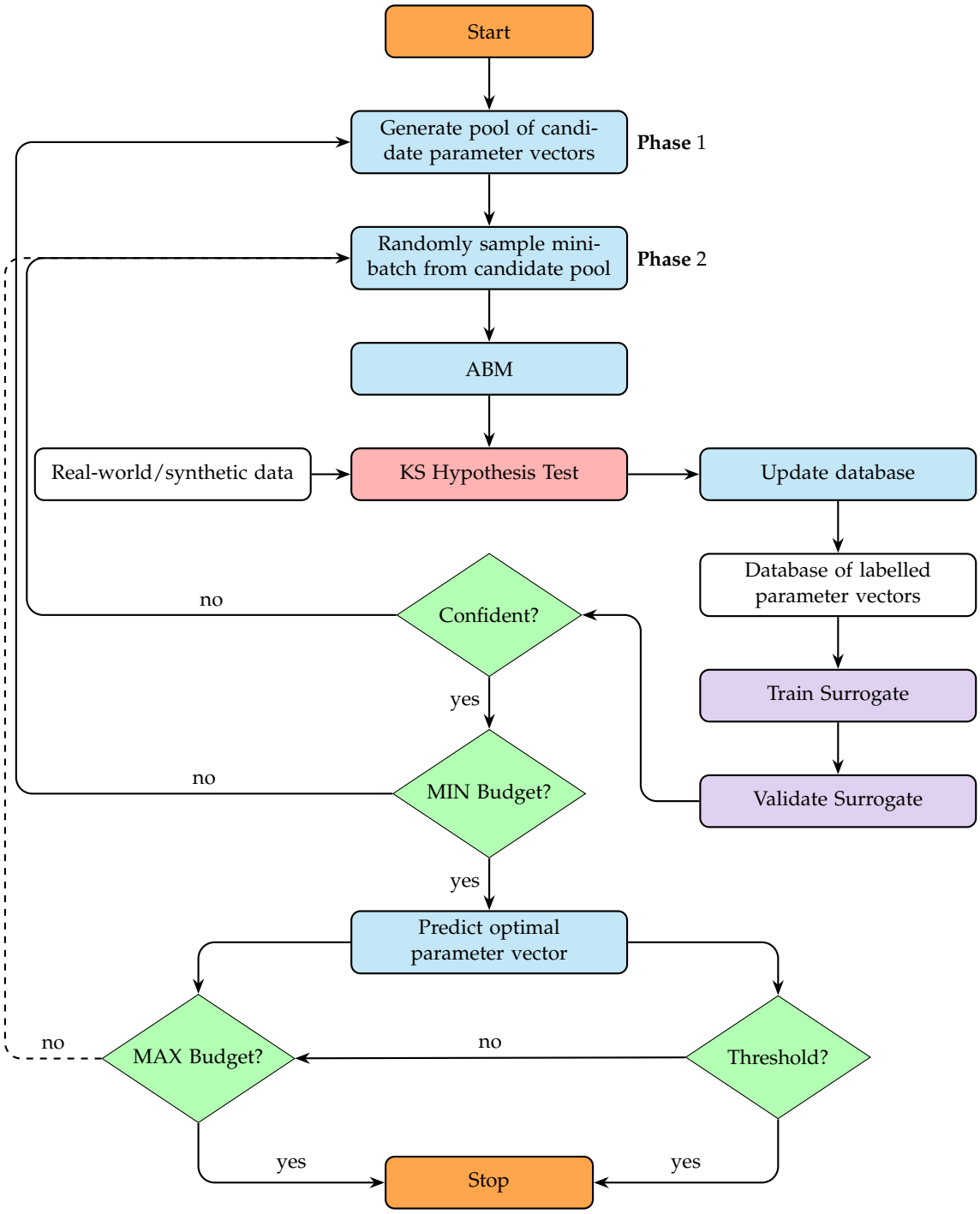


FIGURE 2.2: Agent-based modelling and simulation (ABMS) framework used to evaluate the different surrogate assisted sampling strategies.

and Kuo [45]. This sampling method generates low discrepancy sequences of points that are equally distributed around an M-dimensional hyper-cube. The sequence of points generated tends to be more evenly distributed for a discrete set of points than the pseudo-random sampler.

2.5.4 Surrogate Assisted Quasi-Random Sobol Sampler

Following the approach of the surrogate assisted random sampler, a quasi-random Sobol sampler is developed. The difference in this approach comes with the generation of quasi-random Sobol samples instead of pseudo-random samples. Once we have a confident SM, we re-generate a new candidate parameter vector pool using the quasi-random Sobol sampler. The new set is passed as input to the SM, classifying the candidates as positive or negative parameter calibrations. The parameter pool is then re-initialised using an ϵ-greedy algorithm, where $\epsilon = 0.1$. Positively predicted candidates are selected 90% of the time and negatively predicted candidates 10% of the time. This encourages exploration of the parameter space utilising the nature of the quasi-random Sobol sampler.

2.6 Surrogate Models

We evaluated the following machine learning (ML) algorithms for learning surrogate models (SMs):

- **eXtreme Gradient Boosting (XGBoost):** A decision tree ensemble machine learning algorithm, based off the framework by [46], that is scalable and efficient in its implementation [47].
- **Decision Tree (DT):** A classification algorithm based off a tree-like structure, where the leaves represent a feature/attribute and the branches represent the decision rule which leads to the outcome of that decision [48].
- **Support Vector Machine (SVM):** A classification algorithm which finds a hyperplane in an M-dimensional space in order to differentiate between different classes of data points [49].

The database of labelled parameter vectors is used to construct an SM. The database is split into a training and validation set using an 80/20 split. The SM is constructed using an ML algorithm, where the SM is trained using the training set and validated with the validation set. The confidence of a SM is dependent on the *F1 Score*, which we can see in Equation 2.4, obtained on validation set.

$$\text{F1 Score} = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{2.4}$$

An SM is confident if the *F1 Score* on the validation set is greater than or equal to the confidence criteria that the ABMS framework is initialised with, which corresponds to an *F1 Score* value that is deemed feasible.

2.7 Sanity Check

To ensure that our implemented ABMS framework in Figure 2.2 is reliable, we need to conduct a sanity check. Given a set of known optimal parameters, θ^*, we use this as input for the constructed ABM. The ABM will then generate a synthetic dataset based on the value of θ^* as output. We subsequently use the synthetic dataset as the real-world/synthetic data in Figure 2.2. After that, we run through the ABMS framework and observe if we can approximate θ^*.

2.8 Experiment Setup

The following initial configurations are set as: *MIN Budget* = 500, *MAX Budget* = 2500, *Batch Size* = 50 and KSTS *Threshold* = 0.005 ($\approx$ 99.5% similar to the true distribution). Lastly, the confidence criteria of the SM was set so that we have at least evaluated a proportional number of candidate parameter vectors, dependent on the *batch size* and the number of parameters we are parameterising, at a validation *F1 Score* $\geq$ 0.90. A total of 56 experiments were conducted, averaging each experiment 10 times. For each average, the real-world/synthetic data used was generated by simulating an epidemic for 10 random parameter vectors. All of the sampling methods and surrogate assisted sampling methods implemented were compared, parametersing parameters $1, \ldots, 7$ as seen in Table 2.1.

2.8.1 Hardware Specifications

The machine used to run our experiments consisted of an Intel Xeon CPU E5-2683 v4 @ 2.10GHz processor with 64 CPUs and 256GB of RAM using the Ubuntu 18.04.4 LTS operating system.

2.9 Results and Discussion

We evaluate the performance of the various surrogate models, sampling methods, and surrogate assisted sampling methods in the context of the following metrics:

- **Standardised L_2 Norm:** The euclidean distance between the true input parameter vector and the predicted optimal parameter vector.
- **Kolmogorov-Smirnov Test Statistic (KSTS):** The maximum distance between two empirical time-series cumulative distributions functions. The similarity of the time-series distributions increases as this value tends to 0.
- **Mini-batch Evaluations to Success (MBS):** Success is defined as the quality criterion that needs to be achieved, i.e. a solution within 99% or 95% of the known optimal parameter values. MBS is the number of mini-batches the framework required to reach success, i.e. the MBS at a success of 99% would be the number of mini-batches used to get to within a 99% of the optimal value.

In Table 2.2 and Table 2.3, we can see the Standardised L_2 Norm and Kolmogorov-Smirnov test statistic (KSTS) values, respectively, for each of the methods that we have implemented, calibrating parameters $1, \ldots, 7$. The values presented in Table 2.3, are relatively close to zero, which implies the difference between the true input parameter vector and the predicted optimal is minimal. In Figure 2.3, we can see a visual representation of the KSTS values shown in Table 2.2. The results obtained clearly illustrate that trying to parameterise more than three parameters results in a gradual decrease in performance as measured by the KSTS.

In Figure 2.3, we can see that Random DT consistently outperforms Random for parameters 1, 2, 3, 4, 5 and 7. Random XGBoost outperforms Random for parameters 1, 2, 3 and 7. In Figure 2.4, it can be seen that for parameterising parameters $1, \ldots, 7$, that Sobol DT outperforms Sobol for all parameters. The results from Table 2.2, Figure 2.3 and Figure 2.4 clearly illustrate that XGBoost and DT surrogates are able to get the lowest KSTS values and also the lowest L_2 norm values overall.

TABLE 2.2: Standardised L_2 Norm values for the optimal predicted parameter vectors using each of the strategies and SMs implemented. The optimal values are marked in bold.

		Standardised L_2 Norm						
Sampler	**Surrogate**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Random	-	0.0094	0.0930	0.2434	0.9010	**0.7385**	1.0194	1.6187
Sobol	-	0.0207	0.1544	0.2294	0.6363	1.0808	1.0739	**1.0879**
	XGBoost	**0.0088**	0.0794	0.2033	0.7192	0.9712	1.0922	1.6548
Random	**DT**	0.0169	**0.0760**	0.2449	**0.6302**	0.9345	1.6228	1.4602
	SVM	0.0217	0.1414	0.2595	0.8081	9.8898	1.1399	1.3258
	XGBoost	0.0169	0.0839	0.2193	0.5524	0.8779	**1.0059**	1.6922
Sobol	**DT**	0.0147	0.1201	**0.1454**	0.6898	0.7487	1.1827	1.3977
	SVM	0.0130	0.1331	0.2674	0.6960	0.9751	1.4954	1.1180

TABLE 2.3: Kolmogorov-Smirnov Test Statistic (KSTS) values for the optimal predicted parameter vectors using each of the strategies and SMs implemented. The optimal values are marked in bold.

		Kolmogorov-Smirnov Test Statistic (KSTS)						
Sampler	**Surrogate**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Random	-	0.0014	0.0035	0.0085	0.0264	0.0140	0.0242	0.0209
Sobol	-	0.0018	0.0037	0.0050	0.0290	0.0150	0.0274	0.0226
	XGBoost	**0.0006**	0.0028	**0.0044**	0.0310	0.0160	0.0295	**0.0174**
Random	**DT**	0.0008	**0.0023**	0.0055	**0.0216**	**0.0124**	0.0286	0.0221
	SVM	0.0014	0.0034	0.0053	0.0266	0.0130	0.0307	0.0210
	XGBoost	0.0014	0.0024	0.0081	0.0277	0.0157	0.0251	0.0237
Sobol	**DT**	0.0010	0.0028	0.0046	0.0282	0.0134	**0.0219**	0.0213
	SVM	0.0017	0.0038	0.0070	0.0276	0.0194	0.0272	0.0206

The values presented in Table 2.4 correspond to the averaged minimum number of mini-batch evaluations it took to reach the optimal predicted parameter vector. The table also details the number of mini-batch evaluations to succeed at different percentage intervals for parameterising seven of the ABM's parameters. Sobol XGBoost is able to reach success at 97% and 97.5% in only 14.3 and 20 mini-batch evaluations on average respectively. Random XGBoost and Sobol SVM are able to reach success at 98% and 98.5% respectively in the least amount of mini-batch evaluations on average. As we scale the problem size and the complexity of the ABM, we may find that doing so many mini-batch evaluations is computationally infeasible to reach that level of success.

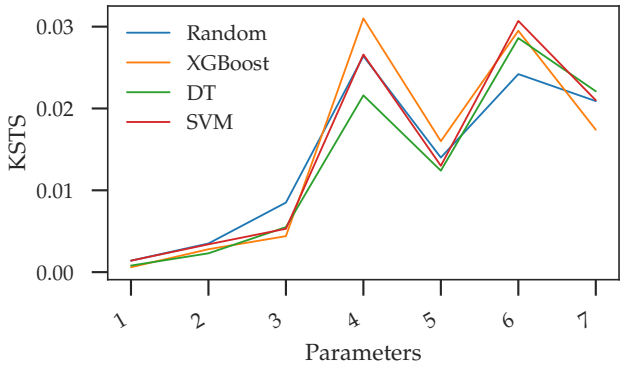


FIGURE 2.3: Comparison between the KSTS values of the different surrogate assisted random sampling methods implemented and the number of parameters being estimated.

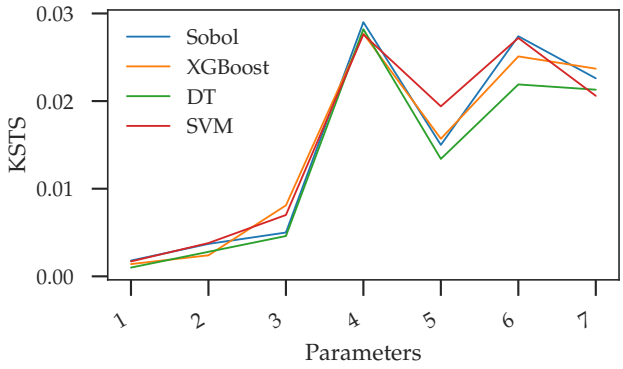


FIGURE 2.4: Comparison between the KSTS values of the different surrogate assisted sobol sampling methods implemented and the number of parameters being estimated.

TABLE 2.4: Number of mini-batch evaluations on average to success at (97%, 97.5%, 98% and 98.5%) for estimating 7 ABM parameters. The optimal values are marked in bold.

		Standardised L_2 Norm			
Sampler	**Surrogate**	**97%**	**97.5%**	**98%**	**98.5%**
Random	-	15.6	25.4	34.4	43.0
Sobol	-	20.9	26.8	35.7	40.5
Random	**XGBoost**	17.0	25.6	32.9	**39.1**
	DT	23.0	26.6	30.2	40.9
	SVM	22.4	27.3	31.6	40.1
Sobol	**XGBoost**	**14.3**	**20.0**	37.2	41.1
	DT	24.5	25.2	29.2	40.5
	SVM	20.5	27.7	**27.7**	43.1

2.10 Conclusion

This chapter presented an implementation of an ABMS framework that can effectively swap out and replace alternative sampling methods and surrogate models. In addition, the framework allows for evaluating the performance of parameter calibration and exploration challenges in ABMS. The results demonstrate that the surrogate assisted methods perform better than the Random and Quasi-Random Sobol sampling methods. Further, we show that the ABMS framework can predict an optimal parameter vector relatively close to the actual input parameters used to generate the synthetic epidemic. In addition, employing an XGBoost and DT surrogate outperformed competing methods at assisting the sampling methods in approximating the synthetic data distribution.

Chapter 3

Comparison of Surrogate Assisted Parameterisation Strategies

3.1 Introduction

In this chapter, we expand on our work conducted in Chapter 2. An improved agent-based modelling and simulation (ABMS) framework is presented to facilitate the evaluation of different parameterisation strategies. The Metric Stochastic Response Surface (MSRS), introduced by Regis and Shoemaker [22], and Dynamic Coordinate Search Using Response Surface Models (DYCORS), introduced by Regis and Shoemaker [21], global optimisation strategies have been modified and re-implemented to enable ABM parameterisation within our framework. Further, we implement an improved version of the surrogate assisted sampling strategies and a new surrogate assisted evolutionary strategy. The best strategy-surrogate combination is selected in terms of accuracy and efficiency (speedup). The results obtained in this chapter show that:

1. Significant speedup was obtained between two to four times above the baseline models using surrogate assisted data-driven optimisation.
2. Better than parity accuracy was achieved across multiple parameters using surrogate assisted strategies compared to the baselines.
3. The MSRS SVM strategy-surrogate combination is able to minimise the distance to the synthetic ABM parameter vector the best overall.
4. The best overall method in terms of both accuracy and speedup is DYCORS XGBoost.

The remainder of this chapter is structured as follows. Section 3.2 explains the improved ABMS framework. Section 3.3 covers the modified surrogate assisted optimisation strategies. Section 3.4 describes the updated surrogate assisted sampling strategies. Section 3.5 introduces a new surrogate assisted evolutionary strategy. Section 3.6 lists the surrogate models used. Section 3.7 explains the experiment setup. Section 3.8 presents the results obtained and a discussion. Lastly, Section 3.9 concludes the chapter.

3.2 Improved Agent-Based Modelling and Simulation Framework

Figure 3.1 presents an improved version of the agent-based modelling and simulation (ABMS) framework introduced in Chapter 2. The improved ABMS framework

can now integrate different surrogates, parameterisation strategies and sampling methods. Before the framework starts, an initial configuration must be set. The details of the initial configuration are as follows:

- selection of the agent-based model (ABM) to be parameterised;
- selection of a sampling method, which will be used to generate candidate parameters from the parameter space;
- selection of a machine learning (ML) algorithm that will construct the surrogate model (SM);
- selection of a parameterisation strategy;
- input the real-world/synthetic data distribution we are parameterising the ABM towards;
- setting the values for the MIN and MAX Budget, which represent the minimum and maximum number of samples required to be evaluated by the ABM, respectively;
- setting the batch size that corresponds to the size of the mini-batch sampled from the parameter pool;
- defining a Threshold value, corresponding to the Kolmogorov-Smirnov test statistic (KSTS) that the predicted optimal should at least attain in terms of similarity to the real-world/synthetic data distribution and
- a confidence criteria for a given surrogate model is defined.

After initialisation, a pool of candidate parameter vectors is generated utilising the selected sampling method. During initialisation, we exhaust the *MIN Budget*. A subset of candidates, equal to *MIN Budget*, are sampled from the parameter pool, which the ABM then evaluates. For each candidate parameter vector, the ABM generates a simulated data distribution. We compare the similarity between the real-world/synthetic cumulative data distribution and the simulated cumulative data distribution using the Approximate Two-Sample Kolmogorov-Smirnov Test, introduced in Chapter 2, and the corresponding candidate is labelled accordingly. The labelled candidates are then included in the database of labelled parameter vectors (ground-truth). We then construct the SM employing the ground-truth database and the selected ML algorithm at the first iteration of the *Main Loop*. After the SM has been constructed, we execute the selected strategy. If we are not at the first iteration of the *Main Loop*, we check whether the SM has diverged from the *Confidence Criteria*. Depending on whether we have diverged from the *Confidence Criteria*, we either go straight to execute the strategy or update the SM using the newly evaluated batch of candidates. Once the strategy has been executed, we predict the optimal candidate. The corresponding KSTS value of the predicted optimal is compared to the predefined *Threshold*, which represents the percentage of similarity deemed to be accurately sufficient. Depending on whether the *Threshold* or *MAX Budget* is met, we either stop or go into the next iteration of the framework. During the next iteration, a mini-batch of candidates are randomly sampled from the pool, and we continue within the *Main Loop*.

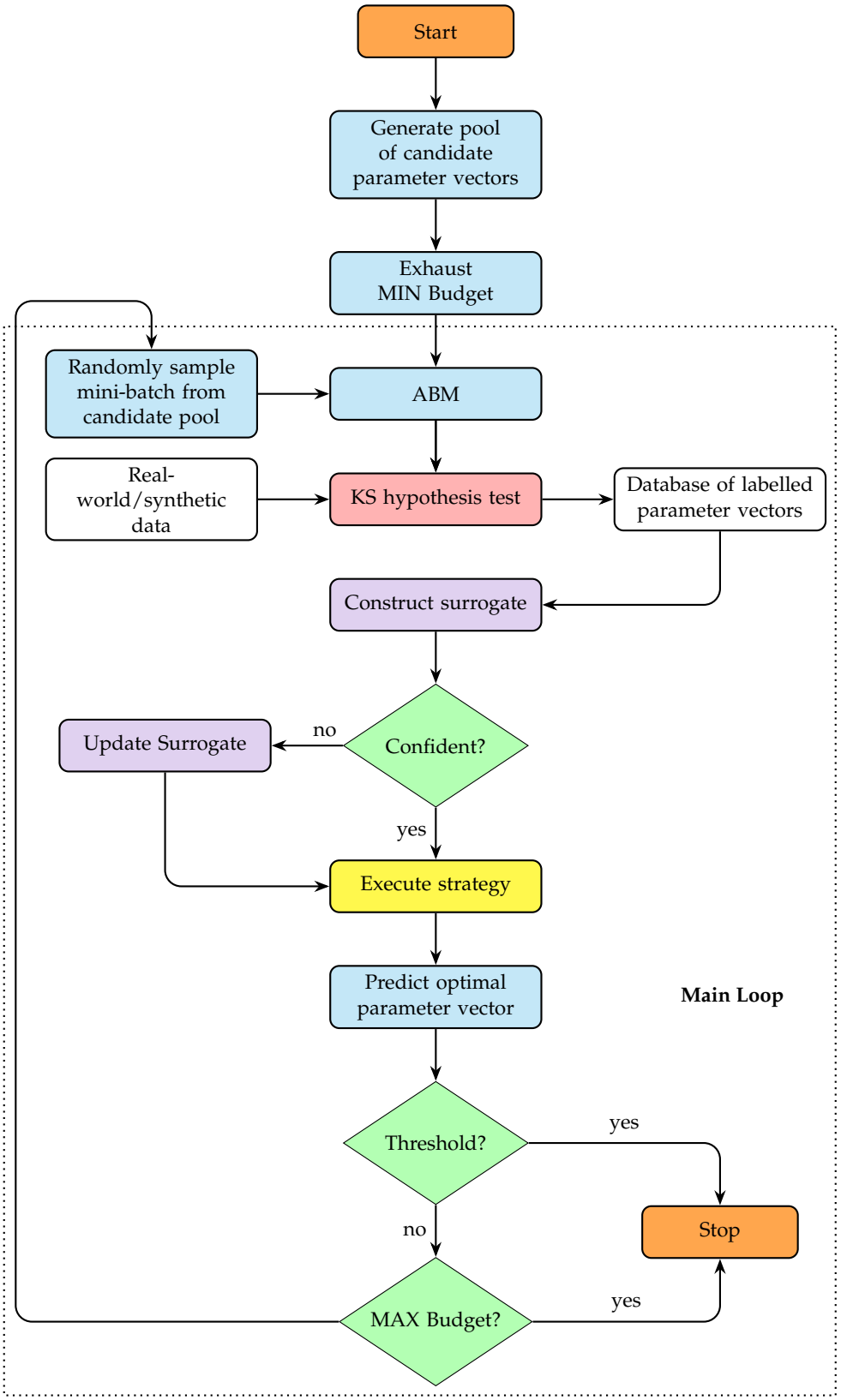


FIGURE 3.1: Improved agent-based modelling and simulation (ABMS) framework used to evaluate the different parameterisation strategies implemented.

3.3 Surrogate Assisted Optimisation Strategies

The following strategies have been modified and re-implemented from the literature into surrogate assisted parameterisation strategies. The strategies are used with the ABMS framework to facilitate parameter calibration of an infectious disease agent-based model.

3.3.1 Metric Stochastic Response Surface

The initial lower and upper bounds of the agent-based model (ABM) parameters are reassigned such that each new candidate is generated as a $Normal(0, \sigma^2)$ distributed perturbation around the current best candidate [22]. The current best candidate is selected with the lowest Kolmogorov-Smirnov test statistic (KSTS) value. As the KSTS value tends to zero, the approximation to the real-world/synthetic data becomes more similar and less similar as the value tends to one. During the reassignment of the upper and lower bounds, the value of σ increases if we have an accurate surrogate model (SM) and decreases if it is inaccurate. A sample of candidates is generated using a sampling method passed through by the ABMS framework. The SM predicts the KSTS value of the newly generated candidate points and then computes the previously evaluated candidates' distance. The predicted KSTS value represents the expected similarity between the real-world/synthetic and simulated data distributions as if the ABM evaluated the candidate. The SM's prediction and the distance to the previously evaluated points are computed and then rescaled through a linear transform on the interval $[0, 1]$. A candidate that minimises the weight-distance merit function,

$$\text{merit}(\bar{x}) = ws(\bar{x}) + (1 - w)d(\bar{x}), \tag{3.1}$$

is selected as the optimal candidate, where $s(x)$ is the SM's prediction of candidate x, $d(x)$ is the minimum distance to the previously seen candidates and $0 \leq w \leq 1$. The weight w is commonly cycled through a finite set of values in order to encourage exploration and exploitation, we chose $w \in \{0.3, 0.5, 0.7, 0.95\}$. When w is close to 0, we do exploration, while w close to 1 does exploitation. At the end of each iteration, the predicted best candidate is evaluated using the ABM, and that candidate is added to the database of labelled parameter vectors.

3.3.2 Dynamic Coordinate Search Using Response Surface Models

Dynamic Coordinate Search Using Response Surface Models (DYCORS) is a modification of the Metric Stochastic Response Surface (MSRS) method presented by Regis and Shoemaker [22]. This method incorporates an idea from the dynamically dimensioned search (DDS) strategy presented by Tolson and Shoemaker [50]. DDS is a heuristic method for box-constrained optimisation, which scales the search for global solutions based on the maximum number of evaluations. In this method, we use DDS to create a new set of candidates perturbed around the current best solution. In each iteration, a set of new candidate vectors are created by adding random perturbations of the current best solution x_{best}, where x_{best} is a candidate vector with the lowest KSTS value. We probabilistically determine a subset of values to perturb where the perturbations are normally distributed with mean zero and a fixed standard deviation within each candidate vector. An SM is then used to predict the KSTS value for each of the new candidates. The candidate with the lowest predicted KSTS value is selected and evaluated by the agent-based model. Then the candidate is added to the ground-truth database and the algorithm iterates.

3.4 Surrogate Assisted Sampling Strategies

The following strategies are modified versions of the surrogate assisted sampling strategies introduced in Chapter 2. The baselines for the surrogate assisted random, and Sobol samplers are the standard Random and Sobol sampling methods, respectively, used in Chapter 2.

3.4.1 Surrogate Assisted Random Sampler

An SM is trained and validated with 3-fold cross-validation. We use the database of labelled parameter vectors for this task, where the best surrogate is selected based on the F1-Score. If the best surrogate's F1-Score is greater than or equal to a specified confidence criteria, predefined by the ABMS framework, and the number of positive and negative samples seen in the database is greater than or equal to ϕ, where

$$\phi = (n_{folds} + n_{parameters}) + 1, \tag{3.2}$$

we are then able to proceed with the strategy. A temporary pool of candidate parameters is generated using the corresponding sampling method. The SM then classifies the candidates as positive or negative parameter calibrations to the real-world/synthetic cumulative data distribution. The new candidate pool is then generated using an ϵ-greedy algorithm, where $\epsilon = 0.10$. We select positively predicted candidates at a rate $1 - \epsilon$ and select negatively predicted candidates at a rate ϵ from the temporary pool. The ϵ-greedy algorithm aims to encourage exploration whilst still exploiting the optimal region within the parameter space.

3.4.2 Surrogate Assisted Quasi-Random Sobol Sampler

The Quasi-Random Sobol sampling method, proposed by [51], generates low discrepancy sequences of points that are equally distributed points around an M-dimensional hypercube, where M refers to the number of parameters we are parameterising. We utilise this sampling method to generate a pool of candidate parameter vectors. Candidate points are similarly classified as positive or negative, and then an ϵ-greedy algorithm is used to generate a new candidate pool. Like the new surrogate assisted random sampler, the SM used is validated using the predefined confidence criteria.

3.5 Covariance Matrix Adaptation Evolutionary Strategy

We introduce a new surrogate assisted parameterisation strategy built upon the Covariance Matrix Adaptation Evolutionary Strategy ($(\mu/\mu, \lambda)$-CMA-ES). We select the parent pool size, λ, to equal our *Batch Size* size, predefined by the ABMS framework. We follow the standard algorithm using the SM to evaluate the fitness for each of the offspring. Once the new elite parents have been selected, we isolate them for our batch to be evaluated by the ABM. The mean of the next generation is calculated using the elite population. The next generation's covariance matrix is calculated using the elite population along with the mean value of the entire population at the current generation. A new set of candidate points are sampled using a Gaussian distribution with the mean and covariance of the next generation [52].

3.6 Surrogate Models

The following machine learning algorithms for learning surrogate models (SMs) are evaluated:

- **eXtreme Gradient Boosting (XGBoost):** A decision tree ensemble machine learning algorithm, based off the framework by [46], that is scalable and efficient in its implementation [47].
- **Decision Tree (DT):** A classification algorithm based on a tree-like structure, where the leaves represent a feature/attribute, and the branches represent the decision rule which leads to the outcome of that decision [48].
- **Support Vector Machine (SVM):** A classification algorithm which finds a hyperplane in an n-dimensional space in order to differentiate between different classes of data points [49].

The newly sampled batch of labelled parameter vectors at each iteration of the ABMS framework are used to validate the SM's performance. In order to validate a SM for classification, we use the:

$$\text{F1 Score} = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{3.3}$$

In order to validate a SM predicting a real value we use the:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{B}(y_i - \hat{y}_i)^2}{B}}, \tag{3.4}$$

where y_i is the predicted real value of the ith candidate from the newly evaluated batch, $\hat{y}_i$ is the true value of the ith candidate and $B = Batch\ Size$.

3.7 Experiment Setup

The following initial configurations were set as: $MIN\ Budget = 500$, $MAX\ Budget = 2500$, batch size $= 250$ and KSTS $Threshold = 0.005$ ($\approx 99.5\%$ similar to the input distribution). The surrogate confidence criteria was split into two cases: when the class label was predicted, *F1 Score* was used, where the $F1\ Score\ Threshold = 0.90$ and when the real label (KSTS value) was predicted, *RMSE* was used, where the $RMSE\ Threshold = 0.001$. The *F1 Score Threshold* and the *RMSE Threshold* are hyperparameters which the ABMS framework was initialised with. A total of 126 experiments were conducted, averaging each experiment 20 times, where for each average the true parameters were varied and different combinations of parameters were tested. All of the strategies implemented were compared, parameterising $1, \ldots, 7$ parameters as seen in Table 3.1. Each surrogate-assisted optimisation strategy was initialised to generate 1000 new samples and perform three iterations every time the strategy was executed within the framework.

3.7.1 Hardware Specifications

The machine used to run our experiments consisted of an Intel Xeon CPU E5-2683 v4 @ 2.10GHz processor with 64 CPUs and 256GB of RAM using the Ubuntu 18.04.4 LTS operating system.

TABLE 3.1: Table of the ranges for each parameter value of the infectious disease ABM, presented in Chapter 2, that we have considered for parameterisation. Parameters $1, 2, 3$, and 6 are sampled between the range $(0, 1)$. Parameter 7 is sampled between the range $(0, 0.022)$ to mimic real world interaction as the space defined is bounded by $(1, 1)$. Parameters 4 and 5 are sampled between the range $(0, 41)$ days.

Parameters	Range
1. Transmission Probability (β)	(0.0, 1.0)
2. Reinfection Probability	(0.0, 1.0)
3. Death Probability	(0.0, 1.0)
4. Infection Period	(0, 41) days
5. Detection Time	(0, 41) days
6. Speed	(0.0, 1.0)
7. Interaction Radius	(0.0, 0.022)

3.8 Results and Discussion

TABLE 3.2: Standardised L_2 Norm values for the optimal predicted parameter vectors using each of the strategies and SMs implemented. The top three values have been highlighted varying in intensities of grey, the darkest grey represents the best value and the lightest grey represents the third best value.

Strategy	Surrogate	Standardised L_2 Norm 1	2	3	4	5	6	7
Random	-	0.1778	0.5363	0.7105	1.5249	2.8104	1.9735	3.1206
Sobol	-	0.0546	0.6204	0.9194	1.3720	2.1372	2.9459	4.6424
	XGBoost	0.0695	0.6866	1.0335	0.9920	3.2106	3.2145	4.3614
Random	**DT**	0.0421	0.5888	0.7117	1.2247	2.5250	2.1988	4.3225
	SVM	0.0541	0.5378	0.7395	1.1894	1.8206	2.1682	3.6745
	XGBoost	0.1076	0.7249	0.8582	1.2848	2.2395	2.8396	4.9996
Sobol	**DT**	0.1092	0.7215	0.8129	1.3268	2.0950	3.0756	4.0618
	SVM	0.0795	0.9444	1.0304	1.1763	2.1551	3.5414	3.3606
	XGBoost	0.1442	0.6426	0.7628	0.9560	1.5134	1.6819	3.0205
MSRS	**DT**	0.0463	0.7482	0.6837	1.0738	1.2975	1.8974	2.8574
	SVM	0.0463	0.7436	0.6654	0.9479	1.3820	2.6611	2.3829
	XGBoost	0.0679	0.6729	0.8196	1.3433	2.6203	1.6127	3.2048
DYCORS	**DT**	0.0281	0.5231	0.8930	0.9465	2.6397	3.5522	2.7680
	SVM	0.0289	0.5827	0.7838	1.2029	1.4711	2.2533	3.9010
	XGBoost	0.1175	0.5679	0.9984	2.2796	2.2526	4.4795	4.2173
CMAES	**DT**	0.1423	1.0773	0.7987	1.1261	2.8731	4.6390	3.9230
	SVM	0.0369	0.4593	0.5793	1.0855	3.1720	4.5925	3.9130

In the following tables and figures, we present the results of the above experiment, followed by a discussion of the results. The Standardised L_2 Norm values

TABLE 3.3: Kolmogorov-Smirnov Test Statistic (KSTS) values for the optimal predicted parameter vectors using each of the strategies and SMs implemented. The best KSTS values for each parameter are marked in bold.

		Kolmogorov-Smirnov Test Statistic (KSTS)						
Strategy	**Surrogate**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Random	-	.0	0.0004	0.0018	0.0034	0.0047	0.0045	0.0043
Sobol	-	.0	**0.0000**	0.0035	0.0035	0.0037	0.0042	0.0039
	XGBoost	.0	0.0008	0.0026	0.0034	0.0041	0.0046	0.0043
Random	**DT**	.0	**0.0000**	0.0033	0.0028	0.0035	0.0047	0.0038
	SVM	.0	0.0004	0.0019	0.0037	0.0044	0.0049	0.0047
	XGBoost	.0	0.0003	0.0029	0.0034	0.0045	0.0050	0.0045
Sobol	**DT**	.0	**0.0000**	0.0025	0.0027	0.0033	0.0044	0.0039
	SVM	.0	**0.0000**	0.0030	0.0029	0.0041	0.0048	0.0043
	XGBoost	.0	0.0009	0.0039	0.0034	0.0038	0.0043	0.0039
MSRS	**DT**	.0	0.0002	0.0019	0.0032	0.0040	0.0047	0.0045
	SVM	.0	0.0003	0.0036	0.0031	0.0045	0.0042	**0.0034**
	XGBoost	.0	0.0003	0.0023	0.0031	0.0046	0.0042	0.0044
DYCORS	**DT**	.0	0.0003	0.0039	0.0030	**0.0033**	0.0041	0.0045
	SVM	.0	**0.0000**	0.0036	0.0026	0.0041	**0.0036**	0.0041
	XGBoost	.0	0.0004	**0.0006**	0.0023	0.0051	0.0101	0.0055
CMA-ES	**DT**	.0	0.0005	0.0024	**0.0018**	0.0034	0.0067	0.0046
	SVM	.0	0.0005	0.0031	0.0018	0.0048	0.0109	0.0060

show the distance between the synthetic parameter vector and the ABMS framework's optimal prediction. The optimal prediction is the parameter vector (estimate) with the lowest Kolmogorov-Smirnov Test Statistic (KSTS) value. The sanity check, from Chapter 2, assesses whether our ABMS framework and a strategy-surrogate combination can approximate the parameters that generated the synthetic distribution. The values presented in Table 3.2 are relatively close to zero, which implies our sanity check holds. Also, we have highlighted the top three lowest Standardised L_2 Norm values for each of the strategy-surrogate combinations considered. The results show that overall the MSRS and DYCORS strategies can best approximate the synthetic parameters. In particular, MSRS is relatively good at attaining the synthetic parameters for three or more parameters, whereas DYCORS is relatively good across all parameters. CMA-ES is relatively good for 1–3 parameters, however, it falls short compared to the MSRS and DYCORS strategies as the problem's dimensionality increases. The values presented in both Table 3.2 and Table 3.3 show us that the Support Vector Machine (SVM) and Decision Tree (DT) surrogate gets a more significant majority of optimal values (the lowest values), whereas this is not the case for XGBoost. However, XGBoost is still relatively optimal.

When parameterising all seven parameters, we see that MSRS SVM, on average, can approximate the synthetic parameters the best with the lowest Standardised L_2

TABLE 3.4: Probability of reaching success and the speedup acquired within 98% and 99% of the optimal for seven parameters. The optimal stratey surrogate combination in terms of average accuracy and speedup are highlighted. In addition, the highest average accuracy and speedup are marked in bold.

Strategy	Surrogate	Success @98%	Success @99%	Speedup @98%	Speedup @99%
Random	-	**0.85**	**0.75**	0.00	0.00
Sobol	-	**0.85**	0.65	0.00	0.00
Random	**XGBoost**	0.80	0.70	3.60	2.61
	DT	**0.85**	**0.75**	3.46	2.61
	SVM	0.80	0.70	3.53	2.50
Sobol	**XGBoost**	0.80	0.70	3.21	2.61
	DT	0.80	0.70	**4.00**	**3.10**
	SVM	**0.85**	0.70	3.40	2.40
MSRS	**XGBoost**	**0.85**	0.70	**4.00**	**2.86**
	DT	**0.85**	0.65	3.53	2.31
	SVM	0.80	0.70	2.81	2.65
DYCORS	**XGBoost**	**0.85**	**0.75**	**3.75**	**2.90**
	DT	0.75	0.70	3.16	2.73
	SVM	**0.85**	0.70	3.39	2.81
CMA-ES	**XGBoost**	0.70	0.55	2.90	2.11
	DT	0.75	0.65	3.10	2.65
	SVM	0.75	0.60	3.16	2.25

Norm value of 2.3829. In general, looking at Table 3.2 the MSRS and DYCORS strategies can attain the lowest Standardised L_2 Norm values overall. The result implies that MSRS and DYCORS can predict a parameter vector that generates a simulated infection distribution when run through the ABM, most similar to the synthetic infection distribution.

The KSTS values tell us how close we can get to the synthetic infected cumulative data distribution. Table 3.3 shows that for one and two parameters, we are still in a relatively low dimensional space, and as such, the baselines and the surrogate assisted samplers perform the best. As we increase the dimensionality of the parameter space, the CMA-ES strategy is optimal for three and four parameters. When moving towards a higher dimensional space of five or six parameters, the DYCORS strategy attains the lowest KSTS value, and for seven parameters, MSRS has the lowest value. Inspecting Table 3.3 more closely, we note that all techniques can reasonably approximate the synthetic distribution and that no strategy-surrogate combination stands out clearly from the others. In addition, the KSTS values presented in Table 3.3 are significant regarding the p-values of the Approximate Two-Sample Kolmogorov-Smirnov test.

The probabilities of reaching an optimal solution (success) within 98% and 99% of the optimal value are captured in Table 3.4. Success is defined as a measure of similarity to the synthetic data distribution. For example, success at 98% implies that

a method can produce a simulated distribution that is 98% similar to the synthetic distribution that we try to replicate. In the same table, we also show the speedup attained by each strategy-surrogate combination. Across all implemented strategy-surrogate combinations, we obtain a significant speedup compared to the baselines.

Table 3.4 shows that Random, Random DT and DYCORS XGBoost have the highest probability of reaching success at 98% and 99% in that specific order. Also, the Sobol DT, MSRS XGBoost, and DYCORS XGBoost strategy-surrogate combinations achieve the most speedup. Although Random attains a high probability of reaching success, it is not efficient because it provides no speedup compared to the other methods. The DYCORS XGBoost strategy-surrogate combination is optimal when considering both probabilities of reaching success and speedup. One of the limitations is that we can only get 99% accuracy with a 75% probability. This suboptimal probability means that we would need to run the model more than once to ensure the correct outcome. Running more than once would negate some of the speedup achieved. In general, utilising a strategy-surrogate combination that can predict an optimal parameter vector that perfectly matches the synthetic data distribution in terms of accuracy and speedup (DYCORS XGBoost) is preferred.

3.9 Conclusion

This chapter implemented a more extensive and adaptive agent-based modelling and simulation (ABMS) framework. The ABMS framework can effectively swap out parameterisation strategies and surrogate models (SMs) to parameterise an infectious disease agent-based model (ABM). We show that in terms of the lowest Kolmogorov-Smirnov Test Statistic (KSTS) values, we achieve better than parity across all parameters than the surrogate assisted sampling strategies and the baselines. The Decision Tree (DT) and the Support Vector Machine (SVM) surrogates are on par to attain the lowest KSTS values overall, whereas XGBoost falls short slightly. Using the Standardised L_2 Norm values, we show that MSRS and DYCORS are the best strategies overall for getting closest to the synthetic parameters. In particular, MSRS SVM gets the closest to the synthetic parameters. One of the significant challenges in ABMS is that the time required to parameterise an ABM is cumbersome. We have shown that DYCORS XGBoost attains the highest probability of replicating the synthetic cumulative data distribution within 98% and 99% success and achieves the most considerable speedup compared to the baselines and the evaluated strategy-surrogate pairs.

Chapter 4

SIRD vs Complex Agent-Based Model

4.1 Introduction

This chapter introduces a new Complex Agent-Based Model (ABM) that models a secondary diffusive process parallel to the infectious disease spread. The secondary diffusive process models individuals' awareness (interest) towards the spread of the infectious disease. In addition, we introduce an approach to the parameterisation problem calculating the average Kolmogorov Smirnov test statistic (KSTS) value across all output distributions of the ABM. The results obtained in this chapter show that:

1. the Complex ABM can approximate the distribution of the real-world infections better than the SIRD ABM;
2. taking an average KSTS value across all output distributions of the Complex ABM is more accurate than a single distribution for parameterisation and
3. the Complex ABM using the average KSTS approach can approximate the real-world cumulative distributions with 77% and 94% accuracy in the worst and best cases, respectively.

The rest of this chapter is structured as follows. Section 4.2 presents the datasets used and the relevant pre-processing that has been applied. Section 4.3 introduces the newly developed Complex ABM. Section 4.4 explains the improved parameterisation approach utilising the Approximate Two-Sample Kolmogorov-Smirnov test. Section 4.5 describes the experiment setup. Section 4.6, presents the results obtained from the experiments conducted along with a discussion. Lastly, Section 4.7, concludes the chapter.

4.2 Data

4.2.1 Oxford COVID-19 Government Response Tracker

This dataset was downloaded on 02/07/2021 from the Oxford COVID-19 Government Response Tracker[1] (OXCGRT), where information is collected systematically regarding government measures taken during the COVID-19 pandemic. The different policy measures are tracked from 01/01/2020–02/07/2021, covering more than 180 countries and territories. The policy measures taken are coded into 23 indicators which are categorised into five groups of which we use the following two:

[1] https://github.com/OxCGRT/covid-policy-tracker

- **C** - containment and closure policies and
- **H** - health system policies.

The indicators are scaled reflecting the severeness of government policy, and in addition, these values are aggregated into a collection of policy indexes [53]. An index is calculated using Equation 4.1, where k is the number of component indicators used and I_j is the sub-index score for an individual indicator.

$$index = \frac{1}{k}\sum_{j=1}^{k} I_j \tag{4.1}$$

The sub-index score (I) for a given indicator (j) on a particular day (t) is calculated using Equation 4.2, where N_j is the maximum value for the indicator, F_j is a Boolean value (0 or 1) which is dependant on the indicator having a flag, $v_{j,t}$ is the recorded policy value and $f_{i,j}$ is the recorded binary flag.

$$I_{j,t} = 100\frac{v_{j,t} - 0.5(F_j - f_{j,t})}{N_j} \tag{4.2}$$

The recorded binary flag indicates whether the policy measure applies to a specific geographic region within a country or a general policy applied to the entire country. It is worth noting that the data is not fully complete. Where the recorded policy value is missing, a conservative assumption was made that the sub-index score is zero [53].

We use the indicators shown in Table 4.1 and Table 4.2 to calculate a speed restriction index (SRI), using Equation 4.1, and an interaction radius restriction index (IRRI), using Equation 4.2, respectively. The ordinal scale represents the range of values a particular indicator can take, where each varies in the policy measure's intensity, with zero being no measure taken.

TABLE 4.1: Table of containment and closure policy indicators used to calculate the speed restriction index.

Index	Name	Ordinal Scale	Binary Flag
C1	School Closing	0-3	Yes
C2	Workplace Closing	0-3	Yes
C3	Cancel Public Events	0-2	Yes
C4	Restriction on Gatherings	0-4	Yes
C5	Close Public Transport	0-2	Yes
C6	Stay at Home Requirements	0-3	Yes
C7	Restriction on Internal Movement	0-2	Yes
C8	International Travel Controls	0-4	No

TABLE 4.2: Table of health system policy indicators used to calculate the interaction radius restriction index.

Index	Name	Ordinal Scale	Binary Flag
H1	Public Information Campaigns	0-2	Yes
H2	Testing Policy	0-3	No
H3	Contact Tracing	0-2	No
H6	Facial Coverings	0-4	Yes
H7	Vaccination Policy	0-5	No
H8	Protection of Elderly People	0-3	Yes

4.2.2 COVID-19 Data Repository

This dataset was downloaded on 02/07/2021 from the COVID-19 Data Repository published by the Center for Systems Science and Engineering[2] (CSSE) at John Hopkins University [54]. The number of confirmed COVID-19 cases (infections), recoveries and deaths for all affected countries are collected and updated daily, starting from the 22/01/2020–02/07/2021. In Figure 4.1 we show a visual representation of the COVID-19 data obtained. We pre-processed the dataset to obtain the COVID-19 statistics daily by subtracting the total value on day $t+1$ from day t. A seven-day moving average is calculated across each distribution in order to smooth out the data. We then segment out a portion of the total dataset between the dates 05/10/2020–02/07/2021. The segmented portion is broken up into three time periods which can be seen in Figure 4.2, Figure 4.3 and Figure 4.4, where each time period is equal to 90 days. In Figure 4.2, we can see an increasing trend is captured regarding infections, recoveries and deaths. In Figure 4.3, both a peak of one of the waves of the pandemic experienced in South Africa is captured. Lastly, in Figure 4.4 we observe another increasing trend. It is worth noting that these figures represent the data explicitly for South Africa. The other countries that we have included in our experimentation, as seen in Table 4.5 may exhibit different trends.

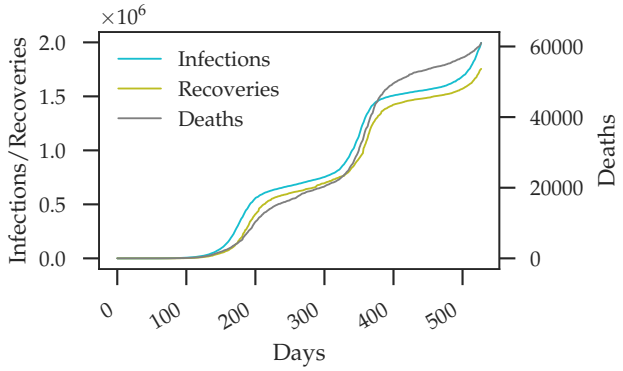


FIGURE 4.1: CSSE dataset containing the confirmed COVID-19 cases (infections), recoveries and deaths for South Africa between dates 22/01/2020–02/07/2021.

[2] https://github.com/CSSEGISandData/COVID-19

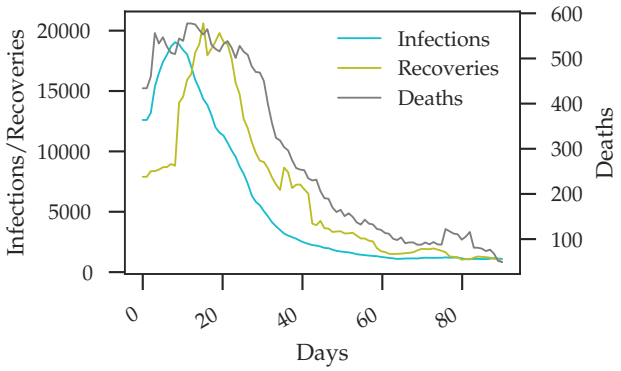


FIGURE 4.2: Time Period 1 – Pre-processed CSSE dataset containing the number of daily COVID-19 cases (infections), recoveries and deaths for South Africa between dates 05/10/2020–02/01/2021.

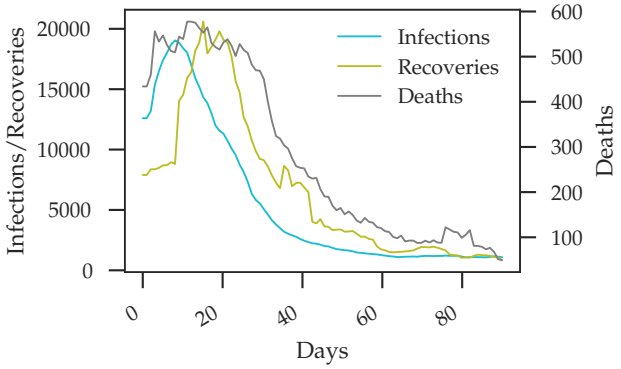


FIGURE 4.3: Time Period 2 – Pre-processed CSSE dataset containing the number of daily COVID-19 cases (infections), recoveries and deaths for South Africa between dates 03/01/2021–02/04/2021.

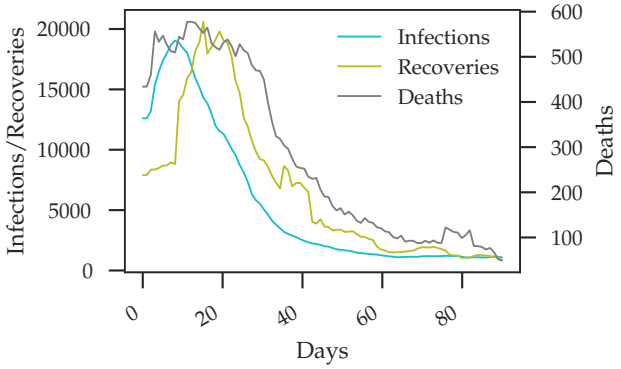


FIGURE 4.4: Time Period 3 – Pre-processed CSSE dataset containing the number of daily COVID-19 cases (infections), recoveries and deaths for South Africa between dates 03/04/2021–02/07/2021.

4.2.3 Google Trends Dataset

Google Trends is a search trend feature, which shows how often a given search term is entered into Google's search engine. The search frequency relative to the total search volume over a given period of time is recorded weekly. The data recorded is anonymous, aggregated, and categorised into groups (regions). We have downloaded the dataset, using the search term *covid19* between the period 05/07/2020–27/06/2021, on the 02/07/2021. The values for each week have been interpolated to convert the data to a daily scale. In addition, we have calculated a seven day moving average over the data in order to smooth out the distribution. In Figure 4.5, Figure 4.6 and Figure 4.7, we show the processed Google Trends dataset across the time periods 05/10/2020–02/01/2021, 03/01/2021–02/04/2021 and 03/04/2021–02/07/2021, respectively. The search data shows increasing and decreasing trends for the different time periods.

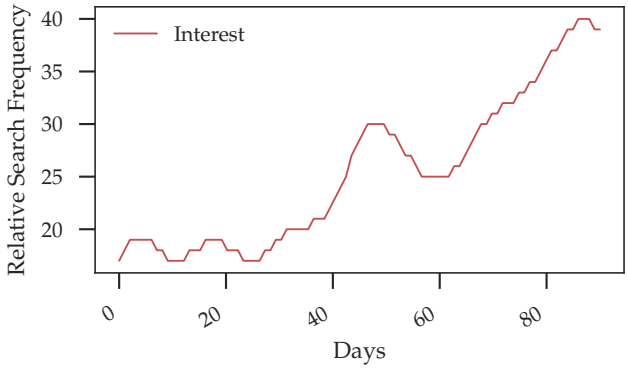


FIGURE 4.5: Google Trends Data using the search term *covid19*, across the time period 05/10/2020–02/01/2021 for South Africa.

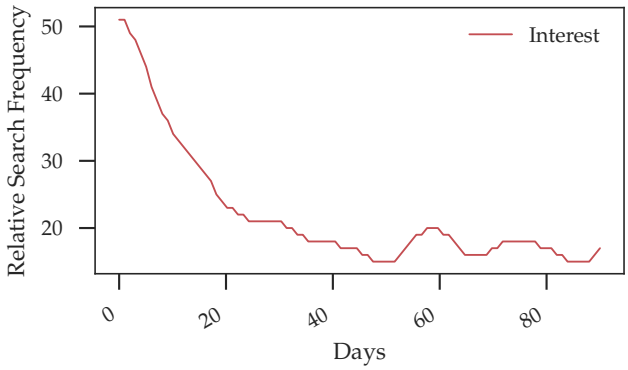


FIGURE 4.6: Google Trends Data using the search term *covid19*, across the time period 03/01/2021–02/04/2021 for South Africa.

4.3 Complex Agent-Based Model

We have developed a Complex agent-based model (ABM), which builds on the Susceptible-Infected-Recovered-Dead (SIRD) ABM, introduced in Chapter 2, incorporating two data streams that influence agents' behaviour. In addition, the spread

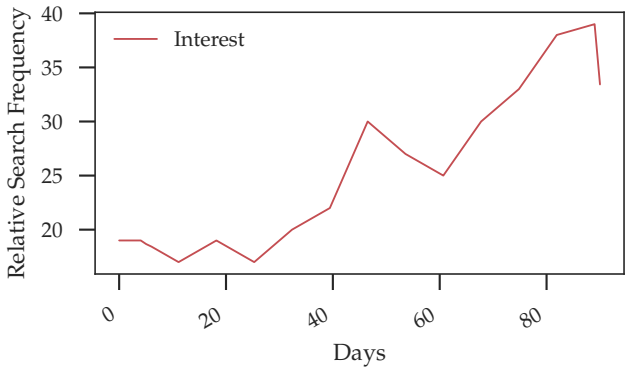


FIGURE 4.7: Google Trends Data using the search term *covid19*, across the time period 03/04/2021–02/07/2021 for South Africa.

of interest towards the virus within the population is modelled in parallel. The SRI and IRRI indexes from Section 4.2.1 are used to modify the speed at which agents move within the environment and the maximum distance for the contact between two agents, respectively. The new ABM takes in 13 parameters, which can be seen in Table 4.3, where we try to find parameter values that allow the ABM to match the real-world data. The model's output contains four distributions relating to daily infections, recoveries, deaths and interest of the modelled population. We introduced two new Δ parameters, parameters 8 and 10 in Table 4.3, which are used to update an agent's speed and interaction radius, respectively. These Δ parameters are used to reduce an agent's speed and interaction radius proportionally to the SRI and IRRI indexes. In addition, we allow the number of initially infected and initially interested agents to be set as parameters that we parameterise. The initially infected agents are divided into bins, where the number of bins is equal to the disease infection period. We then set the agents' days infected attribute equal to the bin in which the agent resides. A similar process is carried out for initially interested agents, where the bin size is equal to the interest infection period. The purpose of this is to allow the ABM to step in at any given point during an epidemic. Lastly, we model the spread of awareness (interest) using a renamed version of the standard Susceptible-Infected-Recovered (SIR) framework. The new framework, Not Informed-Interested-Uninterested (NIU), which we can see in Figure 4.8, is renamed to relate to the transmission dynamics of interest. Modelling the spread of interest to the virus within the population shows the increased complexity an ABM can incorporate. The spread of interest within the model does not affect the spread of the disease at this stage as we have no real-world data which maps the relationship between the spread of interest and COVID-19. The Google Trends dataset is used to parameterise the interest transmission dynamics of the model.

4.4 Average Kolmogorov-Smirnov Test Statistic

Previously, in Chapter 2 we have used the Approximate Two-Sample Kolmogorov-Smirnov Test to measure the similarity between two time-series distributions. The Kolmogorov-Smirnov test statistic (KSTS) is a value between $(0, 1)$, which servers as the metric for the similarity between the distributions in comparison. As the KSTS value tends to zero, the similarity between the two distributions increases and

TABLE 4.3: Table of the ranges for each parameter value of the ABM that we have considered for parameterisation. Parameters $1, 2, 4, 5, 9, 10$ and 11 are sampled between the range $(0, 1)$. Parameters 7 and 8 are sampled between the range $(0, 0.022)$ to mimic real world interaction as the space defined is bounded by $(1, 1)$. Parameters 3 and 6 are sampled between the range $(0, 90)$ days. Lastly, parameters 12 and 13 are sampled between $(1,\ N_{Agents})$, where N_{Agents} represents the number of agents within the modelled population.

Parameters	Range
1. Disease Transmission Probability	(0.0, 1.0)
2. Disease Reinfection Probability	(0.0, 1.0)
3. Disease Infection Period	(0, 90) days
4. Interest Transmission Probability	(0.0, 1.0)
5. Interest Reinfection Probability	(0.0, 1.0)
6. Interest Infection Period Time	(0, 90) days
7. Interaction Radius	(0.0, 0.022)
8. ΔInteraction Radius	(0.0, 0.022)
9. Speed	(0.0, 1.0)
10. ΔSpeed	(0.0, 1.0)
11. Death Rate	(0.0, 1.0)
12. Initial Infected	$(0,\ N_{Agents})$
13. Initial Interested	$(0,\ N_{Agents})$

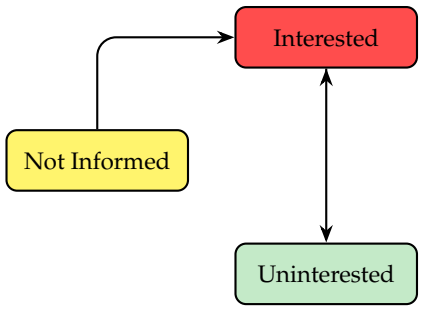


FIGURE 4.8: Not Informed-Interested-Uninterested Transmission Dynamics.

as the KSTS value tends to one, the similarity between the two distributions decrease. The Kolmogorov-Smirnov test used in Chapter 2 compares the similarity between the cumulative daily infections distribution produced by the ABM and the synthetic daily infections distribution. The average KSTS is calculated by computing the KSTS value between each output distribution produced by the ABM to its corresponding real-world/synthetic distribution. The average KSTS is used by the improved ABMS framework, introduced in Chapter 3, to calibrate the ABM towards real-world/synthetic distributions.

4.5 Experiment Setup

In this chapter, we have performed two main experiments. The first experiment compared the performance between the SIRD ABM used in Chapter 2 and the newly

constructed Complex ABM. In experiment one, we parameterised both ABMs towards the distribution of the daily infections obtained from the real world for each time period. The second experiment evaluated the benefit of including all the output distributions within the KSTS calculation versus a single output distribution. The real-world data has been split up into three time periods listed below:

- **Period 1**: 05/10/2020 – 02/01/2021
- **Period 2**: 03/01/2021 – 02/04/2021
- **Period 3**: 03/04/2021 – 02/07/2021

where each time period is of length 90 (days). We have arbitrarily selected seven countries for our experiment shown in Table 4.4, where we see the corresponding number of confirmed COVID-19 cases to date which vary between each country. We have conducted a total of 63 experiments.

The DYCORS XGBoost strategy-surrogate pair was chosen for this experiment based off the results seen in Chapter 3. The following initial configurations were set as: *MIN Budget* $= 500$, *MAX Budget* $= 2500$, *Batch Size* $= 250$ and KSTS *Threshold* $= 0.005$ ($\approx 99.5\%$ similar to the input distribution). The surrogate-strategy pair selected predicts the real label (KSTS value), hence, we used *RMSE* to measure the surrogates performance after each mini-batch, where the *RMSE Threshold* $= 0.001$.

4.5.1 Hardware Specifications

The machine used to run our experiments consisted of an Intel Xeon CPU E5-2683 v4 @ 2.10GHz processor with 64 CPUs and 256GB of RAM using the Ubuntu 18.04.4 LTS operating system.

TABLE 4.4: Table of the countries we have used in our experiment along with their confirmed number of COVID-19 cases to date (02/07/2021).

Countries	Confirmed Cases
South Africa	2 019 826
Egypt	281 722
France	5 780 648
Germany	3 737 611
India	30 501 189
Italy	4 261 579
United Kingdom	4 854 367

4.6 Results and Discussion

In the following tables and figures, we present the results of the above experiments, followed by a discussion of the results. The values presented in Table 4.5 shows a comparison between the Susceptible-Infected-Recovered-Dead (SIRD) agent-based model (ABM), introduced in Chapter 2, and the Complex ABM for each time period, parameterising the model towards the distribution of the real-world infections only. The values presented in Table 4.6, Table 4.7 and Table 4.8 show parameterising the

Complex ABM towards a single data distribution in comparison to all data distributions.

The Kolmogorov-Smirnov test statistic (KSTS) values presented in Table 4.5 shows that the Complex ABM is able to approximate the cumulative daily infections distribution with a tighter fit than the SIRD ABM for each country. The SIRD ABM outperforms the Complex ABM in Time Period 2 for France. The same result is observed in Time Period 3 for South Africa and the United Kingdom. However, on average, the Complex ABM across each time period outperforms the SIRD ABM by attaining a KSTS value that is closer to zero.

TABLE 4.5: Comparison of KSTS values between the SIRD ABM and the Complex ABM, calibrating the models towards number of daily infections, between the time periods 05/10/2020 – 02/01/2021 (Time Period 1), 03/01/2021 – 02/04/2021 (Time Period 2) and 03/04/2021 – 02/07/2021 (Time Period 3). The time periods and corresponding countries where the SIRD model was optimal is highlighted. In addition, we have marked the optimal average for each time period in bold.

Country	ABM	Time Period 1	Time Period 2	Time Period 3	Average
South Africa	**SIRD**	0.07	0.18	0.04	0.10
	Complex	0.04	0.09	0.06	**0.06**
Egypt	**SIRD**	0.09	0.25	0.28	0.21
	Complex	0.05	0.03	0.04	**0.04**
France	**SIRD**	0.47	0.04	0.22	0.24
	Complex	0.09	0.06	0.03	**0.06**
Germany	**SIRD**	0.29	0.25	0.30	0.28
	Complex	0.04	0.03	0.06	**0.05**
India	**SIRD**	0.17	0.15	0.63	0.31
	Complex	0.02	0.18	0.05	**0.08**
Italy	**SIRD**	0.53	0.15	0.15	0.28
	Complex	0.08	0.05	0.04	**0.06**
United Kingdom	**SIRD**	0.08	0.21	0.10	0.13
	Complex	0.08	0.02	0.14	**0.08**

The values presented in Table 4.6, Table 4.7 and Table 4.8 show the difference between parameterising towards a single KSTS value in comparison to taking an average of each KSTS value per output distribution. The Single parameterisation model is able to approximate the distribution of the daily infections with a tighter fit to the real-world data in contrast to the Average model. However, the Average model approximates the remaining three real-world data distributions better than the single model. Hence, on average, across each time period, the Average model performs better overall as it is able to attain an average KSTS value closer to zero.

The performance on average between the Single and Average Complex ABM is shown in Figure 4.9, Figure 4.10 and Figure 4.11. Performance is measured in terms of the percentage of similarity to the real-world distributions. In Figure 4.9, the single model does poorly for Egypt, South Africa and the United Kingdom, whereas the average model appears more stable given the variety of countries. A similar result is

TABLE 4.6: Comparison of KSTS values between the single KS Test and the average KS Test for the time period 05/10/2020 – 02/01/2021. The average optimal values are marked in bold.

Country	KS Test	Infections	Recoveries	Deaths	Interest	Average
South Africa	**Single**	0.04	0.73	0.67	0.57	0.50
	Average	0.47	0.07	0.07	0.11	**0.18**
Egypt	**Single**	0.05	0.54	1.00	0.60	0.55
	Average	0.38	0.11	0.09	0.19	**0.19**
France	**Single**	0.09	0.26	0.36	0.36	0.26
	Average	0.11	0.10	0.16	0.06	**0.11**
Germany	**Single**	0.04	0.09	0.12	0.80	0.26
	Average	0.04	0.11	0.08	0.27	**0.12**
India	**Single**	0.02	0.12	0.15	0.48	0.19
	Average	0.03	0.07	0.05	0.25	**0.10**
Italy	**Single**	0.08	0.25	0.32	0.61	0.32
	Average	0.10	0.13	0.06	0.32	**0.15**
United Kingdom	**Single**	0.08	0.96	0.98	0.54	0.64
	Average	0.14	0.16	0.06	0.07	**0.11**

TABLE 4.7: Comparison of KSTS values between the single KS Test and the average KS Test for the time period 03/01/2021 – 02/04/2021. The average optimal values are marked in bold. Additionally, the optimal results are highlighted in grey.

Country	KS Test	Infections	Recoveries	Deaths	Interest	Average
South Africa	**Single**	0.09	0.49	0.49	0.72	0.45
	Average	0.12	0.11	0.08	0.10	**0.10**
Egypt	**Single**	0.03	0.10	0.05	0.61	0.20
	Average	0.04	0.05	0.04	0.10	**0.06**
France	**Single**	0.06	0.04	0.05	0.61	0.19
	Average	0.13	0.09	0.04	0.04	**0.08**
Germany	**Single**	0.03	0.22	0.29	0.33	0.22
	Average	0.13	0.05	0.08	0.09	**0.09**
India	**Single**	0.18	0.15	0.17	0.28	0.19
	Average	0.27	0.09	0.11	0.10	**0.14**
Italy	**Single**	0.05	0.17	0.28	0.28	0.19
	Average	0.07	0.04	0.04	0.11	**0.06**
United Kingdom	**Single**	0.02	0.19	0.23	0.65	0.27
	Average	0.08	0.10	0.10	0.18	**0.11**

shown in Figure 4.10 and Figure 4.11 for South Africa using where the Single model does poorly, and the Average model appears more stable than the Single model.

We have selected the worst and best configurations from the results presented in Table 4.6, Table 4.7 and Table 4.8 for the Average model to visualise. In Figure 4.12 we show a comparison between the real-world and the predicted data distributions for

TABLE 4.8: Comparison of KSTS values between the single KS Test and the average KS Test for the time period 03/04/2021 – 02/07/2021. The average optimal values are marked in bold. Additionally, the best and worst results are highlighted in dark and light grey, respectively.

Country	KS Test	Infections	Recoveries	Deaths	Interest	Average
South Africa	**Single**	0.06	0.64	0.78	0.74	0.56
	Average	0.35	0.35	0.12	0.11	**0.23**
Egypt	**Single**	0.04	0.12	0.10	0.35	0.15
	Average	0.05	0.07	0.06	0.06	**0.06**
France	**Single**	0.03	0.24	0.26	0.32	0.21
	Average	0.19	0.08	0.11	0.05	**0.11**
Germany	**Single**	0.06	0.45	0.42	0.47	0.35
	Average	0.16	0.08	0.10	0.09	**0.11**
India	**Single**	0.05	0.20	0.16	0.49	0.22
	Average	0.14	0.14	0.08	0.14	**0.12**
Italy	**Single**	0.04	0.48	0.48	0.64	0.41
	Average	0.14	0.05	0.20	0.03	**0.11**
United Kingdom	**Single**	0.14	0.37	0.23	0.58	0.33
	Average	0.29	0.13	0.08	0.21	**0.18**

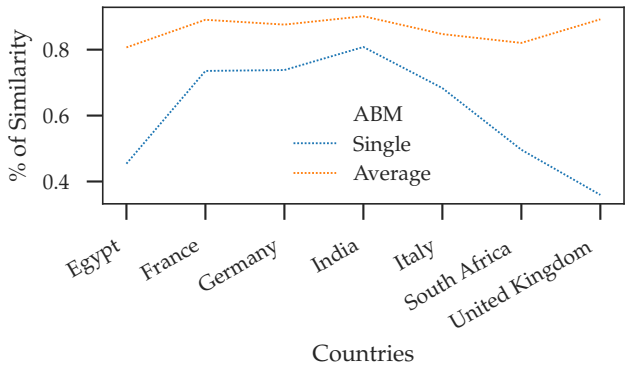


FIGURE 4.9: Performance comparison plot between the single KS Test and the average KS Test for time period 05/10/2020–02/01/2021.

the worst average KSTS value obtained, where the similarity to the real-world distributions is 77%. In addition, we show the corresponding cumulative distributions plot in Figure 4.13, where the cumulative distributions are what the ABMS framework uses to parameterise the ABM. It can be seen in Figure 4.12 that the Average model's prediction of the real-world for infections and recoveries are less similar in comparison. While in Figure 4.13 the predicted distributions for deaths and interest approximately resembles the trend of the real world. The rate at which infections and recoveries spike in Figure 4.12 is noticeable in comparison to the real world, which shows a gradual increase. In Figure 4.14 we show a comparison between the real-world and the predicted data distributions for the best average KSTS value obtained, where the similarity to the real-world distributions is 94%. The best results were obtained in Table 4.7 (Time Period 2) for Egypt and Italy and in Table 4.8 (Time

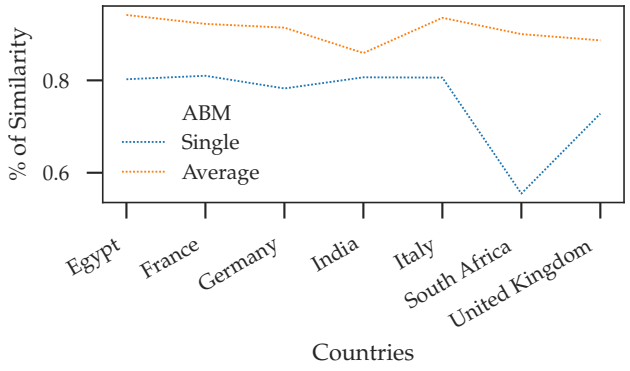


FIGURE 4.10: Performance comparison plot between the single KS Test and the average KS Test for time period 03/01/2021–02/04/2021.

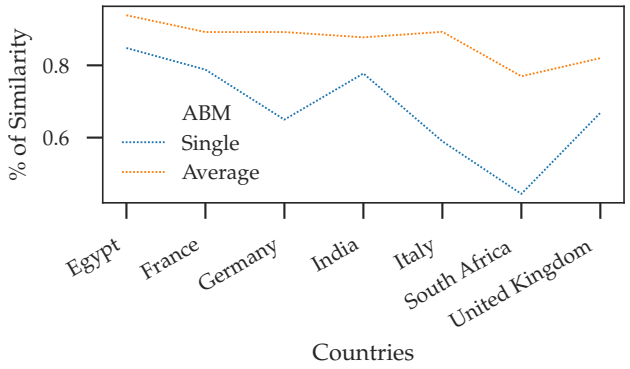


FIGURE 4.11: Performance comparison plot between the single KS Test and the average KS Test for time period 03/04/2021–02/07/2021.

Period 3) for Egypt. The trend is similar to the real world but not exact. In Figure 4.15 we show the corresponding cumulative distributions plot, where it can be seen that the fit between the real-world and the prediction is significantly closer in comparison to Figure 4.13.

4.7 Conclusion

This chapter presents a new Complex agent-based model (ABM), which models a secondary diffusive process parallel to the infectious disease spread. In addition, the new model can incorporate additional data distributions, which in turn moderate the rate at which an agent moves and interacts within the environment. This mechanism allows the agents to assimilate real-world dynamics corresponding to the policy measures enforced in the real world. We show that the Complex ABM performs better than the Susceptible-Infected-Recovered-Dead (SIRD) ABM when parameterising the real-world daily infections distribution. We have also evaluated the benefit of parameterising the ABM towards a single data distribution compared to averaging across them all. We show that there is significant merit to parameterise an ABM taking into account all output data distributions. We also note there is inherent scale variance between our models' number of agents and the number of people in the real world. Hence, we parameterise the cumulative distributions, which are

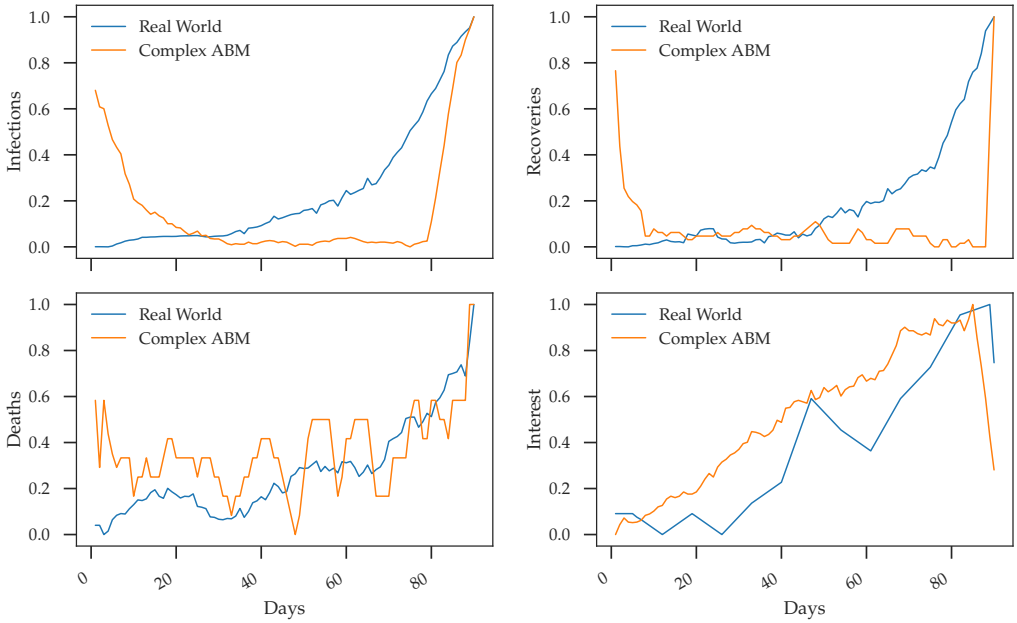


FIGURE 4.12: Comparison between the real world and the predicted data distributions for South Africa using the Average model during the third time period.

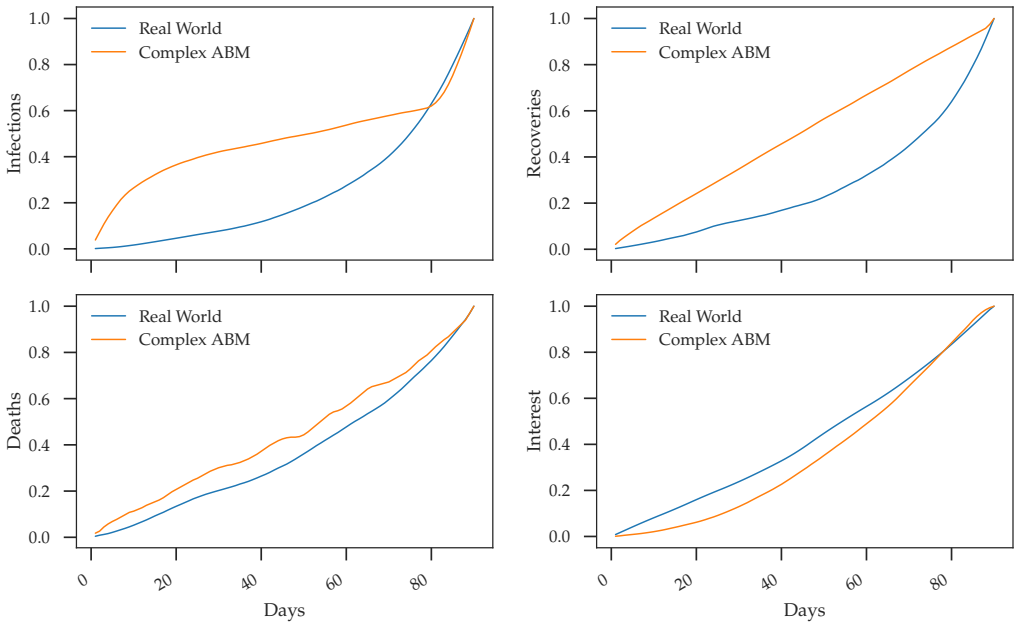


FIGURE 4.13: Comparison between the real world and the predicted cumulative distributions for South Africa using the Average model during the third time period.

scale-invariant. Lastly, we have also shown that we can approximate the real-world distributions with 77% and 94% accuracy in the worst and best cases, respectively.

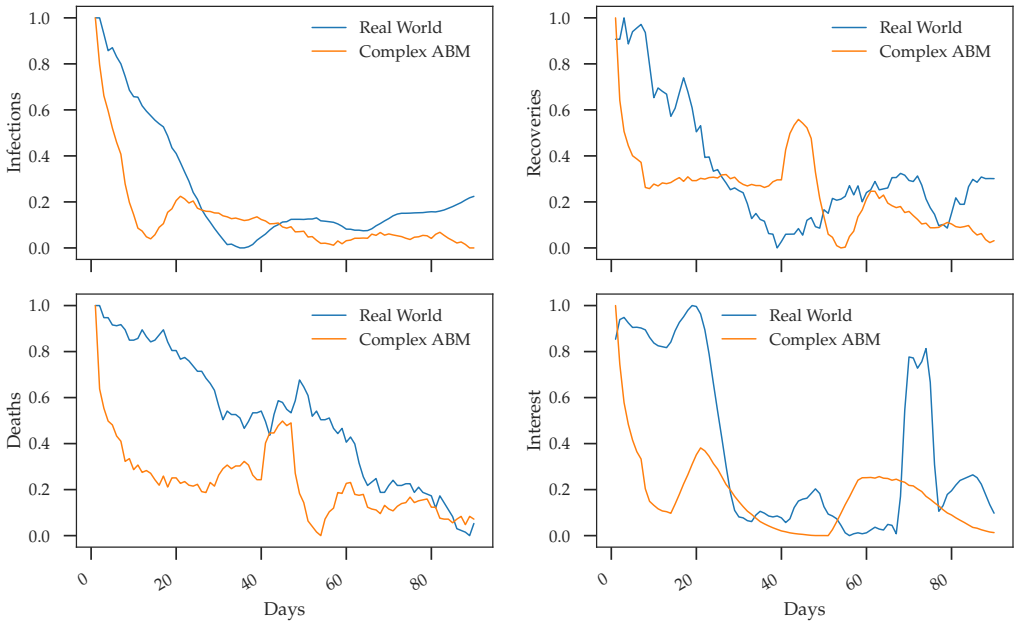


FIGURE 4.14: Comparison between the real world and the predicted data distributions for Egypt using the Average model during the second time period.

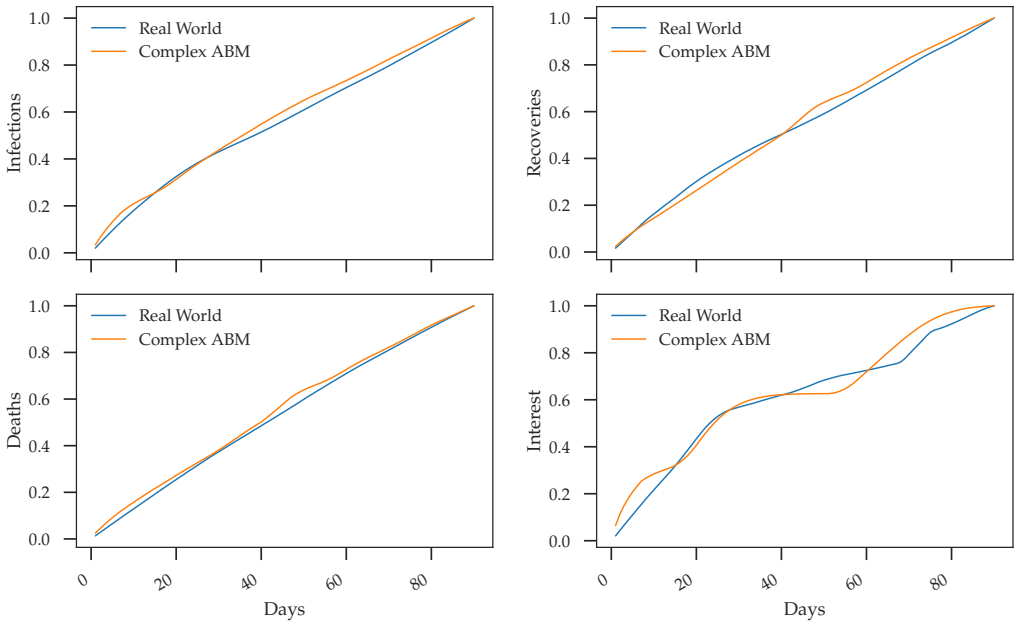


FIGURE 4.15: Comparison between the real world and the predicted cumulative distributions for Egypt using the Average model during the second time period.

Chapter 5

Surrogate Agent-Based Model

5.1 Introduction

This chapter evaluates the Long Short-Term Memory recurrent neural network (RNN) to replace the simulation model used within the agent-based modelling and simulation (ABMS) framework. The results obtained in this chapter show that:

- the LSTM network can assimilate the overall accuracy of the Complex ABM and
- using the LSTM network to replace the simulation model significantly reduces the parameterisation time within the ABMS framework.

The rest of this chapter is structured as follows. Section 5.2 explains the training data used to train the LSTM model. Section 5.3 presents the LSTM network architecture used. Section 5.4 describes the experiments conducted in this chapter. Section 5.5 presents the results obtained from carrying out the experiments along with a discussion. Lastly, Section 5.6, concludes the chapter.

5.2 Data

The data that is used in this chapter is generated using the Complex agent-based model (ABM) introduced in Chapter 4. The Complex ABM requires an input policy measure distribution, which governs the speed at which agent's move and interact within the environment. Based on the results obtained from Chapter 4, we have selected the configuration that attains the best approximation to the real world. The policy measure distribution for the country Egypt between the time period 03/01/2021 – 02/04/2021 was selected. We have randomly created 10000 parameter vector combinations between the ranges outlined in Table 5.1 using a pseudo-random sampling mechanism. Each vector is fed into the Complex ABM, and the corresponding simulation data is generated. The simulation data records the number of daily infections, recoveries, deaths and interest for the entire population across a 90 day time period.

5.3 Long Short-Term Memory Recurrent Neural Network

The Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that was proposed by Hochreiter and Schmidhuber [32] to overcome the vanishing gradient problem in standard RNNs. The LSTM architecture is well known for its ability in capturing long-term temporal dependencies and has been used previously in speech recognition, traffic flow prediction and text generation problems [34]–[36].

TABLE 5.1: Table of the ranges for each parameter value of the ABM that we have considered for parameterisation. Parameters $1, 2, 4, 5, 9, 10$ and 11 are sampled between the range $(0, 1)$. Parameters 7 and 8 are sampled between the range $(0, 0.022)$ to mimic real world interaction as the space defined is bounded by $(1, 1)$. Parameters 3 and 6 are sampled between the range $(0, 90)$ days. Lastly, parameters 12 and 13 are sampled between $(1,\ N_{Agents})$, where N_{Agents} represents the number of agents within the modelled population.

Parameters	Range
1. Disease Transmission Probability	(0.0, 1.0)
2. Disease Reinfection Probability	(0.0, 1.0)
3. Disease Infection Period	(0, 90) days
4. Interest Transmission Probability	(0.0, 1.0)
5. Interest Reinfection Probability	(0.0, 1.0)
6. Interest Infection Period Time	(0, 90) days
7. Interaction Radius	(0.0, 0.022)
8. ΔInteraction Radius	(0.0, 0.022)
9. Speed	(0.0, 1.0)
10. ΔSpeed	(0.0, 1.0)
11. Death Rate	(0.0, 1.0)
12. Initial Infected	(1, N_{Agents})
13. Initial Interested	(1, N_{Agents})

In this chapter, we use the LSTM network to replace the Complex agent-based model (ABM) within the improved ABMS framework, introduced in Chapter 3. In Figure 5.1, we can see a diagrammatic representation of the LSTM network architecture used. The LSTM network consists of four layers in total. The first layer in the network is a dense layer, where the number of neurons matches the input's size. The subsequent two layers are hidden layers consisting of 180 and 270 LSTM units, respectively. The last layer in the network is a dense layer, where the number of neurons matches the output's size. The size of the input and output layers of the LSTM network corresponds to the input and output of the Complex ABM, introduced in Chapter 4. The network is trained using machine learning techniques, where Equation 5.1 is used to calculate the training and validation loss at each epoch.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{5.1}$$

5.4 Experiment Setup

The experiments carried out in this chapter were split into two parts. First, we trained a Long Short-Term Memory (LSTM) recurrent neural network using the generated agent-based model (ABM) data. The dataset was split using an $80/20$ train test split. In order to maximise the variation in candidate type seen by the LSTM, k-fold cross-validation was used during training, where $k = 7$. The model was trained for 2000 epochs with early stopping. The model which achieved the lowest validation *RMSE* was selected to be used as a replacement to the Complex ABM within the improved ABMS framework, introduced in Chapter 3. The second experiment

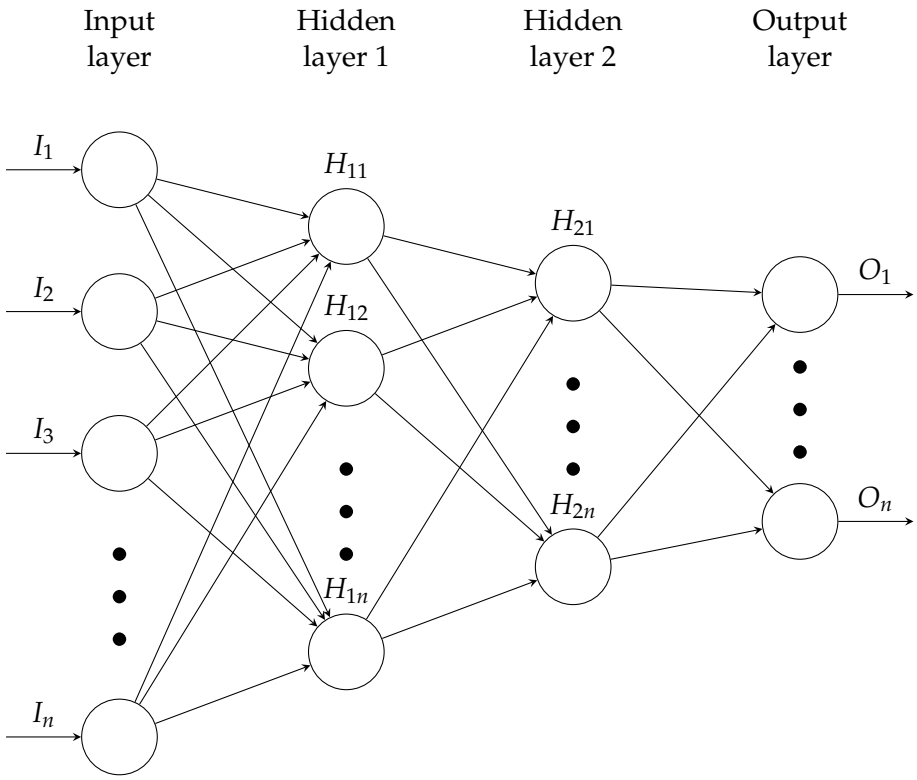


FIGURE 5.1: Long Short-Term Memory Network Architecture.

parameterised the LSTM network within the ABMS framework to attain a cumulative data distribution close to the real world. This experiment aimed to validate the use of the LSTM network as a suitable replacement to the simulation model for the parameterisation task within the framework.

The following initial ABMS framework configurations were set as: *MIN Budget* $=$ 500, *MAX Budget* $=$ 2500, *Batch Size* $=$ 250 and KSTS *Threshold* $=$ 0.005 ($\approx$ 99.5% similar to the input distribution). The strategy-surrogate pair selected predicts the real label (KSTS value), hence, we used *RMSE* where the *RMSE Threshold* $=$ 0.001 to measure the surrogates performance. The DYCORS XGBoost parameterisation strategy-surrogate pair was chosen for this experiment based off the optimal results seen in Chapter 3. In addition, the average candidate evaluation time, average strategy time and total ABMS framework time are tracked.

5.4.1 Hardware Specifications

5.4.1.1 Experiment 1

The machine used to run this experiment consisted of an AMD Ryzen 7 3700x 8-core CPU, 16GB of RAM and a GeForce RTX 2070 SUPER/PCIe/SSE2 using the Ubuntu 18.04.5 LTS operating system.

5.4.1.2 Experiment 2

The machine used to run this experiment consisted of an Intel Xeon CPU E5-2683 v4 @ 2.10GHz processor with 64 CPUs and 256GB of RAM using the Ubuntu 18.04.4 LTS operating system.

5.5 Results and Discussion

In the following tables and figures, we present the results of the above experiments, followed by a discussion of the results. The parameterisation results obtained for the optimal results presented in Chapter 3 and Chapter 4 are used as a baseline for the second experiment. The configuration of the optimal results consisted of the DYCORS XGBoost parameterisation strategy, the average Kolmogorov-Smirnov test statistic (KSTS) approach and the Complex agent-based model (ABM) for Egypt during the time period 03/01/2021–02/04/2021.

In experiment one, we train the Long Short-Term Memory (LSTM) network using the ABM data. The performance of the LSTM network was evaluated using Equation 5.1, where the model with the lowest validation RMSE was selected as the optimal. In Figure 5.2 and Figure 5.3, we can see the RMSE during training and validation for each epoch, respectively. The early stopping criteria is activated during folds 1, 2, 3, 5 and 6. In folds 4 and 7, the RMSE loss decreases until the total number of epochs are reached. In Figure 5.3, the LSTM network trained on Fold 4 attains the lowest validation accuracy and hence was selected as the optimal model.

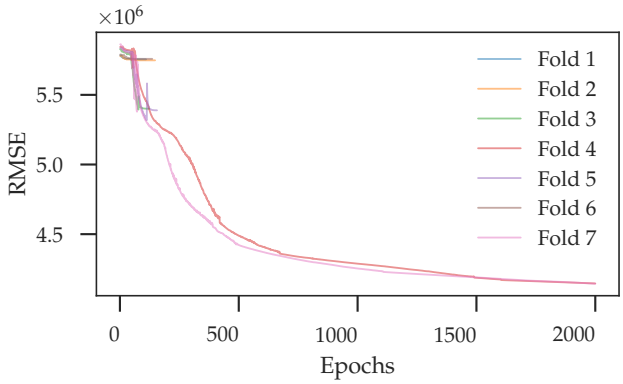


FIGURE 5.2: Comparison between the training RMSE obtained by the LSTM network in Figure 5.1 for each *k*th fold, using early stopping.

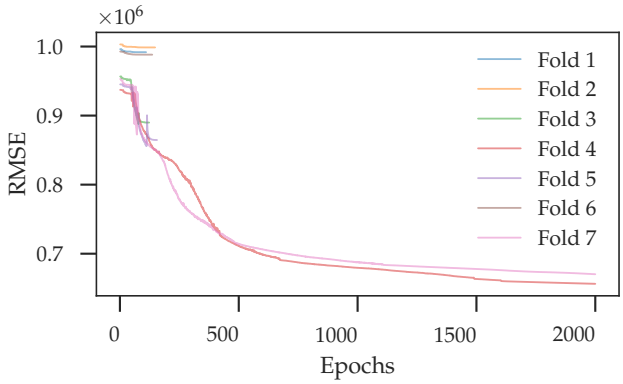


FIGURE 5.3: Comparison between the validation RMSE obtained by the LSTM network in Figure 5.1 for each *k*th fold, using early stopping.

In order to assess that the LSTM works as expected within the ABMS framework, we parameterise the LSTM network to the cumulative data distributions of a simulated epidemic based on the optimal parameter vector predicted for Egypt during the time period 03/01/2021 – 02/04/2021. We ran 30 ABM simulations using the optimal predicted parameter vector from Chapter 4 and compared that to the prediction of the LSTM. In Figure 5.4, we can see those data distributions of the 30 ABM simulations, which have been plotted with an alpha value and the average prediction plotted by the solid line. In addition, the LSTM and the real-world data are plotted alongside for comparison. The overall trend of the LSTM's prediction is more stable than the Complex ABM. Thus, the LSTM is able to reduce the stochasticity of the ABM, producing a smoother approximation. In Figure 5.5, we can see the cumulative distribution of the same result. In addition, we can see that the LSTM can approximate the distribution of the deaths with a tighter fit, whereas the Complex ABM can approximate interest with a better fit. Looking at Figure 5.5, it is worth noting that the cumulative distribution can also reduce the stochasticity of the simulation.

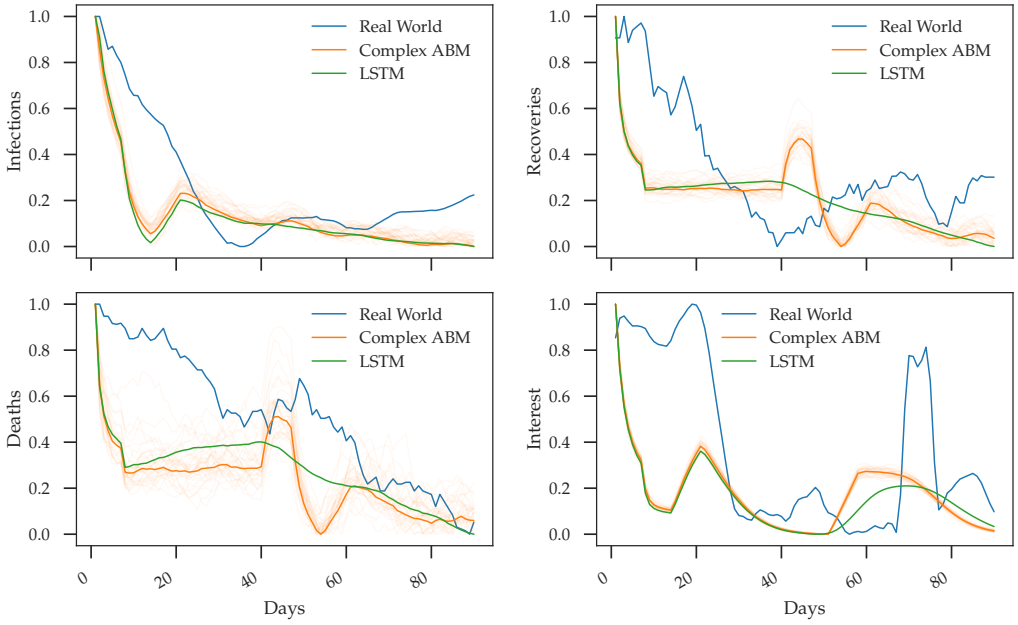


FIGURE 5.4: Data distribution of parameterising the LSTM network to an epidemic simulation of the optimal prediction for Egypt during the time period 03/01/2021 – 02/04/2021.

The second experiment consists of replacing the ABM within the ABMS framework with the pre-trained LSTM network. In Table 5.2 we compared the approximation to the real-world distributions between the Complex ABM and the LSTM network using the average KSTS value. The Complex ABM can approximate the cumulative real-world distributions with an overall similarity of 94%. Similarly, the LSTM network can approximate the cumulative real-world distributions with the same overall accuracy. However, we can see in Table 5.2 that the KSTS value between each cumulative distribution for the Complex ABM and the LSTM varies. In particular, the Complex ABM can better approximate interest, whereas the LSTM is able to better approximate deaths.

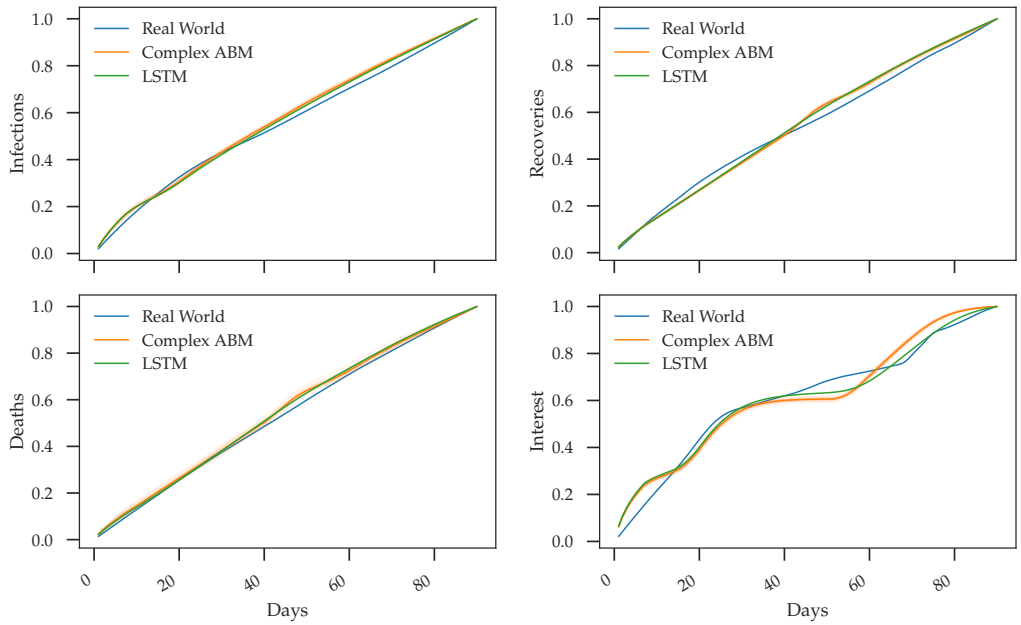


FIGURE 5.5: Cumulative distribution of parameterising the LSTM network to an epidemic simulation of the optimal prediction for Egypt during the time period 03/01/2021 – 02/04/2021.

TABLE 5.2: Comparison of the KSTS values between the using the Complex ABM and the LSTM within the ABMS framework for Egypt during the time period 03/01/2021 – 02/04/2021.

Simulation Type	Infections	Recoveries	Deaths	Interest	Average
Complex ABM	0.04	0.05	0.04	0.10	0.06
LSTM	0.05	0.04	0.02	0.14	0.06

The values presented in Table 5.3 represent the time in seconds for three different areas within the ABMS framework, which deals with the ABM. The average parameter evaluation time, average strategy time and the overall ABMS framework time is recorded. We can see looking at Table 5.3 there is a significant reduction in speed attained by using the LSTM as a replacement for the Complex ABM. In Table 5.4, we measure the speed up achieved in each sector that was measured by time. It is clear to see that the speedup attained by the LSTM over the Complex ABM is remarkable.

TABLE 5.3: Comparison between the Complex ABM and the LSTM network for the time taken in seconds for each part of the ABMS framework that deals with parameter evaluation.

Simulation Type	Parameter Evaluation	Strategy	ABMS Framework
Complex ABM	2.766	1369.467	26799.089
LSTM	0.001	33.957	9.481

TABLE 5.4: Comparison between the speed up achieved by the LSTM as a replacement for the Complex ABM within the ABMS framework.

Simulation Type	Parameter Evaluation	Strategy	ABMS Framework
Complex ABM	-	-	-
LSTM	2709	40	2827

5.6 Conclusion

In this chapter, we have evaluated the Long Short-Term Memory (LSTM) network as a replacement to the Complex agent-based model (ABM) used within the agent-based modelling and simulation (ABMS) framework. A set of random parameter vectors were generated and passed to the ABM to generate the training data for the LSTM network. Since we are comparing the outputs of our experiments to the optimal configuration observed in Chapter 4, the ABM was initialised to use the policy distributions for Egypt during the time period 03/01/2021 – 02/04/2021. The LSTM and Complex ABM can approximate the real-world cumulative distributions with the same overall accuracy of 94%. However, the accuracy between the individual output distributions vary. In addition, the speedup obtained using the LSTM as a replacement to the Complex ABM within the ABMS framework is remarkable. The average parameter evaluation time, the average strategy time and the total ABMS framework time are drastically reduced using the LSTM network. While the initial training time of the LSTM is expensive, the overall speedup achieved is still a significant reduction compared to the Complex ABM. The above results lead to the answer in determining the trade-off between accuracy and efficiency regarding using a machine learning model to replace the simulation model. It is also worth noting that the performance of the LSTM is dependent on the accuracy of the Complex ABM to the real world.

Chapter 6

Conclusion

Accurate and efficient simulation models for decision analysis and policy-making during an epidemic are vital for preventing infectious diseases. Agent-based models (ABMs) are more complex than previously used compartmental models to this problem. However, there are limitations in ABMS such as model validation, parameter calibration and long simulation time [2], [3]. Surrogate models (SM) learned through machine learning (ML) algorithms can function as a computational approximation to an agent-based model (ABM). However, they are potentially less accurate than the original ABM [5], [6]. In addition, they can be used to aid parameter space exploration and calibration.

An agent-based modelling and simulation (ABMS) framework was developed to compare surrogate-assisted sampling methods for comparison. The results obtained show that standard sampling approaches such as Random and Sobol cannot efficiently search the parameter space of an ABM. However, a surrogate assisted sampling approach can attain a closer approximation to the cumulative data distributions used to parameterise the ABM.

An improved version of the ABMS framework is created to assist the evaluation of different surrogate assisted parameterisation strategies in general. Surrogate assisted optimisation strategies from the literature have been modified to work as parameterisation strategies. In addition, a new evolutionary parameterisation strategy is presented. Through the improved ABMS framework, we evaluated and identified the optimal strategy-surrogate combination in terms of accuracy and efficiency. The Dynamic Coordinate Search Using Response Surface Models (DYCORS) strategy using the XGBoost SM parameterised the Susceptible-Infected-Recovered-Dead (SIRD) ABM towards a synthetic epidemic with the highest accuracy on average and the most speedup.

A new Complex ABM is introduced, which models two diffusive processes in parallel. The first process models the spread of the infectious disease within the population, and the second models individuals interest in the spread of the disease. In addition, the Complex ABM incorporates the policy measures set out by the government of a particular country to regulate the speed at which agents move and interact within their environment. The accuracy between the SIRD and Complex ABMs are compared through the ABMS framework to the real-world distribution of infections for the COVID-19 virus across three time periods. The Complex ABM approximates the real-world infections better than the SIRD ABM across the time periods considered.
Further, an average similarity value across each output distribution of the Complex

ABM is used for parameterisation compared to using a single distribution. The average approach is superior and more stable than the single distribution approach across the different countries evaluated. The ABMS framework was able to attain an overall similarity score of 77% and 94% accuracy in the worst and best cases, respectively, compared to the real-world distributions.

A Long Short-Term Memory (LSTM) recurrent neural network is proposed as an alternative to the Complex ABM to tackle long simulation time. The LSTM network can assimilate the dynamics of the Complex ABM with on par accuracy. In addition, the LSTM network significantly reduces the parameterisation time.

Through the research conducted in this dissertation, it has been shown that surrogate assisted sampling approaches are better in terms of accuracy than stand-alone sampling methods. Therefore a more complex sampling approach is well suited for parameter space exploration. Surrogate models can aid parameterisation strategies reducing parameter exploration time whilst still maintaining accuracy. An ABM that incorporates more complexity can approximate the real-world data with greater accuracy and stability than more straightforward approaches. Lastly, a surrogate replacement in the form of an LSTM network can maintain accuracy with regards to the ABM whilst significantly reducing simulation time. The future work of this research aims to explore the use of a surrogate model within reinforcement learning, where a simulation model is not available.

Bibliography

[1] R. Beaglehole, R. Bonita, T. Kjellström, *et al.*, *Basic epidemiology*. World Health Organization Geneva, 1993.

[2] E. Hunter, B. Mac Namee, and J. D. Kelleher, "A taxonomy for agent-based models in human infectious disease epidemiology," *Journal of Artificial Societies and Social Simulation*, vol. 20, no. 3, 2017.

[3] F. Miksch, B. Jahn, K. J. Espinosa, J. Chhatwal, U. Siebert, and N. Popper, "Why should we apply abm for decision analysis for infectious diseases?—an example for dengue interventions," *PloS one*, vol. 14, no. 8, 2019.

[4] C. E. Walters, M. M. Meslé, and I. M. Hall, "Modelling the global spread of diseases: A review of current practice and capability," *Epidemics*, vol. 25, pp. 1–8, 2018.

[5] F. Lamperti, A. Roventini, and A. Sani, "Agent-based model calibration using machine learning surrogates," *Journal of Economic Dynamics and Control*, vol. 90, pp. 366–389, 2018.

[6] Y. Zhang, Z. Li, and Y. Zhang, "Validation and calibration of an agent-based model: A surrogate approach," *Discrete Dynamics in Nature and Society*, vol. 2020, 2020.

[7] C. M. Macal and M. J. North, "Agent-based modeling and simulation," in *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 2009, pp. 86–98.

[8] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002, ISSN: 0027-8424. DOI: 10.1073/pnas.082080899. eprint: https://www.pnas.org/content/99/suppl_3/7280.full.pdf. [Online]. Available: https://www.pnas.org/content/99/suppl_3/7280.

[9] M. Tracy, M. Cerdá, and K. M. Keyes, "Agent-based modeling in public health: Current applications and future directions," *Annual review of public health*, vol. 39, pp. 77–94, 2018.

[10] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, vol. 115, no. 772, pp. 700–721, 1927.

[11] D. M. Aleman, T. G. Wibisono, and B. Schwartz, "A nonhomogeneous agent-based simulation approach to modeling the spread of disease in a pandemic outbreak," *Interfaces*, vol. 41, no. 3, 301–315, May 2011, ISSN: 0092-2102. DOI: 10.1287/inte.1100.0550. [Online]. Available: https://doi.org/10.1287/inte.1100.0550.

[12] L. Mao, "Modeling triple-diffusions of infectious diseases, information, and preventive behaviors through a metropolitan social network—an agent-based simulation," *Applied Geography*, vol. 50, pp. 31 –39, 2014, ISSN: 0143-6228. DOI: https://doi.org/10.1016/j.apgeog.2014.02.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0143622814000277.

[13] P. Cooley, S. Brown, J. Cajka, B. Chasteen, L. Ganapathi, J. Grefenstette, C. R. Hollingsworth, B. Y. Lee, B. Levine, W. D. Wheaton, *et al.*, "The role of subway travel in an influenza epidemic: A new york city simulation," *Journal of Urban Health*, vol. 88, no. 5, p. 982, 2011.

[14] J. Hackl and T. Dubernet, "Epidemic spreading in urban areas using agent-based transportation models," *Future Internet*, vol. 11, no. 4, p. 92, 2019.

[15] J. Grazzini, M. G. Richiardi, and M. Tsionas, "Bayesian estimation of agent-based models," *Journal of Economic Dynamics and Control*, vol. 77, pp. 26–47, 2017.

[16] S. van der Hoog, "Surrogate modelling in (and of) agent-based models: A prospectus," *Computational Economics*, vol. 53, no. 3, pp. 1245–1263, 2019.

[17] L. Bargigli, L. Riccetti, A. Russo, and M. Gallegati, "Network calibration and metamodeling of a financial accelerator agent based model," *Journal of Economic Interaction and Coordination*, pp. 1–28, 2018.

[18] G. Dosi, M. C. Pereira, A. Roventini, and M. E. Virgillito, "The effects of labour market reforms upon unemployment and income inequalities: An agent-based model," *Socio-Economic Review*, vol. 16, no. 4, pp. 687–720, 2018.

[19] W. A. Brock and C. H. Hommes, "Heterogeneous beliefs and routes to chaos in a simple asset pricing model," *Journal of Economic dynamics and Control*, vol. 22, no. 8-9, pp. 1235–1274, 1998.

[20] G. Fagiolo and G. Dosi, "Exploitation, exploration and innovation in a model of endogenous growth with locally interacting agents," *Structural Change and Economic Dynamics*, vol. 14, no. 3, pp. 237–273, 2003.

[21] R. G. Regis and C. A. Shoemaker, "Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization," *Engineering Optimization*, vol. 45, no. 5, pp. 529–555, 2013.

[22] ——, "A stochastic radial basis function method for the global optimization of expensive functions," *INFORMS Journal on Computing*, vol. 19, no. 4, pp. 497–509, 2007.

[23] A. E. Eiben, J. E. Smith, *et al.*, *Introduction to evolutionary computing*. Springer, 2003, vol. 53.

[24] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–a comprehensive introduction," *Natural computing*, vol. 1, no. 1, pp. 3–52, 2002.

[25] B. Doerr, C. Doerr, and F. Ebel, "From black–box complexity to designing new genetic algorithms," *Theoretical Computer Science*, vol. 567, pp. 87–104, 2015.

[26] B. Doerr and C. Doerr, "Optimal static and self-adjusting parameter choices for the $(1 + (\lambda,\lambda))$ genetic algorithm," *Algorithmica*, vol. 80, no. 5, pp. 1658–1709, 2018.

[27] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[28] V. S. Dave and K. Dutta, "Neural network based models for software effort estimation: A review," *Artificial Intelligence Review*, vol. 42, no. 2, pp. 295–307, 2014.

[29] T. Katte, "Recurrent neural network and its various architecture types," *International Journal of Research and Scientific Innovation*, vol. 5, 3 2018, ISSN: 2321-2705.

[30] J. Chung, Çaglar Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[31] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Trans. Neur. Netw.*, vol. 5, no. 2, pp. 157–166, 1994, ISSN: 1045-9227.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 103–111.

[34] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*, IEEE, 2013, pp. 273–278.

[35] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328. DOI: 10.1109/YAC.2016.7804912.

[36] D Pawade, A Sakhapara, M Jain, N Jain, and K Gada, "Story scrambler-automatic text generation using word level rnn-lstm," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 10, no. 6, pp. 44–53, 2018.

[37] R. Perumal and T. L. v. Zyl, "Comparison of recurrent neural network architectures for wildfire spread modelling," in *2020 International SAUPEC/RobMech/PRASA Conference*, 2020, pp. 1–6. DOI: 10.1109/SAUPEC/RobMech/PRASA48453.2020.9078028.

[38] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017, ISSN: 2162-237X.

[39] T. Iqbal and S. Qureshi, "The survey: Text generation models in deep learning," *Journal of King Saud University-Computer and Information Sciences*, 2020.

[40] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602 –610, 2005, IJCNN 2005, ISSN: 0893-6080.

[41] J. Bermúdez, P Achanccaray, I. Sanches, L Cue, P Happ, and R. Feitosa, "Evaluation of recurrent neural networks for crop recognition from multitemporal remote sensing images," in *Anais do XXVII Congresso Brasileiro de Cartografia*, 2017, pp. 800–804.

[42] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, 2003, ISSN: 1532-4435.

[43] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[44] P. Bratley and B. L. Fox, "Algorithm 659: Implementing sobol's quasirandom sequence generator," *ACM Trans. Math. Softw.*, vol. 14, no. 1, 88–100, Mar. 1988. DOI: 10.1145/42288.214372.

[45] S. Joe and F. Y. Kuo, "Remark on algorithm 659: Implementing sobol's quasirandom sequence generator," *ACM Trans. Math. Softw.*, vol. 29, no. 1, 49–57, Mar. 2003. DOI: 10.1145/641876.641879.

[46] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[47] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, "Xgboost: Extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.

[48] P. H. Swain and H. Hauska, "The decision tree classifier: Design and potential," *IEEE Transactions on Geoscience Electronics*, vol. 15, no. 3, pp. 142–147, 1977.

[49] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[50] B. A. Tolson and C. A. Shoemaker, "Dynamically dimensioned search algorithm for computationally efficient watershed model calibration," *Water Resources Research*, vol. 43, no. 1, 2007.

[51] W. J. Morokoff and R. E. Caflisch, "Quasi-random sequences and their discrepancies," *SIAM Journal on Scientific Computing*, vol. 15, no. 6, pp. 1251–1279, 1994.

[52] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of IEEE international conference on evolutionary computation*, IEEE, 1996, pp. 312–317.

[53] T. Hale, N. Angrist, R. Goldszmidt, B. Kira, A. Petherick, T. Phillips, S. Webster, E. Cameron-Blake, L. Hallas, S. Majumdar, *et al.*, "A global panel database of pandemic policies (oxford covid-19 government response tracker)," *Nature Human Behaviour*, vol. 5, no. 4, pp. 529–538, 2021.

[54] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track covid-19 in real time," *The Lancet infectious diseases*, vol. 20, no. 5, pp. 533–534, 2020.