On the 3 M's of Epidemic Forecasting: Methods, Measures, and Metrics.

Farzaneh S. Tabataba

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Madhav V. Marathe, Chair
Bryan L. Lewis, Co-Chair
Anil Kumar S. Vullikanti
Naren Ramakrishnan
Dave M. Higdon
John S. Brownstein

September 20, 2017
Blacksburg, Virginia

Keywords: Epidemic forecasting, Data assimilation, Calibration, Forecast Evaluation.

On the 3 M's of Epidemic Forecasting: Methods, Measures, and Metrics.

Farzaneh S. Tabataba

(ABSTRACT)

Over the past few decades, various computational and mathematical methodologies have been proposed for forecasting seasonal epidemics. In recent years, the deadly effects of enormous pandemics such as the H1N1 influenza virus, Ebola, and Zika, have compelled scientists to find new ways to improve the reliability and accuracy of epidemic forecasts. The improvement and variety of these prediction methods are undeniable. Nevertheless, many challenges remain unresolved in the path of forecasting the outbreaks using surveillance data. Obtaining the clean real-time data has always been an obstacle. Moreover, the surveillance data is usually noisy and handling the uncertainty of the observed data is a major issue for forecasting algorithms. Correct modeling assumptions regarding the nature of the infectious disease is another dilemma. Oversimplified models could lead to inaccurate forecasts, whereas more complicated methods require additional computational resources and information. Without those, the model may not be able to converge to a unique optimum solution. Through the last decade, there has been a significant effort towards achieving better epidemic forecasting algorithms. However, the lack of standard, well-defined evaluating metrics impedes a fair judgment on the proposed methods.

This dissertation is divided into two parts. In the first part, we present a Bayesian particle filter calibration framework integrated with an agent-based model to forecast the epidemic trend of diseases like flu and Ebola. Our approach uses Bayesian statistics to estimate the underlying disease model parameters given the observed data and handle the uncertainty in the reasoning. An individual-based model with different intervention strategies could result in a large number of unknown parameters that should be properly calibrated. As particle filter could collapse in very large-scale systems (curse-of-dimensionality problem), achieving the optimum solution becomes more challenging. Our proposed particle filter framework utilizes machine learning concepts to restrain the intractable search space. It incorporates a smart analyzer in the state dynamics unit that examines the predicted and observed data using machine learning techniques to guide the direction and amount of perturbation of each parameter in the searching process.

The second part of this dissertation focuses on providing standard evaluation measures for evaluating epidemic forecasts. We present an end-to-end framework that introduces epidemiologically relevant features (Epi-features), error measures, and ranking schema as the main modules of the evaluation process. Lastly, we provide the evaluation framework as a software package named Epi-Evaluator and demonstrate the potentials and capabilities of the framework by applying it to the output of different forecasting methods.

On the 3 M's of Epidemic Forecasting: Methods, Measures, and Metrics.

Farzaneh S. Tabataba

(GENERAL AUDIENCE ABSTRACT)

Epidemics impose substantial costs to societies by deteriorating the public health and disrupting economic trends. In recent years, the deadly effects of wide-spread pandemics such as H1N1, Ebola, and Zika, have compelled scientists to find new ways to improve the reliability and accuracy of epidemic forecasts. The reliable prediction of future pandemics and providing efficient intervention plans for health care providers could prevent or control disease propagations. Over the last decade, there has been a significant effort towards achieving better epidemic forecasting algorithms. The mission, however, is far from accomplished. Moreover, there has been no significant leap towards standard, well-defined evaluating metrics and criteria for a fair performance judgment between the proposed methods.

This dissertation is divided into two parts. In the first part, we present a Bayesian particle filter calibration framework integrated with an agent-based model to forecast the epidemic trend of diseases like flu and Ebola. We model the disease propagation via a large scale agent-based model that simulates the disease spread across the contact network of people. The contact network consists of millions of nodes and is constructed based on demographic information of individuals achieved from the census data. The agent-based model's configurations are mostly unknown parameters that should be properly calibrated. We present a Bayesian particle filter calibration approach to estimate the underlying disease model parameters given the observed data and handle the uncertainty in the reasoning. As particle filter could collapse in very large-scale systems, achieving the optimum solution becomes more challenging. Our proposed particle filter framework utilizes machine learning concepts to restrain the intractable search space. It incorporates a smart analyzer unit that examines the predicted and observed data using machine learning techniques to guide the direction and amount of perturbation of each parameter in the searching process.

The second part of this dissertation focuses on providing standard evaluation measures for evaluating and comparing epidemic forecasts. We present a framework that introduces epidemiologically relevant features (Epi-features), error measures, and ranking schema as the main modules of the evaluation process. Lastly, we provide the evaluation framework as a software package named Epi-Evaluator and demonstrate the potentials and capabilities of the framework by applying it to the output of different forecasting methods.

To my mother, my father,

and to my love, Milad.

Acknowledgments

I would like to thank my advisor, Prof. Madhav Marathe, for all his support and words of wisdom. This project would have never made it to the end without his confidence in me and his guidance. I am also grateful to my co-advisor, Dr. Bryan Lewis, for insightful comments and encouragement. This project would not have advanced so quickly if it was not for his willingness to pass on his knowledge. I also want to appreciate the time and effort of my committee members, and their willingness to serve on my committee. I am grateful for their feedback along with the hard questions that incented me to widen my research from different perspectives. I would like to extend my sincere gratitude to Erin Raymond for her editorial assistance in reviewing drafts of my publications.

I am grateful to my mother and my father, Fereshteh Kazemi and Hesam Tabataba, who have always been with me, regardless of thousands of physical distance between us, to support me mentally and emotionally throughout this journey. If I have something today, it is because of their sacrifices and unconditional love. I would also like to thank my beloved sisters, Foroogh and Farima, for their encouragement and belief in me. Dr. Foroogh Tabataba is a professor at the Isfahan University of Technology. I was blessed by her professional advice in every step of the way to improve my work.

I am also grateful to many friends who have stood by me in all hardships. Especial thanks to my beloved friend, Shadi Esmaeili, for her unconditional kindness and thoughtfulness. A special thank you is due to Betty and Wayne Coleman, Ginger and James Travis, Mahnaz Asghari, and Kimberly Coleman for all their heart-warming support and prayers. Having them in my life is such a big blessing.

Last, but the most important, I would love to thank my beloved husband, Milad Hosseinipour, who has been my best friend and consultant in this journey. I could not accomplish this undertaking and overcome the challenges without his technical support and encouragement. He carried me through the storms and held my hand in the hard times to facilitate my road to success. I am eternally grateful for having him in my life.

Farzaneh S. Tabataba
September 2017

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Every year epidemics impose incalculable losses on health and welfare of human societies. The implications of epidemics on a population can vary from deteriorating the public health to affecting economy growth and development through loss of productivity. Only in the US, influenza outbreaks cause an annual average of 610000 deaths and 3 million hospitalization days (Molinari et al., 2007). Between April 2009 and April 2010, more than 60 million cases of H1N1 were reported in the US. Wars, loss of social cohesion, and natural disasters have catalyzed many Emerging Infectious Diseases (EIDs). Under these conditions, if an epidemic is not sufficiently contained, it can become a pandemic outbreak with no social or geographical boundaries. Recent outbreaks like Ebola in West Africa (2014), Cholera in Bangladesh, and Zika in Brazil demonstrate an unprecedented threat of pandemics worldwide.

Advances in medicine, science, and technology have helped societies cope with certain infectious diseases. The impact of an epidemic on human societies depends on its imminent and long-term severity. Various factors such as rate of hospitalization, insurance premiums, outpatient visits, and most importantly, death, can influence the impact. Reliable prediction of outbreaks has been proven an effective tool in alleviating the epidemics through providing vital information for interventions' planning and targeted resource distributions.

Over the past few decades, numerous forecasting methods have been proposed. Such methods can be classified based on various criteria such as deterministic vs. probabilistic, or comparative vs. generative. Typically, predictive models receive their input in the form of a time-series of epidemiological data from early stages of an outbreak and predict data points in the future or the remainder of the season. The improvement and variety of these prediction methods are undeniable. Nevertheless, many challenges remain unresolved in the path of forecasting the outbreaks using surveillance data. Obtaining the clean real-time data has always been an obstacle. Moreover, the surveillance data is usually noisy and handling

the uncertainty of the observed data is a major issue for forecasting algorithms. Modeling assumptions regarding the nature of the infectious disease is another problem. Oversimplified models could lead to inaccurate forecasts, whereas more complicated methods require additional computational resources and information. Without those, the model man not be able to converge to a unique optimum solution. Through the last decade, various forecasting algorithms have claimed improvements in accuracy and reliability of their predictions. However, the lack of standard, well-defined evaluation measures to compare the proposed algorithms further hurdles the development of epidemic forecasting techniques.

1.2 Problem Statement

1.2.1 Epidemic Forecasting Problem

Most predictive models receive their input in the form of a time-series of the epidemiological data from early stages of an outbreak. The epidemic data is usually structured in a form such as $y(1), ..., y(t), .., y(T)$, where $y(t)$ denotes the number of newly infected cases observed at time t, and T is the duration of the epidemic season. Weekly time-steps are often preferred over daily steps due to their averaging nature that filters the noise in daily case counts, and hence decreases the uncertainty.

Let us denote the *prediction time* by k and the prediction horizon by w. Given the early time-series up to time k $(y(1), ..., y(k))$ as the observed data, the forecasting algorithm predicts the time-series up to the prediction horizon as $x(k+1), ..., x(k+w)$ (Figure 1.1). The forecasts could be short-term (small w), or long-term ($w = T - k$). Some algorithms are capable of updating the prediction when a new data point is observed. While some forecasting algorithms provide a deterministic output, the others generate stochastic results that bear uncertainty. Some methods are only capable of forecasting the epidemic curve, whereas others can provide more detailed information like endangered sub-populations, or more efficient preventive plans. Some algorithms are vulnerable to noisy data, and some are not.

The primary objective in the area of epidemic forecasting is to provide a more accurate algorithm that can deliver reliable foresight to health providers for better planning and resource allocation. On the other hand, the desired algorithm should be affordable in terms of time and computing resources to supply real-time information.

1.2.2 Evaluation Challenge

The past few decades have witnessed a significant effort in developing better and more reliable forecasting techniques. Various national and international agencies have recently embarked

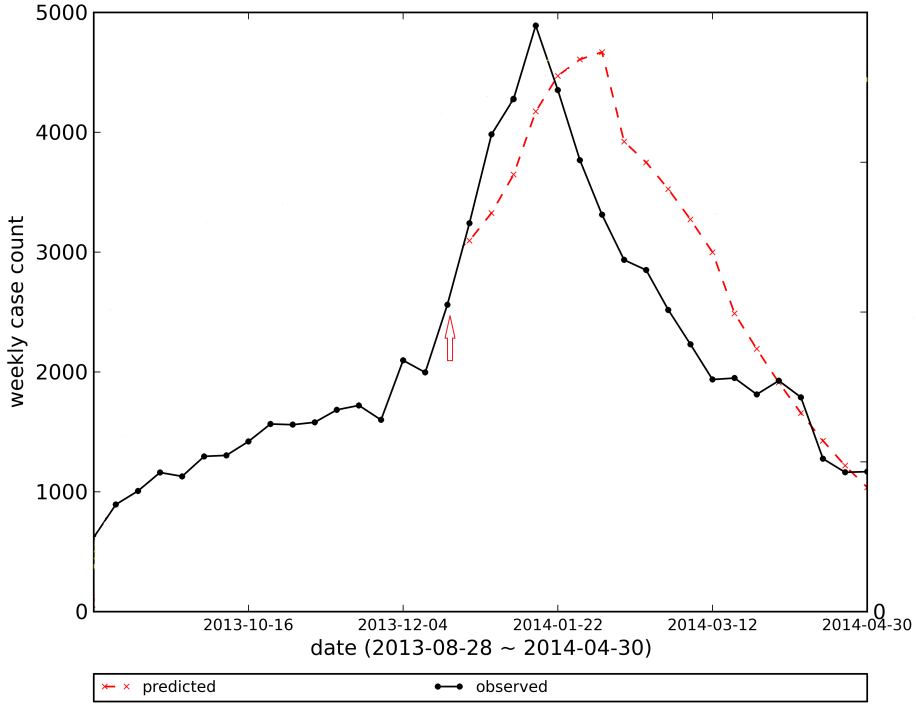


Figure 1.1. Predicting the epidemic curve. The red arrow points to the prediction time k in which prediction occurs based on k initial data points. The dashed red line is the predicted epidemic curve and the black line is the observed one.

on a mission to promote forecasting challenges to create a standard platform for comparing the performance of these methods. Few of these organizations are the Center for Disease Control and Prevention (CDC), National Institutes of Health (NIH), Department of Health and Human Services (HHS), National Oceanic and Atmospheric Administration(NOAA), and Defense Advanced Research Projects Agency (DARPA). These challenges target major cost-imposing outbreaks such as Flu[2], Ebola[3], Dengue[4, 5] and Chikungunya[6].

Fair evaluation and comparison of the proposed outputs of different forecasting methods have remained an open question. In an effort to provide the means for assessing the performance of various time-series forecasting methods[7, 8], three competitions were held under the name of Makridakis Competitions (M-Competitions) in 1982, 1993, and 2000. In their analysis, the accuracy of different methods was estimated by calculating different error measures on business and economic time-series which apply to other disciplines. The competitions were aimed at predicting financial time-series. Although their analysis is generic, it does not consider properties of epidemiologically relevant time-series. The lack of standard, well-defined evaluation measures to select the best algorithm among different ones, as well as for selecting the best possible configuration for a particular algorithm, is still a significant problem.

1.3 Dissertation Objectives and Contributions

This dissertation is divided into two main sections: I. epidemic forecasting algorithms, and II. a framework for evaluating epidemic forecasts. In the first part, we present an agent-based-Bayesian algorithm to forecast the epidemic curve and estimate the underlying disease model parameters. Our approach uses the concept of Bayesian statistics to infer the model parameters given the observed data and handle the uncertainty in the reasoning. Moreover, the proposed model utilizes machine learning concepts to restrain the intractable search space.

The Bayesian approach presented in this dissertation departs from previously proposed methods; it uses an individual-based model to simulate the spread of the disease across the contact network. Built on this foundation, we infer the unknown parameters of a causal model that might have given rise to observed pandemic until the current week of the season. The novelty of this approach lies in:

(i) Integrating an agent-based model with a network representation of different populations, as the disease propagation simulator, combined with Bayesian particle filter. Available particle filter approaches, use ODE compartment models like SIR and SEIR [9, 10, 11] as the disease model and apply Bayesian calibrators to adjust the observed and predicted epidemic curves together. Based on our knowledge at the time of writing this dissertation, no one has used a network agent-based model as the disease model combined with the particle filter framework as the data assimilation method. Besides the fidelity of agent-based model and its consistency with real-world disease propagation, it is able to provide more detailed outputs compared with the naive compartment models.

(ii) Incorporating a smart analyzer in the state dynamics unit, that examines the predicted and observed data using machine learning techniques to guide the direction and amount of perturbation of each parameter in the searching process. An individual-based model with different intervention strategies could result in a large number of unknown parameters that should be properly calibrated. As particle filter could collapse in very large-scale systems (curse-of-dimensionality problem (Bengtsson et al., 2008)), it is more challenging to achieve the optimum solution in this case. The smart diffuser enables the particle filter to look for better results with fewer particles and fewer iterations [12].

The second part of this dissertation focuses on providing standard evaluation measures for evaluating epidemic forecasts [13]. As discussed earlier, there has been a significant effort towards achieving better epidemic forecasting algorithms. However, the lack of standard, well-defined evaluating metrics impedes a fair judgment on the proposed methods. We present an end-to-end framework for evaluating epidemic forecasts. Three main parts constitute this framework:

(i) Epi-Features Extractor: We quantify the output of the forecasting algorithm by extracting epidemiologically relevant features from the predicted epidemic curves. We refer to these epidemiologically relevant features as *Epi-features*.

(ii) Error Measures Unit: The accuracy of a model's estimate of a particular Epi-features is to be quantified by evaluating its error with respect to the Epi-features extracted from the ground truth. This is enabled by using functions that capture their dissimilarity, which we refer to as *error measures*.

(iii) Ranking Schema Unit: In addition to quantifying the performance of each method, the Ranking Schema Unit enable the framework to compare the output of various forecasting methods by ranking them. This unit incorporates two ranking concepts, called *Consensus Ranking* and *Horizon Ranking*, to accumulate the analysis for different Epi-features , and provide a comparative evaluation for the methods' performance over time.

1.4 Organization of the Dissertation

The remainder of the dissertation is organized as follows. Chapter 2 provides a background and literature review on the past and existing methods and efforts in both fields of Bayesian epidemic forecasting, and performance evaluation of forecasts.

The details of the proposed method for modeling the infectious disease dynamics of influenza and Ebola are presented in Chapter 3. A high-resolution agent-based model is combined with a smart particle filter framework as the data-assimilation model. The particle filter role is to optimize the parameters of the agent-based model given the observed data. The experimental results demonstrate that the proposed smart particle filter approach is able to calibrate the parameters of disease model successfully. However, having the agent-based-model embedded in the framework still imposes a huge overhead and time-complexity to the procedure, which makes it computationally expensive for real time systems. In Chapter 4, we perform a complexity analysis on the proposed SBPF methodology and develop new strategies to optimize the running time of the system without compromising performance.

In Chapter 5, we introduce the epidemic evaluation framework to evaluate the performance of forecasting algorithms. This chapter presents epidemiologically relevant features (Epi-features), error measures, and Ranking Schema as the main parts of the evaluation framework. It demonstrates the potentials and capabilities of the framework by applying it to the output of a collection of six different methods used to predict influenza in the United States. The methods employed in this chapter generate deterministic results without handling the uncertainty in the solution.

The measures discussed in Chapter 5 deal primarily with deterministic forecasts. However, there are quite a few stochastic forecasting algorithms with some levels of uncertainty that

require a modified evaluation approach. Moreover, the observed data may be stochastic due to errors in measurements and source of information. In Chapter 6, we extend the evaluation framework and provide new methods to handle the stochastic forecasts and observations.

In Chapter 7, we present a software package named Epi-Evaluator that includes three primary packages named Epi-Feature, Epi-Error and Epi-Ranking which correspond to the three modules of the proposed evaluation framework. This package is going to be released as an open source software to be accessible to epidemiologist and data scientist who are interested in epidemic forecasting.

Chapter 2

Literature Review

2.1 Background on Epidemic Forecasting Methods

Over the past few decades, numerous forecasting methods have been proposed to aid in selection and implementation of interventions to reduce morbidity and mortality of infectious diseases. Such methods can be classified based on different criteria such as deterministic versus probabilistic, mechanistic versus statistical (generative), and so on.

Statistical methods do not aim at modeling the epidemiological process. Rather, they estimate the desired parameters or variables through statistical approaches. Some of the most prominent efforts among proposed statistical approaches in the field of epidemic forecasting are:

- Generalized linear models (GML) [14, 15, 16, 17, 18],
- Autoregressive methods such as AR, ARIMA, SARIMA, GARMA, INAR, etc. [19, 16, 20] for generating the time-series
- Bayesian networks [21, 22]
- Survival analysis [23]
- Prediction market [24],
- Gaussian random process [25]
- Other statistical analyses to model the time-series [26, 27, 28].

Mechanistic approaches employ epidemiological models to analyze and estimate the desired epidemiological variables. This dissertation and the methods discussed herein focus on these efforts. Mechanistic methods are classified into three main categories:

(i) Compartmental methods model the transition of disease across the subpopulations via abstracting the population into compartments. Each compartment represents the health status of its subpopulation. The most promising compartmental models are SIR and SEIR [9, 10, 11].

(ii) Agent-based models (ABMs) simulate the agents' behaviors, their interactions, and disease propagation among them in a sub-population [29, 30, 31, 32, 33].

(iii) Metapopulation models in which sub-populations are represented by separate patches and interact with each other through migration [34].

In some of the more popular mechanistic methods, researchers compare observed epidemiological data from early stages of an outbreak with the output of proposed models to forecast the future trend and prevalence of the pandemic. Usually, mechanistic models are coupled with other optimization and data-assimilation methods to estimate the model parameters given the surveillance data. Nsoesie et al. [35, 36] developed various simulation-optimization algorithms, including Nelder-Mead and Robbins-Monro algorithm. These methods estimate the key epidemiological parameters of the ABM to simulate the spread of influenza across the population. Besides the optimization trials, Nsoesie et al. applied eight supervised classification approaches [37] and one semi-supervised classification method [38] to train the epidemic model in the predefined group of simulated epidemic curves that are stored in a digital library.

Data assimilation or filtering techniques incorporate the available surveillance data into the simulated model of disease propagation and adjust the parameters to forecast the trend of epidemic [39, 40, 41]. A variety of filtering techniques has been utilized in the context of epidemic forecasting, among which are ensemble Kalman filter (EnKF) [40], ensemble adjustment Kalman filter (EAKF) [39], and various forms of particle filter (PF). Among the proposed methods, particle filters stand out due to their promising capabilities in dealing with non-linear and multi-modal systems [42]. Primary examples of particle filter methods coupled with mechanistic epidemic models are bootstrap PF [43, 44], auxiliary PF [45], kernel density PF [41, 46, 47], particle Markov chain Monte Carlo (pMCMC) [48], and maximum likelihood estimation via iterated filtering (MIF) [42]. These methods use compartmental techniques like SEIR [41, 46, 49, 50, 51], SIR [43, 48, 52], and SIRS [42] as the disease dynamics model, which provide a numerical solution to the differential equations subject to the assumption that each individual is in contact with all others. Some models alleviate this generalization through a contact factor that determines the percentage or fix number of contacts between susceptible and infected individuals [44, 45, 47, 53]. Jegat et al.[53] proposed a gravitational model to calculate the amount of flux within and between the regions. They used it as the contact rate in the SIR model to simulate influenza propagation. Nevertheless, none of these contact rates can be a true representative of the real-world interactions between individuals and the disease propagation.

To the best of our knowledge at the time of writing this dissertation, no previous effort has been made towards combining network agent-based models with particle filter framework as the data assimilation method. Osgood and Liu [44], Dawson et al. [51], and Skvortsov et al. [43] did incorporate an agent-based model with small population size (1,000-5,000) to only generate synthetic data to test their particle filter calibrator, while they embedded SIR/SEIR models as the disease propagation model within their calibration frameworks.

In the first part of this dissertation, we model the dynamics of diseases like flu and Ebola and present an agent-based-Bayesian algorithm to forecast the epidemic curve. We further estimate the underlying disease model parameters. Our approach uses the concept of Bayesian statistics to infer the model parameters given the observed data to handle the uncertainty in the reasoning. Moreover, the proposed model utilizes machine learning concepts to restrain the intractable search space.

2.2 Background on Forecast Evaluation

Developing epidemic forecasting methods has picked up pace over the past few decades. The list of proposed methods keeps growing. However, the lack of standard, well-defined evaluation measures to compare the performance of these methods, as well as selecting the best possible configuration for a particular algorithm, is still a significant problem. Armstrong [54], provided a thorough summary of the key principles that must be considered while evaluating time-series forecasting methods. The work presented in this dissertation is built upon his philosophy of objective evaluation, with a particular focus on the domain of epidemiology. To the best of our knowledge at the time of writing this dissertation, there have been no formal studies on comparing the standard epidemiologically relevant features across appropriate error measures for evaluating and comparing epidemic forecasting algorithms.

Nsoesie et al. [55] reviewed different studies in the field of forecasting influenza outbreaks and presented the features used to evaluate the performance of proposed methods. Eleven of the sixteen forecasting methods studied by the authors predicted daily/weekly case counts [55]. Some of the studies used various distance functions or errors as a measure of closeness between the predicted and observed time-series. For example, Viboud et al. [28], Aguirre and Gonzalez [56], and Jiang et al. [21] used correlation coefficients to calculate the accuracy of daily or weekly forecasts of influenza case counts. Other studies evaluated the precision and "closeness" of predicted activities to observed values using a different statistical measure of errors such as root-mean-square-error (RMSE), percentage error [21, 57], etc. However, defining a suitable distance function which demonstrates closeness between the surveillance and predicted epidemic curves is still a challenge. Moreover, the distance function provides a general comparison between the two time-series and ignores their epidemiological relevance between them, i.e. specific features of the epidemic curves that are more significant and meaningful from the epidemiologist perspective; these features could be better criteria to compare epidemic curves together rather than simple distance error. Cha [58] provided a

survey on different distance/similarity functions for calculating the closeness between two time-series or discrete probability density functions. Some other studies have analyzed the overlap or difference between the predicted and observed weekly activities by graphical inspection [59].

The epidemic peak is one of the most relevant quantities of interest in an outbreak, and its magnitude and timing are important from the perspective of health service providers. Consequently, accurate prediction of the peak has been the goal of some forecasting studies [17, 34, 56, 59, 60, 61, 62, 63, 64, 65]. Hall et al. [61], Aguirre and Gonzalez [56], and Hyder et al. [65] predicted the pandemic duration and computed the error between the predicted and real value. A few studies also considered the attack rate for the epidemic season as the feature of interest for their method [34, 57].

In the second part of this dissertation, we present an evaluation framework which allows for combining different features, error measures, and ranking schema to evaluate forecasts. We describe various epidemic features (Epi-features) included to characterize the output of forecasting methods and provide suitable error measures that could be used to evaluate the accuracy of the methods with respect to these Epi-features. We present various ranking schema that each summarizes individual rankings, and thus accounts for different error measures.

Part I

Epidemic Forecasting Algorithms

Chapter 3

Smart Beam Particle Filter Framework for Epidemic Forecasting

3.1 Introduction

Over the past decades, numerous computational and mathematical methodologies have been proposed for forecasting seasonal epidemics. In recent years, the deadly effects of enormous pandemics such as the H1N1 influenza virus, Ebola, and Zika have compelled scientists to find new ways to improve the reliability and accuracy of epidemic forecasts. The output of the forecasting models is of prime importance to epidemiologists and health-care providers and the choice of reliable model to emulate the disease's behavior remains a challenge. There have been multiple efforts to predict epidemics of diseases such as influenza and Ebola by modeling the disease-transmission dynamics and calibrating the model recursively using real-time observation data. Most recent efforts relied on mathematical compartment models like SEIR and SIR [9, 10, 11] as the disease dynamics simulator. Some studies improved model fidelity by considering the individual-level interactions across social networks [29, 30, 31, 32, 33]. However, these individual-based models impose additional complexity to the system by increasing the number of unknown parameters that should be optimized through data assimilation. Moreover, the choice of optimization method can drastically affect the accuracy of the epidemic model and thus forecast performance. Data assimilation, or filtering methods, use observation data recursively to train the model by estimating and updating the unknown model parameters, so that the model's output matches the characteristics of the evolved outbreak. As mentioned in Chapter 2, a variety of filtering techniques have been utilized in the context of epidemic forecasting, which includes ensemble Kalman filter (EnKF) [40], ensemble adjustment Kalman filter (EAKF) [39], and various forms of particle filters (PF). Among these methods, particle filters stand out because of their promising capabilities in dealing with non-linear and multi-modal systems [42].

In this chapter, we model the infectious disease dynamics of influenza and Ebola using a high-resolution agent-based framework that simulates the propagation of the disease over a social contact network of several million agents. The contact network is constructed based on demographic information and census data that reflects the detailed representation of individuals, subpopulations, and their interactions together. The epidemic simulator, named Epifast [29], is capable of implementing policy-based interventions such as school closures, vaccinations, and antiviral distributions as well as individual based interventions like travel or interaction reductions. This model provides a realistic representation of human network structure and simulates real-world transmission dynamics at the detailed individual level. However, increasing the complexity of the model requires calibrating more unknown parameters. We propose a smart particle filter framework to optimize the parameters of the agent-based model given the observed data.

A variety of recent works have used the particle filter as the data assimilator in the context of outbreak forecasting. All of these efforts, however, depended on mathematical compartment models like SEIR [41, 46, 49, 50, 51], SIR [43, 48, 52], and SIRS [42] as the disease dynamic model, which provide a numerical solution to the differential equations subject to the assumption that each individual is in contact with all others. Some models alleviate this generalization via a contact factor that determines a percentage, or fixed number of contacts between susceptible and infected individuals [44, 45, 47, 53]. Nevertheless, none of these contact rates are a true representation of real-world interactions between individuals and the disease propagation.

To the best of our knowledge at the time of writing this dissertation, no prior effort has been made towards combining a network agent-based model with a particle filter framework as the data assimilation method. Osgood and Liu [44], Dawson et al. [51], and Skvortsov et al. [43] did incorporate an agent-based model with small population size (1,000-5,000) to only generate synthetic data to test their particle filter calibrator, while they embedded SIR/SEIR models as the disease propagation model within their calibration frameworks.

Simulating disease propagation by real network agent-based models involves significant costs and imposes complexity to the calibration framework that restricts the number of particles as well as the number of running iterations of the particle filter. Most particle filters used in the context of epidemic forecasting generate 5,000 to one million particles to establish converged best results. Running agent-based methods with large number of particles is not practical and tractable. On the other hand, individual-based models with different intervention strategies require multiple parameters that should be calibrated properly. A particle filter could collapse in large-scale systems due to the curse-of-dimensionality problem [66]. To solve this problem, we propose a smart dynamic diffuser, which defines the state dynamics of the particle filter. The smart diffuser examines the predicted and observed data based on machine learning methods and determines the direction and the perturbation for each parameter of the state vectors. The smart diffuser enables the particle filter to find more accurate results with a lower number of particles and fewer running iterations.

The rest of the chapter is organized as follows. Preliminaries and notations, that are used in the rest of this chapter, are given in Section 3.2. In Section 3.3, we provide an overview of the agent-based model and its transition Markov chain model. In Section 3.4, we present how we model the disease dynamics of the flu and Ebola with the applied agent-based model and provide the detailed parameters that control the disease properties and required interventions. An overview of Bayesian recursive filter and particle filter are provided in Sections 3.5 and 3.6 respectively. In Section 3.7, we present our proposed smart Beam-Particle Filter framework as a data assimilation approach to improve the epidemic forecast. Experimental results showing the utility of the proposed evaluation framework are presented in Section 3.8. Section 3.9 provides the list of findings and achievements from our experimental results. Finally, we conclude the chapter in Section 3.10.

3.2 Preliminaries and Notations

In this chapter, we briefly introduce the agent-based model that we used to model the disease propagation, and then we present the proposed particle filter framework. In the rest of this chapter, we use the following notations listed in Table 3.1.

Symbol	Definition
hs_i^t	the health status of the agent i at time t
N_p	population size for the individual-based model
v_i or u_i	i^{th} vertex of the network, corresponding to an individual in agent-based model
x_k $x_k = \{\theta_k^1, \theta_k^2, .., \theta_k^{n_\theta}\}$	a particle at time step k: a state vector of parameters θ_k^i is the i^{th} parameter.
$x_k^* = \{x_k^{i*}\}_{i=1}^N$	resampled particles at time step k
y_k	single observation at time step k
$Y_k = \{y_i, i = 1, ..., k\}$	all observations received up to time k including y_k
$p(x)$ or pdf	probability density function (of random variable x)
$\tilde{w}_k^i$	unnormalized weight of i^{th} particle at time step k
$w_k^i = \tilde{w}_k^i / \sum_{j=1}^N \tilde{w}_k^j$	normalized weight of i^{th} particle at time step k
ϖ	the width of the window in calculating likelihood
V_k	system dynamics error.
W_k	observation (measurement) error.
$\{I_i\}_{i=1}^T$	the predicted case counts from time step 1 to time step T.
$\delta(x)$	Dirac delta function
N	number of particles

Table 3.1. Notation and Symbols

3.3 Networked Epidemiology

Networked epidemiology studies epidemic processes over networks; over the last decade these models have become popular owing to their ability to incorporate spatial and individual level heterogeneity and complex interventions. See [30, 31, 67, 68] for additional discussion. We briefly describe the terminology here. Let $G(V, E)$ denote a social contact graph on a population V – each edge $e = (u, v) \in E$ denotes that the individuals (also referred to as nodes) $u, v \in V$ come into contact. Let $N(v)$ denote the set of neighbors of v. We will let $|V| = n$ and $|E| = m$.

For the SEIR model on the graph G, we have a dynamical process with each node being in S, E, I or R states. Infection can potentially spread from u to v along edge $e = (u, v)$ with a probability of $\beta(e, t)$ at time instant t after u becomes infected, conditional on node v remaining uninfected until time t.

Let $\tau(u)$ denote the time that node u would remain in the infected state, and let $\tau = \max\{\tau(u) : u \in V\}$. If a node $u \in V$ gets infected at time t_u, it attempts to infect each susceptible neighbor v with probability $\beta((u, v), t - t_u)$ for $t = t_u + 1, \ldots, t_u + 1/\lambda(u)$. After $1/\lambda(u)$ steps (called the period of infection), node u switches to state R. If the susceptible node v contracts the infection it moves to Exposed state E and stays in this state for $1/\gamma(v)$ (called the incubation period) steps and then moves to state I. A node in state E cannot infect other nodes.

We let $I(t)$ denote the set of nodes that become infected at time t. The sequence $I(t)$, along with the (random) subset of edges on which the infections spread, represent a disease outcome, also referred to as a *dendogram*. The time series $(|I(t)|, t = 0, 1, \ldots)$ is referred to as an *epidemic curve* corresponding to a stochastic outcome. The total number of infections for an outcome is given by $\sum_t |I(t)|$. The epidemic peak is the largest time t that maximizes $|I(t)|$. Thus, this dynamical system starts with a configuration in which there are one or more nodes in state I and reaches a fixed point in which all nodes are in states S or R.

3.3.1 Agent-Based Network Model

We use a high performance computing based modeling environment called *Epifast* for simulating epidemic and associated interventions over social contact networks [29]. Epifast simulates the spatiotemporal propagation of the disease through social interactions between individuals [29]. Epifast provides the capability of simulating a broad range of policy-based, as well as individual-based, interventions. Interventions could be pharmaceutical (PI) or non-pharmaceutical (NPI). PIs include the dispensing of antivirals, vaccines, and antibiotics, whereas NPIs refer to any change in individual interactions or social network structure aside from using medicine. More information about mathematical modeling can be found in [33].

The social contact networks we use are based on the method described in [67]; we refer the reader to [67, 68, 69] for further details on how such networks are constructed, modified, calibrated and validated. The process has been extensively reviewed and results in high quality realistic synthetic social contact networks– note that there is no way to build such networks by endless measurements.

3.3.2 Markov Chain Components of Disease Transition model

As mentioned earlier, the agent-based model used in this project categorizes each individual of the population at every snapshot of the simulation in one of the four health states: susceptible, exposed, infectious, and removed. The transmission of disease depends on the demographic information of each individual (e.g their age, health status, susceptibility, etc.) as well as the contact between two individuals in the susceptible and infectious states. The agents could have different activities (such as work, shop, stay at home, travel, etc.) which puts them in contact to each other. In the constructed contact network, the vertices (v_is) represent individuals (agents) in the society and the edges determine the mutual contacts. The disease transition model could be summarized in a discrete Markov Chain model. The configuration of random variables of the corresponding Markov chain would be an N_p-vector $HS^t = \left(hs_1^t, hs_2^t, ..., hs_{N_p}^t\right)$, where $hs_i^t \in \{S, E, I, R\}$ is the health status of the agent i at time t, and N_p is the population size. Epifast starts simulation with $hs_i^0 = S$ for all agents, except those who are selected as the initial infected seeds, $hs_j^0 = I$. The system goes through a sequence of configurations over time, and stops at a true fixed point or when the timer reaches its threshold. True fixed point is a configuration state HS^t that its only successor is itself, e.g. $HS^{t+1} = HS^t$. HS^{t+1} is the successor of HS^t, if the probability of transition from HS^t to HS^{t+1} is not zero: $p\left(HS^{t+1}|HS^t\right) = \prod_{i=1}^{N_p} p\left(hs_i^{t+1}|hs_i^t\right)$. A single transition probability for a vertex v_i is as follows:

$$p(hs_i^{t+1} = S|hs_i^t = E) = \begin{cases} 0, & \text{if } IN(v_i) = \emptyset \\ 1 - \prod_{u \in IN(v_i)}(1 - p(u, v_i)), & \text{otherwise} \end{cases} \tag{3.1}$$

where:

$$p\left(u, v_i\right) = f\left(u, v\right) = 1 - \left(1 - r_u^{out}.r_v^{in}\right)^{w(u,v)} \tag{3.2}$$

where $IN\left(v_i\right)$ are infectious neighbors of v_i, $r_v^{in} \in \mathbb{R}$ is the vulnerability of the vertex v, $r_u^{out} \in \mathbb{R}$ is the infectivity of the vertex u, and $w\left(u, v\right) \in \mathbb{N}$ is the contact duration between the two agents. The other transition probabilities of agent-based-system Markov chain are listed in Table 3.2.

Transition probability
$p(hs_i^{t+1} = S \mid hs_i^t = I) = 0$
$p(hs_i^{t+1} = S \mid hs_i^t = R) = \begin{cases} 1, & \text{if } f^V = PI \text{ vaccination with } efficacy\ r_{vi}^{in} = 0 \text{ occurs} \\ 0, & \text{otherwise} \end{cases}$
$p(hs_i^{t+1} = S \mid hs_i^t = S) = 1 - p(hs_i^{t+1} = S \mid hs_i^t = E) - p(hs_i^{t+1} = S \mid hs_i^t = R)$
$p(hs_i^{t+1} = E \mid hs_i^t = I) = \begin{cases} 1, & \text{if } vi\left(\Delta t^E\right) \text{ gets zero at time } t+1 \\ 0, & \text{otherwise} \end{cases}$
$p(hs_i^{t+1} = E \mid hs_i^t = R) = 0$
$p(hs_i^{t+1} = E \mid hs_i^t = S) = 0$
$p(hs_i^{t+1} = E \mid hs_i^t = E) = 1 - p(hs_i^{t+1} = E \mid hs_i^t = I)$
$p(hs_i^{t+1} = I \mid hs_i^t = R) = \begin{cases} 1, & \text{if } vi\left(\Delta t^I\right) \text{ gets zero at time } t+1 \\ 0, & \text{otherwise} \end{cases}$
$p(hs_i^{t+1} = I \mid hs_i^t = S) = p(hs_i^{t+1} = I \mid hs_i^t = E) = 0$
$p(hs_i^{t+1} = I \mid hs_i^t = I) = 1 - p(hs_i^{t+1} = I \mid hs_i^t = R)$
$p(hs_i^{t+1} = R \mid hs_i^t = R) = 1$

Table 3.2. Transition probability between Markov chain states of the agent-based model.

3.3.3 Agent-Based Model Output

In addition to the high degree of fidelity of the agent-based model and its consistency with the real-world disease propagation, Epifast is able to provide more detailed outputs as compared with the compartment SEIR model. The possible outputs of Epifast are as follows:

- The daily health status of each individual
- Dendogram that provides detailed temporal description of who infects whom and at what time;
- Transmission Network (TN) is a directed acyclic graph where people at level i were infected on the i^{th} day.
- Health status and statistics for different subpopulations: Having complete demographic data for each individual makes it possible to obtain different epidemic statistics for subpopulations in various categories based on age, gender, regions, etc.

3.4 Simulating Flu & Ebola Propagation by Epifast

Epifast provides extensive freedom and capabilities for modeling disease dynamics and interventions. Those capabilities are controlled by different parameters that are mostly unknown. When Epifast uses the SEIR model to simulate either Ebola or flu dynamics, the minimum-required parameters are those that control SEIR dynamics, including transmission rate (β), mean incubation period ($1/\gamma$), and mean infectious period ($1/\lambda$). In our disease simulation, we have defined two discrete probability distribution functions for different values of incubation and infectious times, instead of their mean value. $P_\gamma = (P_{\gamma_1}, P_{\gamma_2}, ..., P_{\gamma_{n_\gamma}})$ denotes the probability distribution function for various incubation periods, where $\sum_{i=1}^{n_\gamma} P_{\gamma_i} = 1$, and P_{γ_i} represents the probability that the incubation period of a patient is equal to γ_i time units. The probability distribution function for different values of infectious duration is denoted by $P_\lambda = (P_{\lambda_1}, P_{\lambda_2}, ..., P_{\lambda_{n_\lambda}})$. Table A.3 and Figure A.1 in appendix A provide more details on the values and probability distributions of incubation and infectious periods for simulating Flu and Ebola diseases, respectively.

Depending on the available information on disease characteristics and type of applied interventions to control the epidemic, additional parameters could show up. Table 3.3 presents the parameters used to characterize the flu and Ebola models. Some of these parameters are estimated and fixed according to prior knowledge while others are calibrated through a data-driven particle filter framework. As mentioned, the Epifast simulator is able to implement the PI interventions such as dispensing antivirals and vaccines. These interventions are adjusted by parameters including *delay in actions*, input-efficacy (or *immunization efficacy*: the effectiveness of antiviral or vaccine on immunizing a person from getting infected), *output-efficacy* (the effectiveness of antiviral or vaccine on decreasing the transmissibility of disease from an infectious person to susceptible ones), total-resources (the available vaccine/antiviral resources), the affected *sub-population*, the *effective duration*, etc [29]. The interventions that are applied to prevent the outbreak of the Ebola disease are different from ordinary vaccination and antivirals. Based on the reports that are provided for the RAPIDD Ebola challenge, the list of applied interventions are: regular hospitalization, safe burial protocol, establishing Ebola treatment units (ETU), contact tracing, and distribution of home disinfection kits [70]. The detailed reports of these actions for various time-points of the season are included in appendix A. As Epifast does not provide such interventions, they are emulated via vaccination and antiviral reactions. The detailed descriptions of various interventions and required parameters to control and adjust them are as follows.

Natural Isolation. The first assumption in our simulation is that infected people reduce their non-household interactions as soon as the symptoms show up. Therefore, the transmissibility of outgoing non-household edges of the contact graph including *work*, *school*, *shopping*, and *others* interactions are decreased by the natural isolation parameter (β_{NI}).

			Calibrated or Fixed				
Parameter		**Flu**	**Ebola**				
			TP_1	TP_2	TP_3	TP_4	TP_5
β	Transmission rate	C	C	C	C	C	C
P_γ	Incubation period distribution	C	F	F	F	F	F
P_λ	Infectious period distribution	C	F	F	F	F	F
I_0	No. of initial infections	F	C	C	C	C	C
β_{NI}	Natural isolation efficacy	NA	F	F	F	F	F
ω_{TR}	Travel reduction ratio	NA	C	C	C	C	C
β_{HE}	Hospitalization efficacy	NA	C	C	C	C	C
β_{HD}	Hospitalization delay	NA	C	C	C	C	C
$beta_{ETU}$	Ebola treatment units' efficacy	NA	NA	F	C	C	C
$beta_{SB}$	Safe burial efficacy	NA	NA	NA	F	F	F
$beta_{CT}$	Contact tracing efficacy	NA	NA	NA	F	F	F
$beta_{BI}$	Behavioral intervention efficacy	NA	F	F	F	F	F
C_{BI}	Behavioral intervention compliance	NA	F	F	F	F	F

Table 3.3. Parameters to adjust and control the epidemic model and interventions. (C) refers to the calibrated parameters and (F) represents the parameters which are initially estimated and then fixed during the the epidemic simulations. TP_1,...,TP_5 denote the time-points 1 through 5 of the Ebola challenges.

Travel Reduction. Same as natural isolation, travel reduction parameter is reducing the interaction of infectious people with the outside world. This intervention changes the long distance traveling of agents during their infectious periods. This parameter denoted by ω_{TR} is calibrated by searching over the range of [0.2,1.0].

Hospitalization. The effect of *hospitalization* is emulated by an antiviral intervention in our simulator. We assume that some of the diagnosed patients (those who comply) are admitted to hospitals with some delay after showing the symptoms. This delay is controlled by *Hospitalization-delay* parameter (β_{HD}), and can vary from 2 to 10 days. As hospitalization and isolation of infectious people could affect the propagation of disease, the outgoing transmissibility of the targeted patients are down-scaled by a parameter named *hospitalization efficacy* (β_{HE}) that is calibrated over the range of [0.1,0.8]. Therefore, the transmissibility of such agents would be multiplied by β_{HE}. The *compliance ratio* determines the portion of infectious agents that comply the hospitalization, which is estimated and fixed during the simulation. *Hospitalization duration* is also estimated based on available datasets and

is fixed for all agents. Table A.4 in appendix A provides more details on the estimated parameters as well as the range of calibrated parameters.

Ebola Treatment Units. During the 2014 Ebola pandemic, various Ebola Treatment Units (ETUs) were established in different counties to provide patients with the best possible care only for Ebola. While these units could decrease the disease transmission to the rest of society, still ETU personnel and health workers could get exposed to the virus. Moreover, the resources and number of beds of these units were insufficient [70]. In our model, the effect of ETU units are emulated by vaccination-type intervention and vaccine resources correspond to the beds and facilities of ETUs. The *efficacy of ETUs* (β_{ETU}) is fixed or calibrated depending on the information about the future or past ETUs provided in various time-points. Table 3.3 presents the details of intervention configurations used for each time-point of the Ebola challenge.

Safe Burial. Similar to hospitalization, safe burial is implemented by antiviral intervention to reduce the transmissibility in the society. The mortality rate is calculated based on the available datasets. The *safe burial efficacy* (β_{SB}) is estimated and fixed for all time-points were the safe-burial protocol was reported for (Table 3.3).

Contact Tracing. Facing the epidemic of Ebola, the health providers implemented the contact tracing protocol to control EVD outbreak [71, 72]. In our network-based epidemic simulator, we trace the agents that have at least one EVD in their 1-distance neighborhood in the contact graph. The duration of this intervention is considered 21 days [73], and the outgoing transmissibility of those who comply is decreased by the *contact tracing efficacy* (β_{CT}).

Behavioral Intervention. In our model, we add a behavioral intervention which simulates the natural behavior reactions of people during the epidemic e.g. washing hands and foods more carefully and using napkins to block coughs and sneezes. As people reactions are different, the *compliance* parameter (C_{BI}), adjusts the human obedience and *efficacy* parameter (β_{BI}) indicates the effect of such actions. Not all of the interventions and protocols mentioned above are applied for every time-point and scenario. Depending on the past and future protocols provided in each time-point report, specific parameters are estimated and fixed while some others are calibrated through our proposed particle filter approach (Table 3.3).

3.5 State-Space Model & Bayesian Sequential Estimation Method

State-space model is a class of probabilistic models that describes the dependence between state variables and observed data. Sequential estimation filters receive new observations and update the estimated value of state variables that best describe the observed behavior. Bayesian recursive filter tries to construct the posterior probability density function (pdf) of

the state vector (x_k) based on all available information. Bayesian filter recursively predicts and update the pdf function of state vector $P(x_k|Y_k)$ given all observations, received up to time k and including $y_k : Y_k = \{y_i, i = 1, ..., k\}$.

The Bayesian filter contains two main steps: prediction and update. The prediction process estimates the posterior of the state vector at time k based on the old observation (measurement) values up to time $k-1$ ($p(x_k|Y_{k-1})$). In the update process, the new observation y_k is added to the previous information and the posterior pdf of the state vector ($p(x_k|Y_k)$) is corrected based on the new information (The proof of Update formula is provided in Appendix A).

Prediction:

$$p(x_k|Y_{k-1}) = \int p(x_k|x_{k-1}).p(x_{k-1}|Y_{k-1})dx_{k-1} \tag{3.3}$$

Update:

$$p(x_k|Y_k) = p(y_k|x_k) \times p(x_k|Y_{k-1})/p(y_k|Y_{k-1}) \tag{3.4}$$

where $p(y_k|Y_{k-1})$ is normalizing value:

$$p(y_k|Y_{k-1}) = \int p(y_k|x_k) \times p(x_k|Y_{k-1})dx_k$$

These equations are not usually in closed form and cannot be calculated analytically, except in particular cases such as Linear-Gaussian model, which results in Gaussian posterior. The Bayesian filter in this case in known as Kalman filter [74]. For other cases, numerical methods like extended-Kalman-filter [75], Gaussian sum filter [76], or Monte Carlo particle filter are the alternate approaches.

3.6 Particle Filter

Particle filter (PF), which is best known as bootstrap filtering [77] or condensation algorithm [78], is a sequential Monte Carlo (SMC) approach to estimate the posterior distribution defined in a discrete Bayesian filter. The main idea of this method is to represent the posterior density function of a state vector by a set of weighted random particles. Each particle is a sample of the state vector and is shown as x_i. The initial values of particles are set randomly by sampling from a prior distribution defined for the state vector. The PF has two fundamental parts: state dynamics, and observation equations. The state dynamics (system model) describes the change of the state vector over time and is assumed to be a function of the state vector in the preceding step:

$$X_k = f_{k-1}(X_{k-1}, V_{k-1})$$

X_k is the state vector to be estimated, f_{k-1} is the known possible non-linear function, and V_{k-1} is the known system error. This equation corresponds to transition probability density

function (pdf) $p(x_k|x_{k-1})$ in the probabilistic description of the state evolution in a Bayesian filter. The observation (measurement) equation shows the relation between the observed data and the state vector:

$$y_k = h_k(X_k, W_k)$$

where W_k is the observation error. This function corresponds to the likelihood function $p(y_k|x_k)$ in a Bayesian model. The initial condition of this model is defined as the prior pdf $p(x_0)$ for the state vector in which no measurement has been received.

Figure 3.1, describes common steps of a particle filter algorithm. PF uses N particles to estimate the posterior distribution of the state vector. The initial particles in step k are the samples from the posterior pdf $p(x_{k-1}|Y_{k-1})$ generated in the previous cycle, where Y_{k-1} is all observations received up to time $k-1$ and including $y_{k-1} : Y_{k-1} = \{y_i, i = 1, ..., k-1\}$. These particles are denoted by $\left\{x_{k-1}^{i*}\right\}_{i=1}^{N}$. As mentioned earlier, the first set of particles are generated randomly. Like the Bayesian filter,the PF has two phases: prediction and update. In the prediction phase, the samples from step $k-1$ are passed through the state dynamics to generate the new set of samples, which are prior particles in step k:

$$x_k^i = f_{k-1}(x_{k-1}^{i*}, v_{k-1}^i)$$

This new set of particles $\{x_k^i\}$ represents the samples of the prior pdf $p(x_k|Y_{k-1})$, that is the prediction of the new state vector based on the previous observed data.
In the update phase, these prior samples $\{x_k^i\}$ are updated based on the new measurement received at step k (Y_k). A weight $\tilde{w}_k^i$ is calculated for each particle based on these measured values. This weight defines the likelihood of the observed value based on the prior sample: $\tilde{w}_k^i = p(y_k|x_k^i)$. The weights are normalized and calculated with the following equation:$w_k^i = \tilde{w}_k^i / \sum_{j=1}^{N} \tilde{w}_k^j$. The prior particles are resampled according to the normalized weights to generate the new set of particles, denoted by $\{x_k^{i*}\}_{i=1}^{N}$. This set of particles are considered as the samples of posterior function $p(x_k^*|Y_k)$. Therefore, the aggregation of these weighted samples constructs the required pdf $p(x_k^*|Y_k)$:

$$p(x_k^*|Y_k) = \sum_{i=1}^{N} w_k^i \times \delta(x_k - x_k^i) = \sum_{i=1}^{N} (1/N) \times \delta(x_k - x_k^{i*}) \tag{3.5}$$

This completes one cycle of the algorithm. This procedure is repeated when a new observation is received (Figure 3.1).

3.7 Smart Beam Particle Filter Epidemic Calibrator Framework

Smart Beam Particle Filter (SBPF) is inspired from the Beam Search (BS) method [79], to explore the search space more actively, alleviate the impoverishment problem, and to

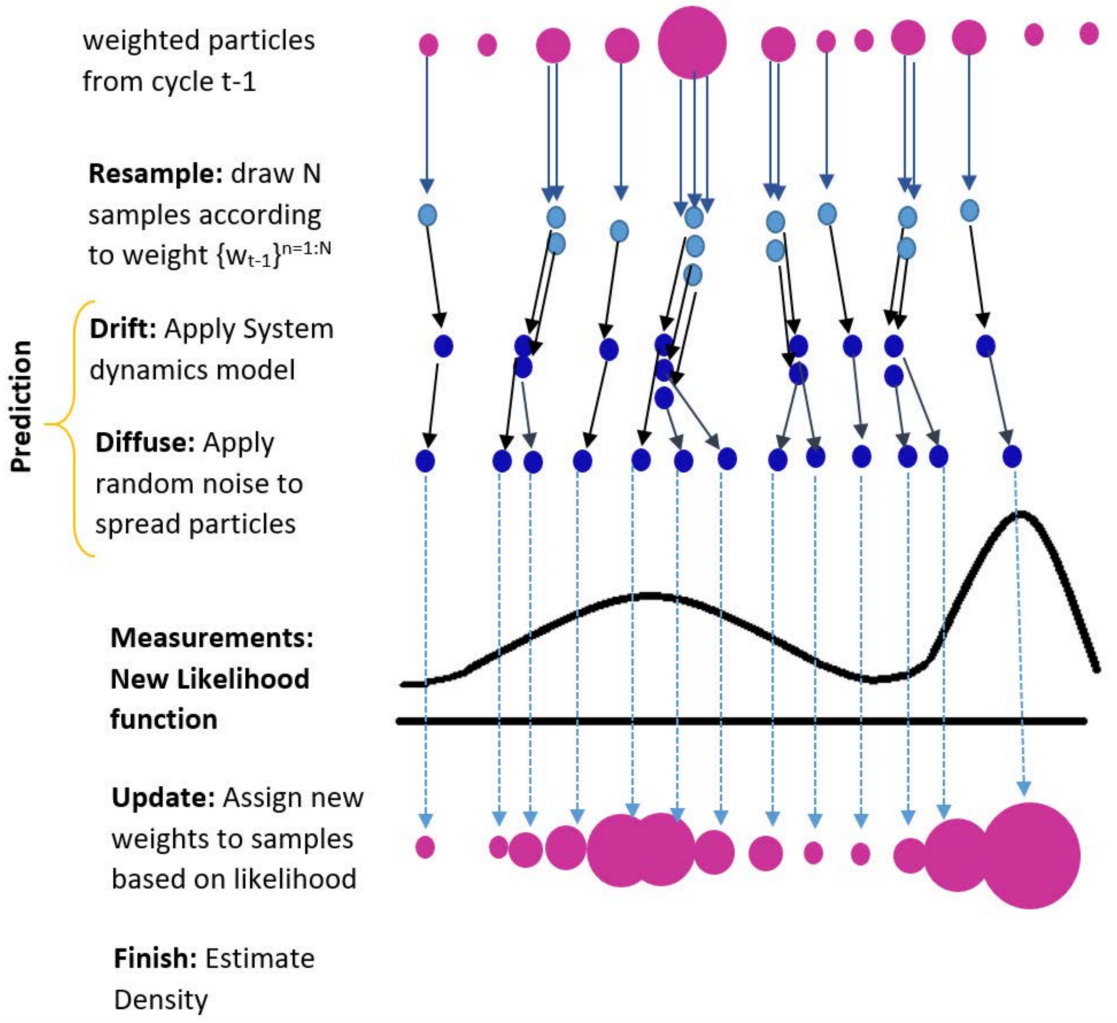


Figure 3.1. Particle filter procedure

escape from local optima traps. The SBPF epidemic calibrator solves a dynamic estimation problem in order to figure out the model's parameters that best describe the trend of the observed epidemic. As mentioned previously, a particle filter uses a set of particles to estimate the posterior distribution of the state vector. The state vector for our model contains the unknown parameters that control the dynamics of the epidemic model (Table 3.3).

The mechanism of our SBPF epidemic calibrator is shown in Figure 3.2. According to the PF process, the first phase of the particle filter is the prediction process in which the sampled particles are received from the previous cycle, and the new states of the particles are predicted based on their previous values and past observations ($p(x_k|Y_{k-1})$). The SBPF framework expands and diffuses each old sample, x_{k-1}, to multiple new state vectors, (x_k), in different directions. This process increases the number of particles from N_1 particles to N_3, where $N_3 >> N_1$. This results in more diverse solutions in each round. The size and direction of perturbation are determined based on predictive decisions made in the smart

state dynamics component. Smart state dynamics is discussed in detail in the section 3.7.2.

At the first cycle, no old sampled particles and observations are available to make any prediction. Therefore, the first state vectors $\{x_0^{i*}\}_{i=1}^{N_2}$ are initialized by sampling from a prior distribution, where $N_2 >> N_1$. Initial particles are passed unchanged to the Epifast as the first prediction for particles. Epifast receives the particles and uses them as base parameters to run the simulation of the agent-based model and generates various epidemic outputs as well as the corresponding epidemic curves.

The next step of the particle filter is calculating the likelihood of observations given the state vectors. Epidemic curves that are generated by epidemic simulator are compared to observed curve to calculate the likelihood of the observed data given the model's parameters. The details of the likelihood function are provided in the section 3.7.1.

In the update procedure, the likelihood score can be used as the weights of the particles after normalization. The SBPF performs the non-replacement Monte Carlo sampling to resample the particles based on their weights and selects N_1 particles among N_2/N_3 ones that are the most promising state-vectors to generate the better forecast. The resampled particles are considered as the samples of posterior function $p(x_k^*|Y_k)$. Therefore, the aggregation of these samples could construct the required pdf $p(x_k^*|Y_k)$. At this stage, one cycle has been completed, and the result of this cycle are fed to next cycle.

Usually, the PF cycle is repeated whenever a new measurement (observation) is received. However, the SBPF repeats the cycle more than once for the same observed data to converge to a better solution (Algorithm 1).

3.7.1 SBPF Components: Likelihood Function

The likelihood function represents the possibility of observed epidemic data given the hidden parameters of the epidemic simulator. Although Epifast generates various outputs, the available surveillance observed data are usually limited to the time series of the weekly number of new infected cases. We use similarity/dissimilarity functions as the criteria of closeness between the observed and simulated curves for assessing the likelihood (Table 3.4). The crucial choice of dissimilarity function could change the functionality of the resampling process. We tested various dissimilarity functions such as wwL1, wwL2, and wMAPE. wwL1 or window-weighted-L1 is the L1-distance function in which recent data points of the time-series are valued more than older ones. As people and public health providers change their behavior and policies after facing the epidemic, some epidemic parameters will change across the time. Therefore, calibrating parameters based on old observations will diverge from the actual values. As a result, we calculate the likelihood based on the latest ϖ recent data points of the time-series and discard the older ones. Table 3.4 lists the equation of the aforementioned distance functions, in which $\{Y_i\}_{i=1}^{T}$ denotes the target time series of newly infected case counts until week T, $\{I_i\}_{i=1}^{T}$ represents the predicted case counts, and ϖ denotes

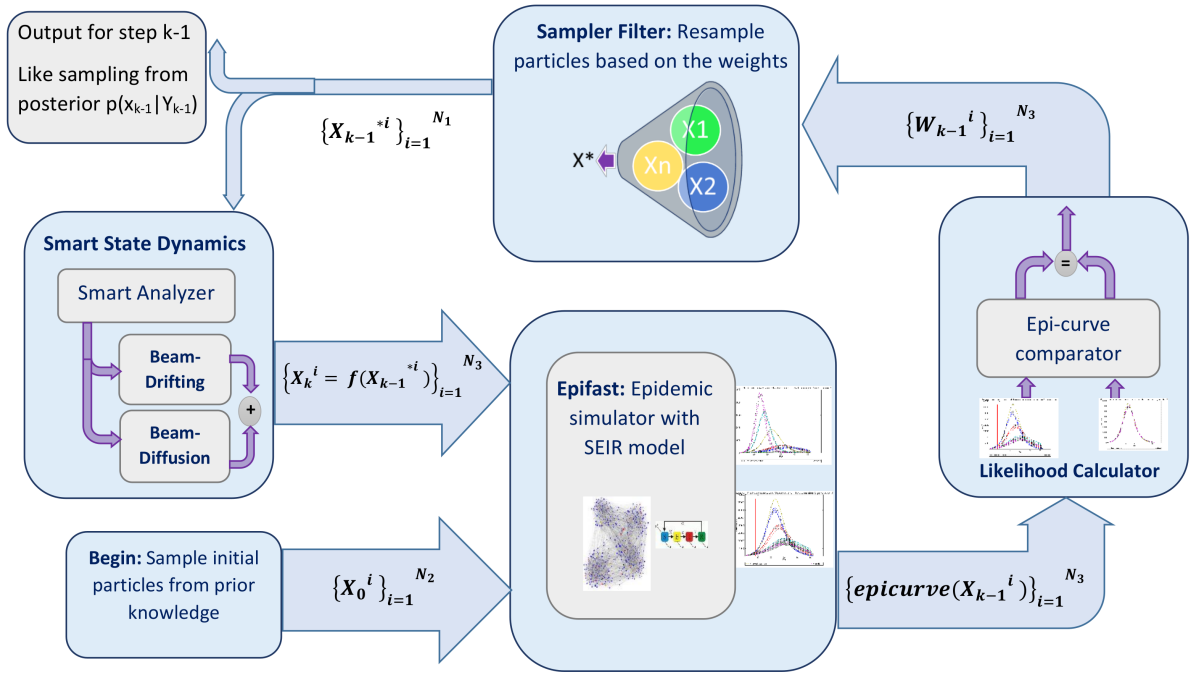


Figure 3.2. Smart Beam-Particle Filter: epidemic calibrator framework. The width of arrows indicates the number of particles transferred between different units.

the width of the window.

Our experiments demonstrated better results using the wMAPE compared with others. For comparison purposes, the similarity score is defined as the inverse of distance score. The scores are normalized over all particles and reported as the likelihood scores of the observed data given the state vector of parameters.

3.7.2 SBPF Component: Smart State Dynamics

State dynamics unit models the changes in the state vector and predicts the value of the state vector in the next cycle based on the current value ($p(x_k|x_{k-1})$). We have designed a smart state dynamics utility that examines the predicted epidemic curves and the observed ones to determine the perturbation's size and direction for each parameter of the state vector. This utility has the following components:

- A *feature selector*
- A *classifier*
- A *smart director*
- An *adaptive tuner*

ALGORITHM 1: Smart Beam Particle Filter Algorithm.

Input: Probability Distribution function $p(x_0)$: The prior knowledge about unknown parameters of the state vector $x = (\theta_1, \theta_2, ..., \theta_{n_x})$;
All observations received up to time $k-1$ including $y_{k-1} : Y_{k-1} = \{y_i, i = 1, ..., k-1\}$;
N_1: Number of resampled particles;
N_2: Number of initial particles;
N_3: Number of expanded particles;
Output: $p(x_T^*|Y_T)$: The posterior function ;
S = {Sample N_2 particles from prior pdf $p(x_0)$};
for $k = 0$ *to* T **do**
  if *k=0* **then** `/* First Round: Does not receive any observation yet. */`
    $S_0^* = \{x_0^{i*}\}_{i=1}^{N_2}$ = {Sample N_2 particles from prior $p(x_0)$}
  else
    $S_{k-1}^* = \{x_{k-1}^{i*}\}_{i=1}^{N_1}$ = {resample N_1 particles from posterior pdf $p(x_{k-1}|Y_{k-1}$) from previous step}
  end
  `/* Prediction Phase: Pass samples through the smart State Dynamic unit. */`
  `/* Purterb and expand previous particles to multiple ones: */`
  $x_k^i = f_{k-1}(x_{k-1}^{i*}, v_{k-1}^i)$;
  $S = \{x_{k-1}^i\}_{i=1}^{N_3}$ = samples from the prior pdf $p(x_k|Y_{k-1})$;
  `/* Update Phase: Pass samples through the State Dynamic unit. */`
  Give weight to particles based on $\widetilde{w}_k^i = p(y_k|x_k^i)$;
  Normalized weights: $w_k^i = \widetilde{w}_k^i / \sum_{j=1}^{N_3} \widetilde{w}_k^j$;
  Resample particles based on normalized weights as $S_k^* = \{x_k^{i*}\}_{i=1}^{N_1}$;
  Posterior is: $p(x_k^*|Y_k) = \sum_{i=1}^{N_1} w_k^i \times \delta(x_k - x_k^i)$;
end

Distance Function Equation	$v_i = \frac{f(x=i)}{f(x=T)}$
$wwL1 = 1/\varpi \sum_{i=T-\varpi}^{T} v_i \times \lvert y_i - I_i \rvert$	
$wwL2 = 1/\varpi \sum_{i=T-\varpi}^{T} v_i \times \lvert y_i - I_i \rvert^2$	where
$wMAPE = 1/\varpi \sum_{i=T-\varpi}^{T} \lvert y_i - I_i \rvert / y_i$	$f \sim N(T, \varpi/3)$

Table 3.4. Dissimilarity functions considered for assessing the likelihood. wMAPE demonstrated better results and was selected among others.

- A *diffuser*

Figure 3.3 demonstrates the connection between all these components. The components of the *smart state dynamics framework* are described as follows.

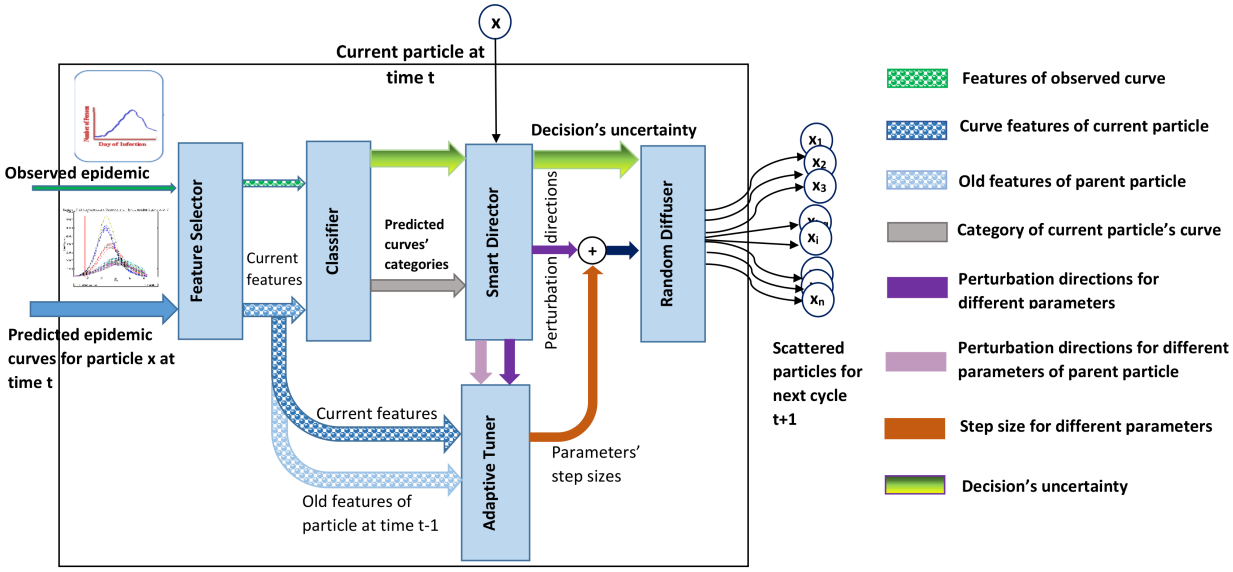


Figure 3.3. Smart state dynamics framework. Inputs are: the current particle, its corresponding epidemic curves simulated by Epifast, history of different features belonging to the particle's parent, and the observed epidemic curve. The feature selector chooses different features from the epidemic curves. Base on the selected features of both predicted and observed epidemic curves, the classifier determines the category to which the state vector (particle) belongs. The smart director makes the directional adjustment for each parameter. The adaptive tuner adjusts the step size for each parameter.

Components of Smart State Dynamics: Feature Selector

The *feature selector* receives both the predicted and observed epidemic curves and extracts different features from them. The most important features of the feature selector are:

- Peak Value (denoted by PV):
  The maximum value of the epidemic curve. For the observed epidemic curve, this feature refers to the maximum value of the observed curve till the current prediction week.

- Peak Time (PT):
  The week that peak happens.

- Epidemic Length (L):
  The length of epidemic curve

- Weighted Error:
  This is the error between the observed and predicted epidemic curve till the current prediction week. In this case we have used ww-L1 error which is defined in Table 3.4 .

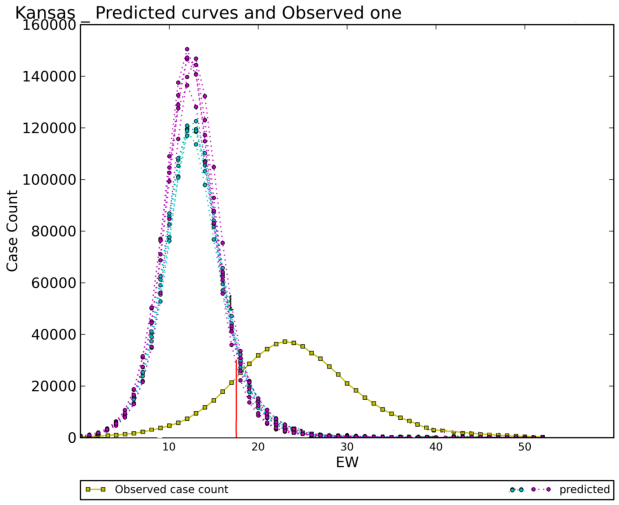


Figure 3.4. Early-Overestimated prediction. The red line shows the prediction week. The blue curves replicate the curves generated by running epidemic simulator with one set of parameters. The predicted epidemic curves corresponding to two different state vectors are shown in purple and blue.

For more information regarding the above features refer to Chapter 5.

Components of Smart State Dynamics: Classifier

The rule-based *classifier* receives the selected features of both the predicted and observed epidemic curves and determines the state vector's category as summarized in Table 3.5. The tree-structure and rules of the classifier are constructed based on the relative value of the predicted and observed features. Algorithm 2 presents the details of the tree-structure of the classifier. The classifier also determines the level of certainty (CL) about its decision. Figures A.2 and A.3 in appendix A, show the dataset of the sample curves from different categories that satisfy various conditions of the tree-structure. True samples represent those predicted curves that are classified correctly, and false positive samples are the possible curves that could be classified incorrectly due to lack of observation data. Certainty level (CL) of each class is extracted by the case-based reasoning approach and is inspired from the number of false positive samples in the dataset for that category. More false positive samples decrease the certainty level of classifier's decision. The certainty level is also affected by the certitude level of epidemiologists in training the smart director and defining rules for each class (Refer to the following section).

State Number	Amplitude Deviation	Time Deviation
1-UL	Underestimate	Late prediction
2-UE	Underestimate	Early prediction
3-UN	Underestimate	Negligible
4-OL	Overestimate	Late prediction
5-OE	Overestimate	Early prediction
6-ON	Overestimate	Negligible
7-NL	Negligible	Late prediction
8-NE	Negligible	Early prediction
9-NN	Negligible	Negligible

Table 3.5. Possible categories of predicted epidemic curve in comparison with the observed curve. Negligible means close to observed data in terms of time or case counts.

Components of Smart State Dynamics: Smart Director

Knowing the class of each state vector (particle), the *smart director* decides about the drifting direction of each parameter. Fig 3.4 presents an example of Overestimate_Early (OE) prediction of two particles that generated huge and fast pandemics in comparison with the ground truth curve. For such a case, the smart director decides to change the epidemic parameters in ways that slow down the epidemic dynamics, such as decreasing the transmission rate and/or infectious period.

- **Training the smart director.** The *director* is trained based on the epidemiologist's opinion about the effect of each epidemic parameter on the trend of the pandemic. In other words, epidemiologist should simply specify the positive, negative, or neutral effect of parameters that should get calibrated on slowing down or speeding up the epidemic. Refer to Table 3.3 for the specified parameters.

  For each possible class of a predicted curve, the epidemiologist determines the perturbation direction of each parameter to push the trend of epidemic towards the ground truth curve:

  $$\forall \theta_i \in \theta, \forall c_j \in C = \{1 : UL,\ 2 : UE, ...,\ 9 : NN\}\ \texttt{define}\ Dir(\theta_i, c_j) \in \{0, +1, -1\}$$

  where θ is the set of parameters, C is the set of all possible classes, and $Dir(\theta_i, c_j) = +1$, $Dir(\theta_i, c_j) = -1$, and $Dir(\theta_i, c_j) = 0$ means the *increment*, *decrement*, and *fixing* of the parameter θ_i, respectively. For each class like c_j, Dir function divides the collection of parameters into two sets of *changing* and *fixed* parameters denoted by θ_{change} and θ_{fix}, respectively:

$\forall\theta_i \in \theta_{change},\ Dir(\theta_i, c_j) \neq 0$, and $\forall\theta_k \in \theta_{fix},\ Dir(\theta_k, c_j) = 0$. The epidemiologist should also define the logical rules between the parameters with logical conjunction ($\wedge$) and disjunction ($\vee$) operators. For each class $c \in C$, the epidemiologist may divide the set of changing parameters (θ_{change}) into two sets of *mandatory* (θ_{man}) and *optional* (θ_{opt}). Mandatory parameters are those whose values must be changed to achieve better results, whereas optional corresponds to those whose change of value could be promising but not necessary. Table 3.6 represents an example of two hypothetical rules for two possible classes. For simplicity, $Dir(\theta_i, c_j) = +1$ and $Dir(\theta_i, c_j) = -1$ are denoted by θ_i^+ and θ_i^-. The first rule implies that the parameters θ_1 and θ_2 *must* be increased and decreased, respectively, in a combination of parameters for generating new particles. For optional parameters such as θ_3, θ_4, and θ_5, both new and old values should be used in various multi-factorial combinations. Considering some parameters as mandatory and fixed results in pruning the unnecessary set of parameters and reducing the number of expanded particles. Table 3.7 lists the set of rules we define for all nine classes (C) and between all calibrating parameters for the Ebola epidemics. *Note that the training procedure of the smart director happens* ***only once*** *during the setup process of the framework.*

- **Smart Director Operation.** The trained *smart director* is embedded in the cycle of particle filter framework. It receives resampled particles and expands each of them into multiple particles by perturbing one or more parameters in the desired directions and generating new combinations of them. Algorithm 3 demonstrates the operation of the smart director in exploring the search space more efficiently.

Components of Smart State Dynamics: Adaptive Tuner

The *adaptive tuner* receives the history of the particle's evolution and determines the amount of perturbation (step size) for each parameter to help converging to the optimal solution. The inputs of the adaptive tuner are some *features* of both current and parent particles like the distance of the current/parent particles from the ideal solution, present and future perturbation directions for each parameter, etc. The error function used for this purpose is the wwL1 error which measures the distance between the observed and the predicted curve. Algorithm 4 presents the procedure of the adaptive tuner. If the perturbation direction of a parameter alternates frequently, it indicates that the step size is too large in that area of the search space. Therefore, the adaptive tuner decreases the step size to avoid particle filter from fluctuating in near-optimal-solution space. On the other hand, if the perturbation direction is fixed in sequential iterations, the tuner evaluates the effect of the step size by considering the amount of change in the prediction error in current and parent particles. When the error of current particle does not change significantly compared to the previous (parent) particle, it implies that the step size of the current parameters is too small for that

area. Adjusting the step size results in larger steps in flat areas of the search space to move faster toward better solutions. On the other hand, when the prediction error gets very small compared with the parent particle, the adaptive tuner assumes that the current particle is getting close to the optimal solution, so it slows down the evolution of the parameters by decreasing their step sizes (Algorithm 4).

Note that each particle carries the step sizes and perturbation directions for all of its parameters; therefore, the step size of one single parameter could be different with various particles.

Components of Smart State Dynamics: Random Diffuser

As mentioned before, the classifier determines the level of certainty for its classification decisions. The *random diffuser* receives the certainty level (CL) or randomness degree (RD where $RD \propto 1/CL$) and generates some random particles somewhat far away from the current particle. This distance will increase at higher randomness degrees. The number of randomly generated particles (n_{RP}) is proportional to the randomness degree ($n_{RP} = 2^{RD}$). More uncertainty results in generating more random particles that are moved in erratic directions.

Deterministic perturbations achieved from smart director are also combined with little randomness in the *random diffuser* to explore the search space and avoid local optima traps.

The smart analysis mentioned above result in non-blind searches that achieve optimum values for the parameters in a faster way.

3.8 Experimental Results

We have tested the robustness and performance of the proposed method by conducting multiple experiments on both dependent and independent synthetic data. We have used our approach to model various diseases, like flu and Ebola, with specific parameters that describe the disease dynamics and intervention behaviors. In this section, we present the step-by-step and total experimental results that demonstrate the trend of evolution and convergence of particles to optimal solutions.

3.8.1 Experimental Tests for Forecasting Influenza Epidemic Data

In order to show the robustness of our method, we generate synthetic data by running the Epifast simulator with predetermined configurations. We use the pre-constructed contact network of Kansas state, with 2.83 million nodes, to simulate the spread of influenza across

it. Epifast generates stochastic output by applying enough randomness in simulating the disease dynamics across the contact network. We run Epifast 50 times and select one of the replicates as the ground truth data. Knowing the ground truth configuration, we design two different scenarios for our experiments:

I For the first scenario, we set the starting point of particles close to the actual solution and traced the evolution of the particles. The ideal behavior is that the particles, which are scattered by the diffusion process and typically got away from the optimal solution, move toward the correct result without unusual roving.

II For the second scenario, we set the starting point of particles far away from the actual solution to examine whether or not they converge to the correct solution.

In our experiments, the Epifast simulator repeats the simulation five times for each particle. In Figure 3.4 the epidemic curves, corresponding to one particle, are plotted in the same color. The vertical red line demonstrates the prediction week (PW). The data points of the ground truth time series before the prediction week are used as the training observed data for the prediction process, and the tail of ground truth time-series after PW is considered as the test data.

Figure 3.5 presents the evolution of particles for the first scenario. Five epidemic-curve replicates of five best particles in each cycle are shown. Replicates corresponding to one particle are represented with the same color. In the first scenario, the start settings of particles are close to the ground truth configurations. Therefore, in Figure 3.5a, all of the predicted curves are close to the ground curve (olive-green curve). Figure 3.5b shows scattered particles that are diffused based on the randomness of the system. However, the smart system dynamics recognizes the underestimation for the new particles (Figure 3.5-c), and the smart director changes the direction of diffusion based on the classifier output. As can be seen, after only running the particle filter for four iterations, all particles converged to one sample close to the ground truth setting (Figure 3.5d). The number of particles used for this test was 10.

In the second scenario, we initialize the parameter state vectors with values far away from the ground truth values. $N_2 = 20$ particles are generated for the first round. The state vector of each particle is run by Epifast five times to generate five different replicates of epidemic curve. The particles are narrowed down to $N_1 = 10$ in the resampling process and then expanded to $N3 = 50$ in the diffusion stage. The whole search cycle is repeated for multiple iterations (7 iterations in this case), for each prediction week (PW), and the 10 best particles are stored as the best sample solutions for the corresponding prediction week. As mentioned before, Epifast simulator repeats the simulation five times and generates five predicted curves for each particle. Each predicted curve is compared with the ground truth curve and is assigned by a likelihood score as its weight. The score of each particle is defined as the average of likelihood scores of the five replicates. This score is used as a criteria for selecting the 10 best

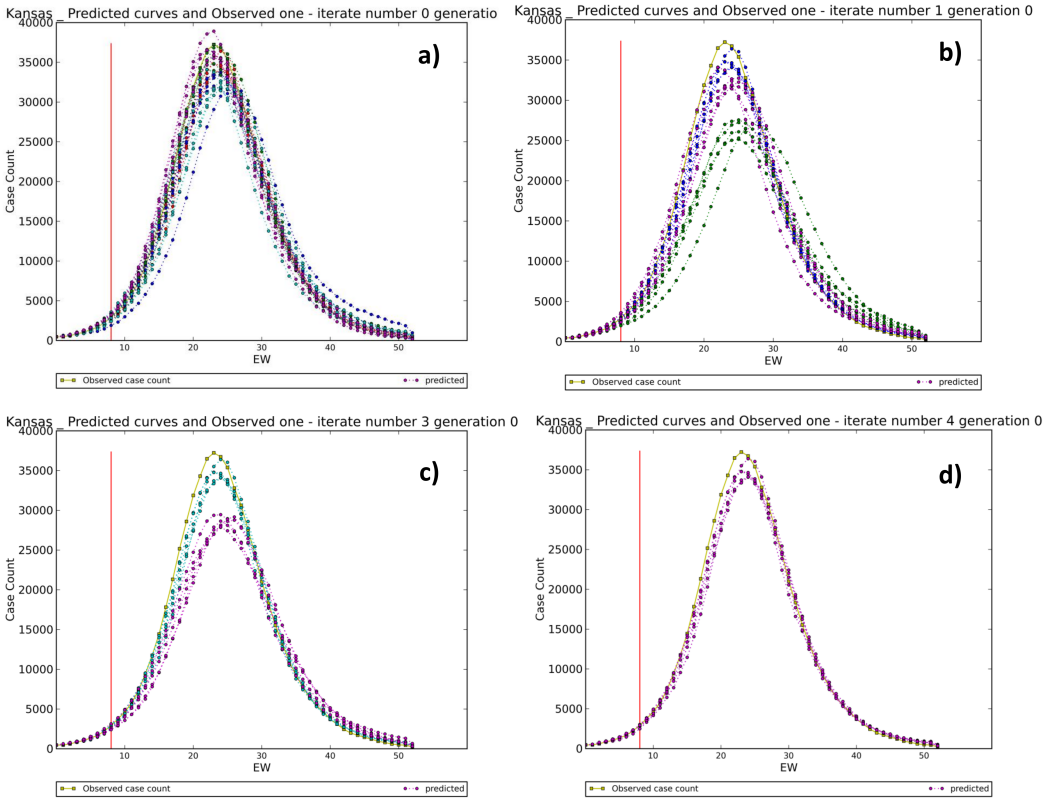


Figure 3.5. Predicted epidemic curves versus ground truth. The ground truth curve is shown in olive-green. a) Epidemic curves corresponding to the initial particles with a configuration near the ground truth. b) Deviation of predicted curves due to random scattering. c) Smart diffuser recognizes the direction of deviation (underestimation in this case) and directs the particles to the opposite direction (d).

particles among all generated ones. Since the resampling process is without replacement, a weighed accumulation is needed to represent the posterior distribution correctly. We calculate the weighted mean and weighted standard deviation of the predicted value in each week for the entire epidemic curve. Figure 3.6 visualizes the weighted epidemic curves corresponding to the best particles. To measure the weighted mean of predicted curves, individual replicates of best samples and their weights are considered. Figure 3.6a shows the weighted average curve associated with the best particles of the initial samples before going through the calibration process. Figure 3.6b demonstrates the improvement in the results after seven iterations of the search cycle. Prediction week is assumed to be week 11, which means the algorithm uses the data up to week 11 as the training data to perform the parameter adjustment. Figures 3.6c and 3.6d show the output of the SBFP for predicting weeks 13 and 14, respectively. Since a few more data points of ground truth data are used for the training process, more accurate predictions are achieved.

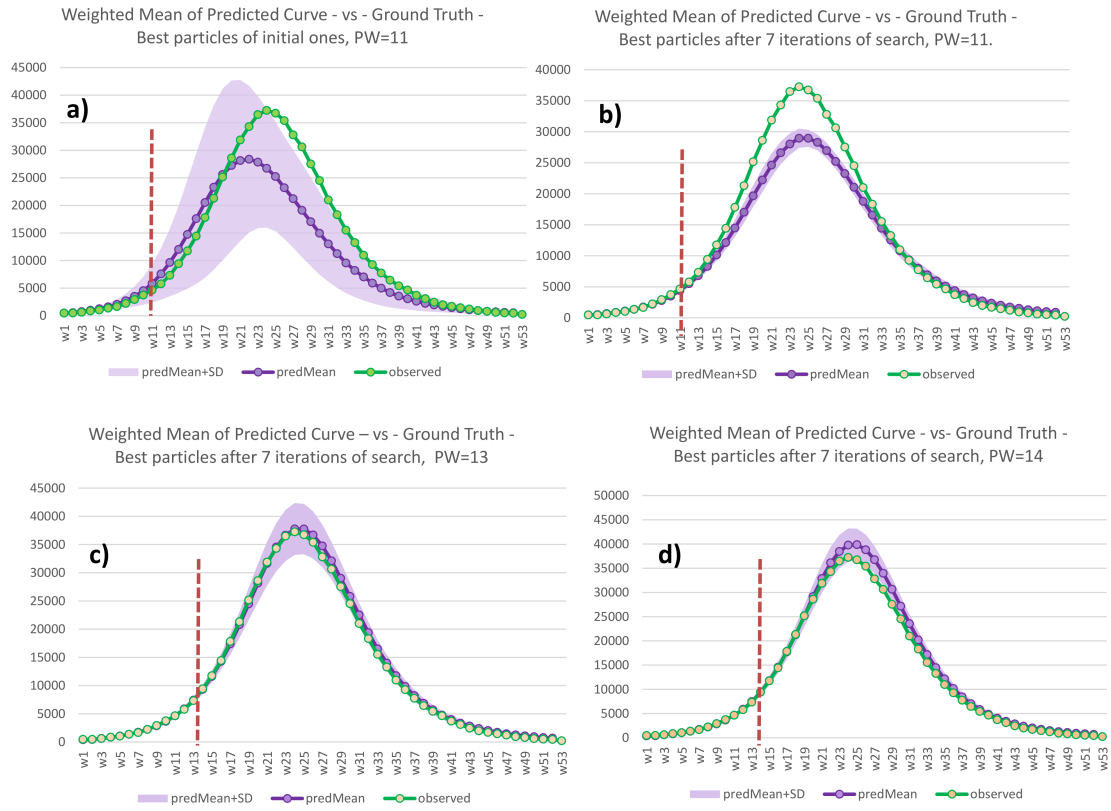


Figure 3.6. The weighted mean of the predicted epidemic curves versus ground truth. The ground truth curve is shown in green. a) Weighted mean ± one standard deviation of epidemic curves corresponding to the initial particles with a configuration far away from the ground truth. b) Prediction for week 11: Improvement in the results by performing search cycle for seven iterations. c) Prediction week is assumed week 13, d) and week 14 respectively: Updating the particles using more observed data.

We calculated some Epi-features [13] of the predicted epidemic curves and the ground truth to quantify the output of the forecasting algorithm based on epidemiologically relevant features. The selected Epi-features are: Peak value and time, Take-off value and time, and Intensity Duration length and start-time. For more information about the epidemiologically relevant features (Epi-features) and evaluation error measures refer to Chapter 5. Figure 3.7 demonstrates the weighted mean and weighted standard deviation of predicted Epi-features for different prediction weeks. PW=0 corresponds to the initial particles that are samples of prior distributions of parameters. This figure shows how the accuracy of prediction for different Epi-features improved by our proposed method and the forecasted values of Epi-features approached the ground truth values.

These two tests performed on the synthetic data, show the robustness and capabilities of the smart state dynamics of the SBPF framework for determining the search direction and

for converging to better results without roving around the search space. In the next set of experiments, we evaluate the accuracy and robustness of our agent-based model in simulating the propagation of Ebola. It is intended to demonstrate how the SBPF algorithm can adjust multiple unknown parameters of the agent-based model for forecasting Ebola pandemics given independent data.

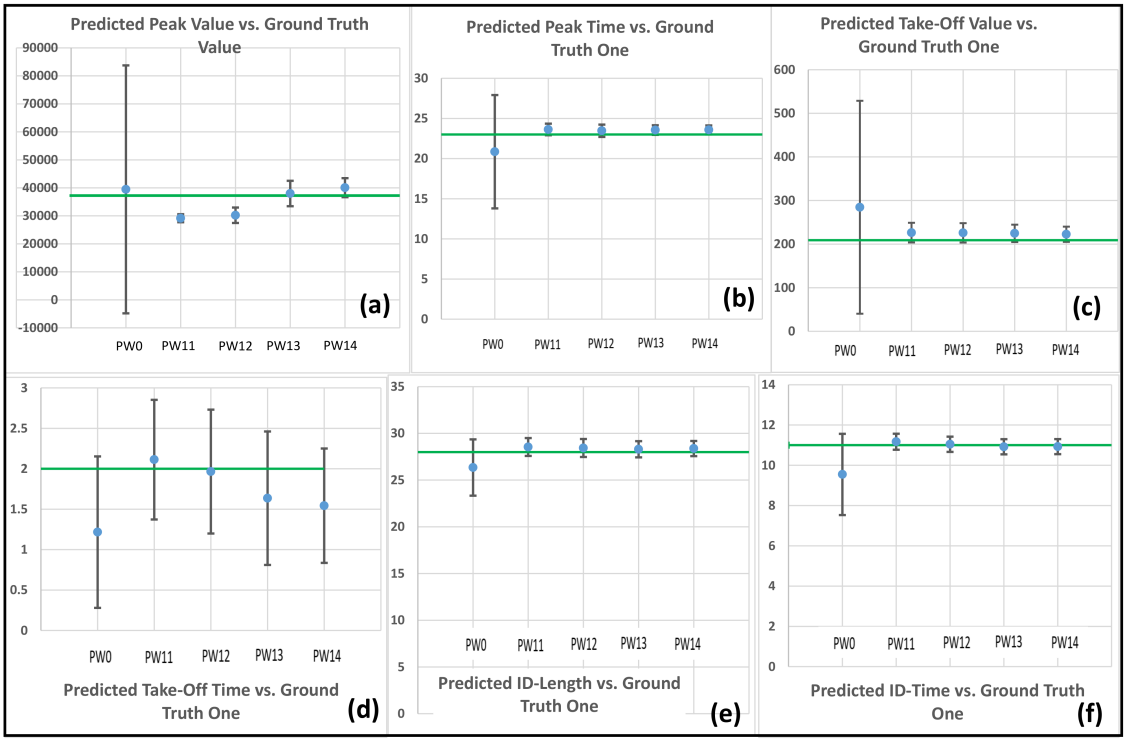


Figure 3.7. Weighted mean ± one standard deviation of different Epi-features calculated for long-term prediction of epidemic curves in prediction weeks 11, 12, 13, and 14. PW=0 represents the Epi-feature values corresponding to the initial particles before going through the calibration process. a) Predicted Peak Value, b) Predicted Peak Time, c) Predicted Take-Off Value, d) Predicted Take-Off Time, e) Predicted Intensity Duration Length, and f) Predicted Intensity Duration Start Time, all versus the ground truth one.

3.8.2 Experimental Tests for Forecasting Ebola Epidemic Data

The dataset we used for modeling and calibrating Ebola disease parameters was provided for the Ebola Challenge [70], organized under the Research and Policy for Infectious Disease Dynamics (RAPIDD) program at the NIH. This Ebola disease dataset was generated by a previously published agent-based model [80] calibrated with real data of 2014 Ebola epidemic for Liberia under four different scenarios. These scenarios emulate various epidemiological

changes, intervention reactions, and availability of data in the real world. Epidemic data for each scenario was released in five time points, and each time point contained outbreak situation reports, weekly reported new Ebola Virus Disease (EVD) cases at the county and country level, and current/future intervention plans for preventing and fighting the outbreak. The detailed description of the data and plans for each time point are included in appendix A. As the population of Liberia is about 4.5 million, the constructed contact network which is used by Epifast is made of 4.5 million nodes representing the individuals. Table 3.8 summarizes and compares the settings used for running the forecasting experiments for both influenza and Ebola epidemics.

New EVD cases were forecasted by each team at one, two, three and four weeks after each time point as the short-term prediction. As the intervention plans and strategies change from one time point to another, the long-term prediction at the first and second time points would not be expected to be sufficiently accurate. We have used the country-level incidence time-series data and have predicted the short-term epidemic curve. Figures 3.8 and 3.9 represent the short-term predicted epidemic curves for the first, second, third, and forth time points of the first scenario. The weighted mean and weighted standard deviation of new cases of EVD are calculated based on the best weighted particles among all particles generated in the repetitive search iterations. Each particle is run by Epifast five times to generate five different replicates of epidemic curve. Figures 3.8 and 3.9 demonstrate that the SBPF could perfectly calibrate the configuration parameters of the epidemic model such that the weighted-mean of predicted curves matches the ground truth curve in the training part. The short-term prediction, shows a little over-estimation, especially for the second time point. The over-estimation originates from incomplete information that is provided at each time point regarding future intervention plans. For example, in the second time point at week 20, it is claimed that three Ebola Treatment Units (ETU) will be opened in three counties between week 20 and week 24, and there is no other plan for neither contact tracing [73, 72], nor for a safe burial protocol. However, in the next report for the time point three, it is announced that 10292 individuals had been contact traced between week 20 and 26 to prevent the disease propagation. Also, an effective safe burial protocol had enforced since week 21. This shows the fidelity and consistency of our model with regard to the real-life intervention plans. Figures 3.8 and 3.9 demonstrate that the SBPF could perfectly calibrate the configuration parameters of the epidemic model such that the weighted-mean of predicted curves matches the ground truth curve in the training part. The short-term prediction, shows a little over-estimation, especially for the second time point. The over-estimation originates from incomplete information that is provided at each time point regarding future intervention plans. For example, in the second time point at week 20, it is claimed that three Ebola Treatment Units (ETU) will be opened in three counties between week 20 and week 24, and there is no other plan for neither contact tracing [73, 72], nor for a safe burial protocol. However, in the next report for the time point three, it is announced that 10292 individuals had been contact traced between week 20 and 26 to prevent the disease propagation. Also, an effective safe burial protocol had enforced since week 21. This shows the fidelity and consistency of our model with regard to the real-life

intervention plans.

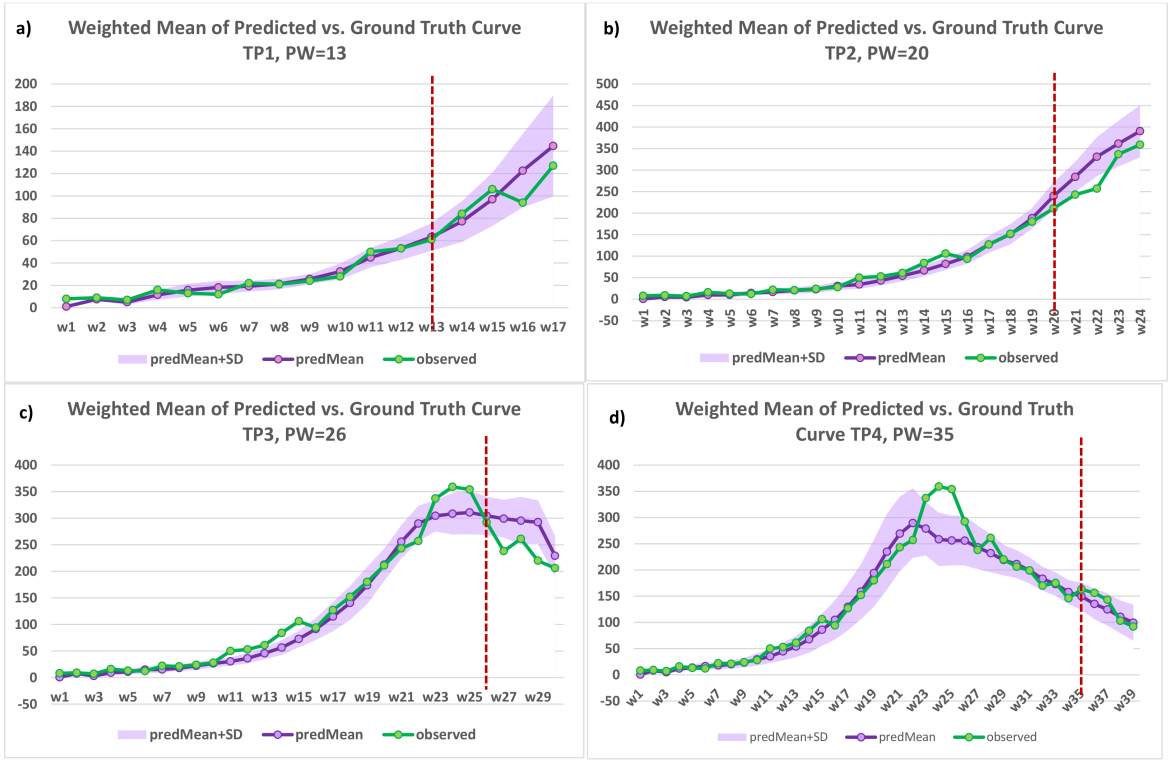


Figure 3.8. The weighted mean of predicted versus ground truth curve - The weighted mean of best particles after 20 iterations of search. a) Demonstrates the short-term prediction for time point 1 (PW=13), b) Shows the prediction for time point 2 (PW=20), c) Shows the prediction for time point 3 (PW=26), and d) Shows the prediction for time point 4 (PW=35).

We calculated several error metrics over the short-term predicted curves. The predefined set of error metrics included: Pearson's correlation coefficient, mean absolute error (MAE), the mean absolute percentage error (MAPE), and root mean square error (RMSE). We compared our results with the available output of other teams participated in the 2015 Ebola challenge [81, 82, 83, 84, 85, 86, 87, 88]. Among these eight teams, five of them [82, 83, 87, 85, 86] mainly published visualized graphs rather than detailed error-measures. We compared the aforementioned error-measures with those provided by Pell et al. [81] for their two phenomenological models, with the ensemble method of Venkatramanan et al. [88], and with overall results of Gaffey and Viboud approach [84]. The first method of Pell et al. named Logistic Equation (LGM), consistently underestimated the epidemic curve and consequently, the Epi-features. Although it shows lower MAPE error compared with our method, the negative value of Pearson's correlation coefficient demonstrates that the LGM's epidemic-curve does not follow the trend of the epidemic in the correct direction [81]. Pell et al. proposed another method, named Generalized Richards Model (GRM), that generates

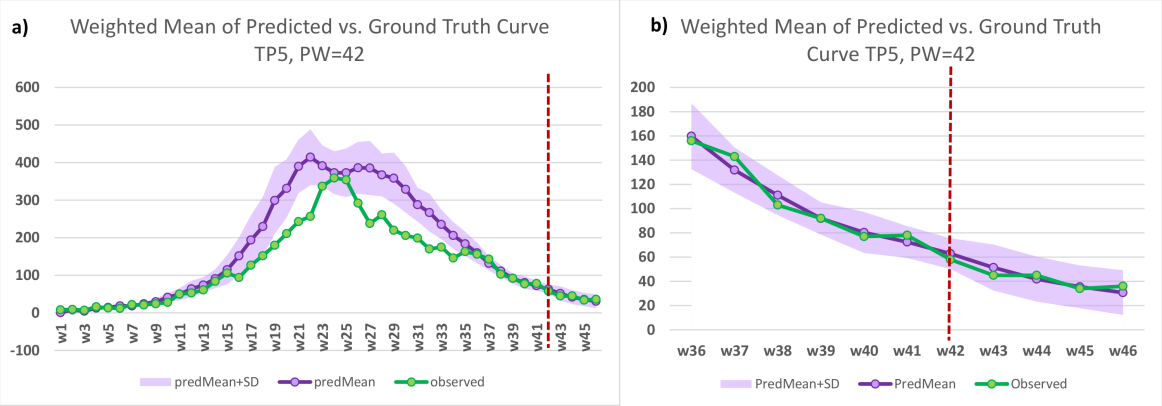


Figure 3.9. The weighted mean of predicted versus ground truth curve - The weighted mean of best particles after 15 iterations of search. Both curves belong to time point 5 (PW=42). a) Shows the whole curve up to four data points after PW=42 , b) Close up view of the curves around prediction week (PW=42).

positive values for Pearson's correlation coefficient for most of the data points. Our SBPF approach demonstrates Pearson value of 0.80, 0.94, 0.73, 0.98, and 0.83 for time points 1 to 5, respectively, that shows agreement between the output and the trend of incidence data. The ensemble method of Venkatramanan et al. [88] is represented by VTC in the charts. Our proposed SBPF generates lower MAPE and RMS errors in all time points comparing with both GRM and LGM approaches. SBPF approach demonstrates better performance in all time points except the second one in comparison with VTC method (see Figs 3.10, 3.12, and 3.11). For our method, the error measures are calculated on the weighted mean of the curves, and for other stochastic approaches, they are measured on median of predicted curves.

Gaffey and Viboud presented the performance of an extension of the CDC-initiated Ebola response modeling tool for the Ebola forecasting challenge [84]. They only provided the overall error measures calculated across the short-term predictions on all five time points. We refer to their method with the term 'CDC'. Figure 3.13 compared the overall value of some error measures for the aforementioned five methods: GRM, LGM, SBPF, VTC, and CDC. As can be seen, SBPF framework, shows pretty good performance in comparison to all these methods.

We have also calculated some Epi-features on the predicted epidemic curves generated by SBPF algorithm (see Chapter 5 for the definitions). Figure 3.14 represents the weighted average and percentiles of the following features achieved from multiple epidemic curves: Peak Value, Peak Time, Take-Off-Value, Take-Off-Time, ID-Length, and ID.Time.

As CDC method is a deterministic, mechanistic algorithm, the reported values of the Epi-features are single values for each time-point without uncertainty percentiles. Pell et al. [81], did not provide numerical values of Epi-features for none of their methods. Figure 3.15

Figure 3.10. Error metrics: Pearson's correlation coefficient, MAPE, and RMSE measure the direction and amount of deviation of short-term predictions of GRM, LGM, SBPF and VTC. Figures a, b, and c are the statistics corresponding to the first time point. Figures d, e, and f are associated with the second time point.

compares the precision of SBPF, VTC and CDC approaches in predicting Peak Value and Peak Time. As can be seen, SBPF demonstrates a little overestimation in predicting the Peak Value for the first and second time points because of incomplete information about future interventions. As the data reported for time point 3 are more accurate and consistent with future intervention plans, the agent-based model generates more precise predicted curves as well as the predicted Peak Values.

Figure 3.11. Error metrics: Pearson's correlation coefficient, MAPE, and RMSE measure the direction and amount of deviation of short-term predictions of GRM, LGM, SBPF and VTC. Figures a, b, and c are the statistics corresponding to the third time point. Figures d, e, and f are associated with the forth time point.

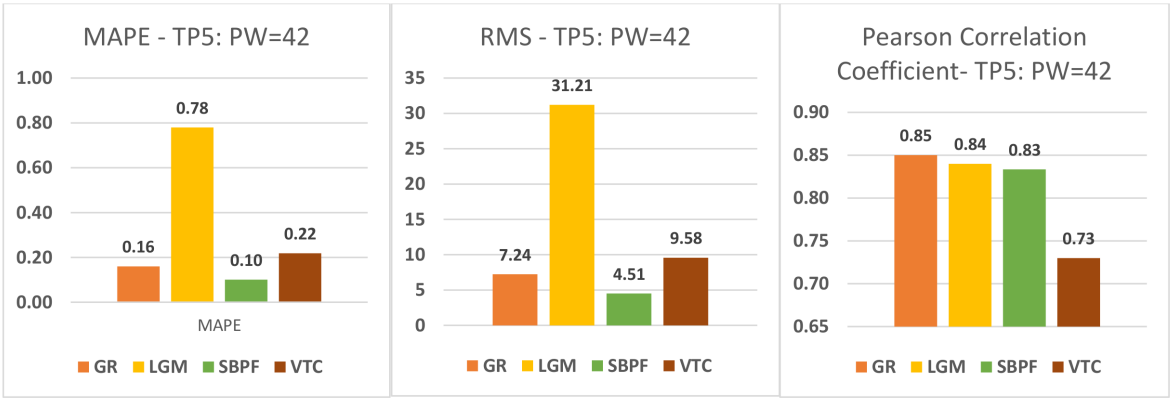


Figure 3.12. Error metrics: Pearson's correlation coefficient, MAPE, and RMSE measure the direction and amount of deviation of short-term predictions of GRM, LGM, SBPF and VTC for time point 5.

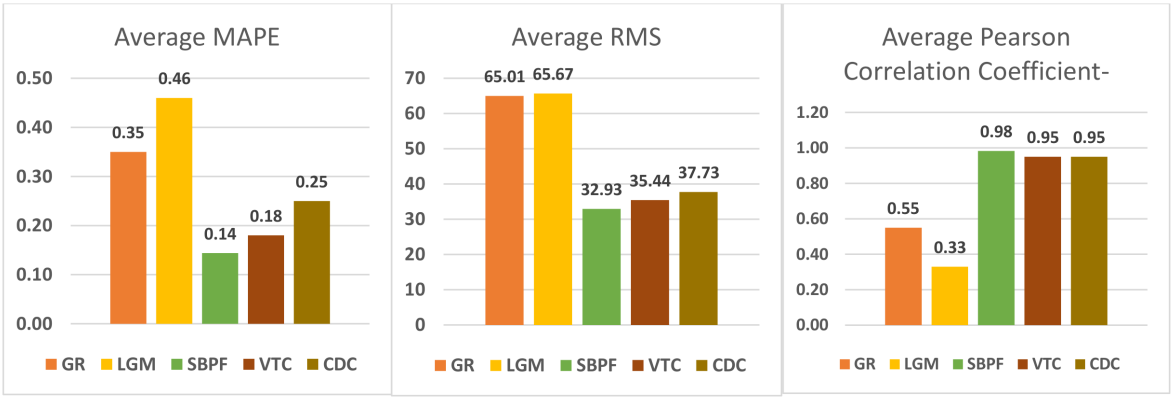


Figure 3.13. Overall value of error measures across all five time points: Pearson's correlation coefficient, MAPE, and RMSE measure the direction and amount of deviation of short-term predictions of GRM, LGM, SBPF, VTC and CDC.

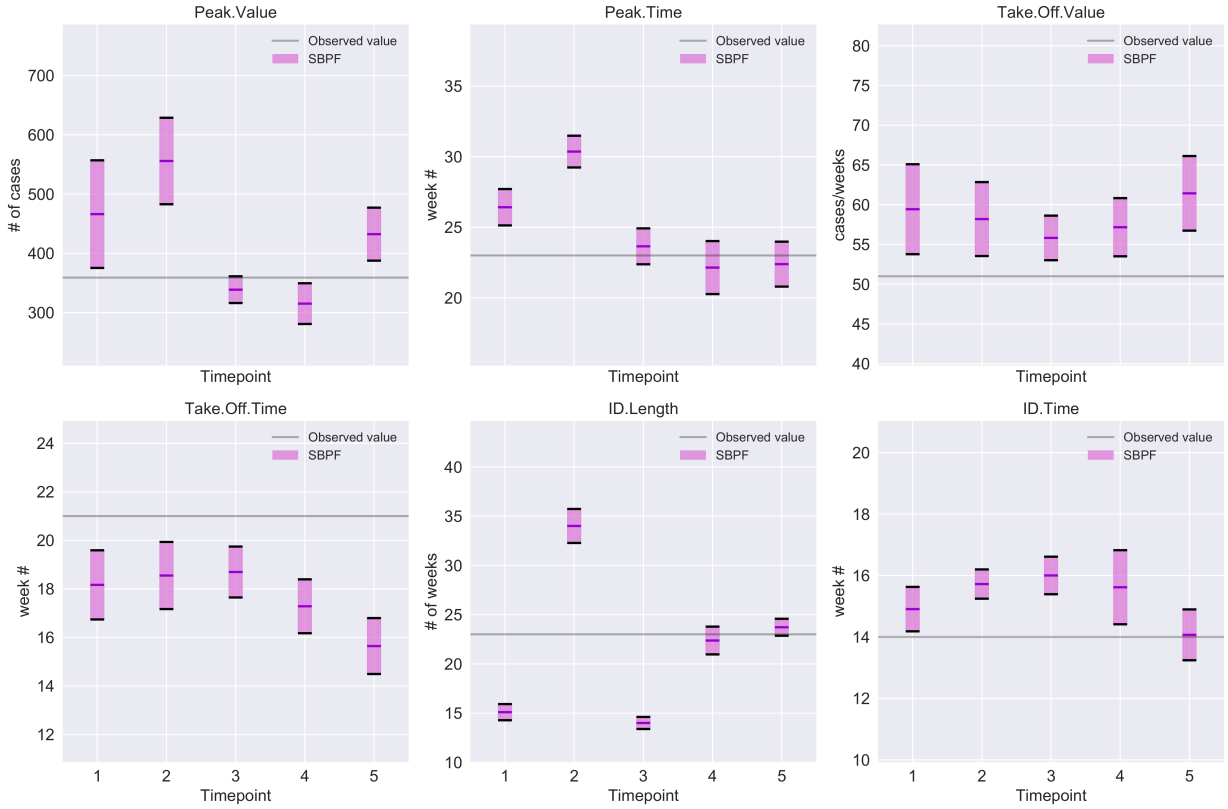


Figure 3.14. Distribution of predicted Epi-features by SBPF approach for time points 1 to 5, from left to right.

ALGORITHM 2: The epidemic curve classifier

Input: Features of the predicted and ground truth epidemic curves:
L_P, L_G: Length of the epidemic season for the predicted and ground truth curves, respectively;
PV_P, PV_G: Predicted and observed peak values so far (till prediction week); PT_P, PT_G: Predicted and observed peak times so far (till prediction week); PW: Prediction week.
Output: Class: Category of the predicted epidemic curve in comparison with the observed curve;
$CL \in$ [medium, very high]: Certainty level about the classification decision;
RD: Randomness degree, where $RD \propto 1/CL$.

if *($L_P < PW$ and $L_G > PW$):* **then** `/* Predicted curve terminated soon: */`
  if *($PV_P >> PV_G$):* **then** `/* Prediction is overestimated and early: */`
    Leaf node A_1: Class=5-OE; CL =Very high; $RD = 0$;
  else if *($PV_P << PV_G$):* **then** `/* Prediction is low and early: */`
    Leaf node B_1: Class=2-UE; CL =Very high; $RD = 0$;
  else `/*` $PV_P \approx PV_G$ `*/`
    Leaf node C_1: Class=8-NE; CL =High-medium; $RD = 2$;
  end
else if *($PT_P << PT_G$):* **then** `/* Prediction is soon: */`
  if *($PV_P >> PV_G$):* **then** `/* Prediction is overestimated and early: */`
    Leaf node A_2: Class=5-OE; CL = Medium; $RD = 3$;
  else if *($PV_P << PV_G$):* **then** `/* Prediction is underestimated and early: */`
    Leaf node B_2: Class=2-UE; CL =High; $RD = 1$;
  else `/* Prediction is early, but with similar amplitude as the observed data: */`
    Leaf node C_2: Class=8-NE; CL =Medium; $RD = 3$;
  end
else if *($PT_P >> PT_G$):* **then** `/* Prediction is late: */`
  if *($PV_P >> PV_G$):* **then** `/* Prediction is overestimated and late: */`
    Leaf node D: Class=4-OL; CL =High; $RD = 1$;
  else if *($PV_P << PV_G$):* **then** `/* Prediction is underestimated and late: */`
    Leaf node E: Class=1-UL; CL =High-medium; $RD = 2$;
  else `/* Prediction is late, but with similar amplitude as the observed data: */`
    Leaf node F: Class=7-NL; CL =Medium; $RD = 3$;
  end
else `/*` $PT_P \approx PT_G$ `*/`
  if *($PV_P >> PV_G$)* **then** `/* Prediction is only overestimated: */`
    Leaf node G: Class=6-ON;
  else if *($PV_P << PV_G$):* **then** `/* Prediction is only underestimated: */`
    Leaf node H: Class=3-UN;
  else `/* Prediction is in the same range as the observed data */`
    Leaf node I: Class=9-NN;
  end
  if *($T_1 = PW$):* **then** `/* Peak time=PW: Higher chance of false decision: */`
    Leaf node $\{G,H,I\}_1$: CL=Medium, $RD = 3$
  else
    Leaf node $\{G,H,I\}_2$: CL=High, $RD = 1$
  end
end

Class Label	**Mandatory Parameters**	**Optional Parameters**	**Fixed Parameters**
c_1	(θ_1^+, θ_2^-)	$\{\theta_3^+, \theta_3\} \times \{\theta_4^+, \theta_4\} \times \{\theta_5^-, \theta_5\}$	θ_6
c_2	-	$\{\theta_1^+, \theta_1\} \times \{\theta_3^+, \theta_3\} \times \{\theta_4^+, \theta_4\} \times \{\theta_5^-, \theta_5\}$	(θ_2, θ_6)

Table 3.6. Example of two hypothetical rules for two possible classes. For simplicity, $Dir(\theta_i, c_j) = +1$, $Dir(\theta_i, c_j) = -1$, and $Dir(\theta_i, c_j) = 0$ are denoted by θ_i^+, θ_i^-, and θ_i, respectively. $'\times'$ denotes the Cartesian product of the sets.

Class Label	**Mandatory Parameters**	**Optional Parameters**	**Fixed Parameters**
$c = 1 : UL$	β^+	$\{I_0^+, I_0\} \times \{\omega_{TR}^+, \omega_{TR}\} \times \{\beta_{HE}^+, \beta_{HE}\} \times \{\beta_{HD}^+, \beta_{HD}\} \times \{\beta_{ETU}^+, \beta_{ETU}\}$	-
$c = 2 : UE$	-	$\{\beta^+, \beta\} \times \{I_0^+, I_0\} \times \{\omega_{TR}^+, \omega_{TR}\} \times \{\beta_{HE}^+, \beta_{HE}\} \times \{\beta_{HD}^+, \beta_{HD}\} \times \{\beta_{ETU}^+, \beta_{ETU}\}$	-
$c = 3 : UN$	-	$\{\beta^+, \beta\} \times \{I_0^+, I_0\} \times \{\omega_{TR}^+, \omega_{TR}\} \times \{\beta_{HE}^+, \beta_{HE}\} \times \{\beta_{HD}^+, \beta_{HD}\} \times \{\beta_{ETU}^+, \beta_{ETU}\}$	-
$c = 4 : OL$	(β^-, I_0^+)	$\{\omega_{TR}^-, \omega_{TR}\} \times \{\beta_{HE}^-, \beta_{HE}\} \times \{\beta_{HD}^-, \beta_{HD}\} \times \{\beta_{ETU}^-, \beta_{ETU}\}$	-
$c = 5 : OE$	β^-	$\{I_0^-, I_0\} \times \{\omega_{TR}^-, \omega_{TR}\} \times \{\beta_{HE}^-, \beta_{HE}\} \times \{\beta_{HD}^-, \beta_{HD}\} \times \{\beta_{ETU}^-, \beta_{ETU}\}$	-
$c = 6 : ON$	-	$\{\beta^-, \beta\} \times \{I_0^-, I_0\} \times \{\omega_{TR}^-, \omega_{TR}\} \times \{\beta_{HE}^-, \beta_{HE}\} \times \{\beta_{HD}^-, \beta_{HD}\} \times \{\beta_{ETU}^-, \beta_{ETU}\}$	-
$c = 7 : NL$	β^+	$\{\omega_{TR}^-, \omega_{TR}\} \times \{\beta_{HE}^-, \beta_{HE}\} \times \{\beta_{HD}^-, \beta_{HD}\} \times \{\beta_{ETU}^-, \beta_{ETU}\}$	I_0
$c = 8 : NE$	β^-	$\{\omega_{TR}^+, \omega_{TR}\} \times \{\beta_{HE}^+, \beta_{HE}\} \times \{\beta_{HD}^+, \beta_{HD}\} \times \{\beta_{ETU}^+, \beta_{ETU}\}$	I_0
$c = 9 : NN$	-	-	$(\beta, I_0, \omega_{TR}, \beta_{HE}, \beta_{HD}, \beta_{ETU})$

Table 3.7. Perturbation direction rules for calibrating parameters. For each possible class of a predicted curve, the epidemiologist determines the perturbation direction of each parameter to push the trend of epidemic towards the ground truth curve. For simplicity, $Dir(\theta_i, c_j) = +1$, $Dir(\theta_i, c_j) = -1$, and $Dir(\theta_i, c_j) = 0$ are denoted by θ_i^+, θ_i^-, and θ_i, respectively. $'\times'$ denotes the Cartesian product of the sets.

ALGORITHM 3: Smart director operation

Input: Particle (state vector) $x = (\theta_1 = \theta_1^{old}, \theta_2 = \theta_2^{old}, ..., \theta_{n_x} = \theta_{n_x}^{old})$;
$M = (M_1, .., M_{n_x})$, where M_i is the set of meta data corresponding to i^{th} parameter of particle x;
and $c \in C = \{1 : UL,\ 2 : UE, ...,\ 9 : NN\}$: Category of the predicted epidemic curve generated by the set of parameters in particle x;
Output: *Set newXs*: A set of new particles generated by perturbing the parameters of particle x;

Set $newXs = \emptyset$;
$x_{new} = (\acute{\theta_1} = \emptyset, \acute{\theta_2} = \emptyset, ..., \acute{\theta_{n_x}} = \emptyset)$;
forall $\theta_i \in \theta$ **do**
  $M_i.newDirection = Dir(\theta_i, c)$;
  `/* Tune step size by calling the adaptive tuner */`
  $M_i.newStepSize$ = adaptive_tuner (M_i containing $M_i.oldDirection$, $M_i.newDirection$, $M_i.oldStepSize$; $x.currError$; $x.prevError$);
  `/* Perturb` θ_i^{old} `in the new direction based on the new step size. */`
  θ_i^{new} = Perturb (θ_i^{old}, $M_i.newDirection$, $M_i.newStepSize$);
end
forall *parameters* $\theta_i \in \theta$ **do**
  if $\theta_i \in \theta_{fixed}(c)$ **then**
    `/* Use the old value for` i^{th} `parameter from the parent particle. */`
    $\acute{\theta_i} = \theta_i^{old}$;
  if $\theta_i \in \theta_{man}(c)$ **then**
    `/* Use the new value for` i^{th} `parameter generated by` *`Perturb method`* `. */`
    $\acute{\theta_i} = \theta_i^{new}$;
end
for $k = 0$ *to* n_{opt} **do** //Among optional parameters
  forall *possible variation of selecting k parameters from* n_{opt} *optional parameters* **do**
   `/*` $\binom{n_{opt}}{k}$ `different variations */`
    foreach $j \in k-$*selected parameters* **do**
      $i = map(j)$; `/* Get corresponding index of parameter` $\theta_j \in \theta_{opt}$ `in state vector` x `*/`
      $\acute{\theta_i} = \theta_j^{new}$;
    end
    foreach $j \in$ *non-selected parameters* **do** `/*` $(n_{opt} - k)$ `parameters */`
      $i = map(j)$; `/* Get corresponding index of parameter` $\theta_j \in \theta_{opt}$ `in state vector` x `*/`
      $\acute{\theta_i} = \theta_j^{old}$;
    end
    Add a copy of x_{new} to $newXs$;
  end
end

ALGORITHM 4: Adaptive tuner procedure

Input: M_i: The set of meta data corresponding to i^{th} parameter of particle x containing $M_i.currDirection$, $M_i.newDirection$,and $M_i.currStepSize$;
$x.currError$: Meta data about prediction error of current particle;
$x.prevError$: Meta data about prediction error of previous (parent) particle from which particle x is originated;
Output: $M_i.newStepSize$: The new step size for perturbing parameter θ_i;
if *($M_i.newDirection \neq M_i.currDirection$) and ($M_i.currDirection \neq 0$)* **then**
`/* Fluctuation in direction: decrease the step size for` θ_i `*/`
$M_i.newStepsize = M_i.currStepsize/2$;
else
if $M_i.newDirection = M_i.currDirection$ **then**
$f = |\frac{|x.currError|-|x.prevError|}{|x.currError|}|$;
if f ¡ *minThreshold* **then** `/* Area is flat:step size is small. */`
$M_i.newStepsize = M_i.currStepsize \times \alpha$, where $\alpha > 1$;
else
if f ¿ *maxThreshold* **then** `/* Getting close to ideal solution: step size is large. */`
$M_i.newStepsize = M_i.currStepsize \times \rho$, where $\rho < 1$;
else `/* step size is fine. */`
$M_i.newStepsize = M_i.currStepsize$;
end
end
if $M_i.currDirection = 0$ **then**
$M_i.newStepsize = M_i.currStepsize$;
end

Disease	Type of Data	Data Simulator	Network	#Individuals	# Particles	#Iterations
Flu	Synthetic/ Dependent	Epifast	Kansas	2.83 M	$N_2 = 20$, $N_1 = 10$, $N3 \approx 50$	7
Ebola	Synthetic/ Independent	NIH simulator [80]	Liberia	4.5 M	$N_2 = 20$, $N_1 = 10$, $N3 \approx 140 - 275$	20

Table 3.8. Settings used for running the forecasting experiments for influenza and Ebola epidemics.

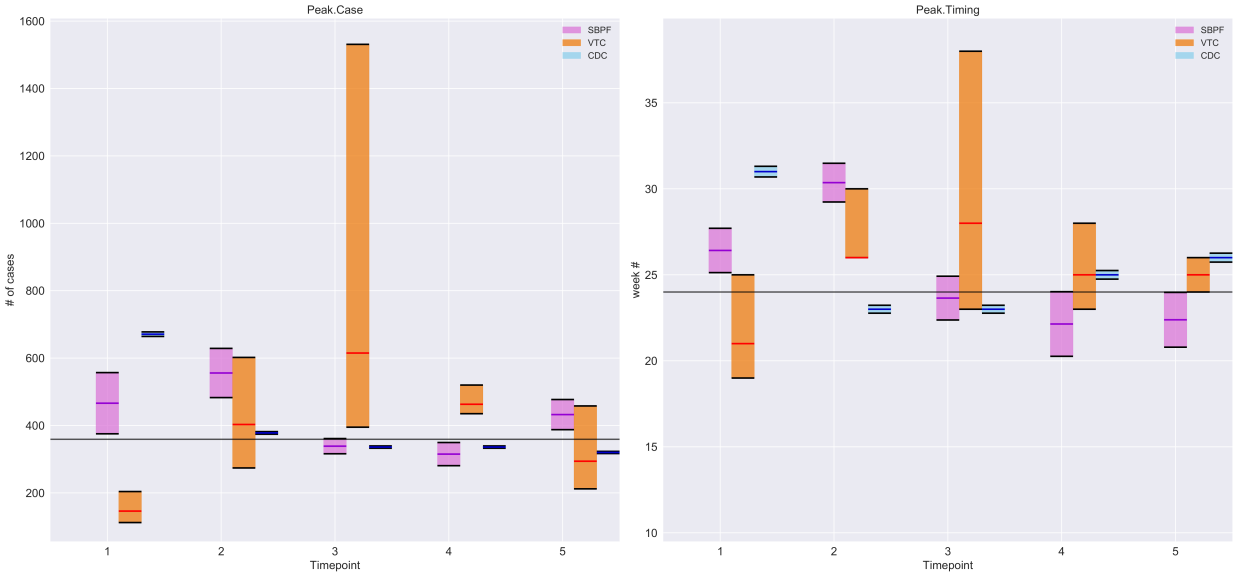


Figure 3.15. Distribution of predicted Peak-Value and Peak-Time around the observed value for three methods - SBPF, VTC, and CDC - for time points 1 to 5.

3.9 Summary of Results and Achievements

3.9.1 *Finding 1. The Smart Beam-Particle Filter is able to successfully calibrate/adjust the epidemic curve in the training part with a low rate of training error.*

Support: As can be seen in Figures 3.5, 3.6, and 3.8, our proposed SBPF framework is able to successfully calibrate the configuration parameters of the agent-based model such that the weighted-mean of the corresponding epidemic curves matches the ground truth curve in the training part.

3.9.2 *Finding 2. Our proposed particle filter method combined with agent-based model could successfully predict the trend of the epidemic of flu and Ebola with reasonable prediction errors.*

Support: The experimental results in Figures 3.10 and 3.12 demonstrate high value of Pearson correlation coefficient that represents the high degree of correlation between the output of the SBPF and the trend of incidence data. This figure also demonstrates that the SBPF has lower prediction error in comparison to most of available approaches

3.9.3 *Finding 3. The proposed agent-based-model overestimates the epidemic forecast in the cases that knowledge about future interventions is incomplete. While these results show the sensitivity of the model to correct/noisy information, they also confirm the fidelity of the model and its consistency with real-life facts and activities.*

Support: As it was shown in Figure 3.8, the applied agent-based model shows a little over-estimation, especially in the second time point at week 20. This over-estimation originate from the incomplete information about future interventions. While these results show the sensitivity of the model to correct/noisy information, they also confirm the fidelity of the model and its consistency with real-life facts and activities.

3.10 Conclusion

In this chapter, we have modeled the dynamics and propagation of diseases like influenza and Ebola by using an agent-based model that simulates the pandemic across a large contact network with millions of vertices as individuals. Moreover, this model provides capabilities to apply a wide range of policy-based and individual-based interventions. However, more capabilities of the method, impose more complexity and unknown parameters to the system. Consequently, We have proposed a smart particle filter framework to calibrate the desired parameters given the observed data as the training dataset. Using machine learning techniques, the proposed particle filter framework implements a smart state dynamics unit that regulates the direction and perturbation of the particles. The smart director diffuses and scatters the particle in targeted directions to yield a non-blind search that swiftly achieves optimum values for parameters. We have tested the robustness and performance of the proposed method by running several experiments on the synthetic data of influenza and Ebola. The experimental results show the successful calibration/adjustment of the curve in the training part with low rate of the training error. However, the proposed agent-based-model overestimates the epidemic forecast in the cases that the knowledge about future interventions is incomplete. While these results show the sensitivity of the model to correct/noisy information, they also confirm the fidelity of the model and its consistency with real-life facts and activities.

Chapter 4

Improving the time-complexity of Smart Beam Particle Filter framework

4.1 Introduction

In the previous chapter, we proposed a Smart Beam Particle Filter (SBPF) framework to calibrate the unknown parameters of a large-scale agent-based-model as a disease propagation simulator. Using machine learning techniques, the proposed particle filter framework implements a smart director that diffuses and scatters the particles in targeted directions to yield a non-blind search that swiftly achieves optimum parameter values. The SBPF was able to successfully calibrate and adjust the curve with low errors for both training and test data sets. However, having the agent-based-model embedded in the framework still imposes a huge overhead and time-complexity to the procedure, which makes it computationally expensive for real time systems. In this chapter, we perform a complexity analysis on the proposed SBPF methodology and develop new strategies to optimize the running time of the system without compromising performance.

The notations used in this chapter are the same as Chapter 3. The rest of the chapter is organized as follows. In Section 4.2, we discuss the time-complexity of the Smart Beam Particle Filter. In section 4.3, we present various methods to improve the framework running time, followed by experimental results in Section 4.4 that elaborate the trade-off between time-improvement and performance compromise. Section 4.5 provides the list of findings and achievements from our experimental results. Finally, we conclude the chapter in Section 4.6.

4.2 Complexity Evaluation

The complexity of the proposed framework, depends on many factors from which, the number of particles and the size of contact network are the dominant ones. Assuming that the SBPF cycle is repeated for n_{It} iterations, all units run for n_{It} times. For the first iteration, the epidemic simulator (Epifast) runs only for the initial number of particles (N_2). For consecutive iterations, the number of particles changes to N_3. Since Epifast output is stochastic, we repeat the simulator for R times for each particle to achieve non-biased results. In order to have an estimation of N_3, we need to dissect the Smart Dynamics unit. For each particle, the time complexity of the *feature selector* and the *classifier* components are $O(R \times L)$ and $O(R)$, respectively, where L denotes the length of the epidemic curve. For each parameter $\theta_i \in \theta_1, .., \theta_{n_x}$ that needs to be calibrated, the smart director calls the adaptive tuner, which is $O(1)$. The *smart director* expands each particle to multiple ones in the number of $O(2^{n_x})$, and then *random diffuser* adds some randomness to each parameter of the expanded particles. Moreover, the random diffuser generates new random particle to escape from local optimal traps in the number of $O(2^{RD})$, where RD is the randomness degree and is proportional to the uncertainty of the classifier and smart director. Overall, the time complexity of the Smart Dynamics unit for each particle is $O(R.L + 2^{RD} + 2^{n_x})$ and the number of expanded particles from a single one is $O(2^{RD} + 2^{n_x})$. Therefore, $N_3 = N_1 \times (2^{RD} + 2^{n_x})$, where N_1 is the number of resampled particles in each iteration.

The time complexity of Epifast is linear in terms of the contact network ($O(|V| + |E|)$), and creates a bottle neck int the SBPF framework. The number of Epifast runs is equal to the total number of particles ($O(N_3 \times n_{It})$), where $N_3 = O(N_1 \times (2^{RD} + 2^{n_x}))$.

4.3 Time-complexity Improvement

According to the system complexity, we suggest three main strategies to improve the running time of the framework:
i) Reduce the number of expanded particles (2^{n_x})
ii) Reduce the number of random particles (2^{RD})
iii) Reduce the number of resampled particles (N_1)

4.3.1 Reduce the Number of Expanded Particles

In a regular SBPF, the search process follows the Breadth-first search (BFS) strategy, which means that it generates all the neighbors of each particle by changing 0 to n_x parameters and producing 2^{n_x} various combinations of them (Figure 4.1). In this figure, a green node means the parameter is changed, while a red node means the parameter is keeping its last value. Our first strategy to speed up the algorithm is to perform partial search instead of

the breadth search to reduce the number of expanded particles. Therefore, we only perturb k parameters out of the n_x ones, where $1 \leq k \leq n_x$. This results in only $\binom{n_x}{k}$ expanded particles. Figure 4.2 represents the perturbation tree, where 5 or 6 parameters are drifted. The number k is proportional to the prediction error; in other words, if the prediction error is high, more parameters should be changed to improve the results; while if the prediction error is low, a slight change in one or two parameters could lead to a better result (Figure 4.3).

We have selected Mean Absolute Percentage Error (MAPE) as the criteria to measure the prediction error because of its specific properties to scale the error. Different error measures and their specifications will be discussed in Chapter 5. Table 4.1 provides the corresponding domains and ranges of the MAPE error. Figure 4.4 demonstrates the corresponding colored spectrum of the MAPE ranges. The black curve shows the hypothetical observed curve (y). Each color demonstrates the predicted values (curves) that can generate the corresponding MAPE errors as listed in the table. According to Table 4.1, we selected five levels of threshold ($[0, 1/4, 1/2, 3/4, 1]$) to classify the prediction error to one of the five categories of $c = [1 : \textit{very low},\ 2 : \textit{low},\ 3 : \textit{medium},\ 4 : \textit{high},\ 5 : \textit{very high}]$. Then, for each class of prediction error like $c = i \in [1, ..., 5]$, we only perturb k parameters that $k \in [(i-1) \times \Delta k + 1, i \times \Delta k]$, and $\Delta k = \lceil n_x/5 \rceil$.

4.3.2 Reduce the Number of Random Particles

As mentioned before, random particles are generated by the random diffuser unit to help the system escape from local traps. Moreover, they compensate the uncertainty of the classifier and the smart director decisions. The number of randomly generated particles (n_{RP}) is proportional to the randomness degree ($n_{RP} = 2^{RD}$). More uncertainty results in generating more random particles that are moved in erratic directions. In the efficient version of SBPF, we decided to decrease the number of random samples by half ($n_{RP} = 2^{RD-1}$) to reduce the overhead. In the regular SBPF algorithm, all random and non-random particles are compared with each other in the weighting and resampling stages. Random particles usually have a lower chance of getting selected in competition with the old samples with higher likelihoods. Therefore, reducing the population of random particles decreases their chance of staying in the system and their efficacy. To solve this problem, we suggest an aged-based resampling algorithm, in which all samples are assigned with an age. This age will increase in each cycle of the particle filter algorithm. The random particles are assigned with age zero immediately after generation. The young particles are resampled separately from the adults in order to give them a chance of staying in the system for a few iterations and grow. The young particles gradually grow into adults after a few cycles and are compared with other adult particles in the resampling process. Algorithm 5 summarizes these steps.

4.3.3 Reduce the Number of Resampled Particles

The other strategy to improve the running time of SBPF is to reduce the number of resampled particles (N_1). We followed the *explore then exploit* policy, which means that we start with N_1 resampled particles and gradually decrease them to $N_1/2$. This allows the algorithm to explore more around the search space during the first iterations and then exploit the achieved knowledge in later tries.

4.4 Experimental Results

In this section, we analyze the performance and time-improvement of the efficient SBPF algorithm after applying all three strategies mentioned earlier. We have tested both regular and efficient SBPF frameworks for forecasting Ebola epidemics on time point 3, which is the most sensitive time point among the prediction weeks. Time point 3 is a few weeks after the epidemic peak time. A slight latency in the predicted curve and predicted peak time could result in a large prediction error for the short-term predictions, such as the 4 steps ahead forecasts. Figure 4.5 compares the performance and running time of the regular and efficient SBPF. The average number of particles, and consequently the running time of the efficient SBPF, decreased to 22% compared with the regular SBPF. On the other hand, the performance ratio dropped to 43% and 47% based on the weighted MAPE and weighted RMS errors, respectively.

Performance depreciation is mainly due to the partial search that prunes promising neighbors incorrectly. To avoid this issue, we only employ the last strategy by decreasing the number of resampled particles by half ($N_1/2$) for all SBPF iterations. As a result, the number of generated samples and running time dropped by 50%. Figure 4.6 represents the performance ratio and running time of the second efficient SBPF in comparison with the regular framework. As can be seen, the performance decreased slightly to 93% and 95% based on the weighted MAPE and weighted RMS errors, while the number of generated samples dropped by half. This experiment is performed for Ebola forecast at time point 3.

By tracing the history and evolution of the particles, we realized that selecting too few resampled particles (N_1) from a big pool of expanded particles ($N3$) may result in loosing the best particle in the resampling process. To resolve this issue, we ensure that the best particle of each iteration is among the selected ones. This would allow that particle to improve during the next iterations. Based on the observations for time point 5, pushing the best sample forward is most effective when the number of resampled particles is very low ($N_1 = 5$) (see Figure 4.7).

4.5 Summary of Results and Achievements

4.5.1 *Finding 1. The efficient Smart Beam-Particle Filter that incorporates all three policies to reduce the number of particles is able to successfully improve the running time. However, it compromises the performance to less than 50%.*

Support: As can be seen in Figures 4.5, applying all three proposed methods, reduces the average number of particles and consequently the running time of the efficient SBPF to 22%. The main cause of performance drop is utilizing the partial search in the smart director that leads to deleting promising neighbors of a particle.

4.5.2 *Finding 2. Reducing the number of resampled particles does not compromise the performance of SBPF significantly.*

Support: By decreasing the number of resampled particles to half for all SBPF iterations, the number of generated samples and running time dropped by half, while the performance was not compromised significantly (see Figure 4.6). Moreover, the experiments showed that selecting the best particle in the resampling process and pushing it forward to the next iteration could improve the performance, especially when the number of resampled particles (N_1) is very low in comparison with the number of expanded particles ($N3$). See Figure 4.7.

4.6 Conclusion

In this chapter, we performed a complexity analysis on the Smart Beam Particle Filter framework and presented three strategies to improve the running time of the system without compromising the performance. The time complexity of the disease propagation simulator, Epifast, changes linearly with the size of the contact network that comprises millions of nodes as the agents and millions of edges representing their connections. Having Epifast embedded in the SBPF framework imposes a huge overhead to the system. We proposed three methods to improve the time-complexity. These methods include decreasing the number of expanded, random, and resampled particles to reduce the number of Epifast runs. Through heuristic approaches, we generated and kept the promising samples, while filtering the rest. Experimental analysis demonstrates that not all of the discussed approaches are able to keep the performance as high as the original SBPF due to loosing good particles. Among the three proposed strategies, reducing the number of resampled particles shows the most promising results in terms of reducing the cost while retaining the high performance.

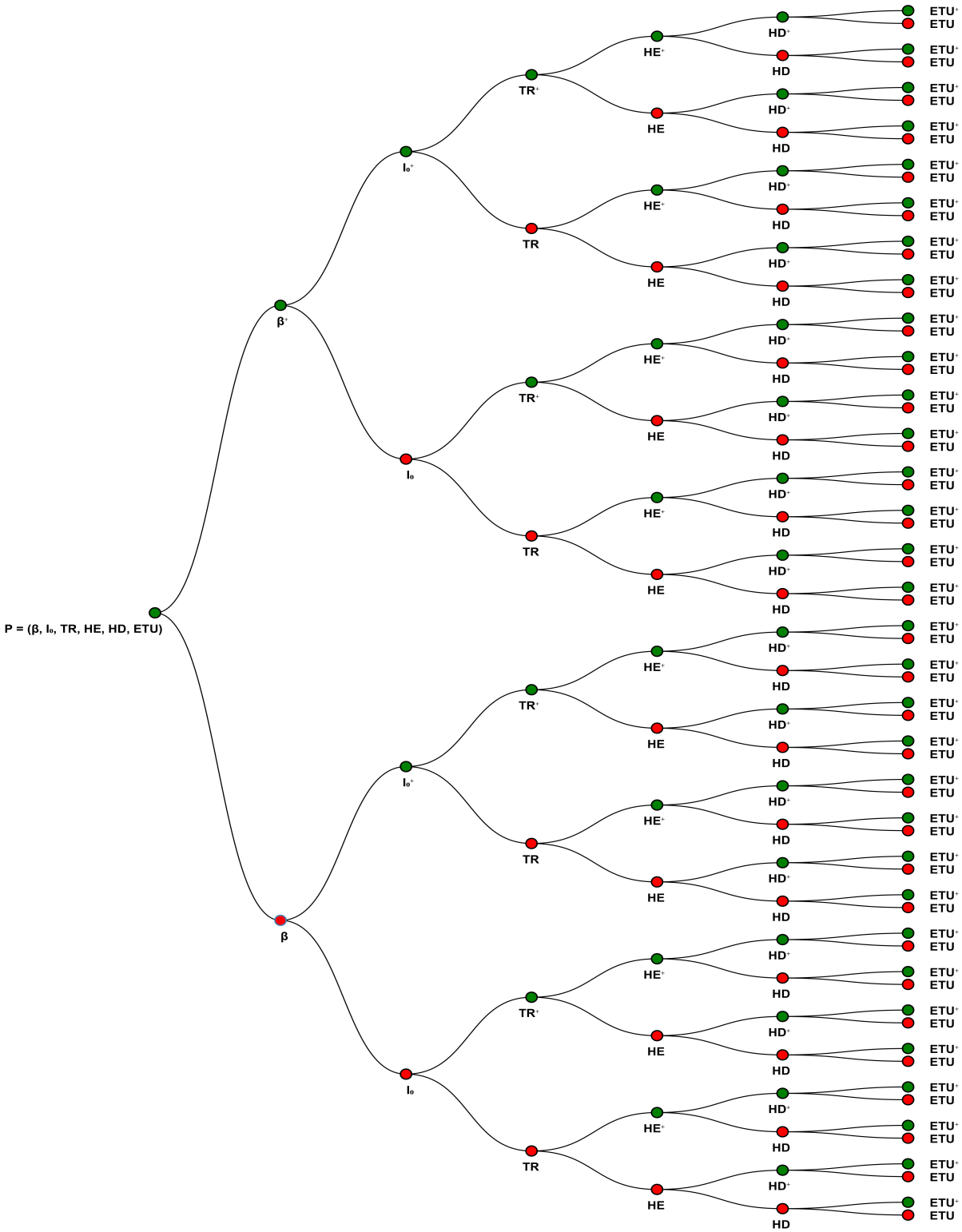


Figure 4.1. Perturbation Tree: Producing the neighbors of a particle by perturbing its parameters. The parameters belong to Ebola configurations: $P = (\beta, I_0, TR, HE, ETU)$. Number of neighbors: $2^6 = 64$.

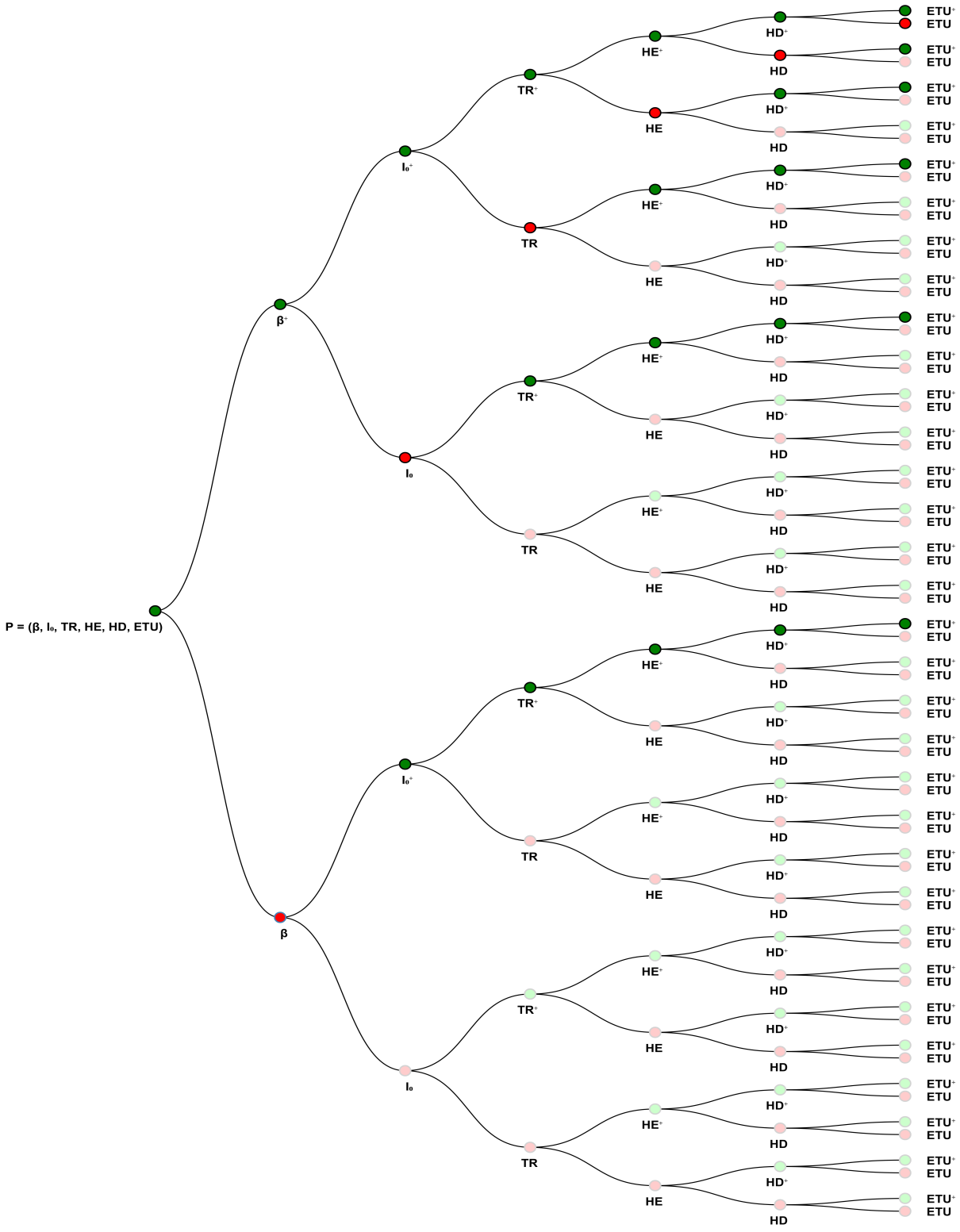


Figure 4.2. Perturbation Tree: Producing the neighbors of a particle by perturbing only 5 and 6 parameters out of 6 ones. Number of neighbors: $\binom{6}{5} + \binom{6}{6} = 6 + 1 = 7$

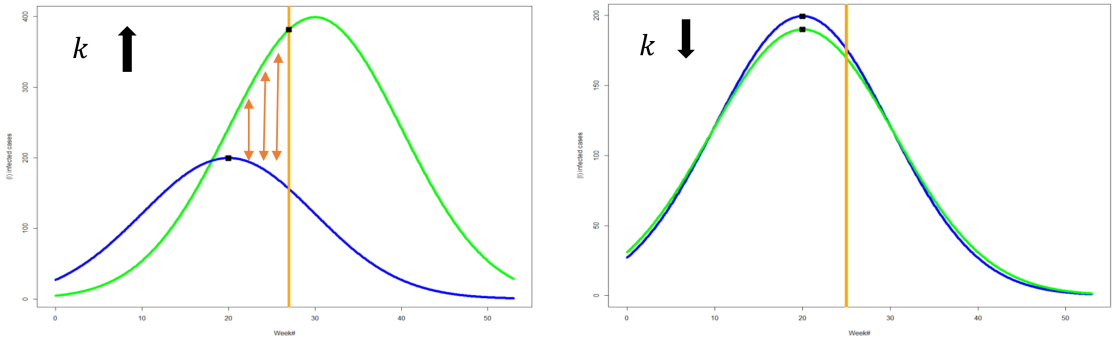


Figure 4.3. Choosing the value of k as the number of parameters to perturb: $k \propto$ prediction error.

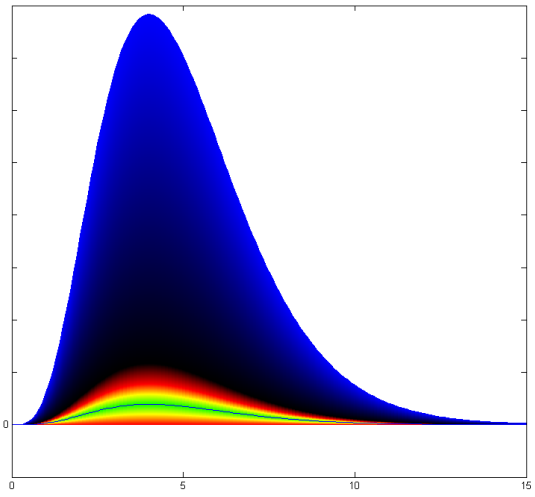


Figure 4.4. Colored Spectrum of the MAPE range.

Curves' Color	Range	Domain
	e=2	–
	e=1	x=0
	e=3/4	x=y/4
	e=1/2	$x = \frac{y}{2}$
	e=1/4	x=5y/4
	e=0	$x = y$
	e=1/4	x=3y/4
	e=1/2	$x = \frac{3y}{2}$
	e=3/4	x=7y/4
	e=1	$x = 2y$
	e=2	$x = 3y$
	$e \succ 2$	$x \succ 3y$
	$e \to \infty$	$x \to \infty$

Table 4.1. Corresponding domains and ranges of MAPE error

ALGORITHM 5: Efficient Smart Beam Particle Filter Algorithm.

Input: Probability distribution function $p(x_0)$: The prior knowledge about the unknown parameters of the state vector $x = (\theta_1, \theta_2, ..., \theta_{n_x})$; All observations received up to time $k-1$ including $y_{k-1} : Y_{k-1} = \{y_i, i = 1, ..., k-1\}$; N_1: Number of resampled particles from the adult set; $N_1^{'}$: Number of resampled particles from the young set; N_2: Number of initial particles; N_3: Number of adult expanded particles; $N_3^{'}$: Number of young expanded particles.

Output: $p(x_T^*|Y_T)$: The posterior function;

Set $S_A = \emptyset$; /* S_A : Collection of adult particles */

/* S_y : Collection of young particles */

$S_y = \{$Generate N_2 particles with age $= 0$ by sampling from prior pdf $p(x_0)\}$;

for $k = 0$ *to* T **do**

  if $k = 0$ **then** /* First Round: Does not receive any observations yet. */

    $S_0^* = \{x_0^{i*}\}_{i=1}^{N_2} = \{$Sample N_2 particles from prior $p(x_0)\}$

  else

    $S_{k-1}^* = \{x_{k-1}^{i*}\}_{i=1}^{N_1} = \{$resampled particles from posterior pdf $p(x_{k-1}|Y_{k-1}$) from previous step$\}$

  end

  /* Prediction Phase: Pass samples through the smart State Dynamic unit. */

  /* Perturb and expand previous particles to multiple ones: */

  $\forall x_{k-1}^i \in \{S_A \cup S_y\}$, *do* $x_k^i = f_{k-1}(x_{k-1}^{i*}, v_{k-1}^i)$;

  $\forall x_k^i \in S_A$, *do* $x_k^i.Age++$;

  $\forall x_k^i \in S_y$, do

    $x_k^i.Age++$;

    if $x_k^i.Age >$*threshold* **then**

      move x_k^i from S_y to S_A;

  $S_y \leftarrow S_y +$ {Generate 2^{RD-1} new random particles with age=0} /* */

  $S = S_A + S_y = \{x_{k-1}^i\}_{i=1}^{N_3+N_3^{'}} =$ samples from the prior pdf $p(x_k|Y_{k-1})$;

  /* Update Phase: Pass samples through the State Dynamic unit. */

  forall $x_k^i \in S_A$ **do**

    Give weight to particle x_k^i based on $\widetilde{w}_k^i = p(y_k|x_k^i)$;

    Normalized weights: $w_k^i = \widetilde{w}_k^i / \sum_{j=1}^{N_3} \widetilde{w}_k^j$;

  end

  forall $x_k^{'i} \in S_Y$ **do**

    Give weight to particle $x_k^{'i}$ based on $\widetilde{v}_k^i = p(y_k|x_k^{'i})$;

    Normalized weights: $v_k^i = \widetilde{v}_k^i / \sum_{j=1}^{N_3^{'}} \widetilde{v}_k^j$;

  end

  $S_k^* \leftarrow$Resample N_1 particles from S_A based on the normalized weights (w_k^i): $S_k^* = \{x_k^{i*}\}_{i=1}^{N_1}$;

  $S_k^{'*} = \leftarrow$Resample $N_1^{'}$ particles from S_A based on the normalized weights (v_k^i) : $S_k^{'*} = \{x_k^{'i*}\}_{i=1}^{N_1^{'}}$;

  $S_{k-total}^* = S_k^* + S_k^{'*}$;

  foreach $x_k^{i*} \in S_{k-total}^*$ **do**

    $\psi_k^i = (w_k^i \text{ or } v_k^i)/(\sum_{j=1}^{N_1} w_k^j + \sum_{l=1}^{N_1^{'}} v_k^l)$

  end

  Posterior is: $p(x_k^*|Y_k) = \sum_{i=1}^{N_1^*} \psi_k^i \times \delta(x_k - x_k^i)$, where $N_1^* = N_1 + N_1^{'}$;

  $N_1 = max(N_1$- -, $minThreshold(N_1))$; $N_1^{'} = max(N_1^{'}$- -,$minThreshold(N_1^{'}))$;

end

Figure 4.5. Performance vs. Running-Time: regular SBPF vs. efficient-SBPF 1.

Figure 4.6. Performance vs. Running-Time: regular SBPF vs. efficient-SBPF 2.

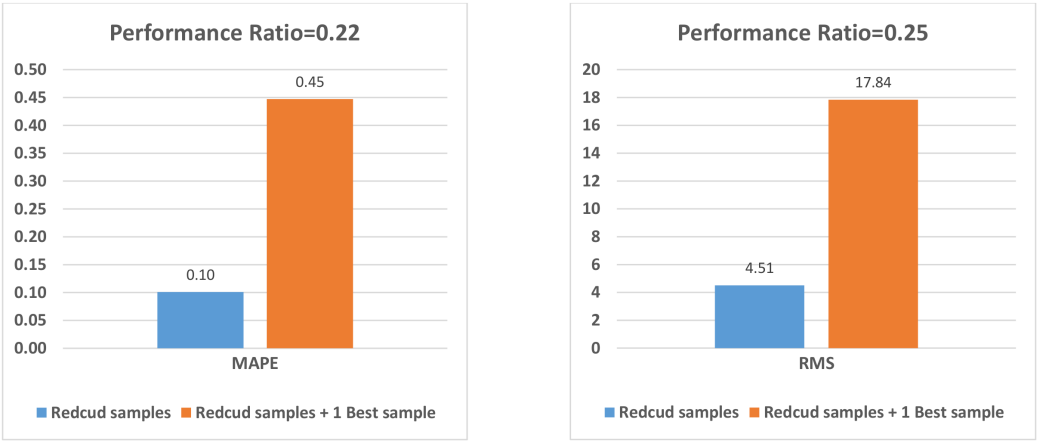


Figure 4.7. Performance of efficient-SBPF2 with and without forwarding the best sample in the resampling process.

Part II

Evaluating Epidemic Forecasts

Chapter 5

Framework for Evaluating Epidemic Forecasts: Deterministic Algorithms

5.1 Introduction

In recent years, numerous forecasting methods have been proposed in the field of epidemic forecasting. There have been a few studies on evaluating time-series forecasting methods and the lack of standard well-defined evaluation measures to select the best forecasting algorithm among different ones is still a significant problem. Armstrong [54], provides a thorough summary of the key principles that must be considered while evaluating time-series forecasting methods. To the best of our knowledge, at the time of writing this chapter, there have been no formal studies on comparing the standard epidemiologically relevant features across appropriate error measures for evaluating and comparing epidemic forecasting algorithms. Our work expands upon his philosophy of objective evaluation, with specific focus on the domain of epidemiology. In this chapter, we present an evaluation framework which allows for combining different features, error measures, and ranking schema to evaluate forecasts. We describe the various epidemic features (Epi-features) included to characterize the output of forecasting methods and provide suitable error measures that could be used to evaluate the accuracy of the methods with respect to these Epi-features. We focus on long-term deterministic predictions rather than short-term forecasting and demonstrate the utility of the framework by evaluating six forecasting methods for predicting influenza in the United States. In addition to quantifying the performance of each method, we also show how the framework allows for comparison among the methods by ranking them.

The rest of the chapter is organized as follows. Preliminaries and notations, that are used in the rest of this chapter, are given in Section 5.2. In Section 5.3, we provide the Epi-features we will use to characterize the features of an epidemic time-series. A collection of convenient error measures are discussed in Section 5.4. In Section 5.5, we present two

different ranking approach to compare multiple forecasting methods. In Section 5.6, we briefly describe the dataset used in this chapter. Experimental results showing the utility of the proposed evaluation framework are presented in Section 5.7. Section 5.8 provides the list of findings and achievements from our experimental results. Finally, we conclude the chapter in Section 5.9.

5.2 Preliminaries and Notations

In our evaluation framework (Figure 5.1), an epidemic forecast generated by a model/data-driven approach is quantified based on epidemiologically relevant features which we refer to as *Epi-features*. Further, the accuracy of a model's estimate of a particular Epi-feature is quantified by evaluating its error with respect to the Epi-features extracted from the ground truth. This is enabled by using functions that capture their dissimilarity, which we refer to as *error measures*. The evaluation framework, compares the methods by ranking them through the two ranking schema called *Consensus Ranking* and *Horizon Ranking*. In the rest of this chapter, the detailed description of all these three modules are included. The notations used in the following sections are listed in Table 5.1.

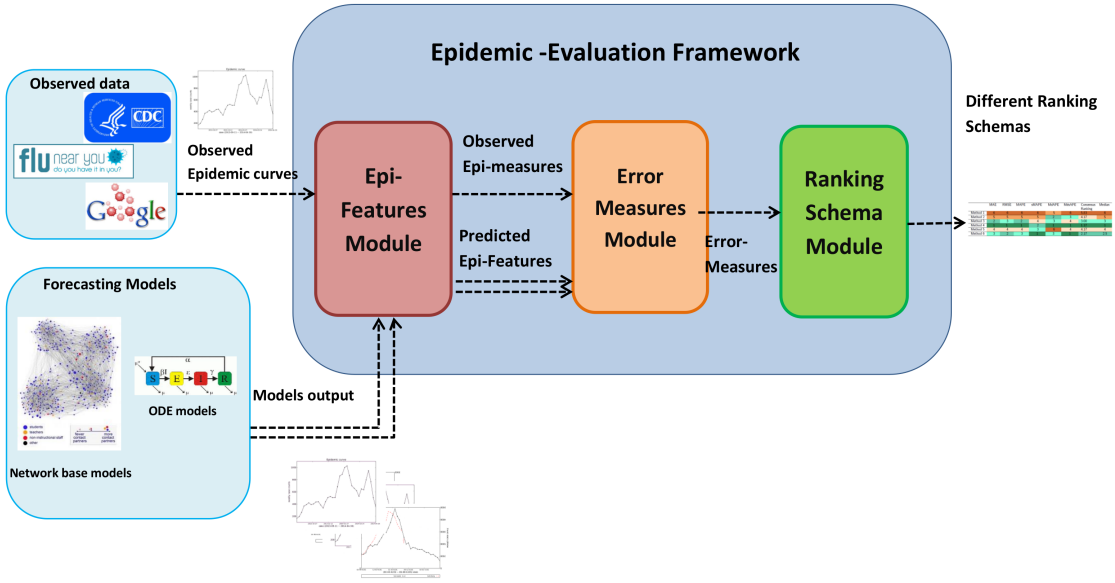


Figure 5.1. Epidemic Evaluation Framework included three modules: Epi-features module, Error Measure module, and Ranking schema.

Symbol	Definition
y_t *or* $y\,(t)$	number of new cases of disease in the t^{th} week observed in surveillance data
x_t *or* $x\,(t)$	number of new cases of disease in the t^{th} week predicted by forecasting methods
x_{start}	number of new cases of disease predicted at the start of epidemic season
x_{peak}	predicted value of the maximum number of new cases of the disease
e_t	$e_t = y_t - x_t$: the prediction error
T	duration of the epidemic season
$\bar{y}$	$\bar{y} = \frac{1}{T}\sum_{t=1}^{T}(y_t)$: the mean for y values over T weeks
σ^2	$\sigma^2 = \frac{1}{T-1}\sum_{t=1}^{T}(y_t - \bar{y})^2$: The variance of y values over T weeks
n_{tot}	Total number of infected persons during specified period
n_{ps}	The population size at the start of specified period
$n_{tot}\,(age)$	Total number of infected persons with specific age during the specified period
$n_{ps}\,(age)$	The population size with specific age at the start of specified period
n_c	or $n_{contacts}$ is the number of contacts of primary infected persons
n_{sg}	or $n_{second-generation}$ is the new number of infected persons among the contacts of primary infected individuals during a specified period
$GM\{Error\}$	$GM(e) = (\prod_{i=1}^{n}(e_i))^{(1/n)}$: Geometric Mean of a set of Errors
$M\{Error\}$	Arithmetic Mean of a set of Errors
$Md\{Error\}$	Median value of a set of Errors
$RMS\{Error\}$	Root Mean Square of a set of Errors

Table 5.1. Notation and Symbols

5.3 Epidemiologically Relevant Features

In this section, we list the Epi-features we will use to characterize the features of an epidemic time-series. While some of these features are generic and applicable to any time-series, the others are specific to epidemiology. Table 5.2 lists the brief definition of Epi-features.

Peak Value & Time

Peak value is the highest value in a time-series. In the epidemic context, it refers to the highest number of newly infected individuals at any given week during an epidemic season. Closely associated with peak value is peak time, which is the week in which the peak value is attained. Predicting these values accurately helps healthcare providers with resource planning.

First-Take-Off (Value & Time):

Seasonal outbreaks, like the flu, usually remain dormant and exhibit a sharp rise in the number of cases just as the season commences. A similar phenomenon of sharp increase is exhibited by emerging infectious diseases. The early detection of "first-take-off" time will help the authorities alert the public and raise awareness. Mathematically, it is the time at which the first derivative of the epidemic curve exceeds a specific threshold. Since the epidemic curve is discretized in weekly increments, the approximate slope of the curve over Δt time steps is defined as follows:

$$s(x, \Delta t) = \frac{x(t + \Delta t) - x(t)}{\Delta t} \tag{5.1}$$

where x is the number of new infected case-counts and t indicates the week number. In our experiment, we set $\Delta t = 2$. The value of $s(x, \Delta t)$ is the slope of the curve and shows the take-off-value while the start time of the take-off indicates the take-off-time. The threshold used in calculating the first-take-off depends on the type of the disease and how aggressive and dangerous the outbreak could be. The epidemiologists determine the threshold value and is also based on the geographic area. In this case, we set the threshold to 150.

Intensity Duration

Intensity Duration (ID) indicates the number of weeks, usually consecutive, where the number of new infected case counts is greater than a specific threshold. This feature can be used by hospitals to estimate the number of weeks for which the epidemic will stress their resources (Figure 5.2).

Speed of Epidemic

The Speed of Epidemic (SpE) indicates how fast the infected case counts reach the peak value. This feature includes peak value and peak time simultaneously. The following equation shows the definition of speed of epidemic:

Epi-feature name	**Definition**
Peak value	Maximum number of new infected cases in a given week in the epidemic time-series
Peak time	The week when peak value is attained
Total attack rate	Fraction of individuals ever infected in the whole population
Age-specific attack rate	Fraction of individuals ever infected belonging to a specific age window
First-take-off-(value):	Sharp increase in the number of new infected case counts over a few consecutive weeks
First-take-off-(time):	The start time of sudden increase in the number of new infected case counts
Intensity duration	The number of weeks (usually consecutive) where the number of new infected case counts is more than a specific threshold
Speed of epidemic	The rate at which the case counts approach the peak value
Start-time of disease season	The time at which the fraction of infected individuals exceeds a specific threshold

Table 5.2. Definitions of different Epidemiologically Relevant features

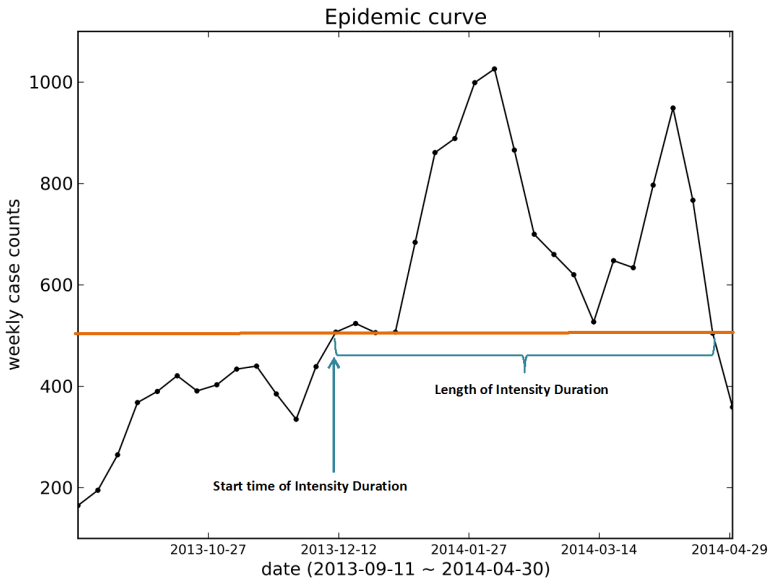


Figure 5.2. Intensity Duration: Intensity Duration's length (ID) indicates the number of weeks where the number of new infected case counts are more than a specific threshold.

$$SpE = \frac{x_{peak} - x_{start}}{t_{peak} - t_{start}} \tag{5.2}$$

where x_{peak} and x_{start} are the number of new case count diseases at peak time and the start time of the season, respectively. In other words, speed of epidemic is the steepness of the line that connects the start data-point of time-series sequence to the peak data-point(Figure 5.3).

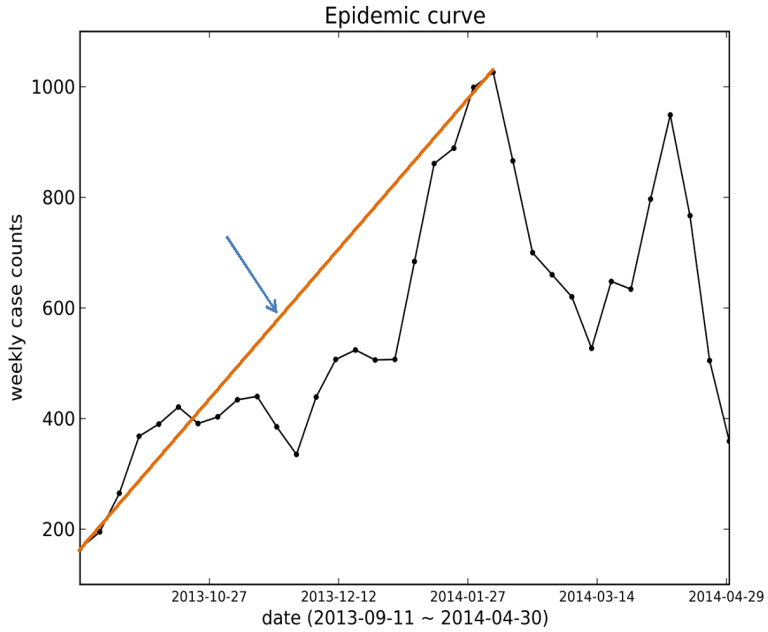


Figure 5.3. Figure explaining Speed of Epidemic: Speed of Epidemic (SpE) is the steepness of the line that connects the start data-point of time-series sequence to the peak data-point. SpE indicates how fast the infected case counts reach the peak value.

Total Attack Rate (TAR):

Attack rate (TAR) is the ratio of the total number of infected cases during a specified period, usually one season, to the size of the whole population at the start of the period.

$$TAR = \frac{n_{tot}}{n_{ps}} \tag{5.3}$$

where n_t is the total number of infected people during specified period.

Age-Specific Attack Rate (Age-AR)

This is similar to the total attack rate but focuses on a specific sub-population. Specific attack rate is not only limited to age-specific attack rate, but the sub-population could be restricted by any feature like age, gender, or any special group.

$$AgeAR\,(age) = \frac{n_{tot}\,(age)}{n_{ps}\,(age)} \tag{5.4}$$

Secondary Attack Rate (SAR):

Secondary attack rate (SAR) means the ratio of new infected cases of a disease, during a particular period, among the contacts of primary cases who are infected first; in other words, it is a measure of the spreading of disease in the contact network.

$$SAR = \frac{n_{sg}}{n_c} \tag{5.5}$$

where n_c is the number of contacts of primary infected persons and n_{sg} is the number of infected persons among those contacts during a specified period[89]. In order to calculate the secondary attack rate, individual information about households and their contact networks are needed. Epidemiologists estimated the secondary attack rate in household contacts of several states in the U.S. to be 18% to 19% for acute-respiratory-illness (ARI) and 8% to 12% for influenza-like-illness (ILI)[90].

Start-Time of a Disease Season

We define the "Start-time of a flu season" as the week when the flu-percentage exceeds a specified threshold. The flu-percentage is defined as follows:

$$Per\,(Flu) = \frac{n_i\,(Flu)}{n_i\,(All)} \tag{5.6}$$

where $n_i\,(Flu)$ is weekly influenza related illnesses in i^{th} week and $n_i\,(All)$ is the weekly number of all patients including non-ILI ones seen by health providers for any reason and/or all specimens tested by clinical laboratories. The value of threshold that is used as the criteria is determined by the epidemiologist and could be calculated in different ways. We define the threshold by analyzing the past flu seasons based on the flu baseline definition given by the CDC [91]. The CDC defines the baseline as the mean percentage of visits for influenza during non-influenza weeks for the previous three seasons plus two standard deviations [91]. The non-influenza weeks are defined as two or more consecutive weeks in which the number of counted ILI diagnoses for each week is less than 2% of total seasonal ILI case counts. The definition of start-of-season could be generalized for any disease such as Ebola, Zika, etc.

5.4 Error Measures

The second step of evaluating epidemic forecasting algorithms is to measure the error for each predicted Epi-feature. There are a variety of measures that can be used to assess the error between the predicted time-series and the observed one. The error measures that we consider in this study are listed in Table 5.3 along with their features. The notations used in the error measure equations are described in Table 5.1. Note that all the error measures considered only handle the absolute value of the error. They do not distinguish between under and over-estimation of the time-series. The signed versions of some of these absolute error measures are listed in the Appendix. These signed measures include the direction of error (i.e. the positive sign demonstrates the underestimation while the negative one indicates overestimation). Moreover, all the measures referred to in Table 5.3 use Arithmetic Mean to get an average value of the error. Variants that use geometric mean, median, etc. are listed in Table B.2 in the Appendix B.

Measure name	Formula	Description	Scaled	Outlier Protection	Other forms
Mean Absolute Error (MAE)	$MAE = \frac{1}{T}\sum_{t=1}^{T}\lvert e_t\rvert$	Demonstrates the magnitude of overall error	No	Not Good	GMAE
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{t=1}^{T} e_t^2}{T}}$	Root square of average squared error	No	Not Good	MSE
Mean Absolute Percentage Error (MAPE)	$MAPE = \frac{1}{T}\sum_{t=1}^{T}\lvert\frac{e_t}{y_t}\rvert$	Measures the average of the absolute percentage error	Yes	Not Good	MdAPE[a], RMSPE[b]
symmetric Mean Absolute Percentage Error (sMAPE)	$sMAPE = \frac{2}{T}\sum_{t=1}^{T}\lvert\frac{e_t}{y_t+x_t}\rvert$	Scale the error by dividing it by the average of y_t and x_t	Yes	Good	MdsAPE
Mean Absolute Relative Error (MARE)	$MARE = \frac{1}{T}\sum_{t=1}^{T}\lvert\frac{e_t}{e_{RWt}}\rvert$	Measures the average ratio of absolute error to Random walk error	Yes	Fair	MdRAE, GMRAE
Relative Measures: e.g. RelMAE (RMAE)	$RMAE = \frac{MAE}{MAE_{RW}} = \frac{\sum_{t=1}^{T}\lvert e_t\rvert}{\sum_{t=1}^{T}\lvert e_{RWt}\rvert}$	Ratio of accumulation of errors to cumulative error of Random Walk method	Yes	Not Good	RelRMSE, LMR [92] , RGRMSE [93]
Mean Absolute Scaled Error (MASE)	$MASE = \frac{1}{T}\sum_{t=1}^{T}\lvert\frac{e_t}{\frac{1}{T-1}\times\sum_{i=2}^{T}\lvert y_i - y_{i-1}\rvert}\rvert$	Measures the average ratio of error to average error of one-step Random Walk method	Yes	Fair	RMSSE
Percent Better (PB)	$PB = \frac{1}{T}\sum_{t=1}^{T}[I\{e_t, e_{RWt}\}]$ $\lvert e_{s,t}\rvert \leq \lvert e_{RWt}\rvert \leftrightarrow I\{e_t, e_{RWt}\} = 1$	Demonstrates average number of times that method overcomes the Random Walk method	Yes	Good	-
Mean Arctangent Absolute Percentage Error (MAAPE)	$MAAPE = \frac{1}{T}\sum_{t=1}^{T} arctan\lvert\frac{e_t}{y_t}\rvert$	Calculates the average arctangent of absolute percentage error	Yes	Good	MdAAPE
Normalized Mean Squared Error (NMSE)	$NMSE = \frac{MSE}{\sigma^2} = \frac{1}{\sigma^2 T}\sum_{t=1}^{T} e_t^2$	Normalized version of MSE: value of error is balanced	No	Not Good	NA

Table 5.3. List of main Error Measures . Arithmetic mean and absolute errors are used to calculate these measures in which positive and negative deviations do not cancel each other out and measures do not provide any information about the direction of errors.

[a]Md represent Median

After careful consideration, we selected MAE, RMSE, MAPE, sMAPE, MdAPE and MdsAPE as the error measures for evaluating the Epi-features and ignored others based on different reasons. We list our reasons and observations on the eliminated error measures in part B of Supplementary Information. Also, instead of using MAPE, we suggest corrected MAPE (cMAPE) to solve the problem of division by zero:

$$cMAPE = \begin{cases} \frac{1}{T}\sum_{t=1}^{T} |\frac{e_t}{y_t}|, & \text{if } y_t \neq 0 \\ \frac{1}{T}\sum_{t=1}^{T} |\frac{e_t}{y_t+\epsilon}|, & \text{otherwise} \end{cases} \tag{5.7}$$

where ϵ is a small value. It could be equal to the lowest non-zero value of observed data. We have also added two error measures based on the median: Median Absolute Percentage Error (MdAPE) and Median symmetric Absolute Percentage Error (MdsAPE). However, as median errors have low sensitivity to change in methods, we do not recommend them for isolated use as the selection or calibration criteria.

5.5 Ranking Methods

The third step of the evaluation process is ranking different methods based on different Epi-features and the result of different error measures. For this purpose, we have used two kinds of ranking methods: Consensus Ranking and Horizon Ranking.

- **Consensus Ranking**: Consensus Ranking (CR) for each method is defined as the average ranking of the method among others. This kind of Consensus Ranking could be defined in different scenarios. For example, the average ranking that is used in Table 5.5 in the Result section is Consensus Ranking of a method based on one specific Epi-feature integrated across different error measures.

$$CR_{EM}^{m} = \frac{1}{n_{EM}} \sum_{i=1}^{n_{EM}} |R_{i,m}| \tag{5.8}$$

  where $R_{i,m}$ is the individual ranking assigned to method m among other methods for predicting one Epi-feature based on error measure i, n_{EM} is the number of error measures, and Consensus Ranking CR_{EM}^{m} is the overall ranking of method m based on different error measures.

  Consensus Ranking could also be defined across different Epi-features. In this case, CR over error measures could be considered as the individual ranking of a method, and the average is calculated over different Epi-features. It is important to consider the variance of ranking and the intensity of quartiles besides the mean value of CR.

In the Results section we demonstrate how to process and analyze these rankings in a meaningful way.

- **Horizon Ranking**: While Consensus Ranking considers the average performance of methods over prediction times, Horizon Ranking demonstrates the performance trend of various forecasting methods in predicting a single Epi-feature across different prediction times. First, for each Epi-feature, we compute an error measure like Absolute Percentage Error (APE) or its symmetric variant (sAPE) per prediction time. For each prediction time, APE values of different forecasting methods are sorted from smallest to largest to determine the ranking of the methods. The average value of this ranking over different error measures determines the overall Horizon Ranking of the methods in each time-step.

5.6 Data

The ILI surveillance data used in this chapter was obtained from the website of the United States Centers for Disease Control and Prevention (CDC). The information of patient visits to health care providers and hospitals for ILI was collected through the US Outpatient Influenza-like Illness Surveillance Network since 1997 and lagged by two weeks(ILINet)[94, 95]; this Network covers all 50 states, Puerto Rico, the District of Columbia and the U.S. Virgin Islands. The weekly data are separately provided for 10 regions of HHS regions [1] that cover all of the US. The forecasting algorithms have been applied to CDC data for each HSS region. We applied our forecasting algorithm on the 2013-2014 flu season data where every season is less than or equal to one year and contains one major epidemic. Figure 5.4 shows the HHS Region Map that assigned US states to the regions.

5.7 Experimental Results

Past literature in the area of forecasting provides an overall evaluation for assessing the performance of the predictive algorithm by defining a statistical distance/similarity function to measure the closeness of the predicted epidemic curve to the observed epidemic curve. However, they rarely evaluate the robustness of a method's performance across epidemic features of interest and error measures. Although the focus of the paper is not on a specific method, it is instructive to observe the functionality of the software framework in action applied on the sample methods.

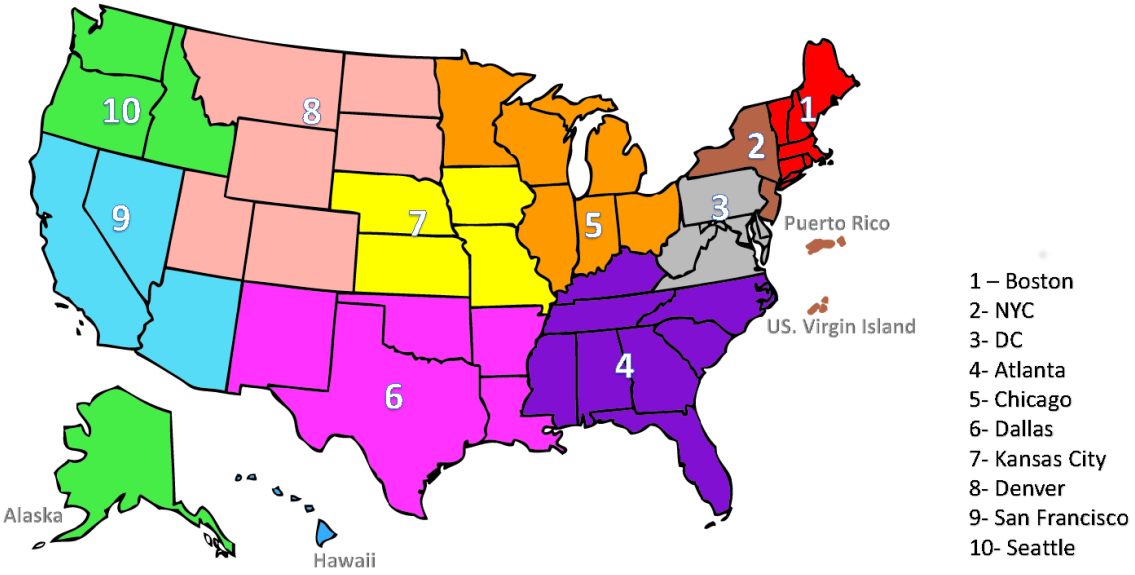


Figure 5.4. HHS region map based on "U.S. Department of Health & Human Services" division[1]

5.7.1 Rankings Based on Error Measures Applied to Peak Value

In Table 5.4, we calculated six error measures, MAE, RMSE, MAPE, sMAPE, MdAPE, and MdsAPE for the peak value predicted by six different forecasting methods. The corresponding ranks are provided in the Ranking Table (Table 5.5). The most successful method is assigned rank 1 (R1); As can be seen, even similar measures like MAPE and sMAPE do not behave the same for the ranking process. The fourth algorithm wins six first places among other methods for seven error measures and shows almost the best performance. However, it is hard to come to a similar conclusion for other methods. The last column in the table is Consensus Ranking, which shows the average ranking of the method over different error measures. Figure 5.5 shows the Box-Whisker diagram of method rankings. Note that, Methods 2 and 5 despite having identical Consensus Ranking, have different interquartile ranges, which represent Method 5 as a more reliable approach. Based on such analysis, the fourth method (M4) is the superior for predicting the peak value. After that, the order of performance for other methods will be: Method 6 (M6), Method 3, Method 5, Method 2 and Method 1. Note however, this analysis is specific to using peak value as the Epi-feature of interest.

	MAE	RMSE	MAPE	sMAPE	MdAPE	MdsAPE
Method 1	4992.0	9838.6	4.9	1.04	1.7	1.03
Method 2	4825.2	9770.4	4.7	0.99	1.4	0.95
Method 3	3263.0	5146.5	3.2	0.96	1.5	1.01
Method 4	2990.7	4651.3	2.9	0.899	1.1	0.85
Method 5	3523.2	5334.8	3.4	0.95	2.1	1.01
Method 6	3310.9	4948.5	3.2	0.896	1.5	0.85

Table 5.4. Different errors for predicting peak value for Region 1 over whole season (2013-2014).

	MAE	RMSE	MAPE	sMAPE	MdAPE	MdsAPE	Consensus Ranking	Median
Method 1	6	6	6	6	5	6	5.83	6
Method 2	5	5	5	5	2	3	4.17	5
Method 3	2	3	2	4	3	4	3.00	3
Method 4	1	1	1	2	1	1	1.17	1
Method 5	4	4	4	3	6	4	4.17	4
Method 6	3	2	3	1	3	1	2.17	2.5

Table 5.5. Ranking of methods for predicting peak value based on different error measures for Region 1 over whole season (2013-2014). The color spectrum demonstrates different ranking levels. Dark green represents the best rank, whereas dark orange represents the worst one.

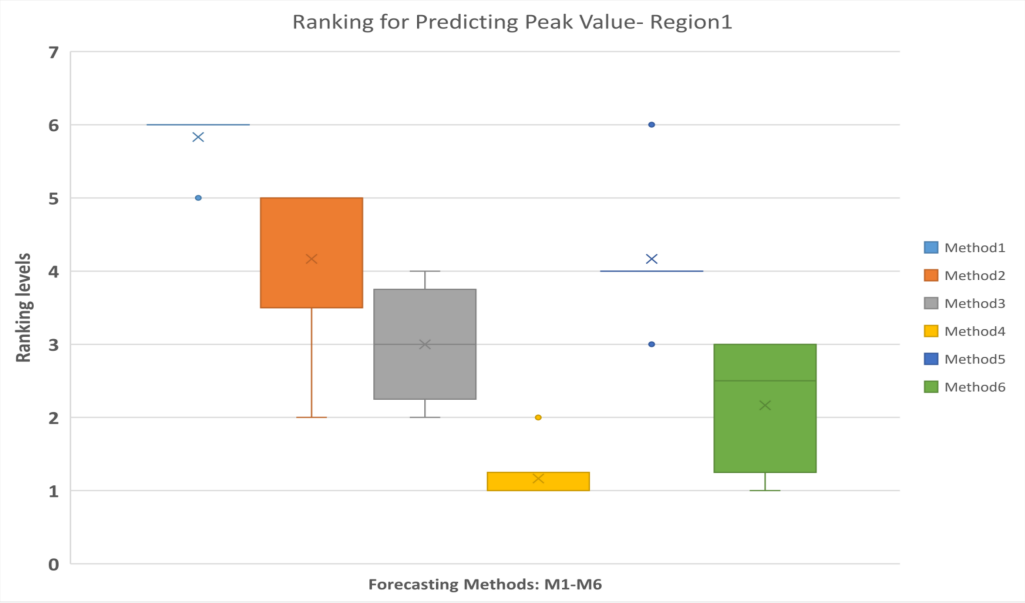


Figure 5.5. Box-Whisker Plot shows the Consensus Ranking of forecasting methods in predicting **Peak value** for Region 1, aggregated on different error measures.

	Peak value	Peak time	Take-off-value	Take-off-time	ID length	ID start time	Start of flu season	Speed of Epidemic	Average	Median
M1	5.83	3.83	6	1	3.33	5.67	6	5.83	4.69	5.67
M2	4.17	4.5	5	2	1	4.33	5.0	4.5	3.81	4.33
M3	3	2.83	3.83	3	3.33	3.17	3	3.17	3.17	3.17
M4	1.17	3.33	1.17	5	4.00	1.0	1	1.17	2.23	1.17
M5	4.17	1.17	3	4	4.33	4.67	3	4.17	3.56	4
M6	2.17	2.33	1.50	6	4.67	2.00	1.00	1.67	2.67	2.17

Table 5.6. Average Consensus Ranking over different error measures for all Epi-features- Region 1.

5.7.2 Consensus Ranking Across All Epi-features

In order to make a comprehensive comparison, we have calculated the error measures on the following Epi-features: Peak value and time, Take-off-value and Take-off-time, Intensity Duration's length and start time, Speed of epidemic, and start of flu season. We do not include demographic-specific Epi-features, such as age-specific attack rate or secondary attack rate, since such information is not available for our methods.

Figure 5.6 shows the Consensus Ranking of the methods in predicting different Epi-features for Region 1. Note that Method 4, which is superior in predicting some Epi-features such as Peak value and start of Flu season, is worse than other methods in predicting other Epi-features such as Take-off time and Intensity Duration. The tables corresponding to the box-plots are included in Appendix B.3.

Figure 5.7 shows the second level of Consensus Ranking over various Epi-features for Region 1. This figure summarizes the performance of different methods based on the average Consensus Rankings that are listed in Table 5.6. It is evident that Method 1, Method2, and Method 5 have similar performance, while the third method performs moderately well across Epi-features. Method 4, which performs best for five out of eight Epi-features, is not among the top three methods for predicting Take-off time and Intensity Duration. Method 6 comes in as the second best method when considering the Consensus Ranking.

Figures 5.8 and 5.9 represent the second level of Consensus Rankings of the six approaches over all Epi-features for regions 1 to 10. Often, experts need to select one method as the best predictor for all regions, hence we propose the third level of Consensus Ranking to aggregate the results across different regions. Figure 5.10 represents the Consensus Ranking over all 10 HHS regions, based on the average of Consensus Rankings across all Epi-features for each region listed in Table 5.7 . As can be seen in Figure 5.10, the performance of the first and the second methods are behind the other approaches and we can exclude them from the pool of selected algorithms. However, the other four methods show very competitive performance and are considered the same according to the total rankings. The sequential aggregations provide a general conclusion which eliminates the nuances of similar methods.

	Region1	Region2	Region3	Region4	Region5	Region6	Region7	Region8	Region9	Region10	Ave
M1	4.69	3.31	4.6	3.94	3.65	2.21	4.3	3.94	3.46	4.29	3.84
M2	3.81	2.77	4.23	4.0	3.71	1.29	3.73	3.69	3.79	3.96	3.50
M3	3.17	3.46	1.96	2.68	2.67	2.21	3.03	2.73	2.17	2.33	2.64
M4	2.23	3.19	2.04	2.7	3.08	1.29	2.93	2.60	2.44	3.71	2.62
M5	3.56	1.79	1.79	2.41	2.77	2.21	2.67	3.06	2.88	2.67	2.58
M6	2.67	3.23	2.13	2.48	2.83	1.29	2.60	3.27	3.13	3.58	2.72

Table 5.7. Average Consensus Ranking of methods over different Epi-features- Regions 1 - 10.

5.7.3 Horizon Rankings for Each Epi-feature

Horizon Ranking helps track the change in accuracy and ranking of the methods over prediction time. Higher fluctuations in the Horizon Ranking across the time steps, hints at the unsuitability of Consensus Ranking as selection criteria for the best method. It is possible that the method that performs best during early stages of prediction may not perform the best at later time-points. Figure 5.11 shows the evolution of Horizon Ranking of the six methods for predicting the peak value calculated based on APE and sAPE. As shown in Figure 5.6, Methods 4 and 6 have the best average Consensus Ranking in predicting peak value and is consistent with observations on Horizon Ranking. In Figure 5.11 the ranking of Methods 4 and 6 demonstrates a little fluctuation at the first time-steps. However, as prediction time moves forward these methods provide more accurate forecasts causing them to rank higher. The most interesting case for Horizon Rankings concerns the prediction of peak time. The Consensus Ranking in Figure 5.6 selects Method 5 as superior in predicting peak time and Methods 6 and 4 as the second and third best approaches. However, by observing the trends of ranks over prediction times (Figure 5.12), Methods 4 and 6 are dominant for the first eight weeks of prediction, then Method 1 wins the first place for seven weeks. In the next eight weeks, Methods 1, 3, and 5 are superiors simultaneously.

Figure 5.13, 5.14, 5.15 show Horizon Ranking graphs for leveraging forecasting methods in predicting other Epi-features. These Horizon Rankings are almost consistent with their corresponding Consensus Rankings which confirms the best methods from the Consensus Ranking perspective could be used for any prediction time.

5.7.4 Visual Comparison of Forecasting Methods

In order to visualize the output of forecasting methods, we generate the one-step-ahead epidemic curve. Given the early time series up to time k $(y(1), ..., y(k))$ as observed data, the forecasting algorithm predicts the next data point of time series $x(k+1)$ and this process is repeated for all values of prediction time k where $t_b \leq k \leq t_e$. By putting together the short-term predictions, we construct a timeseries from t_b to t_e as a one-step-ahead predicted epidemic curve. Figure 5.16 depicts the one-step-ahead predicted epidemic-curves for HHS region 1 that are generated by the six forecasting methods (refer to figures S11-S19 in Sup-

plementary Information for other Regions). We used $t_b = 2$ and $t_e = T - 1$ as the beginning and end for the prediction time. As can be seen in figure 5.16, the first and second methods show bigger deviations from the observed curve, especially in the first half of the season. As these six methods are different configurations of one algorithm, their outputs are competitive and sometimes similar to each other. Methods 3 and 5, and Methods 4 and 6 show some similarity in their one-step-ahead epidemic curve that is consistent with Horizon Ranking charts for various Epi-features. However, Horizon Ranking graphs contain more information regarding long-term predictions; therefore, the ranking methods, especially Horizon Ranking, could help experts to distinguish better methods when the outputs of forecasting methods are competitive and judgment based on the visual graph is not straightforward.

5.8 Summary of Results and Achievements

5.8.1 *Finding 1. None of the forecasting algorithms may outperform the others in predicting all Epi-features. Different levels of Consensus Ranking across multiple error measures, and multiple Epi-features are recommended to obtain a comprehensive comparison.*

As expected, the performance of a particular method depends on the Epi-features and error measures of choice. Our experimental results demonstrate that some algorithms perform well with regard to one Epi-feature, but fall short with respect to the others (Figure 5.6). It is possible that none of the forecasting algorithms dominates others in every Epi-feature and error measure. As a single Epi-feature cannot describe all attributes of a forecasting algorithm's output, all of them should be considered in the ranking process to obtain a comprehensive comparison. We suggest aggregation of multiple error measures , and multiple Epi-features in the ranking procedure, respectively. Figures 5.5 and 5.7 depict various levels of Consensus Ranking to support this claim.

5.8.2 *Finding 2. Horizon Ranking should be used with Consensus Ranking as a supplement. While Consensus Ranking provides an overall perspective for the method rankings, Horizon Ranking dispenses a comparative evaluation of the performance of the methods over time.*

Horizon Ranking scrutinizes the detailed facts about the rankings over time-steps that are concealed in the aggregation process of Consensus Rankings. If the Horizon Ranking fluctu-

ates excessively over the time steps, that would give lower credit to the average Consensus Ranking as selection criteria for the best method. Based on experimental results of Horizon Ranking, it is noticed that for a single Epi-feature, one method may show the best performance in early stages of the prediction, whereas another algorithm is the dominant in other time intervals (Figures 5.11 and 5.12). Finding patterns in Horizon Ranking plots help in selecting the most appropriate method for different forecasting periods. We suggest identifying a few time intervals in which the Horizon Rankings of the best methods are consistent. Then, in each time interval, the best method based on Horizon Ranking could be selected, or the Consensus Ranking could be calculated for each period by calculating the average errors (error measures) over time steps. The superior method for each time interval is the one with first Consensus Ranking in that period.
The other advantage of Horizon Ranking is to detect and reduce the effect of outliers *across time horizons*, whereas Consensus Ranking aggregates the errors across time steps that results in a noticeable change in total value of error measures by outliers. On the other hand, the aggregation of errors in Consensus Ranking is not always a disadvantage, because sometimes a slight difference in errors could change the Horizon Ranking level while the Consensus Ranking accumulates the errors for a whole time-series which gives an overall perspective of each method's performance.

5.8.3 *Finding 3. While sMAPE looks like a symmetric error measure and is claimed to be unbiased in other literature, we believe MAPE has a symmetric domain that provides the unbiased error.*

MAPE and sMAPE have been the two important error measures in assessing forecast errors since 1993. MAPE was used as the primary measure in M2-Competition, and it was replaced by sMAPE in M3-Competition to overcome the disadvantages of MAPE. One of the drawbacks is that MAPE could get a large or undefined value when the observed data point gets close to zero. This is alleviated to some extent by using the average of observed and predicted value in the denominator for sMAPE. The other issue that has been claimed for MAPE in some literature is biasing in favor of small forecasts. Therefore, critics believe that MAPE leads to a higher penalty for large overestimation rather than any underestimation. sMAPE, as the symmetric version of MAPE, normalized the error value with the mean of predicted and observed data which limits the range of sMAPE error between 0 and 2 for both overestimation and underestimation of the prediction. However, we believe that although the range of sMAPE function is symmetric, it does not provide a uniform scoring of the errors. We believe sMAPE is significantly biased toward large forecasts. Figure 5.17 and Table B.10 in the Appendix B.3 demonstrate the corresponding domains that generate equal MAPE or sMAPE errors in term of magnitude. The figures in the left column belong to MAPE and the right ones are sMAPE's. In figure 5.17, the black line represents the observed

epidemic curve (y), and the horizontal axis is the weekly time steps (t). The yellow borders show the predicted curves as overestimated or underestimated predictions which both result in MAPE= 0.5 or sMAPE = 0.5. The green spectrum shows the predicted curves with low values of MAPE or sMAPE. Equal colors in these figures correspond to equal values for the discussed error measure. The red borders in the left graph belong to predicted curves $x(t) = 2 \times y(t)$ and $x(t) = 0 \times y(t)$ with MAPE = 1 and the red borders in the right chart correspond to $x(t) = 3 \times y(t)$ and $x(t) = (1/3) \times y(t)$ which generate sMAPE = 1. As can be seen, MAPE grows faster than sMAPE which means MAPE reaches 1 with smaller values in the domain. Moreover, MAPE demonstrates symmetrical growth around the observed curve that results in fair scoring toward over and underestimation.

The black borders in the lower charts of figure 5.17 correspond to the predicted epidemic curve which generates MAPE=2 and sMAPE =2 in the left and right charts sequentially. The color spectrum of sMAPE in the right chart represents the non-symmetric feature of this error measure which is in favor of large predictions. As we could not show the infinity domain for sMAPE, we limited it to the predicted curve $x(t) = 20 \times y(t)$. Figure 5.18 shows the blue spectrum of MAPE that corresponds to large predictions where $x(t) >> 3y(t)$ and MAPE approaches infinity. This error measure provides more sensible scoring for both calibration and selection problems.

5.9 Conclusion and Future work

Evaluating epidemic forecasts arising from varied models is inherently challenging due to the wide variety of epidemic features and error measures to choose from. In this chapter, we proposed different Epi-features for quantifying the prediction accuracy of forecasting methods and demonstrated how suitable error measures could be applied to those Epi-features to evaluate the accuracy and error of prediction. We have applied the proposed Epi-features and error measures on the output of six forecasting methods to assess their performance. As the experimental results showed, different error measures provide various measurements of the error for a single Epi-feature. Therefore, we provided the Consensus Ranking method to aggregate the rankings across error measures and summarize the performance of forecasting algorithms in predicting a single Epi-feature. Based on the first round of rankings, none of the forecasting algorithms could outperform the others in predicting all Epi-features. Therefore, we recommended the second set of rankings to accumulate the analysis for various Epi-features and provide a total summary of the forecasting method capabilities. We also proposed Horizon Ranking to trace the performance of algorithms across the time steps to provide better perspective over time.
Choosing the best forecasting method enables policy planners to make more reliable recommendations. Understanding the practical relevance of various Epi-features of interest, and the properties offered by different error measures, will help guide the method selection. We hope that our work allows for a more informed conversation and decision process while using

and evaluating epidemic forecasts. The error measures discussed in this chapter, primarily deal with deterministic forecasts. However, numerous stochastic forecasting algorithms with some levels of uncertainty have been studied in the literature, and an evaluation framework for stochastic methods is a prime requirement in this area. Moreover, because of possible errors in measurements and sources of information, the observed data could be stochastic as well. In next chapter, we are going to extend our proposed measures and provide new methods to handle the evaluation of stochastic forecasts.

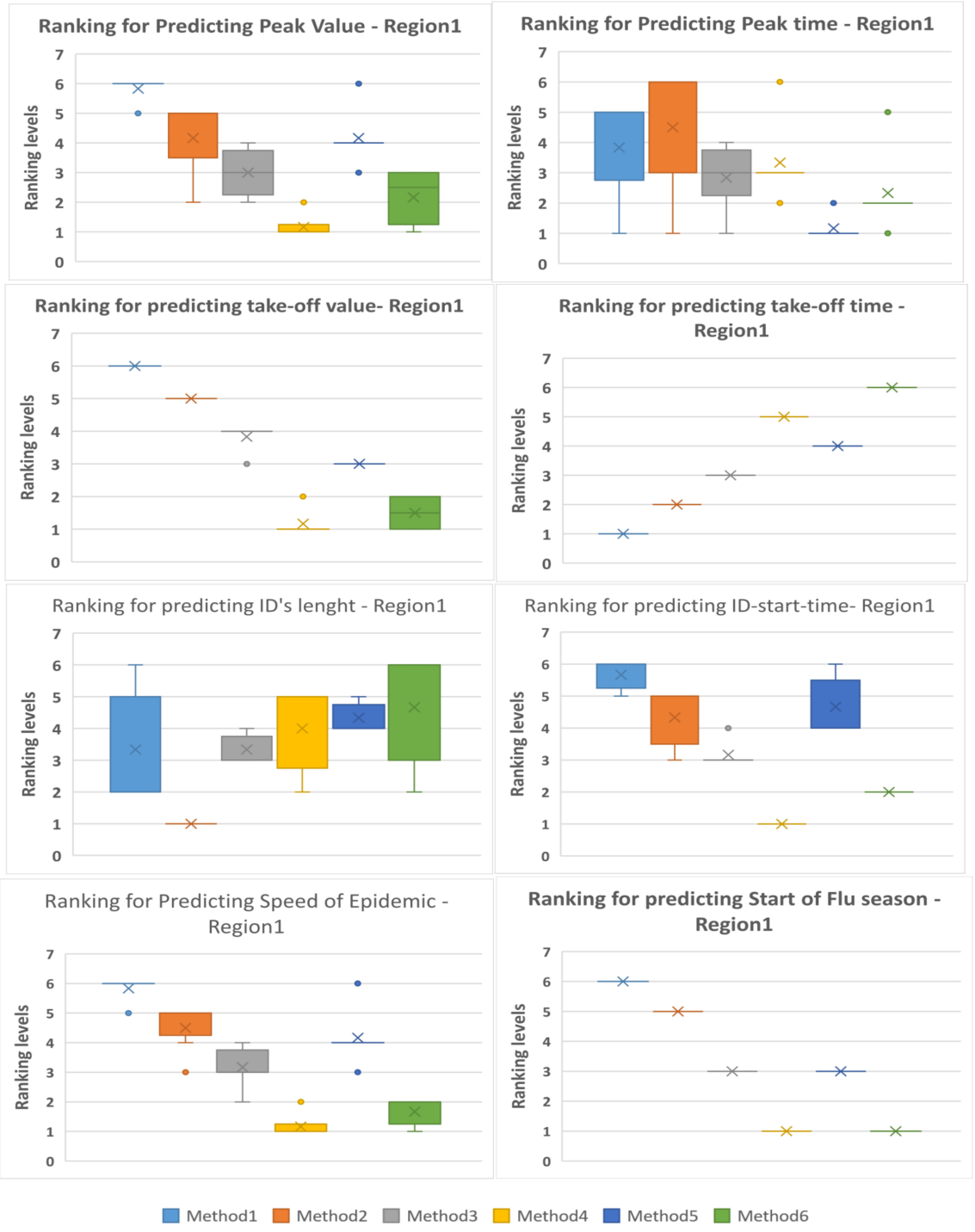


Figure 5.6. Consensus Ranking of forecasting methods over all error measures for predicting different Epi-features for Region 1. Method 4 is superior in predicting five Epi-features out of eight,but is far behind other methods in predicting three other Epi-features.

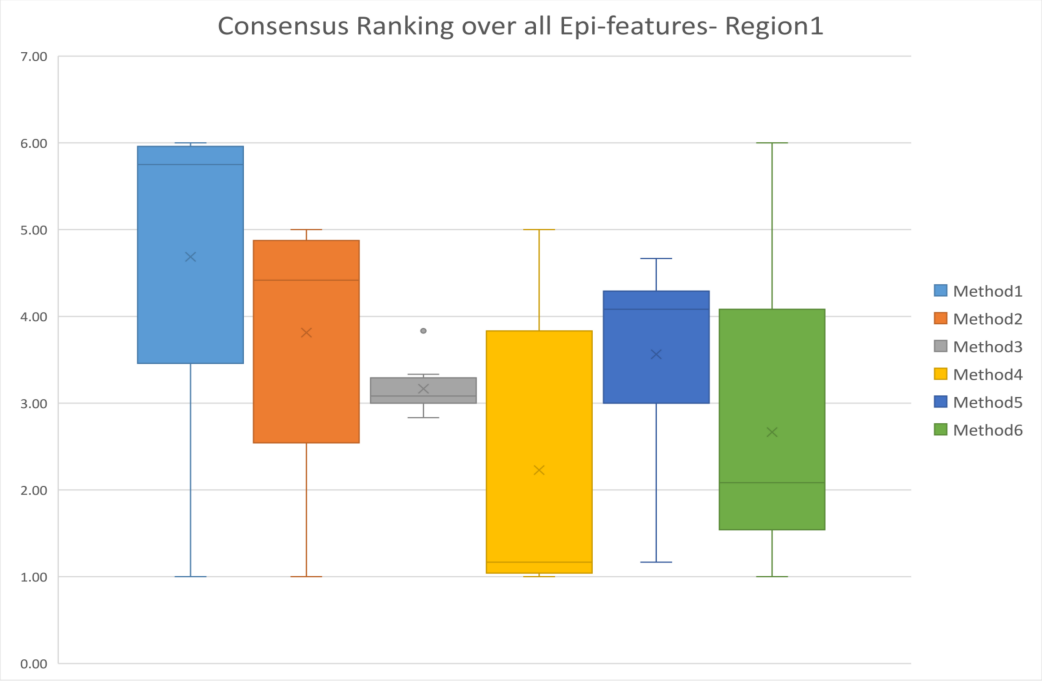


Figure 5.7. The box-whisker diagrams shows the median, mean and the variance of Consensus Ranking of methods over all Epi-features for Region 1.

Figure 5.8. Consensus Ranking over all Epi-Features - Regions 1-6. The box-whisker diagrams show the median, mean and the variance of Consensus Ranking of methods in predicting different Epi-features.

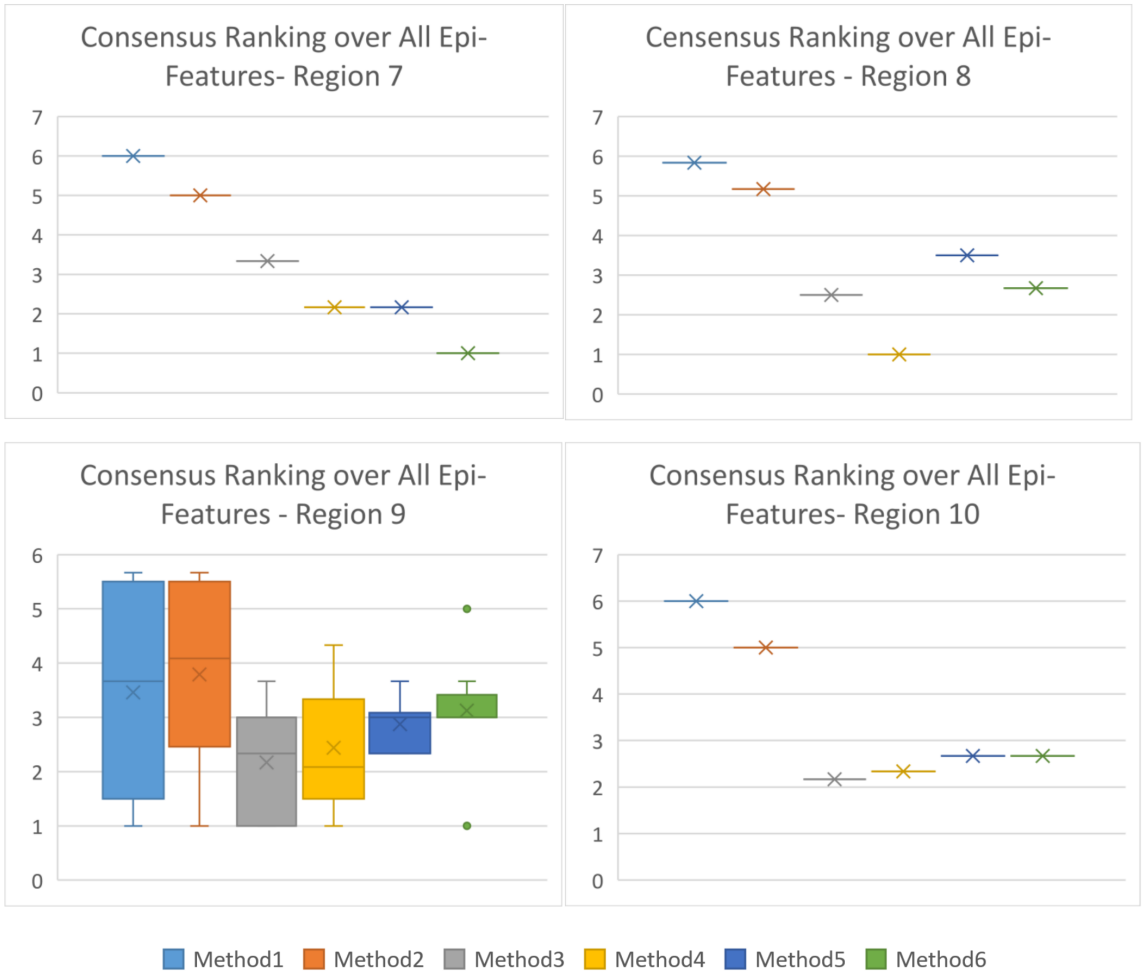


Figure 5.9. Consensus Ranking over all Epi-Features- Regions 7-10: The box-whisker diagrams show the median, mean, and quantiles of Consensus Ranking of methods in predicting different Epi-features.

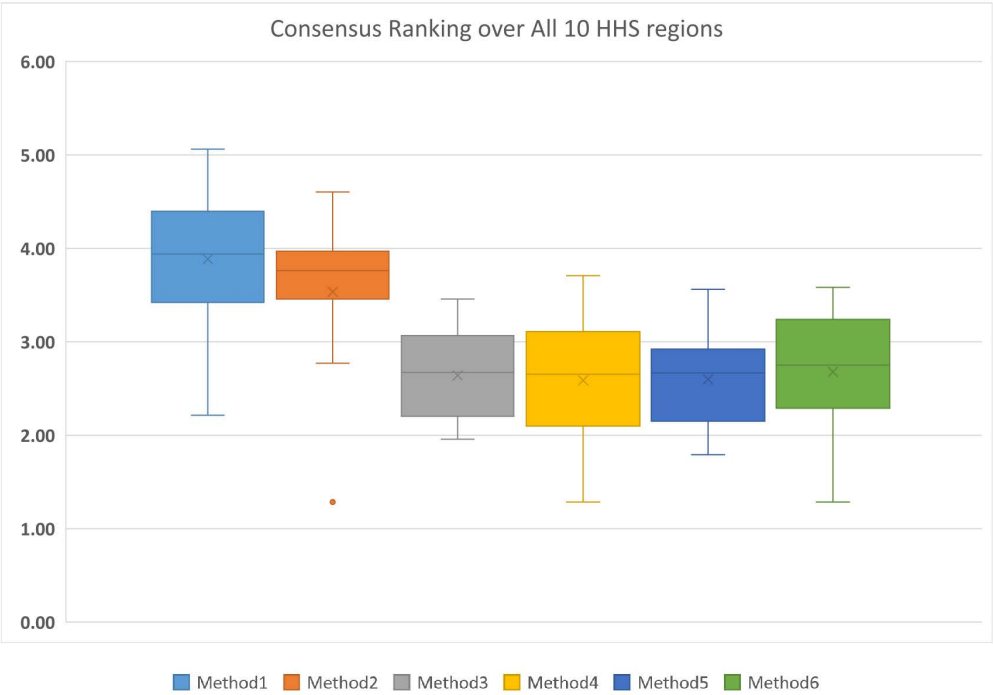


Figure 5.10. Consensus Ranking over all 10 HHS-Regions: The box-whisker diagrams show the median, mean, and quantiles of Consensus Ranking of methods in predicting the Epi-features for all HHS regions.

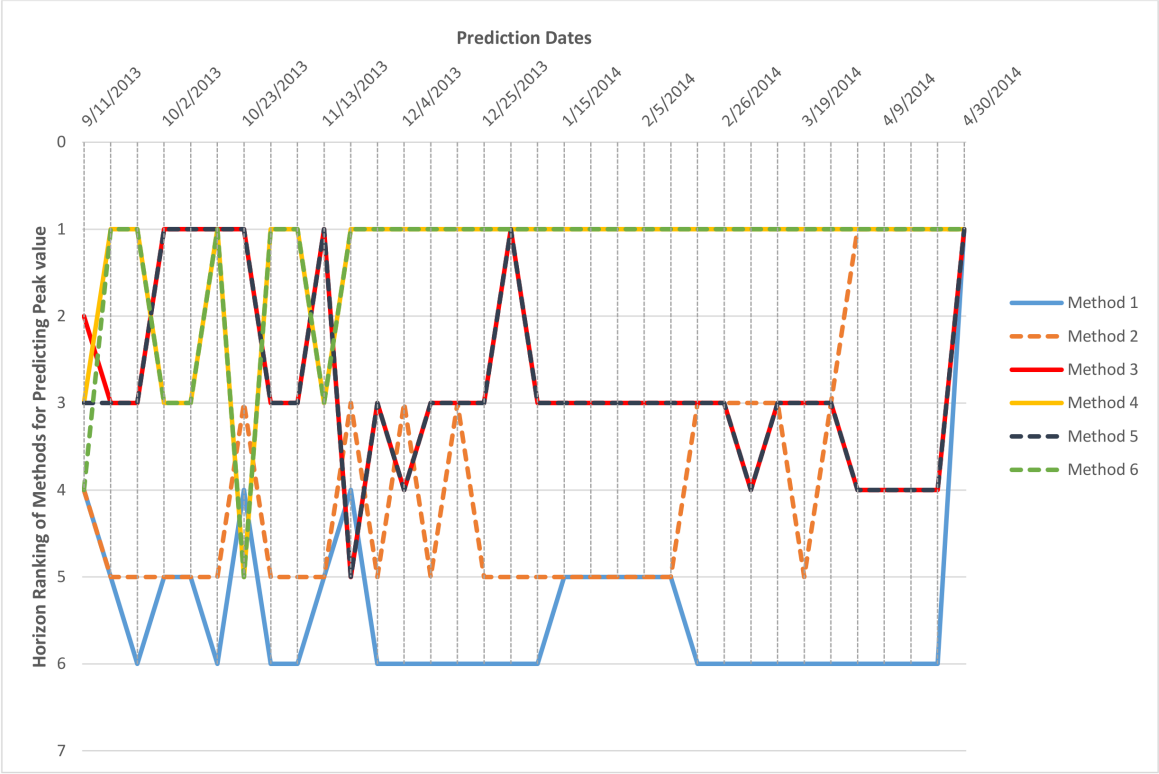


Figure 5.11. Horizon Ranking of six methods for predicting the peak value calculated based on APE, and sAPE, on Region 1.

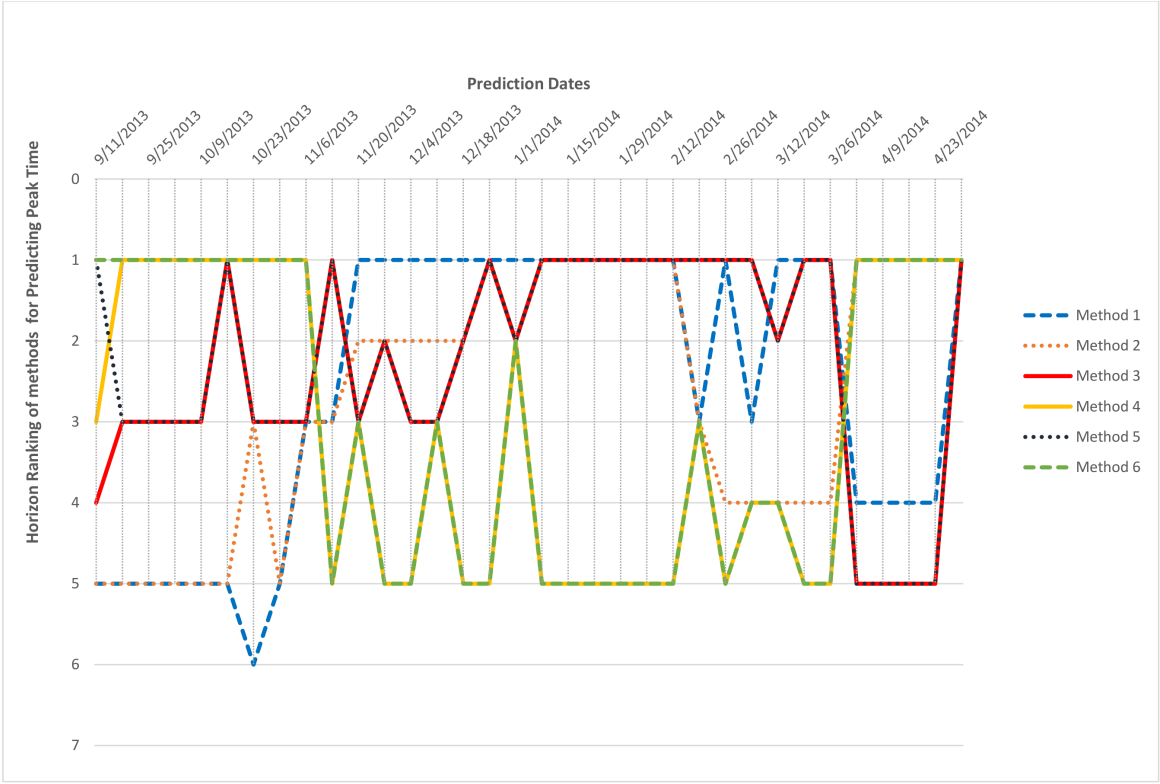


Figure 5.12. Horizon Ranking of six methods for predicting the peak time calculated based on APE, and sAPE, on Region 1. Methods 4 and 6 are the dominant for the first eight weeks of prediction, and then method 1 wins the first place for seven weeks. In the next eight weeks, methods 1, 3, and 5 are superiors simultaneously.

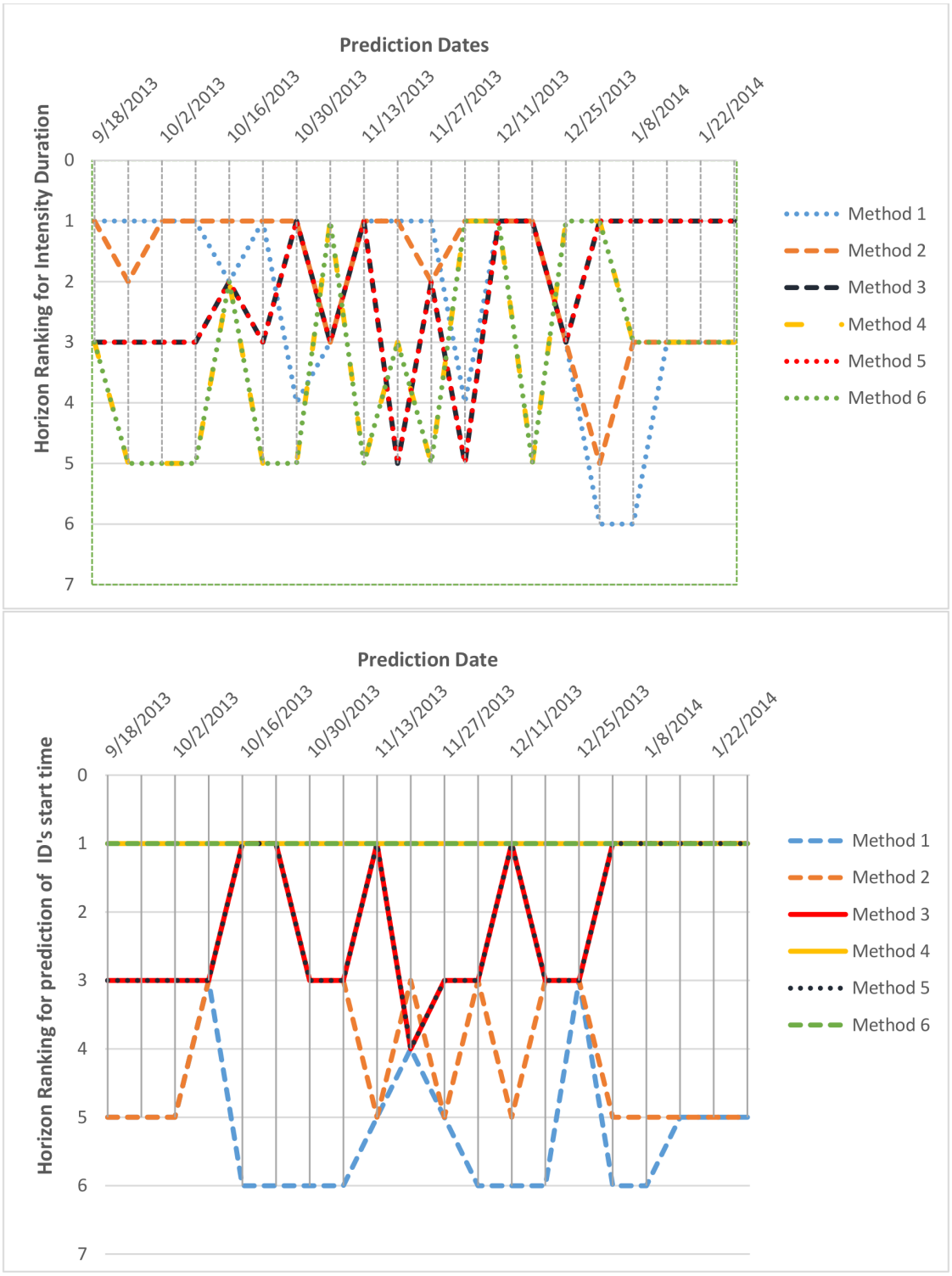


Figure 5.13. Horizon Ranking of six methods for predicting the Intensity Duration length and start time calculated based on APE, and sAPE, on Region 1.

Figure 5.14. Horizon Ranking of six methods for predicting the Take-off value and time calculated based on APE, and sAPE, on Region 1.

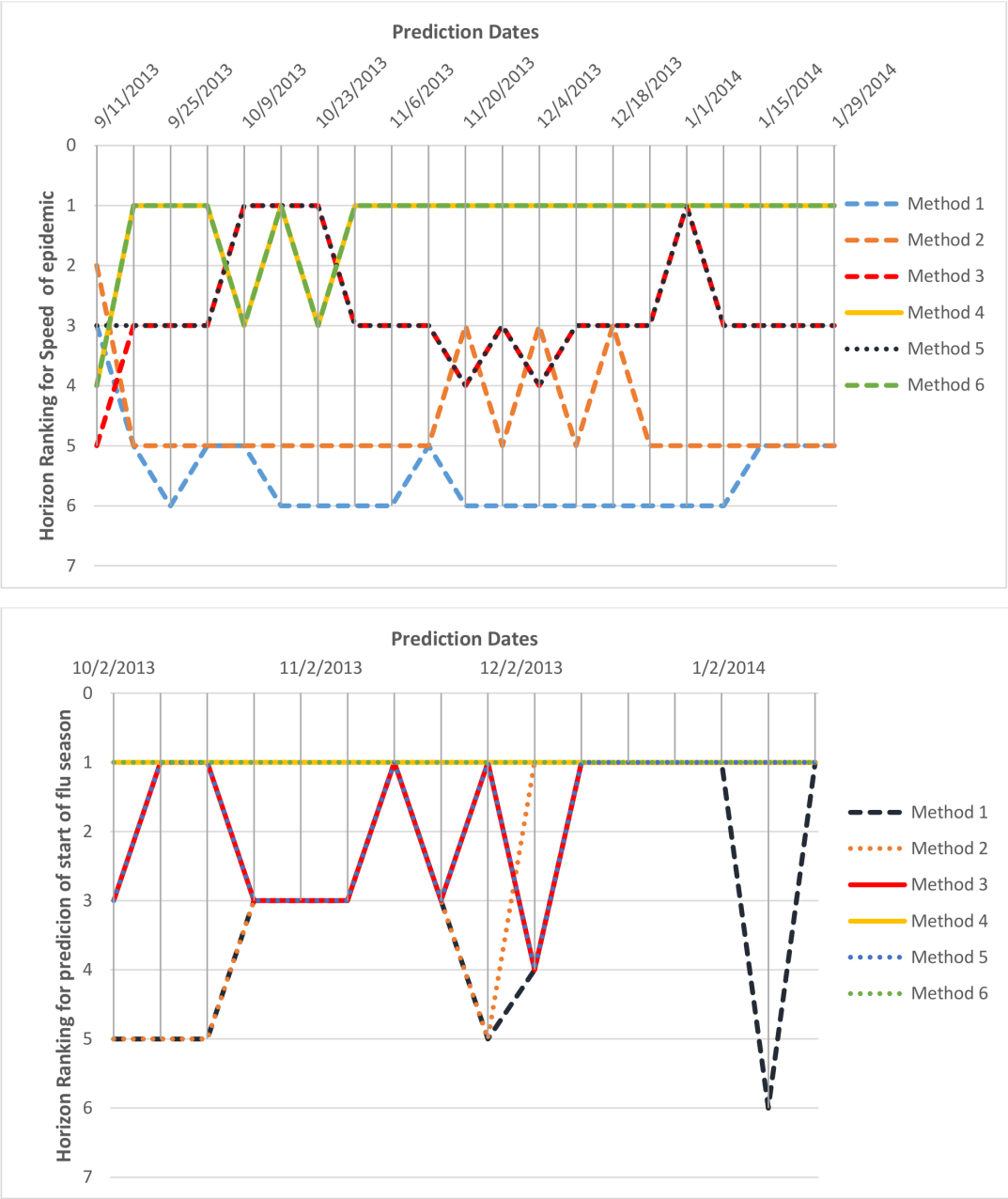


Figure 5.15. Horizon Ranking graphs for leveraging forecasting methods in predicting Speed of Epidemic and Start of flu season, on Region 1.

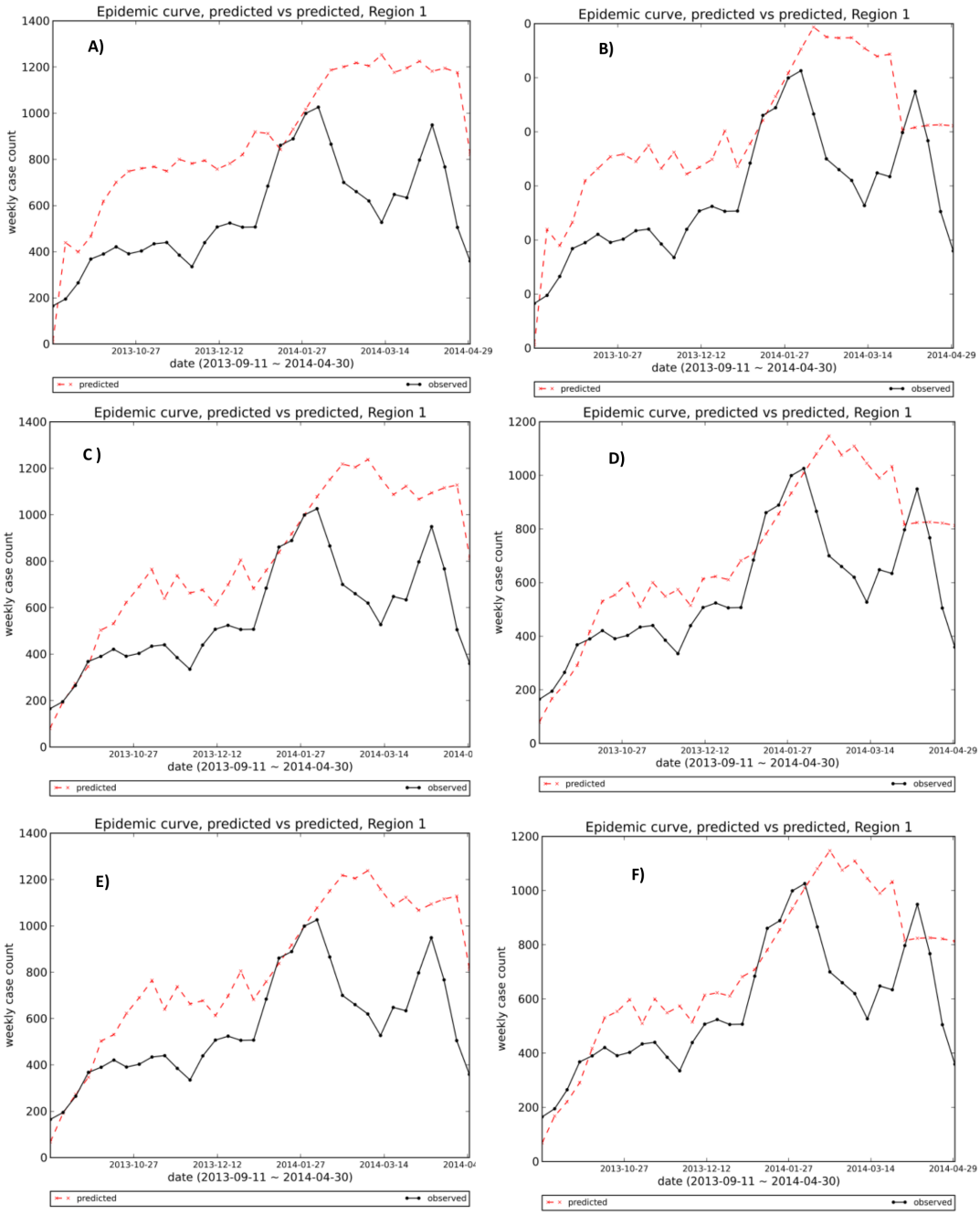


Figure 5.16. Visual comparison of one-step-ahead predicted curves generated by six methods vs. the observed curve, Region 1: The first and second methods show bigger deviations from observed curve, especially in the first half of the season. As the six methods are different configurations of one algorithm, their outputs are so competitive and sometimes similar to each other; methods 3 and 5, and methods 4 and 6 show some similarity in their one-step-ahead epidemic curve that is consistent with Horizon Ranking charts for various Epi-features.

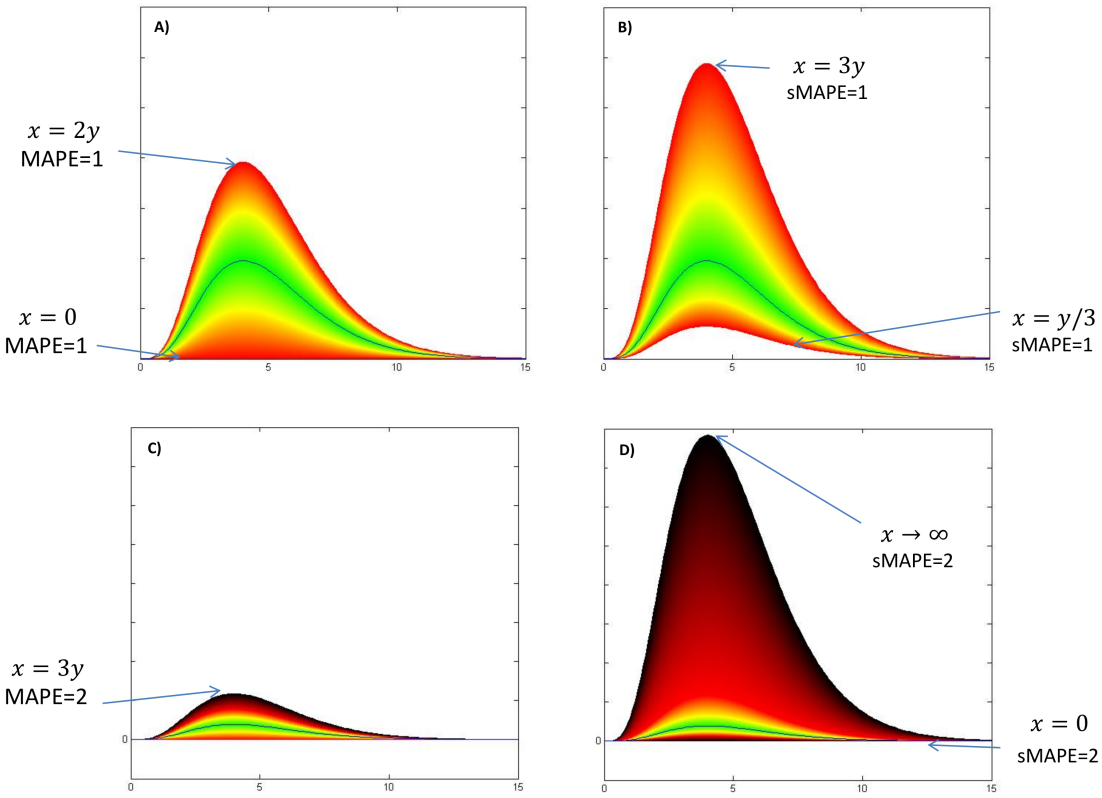


Figure 5.17. Comparison of MAPE and sMAPE domains and ranges spectrum: Red borders in the left graph (A) belong to predicted curves $x(t) = 2 \times y(t)$ and $x(t) = 0 \times y(t)$ with MAPE = 1 and the red borders in the right chart (B) corresponds to $x(t) = 3 \times y(t)$ and $x(t) = (1/3) \times y(t)$ which generate sMAPE = 1. The black borders in graphs C & D are corresponding to predicted epidemic curves which generates MAPE=2 and sMAPE =2 in the left and right charts sequentially.

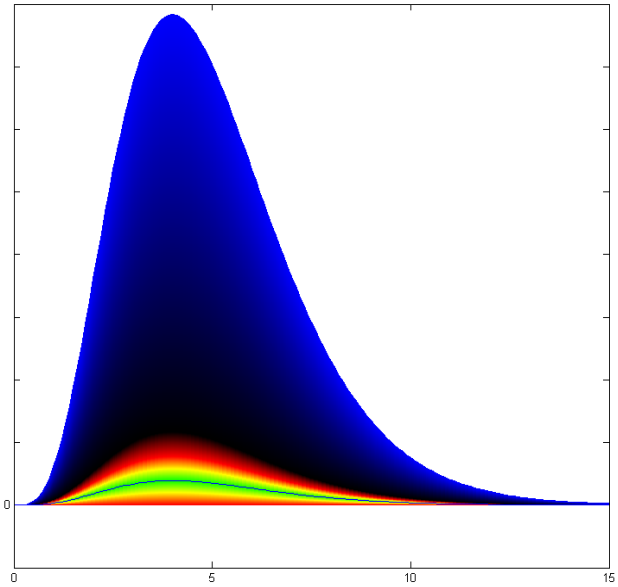


Figure 5.18. Colored Spectrum of MAPE range: MAPE does not have any limitation from the upper side that results in eliminating the large overestimated forecasting.

Chapter 6

Framework for Evaluating Epidemic Forecasts: stochastic Algorithms

6.1 Introduction

The error measures discussed in Chapter 5 primarily deal with deterministic forecasts. However, numerous stochastic forecasting algorithms with some levels of uncertainty have been studied in the literature [39, 40, 46, 41, 47, 48, 42, 45]. Therefore, an evaluation framework for stochastic methods is a prime requirement in this area. Moreover, because of possible errors in measurements and sources of information, the observed data could be stochastic as well. In this chapter, we extend our proposed measures in Chapter 5 and provide new methods to handle the evaluation of stochastic forecasts. we also provide some tips about how to deal with stochastic observations.

The rest of the chapter is organized as follows. Preliminaries and notations, required for this chapter, are given in Section 6.2. In Section 6.3, we present how to generate Epi-features and error measures for the stochastic algorithms with multiple replicates of time-series as their output. In Section 6.4, we extend the evaluation measures and metrics to be able to handle stochastic algorithms that generate time-series of mean and variance of the predicted values. Finally, we conclude the chapter in Section 6.5.

6.2 Preliminaries and Notations

Table 6.1 lists the notations that are used in rest of this chapter. These notation are the complement of those previously defined in preceding chapters.

The output of stochastic forecasting algorithms could be in one of the following formats:

Symbol	Definition
X	Random variable X (or X_t) that is the predicted estimate of a data point at one week(t^{th} week)
$f(x)\|f_x$	Probability density function (pdf) of random variable X
μ_x	Mean value for the random variable X
$\sigma_x = \sigma/\sqrt{N_x}$	Standard deviation for the random variable X
$\overline{x}$	Mean value of the samples belonging to random variable X
σ	Standard deviation of the samples belonging to random variable X
v	$v = N_x - 1$ Degree of freedom of t-distribution
$\bar{y}$	$\bar{y} = \frac{1}{n}\sum_{t=1}^{n}(y_t)$: the mean for y values over n weeks
$S_x = \{s_i\}$	where s_i is the sample from distribution f_x
$N_{s_x} = \|S_x\|$	Number of sample set S_x
Y	Random variable Y (or Y_t) that is the estimate of observed value at one week(t^{th} week)
$g(y)\|g_y$	Probability density function (pdf) of random variable Y
$S_y = \{s_j\}$	where s_j is the sample from distribution g_x

Table 6.1. Notation Table II

i) Multiple replicates of the time-series (Figure 6.1-a)
ii) A time-series of mean and variance of the predicted values (Figure 6.1-b)
In this chapter we divide the problem into multiple sub-problems to conquer the difficulty of the major challenge.

6.3 Stochastic Forecasts as Multiple Replicates

Most of the stochastic algorithms generate multiple replicates of series and/or state vectors to generate the posterior density function by aggregating discrete values together. A state vector contains the parameters that are used by the epidemic model to generate the epidemic curve (time-series of new infected cases). Therefore, the best state vectors (models) are those that generate an epidemic-curve closer to the observed one (i.e., models with higher likelihood). When the forecasting method's output is a collection of replicates of state vectors and time-series, we have the option to calculate Epi-features on each series, for each prediction time, and assess the error measures on each series. The error measures can be accumulated across the series through getting Arithmetic Mean, Median, Geometric Mean, etc. to provide a unique comparable value per each method. Table 6.2 provides advanced error measures to aggregating the error values over the series.

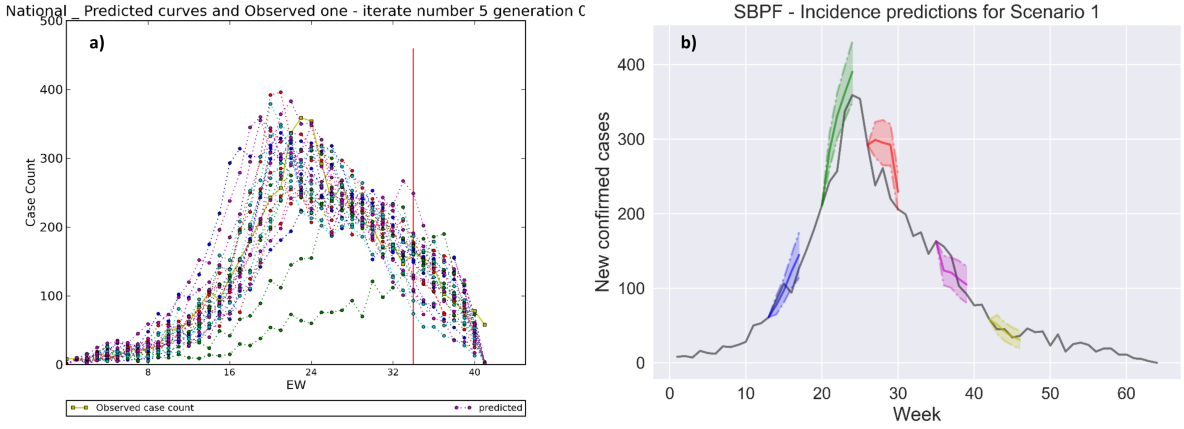


Figure 6.1. Two formats for the output of stochastic forecasting algorithms a) Multiple replicates of the time-series and b) A time-series of mean and variance of the predicted values.

Armstrong [96] performed an evaluation over some of these measures and suggested the best ones in different conditions. In calibration problems, a sensitive error measure is needed to demonstrate the change in parameters in the error measure values. The EMs with good sensitivity are RMSE, MAPE, and GMRAE. He suggested GMRAE because of poor reliability of RMSE and claimed that MAPE is biased towards the low forecasts [96]. As we mention in the discussion section, we believe that MAPE is not biased in favor of the low forecasts and could also be a good metric for calibration (refer to Discussion section). Also, GMRAE could drop to zero when the error contains at least one zero, thus lowering its sensitivity to zero too.

For selecting among forecasting methods, Armstrong suggested MdRAE when the output has a small set of series and MdAPE for a moderate number of series. He believes that reliability, protection against outliers, construct validity, and the relationship to decision-making are more important criteria than sensitivity. MdRAE is reliable and has better protection against outliers. MdAPE has a closer relationship to decision making and is protected against outliers [96].

Some stochastic forecasting algorithms, like our proposed Particle Filter method in Chapter 3, generate multiple time-series with uneven weights, which implies the likelihood level of the time-series to be the correct answer. In this case, it is important to consider the weight of the series in accumulating the values of Epi-features and error measures. As an illustration, instead of calculating MAPE, sMAPE, RMSE, and MdAPE across the time-series, we suggest measuring weighted-MAPE, weighted-sMAPE, weighted-RMSE, and weighted-MdAPE respectively. We also suggest measuring the weighted variance instead of the standard one, to demonstrate the weighted uncertainty.

6.4 Stochastic Forecasts with Uncertainty Estimates

Sometimes the output of a stochastic forecasting method is in the form of mean value and variance/uncertainty interval for the predicted value. In statistics theory, the summation of Euclidean distance between the data points and a fixed unknown point in n-dimensional space is minimized in the mean point. Therefore, the mean value is a good representative of other data points. As a result, we can simply calculate the epi-measure on the predicted mean value of an epidemic curve and compare them through error metrics. However, this comparison is not comprehensive enough because the deviation from the average value is not included in the discussion. To handle this kind of evaluation, we divide the problem to two sub-problems:

- Deterministic observation and stochastic forecasts with uncertainty estimates
- Stochastic observation and stochastic forecasts with uncertainty estimates

6.4.1 Deterministic Observation and Stochastic Forecasts with Uncertainty Estimates

In this case, we assume that each forecasting method's output is a time-series of uncertain estimates of predicted case counts and is reported by the mean value $\overline{x_t}$, variance σ_t^2 for data point at t^{th} week, and the number of samples N_x. For simplicity, we eliminate the subscript t. Table 6.1 lists the required notations used in the following sections. Sample size refers to the number of predicted samples from which the mean and variance are obtained. In the best situation, the forecast algorithm could provide with the probability density function (pdf) of each predicted data point denoted by $f(x)$, unless we assume the pdf is Normal distribution $f_x \sim N(\mu_x, \sigma_x)$ for the large enough sample size, or t-distribution $f_x \sim t(\mu_x, v)$ if the sample size is low. T-distribution has heavier tails, which means it is more subject to producing values far from the mean. $N_x \geq 30$ is assumed as a large sample size. N_x is used to calculate the standard deviation of the random variable X, from the standard deviation of its samples: $\sigma_x = \sigma/\sqrt{N_x}$. When the sample size is low, the degree of freedom of t-distribution is calculated by N_x: $v = N_x - 1$.

In order to evaluate the performance of stochastic methods, we suggest performing the Bootstrap sampling from the distribution $f(x)$ and generate the sample set $S_x = \{s_i\}$ for each data point of time-series where $|S_x| >> N_x$. Note that we do not have access to the instances of the first sample size, so we generate a large enough sample set from its pdf function $f(x)$. Then, the six selected error measures , MAE, RMSE, MAPE, sMAPE, MdAPE, and MdsAPE, are calculated across the sample set S_x *for each week*. Table 6.3 contains the extended formulation of the error measures used for stochastic forecasts. Using the equations in this table we can estimate different expected/median errors for each week for a stochastic forecasting method. The weekly errors could be aggregated by deriving Mean or

Median across the time to calculate the total error measures for each method. The aggregated error measures can be used to calculate the Consensus Ranking for the existing forecasting approaches. Moreover, having the errors for each week, we can depict the Horizon Ranking and evaluate the trend of rankings across the time similar to the graphs for deterministic approaches.

6.4.2 Stochastic Observation and Stochastic Forecasts with Uncertainty Estimates

There are many sources of errors in measurements and data collections which result in uncertainty for the observation data. This makes evaluation more challenging. We suggest two categories of solutions to deal with this problem:

- A) Calculating the distance between probability density functions
- B) Calculating the proposed error measures between two probability density functions

A) Calculating the distance between probability density functions

Assuming that both predicted and observed data are stochastic, they are represented as the time-series of probability density functions (pdfs). There are many distance functions that can calculate the distance between two pdfs [58]. Three most common distance functions for this application are listed in Table 6.4.

Bhattacharyya distance function [58] and Hellinger [97] both belong to the squared-chord family, and their continuous forms are available for comparing continuous probability density functions. In special cases, e. g. when the two pdfs follow the Gaussian distribution, these two distance functions can be calculated by the mean and variances of pdfs as follows [98, 99]:

$$D_B(P,Q) = \frac{1}{4} ln \left(\frac{1}{4} \left(\frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2}{\sigma_p^2} + 2 \right) \right) + \frac{1}{4} \left(\frac{(\mu_p - \mu_q)^2}{\sigma_p^2 + \sigma_q^2} \right) \tag{6.1}$$

$$D_H^2(P,Q) = 2 \left(1 - \sqrt{\frac{2\sigma_1.\sigma_2}{\sigma_1^2 + \sigma_2^2}}.exp(\frac{-(\mu_1 - \mu_2)^2}{4(\sigma_1^2 + \sigma_2^2)}) \right) \tag{6.2}$$

However, calculating the Integral may not be straightforward for an arbitrary pdf. Also, Jaccard distance function is in the discrete form. To solve this problem, we suggest Bootstrap sampling from both predicted and observed pdfs and generating the sample set $S = S_x \cup S_y$

where $S_x = \{s_i^x | s_i^x \sim f(x)\}$, $S_y = \{s_j^y | s_j^y \sim g(y)\}$, and $|S_x| = |S_y| >> N_x$. Then we calculate the summation for the distance function over all the items that belong to the sample set S. As an example for Jaccard distance function:

$$D_{Jac} = 1 - \frac{\sum_{k=1}^{|S|} f(s_k) \times g(s_k)}{\sum_{k=1}^{|S|} f(s_k)^2 + \sum_{k=1}^{|S|} g(s_k)^2 - \sum_{k=1}^{|S|} f(s_k) \times g(s_k)} \tag{6.3}$$

Jaccard distance function belongs to the inner product class and incorporates both similarity and dissimilarity of two pdfs. Using one of the aforementioned distance functions between the stochastic forecasts and stochastic observation, we can demonstrate Horizon Ranking across time and also aggregate the distance values by getting the mean value over the weeks, and then, calculate the Consensus Ranking.
Although these distance functions between the two pdfs seem to be a reasonable metric for comparing the forecast outputs, it ignores some information about the magnitude of error and its ratio to the real value. In other words, any pair of distributions like (P1,Q1) and (P2,Q2) could have the same distance value if : $|\mu_{P_1} - \mu_{Q_1}| = |\mu_{P_2} - \mu_{Q_2}|$ and $\sigma_{P_1} = \sigma_{P_2}$ and $\sigma_{Q_1} = \sigma_{Q_2}$. Therefore, the distance functions lose the information about the relative magnitude of error to the observed value. In the ranking process of different forecasting approaches, as the observed data is assumed to be fixed, this issue will not be a concern. The other problem of using distance functions between pdfs arises when some forecasting methods are stochastic and others are deterministic. As the proposed error measures are not compatible with distance functions, we cannot compare them together.

B) Calculating the error measures between two probability density functions

In order to compare stochastic and deterministic forecasting approaches together, we suggest estimating the same error measures used for deterministic methods. We perform Bootstrap sampling from both predicted and observed pdfs for each data point of time-series and generate two separate sample sets S_x and S_y where $S_x = \{s_i^x | s_i^x \sim f(x)\}$, $S_y = \{s_j^y | s_j^y \sim g(y)\}$ and $|S_x| = |S_y| >> N_x$. The six selected error measures, MAE, RMSE, MAPE, sMAPE, MdAPE, and MdsAPE, could be estimated through the equations listed in Table 6.5. These measures incorporate the variance of pdfs through the sampling and represent the difference between the predicted and observed densities by weighted expected value of the error across the samples.

6.5 Conclusion

The evaluation framework proposed in Chapter 5 primarily deal with deterministic forecasts. In this chapter, we extended the evaluation framework to be able to assess stochastic fore-

casting algorithms too. We divided the stochastic forecasts into two categories regarding the output format: i) algorithms with multiple replicates of the time-series as output and ii) algorithms that generate time-series of mean and variance of the predicted values. We provided some tips about how to calculate error measures for each category and also how to deal with stochastic observations. In the next chapter, we present a software package named Epi-Evaluator which embeds the proposed evaluation frameworks within three packages. To avoid the redundancy, the sample inputs and outputs of the software are included in the next chapter which also demonstrates how the Epi-features and error measures, which are proposed in this chapter, could quantify the output of stochastic methods.

Measure name	Formula	Description
Absolute Percentage Error ($APE_{t,s}$)	$APE_{t,s} = \lvert\frac{y_t - x_{t,s}}{y_t}\rvert$	where t is time horizon and s is the series index.
Mean Absolute Percentage Error ($MAPE_t$)	$MAPE = \frac{1}{S}\sum_{s=1}^{S} APE_{t,s}$	where t is time horizon, s is the series index S is the number of series for the method.
Median Absolute Percentage Error ($MdAPE_t$)	Median Observation of APE_s	Obtaining median of APE errors over series.
Relative Absolute Error ($RAE_{t,s}$)	$RAE_{t,s} = \frac{\lvert y_t - x_{t,s}\rvert}{\lvert y_t - x_{RW_{t,s}}\rvert}$	Measures the ratio of absolute error to Random walk error in time horizon t.
Geometric Mean Relative Absolute Error ($GMRAE_t$)	$GMRAE_t = [\prod_{s=1}^{S} \lvert RAE_{t,s}\rvert]^{1/S}$	Measures the Geometric average ratio of absolute error to Random walk error
Median Relative Absolute Error ($MdRAE_t$)	Median Observation of RAE_s	Measures the median observation of RAE_s for time horizon t
Cumulative Relative Error ($CumRAE_s$)	$CumRAE_s = \frac{\sum_{t=1}^{T} \lvert y_{t,s} - x_{t,s}\rvert}{\sum_{t=1}^{T} \lvert y_{t,s} - x_{RW_{t,s}}\rvert}$	Ratio of accumulation of errors to cumulative error of Random walk Method
Geometric Mean Cumulative Relative Error ($GMCumRAE$)	$GMCumRAE = [\prod_{s=1}^{S} \lvert CumRAE_s\rvert]^{1/S}$	Geometric Mean of Cumulative Relative Error across all series.
Median Cumulative Relative Error ($MdCumRAE$)	$MdCumRAE = Median(\lvert CumRAE_s\rvert)$	Median of Cumulative Relative Error across all series.
Root Mean Squared Error ($RMSE_t$)	$RMSE_t = \sqrt{\frac{\sum_{s=1}^{S}(y_t - x_{t,s})^2}{S}}$	Square root of average squared error across series in time horizon t
Percent Better (PB_t)	$PB_t = \frac{1}{S}\sum_{s=1}^{S}[I\{e_{s,t}, e_{WRt}\}]$ $\lvert e_{s,t}\rvert \leq \lvert e_{WRt}\rvert \leftrightarrow I\{e_{s,t}, e_{WRt}\} = 1$	Demonstrates average number of times that method overcomes the Random Walk method in time horizon t.

Table 6.2. List of advanced error measures to aggregating the error values across multiple series.

Measure name	Formula	Weighted-Expectation/Median of Measure
Absolute Error (AE)	$AE = \|s_i - y\|$	$E(AE) = \sum_{i=1}^{N_{sx}} f(s_i) \times \|s_i - y\|$
Absolute Percentage Error (APE)	$APE = \frac{\|s_i - y\|}{\|y\|}$	$E(APE) = \sum_{i=1}^{N_{sx}} f(s_i) \times \frac{\|s_i - y\|}{y}$
symmetric Absolute Percentage Error (sAPE)	$sAPE = \frac{2 \times \|s_i - y\|}{\|s_i + y\|}$	$E(sAPE) = 2 \times \sum_{i=1}^{N_{sx}} f(s_i) \times \frac{\|s_i - y\|}{s_i + y}$
Squared Error (SE)	$SE = (s_i - y)^2$	$RE(SE) = \sqrt{\sum_{i=1}^{N_{sx}} f(s_i) \times (s_i - y)^2}$
Absolute Percentage Error (APE)	$APE = \frac{\|s_i - y\|}{\|y\|}$	$Md(APE) = weightedMedian_{i=1}^{N_{sx}} \left\{ \frac{\|s_i - y\|}{y} \right\}$ where $w_i = f(s_i)$
symmetric Absolute Percentage Error (sAPE)	$sAPE = \frac{2 \times \|s_i - y\|}{\|s_i + y\|}$	$Md(sAPE) = weightedMedian_{i=1}^{N_{sx}} \left\{ 2. \frac{\|s_i - y\|}{s_i + y} \right\}$ where $w_i = f(s_i)$

Table 6.3. Error measures to measure error between the deterministic observation and stochastic predicted outputs.

Distance Function	Formula (Continues)	Formula (Discrete form)
Bhattacharyya	$D_B(P,Q) = -Ln(BC(P,Q))$, $BC(P,Q) = \int \sqrt{P(x)Q(x)}dx$	$D_B(P,Q) = -Ln(BC(P,Q))$, $BC(P,Q) = \sum \sqrt{P(x)Q(x)}$
Hellinger	$D_H = \sqrt{2 \int (P(x) - Q(x))^2 dx}$ $= 2\sqrt{1 - \int \sqrt{P(x)Q(x)}dx}$	$D_H(P,Q) = \sqrt{2 \sum_{k=1}^{d} (P(x_k) - Q(x_k))^2}$ $= 2\sqrt{1 - \sum_{k=1}^{d} \sqrt{P(x_k)Q(x_k)}}$
Jaccard	-	$D_{Jac} = 1 - S_{Jac}$ $S_{Jac} = \frac{\sum_{k=1}^{d} P(x_k) \times Q(x_k)}{\sum_{k=1}^{d} P(x_k)^2 + \sum_{k=1}^{d} Q(x_k)^2 - \sum_{k=1}^{d} P(x_k).Q(x_k)}$

Table 6.4. Distance functions to measure dissimilarity between probability density functions of stochastic observation and stochastic predicted outputs.

Measure name	Formula	Weighted-Expectation/Median of Measure
Absolute Error (AE)	$AE_{i,j} = \lvert s_i^x - s_j^y \rvert$	$E(AE) = \sum_{i=1}^{N_{sx}} \sum_{j=1}^{N_{sy}} f(s_i^x).g(s_j^y) \times \lvert s_i^x - s_j^y \rvert$
Absolute Percentage Error (APE)	$APE_{i,j} = \frac{\lvert s_i^x - s_j^y \rvert}{\lvert s_j^y \rvert}$	$E(APE) = \sum_{i=1}^{N_{sx}} \sum_{j=1}^{N_{sy}} f(s_i^x).g(s_j^y) \times \frac{\lvert s_i^x - s_j^y \rvert}{\lvert s_j^y \rvert}$
symmetric Absolute Percentage Error (sAPE)	$sAPE_{i,j} = \frac{2 \times \lvert s_i^x - s_j^y \rvert}{\lvert s_i^x + s_j^y \rvert}$	$E(sAPE) = 2 \times \sum_{i=1}^{N_{sx}} \sum_{j=1}^{N_{sy}} f(s_i^x).g(s_j^y) \times \frac{\lvert s_i^x - s_j^y \rvert}{s_i^x + s_j^y}$
Squared Error (SE)	$SE_{i,j} = (s_i^x - s_j^y)^2$	$RE(SE) = \sqrt{\sum_{i=1}^{N_{sx}} \sum_{j=1}^{N_{sy}} f(s_i^x).g(s_j^y) \times (s_i^x - s_j^y)^2}$
Absolute Percentage Error (APE)	$APE_{i,j} = \frac{\lvert s_i^x - s_j^y \rvert}{\lvert s_j^y \rvert}$	$Md(APE) = weightedMedian_{(i=1,j=1)}^{(N_{sx},N_{sx})} \left\{ \frac{\lvert s_i^x - s_j^y \rvert}{\lvert s_j^y \rvert} \right\}$ where $w_{i,j} = f(s_i^x).g(s_j^y)$
symmetric Absolute Percentage Error (sAPE)	$sAPE_{i,j} = \frac{2 \times \lvert s_i^x - s_j^y \rvert}{\lvert s_i^x + s_j^y \rvert}$	$Md(sAPE) = weightedMedian_{(i=1,j=1)}^{(N_{sx},N_{sx})} \left\{ 2.\frac{\lvert s_i^x - s_j^y \rvert}{\lvert s_i^x + s_j^y \rvert} \right\}$ where $w_{i,j} = f(s_i^x).g(s_j^y)$

Table 6.5. Error measures to measure error between the stochastic observation and stochastic predicted outputs.

Chapter 7

Epi-Evaluator: Software Package for Evaluating Epidemic Forecasts

7.1 Introduction

In the two previous chapters, we addressed the problem of evaluating epidemic forecasts by providing standard features, error measures, and ranking approaches. We proposed an evaluation framework that allows for combining different Epi-features, error measures, and ranking schema to evaluate both deterministic and stochastic forecasting algorithms.

In this chapter, we present a software package named Epi-Evaluator that includes three primary packages named Epi-Feature, Epi-Error and Epi-Ranking, which correspond to the three modules of the evaluation framework. This package is going to be released as an open source software to be accessible to epidemiologist and data scientist who are interested in epidemic forecasting.

As the concepts are introduced before, we skip the preliminary and notation section. The rest of the chapter is organized as follows. In Section 7.2, we present the structure of the Epi-Evaluator using Unified Modeling Language diagrams. Section 7.3 provides the sample inputs and outputs of each individual package. Finally, we conclude the chapter in Section 7.4.

7.2 Epi-Evaluator Structure

We provide a top-down presentation of Epi-Evaluator by depicting the organizations and structures of the software. Different standard diagrams in Unified Modeling Language (UML) [100] are used to describe the detail of the packages. In this section, we present a *use case*

diagram and a *package diagram* for the software to better elaborate the interaction between objects with various *sequence diagrams*.

7.2.1 Use Case Diagram

Figure 7.1 presents the *use case diagram* of Epi-Evaluator that depicts the graphical overview of different actors, packages, and objects in the system and also their interaction with each other.

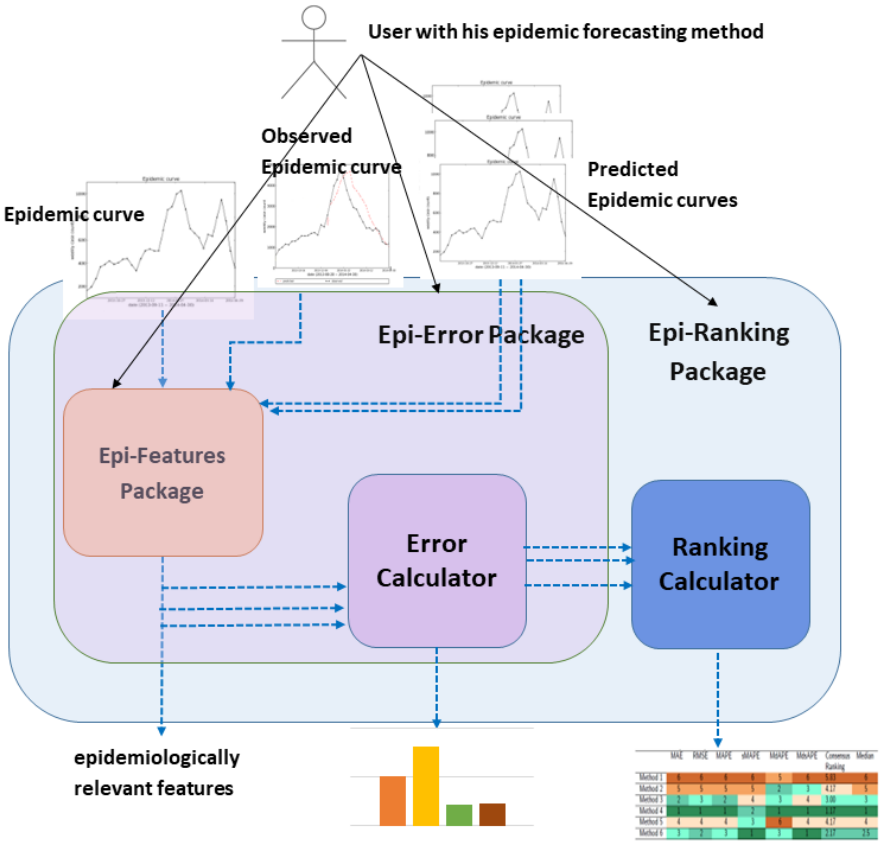


Figure 7.1. Epi-Evaluator use case diagram: The main actor in the system is the user who needs to analyze the predicted epidemic curves and compare them with the observed data and/or other predicted curves.

This software contains three sub-packages called *Epi-Feature*, *Epi-Error*, and *Epi-Ranking* packages. The system is designed as a collection of scripts that are loosely coupled through the data they exchange. As discussed before, *Epi-Feature* package receives the output of a forecasting method in the form of epidemic curves, and extracts the epidemiological related features from them. *Epi-Error* package is designed to measure various error metrics between the features of predicted and ground truth epidemic curves. It invokes the *Epi-Feature*

package to measure the curve features and then calculates the desired error measures as a criteria for quantifying the performance of epidemic forecasts. The *Ranking package* contains both Epi-Feature and Epi-Error packages and uses different ranking schema to rate the performance of various methods in comparison with each other based on the calculated error measures. A user is able to call any of these three packages individually according to specific requirements.

7.2.2 Epi-Evaluator's Package Diagram

Figure 7.2 presents the package diagram of the Epi-Evaluator software, showing the content of its packages and their dependencies.

The *Util* sub-package provides the basic utility functions for each module, and the *Vis* sub-package visualizes the outputs in the form of diagrams and charts. In the Epi-Feature package, the two main classes are *Epi-Feature* and *runEpi-Feature*. The Epi-Feature class contains all necessary functions for generating various Epi-features. The runEpi-Feature class represents how to invoke methods of the *Epi-Feature* class. The same structure is used for the main sub-packages of Epi-Error and Epi-Ranking modules. *Epi-Error* class contains the primary functions for calculating the error measures from epidemic curves and their corresponding Epi-features. *runEpi-Error* class runs and tests these functions by making connections between them and the methods in the *Epi-feature* class. A class named *runEpi-Rank* in Epi-Ranking module is responsible for making the required connections with other classes such as *Epi-Feature*, *Epi-Error* and *Epi-Rank*. The *Epi-Rank* class contains primary functions to rank various forecasting approaches based on different error measures.

7.2.3 Epi-Evaluator's Sequence Diagram

In this section, we depict the sequence diagrams for all three packages of the Epi-Evaluator software. This type of graph describes various interactions among classes in forms of sending and receiving messages and data over time. The sequence diagram visualizes the relationship between the classes, their responsibilities, and various run-time scenarios that may happen. Figure 7.3, represents the sequence diagram for the Epi-Feature package. The *runEpi-feature* class is the main active class that is called by a external user and is a wrapper over other classes. It receives arguments and inputs from the user and calls functions of other classes and makes the required links. The user provides one or multiple epidemic curves as input and can ask for desired epidemic features as output. The details of the arguments and the format of input data are included in the software manual.

Figures 7.4 and 7.5 demonstrate sequence diagrams of the Epi-Error and Epi-Ranking packages. *runEpi-Error* and *run-EpiRank* classes are the wrappers over other classes for Epi-Error and Epi-Ranking, respectively. Each wrapper is responsible for receiving the input

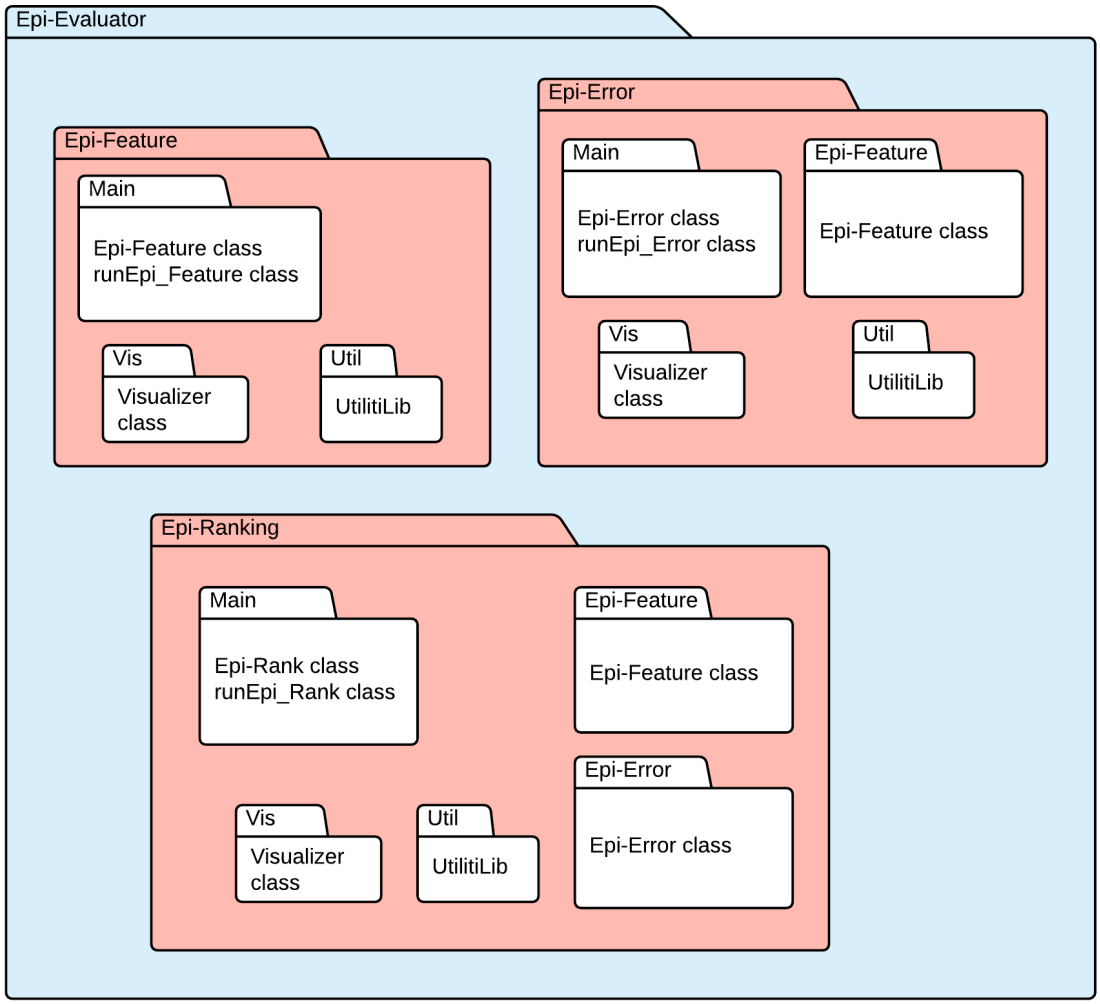


Figure 7.2. Epi-Evaluator package diagram: Showing the content of the packages and their dependencies.

arguments, storing outputs and logging the messages.

7.3 Epi-Evaluator Outcomes

In this section, we provide sample input and outputs for the three packages of the Epi-Evaluator. Each package communicates with the user through the arguments that are described in details in the software manual.

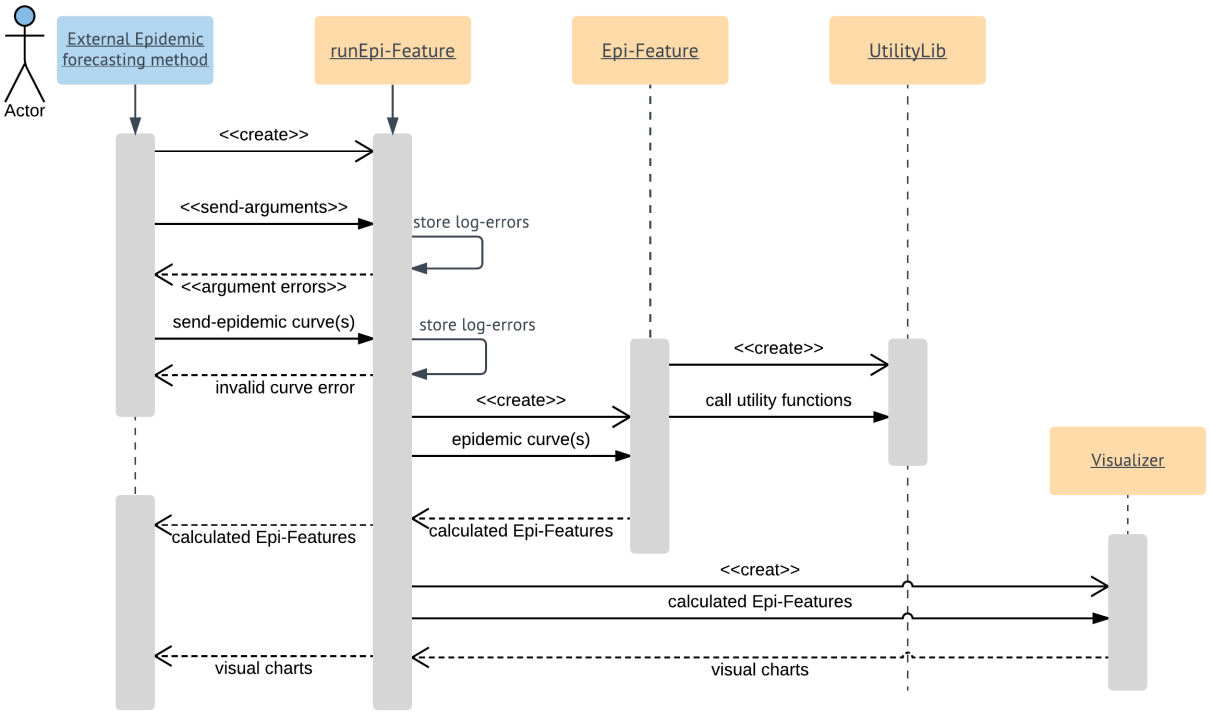


Figure 7.3. Sequence diagram of the Epi-Feature package: Showing the interactions between the classes of Epi-Feature package and their responsibilities.

7.3.1 Epi-Feature Package

Epi-Feature package receives the output of a forecasting method in the form of an epidemic curve and extracts epidemiological related features from it. The software package is provided in two modes for deterministic and stochastic algorithms.

Epi-Feature Package for Deterministic Algorithms

Assuming a forecasting algorithm belongs to the deterministic class, it only generates one time-series of new infected cases as a predicted epidemic curve in a comma delimited file. The numbers in the first row of the file refer to week numbers of the season and the second row lists the number of new infected cases corresponded to the above weeks. Moreover, the output of a deterministic algorithm could be generated for multiple prediction weeks (PW), and stored in a comma delimited file. In this case, as can be seen in Figure 7.6, each pair of rows corresponds to one epidemic curve predicted at a prediction week that is specified in the first column. The numbers in the first and second rows of each pair refer to week numbers and number of new infected cases, respectively.

Epi-Feature package receives a file name and a list of desired Epi-features as input, and

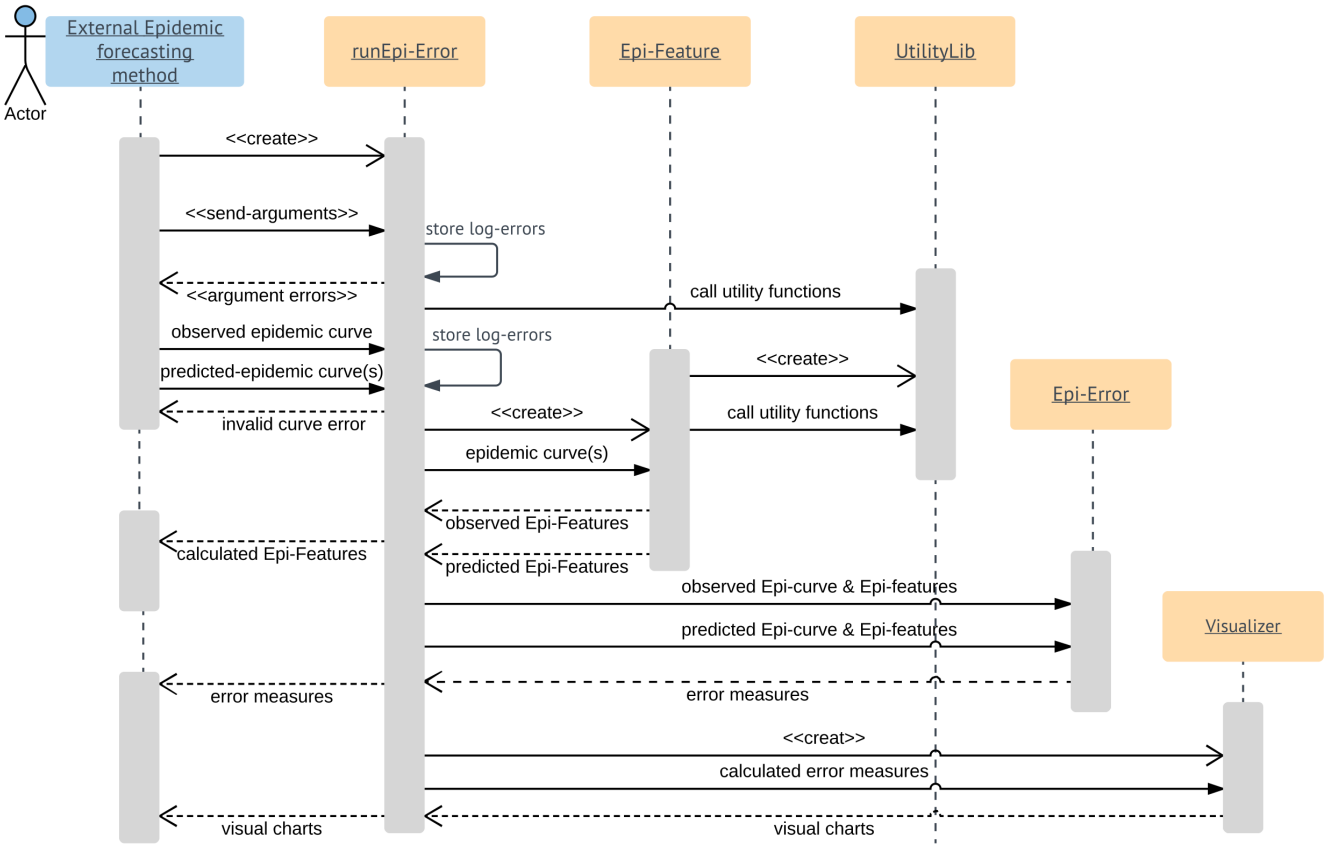


Figure 7.4. Sequence diagram of the Epi-Error package: Showing the interactions between the classes and their responsibilities.

generates and stores the results in the form of comma delimited files and visualized charts. Figure 7.7 shows the output of the Epi-feature package as a table of desired Epi-features.

If multiple epidemic curves are predicted in different prediction times, the Epi-Feature package is capable of depicting diagrams of Epi-features (see Figure 7.8).

Epi-Feature Package for Stochastic Algorithms

As mentioned before, stochastic forecasting algorithms could be in one of the following formats: *i)* Multiple replicates of the time series and *ii)* A timeseries of mean and variance of the predicted values.
The current version of Epi-Evaluator package is able to handle epidemic forecasts with multiple replicates of time-series. These replicates could be weighted or unweighted. The format of the input file will simply change to multiple curves per each prediction week as shown in Figure 7.9.

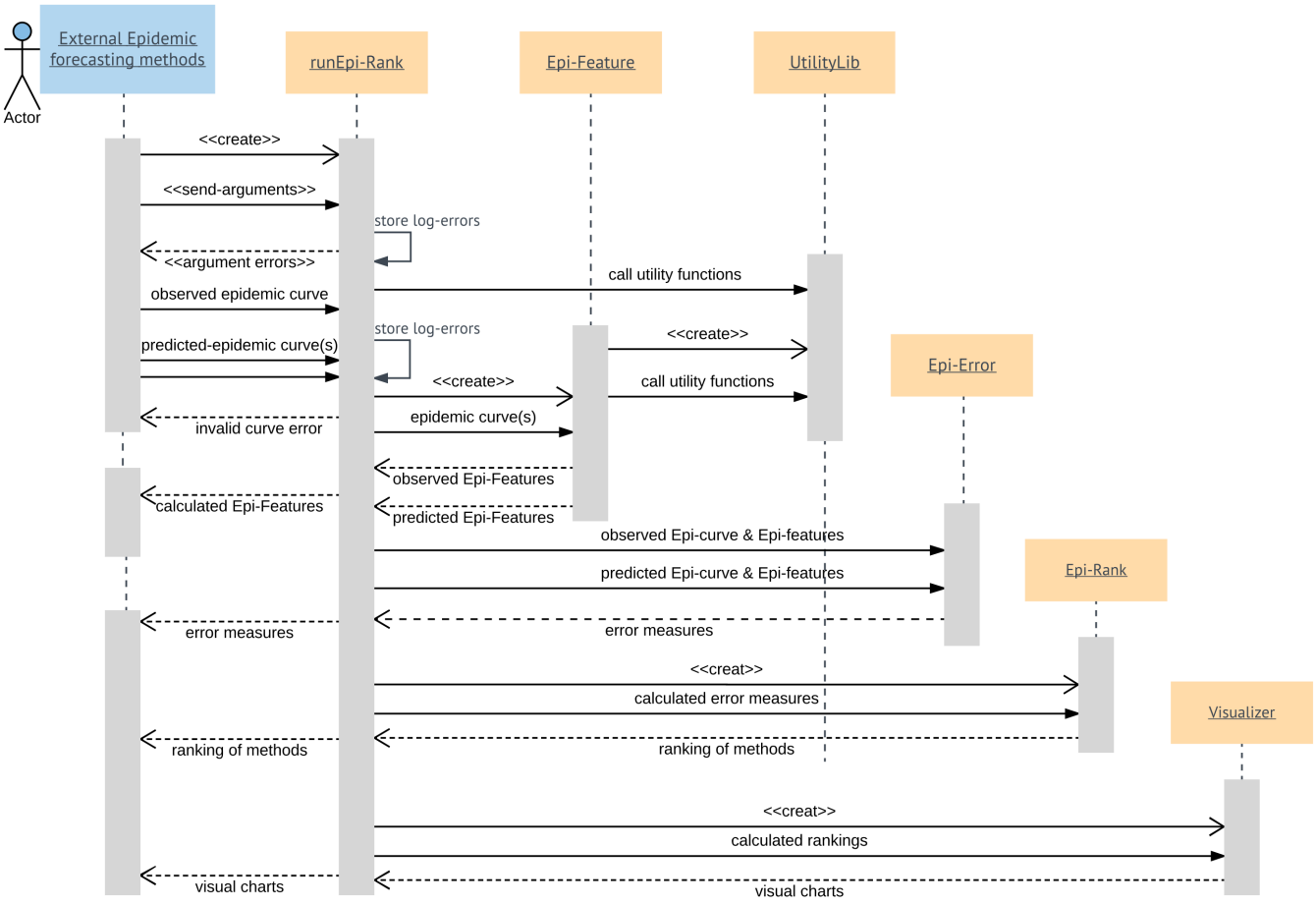
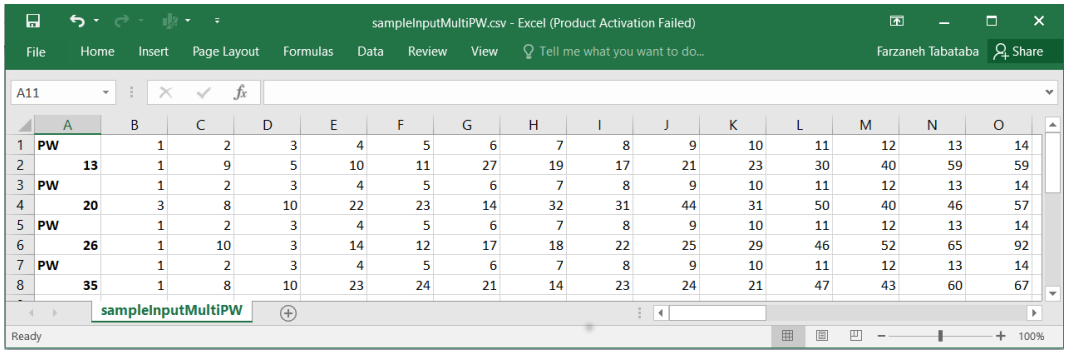


Figure 7.5. Sequence diagram of Epi-Ranking package: Showing the interactions between the classes and their responsibilities.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	PW	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	13	1	9	5	10	11	27	19	17	21	23	30	40	59	59
3	PW	1	2	3	4	5	6	7	8	9	10	11	12	13	14
4	20	3	8	10	22	23	14	32	31	44	31	50	40	46	57
5	PW	1	2	3	4	5	6	7	8	9	10	11	12	13	14
6	26	1	10	3	14	12	17	18	22	25	29	46	52	65	92
7	PW	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	35	1	8	10	23	24	21	14	23	24	21	47	43	60	67

Figure 7.6. Sample inputs for Epi-Feature package: Output of a deterministic forecasting algorithm for different prediction weeks: PW = 13, PW = 20, PW = 26, and PW = 35.

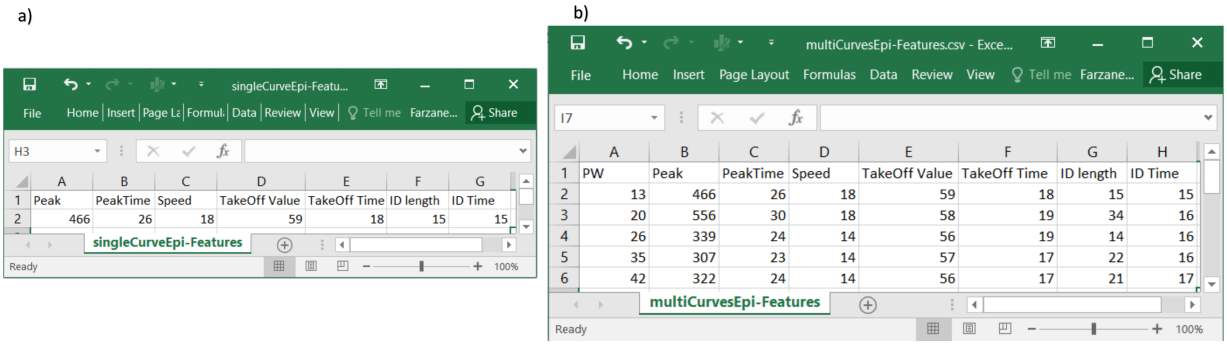


Figure 7.7. Epi-Feature package's output for a deterministic algorithm: Various Epi-features for **a)** single predicted epidemic curve and **b**) multiple epidemic curves predicted in different prediction weeks: PW = 13, PW = 20, PW = 26, PW = 35, and PW = 42.

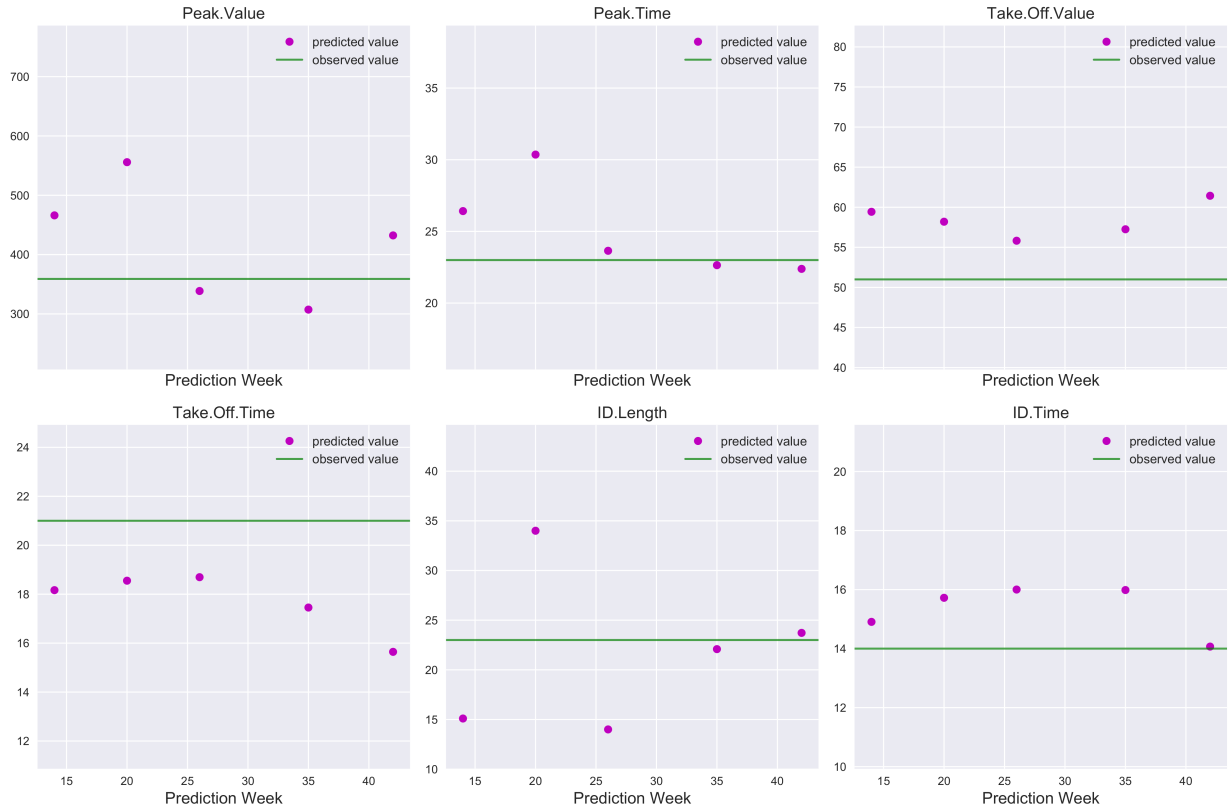
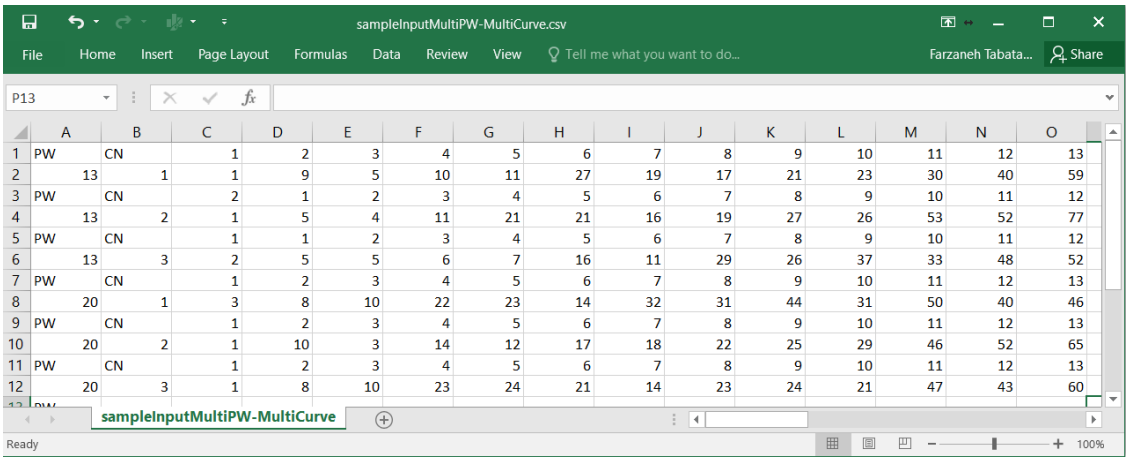


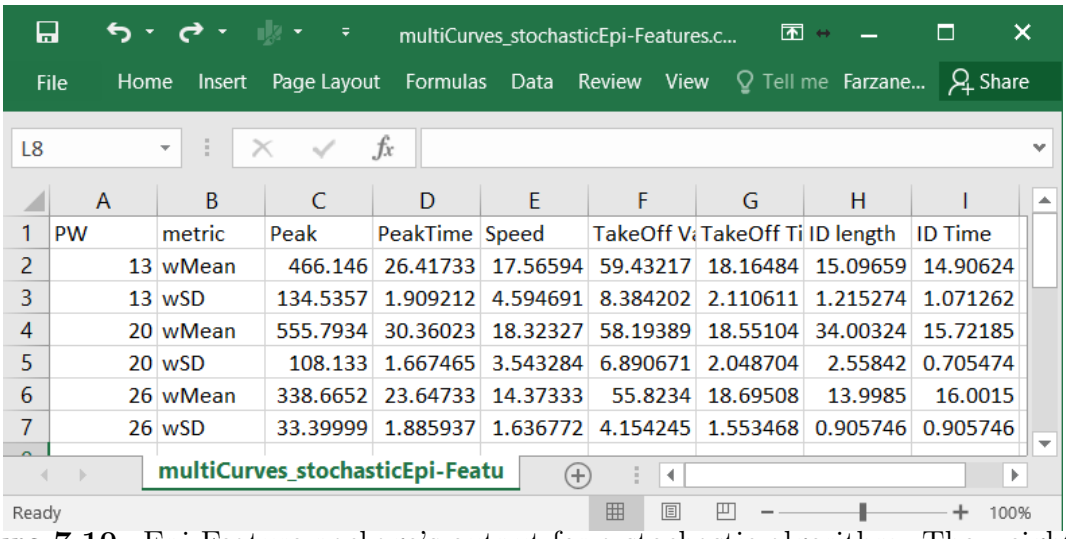
Figure 7.8. Epi-Feature package's output for a deterministic algorithm: Diagram of various Epi-features for multiple epidemic curves predicted in different prediction weeks: PW = 13, PW = 20, PW = 26, PW = 35, and PW = 42.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	PW	CN	1	2	3	4	5	6	7	8	9	10	11	12	13
2	13	1	1	9	5	10	11	27	19	17	21	23	30	40	59
3	PW	CN	2	1	2	3	4	5	6	7	8	9	10	11	12
4	13	2	1	5	4	11	21	21	16	19	27	26	53	52	77
5	PW	CN	1	1	2	3	4	5	6	7	8	9	10	11	12
6	13	3	2	5	5	6	7	16	11	29	26	37	33	48	52
7	PW	CN	1	2	3	4	5	6	7	8	9	10	11	12	13
8	20	1	3	8	10	22	23	14	32	31	44	31	50	40	46
9	PW	CN	1	2	3	4	5	6	7	8	9	10	11	12	13
10	20	2	1	10	3	14	12	17	18	22	25	29	46	52	65
11	PW	CN	1	2	3	4	5	6	7	8	9	10	11	12	13
12	20	3	1	8	10	23	24	21	14	23	24	21	47	43	60

Figure 7.9. Sample input for the Epi-Feature package: Output of a stochastic forecasting algorithm for different prediction weeks. *PW* and *CN* correspond to prediction weeks and curve numbers, respectively.

Epi-feature package receives multiple epidemic curves and measures the mean and the variance value of desired Epi-features. If multiple replicates are weighted, the Epi-feature package applies the weights by calculating the weighted mean and weighted variance of the features (Figure 7.10).



	A	B	C	D	E	F	G	H	I
1	PW	metric	Peak	PeakTime	Speed	TakeOff V	TakeOff Ti	ID length	ID Time
2	13	wMean	466.146	26.41733	17.56594	59.43217	18.16484	15.09659	14.90624
3	13	wSD	134.5357	1.909212	4.594691	8.384202	2.110611	1.215274	1.071262
4	20	wMean	555.7934	30.36023	18.32327	58.19389	18.55104	34.00324	15.72185
5	20	wSD	108.133	1.667465	3.543284	6.890671	2.048704	2.55842	0.705474
6	26	wMean	338.6652	23.64733	14.37333	55.8234	18.69508	13.9985	16.0015
7	26	wSD	33.39999	1.885937	1.636772	4.154245	1.553468	0.905746	0.905746

Figure 7.10. Epi-Feature package's output for a stochastic algorithm: The weighted mean (wMean) and weighted standard deviation (wSD) of various Epi-features for multiple epidemic curves that are predicted in different prediction weeks.

The Epi-Feature package is capable of depicting boxplot diagrams of Epi-features over time and compare them with the observed value (see Figure 7.11).

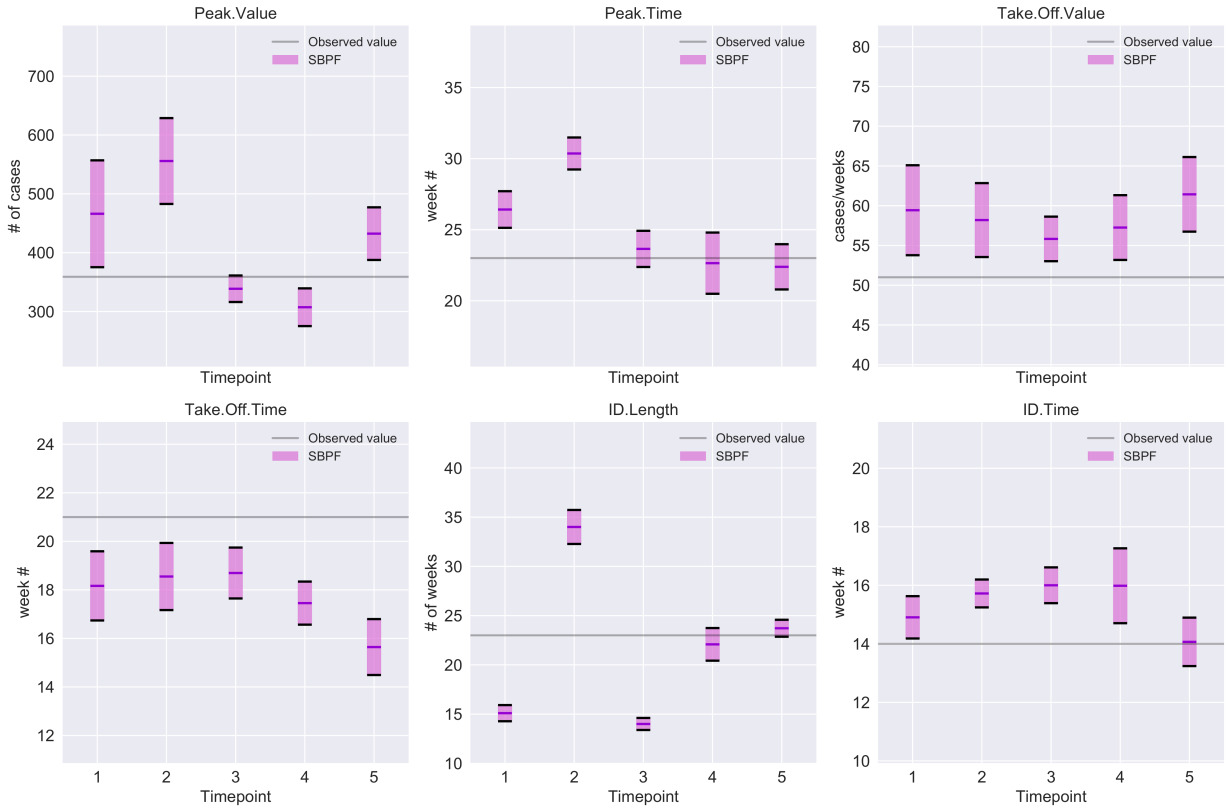


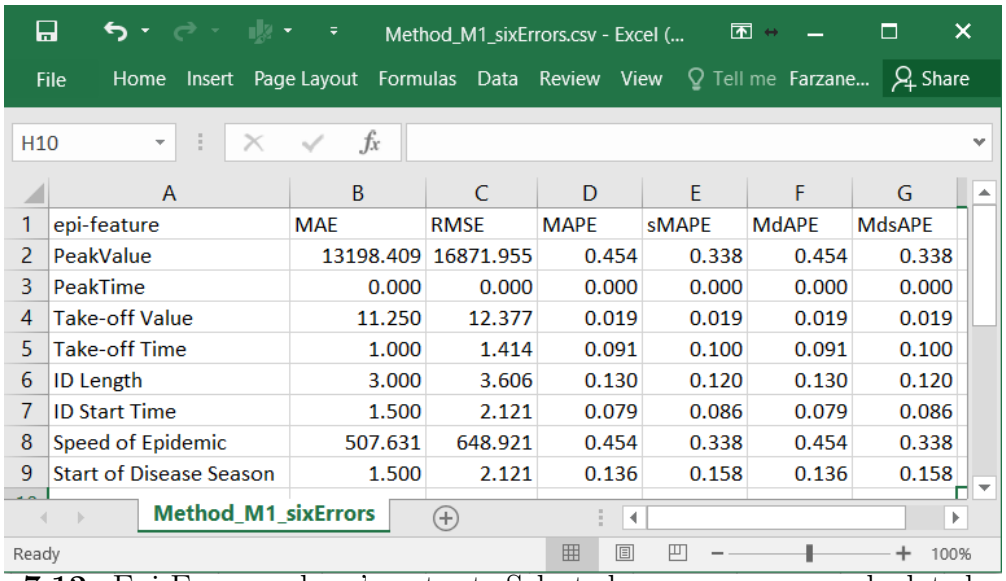
Figure 7.11. Epi-Feature package's output for a stochastic algorithm: Boxplot diagrams of various Epi-features for multiple epidemic curves predicted in different prediction weeks.

7.3.2 Epi-Error Package

Epi-Error Package for Deterministic Algorithms

As discussed before, the error measures assess the distance between the predicted time-series and the observed one. The time-series could be series of new infected case counts (epidemic curve) or series of predicted Epi-feature in multiple prediction weeks such as time-series of peak values, time-series of take-off values, etc. Epi-Error package receives the predicted epidemic curves of a forecasting algorithm and the ground truth curve as input. It first extracts the epidemiological related features from the predicted and ground truth curves by calling the Epi-feature class. Then, it calculates different error measures between the observed and predicted features, and stores them as a table in a comma delimited file (Figure 7.12).

Epi-Error package also generates the column chart of various error measures for each desired Epi-feature (Figure 7.13).



	A	B	C	D	E	F	G
1	epi-feature	MAE	RMSE	MAPE	sMAPE	MdAPE	MdsAPE
2	PeakValue	13198.409	16871.955	0.454	0.338	0.454	0.338
3	PeakTime	0.000	0.000	0.000	0.000	0.000	0.000
4	Take-off Value	11.250	12.377	0.019	0.019	0.019	0.019
5	Take-off Time	1.000	1.414	0.091	0.100	0.091	0.100
6	ID Length	3.000	3.606	0.130	0.120	0.130	0.120
7	ID Start Time	1.500	2.121	0.079	0.086	0.079	0.086
8	Speed of Epidemic	507.631	648.921	0.454	0.338	0.454	0.338
9	Start of Disease Season	1.500	2.121	0.136	0.158	0.136	0.158

Figure 7.12. Epi-Error package's output: Selected error measures calculated on time-series of predicted Epi-feature in multiple prediction weeks.

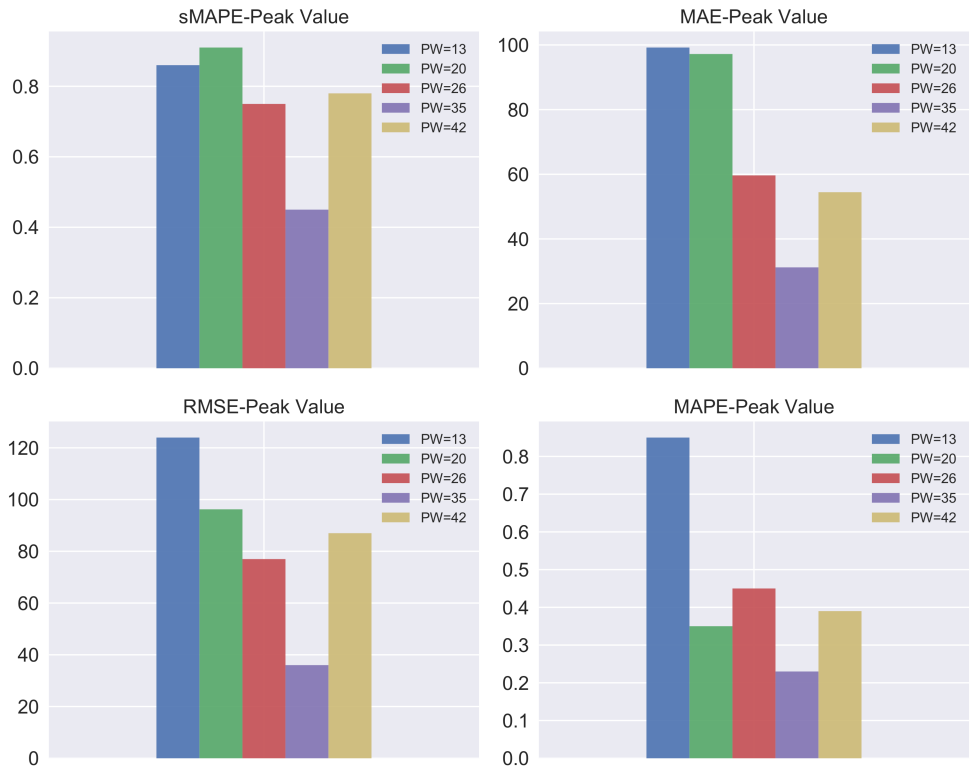


Figure 7.13. Epi-Error package's output: Diagrams of selected error measures on time-series of predicted *peak values* for multiple prediction weeks.

Epi-Error Package for Stochastic Algorithms

Similar to Epi-Feature module, the Epi-Error package is able to handle stochastic forecasts with multiple replicates of the time-series generated for a single prediction time. Epi-Error package receives multiple replicates with their corresponding weights (if they are weighted) and calculates weighted mean and variance of the error measures across multiple replicates and multiple times, consecutively. For example, Figure 7.14-a) shows different replicates of predicted epidemic curves forecasting the trend of the epidemic for 4 weeks ahead. Each error measure, such as Absolute Percentage Error (APE) or symmetric Absolute Percentage Error (sAPE), are computed on each predicted curve for a single week (after prediction time) and the weighted mean of the errors are measured across multiple replicates. The Epi-Error module aggregates the errors over time by measuring the average value of weighted-MAPE or weighted-sMAPE over time (Figure 7.14-b). The same process of measuring errors could be applied to the time-series of the Epi-features.

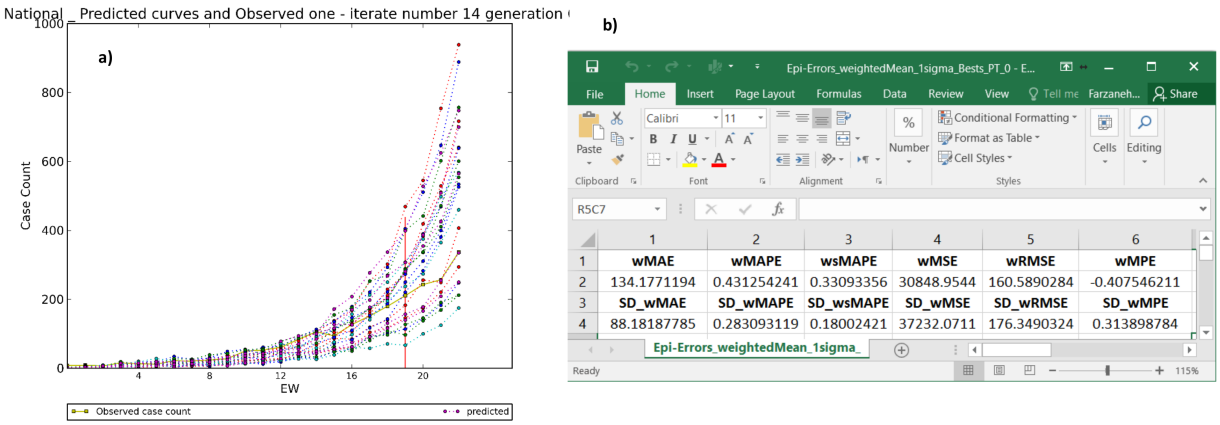
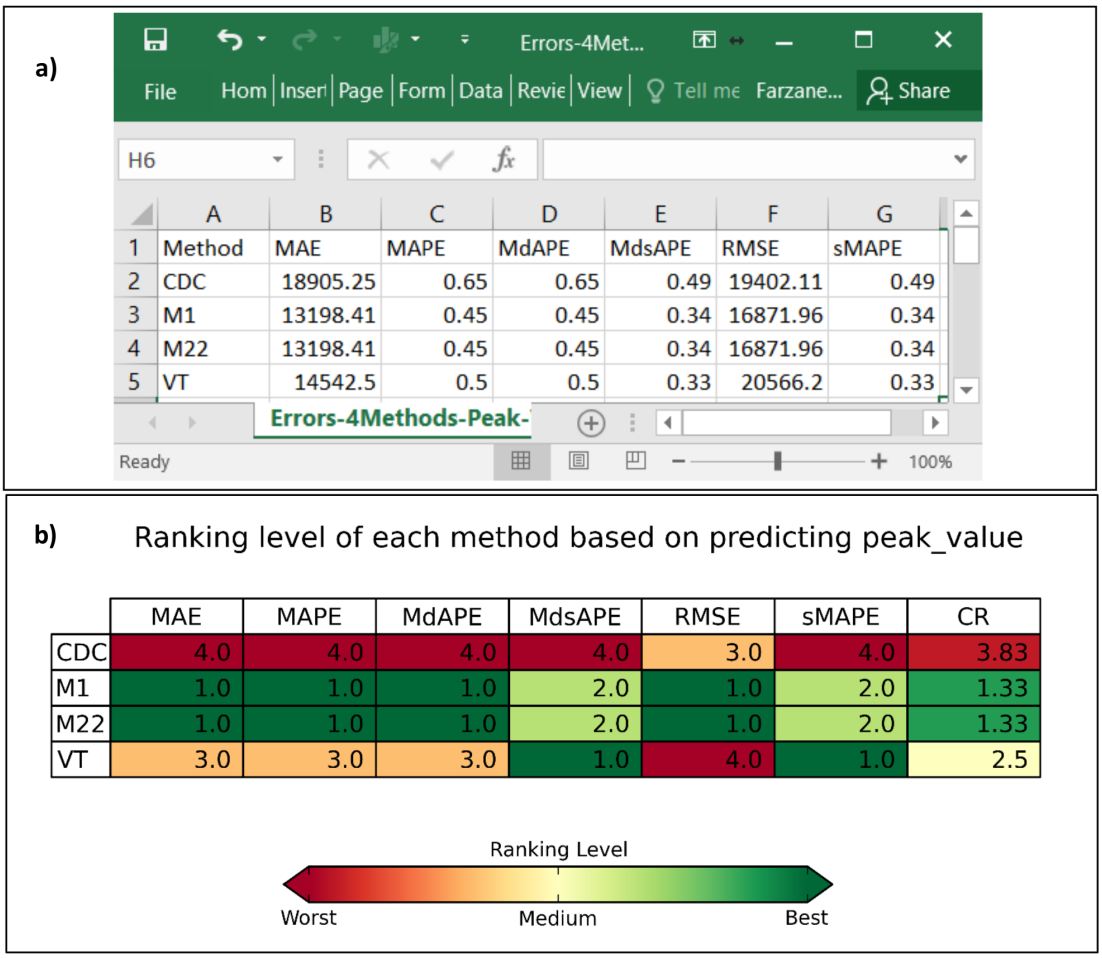


Figure 7.14. a) Different replicates of predicted epidemic curves forecasting the trend of pandemic for 4 weeks ahead. The olive-green curve represents the ground truth. b) (Weighted) error measures calculated across multiple replicates of epidemic curves and over times, consecutively.

7.3.3 Epi-Ranking Package

The Epi-Ranking module receives the output of multiple methods and ranks their performances in predicting various Epi-features based on the proposed error measures. The *runEpi-Rank* class reads a comma delimited file for each forecasting method and calculates the desired Epi-features by invoking the *Epi-Feature* class. The Epi-Error module, which is embedded in the Epi-Ranking package, computes the error measure for all Epi-features and each forecasting method. The Epi-Ranking module ranks various forecasting algorithms based on different error measures in predicting each Epi-feature. Therefore, it generates the

ranking table of the methods for each measured Epi-feature. Figure 7.15 presents a ranking table generated for four hypothetical methods based on their accuracy in predicting the peak value.



	A	B	C	D	E	F	G
1	Method	MAE	MAPE	MdAPE	MdsAPE	RMSE	sMAPE
2	CDC	18905.25	0.65	0.65	0.49	19402.11	0.49
3	M1	13198.41	0.45	0.45	0.34	16871.96	0.34
4	M22	13198.41	0.45	0.45	0.34	16871.96	0.34
5	VT	14542.5	0.5	0.5	0.33	20566.2	0.33

	MAE	MAPE	MdAPE	MdsAPE	RMSE	sMAPE	CR
CDC	4.0	4.0	4.0	4.0	3.0	4.0	3.83
M1	1.0	1.0	1.0	2.0	1.0	2.0	1.33
M22	1.0	1.0	1.0	2.0	1.0	2.0	1.33
VT	3.0	3.0	3.0	1.0	4.0	1.0	2.5

Figure 7.15. a) Comparing the accuracy of different forecasting methods in predicting peak value via various error measures. b) The ranking table corresponds to the above error table generated by Epi-Ranking package. The color spectrum demonstrates different ranking levels. Dark green represents the best rank, whereas dark red represents the worst one. The *CR* represents the Consensus Ranking of methods in predicting peak value across all specified error measures.

The Epi-Ranking package generates a boxplot diagram per each ranking table which summarizes the distribution of each method's ranking in predicting an Epi-feature (Figure 7.16).

The Epi-Ranking module is able to generate all the outputs of Epi-Feature and Epi-Error packages including the error tables, and boxplot diagrams. Moreover, it depicts boxplot charts of various methods in one frame that makes the comparison easier (Figure 7.17).

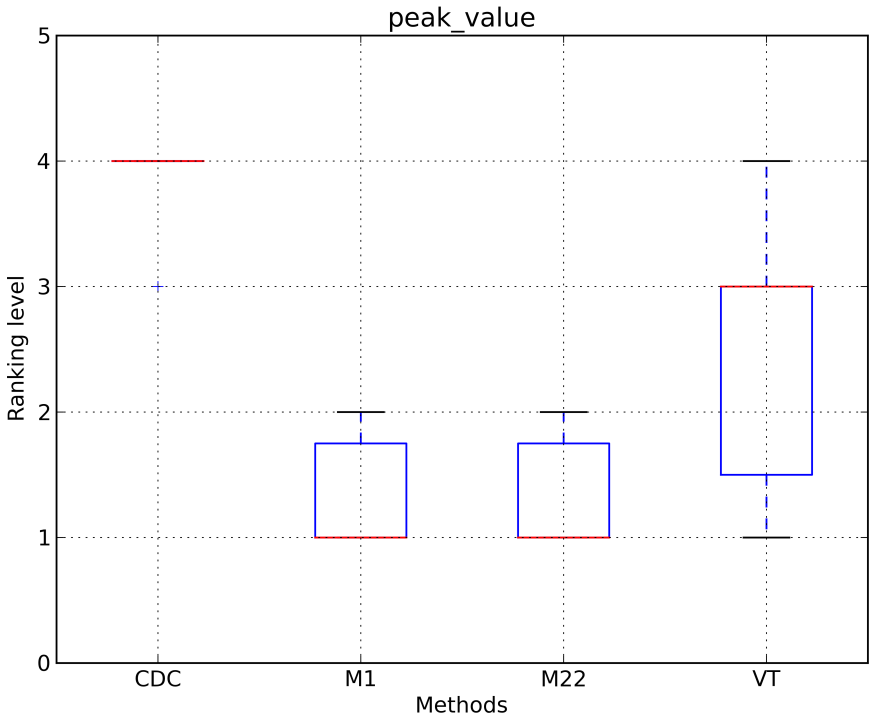


Figure 7.16. The boxplot corresponds to the ranking table in Figure 7.15-b which summarizes the distribution of method rankings in predicting the peak value.

7.4 Conclusion and Future Work

In this chapter, we presented a software package named Epi-Evaluator. It contains three primary packages named Epi-Feature, Epi-Error, and Epi-Ranking which correspond to the three modules of the evaluation framework. The Epi-Feature package generates the epidemiologically relevant features (Epi-features) of epidemic curves. Epi-Error package compares those Epi-features with the ground truths by assessing the error measures. Finally, Epi-Ranking package ranks various forecasting methods based on the measured errors. We dissected the structure of each package by drawing the package and sequence diagrams using Unified Modeling Language (UML). The *Vis* sub-package in each module creates more tangible outputs in the form of visual diagrams and charts. The current version of the software package can handle the stochastic forecasting methods whose outputs are in the form of multiple epidemic curves. In the future version, we are going to expand our software for the stochastic approaches with uncertainty intervals format. Epi-Evaluator is going to be released as an open source software. Its packages are loosely connected, so users are able to embed each of them in their frameworks.

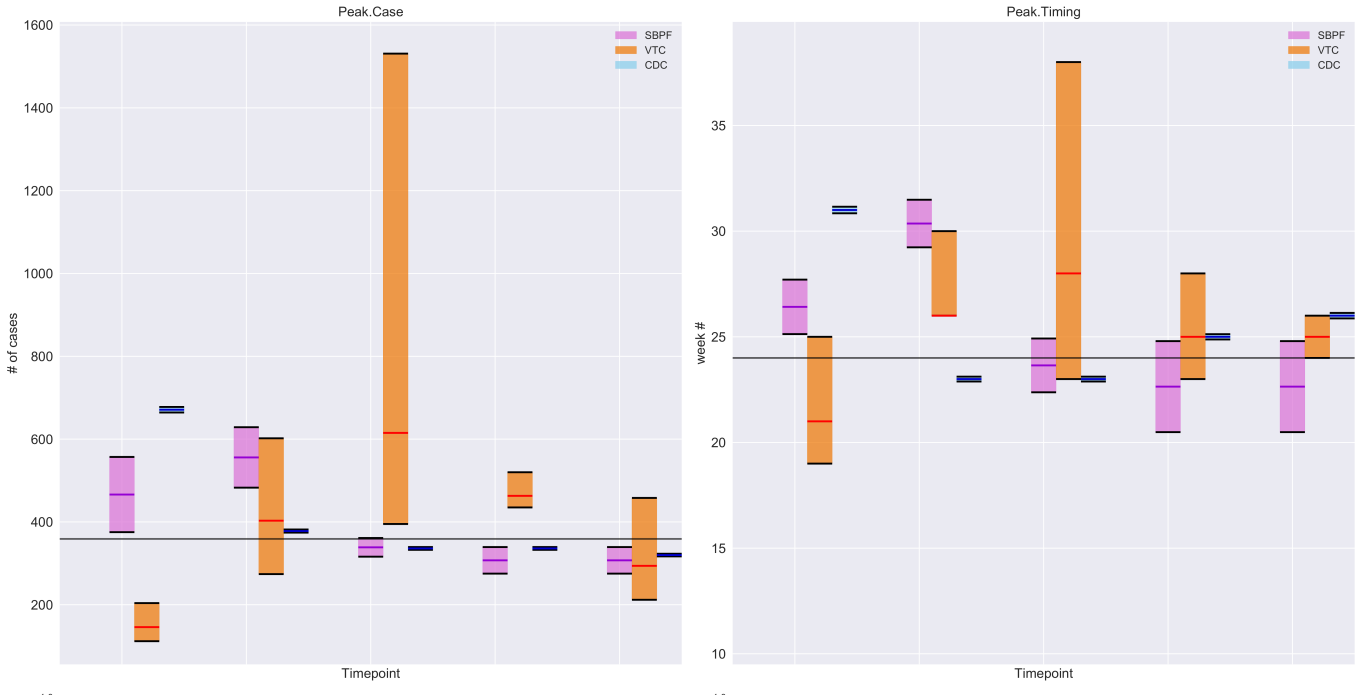


Figure 7.17. The boxplot diagram of various Epi-features predicted in different prediction weeks via multiple forecasting approaches.

Part III

Appendices

Appendix A

Chapter 3: Appendix

A.1 Graphical Discrete Dynamical System (GDDS) model for Epifast

The agent-based framework utilized in this paper is an interaction-based epidemic simulator, named Epifast. Epifast simulates the spatio-temporal propagation of the disease through the social interactions between individuals [29]. Using census data, a contact network is constructed based on the detailed demographic profiles and daily activities of each person. Epifast uses SEIR model to simulate the dynamic of disease among the contact network. SEIR is a stochastic compartmental model that categorizes each individual at every time step according to these four health states: susceptible, exposed, infectious, and removed [9].

Epifast provides the capability of simulating a broad range of policy-based, as well as individual-based, interventions. Interventions could be pharmaceutical (PI) or non-pharmaceutical (NPI). PIs include dispensing antivirals, vaccines, and antibiotics, whereas NPIs refer to any change in individual interactions and social network structure aside from using medicine. We use Graphical Discrete Dynamical System (GDDS) to model the problem of interaction-based epidemic propagation and applied interventions [33]. GDDS, represented by a tuple (G,F) includes two main components:

1. Graph $G(V, E)$: The contact network in Epifast simulator is constructed by a directed graph in which the vertices $V = \{v_1, v_2, ..., v_n\}$ represent individuals (agents) in the society; the edges determine the mutual contacts. Each vertex v_i has a state vector s_i that describes the properties of the vertex in the network: $s_i = \left\{s_i^1, s_i^2, ..., s_i^k\right\} \in D = (D_1 \times D_2 \times .. \times D_k)$ where k is the number of state categories. Table A.1, lists the possible state categories and their corresponding domain in Epifast. The edge set $E = \{e_1, e_2, ..., e_m\} \subseteq (V \times V)$ denotes the contact between the agents. For each edge $e_1 \in E$, there is a label vector that contains the properties of the corresponding

contact: $\left(l_i^1, l_i^2, ..., l_i^h\right) \in L$, where $L = (L_1, L_2, ..., L_h)$ and h represent the number of label categories. In Epifast contact network, edge label categories are *contact type* and *contact duration* (Table A.1).

2. Functions $F = \left\{f, f^V, f^E, f^I\right\}$. There are four function types that simulate the disease dynamics and intervention activities:

   - f is a set of local transition functions: for each vertex v_i, the function $f_i : D \times D^{v_i} \times L^{E_i} \rightarrow D$ defines the transitions that change the state of agent i based on its current state, its neighbors' states (V_i) and the current labels of its incident edges (E_i). These functions simulate the disease propagation between agents (Table A.2).
   - $f^V = \left\{f_1^V, f_2^V, ..., f_{n_V}^V\right\}$ is a set of vertex modification functions that change the state of vertices based on current state of the whole graph. These functions correspond to within-host disease progression, PIs, or individual-behavior changes (see examples of f^V functions in Table A.2).
   - $f^E = \left\{f_1^E, f_2^E, ..., f_{n_E}^E\right\}$ is a set of edge modification functions that change the label of desired edges based on the current state of the graph. These functions simulate NPIs such as quarantine, school closures, travel reductions, and social distancing.
   - f^I denotes the set of intervention functions which is $f^I \subseteq \left\{f^E \cup f^I\right\}$. Each $f^I : SP \times C \rightarrow SP \times A$, where $SP = \{sp^1, ..., sp^p\}$ are the set of possible subpopulations, $C = \{C^1, ..., C^m\}$ is the collection of trigger conditions, and $A = \{A^1, ..., A^z\}$ is the actions that should make changes to the desired subpopulation. Intervention function does not happen unless the required condition C^i is satisfied for the required subpopulation sp^j. Then the corresponding action A^k is applied on the targeted subpopulation sp^l. The condition and target subpopulation could be different (Table A.2).

Epifast loads a pre-constructed contact graph and a configuration file that determines the settings and function types that should be considered during the simulation. After commencing the simulation, it runs the sequences of different functions from the set F and changes the status of the vertices and the structure of contact network accordingly. Algorithm **??** shows the procedure of the disease propagation loop with predefined functions and interventions in the context of GDDS model (Bisset et al., 2014).

A.2 Proof of Update Equation for Bayesian Filter

Theorem.1 *The posterior probability density function (pdf) of the state vector* (x_k) *given all observations received up to time* k *is* $p(x_k|Y_k) = p(y_k|x_k) \times p(x_k|Y_{k-1})/p(y_k|Y_{k-1})$, *where*

GDDS components		**Possible values in Epifast simulator**	
Graph(V,E)	V_i = nodes = Agents	$S^1 = \{PID: \ Patient \ ID\}$	$D^1 = \{PID \in N\}$
		$S^2 = \{health \ status\}$	$D^2 = \{S, \ E, \ I, \ R\}$
		$S^3 = infectivity = \{r^{Out}\}$	$D^3 = \{r^{out} \in R \vert 0 \leq r^{out} \leq 1\}$
		$S^4 = Vulnerability = \{r^{in}\}$	$D^4 = \{r^{in} \in R \vert 0 \leq r^{in} \leq 1\}$
		$S^5 = Incubation - period$	$D^5 = \{\Delta t^E \in R \mid 0 < \Delta t^E \leq T\}$
		$S^6 = Infectious - period$	$D^6 = \{\Delta t^I \in R \mid 0 < \Delta t^I \leq T\}$
		$S^7 = \{Symptomatic - state\}$	$D^7 = \{Symp, \ non - Symp\}$
		$S^8 = \{diagnosis \ state\}$	$D^7 = \{diagnosed, \ not - diagnosed\}$
	$E(i,j) =$ Edges = contacts	$l^1 = \{contact \ type\}$	$l^1 = \{home, \ school, \ work, \ shopping, \ others\}$
		$l^2 = \{contact \ duration\}$	$l^2 = \{w(i,j) \in N \mid 0 < w(i,j) \leq T\}$

Table A.1. Graphical Discrete Dynamical System (GDDS) components: possible state categories and their corresponding domain, examples of different functions including transition, vertex modification, and edge modification functions.

Y_k is all observations received up to time k and including y_k : $Y_k = \{y_i, i = 1, ..., k\}$.

proof. Based on the Bayes' theorem, the conditional probability function of $p(x_k|Y_k)$ could be written like this:

$$p(x_k|Y_k) = \frac{P(x_k, Y_k)}{p(Y_k)}$$

by splitting the $Y_k = \{y_k, Y_{k-1}\}$ we have:

$$p(x_k|Y_k) = \frac{P(x_k, Y_k)}{p(Y_k)} = \frac{P(x_k, y_k, Y_{k-1})}{p(y_k, Y_{k-1})}$$

Using the base rule again, we have:

$$p(x_k|Y_k) = \frac{P(x_k, y_k|Y_{k-1}) \times p(Y_{k-1})}{p(y_k|Y_{k-1}) \times p(Y_{k-1})}$$

Using the base rule again to convert the joint probability function of $P(x_k, y_k)$ to the conditional probability function, we have:

$$p(x_k|Y_k) = \frac{P(y_k|x_k, Y_{k-1}) \times p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})}$$

GDDS components		**Possible values in Epifast simulator**	
$Functions(F)$	f:local transition function	$f(u,v)$: stochastic transition function based on probability of transition of disease from infectious vertex u to susceptible vertex v, based on infectivity of node u, vulnerability of node v, and the duration of contact between u, v	$p(u,v) = 1 - (1 - r_u^{out}.r_v^{in})^{w(u,v)}$
	f^V: vertex	f_1^V : Initial infecting seeding	$D^2 = \{S,\ E,\ I,\ R\}$
		f_2^V: within-host transition of SEIR state	Changing the status of vertices from exposed to infectious based on value of $S^5 = \{\Delta t^E\}$ and exposed time, or from infectious state to removed based on $S^6 = \{\Delta t^I\}$ and infected time
	modification functions	f_3^V: diagnosis	Diagnose symptomatic vertices randomly, and change their S^8 state to diagnosed
		f_4^V : PI antiviral $C = \{> 1\%$ of population are infected$\}$	Administer antivirals to selected diagnosed symptomatic vertices to reduce their infectivity
		f_5^V : PI vaccination	Policy based intervening: vaccinate targeted vertices to decrease their vulnerability and infectivity
	f^E : Edge	f_1^E : NPI: school closure sp = school-age children, $C = \{>$ 1% of population are infected$\}$	A as action: remove in-school edges between them. sp = school-age children
	modification functions	f_2^E : NPI: home isolation sp = diagnosed vertices, C ={vertex is symptomatic, and diagnosed}	A: do natural isolation. Stay at home. Remove or reduce the contact duration with non-household members. sp =diagnosed vertices

Table A.2. Graphical Discrete Dynamical System (GDDS) components, Continued.

Algorithm 1- Procedure of the disease propagation loop in GDDS model of Epifast

for $t = 0$ **to** T **do**

- let $\{g_t^1, g_t^2, .., g_t^{j_t} \mid g_t^i \in F\}$ be the sequences of the functions that are considered for time step t:
- apply sequentially all g_t^i functions where $g_t^i \in \{f^V \cup f^E \cup f^I\}$ /* apply interventions */
  - modify vertex states or edge labels sequentially based on individuals'/policies adaptions
- for each vertex $v_i \in V$ do in parallel:
  - apply functions $g_t^i \in \{f\}$ and change the health state of each individual /* disease spread */
- Check the current status of the whole system & compute new interventions that are triggered.
- Add new interventions to the set of functions $\{g_{t+1}^i\}$

given the value of x_k, $P(y_k|x_k, Y_{k-1})$ become independent from the Y_{k-1}, so we have:

$$p(x_k|Y_k) = \frac{P(y_k|x_k) \times p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})}$$

A.3 Ebola Challenge Dataset

The dataset we used for modeling and calibrating Ebola disease parameters was provided for the Ebola Challenge [70] organized under the Research and Policy for Infectious Disease Dynamics (RAPIDD) program at the NIH. This Ebola disease dataset was generated by a previously published agent-based model [80] calibrated for Liberia under four different scenarios. These scenarios emulate various epidemiological changes, intervention reactions, and availability of data in the real world. Epidemic data for each scenario was released in five time points, and each time point contained outbreak situation reports, weekly reported new Ebola Virus Disease (EVD) cases at the county and country level, and current/future intervention plans to prevent and fight the outbreak. The following sections contains the detailed reports of first and second data points of the first scenario.

A.3.1 Scenario 01: Time point 1

In this data-point prediction week is set to week 13^{th}, and the following information are reported:

- The first confirmed cases of Ebola virus disease(EVD) was reported on day 1 in Gbarpolu.
- A total number of 324 EVD confirmed and probable cases was reported till week 13 across Liberia, 203(the majority) of the cases were reported in Grand Cape Mount, 28 cases in Gbarpolu, 1 case in Margibi, 79 cases in Bomi, 1 case in Lofa, 3 cases in Montserrado, 9 cases in Bong. So far the reported cases are only found in the counties in the northwest of Liberia.
- A total number of 178 deaths were reported till week 13 across Liberia, among which 116 were in Grand Gedeh, 16 in Gbarpolu, 43 in Bomi, 3 in Bong.
- A total number of 13 health care workers(HCW) confirmed and probable cases were reported, 11 in Grand Cape Mount, 2 in Bong. 6 of them were among the fatalities count up to week 13.
- Traditional burial practices were observed among all communities which involves washing/touching/kissing the bodies during ceremony. The practices are confirmed to contribute to the spreading of EVD. No effective safe burial protocol has been enforced till week 13.
- No contact tracing cases were followed till week 13 No Ebola Treatment Unit(ETU) were opened till week 13. No plans of opening ETUs in near future.

A.3.2 Scenario 01: Time point 2

In this data-point prediction week is set to week 20^{th}, and the following information are reported:

- A total number of 954 EVD confirmed and probable cases was reported from week 13 till week 20 across Liberia, 415 (the majority) of the cases were reported in Grand Cape Mount, 38 cases in Gbarpolu, 15 cases in Grand Bassa, 92 cases in Margibi, 3 cases in Nimba, 133 cases in Bomi, 20 case in Lofa, 67 cases in Montserrado, 171 cases in Bong. The disease has extended to county Grand Bassa and Nimba.
- A total number of 614 deaths were reported from week 13 till week 20 across Liberia, among which 285 were in Grand Cape Mount, 29 in Gbarpolu, 5 in Grand Bassa, 63 in Margibi, 2 in Nimba, 99 in Bomi, 13 in Lofa, 27 in Montserrado, 91 in Bong.

- A total number of 40 health care workers(HCW) confirmed and probable cases were reported, 15 in Grand Cape Mount, 3 in Margibi, 12 in Montserrado, 10 in Bong. 19 of new deaths of HCWs were reported among the fatalities count from week 13 up to week 20. Traditional burial practices were observed among all communities which involve washing/touching/kissing the bodies during ceremony. The practices are confirmed to contribute to the spreading of EVD. No effective safe burial protocol has been enforced as of week 20. No contact tracing cases were followed as of week 20. No Ebola Treatment Unit (ETU) was opened as of week 20. However, there is a plan for the opening of the following facilities between week 20 and week 24:
  - An ETU at Grand Cape Mount with 60 beds and 45 HCWs will begin operating sometime between week 20 and week 24.
  - An ETU at Montserrado with 55 beds and 60 HCWs will begin operating sometime between week 20 and week 24. An ETU at Bong with 35 beds and 55 HCWs will begin operating somtime between week 20 and week 24.

A.3.3 Scenario 01: Time point 3

In this data-point prediction week is set to week 26^{th}, and the following information are reported:

- A total number of 1842 EVD confirmed and probable cases was reported from week 20 till week 26 across Liberia, 553 cases were reported in Grand Cape Mount, 53 cases in Gbarpolu, 27 cases in Grand Bassa, 97 cases in Margibi, 81 cases in Nimba, 240 cases in Bomi, 63 case in Lofa, 1 case in Rivercess, 442 cases in Montserrado, 285 cases in Bong.
- A total number of 1341 deaths were reported from week 20 till week 26 across Liberia, among which 430 were in Grand Cape Mount, 39 in Gbarpolu, 30 in Grand Bassa, 79 in Margibi, 39 in Nimba, 178 in Bomi, 35 in Lofa, 1 in Rivercess, 295 in Montserrado, 215 in Bong.
- A total number of 81 health care workers(HCW) confirmed and probable cases were reported, 52 in Grand Cape Mount, 2 in Margibi, 17 in Montserrado, 8 in Bong. 53 of new deaths of HCWs were reported among the fatalities count from week 13 up to week 20.
- An effective safe burial protocol has started being enforced as of week 21.
- Community engagement and social mobilization programs as well as the distribution of home disinfection kits are starting in several parts of the country. It is not yet possible to estimate the effect of these interventions.

- A total number of 10292 individuals has been contact traced from week 20 till week 26. Among which 2830 were in Grand Cape Mount, 435 in Gbarpolu, 146 in Grand Bassa, 480 in Margibi, 604 in Nimba, 1125 in Bomi, 358 in Lofa, 2727 in Montserrado, 1587 in Bong.
- Here is a list of ETUs in operation:
  - An ETU at Grand Cape Mount with 60 beds and 45 HCWs began operating since week 20.
  - An ETU at Montserrado with 55 beds and 60 HCWs began operating since week 22.
  - An ETU at Bong with 35 beds and 55 HCWs began operating since week 24.
- Here is a list of ETUs in plan for opening:
  - An ETU at Nimba with 20 beds and 30 HCWs will begin operating sometime between week 25 and week 27
  - An ETU at Grand Bassa with 30 beds and 35 HCWs will begin operating sometime between week 27 and week 29

A.3.4 Scenario 01: Time point 4

In this time point, prediction week is set to week 35^{th}, and the following information are reported:

- A total number of 1778 EVD confirmed and probable cases was reported from week 26 till week 35 across Liberia, 200 cases were reported in Grand Cape Mount, 29 cases in Gbarpolu, 83 cases in Grand Gedeh, 32 cases in Grand Bassa, 2 cases in Grand Kru, 159 cases in Margibi, 160 cases in Nimba, 171 cases in Bomi, 70 case in Lofa, 6 case in River Gee, 675 cases in Montserrado, 200 cases in Grand Cape Mount, 8 cases in Maryland, 170 cases in Bong, 13 cases in Sinoe.
- A total number of 1557 deaths were reported from week 26 till week 35 across Liberia, among which 221 were in Grand Cape Mount, 28 in Gbarpolu, 52 cases in Grand Gedeh, 31 cases in Grand Bassa, 2 cases in Grand Kru, 115 cases in Margibi, 134 in Nimba, 168 in Bomi, 68 in Lofa, 2 in River Gee, 573 in Montserrado, 6 in Maryland, 152 in Bong, 5 in Sinoe.
- A total number of 82 health care workers(HCW) confirmed and probable cases were reported, 6 in Grand Cape Mount, 20 in Grand Gedeh, 1 in Grand Bassa, 1 in Margibi, 18 in Nimba, 36 in Montserrado. 79 of new deaths of HCWs were reported among the fatalities count from week 26 up to week 35.

- An effective safe burial protocol was enforced from week 26 till week 35.
- Community engagement and social mobilization programs as well as the distribution of home disinfection kits are starting in several parts of the country. It is not yet possible to estimate the effect of these interventions, although it is certainly moderate.
- A total number of 6930 individuals has been contact traced from week 26 till week 35. Among which 675 were in Grand Cape Mount, 166 in Gbarpolu, 392 in Grand Gedeh, 87 in Grand Bassa, 10 in Grand Kru, 760 in Margibi, 595 in Nimba, 809 in Bomi, 307 in Lofa, 38 in River Gee, 2429 in Montserrado, 24 in Maryland, 588 in Bong, 50 in Sinoe.
- All previous opened ETUs are still in operation with full capacity. Here is a list of new ETUs in operation:
  - An ETU at Nimba with 20 beds and 30 HCWs has started operations at week 27.
  - An ETU at Grand Bassa with 30 beds and 35 HCWs has started operations at week 29.
- Here is a list of ETUs in plan for opening:
  - An ETU at Montserrado with 95 beds and 125 HCWs will be opening sometime between week 35 and week 37.
  - An ETU at Grand Bassa with 50 beds and 65 HCWs will be opening sometime between week 35 and week 37.

A.3.5 Scenario 01: Time point 5

In this time point, prediction week is set to week 42^{th}, and the following information are reported:

- A total of 707 EVD confirmed and probable cases were reported from week 35 to week 42 (included) across Liberia; 50 cases were reported in Grand Cape Mount, 25 cases in Gbarpolu, 42 cases in Grand Gedeh, 7 cases in Grand Bassa, 129 cases in Margibi, 27 cases in Nimba, 90 cases in Bomi, 3 case in Lofa, 2 case in Rivercess, 5 cases in River Gee, 205 cases in Montserrado, 92 cases in Bong, 30 cases in Sinoe.
- There is consistent evidence from the field that the epidemic is steadily decreasing in most of the counties. ETUs occupancy is estimated at about 40% in most of the facilities.
- A total number of 662 deaths were reported from week 35 till week 42 across Liberia, among which 50 were in Grand Cape Mount, 19 in Gbarpolu, 50 cases in Grand Gedeh,

6 cases in Grand Bassa, 96 cases in Margibi, 36 in Nimba, 81 in Bomi, 5 in Lofa, 2 in Rivercess, 7 in River Gee, 203 in Montserrado, 78 in Bong, 29 in Sinoe.

- A total number of 9 health care workers(HCW) confirmed and probable cases were reported, 2 in Grand Gedeh, 1 in Nimba, 4 in Montserrado. 2 in Bong. 10 of new deaths of HCWs were reported among the fatalities count from week 35.
- An effective safe burial protocol was enforced from week 35 till week 42. It is estimated that close to 90% of the burial are safe and dignified.
- Community engagement and social mobilization programs as well as the distribution of home disinfection kits are continuing in several parts of the country. It is not yet possible to estimate the effect of these interventions, although it is certainly moderate.
- A total number of 2266 individuals has been contact traced from week 35 till week 42. Among which 110 were in Grand Cape Mount, 103 in Gbarpolu, 223 in Grand Gedeh, 10 in Grand Bassa, 508 in Margibi, 60 in Nimba, 346 in Bomi, 31 in Lofa, 14 in Rivercess, 19 in River Gee, 534 in Montserrado, 256 in Bong, 52 in Sinoe.
- All previous opened ETUs are still in operation with full capacity. Here is a list of new ETUs in operation:
  - An ETU at Montserrado with 95 beds and 125 HCWs has started operations at week 35.
  - An ETU at Grand Bassa with 50 beds and 65 HCWs has started operations at week 36.
- There is no plan for the opening of new ETUs in the next few weeks.

A.4 Supporting Tables

	Domain (days)	**Ground truth distribution**
Incubation Period	{1, 2, 3, 4}	$\{p(1) : 0.06, p(2) : 0.24, p(3) : 0.25, p(4) : 0.45\}$
Infectious Period	{2, 3, 4, 5}	$\{p(2) : 0.2, p(3) : 0.21, p(4) : 0.24, p(5) : 0.35\}$

Table A.3. Probability distribution of incubation and infectious periods used for simulating flu disease dynamics. $p(x)$ denotes the probability that the incubation/infectious period is equal to x days.

A.5 Supporting Figures

Estimated Parameters	**Notation**	**Value**	Description
Natural isolation efficacy	β_{NI}	0.7	Reduce 70% of all non-house hold connections.
Hospitalization Duration	-	21 (days)	Action Duration of Hospitalization
Behavioral intervention efficacy	β_{BI}	0.47	Indicates the effect of behavioral intervention which simulates the natural behavior reactions.
Behavioral reaction compliance	C_{BI}	0.585	Adjusts the human obedience in following behavior reactions.
Safe burial efficacy	$beta_{SB}$	0.1	Reduces the transmission rate of those who comply.
Safe burial compliance	C_{SB}	0.7	Indicates the ratio of population that comply safe burial intervention.
Contact tracing efficacy	β_{CT}	0.3	multiplies the transmissiblity of those agents who receive it by β_{CT}.
Contact tracing duration	-	21 (days)	Action duration.
Calibrated Parameters	**Notation**	**Range**	Description
No. of initial infections	I_0	$I_0 \in \mathbb{N}$, $I_0 \in [1, 10]$	Number of initial infections at week 1.
Transmission rate	β	$\beta \in \mathbb{R}$, $\beta \in [3.5, 7] \times 1e-5$	-
Travel reduction ratio	ω_{TR}	$\omega_{TR} \in \mathbb{R}$, $\omega_{TR} \in [0, 1.2]$	
Hospitalization efficacy	β_{HE}	$\beta_{HE} \in \mathbb{R}$, $\beta_{HE} \in [0.1, 0.8]$	The transmission rate is multiplied by β_{HE} for those who are admitted to hospitals.
Hospitalization delay	β_{HD}	$\beta_{HD} \in \mathbb{N}$, $\beta_{HD} \in [2, 10]$	Adjusts the delay of being admitted to hospitals.
Ebola treatment units' efficacy	β_{ETU}	$\beta_{ETU} \in \mathbb{R}$, $\beta_{ETU} \in [0.05, 1]$	The transmission rate is multiplied by β_{ETU} for those who are admitted in ETUs.

Table A.4. The range of estimated and calibrated parameters utilized to model the disease propagation and interventions of Ebola epidemic.

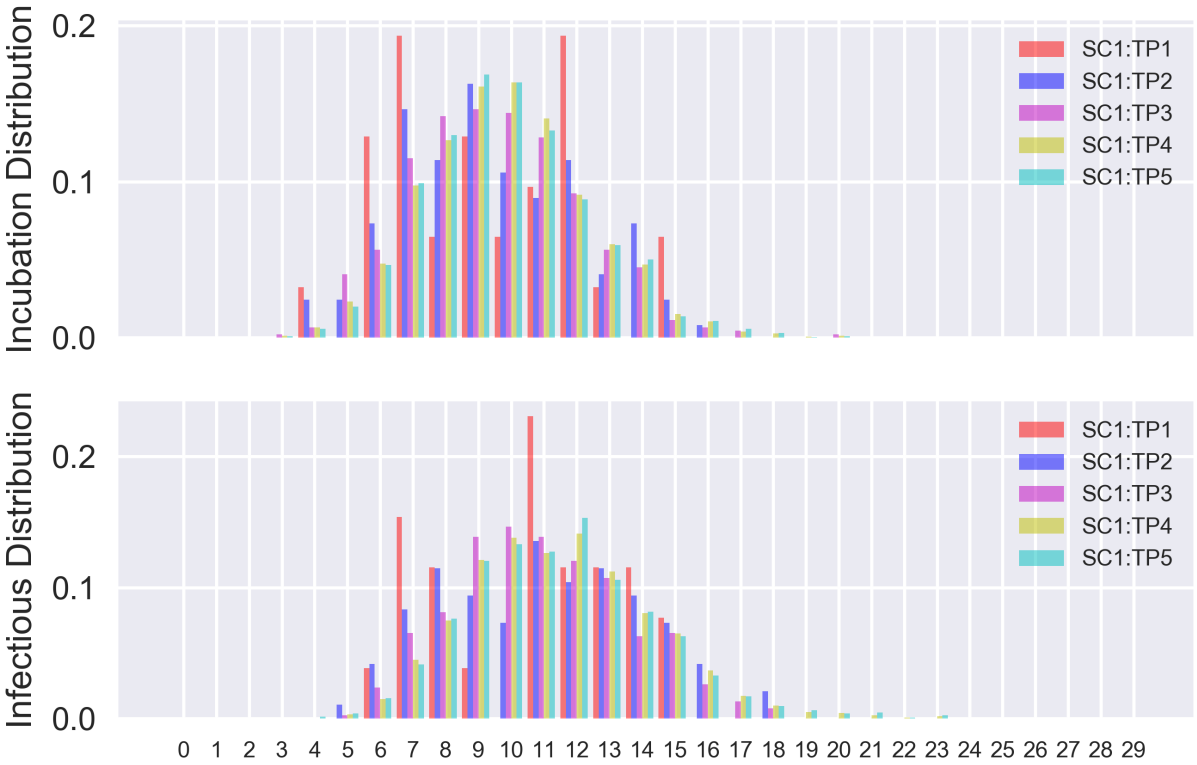


Figure A.1. Probability distribution of incubation and infectious periods used for simulating Ebola disease dynamics, for time points 1 to 5. The numbers on the X axis indicates the period in terms of days, and the numbers on the Y axis shows the probability corresponds to each period. SC1 indicates that these settings are used for the first scenario of the Ebola challenge.

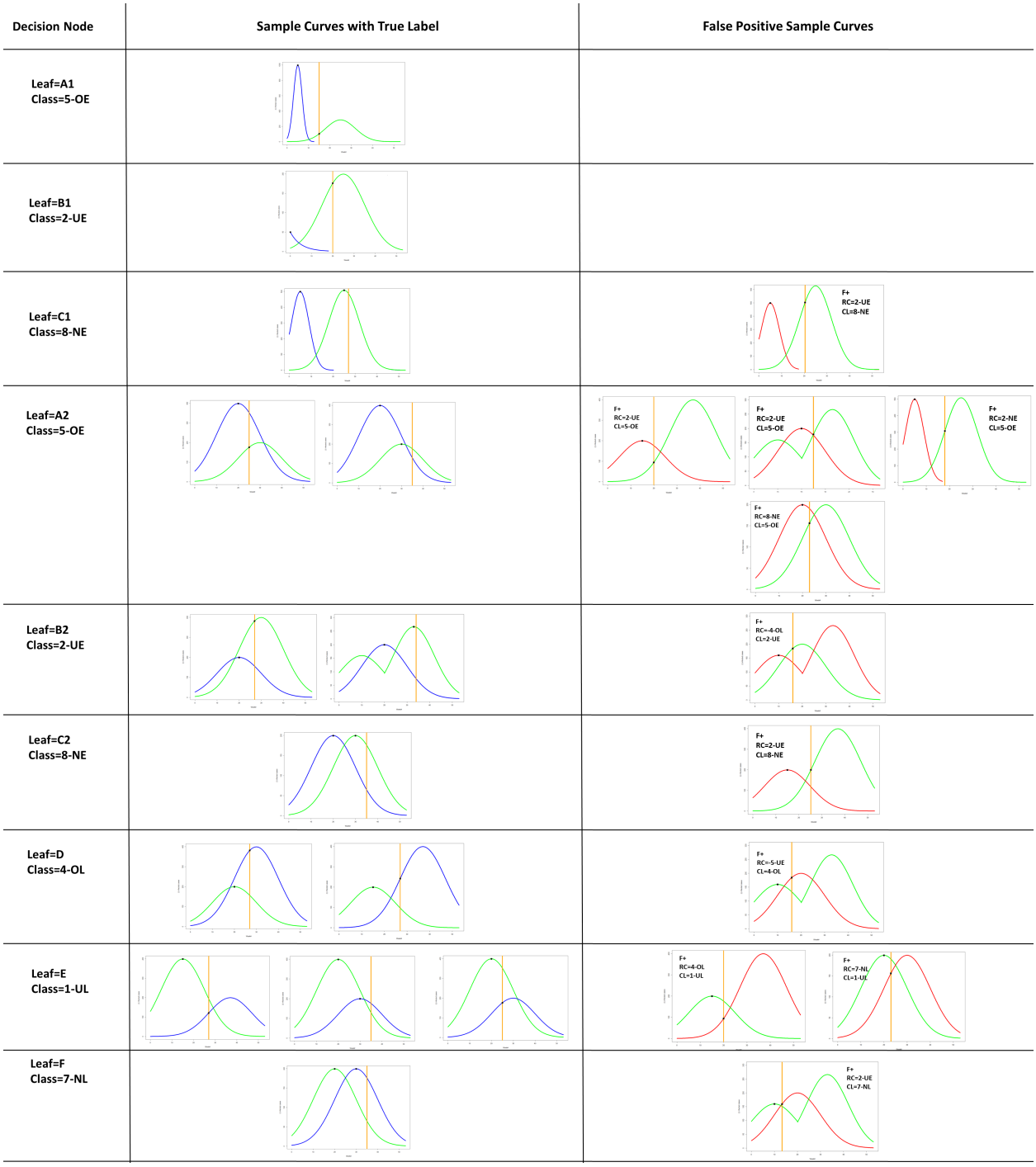


Figure A.2. Dataset of the sample curves, part A: Sample curves from different categories of classifier that satisfy various conditions of the tree-structure (2). True samples represent those predicted curves that are classified correctly, and false positive samples are the possible curves that could be classified incorrectly due to lack of observation data.

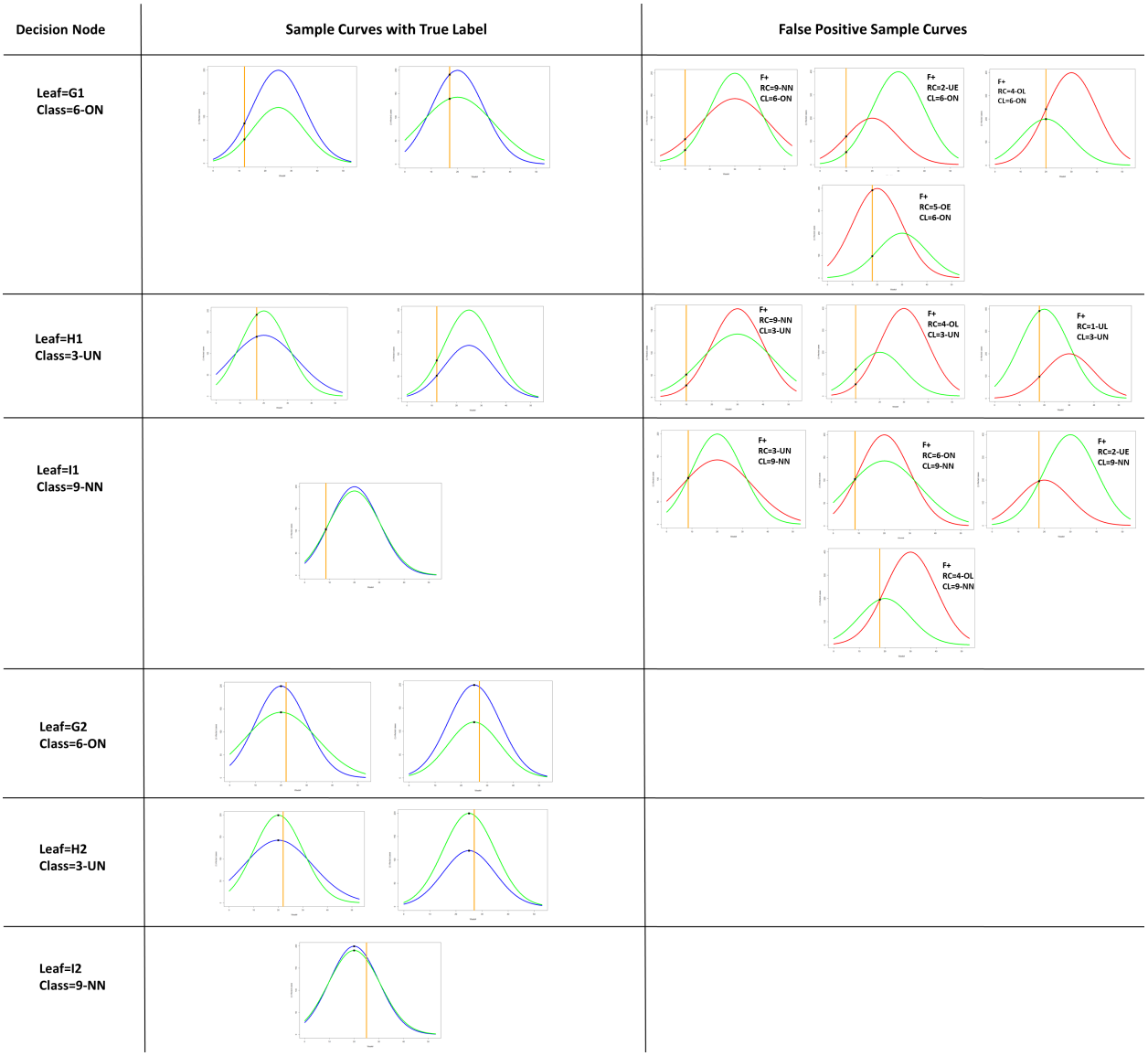


Figure A.3. Dataset of the sample curves, part B: Sample curves from different categories of classifier that satisfy various conditions of the tree-structure (2). True samples represent those predicted curves that are classified correctly, and false positive samples are the possible curves that could be classified incorrectly due to lack of observation data.

Appendix B

Chapter 5: Appendix

B.1 Forecasting Algorithms

In our model, a digital library is initially built from epidemic curves which are simulated or obtained from past epidemics or data-assimilation-based methods. Simulated epidemic curves could be generated by different kinds of epidemiology models like simple ODE models (SEIR) [101, 102], or individual-based models (EpiFast) [103, 29]. In this paper, the digital library contains simulated epidemics curves generated by running the stochastic version of SEIR model [102] with different parameters to keep the variability in the system. The Susceptible-Exposed-Infectious-Removed (SEIR) model is usually represented by a system of differential equations. To take into account the stochastic nature of infectious disease transmission and to facilitate a computer simulation to produce daily (or weekly) case counts, the SEIR model has been extended to a stochastic discrete time SEIR model. In the stochastic SEIR, daily changes of compartment sizes are sampled from binomial distributions with corresponding probability parameters determined by transmission rate, the mean incubation duration, and the mean infectious duration. Each individual is placed in one, and only one, of the following four states at one time step: Susceptible, Exposed, Infectious, and Removed. The details can be found in [102].

We have implemented the stochastic SEIR model in a C++ simulation code which takes initial sizes of the compartments and the aforementioned disease-related parameters and generates the daily number of infections which can be aggregated to weekly case counts. We would like to emphasize that our Epi-Evaluator works for similar time-series prediction data that may be generated by any other method, either a statistical method or an individual based SEIR model like [29, 32].

To generate our predicted epidemic curve, we pre-select 2205 different models, each being a combination of transmission rate, mean incubation duration, and mean infectious duration. The transmission rate ranges from 0.1E-5 to 0.5E-5; the mean incubation duration ranges from 0.9 to 2.9 (days); the mean infectious duration ranges from 3.1 to 5.1 (days). The initial

size of the infectious compartment is set depending on the surveillance data from CDC for the specific HHS region. All 2205 models and corresponding epidemic curves are stored in a digital library. In the forecasting process, given a part of the epidemic curve as a time-series from the surveillance data, the algorithm tries to compare it with all the curves stored in the digital library and find the best match. The tail of the best-matched curve could be taken as the predicted epidemic curve, and its corresponding parameters are used to model the outbreak and predict probable future events about the epidemic (Figure B.1).

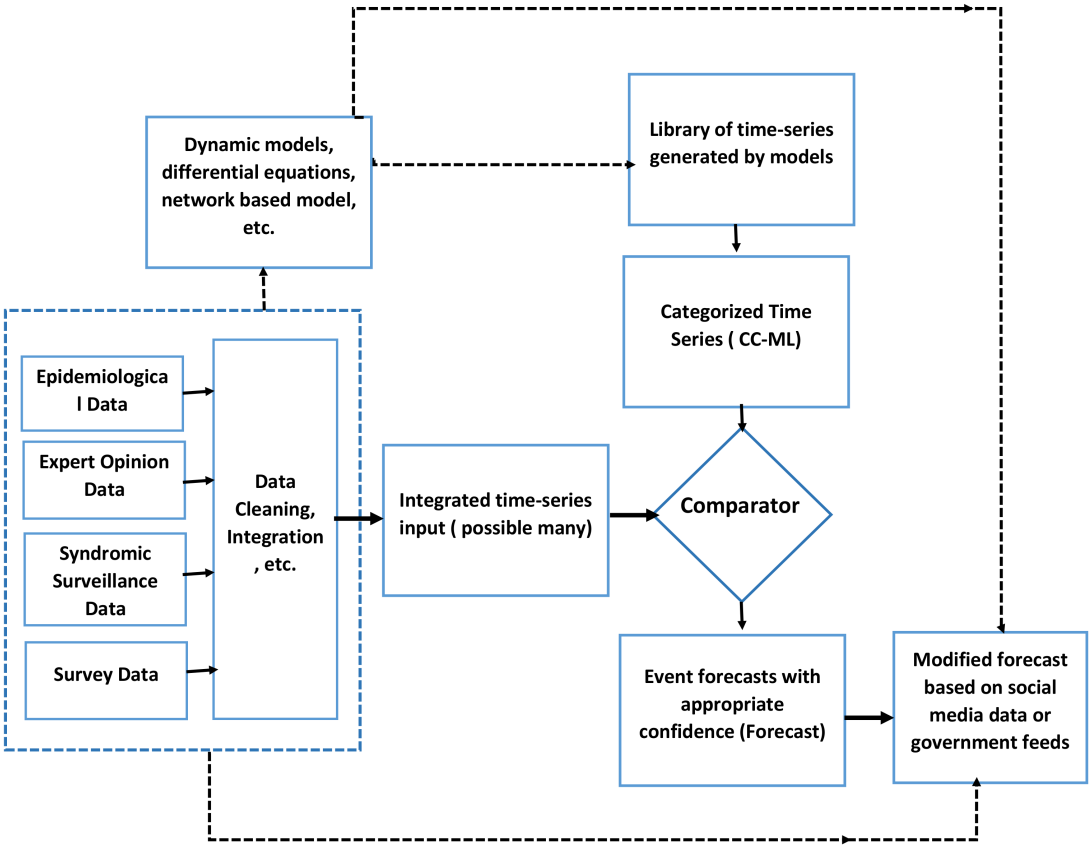


Figure B.1. Summary of Methodology, describing the forecasting pipeline.

To compare a surveillance epidemic curve with the library's curves and find the best-matched parameters, we should calculate the distance between the curves. We use the following measure:

$$dist\left(y, \bar{Z}_P\right) = \sqrt{\sum_{j=1}^{t} d^2\left(y\left(j\right), \bar{Z}_{P^*}\left(j\right)\right)} \tag{B.1}$$

where j is the time step and t indicates the prediction time on which the tail of epidemic

curve is predicted; $\bar{Z}_{P^*}$ is the simulation curve in the digital library generated by running SEIR model with parameters P^*, $y(j)$ is the observed value from surveillance data and d is a distance function which calculates the distance between the data points (See [35] for more details).

The goal of this paper is applying the Epi-features and Error Metrics to the output of the forecasting methods to evaluate their performance and compare them together. To achieve this goal, we consider six different configurations for our algorithm to generate different results and assess them by Epi-features and Error Metrics to find the best configuration. The applied configurations are as follows:

- Configuration 1: Jaccard distance function & No trimming.
- Configuration 2: Euclidean distance function & No-trimming.
- Configuration 3: Jaccard distance function & Automatic Trimming
- Configuration 4: Euclidean distance function & Automatic Trimming
- Configuration 5: Jaccard distance function & Ad-hoc Trimming
- Configuration 6: Euclidean distance function & Ad-hoc Trimming

To compare a surveillance epidemic curve with the library's curves, the configurations 1, 3, and 5 applies Jaccard distance function as the distance function used in equation B.1 while other configurations used Euclidean distance function. Table B.1 demonstrates the definition of Jaccard and Euclidean distance functions. The term "Trimming" which is used in some configurations, refers to the fixing library curves to follow the initially observed data. As noted, the comparator module matches each epidemic curve in the digital library with a given observed curve and determines the best match. Ideally, all the epidemic curves in the digital library should have the same base count as that of a given observed epidemic curve. The base count refers to the number of infected case counts in the first week of epidemic (I_0). However, generating a library for every new initial seed would be time-consuming. Therefore, as an alternative, the Comparator module trims the left portion of digital library curves so that initial count of the input curve (observed data) is less than or equal to a section of digital library's epidemic curves against which comparison distances are computed. We call this process automatic trimming which searches for the trimming point which is the closest value to the base count and trims the head of the epidemic curve. However, we figured out that usually, the best trimming point is the 5th week or near that. Therefore, we suggested the ad-hoc trimming strategy in which only five data points are trimmed to skip the search time. The trimming value 5 is achieved experimentally and heuristically and has shown acceptable results.
The combination of different trimming strategies with various distance functions provide a variety of configurations which results in different outputs for forecasting algorithm.

Jaccard	Euclidean
$d_{Jac} = \frac{\sum_{t=1}^{n}(x_i - y_i)^2}{\sum_{t=1}^{n} x_i^2 + \sum_{t=1}^{n} y_i^2 - \sum_{t=1}^{n} x_i \times y_i}$	$d_{Euc} = \sqrt{\sum_{t=1}^{n} \lvert x_i - y_i \rvert^2}$

Table B.1. Definition of different Distance Functions

B.2 Observations/Proofs on the eliminated error measures

As mentioned before, we have selected MAE, RMSE, MAPE, sMAPE, MdAPE and MdsAPE as the error measures for evaluating the Epi-features and ignored others based on different reasons. Some observations on the eliminated error measures and their relationships with the considered ones are as follow:

- **Observation 1:** The magnitude of Normalized Mean Squared Error (NMSE) between an observed time-series and a predicted one is monotonic of the magnitude of Root Mean Squared Error (RMSE).
  Proof: Normalized Mean Squared Error scales the average squared error by the variance of observed data which is always a fixed number and doesn't change by forecasting errors. As the square root function is monotonically increasing, the NMSE is a monotonic function of RMSE.

- **Observation 2**: The magnitude of Mean Absolute Scaled Error (MASE) between an observed time-series and a predicted one is proportional to the magnitude of Mean Absolute Error (MAE).
  Proof: MASE scales absolute error by the average error of one-step Random walk method that is the average differences of sequential data points of the observed time-series ($\frac{1}{n-1} \times \sum_{i=2}^{n} |y_i - y_{i-1}|$). As the denominator is always fixed for each observed time-series, the MASE is always proportional and monotonic to MAE.

$$MASE = \frac{1}{n}\sum_{t=1}^{n} \left| \frac{e_t}{\frac{1}{n-1} \times \sum_{i=2}^{n} |y_i - y_{i-1}|} \right| \propto \frac{1}{n}\sum_{t=1}^{n} |e_t| = MAE \tag{B.2}$$

- **Observation 3**: The magnitude of Mean Absolute Relative Error(MARE)and Relative Measures (RelMAE) between Epi-features obtained from the observed and predicted time-series is proportional to the magnitude of Mean Absolute Error (MAE).
  Proof: Mean Absolute Relative Error(MARE) scales the error in each horizon (e_i) with the corresponding error achieved by Random walk method (e_{RWi}). In order to achieve the Epi-features from the Random walk results, Seasonal-Adjusted Random

Walk method is a better option which generates the entire seasonal epidemic curve. However, the prediction of strongly Seasonal Random Walk (seasonal random walk without noise) is independent of the prediction time and generates one unique curve for the remainder of the time-series. Therefore, the Epi-features calculated for Strong Seasonal Random Walk is a constant value and independent of the prediction time which means $e_{RWi} = e_{RW} = \alpha$. Even Seasonal Random Walk prediction with noise is also independent of the prediction time and the obtained Epi-features has a fixed mean value with random noise. Consequently, the ranking achieved by Mean Absolute Relative Error(MARE) with relation to strongly seasonal adjusted random walk method is monotonic with the ranking obtained by simple Mean Absolute Error (MAE). The same reasoning could be used to prove the lemma for Relative Measures (RMAE).

$$MARE = \frac{1}{n}\sum_{t=1}^{n}|\frac{e_t}{e_{RWt}}| = \frac{1}{n}\sum_{t=1}^{n}|\frac{e_t}{\alpha}| \propto MAE \tag{B.3}$$

$$RMAE = \frac{MAE}{MAE_{RW}} = \frac{\sum_{t=1}^{n}|e_t|}{\sum_{t=1}^{n}|e_{RWt|}}| = \frac{1}{n \times \alpha}\sum_{t=1}^{n}|e_t| \propto MAE \tag{B.4}$$

- **Observation 4**: Geometric Mean of the Relative Absolute Error (GMRAE) is proportional to the magnitude of Geometric Mean of Absolute Error (GMAE)
  Proof: Using the same reasoning discussed in Observation 3 we have:

$$GMRAE = \left(\prod_{i=1}^{n}|\frac{e_t}{e_{RWt}}|\right)^{(1/n)} = \left(\prod_{i=1}^{n}|\frac{e_t}{\alpha}|\right)^{(1/n)} \propto GMAE \tag{B.5}$$

We have eliminated the Percent Better (PB) from the pool of measures because it has low sensitivity to reveal the effect of change in methods and parameters [96]. MAAPE is the arctangent of Absolute Percentage ratio to solve the problem of division by zero by mapping the undefined infinity values of percentage error to $\Pi/2$ which is not informative and doesn't discriminate the small and large prediction errors from each other.

B.3 Supplement Tables

B.4 Supplement Figures

Error Measure name	Formula	Description
Mean Percentage Error (MPE_t)	$MPE = \frac{1}{n}\sum_{t=1}^{n}[\frac{e_t}{y_t}]$	Similar to MAPE but sign of error demonstrates the direction of error
Signed Mean Squared Error (SMSE)	$SMSE = \frac{1}{n}\sum_{t=1}^{n}[\frac{e_t}{\lvert e_t\rvert}] \times e_t^2$	Similar to MSE but provides the direction of the error
Median Absolute Percentage Error ($MdAPE$)	Median Observation of APE	where Observations are sorted APE_t , t is time horizon.
Root Mean Square Percentage Error ($RMSPE$)	$RMSPE = \sqrt{[\frac{1}{n}\sum_{t=1}^{n}[\frac{e_t}{y_t}]^2]}$	Penalizes large errors.
symmetric Median Absolute Percentage Error ($sMdAPE$)	Median Observation of $sAPE$	where Observations are sorted $sAPE_t$, t is time horizon.
Relative Mean Absolute Error ($RMAE$)	$RMAE = \frac{MAE}{MAE_{RW}}$	Measures the ratio of mean absolute error to Random walk error across time horizons.
Geometric Mean Relative Absolute Error ($GMRAE$)	$GMRAE = [\prod_{t=1}^{N}\lvert RAE_t\rvert]^{1/N}$	Measures the average ratio of relative absolute error to Random walk error
Relative Root Mean Square Error ($RelRMSE$)	$RelRMSE = \frac{RMSE}{RMSE_{RW}}$	Measures the ratio of RMSE to Random walk RMSE across time horizons.
Log Mean Squared Error Ratio, (LMR)	$LMR = log\left(\frac{RMSE}{RMSE_{RW}}\right)$	Measures log form of the ratio of RMSE to Random walk RMSE across time horizons.
Relative Geometric Root Mean Square Error ($RGRMSE$)	$RGRMSE = \frac{GRMSE}{GRMSE_{RW}}$	complex assessment of the relative geometric standard deviation.

Table B.2. Other forms of measure error measures.

	MAPE	sMAPE	RMSE	MdAPE	MdsAPE	MAE	Consensus Ranking
Method 1	5	5	5	2	1	5	3.83
Method 2	6	6	6	2	1	6	4.5
Method 3	3	4	4	2	1	3	2.83
Method 4	3	3	3	2	6	3	3.333
Method 5	1	2	1	1	1	1	1.167
Method 6	2	1	2	2	5	2	2.33

Table B.3. Ranking of methods for predicting peak time based on different error measures for Region 1 over whole season (2013-2014).

	MAPE	sMAPE	RMSE	MdAPE	MdsAPE	MAE	Consensus Ranking
Method 1	6	6	6	6	6	6	6
Method 5	5	5	5	5	5	5	5
Method 3	4	4	4	4	3	4	3.83
Method 4	1	1	2	1	1	1	1.167
Method 5	3	3	3	3	3	3	3
Method 6	1	1	1	2	2	2	1.5

Table B.4. Ranking of methods for predicting take-off value based on different error measures for Region 1 over whole season (2013-2014).

	MAPE	sMAPE	RMSE	MdAPE	MdsAPE	MAE	Consensus Ranking
Method 1	1	1	1	1	1	1	1
Method 2	2	2	2	2	2	2	2
Method 3	3	3	3	3	3	3	3
Method 4	5	5	5	5	5	5	5
Method 5	4	4	4	4	4	4	4
Method 6	6	6	6	6	6	6	6

Table B.5. Ranking of methods for predicting take-off time based on different error measures for Region 1 over whole season (2013-2014).

	MAPE	sMAPE	RMSE	MdAPE	MdsAPE	MAE	Consensus Ranking
Method 1	2	2	2	6	6	2	3.33
Method 2	1	1	1	1	1	1	1
Method 3	3	3	3	4	4	3	3.33
Method 4	5	5	5	2	2	5	4
Method 5	4	4	4	5	5	4	4.33
Method 6	6	6	6	2	2	6	4.67

Table B.6. Ranking of methods for predicting ID's length based on different error measures for Region 1 over whole season (2013-2014).

	MAPE	sMAPE	RMSE	MdAPE	MdsAPE	MAE	Consensus Ranking
Method 1	6	6	6	5	5	6	5.67
Method 2	5	5	5	3	3	5	4.33
Method 3	3	3	3	3	4	3	3.167
Method 4	1	1	1	1	1	1	1
Method 5	4	4	4	6	6	4	4.67
Method 6	2	2	2	2	2	2	2

Table B.7. Ranking of methods for predicting ID's start time based on different error measures for Region 1 over whole season (2013-2014).

	MAPE	sMAPE	RMSE	MdAPE	MdsAPE	MAE	Consensus Ranking
Method 1	6	6	6	5	6	6	5.83
Method 2	5	5	5	4	3	5	4.5
Method 3	3	4	3	2	4	3	3.167
Method 4	1	2	1	1	1	1	1.167
Method 5	4	3	4	6	4	4	4.167
Method 6	2	1	2	2	1	2	1.67

Table B.8. Ranking of methods for predicting Speed of Epidemic based on different Error Measures for Region 1 over whole season (2013-2014).

	MAPE	sMAPE	RMSE	MdAPE	MdsAPE	MAE	Consensus Ranking
Method 1	6	6	6	6	6	6	6
Method 2	5	5	5	5	5	5	5
Method 3	3	3	3	3	3	3	3
Method 4	1	1	1	1	1	1	1
Method 5	3	3	3	3	3	3	3
Method 6	1	1	1	1	1	1	1

Table B.9. Ranking of methods for predicting Start of Flu Season based on different Error Measures for Region 1 over whole season (2013-2014).

Error Metric	MAPE	MAPE	sMAPE	sMAPE
Curves' Color	Range	Domain	Range	Domain
	e=2	–	e=2	x=0
	e=1	x=0	e=1	$x = \frac{y}{3}$
	e=1/2	$x = \frac{y}{2}$	e=1/2	$x = \frac{3y}{5}$
	e=0	$x = y$	e=0	x=y
	e=1/2	$x = \frac{3y}{2}$	e=2	$x = \frac{5y}{3}$
	e=1	$x = 2y$	e=1/2	$x = 3y$
	e=2	$x = 3y$	e=2	$x \to \infty$
	$e \succ 2$	$x \succ 3y$	$e \succ 2$	–
	$e \to \infty$	$x \to \infty$	$e \succ 2$	–

Table B.10. Corresponding domains that generate equal MAPE or sMAPE errors in term of magnitude.

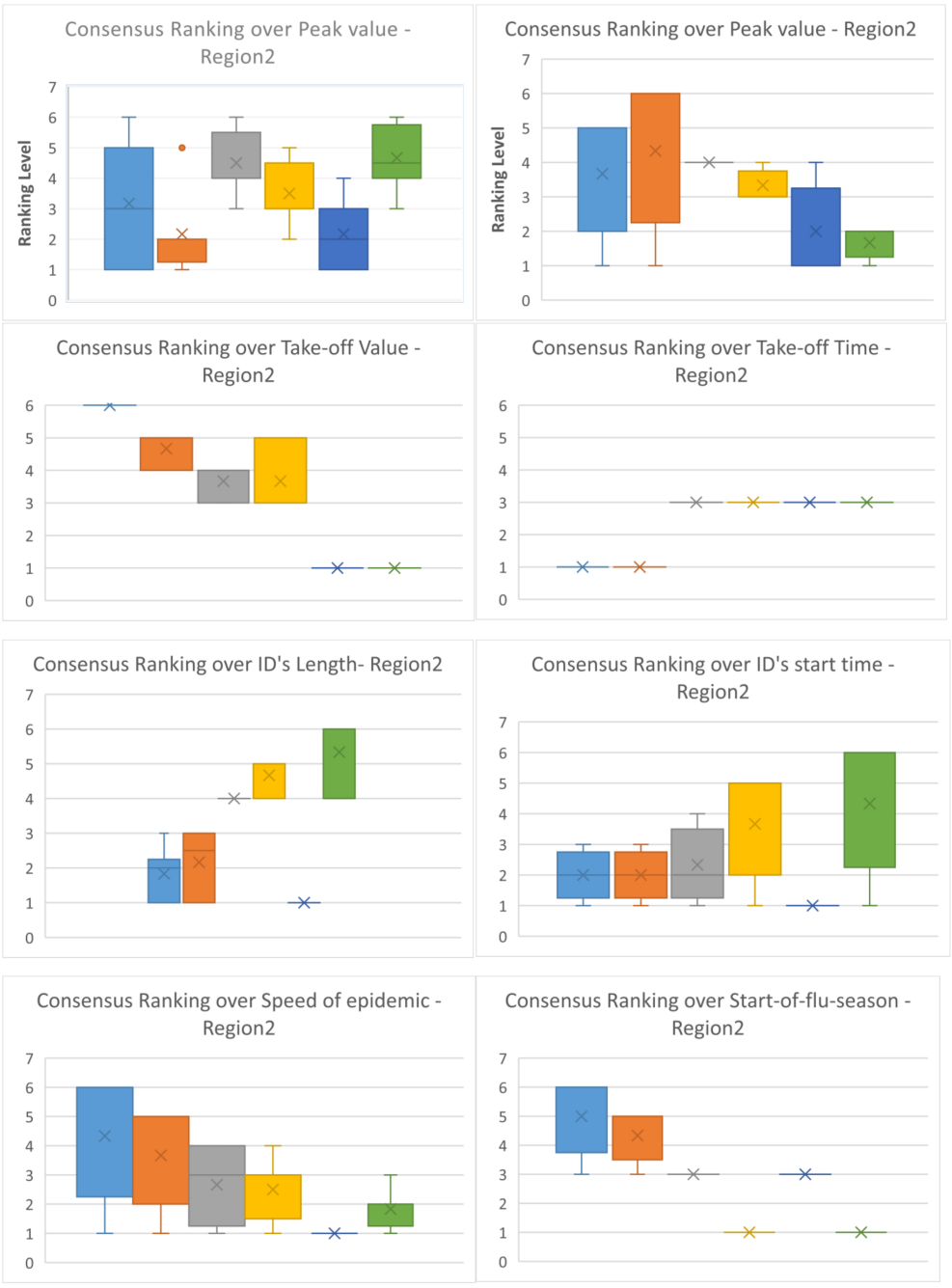


Figure B.2. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 2

Figure B.3. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 3

Figure B.4. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 4

Figure B.5. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 5

Figure B.6. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 6

Figure B.7. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 7

Figure B.8. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 8

Figure B.9. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 9

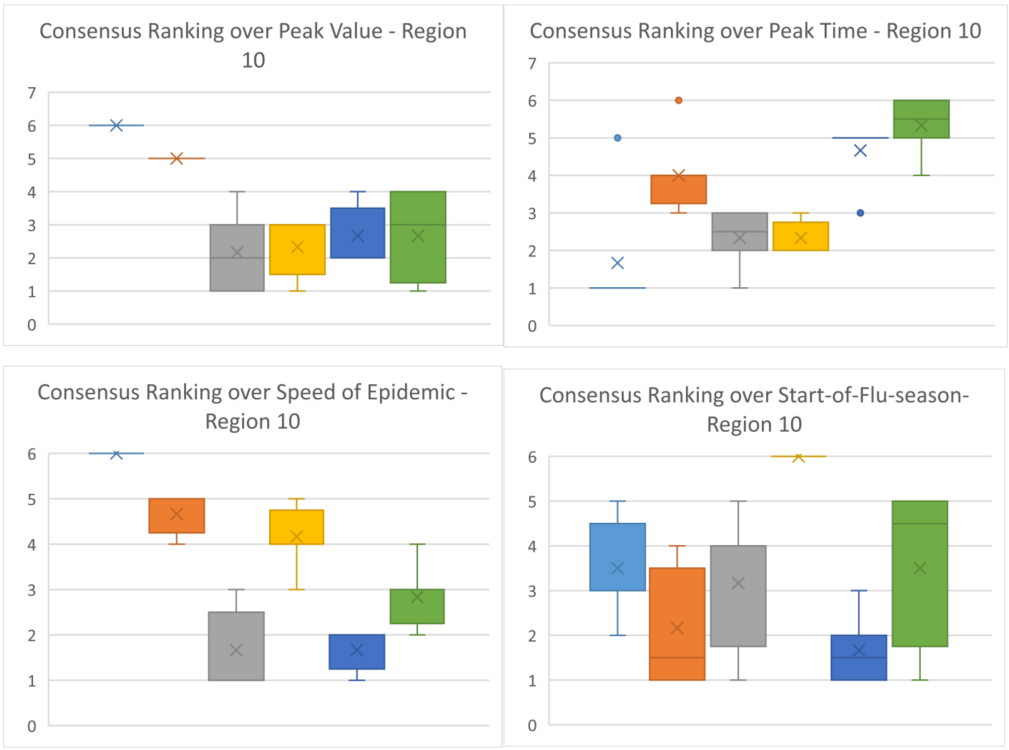


Figure B.10. Consensus Ranking of forecasting methods over all error measuresfor predicting different Epi-features for Region 10

Bibliography

[1] HHS Region Map. http://www.hhs.gov/about/agencies/iea/regional-offices/index.html Accessed 2016-07-07

[2] CDC: Announcement of Requirements and Registration for the Predict the Influenza Season Challenge (2013). http://www.gpo.gov/fdsys/pkg/FR-2013-11-25/pdf/2013-28198.pdf Accessed 2016/07/07

[3] RAPIDD: RAPIDD Ebola challenge: Comparison of disease forecasting models (2015). http://www.ebola-challenge.org/ Accessed 2016/07/07

[4] : Forecasting the Next Dengue Outbreak (2014). https://www.ncdc.noaa.gov/news/forecasting-next-dengue-outbreak Accessed 2016/07/07

[5] : Dengue Forecasting Project (2014). http://dengueforecasting.noaa.gov/docs/project_description.pdf Accessed 2016/07/07

[6] : DARPA CHIKV Challenge to Address Threat of Chikungunya (2014). "http://globalbiodefense.com/2014/08/18/darpa-chikv-challenge-chikungunya/" Accessed 2016/07/07

[7] Makridakis, S., Chatfield, C., Hibon, M., Lawrence, M., Mills, T., Ord, K., Simmons, L.F.: The M2-competition: A real-time judgmentally based forecasting study. International Journal of Forecasting **9**(1), 5–22 (1993). doi:10.1016/0169-2070(93)90044-N

[8] Makridakis, S.: The M3-Competition : results , conclusions and implications. International Journal of Forecasting **16**, 451–476 (2000)

[9] Kuznetsov, Y.A., Piccardi, C.: Bifurcation analysis of periodic SEIR and SIR epidemic models. Journal of Mathematical Biology **32**(2), 109–121 (1994). doi:10.1007/BF00163027

[10] Hethcote, H.W.: The Mathematics of Infectious Diseases. SIAM Review **42**(4), 599–653 (2000). doi:10.1137/S0036144500371907

[11] Bailey, N.T.J.: The Mathematical Theory of Infectious Diseases and Its Applications. Mathematics in Medicine Series, p. 413. Griffin, ??? (1975). https://books.google.com/books?id=A_x-PwAACAAJ

[12] Tabataba, F.S., Lewis, B., Hosseinipour, M., Tabataba, F.S., Venkatramanan, S., Chen, J., Higdon, D., Marathe, M.: Epidemic forecasting by combining agent-based models and smart beam-particle filtering framework. In: Proceedings of the IEEE International Conference on Data Mining (2017)

[13] Tabataba, F.S., Chakraborty, P., Ramakrishnan, N., Venkatramanan, S., Chen, J., Lewis, B., Marathe, M.: A framework for evaluating epidemic forecasts. BMC Infectious Diseases **17**(1), 345 (2017). doi:10.1186/s12879-017-2365-1

[14] Sočan, M., Erčulj, V., Lajovic, J.: Early detection of influenza-like illness through medication sales. Central European Journal of Public Health **20**(2), 156–162 (2012)

[15] Goldstein, E., Cobey, S., Takahashi, S., Miller, J.C., Lipsitch, M.: Predicting the epidemic sizes of influenza A/H1N1, A/H3N2, and B: A statistical method. PLoS Medicine **8**(7), 1–12 (2011). doi:10.1371/journal.pmed.1001051

[16] Dugas, A.F., Jalalpour, M., Gel, Y., Levin, S., Torcaso, F., Igusa, T., Rothman, R.E.: Influenza Forecasting with Google Flu Trends. PLoS ONE **8**, 56176 (2013). doi:10.1371/journal.pone.0056176

[17] Andersson, E., Kühlmann-Berenzon, S., Linde, A., Schiöler, L., Rubinova, S., Frisén, M.: Predictions by early indicators of the time and height of the peaks of yearly influenza outbreaks in Sweden. Scandinavian journal of public health **36**(5), 475–82 (2008). doi:10.1177/1403494808089566

[18] Mooney, J., Holmes, E., Christie, P.: Real-time modelling of influenza outbreaks - a linear regression analysis. Euro Surveill **7**(12) (2002)

[19] Soebiyanto, R.P., Adimi, F., Kiang, R.K.: Modeling and predicting seasonal influenza transmission in warm regions using climatological parameters. PLoS ONE **5**(3), 1–10 (2010). doi:10.1371/journal.pone.0009450

[20] Moriña, D., Puig, P., Ríos, J., Vilella, A., Trilla, A.: A statistical model for hospital admissions caused by seasonal diseases. Statistics in Medicine **30**(26), 3125–3136 (2011). doi:10.1002/sim.4336

[21] Jiang, X., Wallstrom, G., Cooper, G.F., Wagner, M.M.: Bayesian prediction of an epidemic curve. Journal of Biomedical Informatics **42**(1), 90–99 (2009). doi:10.1016/j.jbi.2008.05.013

[22] Sebastiani, P., Mandl, K.D., Szolovits, P., Kohane, I.S., Ramoni, M.F.: A bayesian dynamic model for influenza surveillance. Statistics in medicine **25**(11), 1803–16181725 (2006). doi:10.1002/sim.2566

[23] Hosseini, P., Sokolow, S.H., Vandegrift, K.J., Kilpatrick, A.M., Daszak, P.: Predictive power of air travel and socio-economic data for early pandemic spread. PLOS ONE **5**(9), 1–8 (2010). doi:10.1371/journal.pone.0012763

[24] Polgreen, P.M., Nelson, F.D., Neumann, G.R.: Use of prediction markets to forecast infectious disease activity. Clinical infectious diseases : an official publication of the Infectious Diseases Society of America **44**(2), 272–279 (2007). doi:10.1086/510427

[25] Saltyte Benth, J., Hofoss, D.: Modelling and prediction of weekly incidence of influenza A specimens in England and Wales. Epidemiology and infection **136**(12), 1658–66 (2008). doi:10.1017/S0950268808000307

[26] Held, L., Paul, M.: Modeling seasonality in space-time infectious disease surveillance data. Biometrical Journal **54**(6), 824–843 (2012). doi:10.1002/bimj.201200037

[27] Paul, M., Held, L.: Predictive assessment of a non-linear random effects model for space-time surveillance data. Stat Med **30**(2011), 1118–1136 (2011). doi:10.1002/sim.4177

[28] Viboud, C., Boëlle, P.Y., Carrat, F., Valleron, A.J., Flahault, A.: Prediction of the Spread of Influenza Epidemics by the Method of Analogues. American Journal of Epidemiology **158**(10), 996–1006 (2003). doi:10.1093/aje/kwg239

[29] Bisset, K., Chen, J., Feng, X.: EpiFast: a fast algorithm for large scale realistic epidemic simulations on distributed memory systems. 23rd international conference on Supercomputing, 430–439 (2009). doi:10.1145/1542275.1542336

[30] Meyers, L.A.: Contact network epidemiology: Bond percolation applied to infectious disease prediction and control. Bulletin of the American Mathematical Society **44**(1), 63–86 (2007). doi:10.1090/S0273-0979-06-01148-7

[31] Keeling, M.J., Eames, K.T.D.: Networks and epidemic models. Journal of the Royal Society, Interface / the Royal Society **2**(4), 295–307 (2005). doi:10.1098/rsif.2005.0051

[32] Barrett C, Bisset, K.R., Eubank Stephen G, Feng X, Marathe, M.: EpiSimdemics: An efficient and scalable framework for simulating the spread of infectious disease on large social networks. International Conference for High Performance Computing, Networking, Storage and Analysis (SC08) (November) (2008). doi:10.1145/1413370.1413408

[33] Bisset, K.R., Chen, J., Deodhar, S., Feng, X., Ma, Y., Marathe, M.V.: INDEMICS : An Interactive High-Performance Computing Framework for Data Intensive Epidemic Modeling. ACM Transactions on Modeling and Computer Simulation **24**(1), 4–1432 (2014). doi:10.1145/2501602. NIHMS150003

[34] Tizzoni, M., Bajardi, P., Poletto, C., Ramasco, J.J., Balcan, D., Gonçalves, B., Perra, N., Colizza, V., Vespignani, A.: Real-time numerical forecast of global epidemic spreading: case study of 2009 A/H1N1pdm. BMC medicine **10**, 165 (2012). doi:10.1186/1741-7015-10-165

[35] Nsoesie, E.O., Beckman, R.J., Shashaani, S., Nagaraj, K.S., Marathe, M.V.: A Simulation Optimization Approach to Epidemic Forecasting. PloS one **8**(6), 67164 (2013). doi:10.1371/journal.pone.0067164

[36] Nsoesie, E., Mararthe, M., Brownstein, J.: Forecasting peaks of seasonal influenza epidemics. PLoS currents **5**, 1–14 (2013). doi:10.1371/currents.outbreaks.bb1e879a23137022ea79a8c508b030bc

[37] Nsoesie, E.O., Beckman, R., Lewis, B.: Prediction of an Epidemic Curve: A Supervised Classification Approach. Statistical communications in infectious diseases. **3**(1), 1–27 (2011). doi:10.2202/1948-4690.1038.Prediction

[38] Nsoesie, E.O., Leman, S.C., Marathe, M.V.: A Dirichlet process model for classifying and forecasting epidemic curves. BMC infectious diseases **14**, 12 (2014). doi:10.1186/1471-2334-14-12

[39] Shaman, J., Karspeck, A., Yang, W., Tamerius, J., Lipsitch, M.: Real-time influenza forecasts during the 2012-2013 season. Nature communications **4**, 2837 (2013). doi:10.1038/ncomms3837

[40] Shaman, J., Karspeck, A.: Forecasting seasonal outbreaks of influenza. Proceedings of the National Academy of Sciences of the United States of America **109**(3), 20425–30 (2012). doi:10.1073/pnas.1208772109

[41] Ong, J.B.S., Chen, M.I.-C., Cook, A.R., Lee, H.C., Lee, V.J., Lin, R.T.P., Tambyah, P.A., Goh, L.G.: Real-time epidemic monitoring and forecasting of H1N1-2009 using influenza-like illness from general practice and family doctor clinics in Singapore. PloS one **5**(4), 10036 (2010). doi:10.1371/journal.pone.0010036

[42] Yang, W., Karspeck, A., Shaman, J.: Comparison of filtering methods for the modeling and retrospective forecasting of influenza epidemics. PLoS computational biology **10**(4), 1003583 (2014). doi:10.1371/journal.pcbi.1003583

[43] Skvortsov, A., Ristic, B.: Monitoring and prediction of an epidemic outbreak using syndromic observations. Mathematical Biosciences **240**(1), 12–19 (2012). doi:10.1016/j.mbs.2012.05.010. 1110.4696

[44] Osgood, N., Liu, J.: Towards closed loop modeling: Evaluating the prospects for creating recurrently regrounded aggregate simulation models using particle filtering. In: Proceedings of the Winter Simulation Conference, Savanah, GA, pp. 829–841 (2014). doi:10.1109/WSC.2014.7019944

[45] Sheinson, D.M., Niemi, J., Meiring, W.: Comparison of the performance of particle filter algorithms applied to tracking of a disease epidemic. Mathematical Biosciences **255**(1), 21–32 (2014). doi:10.1016/j.mbs.2014.06.018

[46] Loganathan, P., Ho Chee, S., Hao Ran, L., Lakshminarayanan, S.: Towards forecasting flu dynamics using a regionalized state space model. Advanced Control of Industrial Processes (ADCONIP), 2011 International Symposium on (1), 175–180 (2011)

[47] Shahtori, N.M., Scoglio, C., Pourhabib, A., Sahneh, F.D.: Sequential monte carlo filtering estimation of ebola progression in west africa. In: 2016 American Control Conference (ACC), pp. 1277–1282 (2016). doi:10.1109/ACC.2016.7525093

[48] Koepke, A.A., Longini, I.M., Halloran, M.E., Wakefield, J., Minin, V.N.: Predictive modeling of cholera outbreaks in Bangladesh. Annals of Applied Statistics **10**(2), 575–595 (2016). doi:10.1214/16-AOAS908. 1402.0536

[49] Dukic, V., Lopes, H.F., Polson, N.G.: Tracking Epidemics With Google Flu Trends Data and a State-Space SEIR Model. Journal of the American Statistical Association **107**(January), 1410–1426 (2012). doi:10.1080/01621459.2012.713876

[50] Dureau, J., Kalogeropoulos, K., Baguelin, M.: Capturing the time-varying drivers of an epidemic using stochastic dynamical systems. Biostatistics **14**(3), 541–555 (2013). doi:10.1093/biostatistics/kxs052. 1203.5950

[51] Dawson, P., Gailis, R., Meehan, A.: Detecting disease outbreaks using a combined Bayesian network and particle filter approach. Journal of Theoretical Biology **370**, 171–183 (2015). doi:10.1016/j.jtbi.2015.01.023

[52] Skvortsov, A., Ristic, B., Woodruff, C.: Predicting an epidemic based on syndromic surveillance. 2010 13th International Conference on Information Fusion, 1–8 (2010). doi:10.1109/ICIF.2010.5711847

[53] Jègat, C., Carrat, F., Lajaunie, C., Wackernagel, H.: Early Detection and Assessment of Epidemics by Particle Filtering, pp. 23–35. Springer, Dordrecht (2008). doi:10.1007/978-1-4020-6448-7_2

[54] Armstrong, J.S.: Evaluating Forecasting Methods, pp. 443–472. Springer, Boston, MA (2001). doi:10.1007/978-0-306-47630-3_2. http://dx.doi.org/10.1007/978-0-306-47630-3_20

[55] Nsoesie, E.O., Brownstein, J.S., Ramakrishnan, N., Marathe, M.V.: A systematic review of studies on forecasting the dynamics of influenza outbreaks. Influenza and other respiratory viruses **8**, 309–16 (2014). doi:10.1111/irv.12226

[56] Aguirre, A., Gonzalez, E.: The feasibility of forecasting influenza epidemics in Cuba. Memorias do Instituto Oswaldo Cruz **87**(3), 429–32

[57] Soebiyanto, R.P., Adimi, F., Kiang, R.K.: Modeling and predicting seasonal influenza transmission in warm regions using climatological parameters. PLoS ONE **5**(3), 1–10 (2010). doi:10.1371/journal.pone.0009450

[58] Cha, S.-h.: Comprehensive Survey on Distance / Similarity Measures between Probability Density Functions. International Journal of Mathematical Models and Methods in Applied Sciences **1**(4), 300–307 (2007). doi:10.1007/s00167-009-0884-z

[59] Longini, I.M., Fine, P.E., Thacker, S.B.: Predicting the global spread of new infectious agents. American journal of epidemiology **123**(3), 383–91 (1986)

[60] Chao, D.L., Matrajt, L., Basta, N.E., Sugimoto, J.D., Dean, B., Bagwell, D.A., Oiulfstad, B., Halloran, M.E., Longini, I.M.: Planning for the control of pandemic influenza A (H1N1) in Los Angeles County and the United States. American journal of epidemiology **173**(10), 1121–30 (2011). doi:10.1093/aje/kwq497

[61] Hall, I.M., Gani, R., Hughes, H.E., Leach, S.: Real-time epidemic forecasting for pandemic influenza. Epidemiology and infection **135**, 372–85 (2007). doi:10.1017/S0950268806007084

[62] Ong, J.B.S., Chen, M.I.-C., Cook, A.R., Lee, H.C., Lee, V.J., Lin, R.T.P., Tambyah, P.A., Goh, L.G.: Real-time epidemic monitoring and forecasting of H1N1-2009 using influenza-like illness from general practice and family doctor clinics in Singapore. PloS one **5**(4), 10036 (2010). doi:10.1371/journal.pone.0010036

[63] Towers, S., Feng, Z.: Pandemic H1N1 influenza: predicting the course of a pandemic and assessing the efficacy of the planned vaccination programme in the United States. Euro surveillance : bulletin Europeen sur les maladies transmissibles = European communicable disease bulletin **14**(41), 19358 (2009)

[64] Shaman, J., Karspeck, A.: Forecasting seasonal outbreaks of influenza. Proceedings of the National Academy of Sciences of the United States of America **109**(3), 20425–30 (2012). doi:10.1073/pnas.1208772109

[65] Hyder, A., Buckeridge, D.L., Leung, B.: Predictive Validation of an Influenza Spread Model **8**(6) (2013). doi:10.1371/journal.pone.0065459

[66] Bengtsson, T., Bickel, P., Li, B.: Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. Probability and Statistics **2**, 316–334 (2008). doi:10.1214/193940307000000518. 0805.3034

[67] Eubank, S., Guclu, H., Kumar, V.A., Marathe, M.V., Srinivasan, A., Toroczkai, Z., Wang, N.: Modelling disease outbreaks in realistic urban social networks. Nature **429**(6988), 180–184 (2004)

[68] Marathe, M., Vullikanti, A.K.S.: Computational epidemiology. Communications of the ACM **56**(7), 88–96 (2013)

[69] Barrett, C.L., Beckman, R.J., Khan, M., Kumar, V.A., Marathe, M.V., Stretz, P.E., Dutta, T., Lewis, B.: Generation and analysis of large synthetic social contact networks. In: Simulation Conference (WSC), Proceedings of the 2009 Winter, pp. 1003–1014 (2009). IEEE

[70] RAPIDD Ebola Challenge. `http://ebola-challenge.org` Accessed 2016-08-05

[71] WHO, CDC: Implementation and management of contact tracing for Ebola virus disease (2015). `http://apps.who.int/iris/bitstream/10665/185258/1/WHO_EVD_Guidance_Contact_15.1_eng.pdf?ua=1`

[72] Contact Tracing. `https://www.cdc.gov/vhf/ebola/outbreaks/what-is-contact-tracing.html` Accessed 2016-08-05

[73] What is contact tracing? `https://www.cdc.gov/vhf/ebola/pdf/contact-tracing.pdf` Accessed 2016-08-05

[74] Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME–Journal of Basic Engineering **82**, 35–45 (1960)

[75] Welch, G., Bishop, G.: An Introduction to the Kalman Filter. Technical report (1995)

[76] Alspach, D., Sorenson, H.: Nonlinear Bayesian estimation using Gaussian sum approximations. IEEE Transactions on Automatic Control **17**(4), 439–448 (1972). doi:10.1109/TAC.1972.1100034

[77] Gordon, N.J., Salmond, D.J., a.F.M. Smith: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F Radar and Signal Processing **140**(2), 107 (1993). doi:10.1049/ip-f-2.1993.0015. 9241F (E5)

[78] Isard, M., Blake, A.: Contour Tracking by Stochastic Propagation of Conditional Density. In: Proceedings of the 4th European Conference on Computer Vision, pp. 343–356. Springer, ??? (1996). http://dl.acm.org/citation.cfm?id=645309.648900

[79] Mousavi, S.R., Bahri, F., Tabataba, F.S.: An enhanced beam search algorithm for the shortest common supersequence problem. Eng. Appl. Artif. Intell. **25**(3), 457–467 (2012). doi:10.1016/j.engappai.2011.08.006

[80] Merler, S., Ajelli, M., Fumanelli, L., Gomes, M.F.C., Piontti, A.P.y., Rossi, L., Chao, D.L., Longini, I.M., Halloran, M.E., Vespignani, A.: Spatiotemporal spread of the 2014 outbreak of Ebola virus disease in Liberia and the effectiveness of non-pharmaceutical interventions: A computational modelling analysis. The Lancet Infectious Diseases **15**(2), 204–211 (2015). doi:10.1016/S1473-3099(14)71074-6. 15334406

[81] Pell, B., Kuang, Y., Viboud, C., Chowell, G.: Using phenomenological models for forecasting the 2015 Ebola challenge. Epidemics (2016). doi:10.1016/j.epidem.2016.11.002

[82] Tuite, A.R., Fisman, D.N.: The IDEA model: A single equation approach to the Ebola forecasting challenge. Epidemics, 1–7 (2016). doi:10.1016/j.epidem.2016.09.001

[83] Funk, S., Camacho, A., Kucharski, A.J., Eggo, R.M., Edmunds, W.J.: Real-time forecasting of infectious disease dynamics with a stochastic semi-mechanistic model. Epidemics (2016). doi:10.1016/j.epidem.2016.11.003

[84] Gaffey, R.H., Viboud, C.: Application of the CDC EbolaResponse Modeling tool to disease predictions. Epidemics **2015**(January 2015), 1–7 (2016). doi:10.1016/j.epidem.2017.03.001

[85] Asher, J.: Forecasting ebola with a regression transmission model. Epidemics (2017). doi:10.1016/j.epidem.2017.02.009

[86] Champredon, D., Li, M., Bolker, B.M., Dushoff, J.: Two approaches to forecast ebola synthetic epidemics. Epidemics (2017). doi:10.1016/j.epidem.2017.02.011

[87] Nouvellet, P., Cori, A., Garske, T., Blake, I.M., Dorigatti, I., Hinsley, W., Jombart, T., Mills, H.L., Nedjati-Gilani, G., Kerkhove, M.D.V., Fraser, C., Donnelly, C.A., Ferguson, N.M., Riley, S.: A simple approach to measure transmissibility and forecast incidence. Epidemics (2017). doi:10.1016/j.epidem.2017.02.012

[88] Venkatramanan, S., Lewis, B., Chen, J., Higdon, D., Vullikanti, A., Marathe, M.: Using data-driven agent-based models for forecasting emerging infectious diseases. Epidemics, (2017). doi:10.1016/j.epidem.2017.02.010

[89] Principles of Epidemiology in Public Health Practice, Third Edition An Introduction to Applied Epidemiology and Biostatistics. `http://www.cdc.gov/ophss/csels/dsepd/SS1978/Lesson3/Section2.html` Accessed 2016-07-07

[90] 2009 H1N1 Early Outbreak and Disease Characteristics. `http://www.cdc.gov/h1n1flu/surveillanceqa.htm\#4` Accessed 2016-07-07

[91] Overview of Influenza Surveillance in the United States. `http://www.cdc.gov/flu/weekly/overview.htm` Accessed 2016-07-07

[92] Shcherbakov, M.V., Brebels, A., Shcherbakova, N.L., Tyukov, A.P., Janovsky, T.A., evich Kamaev, V.A.: A survey of forecast error measures. World Applied Sciences Journal **24**(24), 171–176 (2013). doi:10.5829/idosi.wasj.2013.24.itmies.80032

[93] Syntetos, A.A., Boylan, J.E.: On the variance of intermittent demand estimates. International Journal of Production Economics **128**(2), 546–555 (2010). doi:10.1016/j.ijpe.2010.07.005

[94] U.S. Outpatient Influenza-like Illness Surveillance Network (ILINet). https://public.health.oregon.gov/DiseasesConditions/CommunicableDisease/DiseaseSurveillanceData/Influenza/Documents/recruitment_cdc_system.pdf Accessed 2016-07-07

[95] Overview of Influenza Surveillance in the United States. http://www.cdc.gov/flu/pdf/weekly/overview.pdf Accessed 2016-07-07

[96] Armstrong, B.J.S., Collopy, F.: Error Measures For Generalizing About Forecasting Methods: Empirical Comparisons By J. Scott Armstrong and Fred Collopy Reprinted with permission form. International Journal of Forecasting **8**(1), 69–80 (1992). doi:10.1016/0169-2070(92)90008-W

[97] Deza, M.M., Deza, E.: Encyclopedia of Distances, pp. 1–590 (2009). doi:10.1007/978-3-642-00234-2. 0505065

[98] Abou-Moustafa, K.T., Ferrie, F.P.: A note on metric properties for some divergence measures: The Gaussian case. Journal of Machine Learning Research **25**, 1–15 (2012)

[99] Pardo, L.: Statistical Inference Based on Divergence Measures vol. 170, p. 497 (2006). doi:10.1111/j.1467-985X.2007.00485-7.x

[100] Unified Modeling Language. http://www.uml.org/ Accessed 2017-09-11

[101] Kuznetsov, Y.A., Piccardi, C.: Bifurcation analysis of periodic SEIR and SIR epidemic models. Journal of Mathematical Biology **32**(2), 109–121 (1994). doi:10.1007/BF00163027

[102] Lekone, P.E., Finkenstädt, B.F.: Statistical inference in a stochastic epidemic SEIR model with control intervention: Ebola as a case study. Biometrics **62**(December), 1170–1177 (2006). doi:10.1111/j.1541-0420.2006.00609.x

[103] Roche, B., Drake, J.M., Rohani, P.: An agent-based model to study the epidemiological and evolutionary dynamics of Influenza viruses. BMC bioinformatics **12**(1), 87 (2011). doi:10.1186/1471-2105-12-87