DEEP DATA DRIVEN NEURAL NETWORKS FOR LEARNING DYNAMICS OF COVID-19 EPIDEMIC MODELS

By

Thomas Kofi Torku

Dissertation Defense

Submitted in Partial Fulfillment

of the Requirement for the Degree of

Doctor of Computational and Data Science

Middle Tennessee State University

May, 2023

Dissertation Committee:

Prof. Abdul Khaliq (Mathematical Sciences), Chair.

Prof. Wandi Ding (Mathematical Sciences).

Prof. William Robertson (Physics).

ABSTRACT

We present three deep-learning methods to analyze different COVID-19 epidemic models. The first method, an epidemiology-informed neural network, is developed to learn the model parameters and dynamics of a deterministic vaccine efficacy model. Data-driven simulations and error metrics confirm that vaccinating more people curtails the spread of the disease. Data-driven simulations of a hybrid approach comprising residual and recurrent neural networks show that the ResNet-GRU hybrid is superior. In the second method, we learn the dynamics of the stochastic vaccine efficacy model by developing a stochastic epidemiology-informed neural network (SEINN). This method involves discretizing the system of stochastic differential equations using Euler-Murayama and encoding it as a loss function. The SEINN integrates a regularization parameter to enhance training accuracy. Subsequently, this second method aims to investigate the superiority of stochastic models over deterministic ones. Our findings show that stochastic models outperform their deterministic counterparts based on their error metrics. Finally, we examine the impact of stochasticity and vaccination on nonlinear incidence rates. The behavior-epidemiology-informed neural network (BEINN) is the third method. It learns an explicit compliance rate that changes over time and identifies the critical epidemiological parameters of a behavioral epidemic model. This third algorithm incorporates behavioral constraints, attention mechanism, and the regularization parameter to improve training precision. This method aims to analyze how social and behavioral factors impact disease dissemination. The computational analysis of BEINN regarding sensitivity, overfitting, and computational time shows the effectiveness and robustness of the third method. The epidemiological importance of this work includes the following: the outcome of the three methods can help public health officials develop effective vaccination and mitigation strategies by considering randomness in the model and the variability of human behavior in transmitting contagious diseases. We apply these three proposed methods to the spread of COVID-19 in Tennessee. The third methodology is employed to analyze the transmission of COVID-19 in the states of New York and Michigan.

This work is dedicated to:
my wife, Purity Torku, and my children, Praise Torku and Promise Torku
my dad, Mr. Olympio Torku, and my mom, Mrs. Elizabeth Torku
my beloved pastor, Mr. Saheed Hassan.

ACKNOWLEDGMENTS

First and foremost, I am grateful to God for His wisdom and guidance. Next, I am indebted to many people for my Ph.D. journey. They have been my support system since I began my graduate studies. I want to thank my advisor, Professor Abdul Khaliq, for his relentless care, guidance, and mentorship that have helped me this far. The encouragement and support of Professor John Wallin, the program director, are worth mentioning. I thank Professor Wandi Ding and Professor William Robertson, who agreed to be on my committee. I appreciate the advice and professional mentorship from Dr. Marva Lucas, my former department chair. I am grateful to be part of the CRUNCH group at Brown University, under the direction of Professor George Em Karniadakis, which holds weekly seminars to discuss current cutting-edge computational scientific research. Finally, I applaud this group's novel paper on physics-informed neural networks.

I also want to thank the University Studies and Computational and Data Science Program faculty and colleagues for their encouragement: Dr. Vivian Alley, Dr. Matthew Duncan, and Mr. Ebenezer Oluwasakin. Finally, I thank Dr. Khalid Furati of KFUPM for his insights in shaping my research direction.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

1

INTRODUCTION

The SARS-Cov-2 virus spread worldwide, prompting the World Health Organization to classify it as a pandemic [1, 2]. The onset of the disease caused by the virus first emerged in Wuhan, China, in December 2019, as reported by Ghamizi et al. (2020). To mitigate the transmission of the virus, governmental authorities have implemented non-pharmaceutical interventions, including but not limited to social distancing, quarantine measures, and the temporary closure of public institutions. To mitigate the significant consequences of the spread, a pharmaceutical intervention in the form of vaccination was implemented in the latter portion of 2020. Numerous investigations have been conducted regarding efficacious vaccination tactics, vaccine confidence [3], and vaccine effectiveness [2].

During its initial stages, the COVID-19 outbreak presented unparalleled obstacles to worldwide public health infrastructures. Policymakers employ mitigation strategies to restrict the transmission of the virus. However, the fluctuation has hindered the efficacy of these measures of human compliance behavior. Therefore, comprehending the impact of human behavior on the information of COVID-19 is imperative in formulating efficacious standards and regulations for its containment [4].

Differential equations-based mathematical models are utilized across diverse fields to address various problems. For example, ordinary differential equations (ODE), partial differential equations (PDE), and stochastic differential equations (SDE) are widely recognized types of differential equations. The equations in question have been commonly addressed by applying traditional numerical techniques, including finite difference, finite element, and spectral methods. Recent advancements in technology and computational infrastructure have facilitated the development of deep learning models that can effectively learn the dynamics of differential equations, as evidenced by Raissi et al.'s work [5].

An examination of the pertinent literature indicates that a significant amount of research has been conducted on the transmission of COVID-19. Scholars have investigated diverse facets of viral transmission, ranging from simple to complex Susceptible-Infected-Recovered (SIR) models [6]. Using a deterministic model, Biala et al. [7] analyzed the effectiveness of contact tracing in reducing viral transmission. Furati and colleagues (2021) formulated a fractional model contingent upon government intervention. Several scholars have conducted studies on models that incorporate time-varying parameters, as evidenced by the works of Long et al. [8], Ning et al. [9], and Ning et al. [10]. The impact of vaccination programs on virus transmission in Tennessee, USA, was investigated by Torku et al. [2]. The researchers have validated the overarching hypothesis that the transmissibility of infectious diseases diminishes with an increase in the daily vaccination rate. The study conducted by Olumoyin et al. [11] employed a deterministic Asymptomatic Susceptible-Infected-Recovered (A-SIR) model to estimate constant and time-varying transmission rates for selected countries such as the United States, South Korea, and Italy.

Numerous investigations have been conducted on the impact of human behavior on the spread of infectious diseases. Nevertheless, the implications of these outcomes are occasionally considered models. The studies conducted by Kharazmi et al. [12], Long et al. [8], and Olumoyin et al. [11] did not incorporate a specific consideration for human behavior. Verelst and colleagues [13] conducted a comprehensive analysis of multiple behavioral change models aimed at mitigating the transmission of infectious diseases. The researchers thoroughly searched various databases to identify relevant studies published from 2010 to 2015 about behavioral change models in the context of infectious disease transmission. Fifty-four studies were subjected to analysis and evaluation, with particular attention paid to each study's characteristics, methods, and outcomes. The authors Sebastian et al. [14] underscored the importance of understanding the impact of human behavior on the transmission of infectious diseases. According to the assertion made by the author, mathematical models serve as valuable tools in quantifying the effects of behavioral modifications on

disease dynamics, given that direct experimental studies are occasionally unfeasible.

Data-driven modeling approaches are frequently employed to investigate the transmission dynamics of infectious diseases, such as COVID-19. Using real-time data, machine learning algorithms, including neural networks and support vector machines, have been employed to predict the number of COVID-19 cases and fatalities. Nurul and colleagues [15] have devised a deep learning model that utilizes various factors, such as human mobility, meteorological data, and public health interventions, to forecast the daily confirmed cases of COVID-19 in China. The model's efficacy was evaluated in their study. The authors of another study [16] employed machine learning methodologies to forecast the incidence of COVID-19 cases and fatalities in the United States, utilizing social and economic indicators as predictors.

The extensive scientific applications of deep learning, including but not limited to autonomous vehicles, object detection, and image recognition, have contributed to its widespread popularity [17]. Deep neural networks can effectively address partial and ordinary differential equations, a positive development. The Physics-Informed Neural Network (PINN) was proposed by Raissi et al. [5]. This approach incorporates the underlying physics of a dynamical system into the loss function of a neural network, enabling the network to learn the system's parameters and dynamics. As per the Universal Theory of Approximation, it is theoretically possible for a deep neural network to acquire knowledge of any arbitrary function [18, 19].

Chapter 1 provides a comprehensive review of the relevant literature about the background of the study. The notion of deep learning is explained in Section 1.1. Section 1.2 provides an exposition on various types of neural networks, including feedforward neural network (FNN), residual neural network (ResNet), and recurrent neural network (RNN). The study of differential equations dynamics involves the examination of both deterministic models (as discussed in Section 1.3) and stochastic models (as mentioned in Section 1.4).

Chapter 2 exposes the significance and implications of data-driven simulations. The objective of employing data-driven modeling is articulated in Section 2.1. Section 2.2 explains various data processing techniques, including left censoring, right censoring, and data standardization. Section 2.3 delves into the error metrics that function as evaluation metrics. The error metrics can be categorized into four main types: Squared Errors, Absolute Errors, Correlation Errors, and Log Errors. The introduction of cross-validation in Section 2.4 is presented as a supplementary performance metric.

Chapter 3 presents a deterministic model for the efficacy of a COVID-19 vaccine. The present chapter employs a modified version of the feedforward neural network, namely the Epidemiology Informed Neural Network (EINN) (as detailed in Section 3.1.2), to acquire the model's parameters. Sections 3.2 and 3.3 provide the data-driven simulations for the vaccine model and the error metrics for the previous simulations, respectively. Ultimately, the concluding remarks of this particular chapter are presented in Section 3.3.

In Chapter 4, the stochastic model for COVID-19 vaccination is analyzed to estimate the transmission rate that varies with time. In Section 4.1, the mathematical properties of the stochastic model are examined. Section 4.2 presents a deep learning algorithm for the stochastic model, called Stochastic Epidemiology Informed Neural Network (SEINN). Additionally, outcomes derived from data-based simulations about the model are presented in Section 4.3. The conclusive remarks are expounded in Section 4.7.

The work on the behavioral epidemic model is presented in Chapter 5. Section 5.1 accounts for the behavioral susceptible-infected-recovered-death (SIRD) model. Section 5.2 explains the implementation of the behavioral epidemiology-informed neural network (BEINN) algorithm. Section 5.3 presents simulations based on data for the method. The computational analysis of the proposed method is carried out in Section 5.4. Section 5.5 provides an extensive analysis of the outcomes. The final Section of the Chapter is presented in Section 5.6.

The conclusive remarks of the entire study are presented in Chapter 6. This Chapter

outlines the potential research areas and future work.

1.1 Deep Learning

The processing of data representations through multi-layer perceptrons is commonly referred to as deep learning [20]. The method has garnered attention due to its ability to effectively address complex and multi-dimensional issues. Deep learning has been implemented in various fields of study, including but not limited to science, engineering, geography, chemistry, physics, finance, and business.

1.2 Deep Neural Networks

According to Goodfellow et al., [21], the framework of a deep neural network mimics how the brain functions. The neural network architecture comprises three fundamental layers, namely the input layer, hidden layer(s), and output layer. The input layer is responsible for receiving and processing the input data. Subsequently, an activation function is utilized to convert the data within the hidden layer. The output layer receives the net output that is transmitted from the hidden layer. The process of backpropagation guarantees the optimization of the network's parameters through the minimization of the loss function.

1.2.1 Feedforward Neural Network (FNN)

The feedforward neural network (FNN) is a type of deep neural network architecture whose node connections are simple and straightforward but not cyclic [2]. Figure 1.1 explains the basic architecture of a simple feedforward neural network.

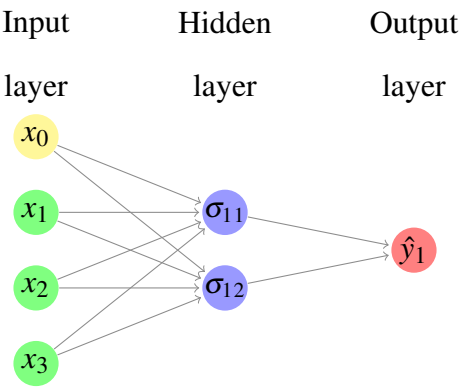

Figure 1.1: A Feedforward Neural Network (FNN)

The formula used to transform the data from one layer to the other is defined as [2]:

$$z_j^{l+1} = \sum_i^{n_l} w_{ij}^l f_{l-1}(z_j^l) + b^l, \tag{1.1}$$

where w_{ij}^l is the weights between the previous node i^{th} and current node j^{th}, b^l is the bias for the l^{th} layer, f represents the activation function, n_l is the number of neurons, $z_i^l, i = 1, \ldots, n_l$, shows the output of the i^{th} node in $(l-1)^{th}$ layer [2]. Examples of popular activation functions used in deep neural networks are:

1. **The Sigmoid Activation Function**. The formula for sigmoid is:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1.2}$$

Equation 1.2 takes in positive and negative values as arguments and returns 1 and 0, respectively.

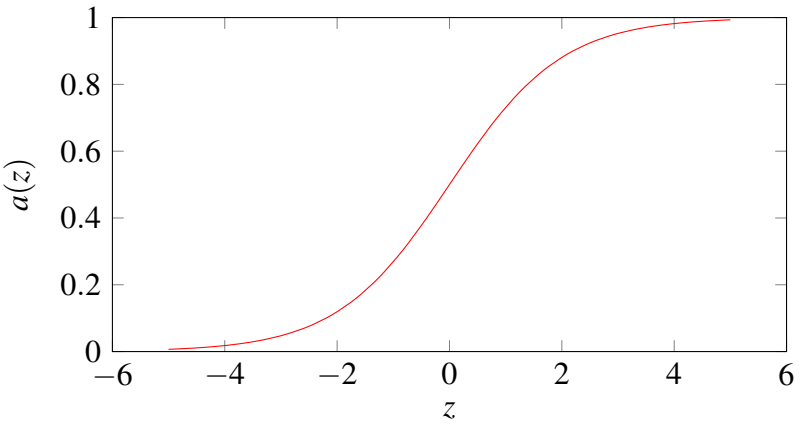

Figure 1.2: Sigmoid

2. **The Tangent Hyperbolic Function**. The mathematical definition is:

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{1.3}$$

This nonlinear activation function in 1.3 squashes positive values to 1 and negative values to -1.

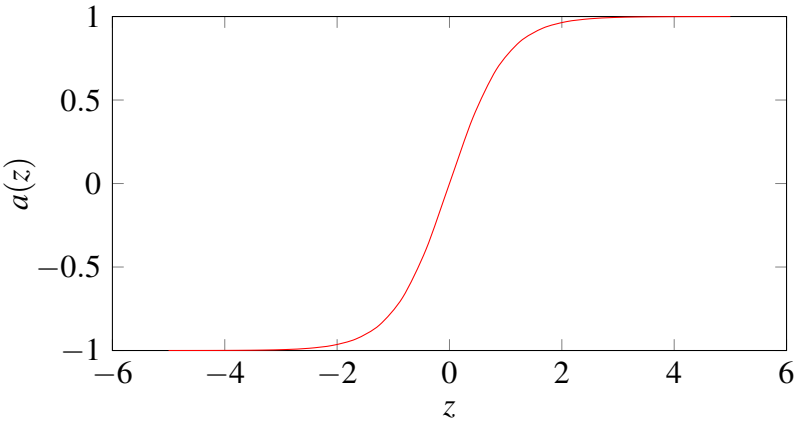

Figure 1.3: Hyperbolic Tangent

1.2.2 Physics Informed Neural Network (PINN)

Physics Informed Neural Networks (PINNs) [5] are a class of neural networks that have been specifically developed to address challenges within the field of physics and beyond.

The fundamental idea pertains to incorporating established principles of physics into a neural network framework. The terminology "Physics Informed Neural Networks" is attributed to the fact that the model is imbued with knowledge from the field of physics. In the feedforward neural network (FNN) context, the predominant objective is to minimize a loss function that quantifies the dissimilarity between the network's predictions and the actual data. In contrast, Physics-Informed Neural Networks (PINNs) distinguish themselves by incorporating a supplementary loss component, referred to as the residual loss, derived from the fundamental principles of the physical phenomena under investigation. The residual loss function renders Physics-Informed Neural Networks (PINNs) highly advantageous in diverse scientific and engineering domains. For example, they can be applied in fluid dynamics, heat transfer, and other problem domains that require an understanding of physics. In addition, traditional neural networks may need help in these areas due to the complex nature of the problem and the significance of adhering to established physical principles.

1.2.3 Regularized Physics Informed Neural Network (RPINN)

Physics Informed Neural Network (PINN) is a framework that allows the underlying physics of the dynamical system into the loss function. It can solve both forward and inverse problems [22, 5]. Using neural networks also introduces challenges such as overfitting, numerical stability, and generalization to unseen data. Regularization techniques [23] can mitigate these challenges and improve the performance and robustness of PINNs. Some standard regularization methods used in PINNs include weight decay, early stopping, dropout, and spectral normalization. These techniques encourage the model to learn simple and smooth solutions that generalize well to unseen data while preventing overfitting and numerical instability. For a vanilla PINN, the loss function

$$Loss = L_{data} + L_{res}, \tag{1.4}$$

L_{data} represents the contribution loss value from the predicted and actual data. L_{res} gives the loss from the underlying physics of the model. We introduce the regularization parameter ε into the loss function in equation (1.4) to improve accuracy and smooth training. Thus, the loss function for the regularized PINN [24] becomes

$$Loss = (1-\varepsilon)L_{data} + \varepsilon L_{res}. \tag{1.5}$$

1.2.4 Recurrent Neural Network (RNN)

The recurrent neural network (RNN) solves problems involving sequential data. The variants of RNN include the following:

1. **Long Short-Term Memory (LSTM)** A variant of recurrent neural network (RNN) was developed to solve the vanishing gradient problem. The input, forget, and output gates are represented by $I_t, F_t, and O_t$ respectively. The cell state is represented by C_t [2].

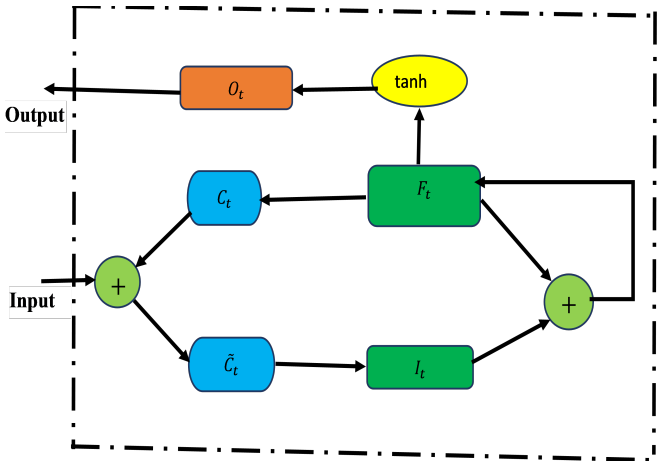

Figure 1.4: Long Short-Term Memory (LSTM)

2. **Bidirectional LSTM (BiLSTM)** as a variant of recurrent neural network (RNN) is an improved version of the LSTM that allows for both backward and forward propagation through the sequential layers. The forward and backward layers are indicated by yellow and green colors, respectively [2].

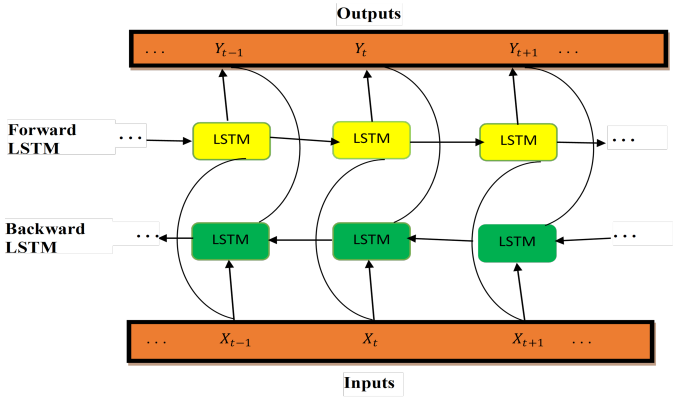

Figure 1.5: Bidirectional LSTM

3. **Gated Recurrent Unit (GRU)** as a variant RNN improves the performance of LSTM and reduces the number of its parameters. The input and forget gates from the LSTM model are merged into one gate called the update gate. It comprises only two updates and reset gates instead of three in LSTM. reset and update gates are represented by R_t and Z_t, respectively.

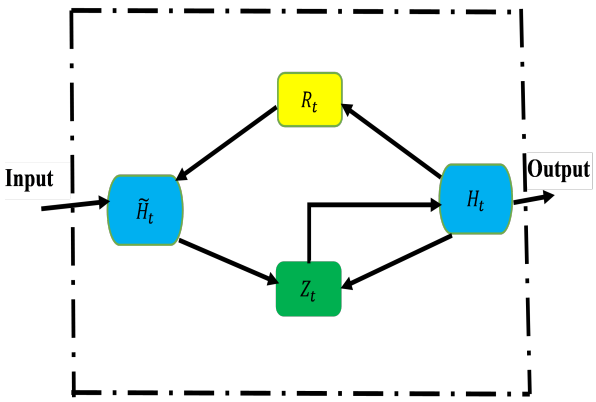

Figure 1.6: Gated Recurrent Unit

1.2.5 Residual Neural Network (ResNet)

The Residual Neural Network is like the Euler method [25].

$$y^{out} = y^{in} + \mathbf{N}(y^{in}, \theta). \tag{1.6}$$

y^{out} is the output, y^{in} is the input (data from solver or fabricated or real measurement), $\mathbf{N}$ is the neural network, and θ is the parameter of the neural network [2].

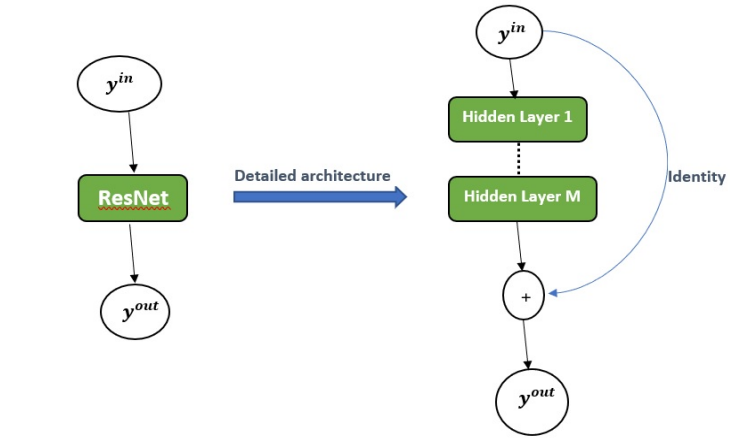

Figure 1.7: Residual Neural Network (ResNet)

1.2.6 Attention Mechanism

The concept of attention mechanism was initially introduced by Vaswani et al. in their 2017 publication [26]. The process of calculating attention weights entails the summation of the absolute values of each channel present in the input tensor. As a result, the attention weights will be maximized for the channels that demonstrate the greatest absolute values. Following this, the input tensor undergoes element-wise multiplication with the attention weights. The resulting tensor is obtained by multiplying the input tensor with the attention weights, resulting in a weighted variant of the input tensor. Utilizing the attention mechanism is a potent approach that can augment the effectiveness of deep learning models. This methodology can guide the model's focus toward the significant characteristics of the input data. This can confer benefits in diverse domains, including machine translation, nat-

ural language processing, and image recognition. The attention weights are computed by summing the absolute values of each channel across the input tensor and subsequently applied to the input tensor element-wise through element-wise multiplication. The attention function can be represented mathematically as follows:

$$\text{Attention}(X) = X \odot \left(\sum_{d=1}^{D} \left| X_{:,:,d} \right| \right) \tag{1.7}$$

where X is the input tensor, a 3D tensor with shape $[batch\_size, height, width]$, D is the number of channels in the input tensor, Attention(X) is the output tensor, a 3D tensor with the same shape as the input tensor. $\odot$ is the element-wise multiplication operator,$\sum$ is the summation operator, $|$ is the absolute value operator, d channel of the input tensor.

1.3 Deterministic Models

A deterministic model is described by an ordinary differential equation (ODE). We have scalar systems of ODEs that describe various physical phenomena. These equations do not have inherent white noise. Consider a deterministic model described by a simple ODE as follows

$$\frac{dx}{dt} = f(x(t)), \tag{1.8}$$

where x is the state, t is the time and f describes the dynamics of the model.

1.4 Stochastic Models

Stochastic models are generally described by stochastic differential equations (SDE). The stochasticity expressed by these equations reveals some hidden functional dynamics of the model. The Gaussian white noise [27] inherent in these models causes different realizations of the dynamics. The realizations usually follow a Brownian Motion. The general form of

SDEs is given by the equation below.

$$dx = f_1(x, h(x))dt + f_2(x, h(x))dw, \tag{1.9}$$

where f_1 is the deterministic part, f_2 is the stochastic part, t is the time, x is the state space, h is the "unknown hidden physics" and w is some Gaussain white noise.

2

DATA DRIVEN SIMULATIONS

The tern simulation in computational science denotes modeling a prototype of a physical process or system. In numerical methods, simulation involves using numerical schemes to find approximate solutions to differential equations. On the other hand, data-driven simulation is where the discovery of parameters and explanations of a defined model entirely relies on data. Torku et al. [2] and Kharami et al. [12] used data-driven simulations to model the parameters and dynamics of COVID-19 model entirely relying on available data.

2.1 Data-driven Modeling

Data-driven modeling [28] is a form of modeling that uses data to construct and validate models. It involves applying statistical and mathematical techniques to identify relationships and patterns in extensive data sets. These patterns, and relationships aid in developing predictive and decision-making models. In healthcare and engineering, for example, data-driven modeling is used to create models that identify optimal solutions to complex problems [29]. Deep neural networks such as Regularized Physics Informed Neural Networks (RPINN) [22, 5] and variants of recurrent neural networks: In this section, long-short-term memory (LSTM) and attention-based LSTM (A-LSTM) are utilized. RPINN is an efficient instrument for discovering the compliance rate and identifying crucial epidemiological parameters. LSTM and A-LSTM are also used to predict the future dynamics of the behavior and epidemiology parameters and the dynamics of the epidemic model.

2.2 Data Preprocessing

Raw data is often noisy and incomplete [30]. The process of transforming raw data into an efficient format for practical data analysis and inference is called data preprocessing mention feng2021seir. Effective data preprocessing improves results from data-driven simula-

tions. Both structured and unstructured data types require some data preprocessing. There are several techniques for accomplishing effective data preprocessing. Scaling, standardization, left and right censoring, one-hot encoding (OHE), data augmentation, and normalization are examples of methods. While left and right censoring and standardization are good data preprocessing techniques for structured data like time series data, OHE and data augmentation are suitable for unstructured data like text data. The choice of data preprocessing technique is data-dependent.

Four main steps are involved in data preprocessing in data science [31]. They include:

1. **Data Cleaning**: This is where missing values, outliers, and noise in the data are handled. Some common ways to take missing values are mean/median/mode substitution and ignoring instances of unknown feature [30].

2. **Data Integration**: This involves storing data from multiple sources in a consistent data warehouse.

3. **Data Specification**: In this step, data is reduced to better represent the data. Dimension Reduction techniques such as Principal Component Analysis (PCA) extract principal components of high dimensional data that apply to the model development.

4. **Data Transformation**: This is the last step that uses techniques like standardization to transform the data into a helpful format for insightful data analysis.

2.2.1 Left Censoring

Data censorship occurs when there is a failure to collect data at a particular time. Left censorship involves the loss of the monitoring and data-reporting system to gather complete information at the beginning of a phenomenon. For example, At the beginning of the COVID-19 pandemic, there was incomplete data on reported cases. However, by deleting insufficient information at the beginning and resetting the start time, left censorship can be reduced [32].

2.2.2 Right Censoring

In the evolution of a phenomenon, some changing factors are introduced. This affects data reporting and monitoring. While mitigation measures were already in place to combat the spread of the COVID-19 virus, vaccines were rolled out to curb the surge. The introduction of vaccines is seen as changing factor [32] which impacts the data reporting. This leads to the correct censorship. Abandoning data after the vaccine was rolled out will help minimize the effect of correct censorship.

2.2.3 Data Standardization

The impact of noise and outliers can be reduced through data standardization. This process involves centering the data with a certain range for smooth training. Popular data standardization techniques are:

1. **The Z-score**: The mean and standard deviation are the two parameters used to standardize data. The values of the parameters are 0 and 1 for mean and standard deviation, respectively. The formula for the z-score is [32]

$$x' = \frac{x - \bar{x}}{\sigma_x}, \tag{2.1}$$

   where the $\bar{x}$ is the mean, σ_x is the standard deviation and x' is the processed data.

2. **The Min-Max Scaling**: This is the scaling method where the minimum and maximum of the raw data are used to transform all the data values into a range from 0 to 1. This serves tends to smooth out noisy data and enhances training. The min-max scaler is defined as follows:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{2.2}$$

   Where the x_{min} and x_{max} represent the minimum and maximum values of the raw data, respectively, x is the raw data, and x' is processed data.

2.3 Error Metrics

Error metrics are evaluation measures used to assess models' performance. In machine learning, error metrics are suitable for regression-based models, while accuracy metrics are suitable for classification-based models. Regression-based models can either be linear or nonlinear. According to Rob Hyndman [33], error metrics can be scale-dependent or scale-free. Scale-dependent means that the metric is affected by the unit of measurement of the data. There are several error metrics. For example, figure 2.1 shows a survey of error metrics for data-driven simulation.

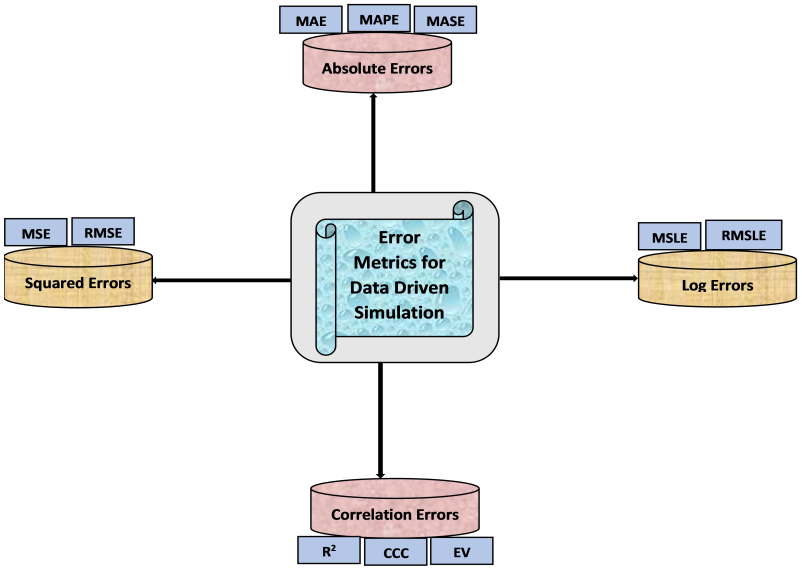

Figure 2.1: A Survey of Error Metrics

Let y be the real data, $\hat{y}$ be the predicted data from the model, and N be the length of the data [2]. Then

2.3.1 Squared Errors

1. **Mean Squared Error (MSE)** is the squared difference between the real and predicted data.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.3}$$

The MSE is commonly used as the loss function for most regression-based deep learning models. Datasets to which MSE can be applied to include: Stock Data, Petroleum Data, and COVID-19 Data

2. **Root Mean Squared Error (RMSE)** is the square root of the MSE. The values are scaled by taking the square root. It can be considered as a normalizer of large error values. This produces better error values and is good for model comparison.

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2.4}$$

RMSE can compare ensemble regression-based models, different regression-based probabilistic models, and hybrid regression-based models [34, 2].

2.3.2 Absolute Errors

1. **Mean Absolute Error (MAE)** refers to the average absolute difference between the real and predicted data values.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.5}$$

The MAE is measured in the same unit as the data, making it scale-dependent. This means that any inaccuracies in the scale can affect model performance. Thus, the MAE is not good for model comparison [35].

2. **Mean Absolute Percentage Error (MAPE)** is the relative error in the mean abso-

lute error. This metric is good for quantile regression since it is a weighted error metric[36].

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}|\frac{y_i - \hat{y}_i}{y_i}|\%. \tag{2.6}$$

One major drawback of this error metric is that it cannot have small values close to zero or zero in the denominator. Since any of such values will make the error metric undefined.

3. **Mean Absolute Scaled Error (MASE)** is a scale-free error metric that can compare forecast methods on single series. Compared to MAPE, it can never give infinite values or undefined values [33].

$$\begin{aligned} q_t &= \frac{e_t}{\frac{1}{N-1}\sum_{i=2}^{N}|y_i - \hat{y}_i|} \\ MASE &= mean(|q_t|) \end{aligned} \tag{2.7}$$

where q_t is the scaled error and e_t is the forecast error from *in-sample* MAE. If q_t is less than one, then we have a better forecast, whiles if q_t is greater than one, then the forecast is worse. MASE can be applied to intermittent datasets like gas data and stock data [33].

2.3.3 Correlation Errors

1. **Coefficient of Determination** (R^2) measures the proportion of explained variance in the model by response variable. It is primarily an error metric for linear models [37].

$$\begin{aligned} SS_{res} &= \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N} e_i^2 \\ SS_{tot} &= \sum_{i=1}^{N}(y_i - \bar{y})^2 \\ R^2 &= 1 - \frac{SS_{res}}{SS_{tot}} \end{aligned} \tag{2.8}$$

where $\bar{y}$ is the mean of the real data, SS_{res} is the sum of square residuals and SS_{tot} is the total sum of squares.

2. **Concordance Correlation Coefficient (CCC)** measures the consistency in evaluating a new method on repeated measurements against a gold standard if it exists. The CCC ensures reliability and consistency in continuous measurement if the reproducibility of outcome is concerned. CCC is applied to health-related data [38, 39].

$$CCC = \frac{2\rho\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2} \tag{2.9}$$

where ρ is the correlation coefficient between two variables x and y, μ_x and μ_y are the means, and σ_x and σ_y are the respective variances.

3. **Explained Variance (EV)** measures the variation in the model explained by the neural network algorithm [2].

$$EV = 1 - \frac{Var(y_i - \hat{y}_i)}{Var(y_i)}, \tag{2.10}$$

where Var is the Variance between the real data and the predicted outcome. Explained Variance (EV) is appropriate for nonlinear models, while the coefficient of determination (R^2) is good for linear models. The closer the value of R^2 or EV gets to 1, the better the data fit the model well.

2.3.4 Log Errors

1. **Mean Squared Log Error (MSLE)** measures the squared differences between the log-transformed real and predicted data values [40].

$$MSLE = \frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2 \tag{2.11}$$

MSLE produces a relative difference between the actual and predicted values by introducing logarithm [40].

2. **Root Mean Squared Log Error (RMSLE)** measures the square root of the squared differences between the log-transformed actual and predicted data values [41].

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(y_i+1)-\log(\hat{y}_i+1))^2} \tag{2.12}$$

The RMSLE reduces the effect of outliers on the model drastically more than RMSE [41].

2.3.5 Relative Errors

1. **REL:** Relative Error measures the difference between the predicted and real data values, expressed as a fraction or percentage of the real data.

$$REL = \sum_{i=1}^{N}\frac{(y_i-\hat{y}_i)^2}{y_i^2}. \tag{2.13}$$

2.4 Cross Validation

Apart from error metrics, cross-validation is another method of evaluating the performance of deep learning algorithms. This evaluation technique is appropriate for small datasets [42, 2]. This technique randomly splits the available data into k partitions, usually 4 or 5. We use $k-1$ cells to train k identical models. The rest of the section is used for testing and scoring the model. The average score from all the test scores is computed. RMSEor MSE can be used as the loss function in the algorithm.

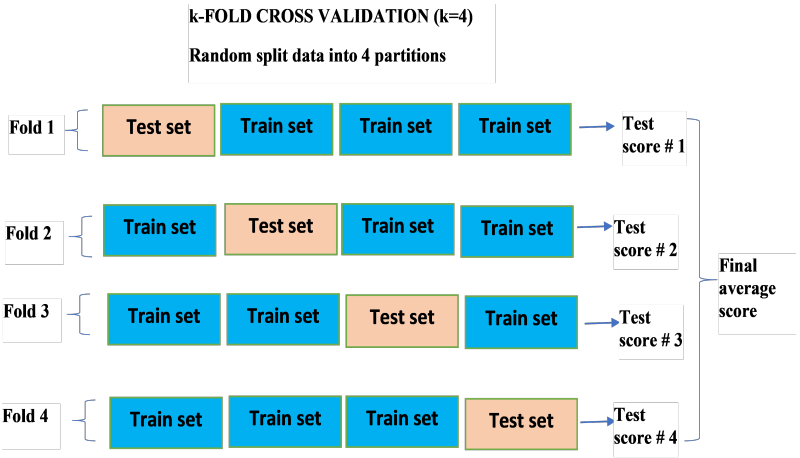

Figure 2.2: Cross Validation

Figure 2.2 shows the partition data into *k* fold for cross-validation.

3

DETERMINISTIC COVID-19 VACCINE MODEL

This chapter presents an application of deep learning to the COVID-19 vaccine efficacy model. Firstly, we analyze the impact of vaccination on controlling the spread of the virus. To achieve this, the feedforward neural network is used to learn the epidemiological parameters of the model. The parameters are then combined with the initial values of the COVID-19 data to obtain different dynamics of the model. Second, we implement and analyze a hybrid approach that combines residual neural networks with variants of recurrent neural networks. Error metrics and cross-validation show that the best approach is the hybrid approach called the residual neural network with the gated recurrent unit (ResNet-GRU). Data-driven simulation demonstrates that the higher the efficacy rate with the vaccination rate, the lower the infectiousness and primary reproduction number [2].

3.1 Materials and Methods

3.1.1 COVID-19 Vaccine Efficacy Model

Given the following population groups,

$S(t)$: The individuals that are susceptible per time

$I(t)$: The individuals that are infected per time

$R(t)$: The individuals that are recovered per time

$N(t)$: The total population per time

$$\begin{aligned} \frac{dS(t)}{dt} &= -\beta \frac{S(t)I(t)}{N(t)} - v\eta S(t) \\ \frac{dI(t)}{dt} &= \beta \frac{S(t)I(t)}{N(t)} - \gamma I(t) \\ \frac{dR(t)}{dt} &= \gamma I(t) + v\eta S(t), \end{aligned} \tag{3.1}$$

with $S(0) \geq 0, I(0) \geq 0, R(0) \geq 0$, $N(t) = S(t) + I(t) + R(t)$, the parameters of the model are: β transmission rate, γ recovery rate efficacy rate η and vaccination rate v. The population of individuals is constant per time throughout the vaccination period.

1. **The Basic Reproduction Number** R_0 is a number that measures the expected number of people affected directly by one infected individual [43]. [2]

$$R_0 = \frac{\beta}{\gamma}. \tag{3.2}$$

   The spread of the virus will decline if $R_0 < 1$ whilst severe effects imply that $R_0 > 1$. However, a stable and persistent situation means that $R_{=}1$ [2].

2. **Current Reproduction Number** measures the reproduction number per time

$$R_t = \frac{\beta_t}{\gamma_t}. \tag{3.3}$$

   The value of R_t is influenced by time-dependent transmission rate β_t and γ_t [2].

3. **The Effective Reproductive Number** R_{eff} is defined as[2]

$$R_{eff} = R_0 \frac{S}{N}. \tag{3.4}$$

   where S and N are the susceptible individuals and populations in an area, and R_0 is the basic reproduction number.

3.1.2 Epidemiology Informed Neural Network (EINN)

In an epidemiology-informed neural network (EINN) [2], the initial values and epidemiological parameters are incorporated into the loss function. EINN is inspired by a physics-informed neural network (PINN) [22]. EINN's output satisfies the differential equations

in model (3.1) because the model's residuals in (3.1) are encoded into the loss function. The derivatives of the output with respect to time are made through automatic differentiation [44]. The Mean Squared Error (MSE) is used as the loss function.

$$Loss = MSE_{data} + MSE_{sir} + MSE_{U_0}, \tag{3.5}$$

where

$$\begin{aligned} MSE_{data} =& \frac{1}{M}\sum_{j=1}^{M} ||I(t_j) - \hat{I}(t_j)||_2^2 \\ MSE_{sir} =& \frac{1}{M}\sum_{i=1}^{3}\sum_{j=1}^{M} ||e_i(t_j,\beta,\gamma)||_2^2 \\ MSE_{U_0} =& \frac{1}{M}\sum_{j=1}^{M} ||S_0 - \hat{S}_0||_2^2 \\ &+ \frac{1}{M}\sum_{j=1}^{M} ||I_0 - \hat{I}_0||_2^2 \\ &+ \frac{1}{M}\sum_{j=1}^{M} ||R_0 - \hat{R}_0||_2^2, \end{aligned} \tag{3.6}$$

where the residual $e_i, i = 1,\ldots,3$

$$\begin{aligned} e_1(t_j,\beta,\gamma) :=& \frac{dS(t_j)}{dt_j} + \beta\frac{S(t_j)I(t_j)}{N(t_j)} + v\eta S(t_j) \\ e_2(t_j,\beta,\gamma) :=& \frac{dI(t_j)}{dt_j} - \beta\frac{S(t_j)I(t_j)}{N(t_j)} + \gamma I(t_j) \\ e_3(t_j,\beta,\gamma) :=& \frac{dR(t_j)}{dt_j} - \gamma I(t_j) - v\eta S(t_j). \end{aligned} \tag{3.7}$$

The minimization of the loss function is done through the use of Adam optimizer where the network's parameters are updated [2].

Figure 3.1 is the diagram showing how the epidemiology-informed neural network (EINN) works. A mini-max scaler is used to preprocess data for the network to smooth training. In this chapter, 80 neurons for each 5 hidden layer. The Latin Hypercube Sam-

pling [5] is used to sample 3000 data points while 5000 data points are generated from spline cubic interpolation to impose the epidemiological constraints. The activation function used is the hyperbolic activation function (all hidden layers) and soft plus activation (output layer) [2]. EINN is implemented in Tensorflow in Python.

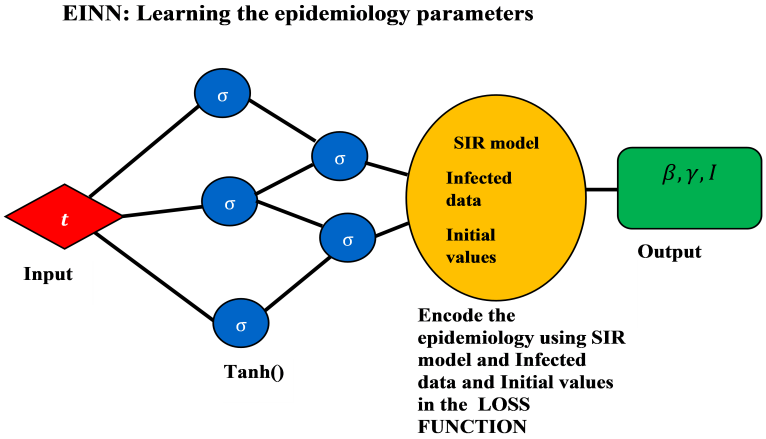

Figure 3.1: Epidemiology Informed Neural Network (EINN

Algorithm 1 Epidemiology Informed Neural Network (EINN)

1: set input arguments

t_j, $j = 1, \ldots, M$

vaccination rate v

efficacy rate η

epidemiology parameters β, γ

Initial values of the SIR model

2: construct and run neural network

specify weights, biases, and activation

3: compute the residuals

$e_i = \hat{y}_i - y_i$

4: calculate the loss function

encode the epidemiology

$$Loss = MSE_{data} + MSE_{SIR} + MSE_{U_0}$$

5: train the neural network

set epochs

6: return the solution

the epidemiology parameters β^*, γ^* and infected data I^*.

3.1.3 Deep Learning Algorithm for Vaccine Model

In this section, the deep learning algorithm for the vaccine model is presented.

Algorithm 2 Deep Learning Algorithm for Deterministic COVID-19 Vaccine Model

1: set parameters

vaccination rate $v = [0.0, 0.005, 0.01, 0.06, 0.1]$

efficacy rate $\eta = [0.94, 0.80]$

initialize epidemiology parameters β, γ

Initial values of the SIR model

2: run the EINN in Algorithm 5

the epidemiology parameters β^*, γ^* and infected data I^*.

3: compute the R_0 in (3.2) for different β, γ

4: compute the R_{eff} in (3.3)

5: use numerical solver to learn the dynamics of S, I, R

with different β^*, γ^*

6: analyze the impact of vaccination on infected group I

In Algorithm 4, different vaccination rates such as $0\%, 0.5\%, 1\%, 6\%, 10\%$ are encoded into the EINN in Algorithm 5 for each fixed efficacy rate $\eta = 94\%$ and $\eta = 80\%$. Different initialization is set for β and γ and the SIR model. The EINN outputs are different learned β^*, γ^*, I^* for different vaccination rates at fixed efficacy rates. Both R_0 and R_{eff} are computed from the learned output. The numerical solver is then used to learn the dynamics of S, I, R for different learned β^*, γ^*. A detailed analysis of the impact of vaccination is shown.

Figure 4.6 is a diagram showing how the hybrid approach works. COVID-19 data is obtained and split into 80% and 20% train and test sets respectively. data is preprocessed and passed into the EINN algorithm to learn different values of β and γ. The Bootstrap algorithm is deployed to construct a confidence interval around the learned β^* and γ^* using the Poisson distribution is used to resample the data [43] by running EINN 200 times. Next, the processed data is passed into the ResNet algorithm to obtain the output. The output from ResNet data is passed into LSTM, BiLSTM, and GRU to have a hybrid method ResNet-

LSTM, ResNet-BiLSTM, and ResNet-GRU. The error metrics are computed from the test data. A short-term prediction into the future is made for 15 days and a confidence interval is constructed. Cross-validation is carried out to evaluate the performance of LSTM, BiLSTM, GRU, and the hybrid approach [2].

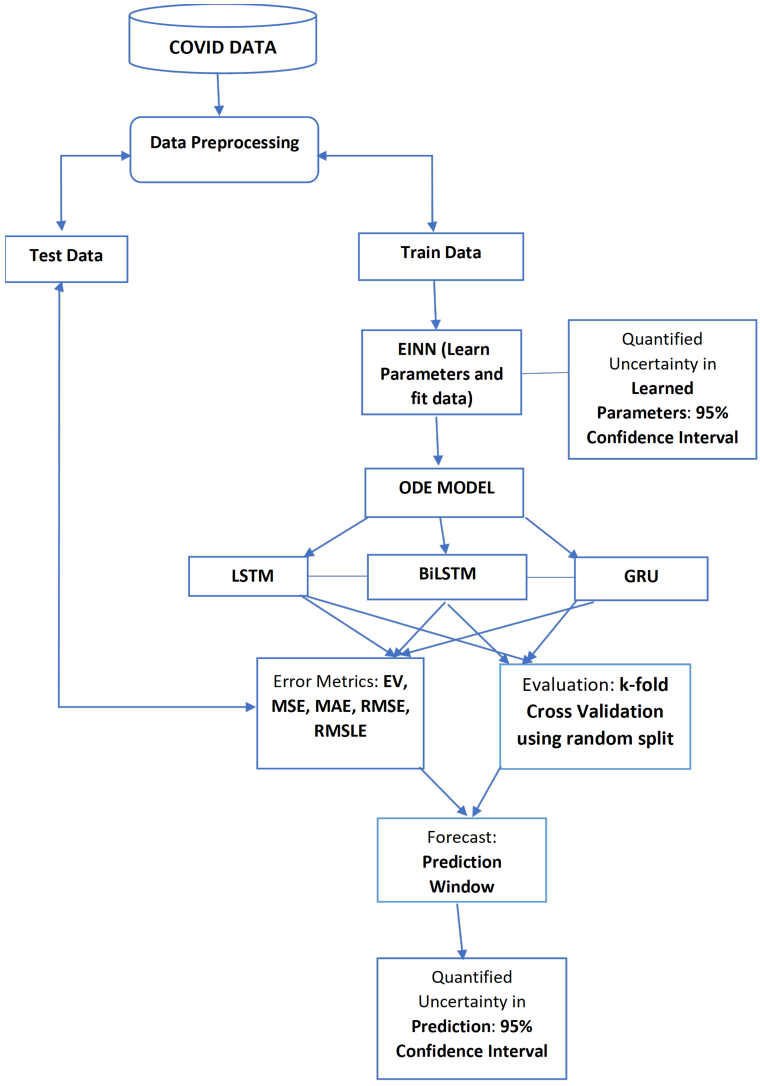

Figure 3.2: Deep learning for a hybrid approach.

3.2 Data Driven Simulations for Vaccine Model

The data for this work was obtained from Tennessee Health Department [45]. The available data as of the time of preparing this manuscript is from March 12, 2020, to September 16, 2021. The cumulative cases are used for the training of the EINN where a 7-day rolling

mean is applied to the data. In particular, the training data is from March 12, 2020, to December 16, 2020. This is the period of no vaccination program. The data is preprocessed using MinMax scaling. The EINN is implemented in Tensorflow while LSTM, BiLSTM, and GRU as well as k-fold Cross Validation are implemented in PyTorch.

State	Min	Max	STD	$Q-0.25$	$Q-0.5$	$Q-0.75$	Skewness	Kurtosis
Tennessee	1	717916	279540.7136	63503.5	275565	656462	0.10058	-1.6981

Table 3.1: Statistical Summary of COVID-19 Data

Table 4.1 gives a statistical summary of Tennessee COVID data.

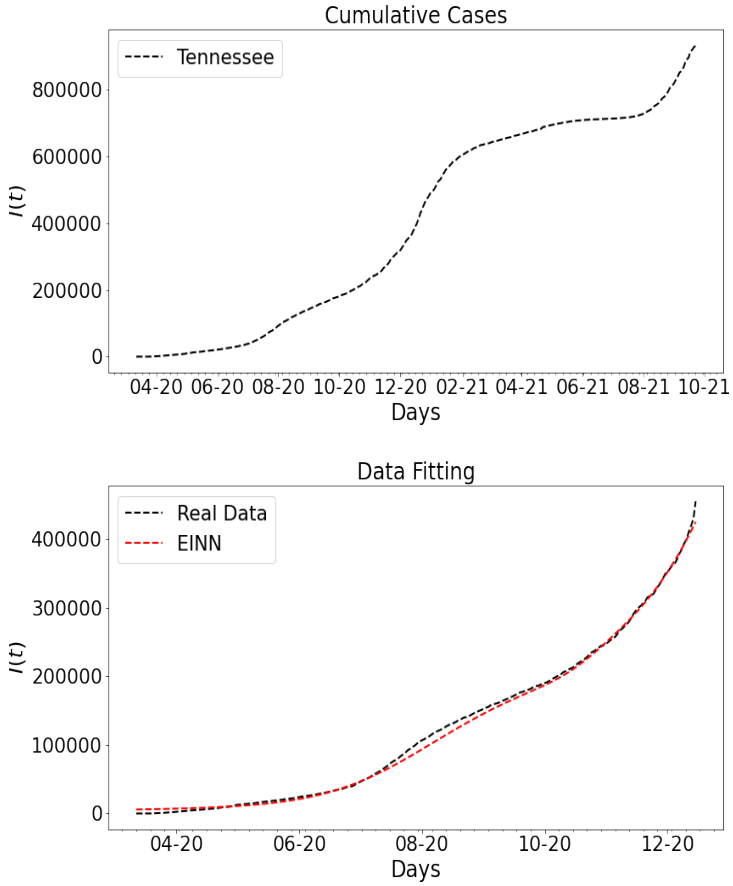

Figure 3.3: Cumulative Cases and Data Fitting.

Figure 3.3 includes the graphs for real COVID-19 data for Tennessee and the data fitting

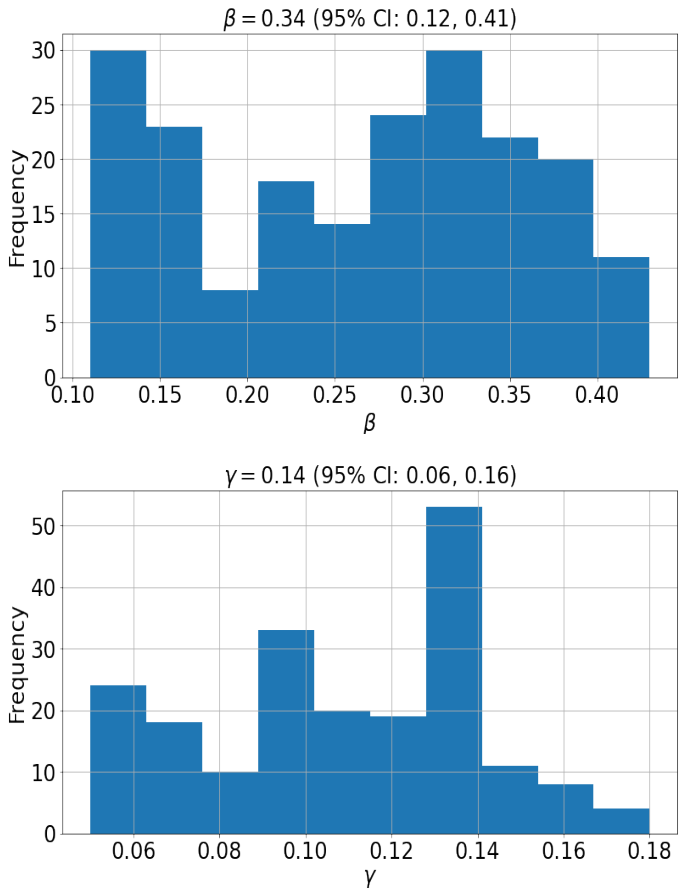

Figure 3.4: Confidence Interval for Learned β, γ.

for learning constant epidemiology parameters using EINN in Algorithm 5. Figure 3.4 shows the confidence interval constructed for learning the epidemiology parameters using Bootstrap algorithm [2].

Figure 4.8 shows the model without vaccination. The epidemiology-informed neural network is used to find the epidemiological parameters of the model and the values are passed into the numerical solver to obtain the shape of the graph. It takes 46 days to have a peak assuming there is no vaccination. Approximately 23.35% of the population will be infected with the virus [2].

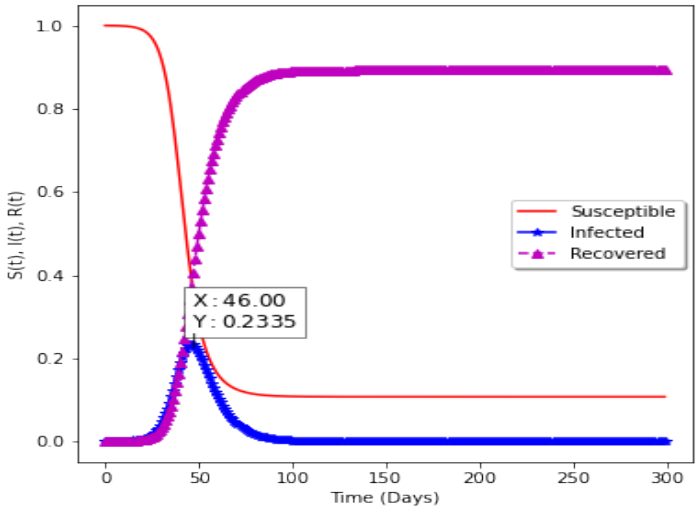

Figure 3.5: Model without vaccination.

Table 5.2 presents the impact of vaccination with an efficacy rate of 80%. In this table, at the different vaccination rates v, the EINN learns different $\beta's$ and $\gamma's$ to produce different basic reproduction numbers R_0 for the entire period. Equation (3.2) is used to compute the basic reproduction number. As vaccination rate v increases from $v = 0$ to $v = 0.5\%$, it leads to the reduction in R_0 value, from 2.5 to 2.05.

Efficacy rate($\eta = 80\%$) for Tennessee					
Vaccination Rate$(\%)$	β	γ	R_0	Infected$(\%)$	Days of Peak
0.00	0.35	0.14	2.5	23.35	46
0.5	0.45	0.22	2.05	15.78	10
1	0.4	0.21	1.90	13.19	10
2	0.4	0.22	1.82	11.19	9
3	0.38	0.22	1.73	9.63	7
6	0.3	0.2	1.5	7.30	4
10	0.18	0.13	1.38	6.67	2

Table 3.2: Impact of vaccination with (80%) efficacy rate.

The impact of vaccination with 94% efficacy is presented in Table 5.3. It can be observed from the table that for $v = 0.5\%$, R_0 decreased from 2.5 to 2.0. This means that the higher the efficacy rate, the faster the decline in the spread of the virus. This claim is further supported by the fact that when vaccination rate v increases to 10%, the R_0 in Table 5.2 is 1.38 and the R_0 in Table 5.3 is 1.31.

Efficacy rate($\eta = 94\%$) for Tennessee					
Vaccination Rate$(\%)$	β	γ	R_0	Infected$(\%)$	Days of Peak
0.00	0.35	0.14	2.5	23.35	46
0.5	0.44	0.22	2.0	15.04	10
1	0.45	0.23	1.96	13.76	9
2	0.42	0.22	1.91	11.92	8
3	0.43	0.24	1.79	10.11	7
6	0.27	0.16	1.89	7.79	5
10	0.17	0.13	1.31	6.57	1

Table 3.3: Impact of vaccination with (94%) efficacy rate.

Figure 4.10 shows the impact of vaccination with fixed efficacy at $\eta = 80\%$ on the infected population and the effective reproduction number that corresponds to the basic reproduction number computed in Table 5.2. In particular, it can be seen that as the respective effective reproduction number for vaccination rates of $0\%, 1\%, 3\%$ and 10%. The decrease in the effective reproduction number as shown in the graphs is indicative of the impact vaccination has on the infected group.

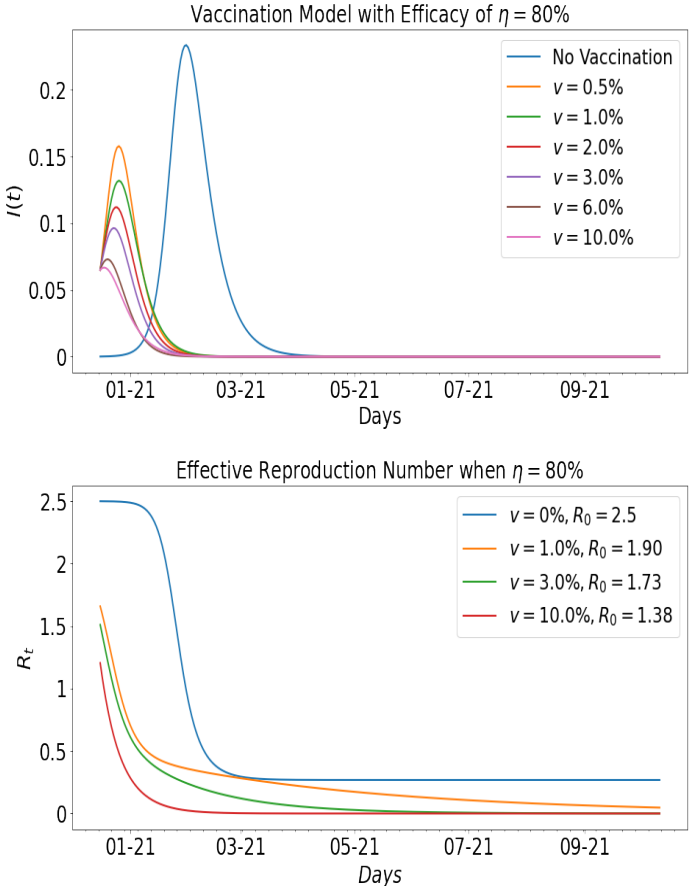

Figure 3.6: Vaccination impact and effective reproduction number for $\eta = 80\%$

Figure 4.11 shows the impact of vaccination with fixed efficacy at $\eta = 94\%$ on the infected population and the effective reproduction number that corresponds to the basic reproduction number computed in Table 5.3. In particular, the lower panel of the graph shows the respective effective reproduction number for vaccination rates of $0\%, 1\%, 3\%$, and 10%.

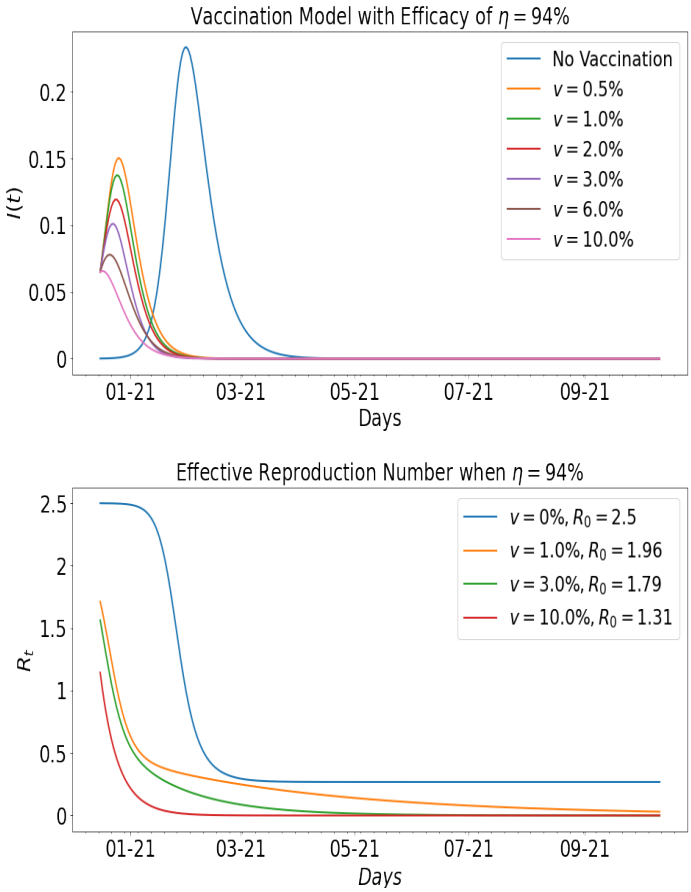

Figure 3.7: Vaccination impact and effective reproduction number for $\eta = 94\%$

Various data-driven simulations (using the real data and ResNet data) are performed for predicting daily infected cases using LSTM, BiLSTM, and GRU for Tennessee. The LSTM, BiLSTM, and GRU are used to learn these dynamics of infected cases. The parameter settings for data-driven simulations are given in Table 5.4.

	Parameter settings	
Approach	Parameter	Value
LSTM/BiLSTM/GRU	Learning rate	0.01
	Training Epochs	500
	Batch Size	100
	Layers	02
	Features	01
	Hidden units	24
ResNet	Learning rate	0.001
	Training Epochs	200
	Batch Size	45
	Layers	03
	Features	01
	Hidden units	50
Cross Validation	Learning rate	0.01
	Training Epochs	1000
	Batch Size	100
	Layers	01
	Features	01
	Hidden units	12

Table 3.4: Parameter settings for Data-Driven Simulations.

Figure 4.12 shows the data-driven simulation for learning the infected group using ResNet compared with the daily infected cases. For comparative analysis, the outcome from the ResNet is compared with the daily infected cases. The magenta-colored graph is ResNet. The data span a nine-month period: 16 January 2021 to 16 September 2021.

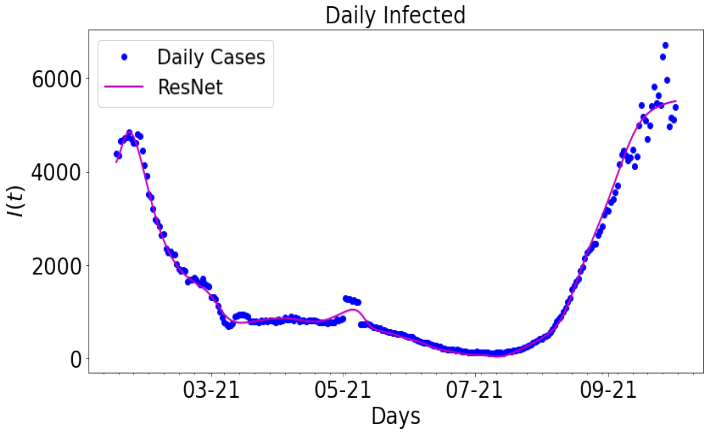

Figure 3.8: Data-driven simulation for real COVID-19 data and ResNet.

In Figure 4.13, the output from LSTM, BiLSTM, and GRU are plotted along with the daily infected cases. The real COVID-19 data is used to train the LSTM, BiLSTM, and GRU architectures. The results are plotted along with the daily infected cases. The graph of daily cases is colored blue. The black-colored graph shows the output for LSTM, the green-colored graph is the graph for BiLSTM, and the magenta-colored graph is the graph for GRU. The data spans nine months period: 16 January 2021 to 16 September 2021.

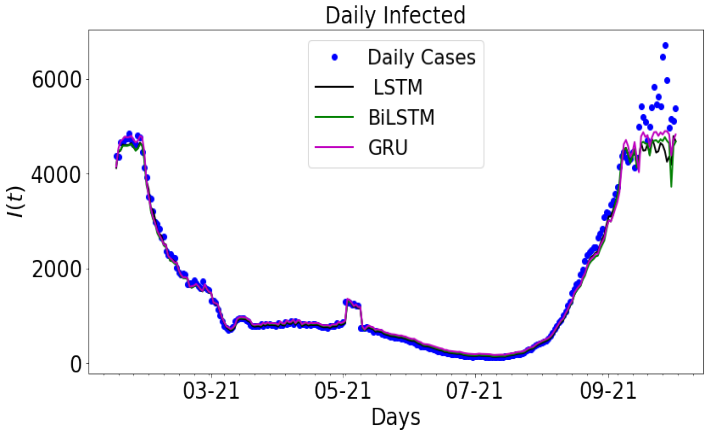

Figure 3.9: Data-driven simulation for LSTM, BiLSTM, and GRU.

A consideration is made for the hybrid approach: ResNet-LSTM, ResNet-BiLSTM, and ResNet-GRU. The outcome ResNet is used as the input data for LSTM, BiLSTM, and GRU. The parameter settings in Table 5.4 are applied to the different hybrid approaches to

generate the ensuring outputs.The outcome for the hybrid approach is shown in Figure 5.9. The daily new cases are plotted along with ResNet-LSTM, ResNet-BiLSTM, and ResNet-GRU. The outcome from the ResNet architecture is called the ResNet data, which is used to train LSTM, BiLSTM, and GRU. The daily cases and ResNet-LSTM are both colored blue while the ResNet-BiLSTM and ResNet-GRU are colored orange and green, respectively. The data spans nine months period: 16 January 2021 to 16 September 2021.

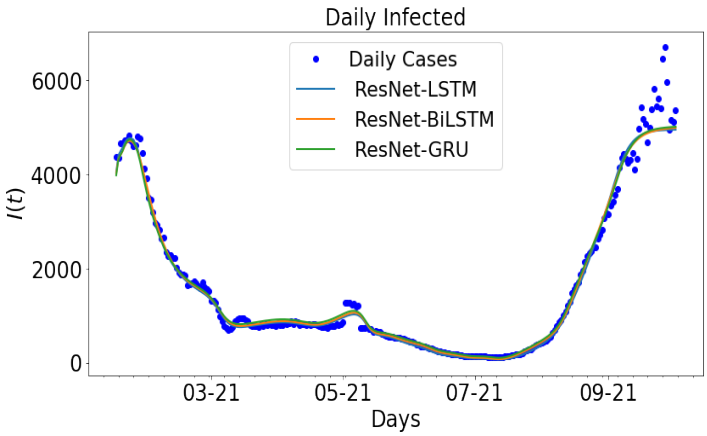

Figure 3.10: Data-driven simulation for ResNet-LSTM, ResNet-BiLSTM, and ResNet-GRU.

3.3 Error metrics for Data-Driven Simulations

The error metrics from data-driven simulation for real COVID-19 data and for ResNet data from Tennessee are shown below. The data is split into two train and test sets, while the train data is used to train the models, the error metrics are obtained from the difference between the test data of the actual COVID-19 data and the test data from the model.

Table 5.5 shows the error metrics table for LSTM, BiLSTM, GRU, ResNet-LSTM, ResNet-BiLSTM, and ResNet-GRU. It can be observed that the RMSE values for LSTM, BiLSTM, and GRU are 650.141249, 611.468730, 536.244478, respectively. GRU has the smallest RSME value. Additionally, the MAPE values (relative errors) are 0.159830, 0.150323, 0.131830, respectively, for LSTM, BiLSTM, and GRU. The model with the least MAPE value is GRU. Besides, the corresponding EV values for LTSM, BiLSTM, and

GRU are 0.891135,0.91171,0.922800. The model with the greatest EV value is GRU. This implies that, using real COVID-19 data, the model with the best error values is GRU. RMSE values for ResNet-LSTM, ResNet-BiLSTM, and ResNet-GRU are 269.863739, 255.473984, and 238.104349, respectively. The corresponding MAPE and EV values are 0.065988, 0.062470,0.058222 and 0.981658,0.981471,0.983856. ResNet-GRU is the hybrid approach with the overall best error values in terms of RMSE, MAPE, and EV.

Error Metrics for Tennessee				
Data	Model	RMSE	MAPE	EV
Real Data	LSTM	650.141249	0.159830	0.891135
	BiLSTM	611.468730	0.150323	0.911171
	GRU	536.244478	0.131830	0.922800
ResNet	ResNet-LSTM	269.863739	0.065988	0.981658
	ResNet-BiLSTM	255.473984	0.062470	0.983471
	ResNet-GRU	**238.104340**	**0.058222**	**0.983856**

Table 3.5: Error Metrics

Figure 5.10 presents each approach's RMSE and MAPE values. Both error metrics show that the approach with the least error values is ResNet-GRU.

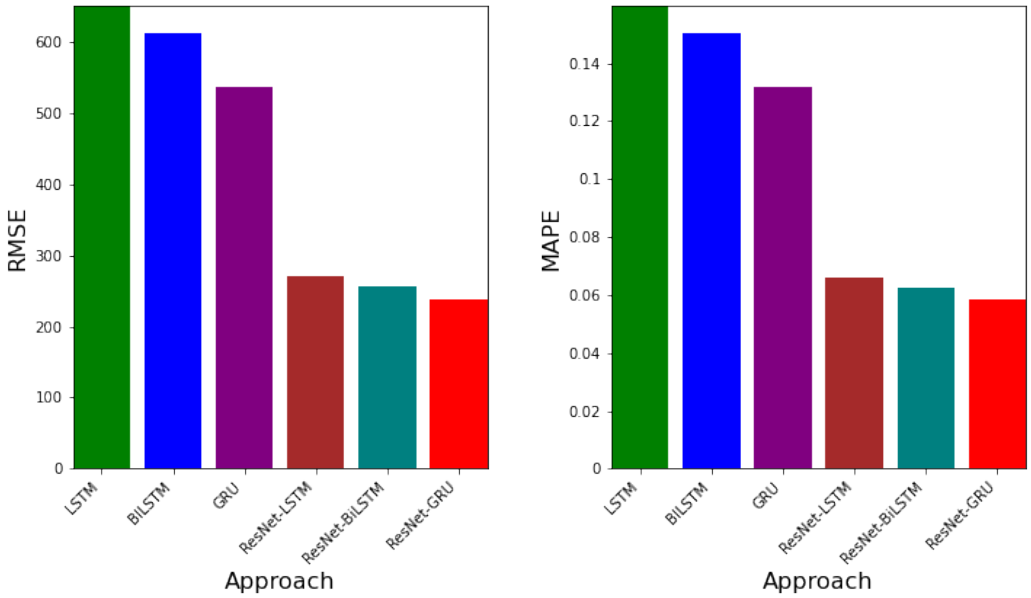

Figure 3.11: The graph of RMSE and MAPE values for each approach.

Figure 5.11 gives the EV values for each approach. It can be seen that the approach with the greatest EV is ResNet-GRU.

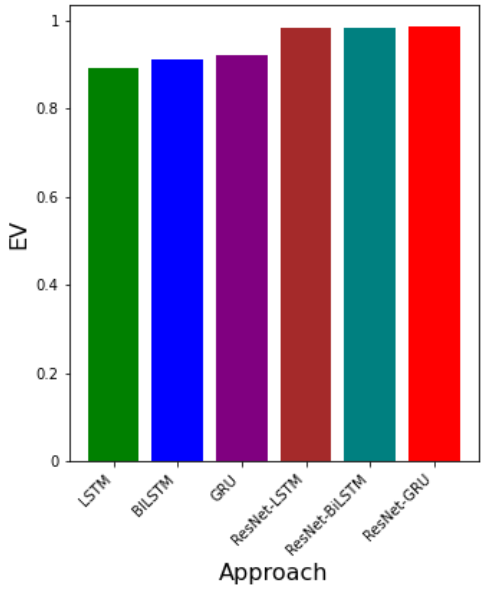

Figure 3.12: The graph of EV values for each approach.

The outcome of the errors obtained from the ResNet-LSTM, ResNet-BiLSTM, ResNet-

GRU, LSTM, BiLSTM, and GRU are presented. The error metrics RMSE, MAPE, and EV determine which algorithm learns the dynamic better. In particular, the RMSE is deployed. Smaller values of RMSE indicate the best algorithm in comparison to the others. The values of EV closer to 1 demonstrates lower variation in the predictive power of the neural network.

The mean values of the RMSE for all four values of k are plotted against each value of k for each model in a bar graph in Figure 5.12 for accurate COVID-19 data. The graph colored blue is LSTM, while the orange colored and green colored charts are BiLSTM and GRU, respectively. When $k = 4$, BiLSTM has the lowest average RMSE value, followed by LSTM and GRU. For $k = 5$, LSTM has the lowest average RMSE value, followed by GRU, then BiLSTM. It can be seen that as the value of k increases to 7, the smallest value of RMSE value is GRU, followed by LSTM, then BiLSTM.

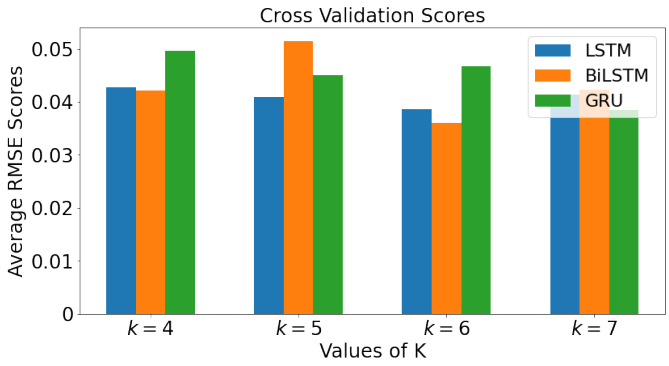

Figure 3.13: Cross Validation for real COVID-19 data.

Figure 5.13 demonstrates the mean value of the RMSE. All four values of k are plotted against each k for each model in a bar graph for ResNet data. The bar graph shows the cross-validation scores for Tennessee. When $k = 4$, the model with the lowest RMSE value is ResNet-BiLSTM, followed by ResNet-GRU, then ResNet-LSTM. With $k = 5$, the model with the lowest RMSE value is ResNet-GRU, followed by ResNet-BiLSTM, then ResNet-LSTM. When $k = 6$, it can be observed that ResNet-GRU has the lowest RMSE value, followed by ResNet-LSTM and ResNet-BiLSTM. As the value of k increases from 5 to 7,

it can be inferred that ResNet-GRU has the smallest average RMSE value.

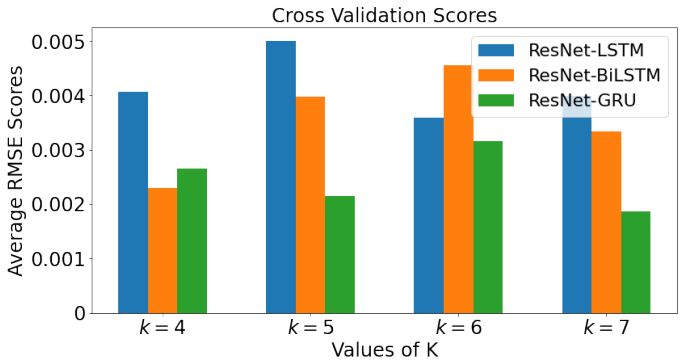

Figure 3.14: Cross Validation for ResNet data.

In Figure 5.14, the Bootstrap algorithm generates the confidence interval band after 10 independent runs. The lower 95% and upper 95% are given in the graph, and the mean prediction is also shown. The Bootstrap algorithm is deployed to run the model 10 times. The Poisson distribution is used to replicate the accurate COVID-19 data 10 times. For all runs, the mean and standard deviation is computed to ascertain a 95% confidence interval for the future predictions (15 days ahead). The confidence interval bound indicates a reasonable bound for future predictions.

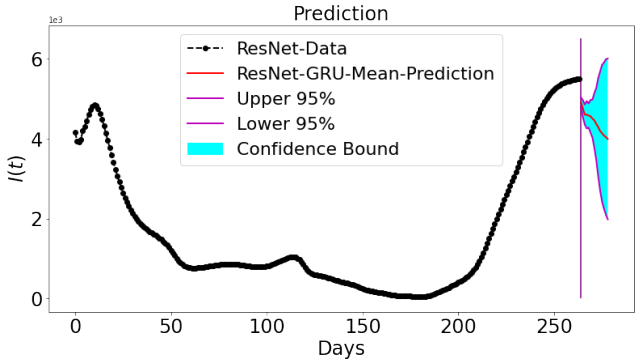

Figure 3.15: A confidence interval for short-term prediction using ResNet-GRU.

3.4 Conclusion

In this work, a COVID-19 vaccination model with vaccine efficacy has been developed and analyzed, where the epidemiological parameters of the model were obtained from the in-

verse problem solved by an epidemiological-informed neural network. A hybrid approach was implemented using a residual neural network with a recurrent neural network. Based on metric error values, the goal of investigating which type of hybrid approach produced a reliable future prediction of the infected cases was achieved. In particular, error metrics for data-driven simulations demonstrate that ResNet-GRU was the best hybrid approach because of its most minor root mean squared error, mean absolute percentage error values and most excellent explained variance value. K-fold cross-validation with random splitting using their average root mean squared error as mean score supported that ResNet-GRU is the best algorithm. Based on quantified uncertainty of 95% confidence interval, the short-term prediction results demonstrated our approach's overall effectiveness.

The epidemiological importance of this study was demonstrated in the vaccination model with vaccine efficacy. The model without vaccination revealed a high peak of infectiousness. However, the model with vaccination showed a lower height of infectiousness. Insights can be drawn from the fact that vaccinating the public using the same vaccination rate daily will curtail the spread of the disease faster. Our study confirmed this.

4

STOCHASTIC COVID-19 VACCINE MODEL

In this chapter, we present a stochastic epidemiology-informed neural network (SEINN) developed to learn the dynamics of a stochastic epidemic vaccine model. The system of stochastic differential equations reflecting the epidemiology of the model is discretized using Euler-Murayama [46] and encoded as a discrete loss. The learned outcomes are compared to the learned outcomes from a deterministic model using error metrics for data-driven simulation. The basic reproduction number(R_0) and the root mean squared error (RMSE) for both deterministic and stochastic models are contrasted.

4.1 Stochastic COVID-19 Vaccine Model

A mathematical model called the stochastic COVID-19 vaccine model simulates the COVID-19 pandemic's dynamics. The differential equations that make up the model's foundation describe how various demographic groupings interact. Given the following population groups

$S(t)$: The individuals that are susceptible per time

$I(t)$: The individuals that are infected per time

$R(t)$: The individuals that are recovered per time

$N(t)$: The total population per time

$w_i(t)$: Gaussian noise for each compartment $(i = 1, ..., 3)$

$$
\begin{aligned}
\frac{dS(t)}{dt} &= -\beta \frac{S(t)I(t)}{N(t)} - v\eta S(t) + \sigma_1 S(t) w_1(t) \\
\frac{dI(t)}{dt} &= \beta \frac{S(t)I(t)}{N(t)} - \gamma I(t) + \sigma_2 I(t) w_2(t) \\
\frac{dR(t)}{dt} &= \gamma I(t) + v\eta S(t) + \sigma_3 R(t) w_3(t)
\end{aligned} \tag{4.1}
$$

The equations mentioned above involve various parameters, including β which denotes the transmission rate, γ which denotes the recovery rate, η which denotes the efficacy rate, v which denotes the vaccination rate, and $\sigma_1, \sigma_2, \sigma_3$ which denote the level of fluctuations (noise) within each compartment. It is postulated that the populace remains invariant during the vaccination regimen, and that the initial quantity of each compartment is non-negative, with $S_0 \geq 0$, $I_0 \geq 0$, and $R_0 \geq 0$. Multiple simulations are carried out using different values of the variable v.

4.1.1 Nonlinear Incidence Stochastic COVID-19 Vaccine Model

The field of epidemiology characterizes the incidence rate as the number of fresh occurrences of illness within a given population during a specified period. A non-linear incidence rate is observed when the occurrence of new cases of illness is not directly proportional to the susceptibility of the population. The following demographic groups are presented: [47]

$S(t)$: The individuals that are susceptible per time

$I(t)$: The individuals that are infected per time

$R(t)$: The individuals that are recovered per time

$N(t)$: The total population per time

$g(S,I)$: Nonlinear incidence rate

$w_i(t)$: Gaussian noise for each compartment $(i = 1, ..., 3)$

$$\begin{aligned}
\frac{dS(t)}{dt} &= b - dS - g(S,I) + \gamma R - v\eta S(t) + \sigma_1 S(t) w_1(t) \\
\frac{dI(t)}{dt} &= g(S,I) - (d+\mu+\delta)I + \sigma_2 I(t) w_2(t) \\
\frac{dR(t)}{dt} &= \mu I - (d+\gamma)R + v\eta S(t) + \sigma_3 R(t) w_3(t)
\end{aligned} \tag{4.2}$$

The mathematical expression denoted by equation (4.2) characterizes the population dynamics, wherein the variable b signifies the recruitment rate, d denotes the natural death rate, μ represents the natural recovery rate, and γ is indicative of the rate at which recovered individuals lose immunity and revert to the susceptible state. The symbol δ denotes the mortality rate linked to the ailment. The variable v is utilized to denote the vaccination rate, whereas η is employed to represent the efficacy rate of vaccination. The variables $\sigma_1, \sigma_2, \sigma_3$ are provided without additional contextual information. The aim is to illustrate the amplitude of oscillations or noise that exists in each compartment, where $g(S,I)$ represents the non-linear incidence rate. The present investigation utilizes the parameter values outlined in the reference [47].The provided parameters are:

$$b = 1, d = 0.1, \mu = 0.05, \delta = 0.01, \gamma = 0.01, \sigma_1 = \sigma_2 = \sigma_3 = 0.1$$

.

$$g(S,I) = \frac{kS^h I}{S^h + \alpha I^h} \tag{4.3}$$

The incidence rate that is nonlinear in nature can be mathematically represented by the equation (4.3). The symbols h and k hold positive numerical values, while α signifies the psychological or inhibitory effect of the viral spread. The equation is formulated as the quotient of the multiplication of k, S^h, and I over the addition of S^h and αI^h, where S and I denote separate variables. The study investigates the impact of modifications in the noise level, ranging from $\sigma_i (i = 1, ..., 3) = 0.1$ to $\sigma_i (i = 1, ..., 3) = 3.0$, on the incidence rate while concurrently altering the values of k, h, and α. It is postulated that the initial value of each compartment is non-negative, with a specific emphasis on $S_0 \geq 0$, $I_0 \geq 0$, and $R_0 \geq 0$, while

the populace remains constant.

4.1.2 The Existence and Uniqueness of Solution

The solution of a stochastic process can be either weak or strong: *strong* implies that the solution is driven by the given Brownian motion; *weak* means the solution is driven by both the Brownian motion and the stochastic process y_t [48]. The existence of a strong solution always implies the existence of a weak the solution, but the reverse is not true. The uniqueness of a solution to the SDE in (4.1) with initial condition $y(0) = y_0$ can be determined using the stochastic Picard iteration [48]. The iteration involves re-writing (5.3) in the integral form:

$$y(t) = y_0 + \int_{t_0}^{\tau} f_1(y(\tau), \tau) d\tau + \int_{t_0}^{\tau} f_2(y(\tau), \tau) dB(\tau). \tag{4.4}$$

Then start with initial guess $\phi_0(t) = y_0$, Given a Brownian motion B, compute the approximations $\phi_1(t)$, $\phi_2(t)$, ... using the following iterative formula

$$\phi_{k+1}(t) = y_0 + \int_{t_0}^{\tau} f_1(\phi_k(\tau), \tau) d\tau + \int_{t_0}^{\tau} f_2(\phi_k(\tau), \tau) dB(\tau). \tag{4.5}$$

The conditions for the existence of a unique solution can be summarized as follows [48]. The iteration in (4.4) *converges to the exact solution in a mean squared sense*

1. *if both of the functions f_1 and f_2 grow at most linearly in y.*
2. *if both of the functions f_1 and f_2 are Lipschitz continuous.*

4.2 Deep Learning Algorithm for Stochastic Vaccine Model

The stochastic epidemiology-informed neural network (SEINN) is an extension of epidemiology-informed neural network (EINN) [2] which is inspired by a physics-informed neural network (PINN) [22] which incorporates the epidemiological parameters of the model, the SDE and initial values into the loss function. The loss function of SEINN satisfies the

system of stochastic differential equations (SDEs) in model (4.1). The Euler-Maruyama is used to discretize the system of SDEs and encode (4.1) into loss function. The method of automatic differentiation [44] is used to compute the derivatives of the output with respect to time for each residual equation. In this work, the loss is computed using the mean squared error

$$Loss = MSE_{data} + MSE_{euler} + MSE_{U_0}, \tag{4.6}$$

where

$$\begin{aligned}
MSE_{data} =& \frac{1}{M}\sum_{j=1}^{M} ||S(t_j) - \hat{S}(t_j)||_2^2 + \frac{1}{M}\sum_{j=1}^{M} ||I(t_j) - \hat{I}(t_j)||_2^2 + \frac{1}{M}\sum_{j=1}^{M} ||R(t_j) - \hat{R}(t_j)||_2^2 \\
MSE_{euler} =& \tau\frac{1}{M}\sum_{i=1}^{3}\sum_{j=1}^{M} ||e_i(t_j,\beta,\gamma)||_2^2 \\
MSE_{U_0} =& \frac{1}{M}\sum_{j=1}^{M} ||S_0 - \hat{S}_0||_2^2 \\
&+ \frac{1}{M}\sum_{j=1}^{M} ||I_0 - \hat{I}_0||_2^2 \\
&+ \frac{1}{M}\sum_{j=1}^{M} ||R_0 - \hat{R}_0||_2^2,
\end{aligned} \tag{4.7}$$

where the discrete loss $e_i, i = 1,\ldots,3$

$$\begin{aligned}
e_1(t_j,\beta,\gamma) :=& S_{k+1} - S \\
e_2(t_j,\beta,\gamma) :=& I_{k+1} - I \\
e_3(t_j,\beta,\gamma) :=& R_{k+1} - R.
\end{aligned} \tag{4.8}$$

and τ is some hyperparameter value for regularization.

Figure 4.1 shows the schematic diagram of the stochastic epidemiology-informed neural network (SEINN). The noisy data is passed into the neural network wherein the nonlin-

ear activation function is applied to the data. To encode the stochasticity in the model, the Euler-Maruyama is used to discretize the system of SDEs in (4.1). The data obtained is corrupted with some noise values. This corrupted data is passed into the input layer. All parameters of the model are initialized. The nonlinear activation function is applied to the processed data from the input layer. The discrete loss function is computed using the Euler-Maruyama Scheme. The output is the learned epidemiology parameters and dynamics of the model

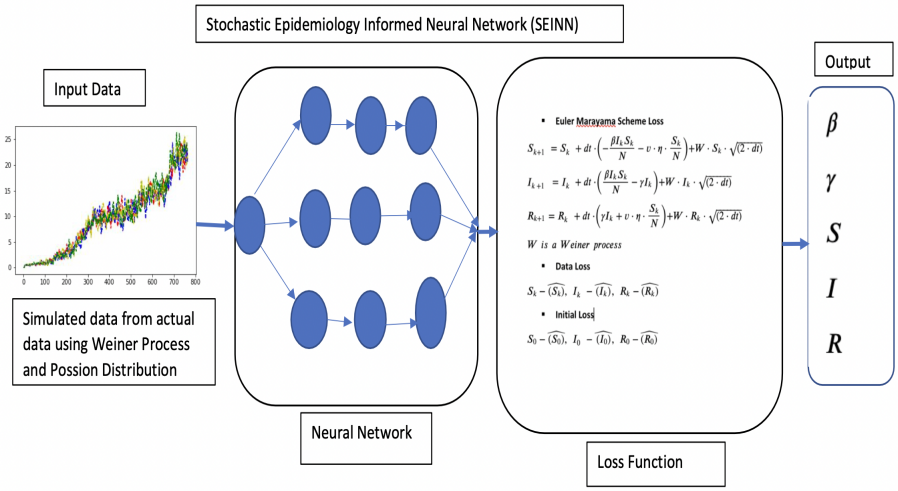

Figure 4.1: Schematic diagram of Stochastic Epidemiology Informed Neural Network (SEINN).

The SEINN Algorithm 5 for estimating the epidemiology parameters of stochastic COVID-19 vaccine model (**??**) is presented as follows.

Algorithm 3 SEINN for Stochastic COVID-19 Vaccine Model

1: set input arguments

t_j, $j = 1, \ldots, M$

vaccination rate v

actual COVID-19 data corrupted with noise level $\sigma_i (i = 1, ..., 3) = 0.1$

efficacy rate η

regularization parameter ε

initial values of epidemiology parameters β, γ

2: construct base neural network

specify weights, biases, and activation for each

3: compute the residuals

$e_i = \hat{y}_i - y_i$

4: calculate the discrete loss

encode the epidemiology using Euler Marayama

$$Loss = MSE_{data} + \varepsilon MSE_{euler} + MSE_{U_0}$$

5: train the composed neural network

set epochs

6: return the solution

the epidemiology parameters β^*, γ^* and S^*, I^*, R^*

Table 4.1: Parameter settings for Data-Driven Simulations for EINN and SEINN.

Model	Parameter	Value
Deterministic Model (EINN) [**?**]	Learning rate	1e-3
	Training Epochs	1900
	Regularization	1e-2
	Layers	07
	Features	03
	Hidden units	100
	Number of runs	10
	Optimizers	Adam
Stochastic Model (SEINN)	Learning rate	1e-5
	Training Epochs	30000
	Regularization	1e-5
	Layers	09
	Sigma Value (σ)	0.1
	Features	03
	Hidden units	150
	Number of runs	10
	Optimizer	Adam

The neural network is trained using Algorithm 5. The objective was to learn various values of β and γ. The COVID-19 data currently available is subject to corruption by a noise level denoted as σ_i, where i ranges from 1 to 3 and is equal to 0.1. Various simulations are conducted using varying vaccination rates, specifically $v = 1\%, 2\%, 3\%, 6\%, 10\%$. The algorithm is utilized to acquire knowledge regarding the susceptible, infected, and recovered individuals. The implementation is carried out utilizing the Pytorch library.

The SEINN Algorithm 4 for learning the dynamics of nonlinear incidence rate (**??**) is

presented as follows.

Algorithm 4 SEINN for Nonlinear Incidence Stochastic Vaccine Model

1: set input arguments

t_j, $j = 1, \ldots, M$

vaccination rate v

efficacy rate η

actual COVID-19 data

regularization parameter ε

epidemiology parameters $b = 1, d = 0.1, \mu = 0.05, \delta = 0.01, \gamma = 0.01$

noise level $\sigma_i (i = 1, ..., 3)$

True $g(S,I)$ with values k, h, α

2: construct base neural network

specify weights, biases, and activation for each

3: compute the residuals

$e_i = \hat{y}_i - y_i$

4: calculate the discrete loss

encode the epidemiology using Euler Marayama

$$Loss = (1-\varepsilon)L_{data} + \varepsilon L_{euler}$$

5: train the composed neural network

set epochs

6: return the solution

predicted nonlinear incidence rate $g^*(S,I)$ and S^*, I^*, R^*

Algorithm 4 specifies the selection of two regularization parameter values, namely $\varepsilon = 1e-1$ and $\varepsilon = 1e-3$, along with a training duration of 30000 epochs, a learning rate of $1e-3$, and the utilization of the Adam optimizer. These parameters are utilized to train a

neural network with the objective of learning diverse nonlinear incidence rates $(g^*(S,I))$. Cubic spline interpolation is employed as a means of generating additional data points. The compartments of susceptible, infected, and recovered are learned concurrently. The Tensorflow library is utilized for the implementation.

4.3 Data-Driven Simulations for Stochastic Vaccine Model

The COVID-19 statistics were obtained from the official website of the Tennessee Health Department. The data underwent preprocessing and scaling methodologies to transform the values into a uniform range of 0 to 100. The process of scaling was implemented in order to optimize the training of the models. The study employed the cumulative counts of individuals who have recovered from and contracted COVID-19 as the primary analytical components. After determining the initial values for the parameters β and γ, and keeping the efficacy rate η consistent, the models were subjected to training and evaluation under different vaccination rates $(0\%, 1\%, 2\%, 3\%, 6\%, 10\%)$.The utilization of the SEINN algorithm, as explained in reference [2], was implemented to obtain unique values of β and γ for every vaccination rate in the stochastic vaccine model. On the other hand, the EINN algorithm, as described in reference [2], was utilized to gain an understanding of the epidemiological parameters in the deterministic vaccine model.

The compartments acquired by the deterministic and stochastic models are depicted in Figure 5.1. The graph's upper segment depicts the acquired knowledge for a vaccination rate of zero percent, while the lower segment represents the acquired knowledge for a vaccination rate of ten percent, with an efficacy rate of 94 percent. The knowledge of compartments acquired by the models provides significant insights into the diverse dynamics of disease transmission in the context of various vaccination scenarios. In order to facilitate comparative analysis and evaluation, the EINN and SEINN experiments were replicated ten times to generate a sufficient quantity of data.

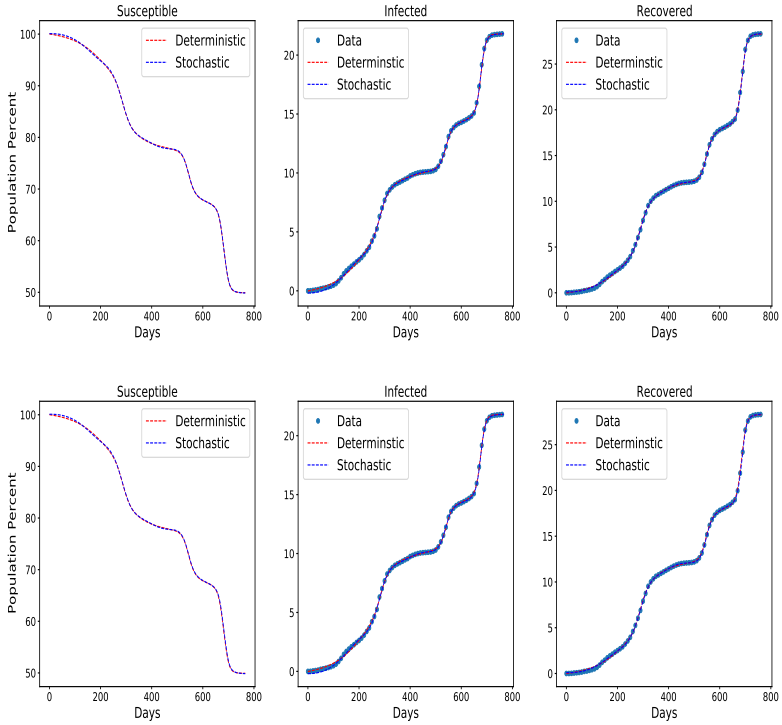

Figure 4.2: Data-Driven Simulation for deterministic and stochastic models when (**a**)$\eta = 94\%$ and $v = 0.0\%$; (**b**) when $\eta = 94\%$ and $v = 10\%$ for Tennessee.

The results of a comparative analysis between a deterministic model that employs EINN and a stochastic model that utilizes SEINN across different vaccination rates are presented in Table 5.2. The tabulated data displays the numerical values of the model parameters, specifically β and γ, alongside the basic reproduction number (R_0) that corresponds to each vaccination rate that has been selected. The presentation in tabular form is structured in a manner that showcases discrete columns that depict the vaccination rate (V), classification of the model (Deterministic or Stochastic), and the corresponding numerical values of β, γ, and R_0. The vaccination rates display a spectrum spanning from 0% to 10%. Distinct values of β and γ are acquired for each respective rate by employing different models. The text highlights that the stochastic model consistently produces lower estimates of R_0 in comparison to the deterministic model, across all vaccination rates, despite a fixed efficacy rate of 94%. The findings indicate that the incorporation of SEINN into the stochastic model leads to a reduction in the values of R_0, signifying a decreased probability of disease

transmission as compared to the deterministic model.

Model Comparison for Efficacy rate($\eta = 94\%$)				
V(%)	**Model**	β	γ	R_0
0.00	Deterministic	0.45	0.19	2.35
	Stochastic	0.30	0.15	**2.00**
1.0	Deterministic	0.45	0.24	1.86
	Stochastic	0.35	0.20	**1.77**
2.00	Deterministic	0.44	0.28	1.63
	Stochastic	0.26	0.18	**1.51**
3.00	Deterministic	0.46	0.31	1.50
	Stochastic	0.35	0.25	**1.43**
6.00	Deterministic	0.47	0.33	1.43
	Stochastic	0.26	0.20	**1.31**
10.00	Deterministic	0.18	0.15	1.19
	Stochastic	0.36	0.31	**1.13**

Table 4.2: This table summarizes the results from data-driven simulations for deterministic and stochastic models using different values of vaccination rates.

The root mean squared error (RMSE) and explained variance (EV) error values for data-driven simulations at different vaccination rates are displayed in Table 5.3. The presented data in tabular form facilitates a comparative analysis of the magnitudes of errors observed in a deterministic model versus a stochastic model. The presented data is organized in a tabular format, where each column represents a unique variable, including the vaccination rate (V), the model type (deterministic or stochastic), and their respective RMSE and EV values. The vaccination rates display a spectrum that varies from 0% to 10%. Fol-

lowing this, the models are evaluated to obtain the root mean square error (RMSE) and explained variance (EV) for each rate. The text provides empirical support for the superior accuracy of the stochastic model over the deterministic model, as indicated by the comparatively lower RMSE values. A lower value of the root mean square error (RMSE) implies a greater level of agreement between the predictions of the stochastic model and the observed data points. The presented data in tabular form provides an overview of the error metrics associated with simulations derived from the data. It demonstrates that the stochastic model outperforms the deterministic model in terms of efficacy. The utilization of the stochastic model results in decreased RMSE values, which suggests a greater level of accuracy in capturing the underlying epidemic dynamics and ultimately producing more effective predictive results. Table 5.3 presents a quantitative analysis of the error metrics for both the deterministic and stochastic models. The outcomes indicate that the stochastic model outperforms the deterministic model owing to its lower RMSE values. This implies that the utilization of the stochastic model is more effective in comprehending the intricate dynamics of the epidemic.

Error Values for Model Comparison for Efficacy rate($\eta = 94\%$)			
V(%)	**Model**	RMSE	EV
0.00	Deterministic	0.099021	0.999779
	Stochastic	**0.078609**	**0.999870**
1.0	Deterministic	0.099750	0.999789
	Stochastic	**0.079221**	**0.999869**
2.00	Deterministic	0.097739	0.999793
	Stochastic	**0.078905**	**0.999134**
3.00	Deterministic	0.099525	0.999790
	Stochastic	**0.079113**	**0.999869**
6.00	Deterministic	0.096687	0.999795
	Stochastic	**0.079043**	**0.999869**
10.00	Deterministic	0.099931	0.999796
	Stochastic	**0.079686**	**0.999863**

Table 4.3: This table summarizes the error values in terms of RMSE and EV from data-driven simulations for deterministic and stochastic models using different values of vaccination rates.

Figure 4.3 presents a bar graph that illustrates the correlation between the basic reproduction number R_0 and the vaccination rate v, as well as the relationship between RMSE and v. The graphical representation illustrates a negative correlation between the vaccination rate and the basic reproduction number (R_0) for both deterministic and stochastic models, as the former increases from 0% to 10%. The stochastic model exhibits a more rapid decrease in R_0 values compared to the deterministic model. The stochastic model exhibits lower RMSE values in comparison to the deterministic model.

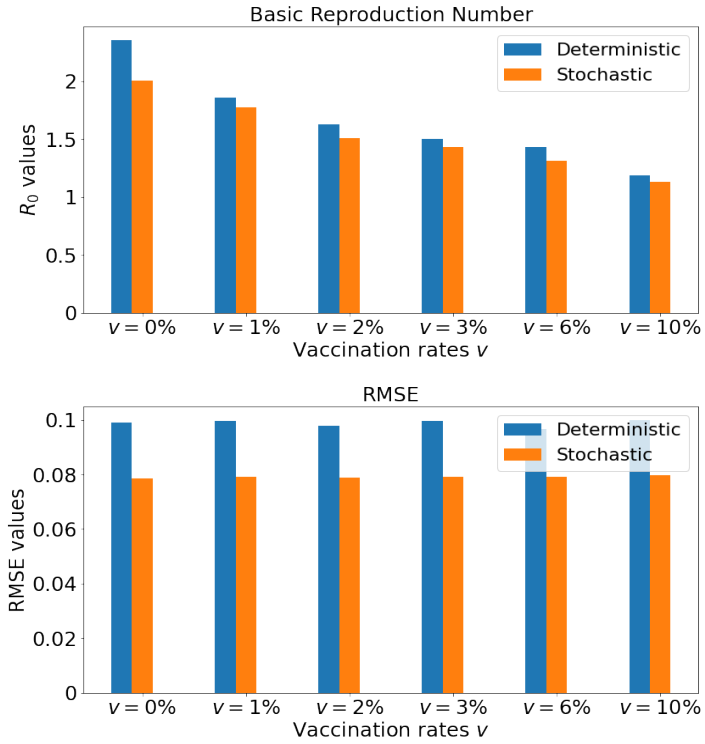

Figure 4.3: Basic Reproduction Number and Root Mean Squared Error. (**a**) Basic Reproduction Number versus vaccination rate (R_0 vs v); (**b**) Root Mean Squared Error versus vaccination rate ($RMSE$ vs v).

4.4 Data-driven simulation for Nonlinear Incidence Rate

The stochastic model was subjected to data-driven simulations at two distinct noise levels, specifically $\sigma = 0.1$ and $\sigma = 3.0$. Simulations were performed with different vaccination rates of $1\%, 6\%, and 10\%$, employing different combinations of parameters such as h, α, and k. The algorithm denoted as Algorithm 4 facilitates the acquisition of the expected non-linear incidence rate.

Figure 4.4 displays the nonlinear incidence rate acquired for a noise level of $\sigma = 0.1$. The left graph depicts the function $g(S,I)$, which was obtained by varying the parameters in equation (**??**) while keeping the vaccination rate constant at 1%. The specific parameter values used were $\alpha = 0.5$, $k = 0.2$, and $h = 2$. The subsequent plot in the sequence exhibits the mathematical function $g(S,I)$, which was obtained by implementing a vaccination rate of 6% while keeping the same parameter values of $\alpha = 0.5$, $k = 0.2$, and $h = 2$. The function

$g(S,I)$, as illustrated in the right graph, was obtained by utilizing a vaccination rate of 10% and the aforementioned parameter values. The nonlinear incidence rates, ascertained through analysis, demonstrate a strong correlation with the observed data.

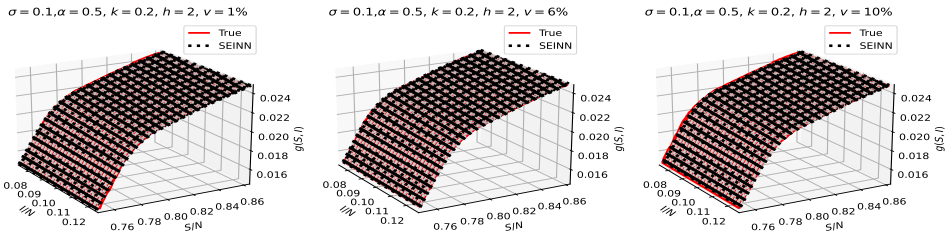

Figure 4.4: Graphical representation of nonlinear incidence rate for noise level $\sigma = 0.1$ when $h = 2, \alpha = 0.5, k = 0.2$ for different values of vaccination rates $v = 1\%$ (**left**), $v = 6\%$ (**middle**), $v = 10\%$(**right**).

Figure 4.5 depicts the nonlinear incidence rate acquired for a noise level of $\sigma = 0.1$ across different parameter combinations. The left graph depicts the function $g(S,I)$, which was obtained through the manipulation of parameters in equation (**??**), specifically $\alpha = 0.5$, $k = 0.2$, and $h = 2$, while utilizing a vaccination rate of 1%. The subsequent plot in the sequence exhibits the mathematical function $g(S,I)$, which was computed based on a vaccination rate of 6% and a unique collection of parameters, specifically $\alpha = 0.6$, $k = 0.3$, and $h = 2$. The function $g(S,I)$, illustrated in the graph on the right was obtained by employing a vaccination rate of 10% and the aforementioned set of parameters. The non-linear incidence rates that were acquired through learning demonstrate a strong correspondence with the data, proficiently encapsulating the underlying dynamics of the epidemic. Differences in parameter combinations result in variations in the shape and magnitude of the incidence rate curve. The aforementioned variations are suggestive of the impact that vaccination rates have on the transmission dynamics of the disease. The visual depictions provide valuable insights into the effectiveness of various vaccination rates in reducing the spread of the

disease and emphasize the importance of exploring multiple parameter combinations when studying the impact of interventions on the dynamics of an epidemic.

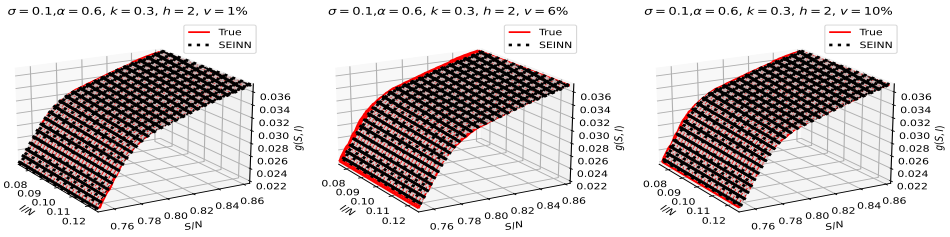

Figure 4.5: Graphical representation of nonlinear incidence rate for noise level $\sigma = 0.1$ when $h = 2, \alpha = 0.6, k = 0.3$ for different values of vaccination rates $v = 1\%$(**left**), $v = 6\%$(**middle**), $v = 10\%$(**right**).

Figure 4.6 illustrates the nonlinear incidence rate acquired for a noise level of $\sigma = 0.1$ across different parameter combinations. The left graph illustrates the function $g(S,I)$, which was obtained by implementing a vaccination rate of 1% and manipulating the parameters in equation (**??**), specifically $\alpha = 0.5$, $k = 0.2$, and $h = 3$. The subsequent plot in the sequence depicts the mathematical function $g(S,I)$, which was derived utilizing a vaccination rate of 6% and an equivalent collection of parameters. The function $g(S,I)$ is depicted in the graph on the right. It was obtained by maintaining the same parameters as those mentioned earlier while setting the vaccination rate at 10%. The results indicate that the model demonstrates proficiency in capturing the dynamics of the epidemic, as demonstrated by the accurate fitting of the learned nonlinear incidence rates to the observed data. It is noteworthy that an increase in the value of h coupled with a 10% increase in the vaccination rate results in a precise correspondence between the nonlinear incidence rates predicted by the model and the observed rates. The findings of this research suggest that a higher vaccination rate, when combined with a more pronounced non-linear incidence rate ($h = 3$), leads to a more accurate representation of the disease transmission dynamics. The

findings emphasize the importance of considering multiple parameter combinations when analyzing the impact of vaccination rates on the spread of the disease.

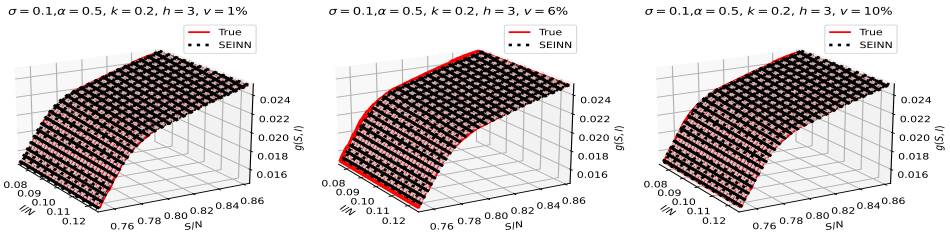

Figure 4.6: Graphical representation of nonlinear incidence rate for noise level $\sigma = 0.1$ when $h = 3, \alpha = 0.5, k = 0.2$ for different values of vaccination rates $v = 1\%$(**left**), $v = 6\%$(**middle**), $v = 10\%$(**right**).

Figure 4.7 illustrates the incidence rate that is nonlinear and has been acquired for a noise level of $\sigma = 0.1$. The parameters α and k underwent a reduction from 0.5 to 0.4 and from 0.2 to 0.1, respectively. The function $g(S,I)$ is depicted in the left graph, where a vaccination rate of 1% and parameter values of $\alpha = 0.4$, $k = 0.1$, and $h = 3$ were utilized in its computation. The subsequent graph in the sequence illustrates the mathematical function $g(S,I)$, which was derived utilizing the same parameter values as the previous graph, albeit with a vaccination rate of 6%. The function $g(S,I)$, as illustrated in the graph on the right, was obtained by utilizing a vaccination rate of 10% and employing the same set of parameters as previously mentioned. The accurate capture of epidemic dynamics by the model is demonstrated by the well-fitted learned nonlinear incidence rates that correspond to the actual data. The model demonstrates a notable level of precision in forecasting nonlinear incidence rates as the value of h is augmented and the values of α and k are diminished, specifically for the vaccination rate of 10%. The findings indicate that the model's ability to capture the nonlinear dynamics of disease transmission is improved by higher vaccination rates and lower values of α and k. The findings suggest that the interplay

between the rate of vaccination and the parameters α and k can exert a significant impact on the disease dynamics. Thus, it is imperative to carefully select these parameters in order to guarantee accurate modeling and prediction.

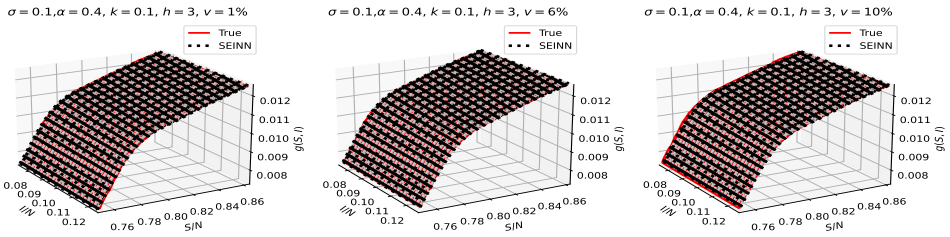

Figure 4.7: Graphical representation of nonlinear incidence rate for noise level $\sigma = 0.1$ when $h = 3, \alpha = 0.4, k = 0.1$ for different values of vaccination rates $v = 1\%$(**left**), $v = 6\%$(**middle**), $v = 10\%$(**right**).

This study aims to examine the dynamics related to the increase in noise levels within the range of $\sigma = 0.1$ to $\sigma = 3.0$, as illustrated in Figure 4.8. The left graph illustrates the acquired non-linear incidence rate $g(S,I)$ when subjected to a vaccination rate of 10% and parameter values of $\alpha = 0.5$, $k = 0.2$, and $h = 2$. In contrast, the second graph depicts the function $g(S,I)$ with a vaccination rate of 10%, but with increased parameter values of $\alpha = 0.6$ and $k = 0.3$, while holding the value of h constant at 2. The nonlinear incidence rates that were acquired through learning exhibit a high level of agreement with the empirical data, indicating that the model is skilled at capturing the underlying dynamics. It is significant to note that an optimal correspondence between the nonlinear incidence rates that are predicted and actual is evident upon augmenting the values of h, α, and k, in conjunction with a vaccination rate of 10%. The results suggest that the model demonstrates competence in comprehending the complexities of disease transmission dynamics in situations where there are increased α and k values, as well as heightened rates of vaccination. It is worth mentioning that the model possesses the ability to attain knowledge of $g(S,I)$ even

in the presence of high levels of noise (e.g., $\sigma = 3.0$). This highlights the model's ability to effectively handle noisy data and its competence in precisely capturing the fundamental patterns and dynamics of the epidemic.

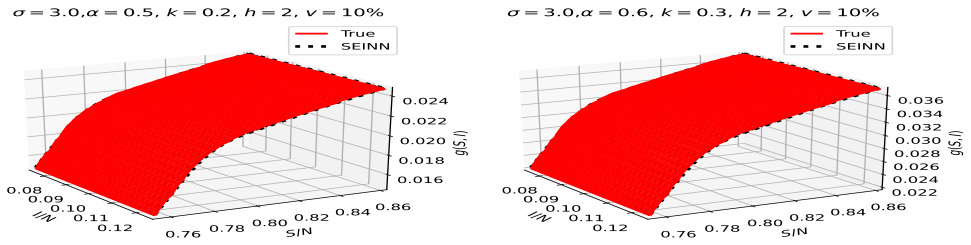

Figure 4.8: Graphical representation of nonlinear incidence rate for noise level $\sigma = 3.0$ when $h = 2, v = 10\%$ for $\alpha = 0.5, k = 0.2$ **(left graph)** and $\alpha = 0.6, k = 0.3$ **(right graph)**.

Figure 4.9 illustrates an investigation into the dynamics of heightened noise levels, which range from $\sigma = 0.1$ to $\sigma = 3.0$, while concurrently increasing the value of h. The left graph depicts the acquired nonlinear incidence rate $g(S,I)$ when the vaccination rate is at 10%, while the parameter values of $\alpha = 0.5$, $k = 0.2$, and $h = 3$ are held constant. In contrast, the graph depicted on the right demonstrates the function $g(S,I)$ when the vaccination rate is held constant at 10%, but with decreased parameter values of $\alpha = 0.4$ and $k = 0.1$, while keeping h as a variable. The observed nonlinear incidence rates exhibit a robust correlation with the empirical data, indicating that the model possesses the ability to effectively depict the underlying dynamics. The alignment between the predicted values and the nonlinear incidence rates is observed to be optimal as the values of h are augmented, and those of α and k are reduced while retaining a constant vaccination rate of 10%. This suggests that the model exhibits the ability to accurately capture the impact of changes in parameters and vaccination rates on the transmission dynamics of the disease. Furthermore, it is significant to mention that the model demonstrates its ability to acquire knowledge of the function g(S, I) even under significant levels of noise, as indicated by the

increased value of sigma equal to 3.0. The results indicate that the model exhibits proficiency in effectively handling data that contains noise, while also successfully capturing the fundamental patterns and dynamics of the system. epidemic.

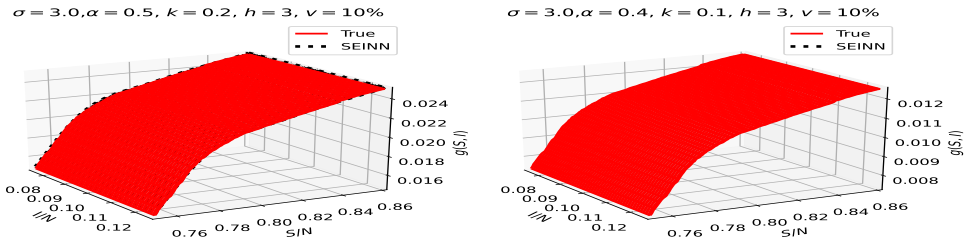

Figure 4.9: Graphical representation of nonlinear incidence rate for noise level $\sigma = 3.0$ when $h = 3, v = 10\%$ for $\alpha = 0.5, k = 0.2$ **(left graph)** and $\alpha = 0.4, k = 0.1$ **(right graph)**.

Table 5.4 presents an analysis of the effect of a decreased noise level of $\sigma = 0.1$ on the acquired nonlinear incidence rate in the model. The presented table exhibits diverse error metrics across discrete simulations conducted under different parameter configurations, while upholding a consistent vaccination rate of 10%. By analyzing these metrics, it is possible to evaluate the effectiveness of the model in capturing the non-linear occurrence rate across various parameter configurations. Decreased values of error measures, such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), indicate improved accuracy in the prediction of nonlinear incidence rates when compared to the actual values.

Table 4.4: Error metrics for noise level $\sigma = 0.1$ in the model

v	α	k	h	RMSE	MAPE	EV	REL
10%	0.5	0.2	2	0.0000812	0.00372	0.9992	0.00268
	0.6	0.3	2	0.000121	0.00370	0.9992	0.00266
	0.5	0.2	3	0.0000734	0.00333	0.9993	0.00189
	0.4	0.1	3	0.0000416	0.00377	0.9992	0.00275

The present study examines the effects of heightened noise levels, specifically with a standard deviation of 3.0, on the acquired nonlinear incidence rate in the model, as depicted in Table 5.5. The table presented exhibits a variety of error metrics for diverse simulations utilizing different parameter combinations while upholding a consistent vaccination rate of 10%. Through the analysis of these metrics, it is possible to evaluate the effectiveness of the model in capturing the non-linear incidence rate across a range of parameter configurations and increased levels of noise. Despite the increased level of noise, the model achieves a satisfactory level of accuracy, as demonstrated by the relatively low values of error metrics such as RMSE and MAPE. The SEINN algorithm's capacity to accurately capture the underlying dynamics of the nonlinear incidence rate is demonstrated by its robustness and effectiveness, even in the presence of elevated noise levels.

Table 4.5: Error metrics for noise level $\sigma = 3.0$ in the model

v	α	k	h	RMSE	MAPE	EV	REL
10%	0.5	0.2	2	0.0000973	0.00446	0.9994	0.00339
	0.6	0.3	2	0.000145	0.00444	0.9994	0.00336
	0.5	0.2	3	0.000102	0.00461	0.9991	0.00407
	0.4	0.1	3	0.0000509	0.00461	0.9991	0.00407

Figure 4.10 illustrates the comparative analysis of the Root Mean Squared Error (RMSE)

values between two models, one with a noise level of $\sigma = 0.1$ and the other with a noise level of $\sigma = 3.0$. The utilization of graphical representation enables a comparative evaluation of the effectiveness of both models in capturing the inherent dynamics of the non-linear incidence rate.After scrutinizing the graph, it is evident that the model with a lower noise level of $\sigma = 0.1$ consistently produces better RMSE values in comparison to the model with a higher noise level of $\sigma = 3.0$. The expected result is attributed to the decreased level of noise, which leads to a more precise and less corrupted input signal. Consequently, this process enables expedited and seamless instruction, resulting in superior results. The effectiveness of the suggested methodology remains unimpaired despite heightened levels of noise, which is a notable observation. The model's root mean square error (RMSE) values demonstrate a marginal rise when subjected to a noise level of 3.0, in contrast to the model operating under a noise level of 0.1. This implies a marginal rise in the discrepancy between the anticipated non-linear frequency of occurrence and the factual frequency of occurrence. However, the overall performance of the model remains satisfactory, and it adeptly captures the fundamental dynamics of the nonlinear incidence rate.

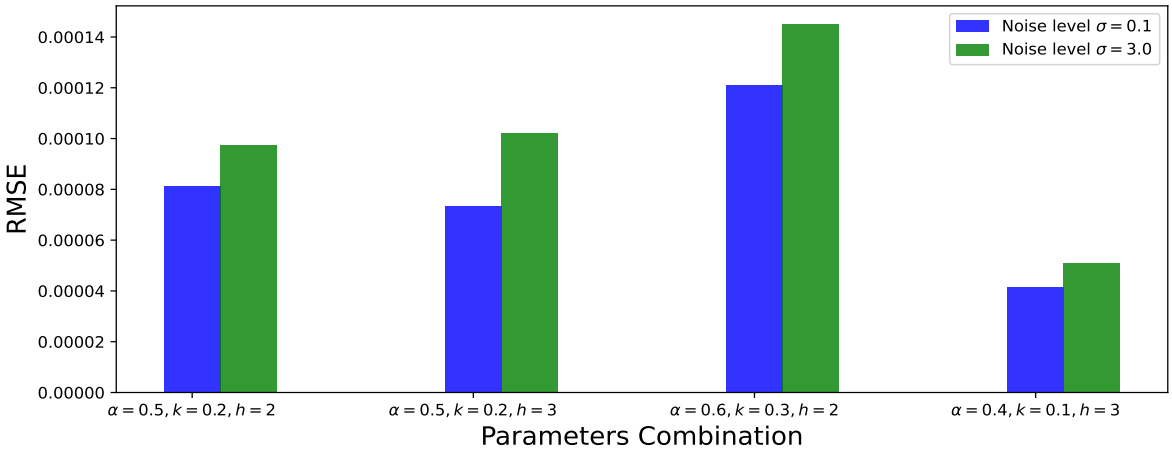

Figure 4.10: Comparative Analysis of model with noise level of $\sigma = 0.1$ and $\sigma = 3.0$.

4.5 Computational Analysis of SEINN

This section provides a comprehensive analysis of the effects of perturbations in the parameters of the model and the issue of overfitting in the proposed approach. By conducting sensitivity analysis, evaluating model performance, and analyzing computational time, significant insights can be gained pertaining to the proposed methodology's resilience, capacity for generalization, and efficacy. The appropriate determination of the regularization parameter ε is crucial in ensuring the reliability and efficacy of the model in real-world situations.

4.5.1 Sensitivity Analysis

Sensitivity analysis is a technique utilized to evaluate the effects of variations in a model's inputs or parameters on the model's output or outcome. The aforementioned refers to a mechanism that is employed to understand the behavior of a model and assess its soundness. This study involves the manipulation of certain parameters within the SEINN model and the subsequent analysis of the resulting impact on error metrics.

The impact of modifying the number of neurons and layers in a stochastic epidemiology-informed neural network (SEINN) model on error metrics, while keeping a consistent regularization parameter of $\varepsilon = 1e-1$, is presented in Table 4.6. The tabular data presented showcases a comprehensive array of performance metrics, such as root mean squared error (RMSE), mean absolute percentage error (MAPE), explained variance (EV), and relative error (REL), for all conceivable permutations of layers and neurons. After performing a comparative analysis between two models with different numbers of neurons (32 and 64) while maintaining a constant number of layers, it was observed that the model with 32 neurons demonstrated superior performance. The fifth row of the table displays the optimal outcomes attained by a model consisting of five layers and 32 neurons. The statistical metrics of root mean square error (RMSE), mean absolute percentage error (MAPE), and relative error (REL) were computed and determined to be 1912, 0.00254, and 0.00131,

respectively.

Table 4.6: Effect of $\varepsilon = 1e-1$ on SEINN using error metrics for different layers $(3,4,5)$ and neurons $(32,64)$.

Regularization parameter $\varepsilon = 1e-1$					
Layers	Neurons	RMSE	MAPE	EV	REL
3	32	4850	0.00644	0.9993	0.00593
3	64	6170	0.00819	0.9957	0.01044
4	32	3493	0.00464	0.9994	0.00378
4	64	4159	0.00553	0.9988	0.00653
5	**32**	**1912**	**0.00254**	**0.9996**	**0.00131**
5	64	2864	0.00361	0.9991	0.00250

The table presented as Table 4.7 indicates that a decrease in the regularization parameter to $\varepsilon = 1e-3$ has the potential to mitigate overfitting and, as a result, enhance the efficacy of error metrics. An alternative approach to mitigate the decrease in regularization and potentially enhance the overall effectiveness is to adjust the number of neurons or add an additional layer. The findings indicate that achieving optimal efficiency within the SEINN framework requires a delicate balance between the regularization parameter and the number of neurons and layers. The combination of 4 layers and 32 neurons, as specified in the fifth row, yields the optimal error metric values.

Table 4.7: Effect of $\varepsilon = 1e-3$ on SEINN using error metrics for different layers $(3,4,5)$ and neurons $(32,64)$.

Regularization parameter $\varepsilon = 1e-3$					
Layers	Neurons	RMSE	MAPE	EV	REL
3	32	3295	0.00438	0.9993	0.00296
3	64	3244	0.00431	0.9990	0.00328
4	32	2175	0.00289	0.9995	0.00199
4	64	2901	0.00385	0.9991	0.00248
5	**32**	**1277**	**0.00160**	**0.9998**	**0.000631**
5	64	6371	0.00847	0.9991	0.01122

Figure 4.11 illustrates the effect of the regularization parameter $epsilon = 1e-1$ on the SEINN method. Root mean squared error (RMSE) and mean absolute percentage error (MAPE) are used as performance measurements in the evaluation of the model's effectiveness. According to the RMSE graph, five layers and 32 neurons are used to produce the ideal arrangement of layers and neurons, yielding a value of 1277. The arrangement that yields the most unsatisfactory results has five layers, each of which has 64 neurons, and an output of 6371 numbers. The design with 5 layers and 32 neurons achieves the maximum degree of performance, as shown by the MAPE value of 0.00160, according to an examination of the MAPE graph. A score of 0.00827 was obtained for the configuration with the least beneficial outcome, which consists of five layers and 32 neural units.

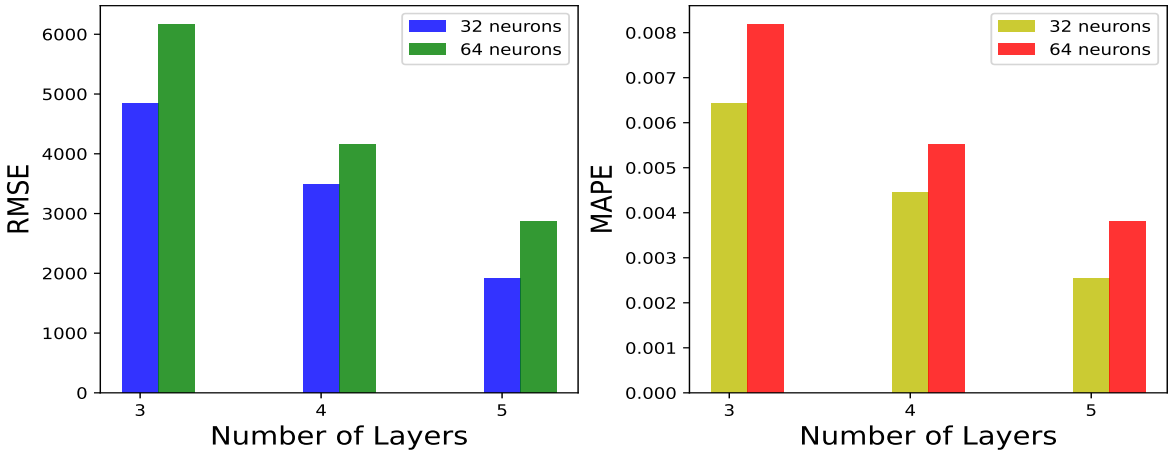

Figure 4.11: Graphical representation of effects of regularization parameter $\varepsilon = 1e-1$ using RMSE and MAPE.

The findings illustrated in Figure 4.12 demonstrate that a reduction in the value of the regularization parameter, from $\varepsilon = 1e-1$ to $\varepsilon = 1e-3$, results in significant improvements in the SEINN model's performance. The graphical representation of the root mean square error (RMSE) displays the optimal configurations that exhibit outstanding performance. The structure is composed of four tiers, with each tier containing 34 individual neurons. The outcome aligns with the discoveries illustrated in Table 4.7. The Mean Absolute Percentage Error (MAPE) graph offers a performance assessment that is akin to that of the Root Mean Square Error (RMSE) graph. The aforementioned diagram highlights the significance of carefully choosing suitable hyperparameters for the SEINN model to achieve maximum efficiency, along with the requirement for meticulous calibration regarding the regularization parameters.

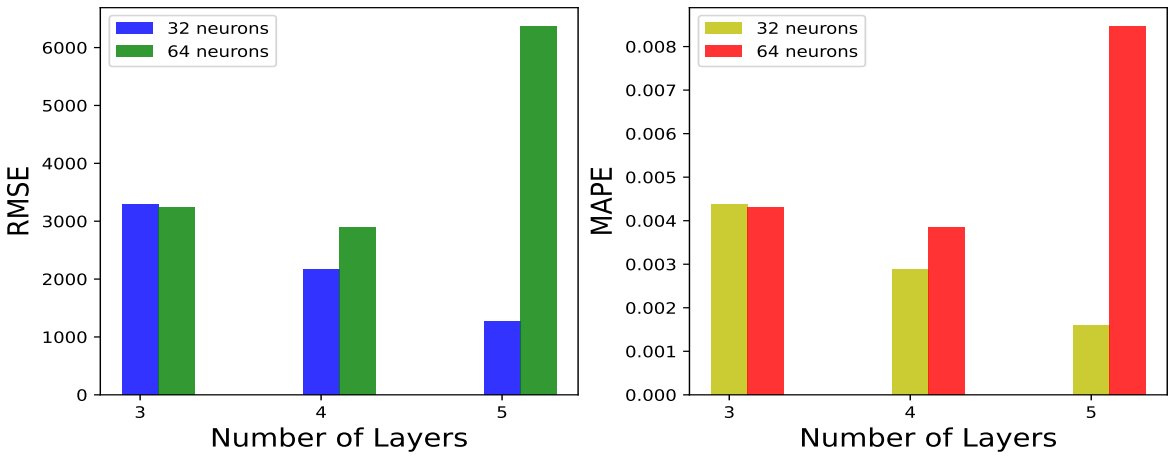

Figure 4.12: Graphical representation of effects of regularization parameter $\varepsilon = 1e-3$ using RMSE and MAPE.

4.5.2 Overfitting Analysis

The process of evaluating a deep learning model's performance on data that it has not been trained on is commonly known as overfitting analysis. The objective is to evaluate if the model exhibits tendencies towards overfitting by analyzing its efficacy on the training dataset in contrast to its efficacy on the validation or test dataset. The scrutiny of overfitting can serve as a technique to steer the choice of suitable hyperparameters for a given model and to evaluate the model's capacity to generalize. The current investigation evaluates the impact of modifying the regularization parameter ε and identifying suitable hyperparameters on the occurrence of overfitting. The SEINN algorithm utilized a partitioning technique to divide the input data into two distinct sets: a training set, which accounted for 80% of the data, and a validation set, which accounted for 20% of the data. The study maintained a constant number of layers at four, while the number of epochs was manipulated within a range of 30000 to 60000.

The SEINN model's ability to mitigate overfitting is demonstrated in Figure 4.13, where the effect of the regularization parameter $\varepsilon = 1e-1$ is depicted. The graph located on the right-hand side of the document illustrates the contrast between the losses incurred during the training and validation phases. The experiment was conducted using a neural network

architecture consisting of four layers and was trained for a total of 30,000 epochs. After applying a logarithmic transformation to the epoch count, it becomes apparent that the training and validation losses display disparity during the early stages. However, they eventually converge to a comparable value after a specific number of epochs. By contrast, the graph depicted on the left was produced by employing 60000 epochs and incorporating four layers. It is apparent that the values of the training and validation losses are identical.

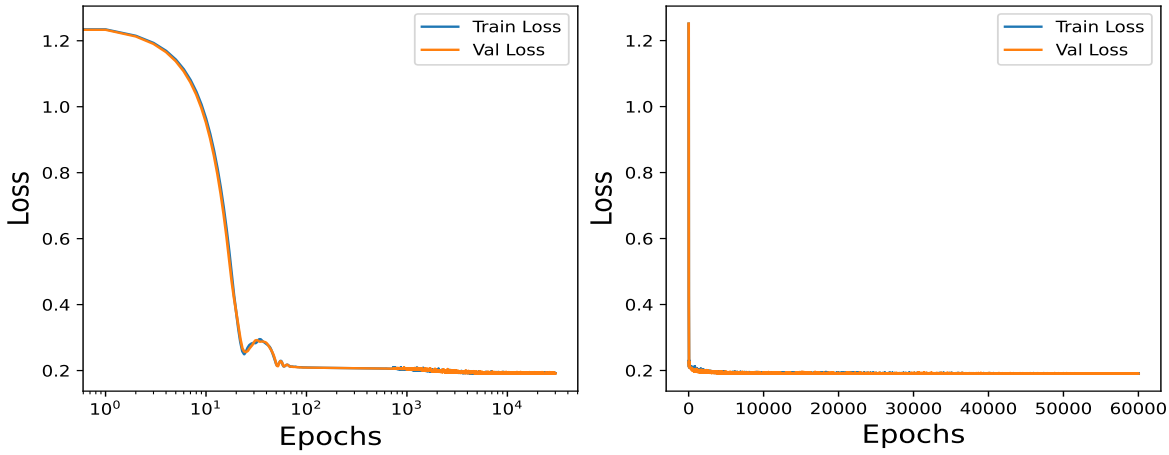

Figure 4.13: Graphical representation of effects of regularization parameter $\varepsilon = 1e-1$ preventing overfitting using epochs 30000 and 60000

Figure 5.9 illustrates the effect of the regularization parameter $\varepsilon = 1e-3$ on reducing overfitting in the SEINN model. The graph on the right-hand side illustrates a comparison between the training and validation losses. These losses were obtained by utilizing 30000 epochs and 4 layers. Upon the application of a logarithmic transformation to the epoch count, it is evident that the training and validation losses exhibit disparities in the initial stages, but ultimately converge to a similar value after a certain number of epochs. The graph illustrated on the left was generated by implementing four layers and utilizing 60000 epochs for the purpose of comparison. The congruity of the losses observed during the training and validation phases is evident.

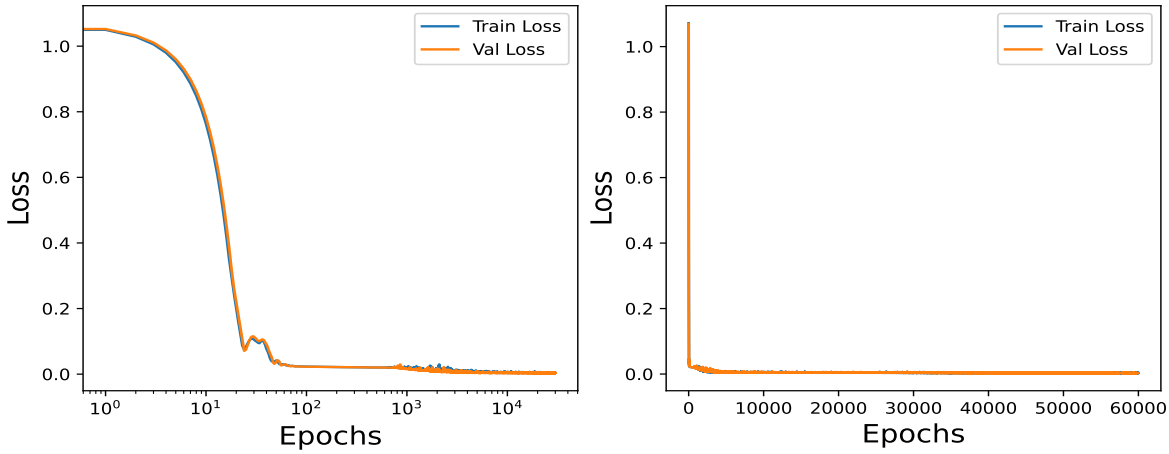

Figure 4.14: Graphical representation of effects of regularization parameter $\varepsilon = 1e-3$ preventing overfitting using epochs 30000 and 60000

Table 4.8 presents an examination of data-driven simulations utilized for training with different numbers of epochs $(30000, 40000,\ 60000)$ and a range of values for the regularization parameter ε. The purpose of the table is to evaluate the impact of modifying the number of epochs on the loss function, which serves as a measure of the model's capacity to adhere to the data. The table presented provides clarification on how the aforementioned factors impact the loss function, which serves as an indicator of the model's ability to accurately represent the data. The findings indicate an optimal number of epochs beyond which further training does not significantly improve the effectiveness of the model. Furthermore, the process of selecting an appropriate value for the regularization parameter is of great significance in addressing the issue of overfitting and enhancing the model's ability to generalize.

Table 4.8: Analyzing SEINN for different epochs $30000, 40000, 6000$

Epochs	Epsilon ε	Loss
30000	$1e-1$	$1.91 \times 10^{-}1$
30000	$1e-3$	$4.21 \times 10^{-}3$
40000	$1e-1$	$1.91 \times 10^{-}1$
40000	$1e-3$	$2.68 \times 10^{-}3$
60000	$1e-1$	$1.91 \times 10^{-}1$
60000	$1e-3$	$3.40 \times 10^{-}3$

4.5.3 Computational Time Analysis

The evaluation of the computational duration required for the SEINN training process involves measuring the time taken by the CPU to execute the training procedure of the SEINN model on a specific dataset. The significance of this analysis pertains to its ability to ascertain the efficacy of the model training methodology. The analysis of computational time may be subject to various factors, including but not limited to the size of the dataset, the parameters and hyperparameters of the SEINN model, and the processing capabilities of the CPU.

Table 4.9 presents an analysis of the aggregate CPU time necessary for training models utilizing two different noise levels, specifically $\sigma = 0.1$ and $\sigma = 3.0$. The tabulated data displays the CPU time in seconds in conjunction with various error metrics, such as root mean squared error (RMSE), mean absolute percentage error (MAPE), explained variance (EV), and relative error (REL). The analysis of the table indicates that the CPU time for model training is influenced by the level of noise that is present. In addition, it can be observed that the computational time for models exhibiting a lower noise level of $\sigma = 0.1$ is comparatively lower than that of models with a higher noise level of $\sigma = 3.0$. The expected result is due to the fact that models with lower noise levels tend to converge more

quickly during the training phase, which in turn requires fewer computational resources. Additionally, the error metric values displayed in the table demonstrate fluctuations based on the regularization parameter ε and the level of noise. The assessment criteria of RMSE, MAPE, EV, and REL provide significant insights into the accuracy and effectiveness of the trained models. Lower values of RMSE and MAPE are indicative of higher accuracy and precision. Conversely, higher values of EV and lower values of REL suggest better model performance and alignment with the data.

Table 4.9: Total CPU time for training SEINN for different value of ε for two noise levels $(0.1, 3.0)$

CPU Time Analysis of Regularization parameter ε						
ε	Noise σ	CPU(seconds)	RMSE	MAPE	EV	REL
$1e-1$	0.1	122496	5094	0.00676	0.9993	0.00687
$1e-1$	3.0	122931	2456	0.00326	0.9995	0.00185
$1e-3$	0.1	117525	2893	0.0038	0.9993	0.00278
$1e-3$	3.0	123270	1997	0.00265	0.9997	0.00135
$1e-5$	0.1	115318	2387	0.00317	0.9994	0.00168
$1e-5$	3.0	120730	3654	0.00486	0.9992	0.00343

Figure 5.10 illustrates the graphical depiction of the aggregate CPU time needed to train the SEINN model under various regularization parameters ε and two distinct noise levels $\sigma = 0.1$ and $\sigma = 3.0$ while maintaining a constant number of training epochs at 30000. The graphical representation illustrates that the model's computational demands are significantly higher for a noise level of $\sigma = 3.0$ compared to $\sigma = 0.1$, for all the assessed values of ε. The present phenomenon is expected to occur due to heightened levels of noise, which introduce greater complexity and variability into the training process, resulting in longer computation times. The visual depiction provides significant insights into the impact of the aforementioned factors on the computational resources needed for

model training. The aforementioned data holds significance in understanding the interplay among regularization, noise levels, and training duration. This enables researchers to make informed decisions when choosing appropriate parameters for effectively and accurately training the SEINN model.

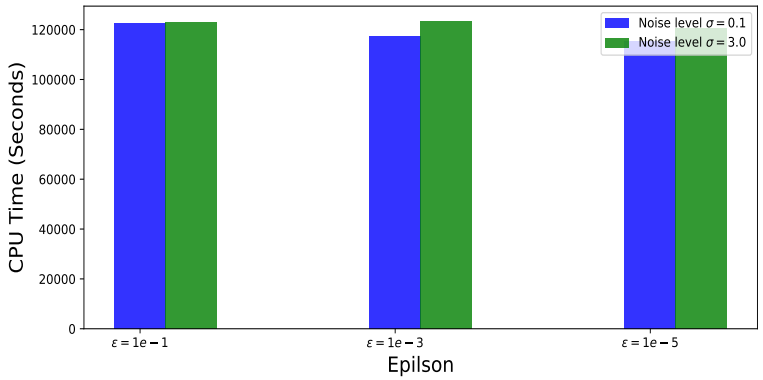

Figure 4.15: Graphical representation of total CPU time in training SEINN for different epsilon values using two noise levels $\sigma = 0.1, 3.0$ assuming 30000 epochs.

4.6 Discussion

This study aims to investigate the impact of environmental noise on transmission and non-linear incidence rates. The phrase "environmental noise" comprises a range of extraneous variables, such as human mobility, societal factors, and demographic characteristics, which may exert an influence on the spread of illnesses. The primary objective of the study's first phase was to establish the superiority of stochastic models over deterministic models. The results suggest that in a range of data-driven simulations with different vaccination rates, stochastic models demonstrate a proclivity towards reduced values of the fundamental reproduction number. The results indicate that the stochastic models demonstrate better performance in comparison to the deterministic models, as demonstrated by the root mean squared error values of both models.

The importance of investigating models that incorporate nonlinear incidence rates in the context of a vaccination protocol was considered. Acknowledging the significance of adhering to a vaccination schedule and the individual reaction to measures designed to

mitigate the transmission of the COVID-19 virus is of paramount significance. The successful execution of a proficient vaccination regimen is of paramount significance in the administration of the present pandemic and in the abatement of its communal ramifications. Immunizations are a significantly efficacious strategy for mitigating the spread of contagious illnesses, diminishing the intensity of symptoms, and decreasing the probability of hospitalization and mortality. In order to attain efficacious management of the pandemic, it is imperative to undertake measures that extend beyond the mere implementation of a vaccination regimen.

The study showcases the effectiveness of the model in precisely forecasting non-linear incidence rates across diverse parameter combinations and noise levels, using simulations based on empirical data. The aforementioned statement offers significant perspectives on the dynamics of disease transmission and the impact of vaccination rates. The findings of the study indicate that the model is efficacious in precisely forecasting non-linear incidence rates across various parameter combinations and noise levels. The results mentioned above offer significant insights into the mechanisms underlying the spread of diseases and the influence of immunization coverage.

The findings derived from the computational examination of SEINN demonstrate that the approach exhibits traits of resilience and efficacy. By manipulating various components of the approach, a more profound comprehension of the methods to reduce overfitting is obtained. The effective mitigation of overfitting can be achieved through the optimization of hyperparameters and regularization parameters. To mitigate performance fluctuations in a model, it is imperative to exercise prudent judgment during experimentation to determine the appropriate degree to which the regularization parameter should be reduced. The examination of computational time within the context of SEINN yields significant findings regarding the quantity of CPU time necessary to train the SEINN model across various regularization parameters. The aforementioned data may aid in the identification of hardware and software optimization strategies to facilitate efficient model training. A comprehensive

understanding of the influence of regularization parameters and neuron values on the total CPU time required for model training can enable researchers to make judicious decisions regarding the computational resources necessary for training their deep-learning models.

The current investigation centers on the research methodology employed to analyze the nonlinear incidence rate in relation to environmental noise. The current investigation is subject to certain constraints. It is advisable to enhance the model by incorporating social determinants of health, such as poverty and employment, among others.

4.7 Conclusion

The present study has devised a neural network named Stochastic Epidemiology-Informed Neural Network (SEINN), which can acquire knowledge regarding the epidemiological parameters and dynamics of a stochastic model. A comparative study assessed the differences between deterministic and stochastic epidemic models. The results of our analysis, which included the computation of the root mean squared error (RMSE) and primary reproduction number R_0 values, indicate that the output generated by the SEINN model exhibits superior performance compared to that of a deterministic model. It was also demonstrated that stochastic models are favored over deterministic models due to their ability to account for environmental noise resulting from randomness in both the model and the data. The significance of nonlinear incidence rates was evaluated by analyzing various parameter combinations. Our pair of algorithms exemplify the efficacy and resilience of acquiring nonlinear incidence rates in the presence of varying degrees of noise. Our proposed method's efficiency is supported by the computational analysis of sensitivity, overfitting, and CPU time.

5

BEHAVIORAL EPIDEMIC MODEL

This chapter presents a novel data-driven deep learning model, the behavior epidemiology-informed neural network (BEINN), designed to explicitly learn a time-dependent compliance rate and identify vital epidemiological parameters. The BEINN model integrates an attention mechanism and regularization parameter to enhance the accuracy of the training process and embeds behavioral constraints into the loss function. The utilization of data-driven simulations and error metrics has been shown to effectively predict the transmission of COVID-19 and identify the crucial epidemiological and behavioral factors that contribute to its spread. The present study conducted a computational analysis of BEINN concerning sensitivity, overfitting, and computational time analysis. The findings suggest that the proposed method is both practical and robust. The model under consideration has the potential to aid policymakers and public health officials in developing more effective mitigation strategies by considering the variability and diversity of human conduct in the transmission of contagious illnesses. Finally, the present study applies and analyzes the proposed BEINN model to spread COVID-19 in Tennessee, New York, and Michigan. A comparison between BEINN and PINN shows the former's superiority over the latter in terms of error metrics and central processing unit (CPU).

5.1 Behavioural SIRD Model

The basic SIR model can include many compartments to increase its complexity. For example, if we define a fourth compartment, such as Death, we have a SIRD model. However, we can use a set of ordinary differential equations to describe the dynamics of these cate-

gories over time.

$S(t)$: The individuals that are susceptible per time

$I(t)$: The individuals that get infected per time

$R(t)$: The individuals that get recovered per time

$D(t)$: The individuals that are dead per time

N : The total population per time

$$\begin{aligned}
\frac{dS(t)}{dt} &= -\beta(t)\frac{S(t)I(t)}{N} \\
\frac{dI(t)}{dt} &= \beta(t)\frac{S(t)I(t)}{N} - \gamma(t)I(t) - \delta(t)I(t) \\
\frac{dR(t)}{dt} &= \gamma(t)I(t) \\
\frac{dD(t)}{dt} &= \delta(t)I(t)
\end{aligned} \tag{5.1}$$

where the variables $\beta(t)$, $\gamma(t)$, and $\delta(t)$ represent the transmission rate, recovery rate, and death rate, respectively, and are subject to change over time. The model (5.1) has been modified to include a behavioral parameter $C(t)$ that quantifies the extent to which individuals adhere to mitigation policies. The investigation pertains to the influence of the function $C(t)$ on the rate of transmission denoted by $\beta(t)$ and the resultant effective reproduction number. The revised version of the model is now available.

$$\begin{aligned}
\frac{dS(t)}{dt} &= -\beta(t)(1-C(t))\frac{S(t)I(t)}{N} \\
\frac{dI(t)}{dt} &= \beta(t)(1-C(t))\frac{S(t)I(t)}{N} - \gamma(t)I(t) - \delta(t)I(t) \\
\frac{dR(t)}{dt} &= \gamma(t)I(t) \\
\frac{dD(t)}{dt} &= \delta(t)I(t).
\end{aligned} \tag{5.2}$$

The authors have previously demonstrated in their work [4] that the behavioral parameter $C(t)$ represents the likelihood of selecting a social distance. The function $C(t)$ is defined

on the interval from 0 to 1, where a value of 0 represents an infrequent occurrence of total non-adherence to mitigation measures, while a value of 1 indicates full adherence. The authors presented an exponential form of the behavioral parameter.

$$C(t) = \frac{1}{1 + e^{-\alpha x(t)}}, \tag{5.3}$$

where α measures the intensity of choice and $x(t)$ represents the observable factors that affect human behavior [49, 50]. They considered daily confirmed cases per capita as one of their observable factors. Our work does not rely on this formulation about the compliance rate but allows the deep learning algorithm to learn the form of the compliance rate directly from data. From the model (5.2), we derive the effective reproduction number as

$$R_{eff} = \frac{\beta(t)(1 - C(t))}{\gamma(t) + \delta(t)}. \tag{5.4}$$

Similarly, the updated transmission rate is

$$\beta(t)^* = \beta(t)(1 - C(t)). \tag{5.5}$$

Equation (5.4) captures the effect of compliance rate on reproduction number while equation (5.5) shows the effect of compliance rate on the transmission rate.

5.2 Behavioral Epidemiology Informed Neural Network

Behavioral Epidemiology Informed Neural Network (BEINN), inspired by Physics Informed Neural Network (PINN), is a novel approach to modeling the transmission of infectious diseases that considers human behavior, such as compliance with public health measures such as donning masks, maintaining social distancing, and receiving vaccinations. BEINN aims to understand the transmission, recovery, death, and compliance rates during the COVID-19 pandemic. This method incorporates deep learning techniques and

epidemiological models to predict and analyze disease progression while considering human behavior accurately. BEINN uses the susceptible-infected-recovered-deceased (SIRD) model, a system of ordinary differential equations (ODEs) that shows the population dynamics of an infectious disease.

In the BEINN method 5, we build a deep neural network for every time-dependent parameter and compartment, such as the transmission rate $\beta(t)$, the recovery rate $\gamma(t)$, the delta rate $\delta(t)$ and the compliance rate $C(t)$. The input data includes the regularization parameter ε and the actual COVID-19 data. We employ the Cubic Spline interpolation for additional points to enhance smooth training and effective learning of the dynamics of the SIRD model. We calculate the residual loss from the encoded system of ODEs in model (5.2). Data loss is the mean absolute difference between the neural network output for each compartment and the corresponding data values. The equation that computes the loss function as the sum of the data loss and the residual loss is given as

$$Loss = (1-\varepsilon)L_{data} + \varepsilon L_{res}, \tag{5.6}$$

where

$$\begin{aligned} L_{data} &= \frac{1}{N}\sum_{i=1}^{N} |Y_i - \hat{Y}i| \\ &= \frac{1}{N}\sum_{i=1}^{N} |S_i - \hat{S}_i| + |I_i - \hat{I}_i| + |R_i - \hat{R}_i| + |D_i - \hat{D}_i| \\ L_{res} &= \sum_{j=1}^{5} |e_j|, j = 1, ..., 5 \end{aligned}$$

$$
\begin{aligned}
e_1 &:= \frac{dS(t)}{dt} + \beta(t)(1-C(t))\frac{S(t)I(t)}{N} \\
e_2 &:= \frac{dI(t)}{dt} - \beta(t)(1-C(t))\frac{S(t)I(t)}{N} + \gamma(t)I(t) + \delta(t)I(t) \\
e_3 &:= \frac{dR(t)}{dt} - \gamma(t)I(t) \\
e_4 &:= \frac{dD(t)}{dt} - \delta(t)I(t) \\
e_5 &:= 1.0 - (S(t) + I(t) + R(t) + D(t))
\end{aligned}
$$

where Y_i is the actual value at time i, $\hat{Y}_i$ is the predicted value at time i, N is the number of time steps, and L_{data} and L_{res} are the data loss and the residual loss, respectively.

After calculating the loss function, we train the neural network to minimize the loss using an optimization algorithm, such as the Adam optimizer. During training, the neural network acquires knowledge of time-varying parameters and compartments, including transmission, recovery, death, and compliance rates. In addition, the trained neural network also determines the SIRD model's time-step solutions. Finally, it allows us to evaluate how compliance influences the spread of COVID-19 and the effect of public health measures on the progression of the disease.

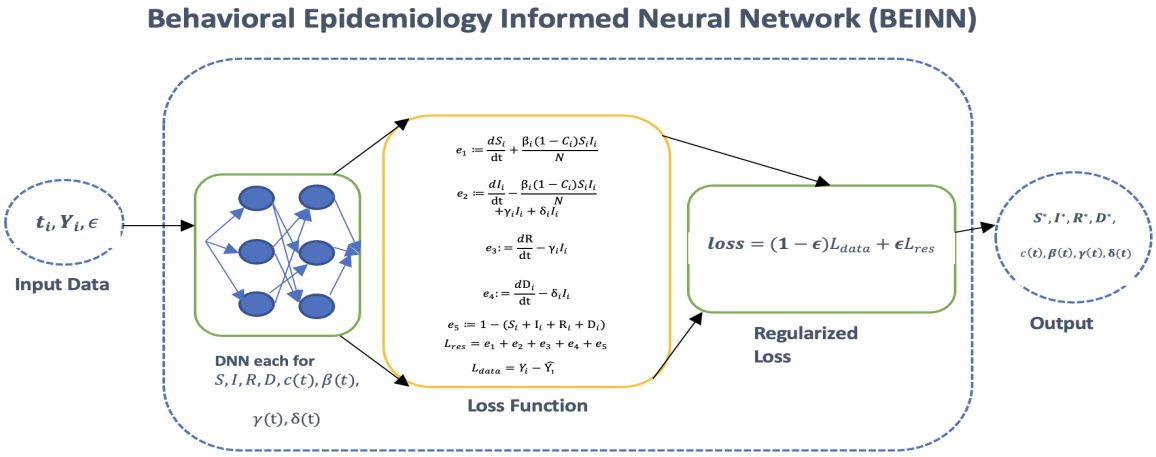

Figure 5.1: Schematic diagram of the behavioral epidemiology-informed neural network (BEINN).

The Behavioral Epidemiology-Informed Neural Network Algorithm (BEINN) Algorithm (Algorithm 5) to learn the compliance rate is presented below.

Algorithm 5 Behavioral Epidemiology Informed Neural Network (BEINN)

1: **Input:**

- Regularization parameter ε
- Actual COVID-19 data $S(t), I(t), R(t), D(t)$
- Cubic Spline interpolation for additional data points

2: **Output:**

- Epidemiology parameters $\beta(t), \gamma(t), \delta(t)$
- Behavioral parameter $C(t)$
- Compartments $S^(t), I^(t), R^(t), D^(t)$

3: **Steps:**

- Construct a neural network for each parameter and compartment
- Apply the attention mechanism to the COVID-19 data
- Compute the residuals for data and epidemiological constraints
- Calculate the loss function:

$$Loss = (1-\varepsilon)L_{data} + \varepsilon L_{res}$$

- Train the neural network using a predefined number of epochs
- Return the learned parameters and compartments

In Algorithm 5, we employ various ε values, 40000 epochs, a learning rate of 1e-3, and an Adam optimizer to train the neural network to learn different compliance rates ($C(t)$) across various COVID-19 timelines. We use cubic spline interpolation to generate additional data points. The susceptible, infected, recovered and death compartments are learned along with epidemiological parameters ($C(t), \beta(t), \gamma(t), \delta(t)$).

5.3 Data Driven Simulations

This study utilized data-driven simulations to examine the impact of compliance on diverse factors linked to the transmission of COVID-19. These variables encompass the transmission rate ($\beta(t)$), recovery rate ($\gamma(t)$), death rate ($\delta(t)$), and effective reproduction number ($R_{eff}(t)$). This research employed COVID-19 data sourced from the Tennessee Health Department [45], alongside data from New York and Michigan, which was accessed from a GitHub repository available here. The temporal scope of the dataset employed in this study spans from March 30, 2020, to December 16, 2020. Cumulative data about the number of individuals infected recovered, and deceased were extracted and processed. The susceptible data for each State was obtained by subtracting the number of infected, recovered, and deceased individuals from the total population, which is a known quantity. Our study aimed to examine the complexity and variability in human conduct regarding adherence in three states, utilizing Physics-Informed Neural Networks (PINN) and Behavior-Informed Neural Networks (BEINN). Furthermore, our objective was to evaluate the comparative advantage of BEINN in contrast to PINN in our analyses. In addition, the present study enhances comprehension of the pandemic's dynamics and furnishes significant insights for policymakers and public health officials to devise efficacious measures for curtailing the transmission of COVID-19.

The COVID-19 data for Tennessee, New York, and Michigan are depicted in Figure 5.2. The figure displays cumulative figures of infected individuals, recovered individuals, and deaths for each state. The figure displays y-values in decimal form, which denote the proportion of each compartment (infected, recovered, or deceased) relative to the total population of the corresponding state. The normalization process facilitates an improved understanding of the influence of COVID-19 in each state about its population size.

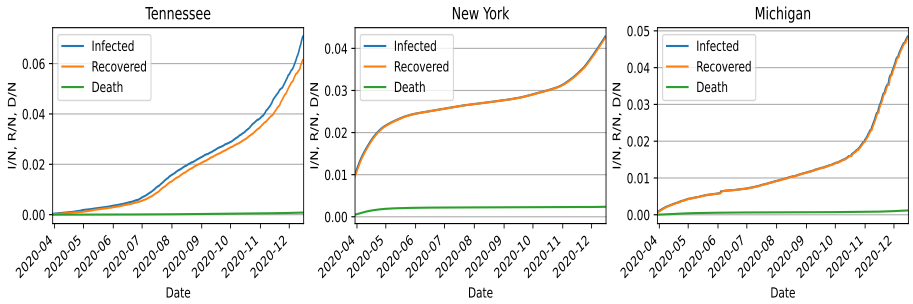

Figure 5.2: Actual COVID-19 data for Tennessee, New York, and Michigan spanning March 30, 2020, to December 16, 2020.

The compliance rates obtained by the Physics-Informed Neural Networks (PINN) and Behavior-Informed Neural Networks (BEINN) for the states of Tennessee, New York, and Michigan are depicted in Figure 5.3. The presented figure demonstrates that the PINN and BEINN algorithms exhibited disparate compliance rates, which is a predictable outcome given the distinct configurations of these two methods. In addition, diverse methodologies are utilized by each algorithm to model and acquire knowledge from the data, leading to discrepancies in the learned compliance rates. Upon analyzing the compliance rates in Tennessee, it is evident that BEINN exhibits a greater degree of compliance with mitigation measures during the initial stages of the COVID-19 outbreak. In contrast, the Physics-Informed Neural Network (PINN) proposes an alternative result. In real-time scenarios, it is plausible for both outcomes to occur, as the compliance behavior of individuals may exhibit significant variability. Therefore, to ascertain the superior accuracy of compliance rate estimations, it is imperative to consider the error metrics linked to the predictions of each algorithm. About New York, the observed compliance rates utilizing PINN and BEINN exhibited similar patterns. The observed alignment suggests that both algorithms have successfully identified comparable compliance patterns among the populace of New York. The consistency observed in the state's compliance rate estimation enhances the confidence level in the results. The compliance rates in Michigan exhibited a parallel trajectory to that of Tennessee, suggesting a similar compliance behavior pattern between the two states. The observed resemblance could be attributed to comparable socio-demographic factors,

cultural influences, or policy measures implemented within these regions.

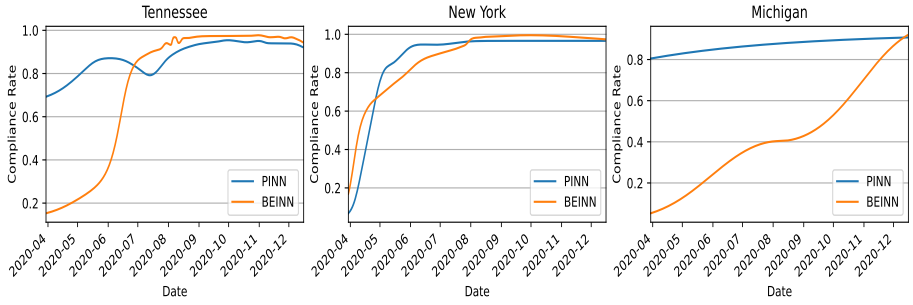

Figure 5.3: Compliance rate for Tennessee, New York, and Michigan using BEINN and PINN.

The impact of compliance rates on transmission rates was analyzed in Figure 5.4 using the Behavior-Informed Neural Networks (BEINN) for Tennessee, New York, and Michigan. Upon analysis of the data about Tennessee, it is evident that the adherence rate has an upward trajectory, as indicated by the ascending numerical values along the horizontal axis. Concurrently, a reduction in the transmission rate was observed, denoted by the hue of blue. The present discovery implies that during the initial stages of the pandemic, adherence to mitigation strategies, such as maintaining physical distance and utilizing mask-wearing, is anticipated to reduce the transmission rate. The association above highlights the significance of adherence to guidelines and regulations in mitigating the transmission of the COVID-19 virus. Similar findings can be discerned in New York and Michigan. The results indicate an inverse relationship between compliance rate and transmission rate, whereby an increase in compliance rate is associated with a decrease in transmission rate. This suggests that an increase in individual compliance results in a reduction in the transmission rate of the virus within the population.

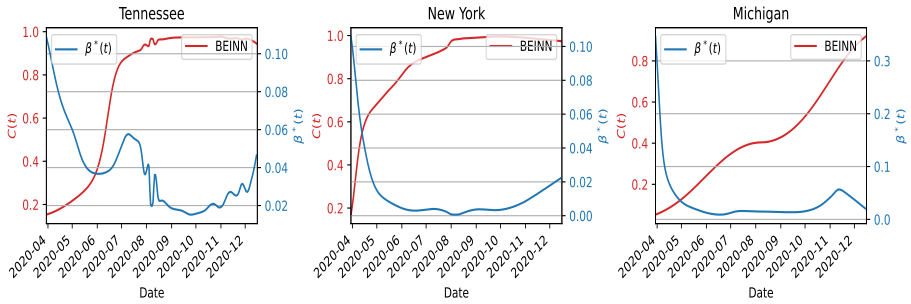

Figure 5.4: Effects of compliance rate on transmission rates for Tennessee, New York, and Michigan using BEINN

The impact of compliance rates on adequate reproduction numbers for Tennessee, New York, and Michigan is illustrated in Figure 5.5. The parameter is the adequate reproduction number, or R_{eff}, which measures the average number of new infections that arise from each infected individual. Upon analysis of the data about Tennessee, it can be observed that a correlation exists between the compliance rate and the effective reproduction number, wherein an increase in the former leads to volatility in the latter. This suggests that adherence to mitigation measures by a larger population is associated with variations in the rate of viral transmission. Notably, within the timeframe under consideration, the propagation of the virus persisted in Tennessee. Hence, it is anticipated that variations in effective reproduction numbers will manifest, indicating the dynamic character of the pandemic. Regarding New York, the situation presents a distinct variation. At the outset, heightened adherence to mitigation measures leads to a decline in effective reproduction numbers, signifying a decrease in virus transmission. Nevertheless, there is an increase in the effective reproduction number during a subsequent phase, even with sustained adherence. The observed growth in the phenomenon above may be attributed to various other factors, including but not limited to the co-occurrence of multiple medical conditions, a more significant proportion of individuals in the senior age range, or pre-existing medical conditions prevalent within the populace. In addition, various factors may contribute to heightened vulnerability and an elevated risk of transmission, resulting in a transient upsurge in the effective reproduction number. The state of Michigan presents a unique nar-

rative regarding the correlation between compliance and effective reproduction numbers. The graphical representation illustrates a decrease in compliance with mitigation measures among the populace of Michigan, leading to a consequent increase in effective reproduction numbers. This implies that failure to adhere to guidelines and regulations may lead to a higher viral spread among the populace.

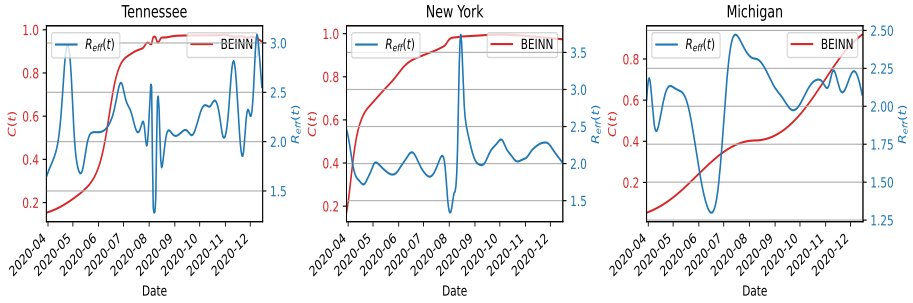

Figure 5.5: Effects of compliance rate on effect reproduction numbers for Tennessee, New York, and Michigan using BEINN

Figure 5.6 demonstrates the satisfactory alignment attained by the Physics-Informed Neural Networks (PINN) and Behavior-Informed Neural Networks (BEINN) models with the COVID-19-infected data for every state. The presented figure illustrates the efficacy of both approaches in capturing the discernible patterns and trends evident in the dataset. It is noteworthy that BEINN performs more in error metrics than PINN. Furthermore, the enhanced proficiency shown by BEINN implies that it offers a more precise depiction of the fundamental dynamics and intricacies associated with disseminating COVID-19 within the specified states. The PINN and BEINN models have demonstrated efficacy in fitting the COVID-19-infected data for Tennessee, indicating a good correspondence between the predicted and actual data points. However, a more conspicuous differentiation between the two models emerges upon examination of the data for New York and Michigan. In certain regions, BEINN performs superior to PINN in capturing the complexities of COVID-19 transmission dynamics, leading to a more accurate representation of the observed data. This implies that BEINN exhibits excellent proficiency in simulating the intricate dynamics and complexities of the pandemic within these geographical areas.

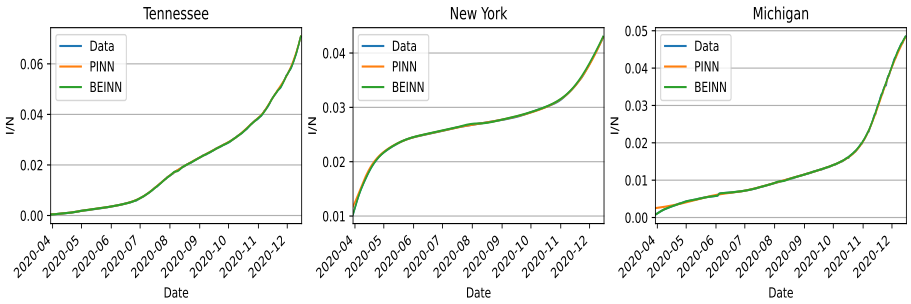

Figure 5.6: Data fitting for Tennessee, New York, and Michigan

Table 5.1 presents error metrics and CPU time values that reinforce the superior performance of the Behavior-Informed Neural Networks (BEINN) in comparison to the Physics-Informed Neural Networks (PINN) for COVID-19 data modeling. The metrics above serve to evaluate the efficacy of the models and their computational proficiency. The table displays various error metrics, including root mean square error (RMSE), mean absolute percentage error (MAPE), relative error (REL), and explained variance (EV), which provide valuable information regarding the precision of the model's forecasts. Reduced RMSE, MAPE, and REL values indicate improved alignment between anticipated and factual information, denoting greater accuracy. In addition, the metric serves as a means of quantifying the extent to which the model can account for the variance in the data. Upon examination of the error metrics presented in Table 5.1, it is evident that the root mean square error (RMSE) values associated with the (BEINN) are notably smaller than those of the physics-informed neural network (PINN) across all three states. The previous observation suggests that BEINN attains a greater degree of concordance between the projected and factual data points, culminating in enhanced precision in modeling the propagation of COVID-19.

Table 5.1: Comparing BEINN with PINN

State	Model	CPU Time (Mins)	RMSE	MAPE	EV	REL
Tennessee	**BEINN**	**43**	**616**	**0.0033**	**0.9999**	**0.1000**
	PINN	45	1129	0.0061	0.9999	0.19402
New York	**BEINN**	**42**	**836**	**0.0035**	**0.9999**	**0.00279**
	PINN	43	1699	0.00728	0.9989	0.0674
Michigan	**BEINN**	**41**	**1108**	**0.0064**	**0.9999**	**0.0110**
	PINN	42	3071	0.0177	0.9992	13.69

Figure 5.7 depicts the CPU time values showing the duration, measured in minutes, necessary to train the models. The values offer valuable insights into the models' computational efficiency. For example, reduced CPU time values signify enhanced processing speed and computational efficiency, critical for conducting extensive simulations or executing applications that necessitate real-time processing. However, comparing the CPU time values shows that BEINN models typically have a shorter training duration than PINN models.

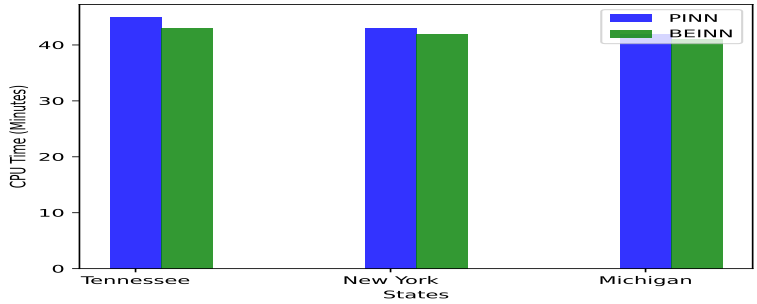

Figure 5.7: Computational Time Analysis for Tennessee, New York, and Michigan using BEINN

Figure 5.8 shows a graphical representation of RMSE and MAPE that provide additional evidence to substantiate the superiority of BEINN over PINN. Reduced values of RMSE and MAPE indicate more substantial alignment between the anticipated and factual

data points, which denotes an elevated level of precision. The values above corroborate that BEINN exhibits superior accuracy in COVID-19 data modeling compared to PINN.

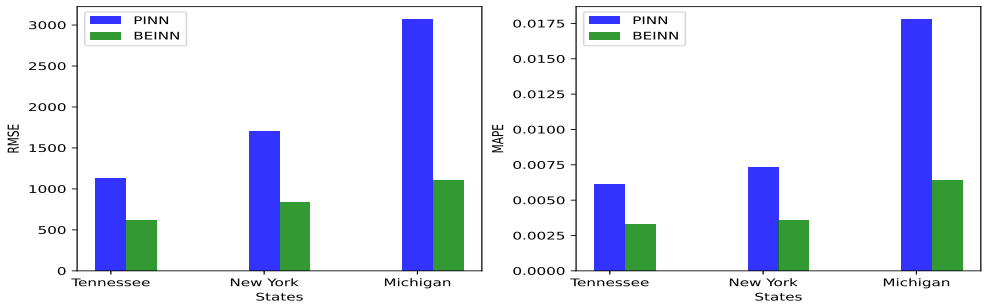

Figure 5.8: RMSE and MAPE for Tennessee, New York, and Michigan comparing BEINN with PINN

5.4 Computational Analysis of BEINN

The section explains the effects of perturbation in the input data on the outcome of the proposed method. We conduct sensitivity analysis by varying the regularization parameter values ε. Next, we examine the issue of overfitting. We want to know how well the proposed method prevents overfitting. We determine the computational time for training different values of the regularization parameter ε.

5.4.1 Sensitivity Analysis

Sensitivity analysis is a method employed to examine the impact of changes in the inputs or parameters of a model on the model's output or result. The aforementioned is a mechanism utilized to comprehend the actions of a model and evaluate its validity. In this work, we vary some parameters in the BEINN and observe the effects on the error metrics.

The impact of changing the number of neurons and layers in a Behavior Epidemiology Informed Neural Network (BEINN) model while keeping the regularization parameter fixed at $\varepsilon = 1e-1$ is presented in Table 5.2. The tabular data showcases the values of root mean squared error (RMSE), mean absolute percentage error (MAPE), explained variance (EV), and relative error (REL) for each combination of layers and neurons. By comparing the outcome from the model with 32 neurons with the model with 64 neurons concerning a

specified number of layers, we observe the model with 64 neurons outperforms the former. In particular, the sixth row of the table displays the best-performing outcomes obtained from a model consisting of 5 layers and 64 neurons. The RMSE is calculated to be 1201, the MAPE is determined to be 0.00646, and the (REL) is computed to be 0.0906.

Regularization parameter $\varepsilon = 1e-1$					
Layers	Neurons	RMSE	MAPE	EV	REL
3	32	1612	0.00866	0.9998	0.543
3	64	1592	0.00855	0.9999	0.288
4	32	1393	0.00748	0.9999	1.23
4	64	1567	0.00842	0.9998	0.101
5	32	1644	0.00884	0.9998	0.332
5	**64**	**1201**	**0.00646**	0.9998	**0.0906**

Table 5.2: Analysis of the effect of $\varepsilon = 1e-1$ on BEINN using error metrics for different layers $(3,4,5)$ and neurons $(32,64)$.

Table 5.3 illustrates the impact of reducing the value of the regularization parameter from $\varepsilon = 1e-1$ to $\varepsilon = 1e-3$ on the performance of BEINN. It indicates that overfitting is prevented when the regularization parameter is reduced from $\varepsilon = 1e-1$ to $\varepsilon = 1e-3$. We achieve better error values in the fourth row by carefully selecting the best combination of hyperparameters of layers and neurons. This row consists of a four-layered model of 64 neurons. The error metric values, such as RMSE and REL, are computed as 712 and 0.0270, respectively.

Regularization parameter $\varepsilon = 1e-3$					
Layers	Neurons	RMSE	MAPE	EV	REL
3	32	2842	0.00153	0.9995	34
3	64	1473	0.00792	0.9999	0.0450
4	32	1277	0.00686	0.9999	0.397
4	**64**	**712**	0.00383	0.9999	**0.0270**
5	32	726	0.00390	0.9999	0.109
5	64	1482	0.00796	0.9992	0.0840

Table 5.3: Analysis of the effect of $\varepsilon = 1e-3$ on BEINN using error metrics for different layers $(3,4,5)$ and neurons $(32,64)$.

According to Table 5.4, it can be observed that decreasing the regularization parameter to $\varepsilon = 1e-7$ may result in overfitting and inferior performance in the BEINN model. On the other hand, adjusting the number of neurons or incorporating an additional layer can compensate for the decline in regularization and potentially enhance the overall performance. Furthermore, the findings indicate that achieving optimal performance in the BEINN model requires a careful balance between the regularization parameter and the number of neurons and layers. The fourth row displays the best combination of the number of layers (4) and neurons (64) to yield the optimal error metric values.

Regularization parameter $\varepsilon = 1e-7$					
Layers	Neurons	RMSE	MAPE	EV	REL
3	32	2268	0.001219	0.9997	0.3177
3	64	1313	0.00705	0.9999	0.355
4	32	1128	0.006058	0.9999	1.0562
4	**64**	**740**	0.00403	0.9999	**0.0567**
5	32	977	0.0052	0.9999	0.9793
5	64	862	0.00433	0.99998	0.1590

Table 5.4: Analysis of the effect of $\varepsilon = 1e-7$ on BEINN using error metrics for different layers $(3,4,5)$ and neurons $(32,64)$.

The impact of the regularization parameter $\varepsilon = 1e-1$ on the Behavior Epidemiology Informed Neural Network (BEINN) model is shown in Figure 5.9. The model's performance is evaluated using the root mean squared error (RMSE) and mean absolute percentage error (MAPE) as performance metrics. The RMSE graph indicates that the optimal configuration of layers and neurons is achieved with five layers and 64 neurons, resulting in a value of 1201. Similarly, a sequence of 4 layers comprising 32 neurons each was observed, yielding a value of 1393. The combination that shows the poorest performance consists of five layers with 32 neurons, producing a value of 1644. Concerning the MAPE graph, it can be observed that the combination that presents the highest performance comprises five layers and 64 neurons, exhibiting a value of 0.006446. The combination with the poorest performance consists of five layers and 32 neurons, with a recorded value of 0.05605.

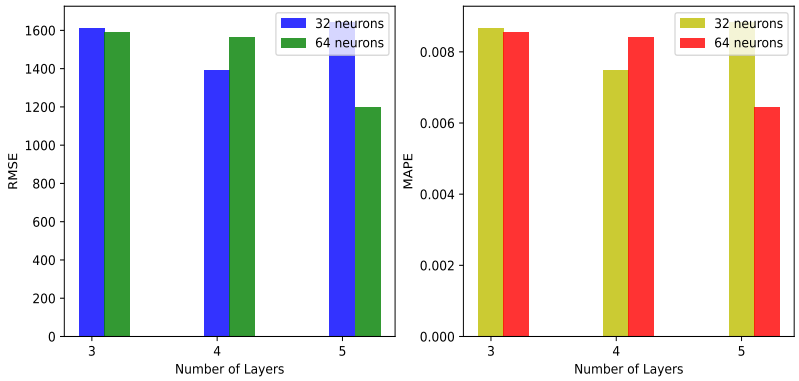

Figure 5.9: Graphical representation of effects of regularization parameter $\varepsilon = 1e-1$ using RMSE and MAPE

Figure 5.10 indicates that reducing the regularization parameter value from $\varepsilon = 1e-1$ to $\varepsilon = 1e-3$ significantly improves the BEINN model's performance. The optimal configurations that exhibit superior performance using the RMSE graph comprise four layers and 64 neurons, consistent with the results in Table 5.3. This figure underscores the importance of carefully choosing suitable hyperparameters for the BEINN model to achieve the best possible outcome. MAPE graph presents a similar performance analysis as the RMSE graph.

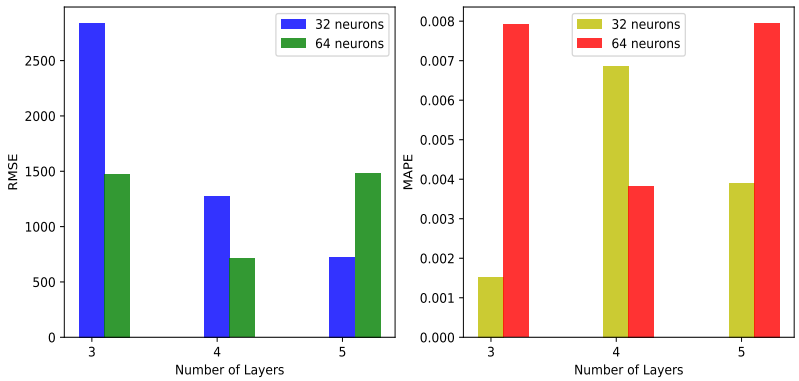

Figure 5.10: Graphical representation of effects of regularization parameter $\varepsilon = 1e-3$ using RMSE and MAPE.

Figure 5.11 shows that reducing the regularization parameter from $\varepsilon = 1e-3$ to $\varepsilon = 1e-7$ significantly improves the performance of the BEINN model. The RMSE graph displays the optimal configurations exhibiting superior performance, entailing four layers

comprising 64 neurons. This outcome is in line with the results in Table 5.4. The MAPE graph presents a similar performance analysis to the RMSE graph. Nonetheless, the improvements in this version exhibit a minor performance compared to Figure 5.11, thereby suggesting a a threshold beyond which the reduction in regularization cannot be extended without compromising performance quality. Finally, the figure emphasizes the significance of choosing suitable hyperparameters for the BEINN model to achieve optimal performance and the necessity of meticulous adjustment of regularization parameters.

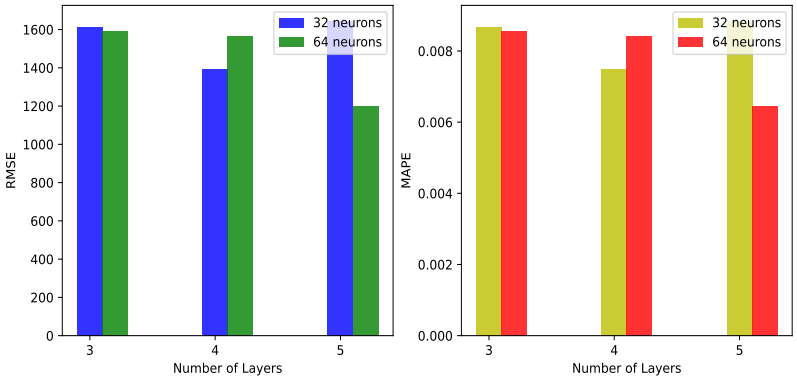

Figure 5.11: Graphical representation of effects of regularization parameter $\varepsilon = 1e-7$ using RMSE and MAPE.

5.4.2 Overfitting Analysis

Evaluating a deep learning model's performance on unseen data is called overfitting analysis. The objective is to ascertain if the model exhibits overfitting tendencies by assess its efficacy on the training dataset against its effectiveness on the validation or test dataset. The analysis of overfitting may serve as a means to direct the choice of suitable hyperparameters for a model and to evaluate the model's capacity for generalization. We assess the impact of varying the regularization parameter ε and the right choice of hyperparameters on overfitting. Finally, we divided the input data for BEINN into 80% training set and 20% validation set. Finally, we set the number of layers at 4 and the number of epochs at 40000 and 100000.

The effect of the regularization parameter $\varepsilon = 1e-1$ on mitigating overfitting in the

BEINN model is visually depicted in Figure 5.12. The right graph compares training and validation losses using 40000 epochs and 4 layers. By taking the log of the number of epochs, we observe that the training and validation losses are initially different but become similar after some epochs. The left graph, on the other hand, is obtained using $100,000$ epochs and 4 layers. It can be seen that both the training and validation losses are the same.

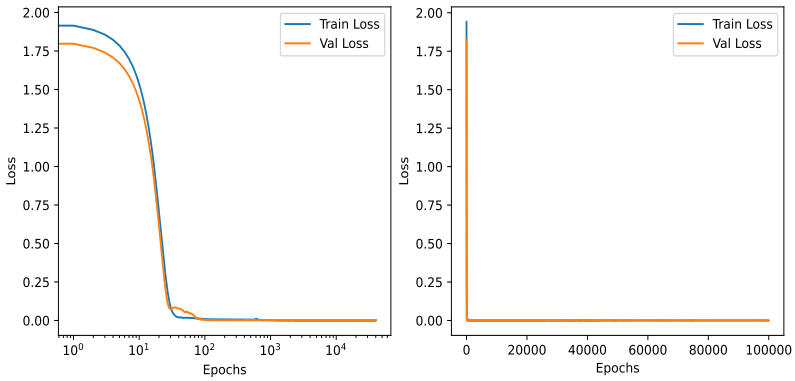

Figure 5.12: Graphical representation of effects of regularization parameter $\varepsilon = 1e-1$ preventing overfitting using epochs 40000 and 10000.

Figure 5.13 illustrates the impact of the regularization parameter $\varepsilon = 1e-3$ in reducing overfitting in the BEINN model. The graph on the proper compares the training and validation losses with the implementation of 40000 epochs and 4 layers. Upon logarithmic transformation of the epoch count, it is evident that the training and validation losses exhibit dissimilarity at the outset but converge to a similar value after a certain number of epochs. In contrast, the graph on the left was generated using 100000 epochs and implementing four layers. The equality of the training and validation losses is evident.

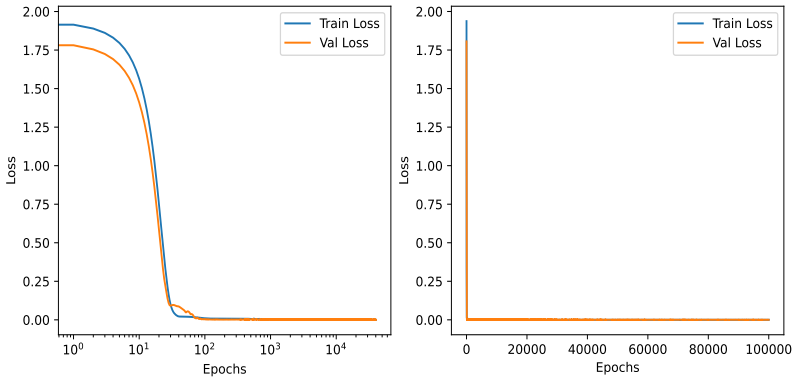

Figure 5.13: Graphical representation of effects of regularization parameter $\varepsilon = 1e-3$ preventing overfitting using epochs 40000 and 10000.

Figure 5.14 illustrates that utilizing a reduced regularization parameter value of $\varepsilon = 1e-7$ with a suitable hyperparameter choice mitigates the overfitting issue. The results from the validation set suggest that the model's ability to generalize to new data is enhanced by reducing the regularization parameter. The graph on the right-hand side compares the training and validation losses obtained through the execution of 40000 epochs and the utilization of 4 layers. After applying a logarithmic transformation to the epoch count, it becomes apparent that the training and validation losses display dissimilarity initially but eventually converge to a similar value after a specific number of epochs. In comparison, the graph depicted on the left was produced by utilizing 100000 epochs and incorporating four layers. Finally, the parity between the training and validation losses is observed.

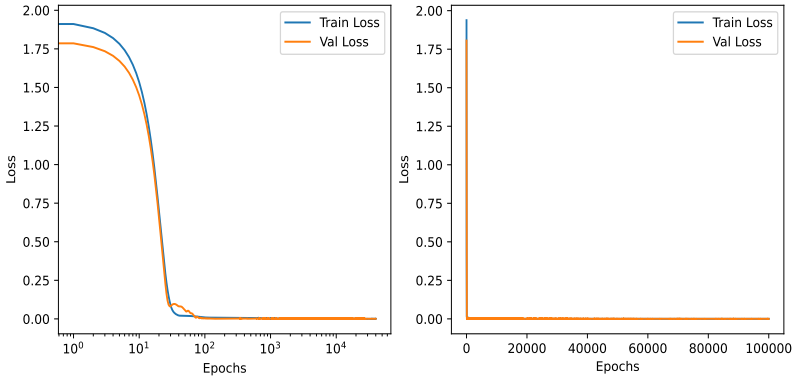

Figure 5.14: Graphical representation of effects of regularization parameter $\varepsilon = 1e-7$ preventing overfitting using epochs 40000 and 10000.

5.4.3 Computational Time Analysis

Evaluating the computational time required for training the BEINN involves measuring the duration taken by the Central Processing Unit (CPU) to train the BEINN model on a specific dataset. The significance of this analysis lies in its ability to ascertain the efficacy of the model training procedure. Several factors, including the dataset size, the BEINN model's parameters and hyperparameters, and the CPU's processing capabilities, may influence the analysis of computational time.

The results of the CPU time analysis for training (BEINN) with varying regularization parameters ε are presented in Table 5.5. The given table showcases the total CPU time necessary to train the BEINN model alongside the corresponding performance metrics, such as the root mean squared error (RMSE), mean absolute percentage error (MAPE), explained variance (EV), and relative error (REL). The values are presented for various ε values while also assuming neural units of 32 and 64. Furthermore, the table shows that the CPU time necessary to train the BEINN model is subject to fluctuations based on specific regularization parameters and neural units. As an illustration, the CPU sufficient time to prepare the model utilizing $\varepsilon = 1e-1$ is more significant than than required for $\varepsilon = 1e-2$ and $\varepsilon = 1e-3$ for both 32 and 64 neurons. Again, the table shows that the utilization of 64 neurons results in greater consumption of CPU time consumption compared to the utilization of 32 neurons across all regularization parameters. We observe the influence of regularization parameters on the performance metrics of the model. For example, reducing the regularization parameter from $\varepsilon = 1e-1$ to $\varepsilon = 1e-3$ leads to decreased RMSE and MAPE values for 32 and 64 neurons.

CPU Time Analysis of Regularization parameter ε						
ε	Neurons	CPU(seconds)	RMSE	MAPE	EV	REL
$1e-1$	32	419117	1024	0.005502	0.9999	0.0405
$1e-1$	64	711102	799	0.004295	0.9999	0.3831
$1e-2$	32	413836	1693	0.009010	0.9998	0.6443
$1e-2$	64	714602	941	0.00506	0.9996	0.0404
$1e-3$	32	424608	1292	0.00694	0.9999	0.1739
$1e-3$	64	728276	984	0.00525	0.9995	0.1022

Table 5.5: otal CPU time for training BEINN for different value of ε assuming neurons $(32,64)$.

The graphical representation of the total CPU time required to train BEINN using various regularization parameters ε, with 32 and 64 neurons and 40000 epochs, is depicted in Figure 5.15. The figure illustrates the fluctuation in the total CPU time necessary for the training of the BEINN model, as influenced by specific regularization parameters and neural units. It further shows that the utilization of 64 neurons results in a higher consumption of CPU time in comparison to the utilization of 32 neurons across all regularization parameters. Additionally, a reduction in the regularization parameter, specifically from $\varepsilon = 1e-1$ to $\varepsilon = 1e-3$, decreases overall CPU time for both 32 and 64 neurons.

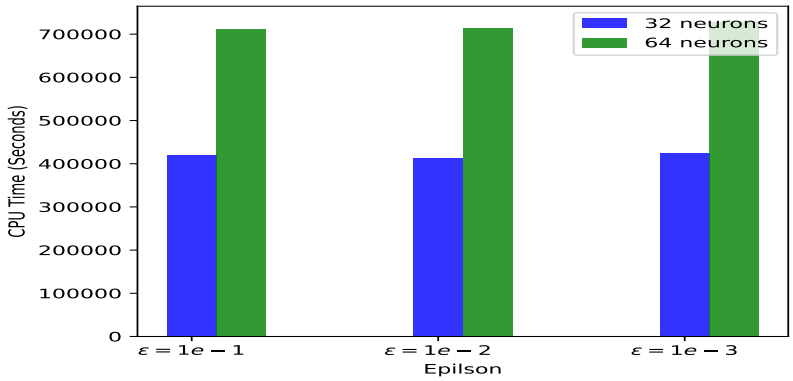

Figure 5.15: Graphical representation of total CPU time in training BEINN for different values of epsilon using 32 and 64 neurons assuming 40000 epochs.

5.5 Discussion

During the early stages of the pandemic, a lack of adherence to preventative measures and the unavailability of a vaccine contributed to a heightened transmission rate. This, in turn, led to an escalation in the effective reproduction rate, which is the impact of the compliance rate on the number of individuals who contracted the virus. The present research examines human conduct's influence on the transmission rate, adequate reproduction numbers, and model dynamics. A nonlinear rate of acquired compliance is inherently observed. This analysis focuses on the temporal sequence spanning from March 30th, 2020, to December 16th, 2020. Three states, specifically Tennessee, New York, and Michigan, were considered. The implementation of preventive measures such as hand hygiene, social distancing, mask-wearing, and remote work has become necessary due to the swift transmission of the disease across the state, as reported in recent studies [11, 4]. The data on compliance rates during this period suggests that a growing proportion of individuals in the three states complied with the mitigation measures, significantly reducing transmission rates. The study examined the comparative effectiveness of BEINN and PINN models utilizing COVID-19 data from three states.

In Tennessee, it has been observed that the rate of learned compliance resulting from two methods exhibits an inverse correlation with transmission rates while displaying a fluc-

tuating association with adequate reproduction numbers. As an increasing number of individuals adhere to mitigation measures during the initial stages of the pandemic, there is a corresponding decrease in transmission rates. The fluctuation in the $R_{eff}(t)$ can be attributed to various factors that affect its efficacy beyond mere compliance. The relationship between compliance rate and transmission rates and effective reproduction numbers exhibited contrasting outcomes for individuals who reside in New York. The decline in transmission rates was observed in conjunction with increased compliance rates; however, there was a notable surge in effective reproduction numbers. This observation demonstrates that in the initial stages of the viral outbreak, the populace of New York City experienced a significant number of confirmed cases of infection. The findings in Michigan exhibited a degree of similarity to those observed in Tennessee. The data analysis results from three states indicate that adherence to nonpharmaceutical interventions (NPIs) is crucial for achieving optimal outcomes. Nonpharmaceutical interventions (NPIs) refer to a range of measures that can be implemented to mitigate the spread of infectious diseases. These measures include but are not limited to using face masks, adherence to social distancing protocols, observance of proper hand hygiene, and avoidance of crowded indoor spaces. The aforementioned prophylactic measures function as an additional strategy to the effects of vaccination, as they aid in reducing the transmission of the virus.

We conduct a comparative analysis of BEINN and PINN by utilizing error metrics and CPU time evaluation. The performance evaluation criteria, namely Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Explained Variance (EV), demonstrate the superior ability of Bayesian Encoder-Decoder Invertible Neural Network (BEINN) in precisely capturing the compliance rate and its influence on the transmission of COVID-19, compared to Physics-Informed Neural Network (PINN). The efficacy of attention mechanisms and regularization techniques in augmenting the training process and elevating the precision of the BEINN model is evidenced by the reduced values of these error metrics. The analysis of CPU time provides additional evidence of the computational

efficiency of BEINN, underscoring its superiority over PINN in the decreased duration of the training process. The utilization of attention mechanisms and regularization techniques in BEINN models has been observed to improve the training process and computational efficiency, as evidenced by the reduced computational time required. Attention mechanisms in machine learning models facilitate the selective concentration on pertinent features and patterns within the data, resulting in enhanced learning efficacy. Regularization methods, such as weight decay or dropout, aid in mitigating overfitting and enhancing the generalization capacity of a model. Upon integration with the BEINN framework, these mechanisms diminish the computational duration required for training, thereby improving the efficiency of BEINN as a preferred option for modeling the behavioral facets of COVID-19 propagation.

The computational analysis of BEINN shows that our method is robust and effective. We gain insights into reducing overfitting by perturbing various aspects of the technique. Making the right choice of hyperparameters and regularization parameters will help decrease overfitting. Careful intuition is required through experimentation to determine the limit to which the regularization parameter is reduced to avoid fluctuations in the model's performance. The computational time analysis of BEINN offers significant insights into the CPU time necessary to train the BEINN model with varying regularization parameters. This information can aid in determining hardware and software optimization strategies for practical model training. Finally, researchers can make informed decisions regarding the computational resources needed for training their deep-learning models by comprehending the influence of regularization parameters and neuron values on the total CPU time necessary for model training.

The research methodology employed in this study is centered on a single social determinant of health. This determinant has been previously identified in the work of Galanis et al. (2021) [51]. Therefore, it is crucial to acknowledge that different social determinants of health can substantially impact the spread of infectious diseases, such as COVID-19,

and individuals' responses to public health measures. Our methodology can incorporate various social determinants of health, such as elevated unemployment rates, educational attainment, poverty, restricted access to healthcare facilities, and age-related variations in compliance with public health measures. In addition, various factors can influence an individual's susceptibility to infectious diseases, ability to comply with public health guidelines and availability of healthcare resources for implementing public health protocols.

5.6 Conclusions

This study has analyzed the impact of compliance rates on the transmission dynamics of COVID-19 by utilizing Behavior-Informed Neural Networks and Physics-Informed Neural Networks models. The study centered on Tennessee, New York, and Michigan, examining the correlation between compliance, transmission rates, and effective reproduction numbers. The results suggested that strict adherence to preventative measures, including but not limited to hand hygiene, social distancing, mask-wearing, and other mitigating measures, is imperative in curtailing the transmission of COVID-19. The data analysis findings indicated a rise in the percentage of people adhering to these measures in the three states, leading to a noteworthy decline in transmission rates. Although there were variations in the effective reproduction numbers, the relationship between compliance and transmission rates remained generally consistent across the states. Tennessee demonstrated a variable correlation between compliance rates and effective reproduction numbers, whereby an increase in compliance was associated with a decrease in transmission rates. The compliance rates in New York have shown a positive correlation with the decline in transmission rates. However, there has been a significant increase in effective reproduction numbers, which may have been influenced by various factors, including comorbidities or underlying medical conditions. For example, the state of Michigan exhibited resemblances to Tennessee, underscoring the significance of adherence to regulations for achieving optimal results. The superiority of the proposed method was confirmed through a comparative analysis of er-

ror metrics and CPU time evaluation compared to PINN models. The results indicate that BEINN demonstrated superior performance in lower root mean square error, mean absolute percentage error, and higher explained variance than PINN. Implementing attention mechanisms and regularization techniques in BEINN has enhanced the training process, increasing accuracy. Moreover, BEINN exhibited computational efficiency by reducing the duration of the training, rendering it a desirable option for modeling the behavioral facets of COVID-19 transmission. A potential avenue for future research encompasses integrating additional behavioral factors that directly influence the transmission of contagious illnesses.

6

CONCLUSION AND FUTURE WORK

We developed three deep-learning methods to analyze different COVID-19 epidemic models. Data-driven simulations for the first method, an epidemiology-informed neural network, showed that the vaccination of more people curbed the spread of the disease. In the same way, simulations based on data from a hybrid approach made up of residual and recurrent neural networks showed that the ResNet-GRU hybrid was superior. In the second method, we used a neural network based on stochastic epidemiology. To do this, we used Euler-Murayama to discretize the system of stochastic differential equations and then encode it as a loss function. Finally, based on error metrics, our findings demonstrated that stochastic models outperform their deterministic counterparts. We investigated the impact of stochasticity and vaccination on nonlinear incidence rates. To analyze the impact of social and behavioral factors on the spread of disease, we developed a third method called a behavior-epidemiology-informed neural network. This method learned an explicit compliance rate that changed over time and identified the critical epidemiological parameters of a behavioral epidemic model. Finally, the computational analysis regarding sensitivity, overfitting, and computational time showed the effectiveness and robustness of the method. Finally, we showed that the results of our work could help public health officials develop effective vaccination and mitigation strategies by considering randomness in the model and the variability of human behavior in transmitting contagious diseases. The potential future direction of our work includes the following:

- the use of graph neural networks to analyze spatial-temporal epidemic models that include the spatial comportment of human movement.
- the inclusion of more social and behavioral parameters in behavioral epidemic models to comprehensively understand the spread dynamics.

References

[1] Salah Ghamizi, Renaud Rwemalika, Maxime Cordy, Lisa Veiber, Tegawendé F Bissyandé, Mike Papadakis, Jacques Klein, and Yves Le Traon. Data-driven simulation and optimization for COVID-19 exit strategies. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3434–3442, 2020.

[2] Thomas K Torku, Abdul QM Khaliq, and Khaled M Furati. Deep-data-driven neural networks for COVID-19 vaccine efficacy. *Epidemiologia*, 2(4):564–586, 2021.

[3] P. Verger and È. Dubé. Restoring confidence in vaccines in the COVID-19 era. *Expert Review of Vaccines*, 19:991 – 993, 2020.

[4] Corrado Di Guilmi, Giorgos Galanis, and Giorgos Baskozos. A behavioural SIR model: Implications for physical distancing decisions. *Review of Behavioral Economics*, 9(1):45–63, 2022.

[5] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[6] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of London. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[7] Toheeb A Biala and AQM Khaliq. A fractional-order compartmental model for the spread of the COVID-19 pandemic. *Communications in Nonlinear Science and Numerical Simulation*, 98:105764, 2021.

[8] Jie Long, A.Q.M Khaliq, and K.M. Furati. Identification and prediction of time-varying parameters of COVID-19 model: a data-driven deep learning approach. *International Journal of Computer Mathematics*, 98:1617–1632, 2021.

[9] Xiao Ning, Xi-An Li, Yongyue Wei, and Feng Chen. Euler iteration augmented physics-informed neural networks for time-varying parameter estimation of the epidemic compartmental model. *Frontiers in Physics*, 10:1300, 2022.

[10] Xiao Ning, Linlin Jia, Yongyue Wei, Xi-An Li, and Feng Chen. Epi-dnns: Epidemiological priors informed deep neural networks for modeling COVID-19 dynamics. *Computers in Biology and Medicine*, page 106693, 2023.

[11] K. D. Olumoyin, A. Q. M. Khaliq, and K. M. Furati. Data-driven deep-learning algorithm for asymptomatic COVID-19 model with varying mitigation measures and transmission rate. *Epidemiologia*, 2(4):471–489, 2021.

[12] Ehsan Kharazmi, Min Cai, Xiaoning Zheng, Guang Lin, and George Em Karniadakis. Identifiability and predictability of integer-and fractional-order epidemiological models using physics-informed neural networks. *medRxiv*, 2021.

[13] Frederik Verelst, Lander Willem, and Philippe Beutels. Behavioural change models for infectious disease transmission: a systematic review (2010–2015). *Journal of The Royal Society Interface*, 13(125):20160820, 2016.

[14] Sebastian Funk, Marcel Salathé, and Vincent AA Jansen. Modelling the influence of human behavior on the spread of infectious diseases: a review. *Journal of the Royal Society Interface*, 7(50):1247–1256, 2010.

[15] Nurul Absar, Nazim Uddin, Mayeen Uddin Khandaker, and Habib Ullah. The efficacy of deep learning based LSTM model in forecasting the outbreak of contagious diseases. *Infectious Disease Modelling*, 7(1):170–183, 2022.

[16] Mohammad Moosazadeh, Pouya Ifaei, Amir Saman Tayerani Charmchi, Somayeh Asadi, and ChangKyoo Yoo. A machine learning-driven spatiotemporal vulnerability appraisal based on socio-economic data for COVID-19 impact prevention in the us counties. *Sustainable Cities and Society*, 83:103990, 2022.

[17] I Goodfellow, Y Bengio, and A Courville. *Deep Learning*. MIT Press, Cambridge, 2016.

[18] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560, 1990.

[19] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Accessed: 05-27-2021.

[22] Maziar Raissi, Niloofar Ramezani, and Padmanabhan Seshaiyer. On parameter estimation approaches for predicting disease transmission through optimization, deep learning and statistical inference methods. *Letters in Biomathematics*, 6(2):1–26, 2019.

[23] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[24] Kevin Linka, Amelie Schäfer, Xuhui Meng, Zongren Zou, George Em Karniadakis, and Ellen Kuhl. Bayesian physics informed neural networks for real-world nonlinear dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 402:115346, 2022.

[25] Tong Qin, Kailiang Wu, and Dongbin Xiu. Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635, 2019.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[27] Jared O'Leary, Joel A Paulson, and Ali Mesbah. Stochastic physics-informed neural networks (spinn): A moment-matching framework for learning hidden physics within stochastic differential equations. *arXiv preprint arXiv:2109.01621*, 2021.

[28] J Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[29] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[30] Sotiris B Kotsiantis, Dimitris Kanellopoulos, and Panagiotis E Pintelas. Data preprocessing for supervised leaning. *International journal of computer science*, 1(2):111–117, 2006.

[31] Main Steps in Data Preprocessing. https://www.programmersought.com/article/98093557423/. Accessed: 04-07-2022.

[32] Lin Feng. SEIR model combined with LSTM and GRU for the trend analysis of COVID-19. 2021.

[33] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[34] Root Mean Squared Error. https://www.sciencedirect.com/topics/engineering/root-mean-squared-error. Accessed: 06-24-2021.

[35] Mean Absolute Error. https://en.wikipedia.org/wiki/Mean_absolute_error. Accessed: 07-12-2021.

[36] Mean Absolute Percentage Error. https://en.wikipedia.org/wiki/Mean_absolute_percentage_error. Accessed: 07-12-2021.

[37] Coefficient of Determination. https://en.wikipedia.org/wiki/Coefficient_of_determination. Accessed: 04-07-2022.

[38] Sara B Crawford, Andrzej S Kosinski, Hung-Mo Lin, John M Williamson, and Huiman X Barnhart. Computer programs for the concordance correlation coefficient. *Computer methods and programs in biomedicine*, 88(1):62–74, 2007.

[39] Junyi Gao, Rakshith Sharma, Cheng Qian, Lucas M Glass, Jeffrey Spaeder, Justin Romberg, Jimeng Sun, and Cao Xiao. Stan: spatiotemporal attention network for pandemic prediction using real-world evidence. *Journal of the American Medical Informatics Association*, 28(4):733–743, 2021.

[40] Mean Squared Log Error. https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/mean-squared-logarithmic-error-(msle). Accessed: 04-07-2022.

[41] Root Mean Squared Log Error. https://medium.com/analytics-vidhya/root-mean-square-log-error-rmse-vs-rmlse-935c6cc1802a. Accessed: 07-12-2021.

[42] How to Configure k-Fold Cross-Validation. https://machinelearningmastery.com/how-to-configure-k-fold-cross-validation/. Accessed: 06-29-2021.

[43] Gerardo Chowell. Fitting dynamic models to epidemic outbreaks with quantified uncertainty: A primer for parameter uncertainty, identifiability, and forecasts. *Infectious Disease Modelling*, 2(3):379–398, 2017.

[44] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

[45] Tennessee Health Department. https://www.tn.gov/health/cedep/ncov/data. Accessed: 06-28-2021.

[46] Peter E Kloeden and Eckhard Platen. Stochastic differential equations. In *Numerical solution of stochastic differential equations*, pages 103–160. Springer, 1992.

[47] Yongli Cai, Yun Kang, and Weiming Wang. A stochastic sirs epidemic model with nonlinear incidence rate. *Applied Mathematics and Computation*, 305:221–240, 2017.

[48] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

[49] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[50] Daniel McFadden. Economic choices. *American economic review*, 91(3):351–378, 2001.

[51] Giorgos Galanis and Adam Hanieh. Incorporating social determinants of health into modelling of COVID-19 and other infectious diseases: a baseline socio-economic compartmental model. *Social science & medicine*, 274:113794, 2021.