



Research article

SEINN: A deep learning algorithm for the stochastic epidemic model

Thomas Torku^{1,*}, Abdul Khaliq² and Fathalla Rihan³

¹ University Studies Department, Middle Tennessee State University, Murfreesboro, TN 37132, USA

² Department of Mathematical Sciences, Middle Tennessee State University, Murfreesboro, TN 37132, USA

³ Department of Mathematical Sciences, United Arab Emirates University, P.O. Box 15551, Al Ain, Abu Dhabi, UAE

* **Correspondence:** Email: thomas.torku@mtsu.edu, ttorku01@gmail.com; Tel: +16158985904.

Abstract: Stochastic modeling predicts various outcomes from stochasticity in the data, parameters and dynamical system. Stochastic models are deemed more appropriate than deterministic models accounting in terms of essential and practical information about a system. The objective of the current investigation is to address the issue above through the development of a novel deep neural network referred to as a stochastic epidemiology-informed neural network. This network learns knowledge about the parameters and dynamics of a stochastic epidemic vaccine model. Our analysis centers on examining the nonlinear incidence rate of the model from the perspective of the combined effects of vaccination and stochasticity. Based on empirical evidence, stochastic models offer a more comprehensive understanding than deterministic models, mainly when we use error metrics. The findings of our study indicate that a decrease in randomness and an increase in vaccination rates are associated with a better prediction of nonlinear incidence rates. Adopting a nonlinear incidence rate enables a more comprehensive representation of the complexities of transmitting diseases. The computational analysis of the proposed method, focusing on sensitivity analysis and overfitting analysis, shows that the proposed method is efficient. Our research aims to guide policymakers on the effects of stochasticity in epidemic models, thereby aiding the development of effective vaccination and mitigation policies. Several case studies have been conducted on nonlinear incidence rates using data from Tennessee, USA.

Keywords: deep learning; data-driven; stochastic differential equation; regularization; stochastic epidemiology informed neural network; computational analysis; nonlinear incidence rate

1. Introduction

In 2020, the World Health Organization [1] designated the transmission of COVID-19 as a pandemic, citing the highly contagious nature of the virus. This declaration was made based on established epidemiological criteria. Following that, governmental and public health organizations implemented interventions to slow the spread of the disease. Numerous virus strains, including Delta and Omicron, have been disseminated throughout various nations. An inquiry has arisen about the underlying factors contributing to these variants and the inadequacy of current epidemic models in terms of comprehending the impact of environmental noise on the transmission of the virus [2].

A comprehensive literature review reveals that much research has been conducted on transmitting the COVID-19 virus. Scholars have investigated diverse facets of viral transmission dynamics, ranging from basic susceptible-infected-recovered (SIR) [3] models to intricate ones. Biala et al. [4], examined a deterministic model that assessed the impact of contact tracing on reducing virus transmission. Furati et al. [5] developed a fractional model with exposure to government intervention and public belief. Several researchers have investigated models featuring parameters that vary over time [6–8]. Torku et al. [9] demonstrated the influence of the vaccination program on viral transmission within the state of Tennessee, located in the United States of America. Empirical evidence has demonstrated a negative correlation between the daily vaccination rate and infectiousness, thus validating the established theoretical concept. A study by Olumoyin et al. [8] involved the development of a deterministic model asymptomatic SIR. The model determined constant and time-dependent transmission rates for several countries, including the USA, South Korea and Italy. The study conducted by Rihan et al. [10] aimed to analyze a fractional-order delayed model of COVID-19, considering the efficacy of vaccination.

SDEs (SDEs) have been extensively employed to address problems that exhibit stochasticity in their parameters or system solutions [11, 12]. The stochasticity observed in the viral spread can be attributed to extrinsic factors, such as environmental noise, encompassing fitness, temperature, geographical area, and population density. Stochastic models are deemed to possess greater informativeness, realism, and utility compared to deterministic models. Mao et al. [13] conducted a comparative analysis between stochastic and deterministic models and found that stochastic perturbations significantly influence the spread of infectious diseases. In their study, Dalal et al. [2] employed parametric methods to assess the impact of environmental noise on stochastic models. Their findings revealed that stochastic noise alters the underlying reproductive numbers. Numerical methods of the classical nature have been employed to solve SDEs, including explicit methods like the Euler-Maruyama method and implicit methods like the backward-Euler method [14].

The mathematical modeling of infectious diseases incorporates a nonlinear incidence rate, as noted in sources [15, 16]. The relationship between the number of new infections and various factors such as population density, age and behavior is often modeled using a proportionality constant and the product of the number of infected and susceptible individuals. According to [17], the incidence rate may exhibit nonlinearity in certain circumstances. This phenomenon can be attributed to super-spreaders, population heterogeneity and various interventions. Numerous scholars have formulated mathematical models to capture the nonlinearity of infectious disease incidence. Miao and colleagues proposed a stochastic SIS epidemic model featuring a nonlinear incidence rate and a double epidemic hypothesis. The details of this model can be found in their publication [14].

The field of deep learning has garnered significant attention owing to its vast potential in scientific

domains, including but not limited to autonomous vehicles, object identification and image classification, among others [18]. The literature [19,20] demonstrates that deep neural networks can successfully address both ordinary differential equations (ODEs) and partial differential equations (PDEs). Raissi et al. [21] have proposed a novel approach called the physics-informed neural network (PINN). This approach involves integrating the physical principles of the dynamical system into the loss function of the neural network, thereby enabling the network to learn the parameters and dynamics of the system simultaneously. It is worth mentioning that the universal approximation theorem is the theoretical basis for the ability of a deep neural network to learn any arbitrary function [22, 23].

A survey of the current state-of-the-art deep learning algorithms for epidemic models reveals the following: A study conducted by Abdelhafid et al. [24] aimed to compare the performance of five deep learning algorithms in terms of forecasting the number of new and recovered cases of COVID-19. The algorithms considered in the study were recurrent neural network, long short-term memory (LSTM), bidirectional LSTM, gated recurrent units (GRUs) and variational autoEncoder (VAE). Abdelkader et al. [25] conducted a comparative study to evaluate the effectiveness of various machine learning methods in forecasting COVID-19 transmission. The study explored deep learning models, including a hybrid convolutional neural networks LSTM (LSTM-CNN), the hybrid GRU-convolutional neural networks, GAN, CNN, LSTM and Restricted Boltzmann machine. Wang et al. [26] introduced a machine learning-assisted framework that combines cellular automata with time sensitive-undiagnosed-infected-removed (SUIR) model to assess the multi-scale risk of COVID-19 transmission. This model not only predicts epidemic dynamics but it also reveals the transmission modes of the coronavirus in different scenarios. By utilizing transfer learning techniques, the framework predicted the prevalence of COVID-19 in all 412 counties in Germany, providing a t-day-ahead risk forecast and assessing the impact of non-pharmaceutical intervention policies. A study by Han et al. [27] introduced a PINN embedded with the SIR model to analyze the temporal evolution dynamics of infectious diseases. The approach was validated using synthetic data from the susceptible-asymptomatic-infected-recovered-dead (SAIRD) model, demonstrating its effectiveness in accurately capturing and predicting disease dynamics.

This study involved the development of a stochastic epidemiology-informed neural network (SEINN) to learn about the dynamics of a stochastic epidemic vaccine model, which does not have a nonlinear incidence rate. We discretize the system of SDEs that reflects the epidemiology of the model by using the Euler-Murayama method [28]. Then we encode the resulting discretized system as a discrete loss. Subsequently, we compare the achieved results with those of a deterministic model by utilizing error metrics for data-driven simulation. Furthermore, we contrast the deterministic and stochastic models based on their basic reproduction numbers [29]. Next, we use a SEINN to analyze and evaluate a stochastic epidemic model with a nonlinear incidence rate. We integrate a regularization technique [30] with our proposed method to improve the accuracy of the training process to learn the nonlinear incidence rate and the model's dynamics. Finally, we present numerous analyses to examine the effects of stochasticity in conjunction with vaccination rates on the dynamics of the incidence rate. The data and details of the implementation of the algorithms in this work are available on **github**. In pursuit of our objective, we have made the following primary contributions:

- 1) We introduce a simulation based on data to illustrate the significance of stochastic models in epidemiology.

- 2) We utilize data-driven simulations to indicate that models with nonlinear incidence rates may offer greater realism and efficacy in capturing the intricacies of disease transmission.
- 3) The efficacy of our proposed approach is demonstrated via a series of computational analyses, including sensitivity analysis and overfitting analysis.

The paper's organization is as follows: Section 2 describes the materials and methods used in the study. Section 2.1 elaborates on the mathematical models utilized in the study. Section 2.2 provides a thorough presentation of the proposed method. Section 2.3 presents the error metrics for data-driven simulation. Section 3 provides an exposition of the findings and analyses. Section 3.1 presents the simulations for a stochastic vaccine model based on data-driven approaches. Section 3.2 discusses the outcomes for nonlinear incidence rates. Section 3.3 details a computational analysis of the SEINN. Section 3.4 provides a comprehensive account of the analysis and interpretation of the results. Section 4 summarizes our work.

2. Materials and methods

2.1. Mathematical models

The present section presents three mathematical epidemiological models utilized in the current paper: the deterministic COVID-19 vaccine model, the stochastic COVID-19 vaccine model, and the nonlinear incidence stochastic COVID-19 vaccine model.

2.1.1. Deterministic COVID-19 vaccine model

The original SIR model [3] is a widely recognized compartmental model that is simple and effective. The assumption is made that the population size N remains constant. The population N is partitioned into three distinct compartments, namely susceptible (S), infected (I) and recovered (R). This model assumes that people in the same group have the same characteristics. This means that each group is homogeneous. The model uses the following differential equations to describe how the groups change over time:

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{SI}{N} - v\eta S \\ \frac{dI}{dt} &= \beta \frac{SI}{N} - \gamma I \\ \frac{dR}{dt} &= \gamma I + v\eta S\end{aligned}\tag{2.1}$$

The deterministic model (2.1) has two trainable parameters: β , which is the rate of infection per contact between a susceptible and an infectious person; γ , which is the rate of recovery for an infectious person. It is the inverse of D , which is the average duration of infection. The vaccination rate, v , and efficacy rate, η are assumed to be fixed in the model. This means that the model also includes a vaccination term $v\eta S$, which accounts for the fraction of susceptible people who get vaccinated at a rate $v\eta$. The model starts with $S(t_0) > 0$, $I(t_0) \geq 0$, and $R(t_0) \geq 0$ at time t_0 . It also assumes that the population size N is constant, so $S(t) + I(t) + R(t) = N$ for any time t . The model ignores the effects of births and deaths on the population.

2.1.2. Stochastic COVID-19 vaccine model

The model (2.1) is modified to include randomness, which can capture the uncertainty and variability in the epidemic dynamics. The model uses SDEs, which are equations that involve random terms or noise. The noise is represented by the parameters w_i , $i = 1, \dots, 3$ and σ_i , $i = 1, \dots, 3$, which are related to the Brownian motion and the noise intensity, respectively. The system of SDEs describes the stochastic vaccine model:

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{SI}{N} - v\eta S + \sigma_1 S w_1 \\ \frac{dI}{dt} &= \beta \frac{SI}{N} - \gamma I + \sigma_2 S w_2 \\ \frac{dR}{dt} &= \gamma I + v\eta S + \sigma_3 S w_3\end{aligned}\quad (2.2)$$

Model (2.2) can be solved numerically by using methods such as Euler-Maruyama, which approximates the SDEs by using discrete steps. However, in this work, these equations are discretized and encoded into the loss function of a deep learning algorithm to learn the dynamics of the model.

2.1.3. Nonlinear incidence stochastic COVID-19 vaccine model

A nonlinear incidence rate [25] is observed when new cases of illness are not directly proportional to the population's susceptibility.

$$g(S, I) = \frac{kS^h I}{S^h + \alpha I^h} \quad (2.3)$$

where h and k are positive constants, and α denotes the psychological or inhibitory impact of the viral spread. The equation is defined as the ratio of the product of k , S^h , and I to the sum of S^h and αI^h , where S and I represent distinct variables. We examine the effects of varying the noise level σ_i in conjunction with alterations in k , h , and α on the incidence rate. It is important to study the effects of varying these parameters to gain insights into how different factors contribute to the spread of an illness and to better understand the dynamics of the population's susceptibility and infected individuals from perspective of the incidence rate.

The nonlinear incidence stochastic COVID-19 vaccine model is described by a system of differential equations, as shown below:

$$\begin{aligned}\frac{dS}{dt} &= b - dS - g(S, I) + \gamma R - v\eta S + \sigma_1 S w_1 \\ \frac{dI}{dt} &= g(S, I) - (d + \mu + \delta)I + \sigma_2 I w_2 \\ \frac{dR}{dt} &= \mu I - (d + \gamma)R + v\eta S + \sigma_3 R w_3\end{aligned}\quad (2.4)$$

where b denotes the recruitment rate, d represents the natural death rate, μ is the natural recovery rate, and γ is the rate at which recovered individuals lose immunity and return to the susceptible state. The variable δ represents the mortality rate associated with the disease. The differential equations describe the dynamic changes in the population's susceptible (S), infected (I), and recovered (R) individuals over time, considering recruitment, death, recovery, vaccination, and the impact of the nonlinear incidence rate function.

2.2. Deep learning algorithms

This section discusses the application of PINNs in the context of epidemiological models and their associated optimization techniques. Specifically, we utilize the epidemiology-informed neural network (EINN) proposed in [9] to learn the dynamics of a deterministic COVID-19 vaccine model. Furthermore, we introduce and apply the SEINN to two stochastic vaccine models. By combining the power of neural networks with the principles and constraints of epidemiology, these models provide a powerful tool for understanding and simulating the spread of infectious diseases, particularly in the context of COVID-19 vaccination strategies.

2.2.1. Physics-informed neural network

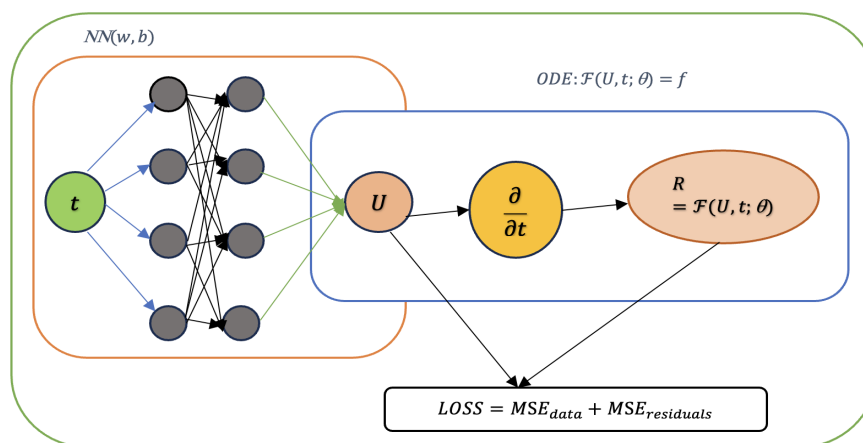


Figure 1. A schematic diagram of PINNs. The orange solid-line block is a neural network that takes time t as input, and the output is U , with w and b being the respective weights and biases. The light blue solid-line block shows the calculation of residual loss. The loss function consists of data loss from a mismatch between the observed data and output U and residuals from the differential equations. By minimizing the loss function ($LOSS = MSE_{data} + MSE_{residuals}$), we obtain the parameters of PINNs.

The core idea behind PINNs is to incorporate the prior knowledge of a system into the learning process of a deep neural network. This prior knowledge can be ODEs/PDEs, which describe the underlying physical laws or domain expertise. PINNs achieve this integration by incorporating these equations into the loss functions used during training. The network weights, biases, and model parameters (physical laws) are optimized during training. The loss function in PINNs typically consists of two terms: the data loss and the residual loss. The data loss ensures that the network fits the available data, while the residual loss satisfies the underlying physics equations or constraints. Figure 1 illustrates the training process for an ODE-dynamics-informed neural network with a PINN. The neural network is trained by minimizing the combined losses derived from the data and the residual terms. By simultaneously optimizing the network and the model parameters, PINNs can effectively tackle the

solution of systems of ODEs.

$$\frac{\partial U}{\partial t}(t) + F(U(t); \theta) = 0, \quad t \in [t_0, T] \quad (2.5)$$

with

$$U(t) = [u_1(t), \dots, u_n(t)], \quad F(U) = [f_1(U), \dots, f_n(U)] \quad (2.6)$$

where $u_k \in \mathbb{R}$ and $f_k : \mathbb{R} \rightarrow \mathbb{R}$, $k = 1, \dots, n$, t_0 is initial time and T is the final time. F is the function and U the solution. $\theta \in \mathbb{R}^q$ denotes the unknown parameters of the system. The unknown parameters θ can be determined from observed data U_o at times t_1, \dots, t_m .

The data loss is computed as

$$L_{data} = \sum_{o=1}^m \|U(t_o) - U_o\|^2. \quad (2.7)$$

In data fitting approaches, the model's parameters are determined by minimizing equation (2.7). This minimization process aims to find a solution $U(t)$ that best fits the observed data by minimizing the sum of squared deviations (least squares). In Figure 1, we have

$$NN_{(w,b)}(t) : \mathbb{R} \rightarrow \mathbb{R}^q \quad (2.8)$$

which approximates the solution

$$U(t) : \mathbb{R} \rightarrow \mathbb{R}^q \quad (2.9)$$

where $U(t)$ represents a system of first-order ODEs and w and b are the respective weights and biases of the neural network $NN(w, b)$. To solve the ODEs with a neural network, the neural network parameters are optimized to fit the observed data by using the idea of least squares.

$$\arg \min_{w,b} (MSE_{w,b}^U) \quad (2.10)$$

This means the loss in terms of observed data ($U_o, o = 1, \dots, m$) can be computed as

$$MSE_{w,b}^U := \frac{1}{m} \sum_{o=1}^m \|NN_{w,b}(t_o) - U_o\|^2. \quad (2.11)$$

The residual loss from ODEs is calculated as

$$\mathcal{F}(NN_{w,b}; t; \theta) = \frac{\partial NN_{w,b}}{\partial t}(t) + F(NN_{w,b}(t)) \quad (2.12)$$

Equation (2.12) enables the PINN to enforce the satisfaction of the underlying physics-based equations or constraints by incorporating the derivatives of the network output with respect to time ($\partial NN_{w,b}/\partial t$) and subtracting the corresponding ODEs. The PINN's loss function is then optimized, guiding the training process to find the optimal values for the network's weights w and biases b that minimize the discrepancy between the neural network predictions and the physics-based constraints. By integrating the residual term into the loss function and leveraging automatic differentiation, PINNs provide a powerful framework for combining neural networks with prior knowledge of physical laws, enabling more accurate modeling and predictions. Combining the residual loss and data loss, the objective function will be as follows:

$$\arg \min_{w,b,\theta} (MSE_{w,b}^U + MSE_{w,b,\theta}^{\mathcal{F}}) \quad (2.13)$$

2.2.2. Epidemiology-informed neural network

EINNs [9], inspired by PINNs, were developed to explicitly incorporate epidemiological constraints into the loss function. The EINN is utilized to capture the dynamics of a deterministic COVID-19 vaccine model, allowing us to analyze and predict the effects of various factors on the spread of the disease.

Figure 2 shows a fully connected dense neural network (marked by the black-solid frame) that is used to evaluate the SIR model. The neural network takes as input the time t and outputs the values of the susceptible population $S(t)$, the infected population $I(t)$ and the recovered population $R(t)$. The values of these compartments obey the SIR model at the given input time t . The residual term (2.12) of the ODEs of the SIR model can be minimized to enforce Eq (2.8).

To impose the epidemiological constraints in the EINN, we define F as

$$\mathcal{F}(NN_{w,b};t;\beta;\gamma) = \begin{bmatrix} \frac{dS(t)}{dt} + \frac{\beta S(t)I(t)}{N} - v\eta S(t) \\ \frac{dI(t)}{dt} - \frac{\beta S(t)I(t)}{N} + \gamma I(t) \\ \frac{dR(t)}{dt} - \gamma I(t) + v\eta S(t) \end{bmatrix} \quad (2.14)$$

This means that the residual loss in terms of mean squared error will be defined as follows:

$$MSE_{SIR} = MSE_{S_{residual}} + MSE_{I_{residual}} + MSE_{R_{residual}} \quad (2.15)$$

where

$$\begin{aligned} MSE_{S_{residual}} &= \frac{1}{n} \sum_{i=1}^n \left\| \frac{dS(t_i)}{dt_i} + \beta \frac{S(t_i)I(t_i)}{N} - v\eta S(t_i) \right\|^2 \\ MSE_{I_{residual}} &= \frac{1}{n} \sum_{i=1}^n \left\| \frac{dI(t_i)}{dt_i} - \beta \frac{S(t_i)I(t_i)}{N} + \gamma I(t_i) \right\|^2 \\ MSE_{R_{residual}} &= \frac{1}{n} \sum_{i=1}^n \left\| \frac{dR(t_i)}{dt_i} - \gamma I(t_i) + v\eta S(t_i) \right\|^2 \end{aligned} \quad (2.16)$$

The variable n represents the total count of discrete time points. It should be noted that the discrete-time points have been selected to align with the observed time step, which has been chosen to be in units of one natural day. Alternatively, the interval separating two consecutive time points is denoted as $\Delta t = 1$.

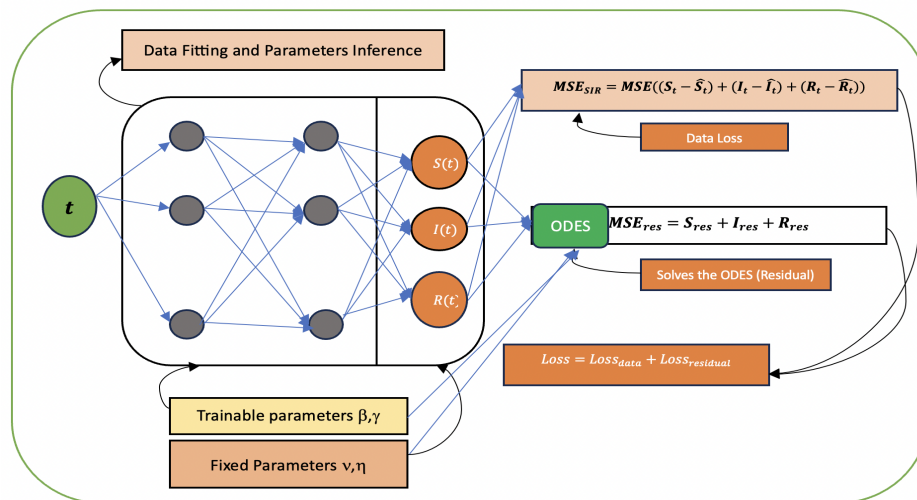


Figure 2. A schematic diagram of EINNs. The black-solid block consists of a dense neural network that takes time t as input and predicts the susceptible (S), infected (I) and recovered (R) populations. The trainable parameters include the transmission rate β and recovery rate γ , while fixed parameters represent the vaccination rate ν and vaccine efficacy η . The ‘ODEs’-labeled box shows the residual calculation that captures the deviation from the expected behavior based on the vaccine model. The loss function incorporates both data mismatch and physical residuals to optimize the network’s performance. By minimizing the loss function, the EINN fits the observed data, infers dynamic parameters β and γ and satisfies the dynamics of the SIR model. This integration enables accurate modeling and prediction of population dynamics in epidemiology.

The mean squared error for the observed data can be expressed as follows:

$$MSE_{data} = MSE_{S_{data}} + MSE_{I_{data}} + MSE_{R_{data}} \quad (2.17)$$

where

$$\begin{aligned} MSE_{S_{data}} &= \frac{1}{o} \sum_{i=1}^o \|S(t_i) - S_{o_i}\|^2 \\ MSE_{I_{data}} &= \frac{1}{o} \sum_{i=1}^o \|I(t_i) - I_{o_i}\|^2 \\ MSE_{R_{data}} &= \frac{1}{o} \sum_{i=1}^o \|R(t_i) - R_{o_i}\|^2 \end{aligned} \quad (2.18)$$

where S_{o_i} , I_{o_i} , R_{o_i} represent the observed data at t_i and o is the total number of observed data points. The total loss will comprise the data loss and residual loss. As the total loss is minimized, the weights and biases, along with the trainable parameters of the model, are optimized.

Algorithm 1 demonstrates the utilization of EINN to determine trainable parameters, including the NN- and deterministic COVID-19 vaccine SIR-model parameters. The input to the algorithm is the

time point, denoted as t , and the output is the corresponding value of each compartment in the SIR model. The weights (w) and biases (b) of the neural network, as well as the model parameters (β and γ) of the SIR model, are initialized randomly. The initializations for w and b are Xavier and zero initializations, respectively, in Tensorflow. For β and γ , random values between 0 and 1 are selected. This algorithm is a guideline for using EINNs to estimate the parameters required for the neural network and the embedded SIR model. By leveraging this approach, it becomes possible to simultaneously train the neural network and determine the optimal values of the SIR model parameters.

Algorithm 1 EINN

Require: t, S_o, I_o, R_o

Randomly initialize weights w , biases b and dynamic parameters β, γ

for epoch **in** epochs **do**

Obtain the values of each compartment of the SIR model by using the forward propagation of the neural network with the input as t :

$$S, I, R = \text{NN}(t)$$

Calculate the composed loss function, including the data loss (with o representing the number of observations in each compartment and thus the number of collected time points):

$$MSE_{SIR} = \frac{1}{o} \sum_{i=1}^o \|S_i - S_{o_i}\|^2 + \|I_i - I_{o_i}\|^2 + \|R_i - R_{o_i}\|^2$$

Denote the mismatch of the output of the neural network and observation data. Here, the residual loss is defined as:

$$MSE_{Residuals} = \frac{1}{n} \sum_{i=1}^n \left(\left\| \frac{dS_i}{dt_i} + \frac{\beta S_i I_i}{N} - v\eta S_i \right\| + \left\| \frac{dI_i}{dt_i} - \frac{\beta S_i I_i}{N} + \gamma I_i \right\| + \left\| \frac{dR_i}{dt_i} - \gamma I_i + v\eta S_i \right\| \right)^2$$

This represents the sum of the squared residual errors for each compartment of the SIR model. The residuals and data loss are calculated by using the same time step $\Delta t = 1$.

The total loss function is given by:

$$Loss = MSE_{SIR} + MSE_{Residuals}$$

Update the weights w and biases b , as well as the dynamic parameters β and γ by using the Adam optimizer toolkit in Tensorflow to minimize the loss function.

end for=0

In our experimental setup, our neural network architecture takes a single input value, denoted as t . The network consists of multiple hidden layers, with each connection between nodes characterized by weights $W[i, j]$. Here, i refers to the starting node position, and j corresponds to the ending node position. The tanh activation function is applied at each node within the hidden layers. The *tanh*

activation function is mathematically defined as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.19)$$

This activation function maps the input value x to a range between -1 and 1 , introducing non-linear characteristics to the neural network. By employing \tanh as the activation function at each node in the hidden layers, the network can capture intricate patterns and discover complex relationships within the data. In the output nodes of the neural network, we employ the sigmoid activation function to account for the normalization applied to $S(t)$, $I(t)$ and $R(t)$. The sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.20)$$

This activation function maps the input x to a value between 0 and 1 , facilitating the representation of probabilities or values within a normalized range. The EINN architecture comprises four hidden layers, each with 64 neurons. Since physics-informed algorithms are data-hungry, we employed interpolation to generate 3000 data points within 150 data points. For optimization, we utilized the Adam optimizer algorithm from the Tensorflow package. The chosen learning rate was set to 0.0001 , and the training process was executed for $40k$ epochs. We employed regularization parameters to prevent overfitting and enhance the model's generalization ability. We chose $\alpha_1 = 1$ and $\alpha_2 = 1$ as regularization values for data loss and residual loss, respectively.

2.2.3. Stochastic epidemiology-informed neural network

The SEINN is introduced as an extension of the EINN [9] framework to incorporate stochasticity [16, 31] into vaccine models. This stochastic component enables the consideration of uncertainty and variability in the model, providing a more comprehensive understanding of the real-world complexities involved in the vaccination process.

In Algorithm 2, the SEINN algorithm is designed to learn the parameters of a neural network and a stochastic vaccine model for the SIR system. It combines deep learning techniques with the ability to capture the stochastic dynamics in the system. The algorithm begins by randomly initializing the neural network's weights, biases and dynamic parameters β and γ . These parameters will be optimized during the training process. Let us denote the weights as w and biases as b . The algorithm then enters a loop over a specified number of epochs. Each epoch represents a complete pass through the entire dataset during training. Within each epoch, an inner loop iterates over the number of Monte Carlo iterations [31] (N_{MC}). This loop captures the stochasticity in the system by repeatedly simulating the SIR model with different random noise inputs. In each Monte Carlo iteration, the algorithm performs forward propagation of the neural network by using the input time t to obtain the predicted values for each compartment of the SIR model: S , I and R . This can be expressed as follows:

$$S, I, R = NN(t; w, b) \quad (2.21)$$

Here, NN represents the forward propagation function of the neural network in the black-solid block of Figure 3. The initializations for w and b are Xavier and zero initializations, respectively, in Tensorflow. Next, the algorithm performs Euler-Maruyama discretization to calculate the next values of S , I and R

based on the SDE terms and random noise. The discrete update equations for each compartment can be written as follows:

$$\begin{aligned} S_{i+1} &= S_i - \left(\beta \frac{S_i I_i}{N} + v\eta S_i\right)\Delta t + \sigma_1 \sqrt{\Delta t} S_i dW_1 \\ I_{i+1} &= I_i + \left(\beta \frac{S_i I_i}{N} - \gamma I_i\right)\Delta t + \sigma_2 \sqrt{\Delta t} I_i dW_2 \\ R_{i+1} &= R_i + (\gamma I_i - v\eta S_i)\Delta t + \sigma_3 \sqrt{\Delta t} R_i dW_3 \end{aligned} \quad (2.22)$$

In these equations, β represents the infection rate, γ represents the recovery rate, N represents the total population size, Δt is the time step size, σ_i ($i = 1, \dots, 3$) is the random noise (scaled by the square root of Δt) and dW_i is a random increment following a standard normal distribution.

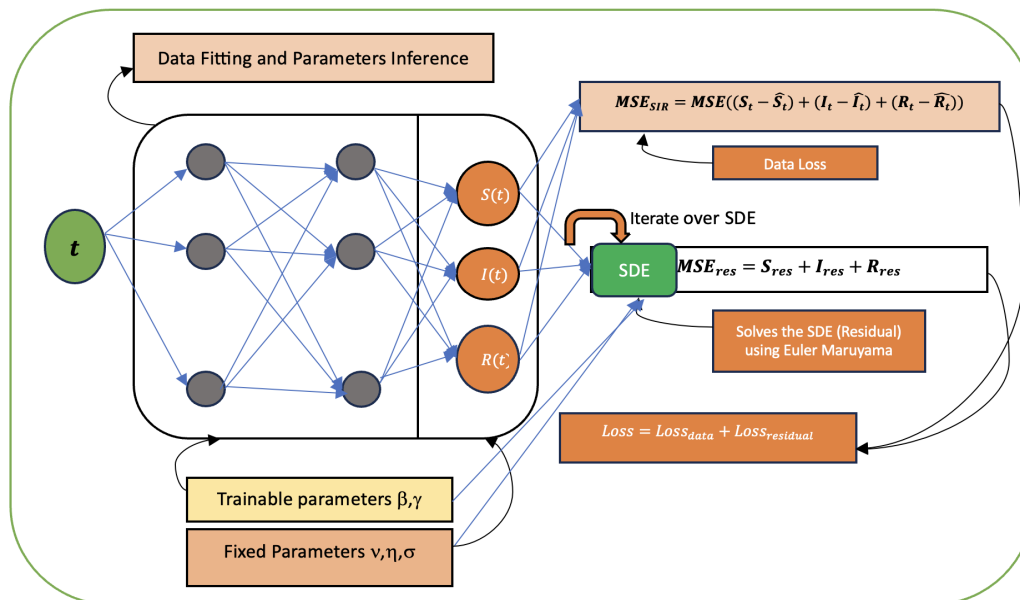


Figure 3. A schematic diagram of SEINNs. The black-solid block consists of a dense neural network that takes time t as input and predicts the susceptible (S), infected (I) and recovered (R) populations. The trainable parameters include the transmission rate β and recovery rate γ , while fixed parameters represent the vaccination rate v , vaccine efficacy η and noise level σ . The individual compartments S, I, R are each corrupted with different choice of σ . ‘SDE’-labeled box shows the residual calculation that captures the deviation from the expected behavior based on the stochastic vaccine model. The Euler Maruyama technique is used to discretize the SDE. The loss function incorporates both data mismatch and physical residuals to optimize the network’s performance. By minimizing the loss function, the SEINN fits the observed data, infers dynamic parameters β and γ , and satisfies the dynamics of the stochastic SIR model. This integration enables accurate modeling and prediction of population dynamics in epidemiology.

Algorithm 2 SEINN

Require: $t, S_o, I_o, R_o, \sigma_1, \sigma_2, \sigma_3, \eta, v, N_{MC}$, where N_{MC} is the number of iteration over the SDE
Randomly initialize weights w , biases b and dynamic parameters β, γ

for epoch in epochs **do**

for $j \leftarrow 1$ to N_{MC} **do**

 Obtain the values of each compartment of the SIR model using the forward propagation of the neural network with the input as t :

$S, I, R = \text{NN}(t)$

for time step $i \leftarrow 1$ to M **do**

 Calculate the SDE terms by using Euler-Maruyama discretization:

$$S_{i+1} = S_i - \left(\beta \frac{S_i I_i}{N} + v \eta S_i\right) \Delta t + \sigma_1 \sqrt{\Delta t} S_i dW_i$$

$$I_{i+1} = I_i + \left(\beta \frac{S_i I_i}{N} - \gamma I_i\right) \Delta t + \sigma_2 \sqrt{\Delta t} I_i dW_i$$

$$R_{i+1} = R_i + (\gamma I_i - v \eta S_i) \Delta t + \sigma_3 \sqrt{\Delta t} R_i dW_i$$

 Calculate the discrete loss after each iteration of the SDE:

$$MSE_{SDE} = \frac{1}{M} \sum_{i=1}^M (||S_{i+1} - S_i||^2 + ||I_{i+1} - I_i||^2 + ||R_{i+1} - R_i||^2)$$

end for

 Obtain a list of the values of each compartment S, I, R after each iteration over the SDE.

 Compute the average from the list of each compartment as the solution to the stochastic model in (2.2)

end for

 Calculate the composed loss function, including the data loss and residual loss:

$$MSE_{SIR} = \frac{1}{M} \sum_{i=1}^M (||S_i - S_{o_i}||^2 + ||I_i - I_{o_i}||^2 + ||R_i - R_{o_i}||^2)$$

$$MSE_{residual} = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} MSE_{SDE}$$

The total loss function is given by:

$$Loss = MSE_{SIR} + MSE_{residual}$$

Update the weights w , biases b and dynamic parameters β and γ by using the Adam optimizer in Tensorflow to minimize the loss function.

end for=0

The algorithm then calculates each time step's discrete loss MSE_{SDE} . This loss measures the discrepancy between the successive values of S, I, and R and helps to quantify the error in approximating the continuous-time SDE with the Euler-Maruyama discretization scheme. The MSE_{SDE} can be com-

puted as follows:

$$MSE_{SDE} = \frac{1}{o} \sum_{i=1}^o (||S_{i+1} - S_i||^2 + ||I_{i+1} - I_i||^2 + ||R_{i+1} - R_i||^2) \quad (2.23)$$

where o is the total number of time steps. Throughout the Monte Carlo iterations, a list is maintained to store the values of each compartment (S , I , and R) after each iteration. This list captures the evolution of the compartments over the iterations and serves as the basis for computing the average solution to the stochastic model. Once the Monte Carlo iterations are completed, the algorithm computes the average of each compartment from the list of values. This average represents the solution to the stochastic model and provides a more stable estimate of the compartment values by mitigating the effect of randomness. To evaluate the performance of the model, the algorithm calculates the composed loss function, which consists of the data loss (MSE_{SIR}) and the residual loss ($MSE_{residual}$). The data loss measures the discrepancy between the dataset's predicted values (S, I, R) and the observed values (S_o, I_o, R_o). It can be computed as follows:

$$MSE_{SIR} = \frac{1}{o} \sum_{i=1}^o (||S - S_o||^2 + ||I - I_o||^2 + ||R - R_o||^2) \quad (2.24)$$

where o represents the number of observations in each compartment, and thus the number of collected time points. The residual loss, $MSE_{residual}$, is calculated as the average of the MSE_{SDE} over all Monte Carlo iterations. This loss captures the discrepancy between the predicted evolution of the compartments based on the SDE simulations and the observed values. It can be computed as follows:

$$MSE_{residual} = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} (MSE_{SDE}) \quad (2.25)$$

N_{MC} is the number of Monte Carlo iterations. The total loss function, $Loss$, is the sum of the data loss (MSE_{SIR}) and the residual loss ($MSE_{residual}$). It can be written as:

$$Loss = MSE_{SIR} + MSE_{residual} \quad (2.26)$$

The algorithm utilizes the Adam optimizer [32], a popular optimization algorithm, to update the weights, biases and dynamic parameters. The Adam optimizer adjusts the parameters based on the gradients of the loss function. By iteratively updating the parameters using the optimizer, the algorithm aims to minimize the loss function and improve the accuracy of the model predictions. The SEINN architecture comprises four hidden layers, each with 64 neurons. The nonlinear activation function in Eq (2.19) is applied to all hidden layers. The sigmoid function in Eq (2.20) is applied to the output layer. Since physics-informed algorithms are data-hungry, we employed interpolation to generate 3000 data points within 150 data points. For optimization, we utilized the Adam optimizer algorithm from the Tensorflow package. The chosen learning rate was set to 0.0001, and the training process was executed for 40k epochs. We employed regularization parameters to prevent overfitting and enhance the model's generalization ability. We chose $\alpha_1 = 1$ and $\alpha_2 = 1$ as regularization values for data loss and residual loss, respectively. The number of Monte Carlo iterations N_{MC} was set at 10.

2.3. Error metrics for data-driven simulations

The present study employed error metrics for data-driven simulations. The variable y_i denotes the actual data, while \hat{y}_i corresponds to the predicted data the models generate.

- 1) **Root Mean Square Error (RMSE):** pertains to the statistical measure of the deviation of prediction errors from their mean value.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}. \quad (2.27)$$

- 2) **The Mean Absolute Percent Error (MAPE):** measures the accuracy of a forecasting method and is expressed as a percentage. It is calculated by taking the absolute percentage difference between the actual and predicted values and averaging it across all observations.

$$MAPE = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y} \right| \%. \quad (2.28)$$

- 3) **Relative Error (REL):** is expressed as the sum of the squared difference between the actual value (y) and the predicted value (\hat{y}_i), divided by the square of the actual value, for each of the N observations.

$$REL = \sum_{i=1}^N \frac{(y_i - \hat{y}_i)^2}{y_i^2}. \quad (2.29)$$

- 4) **Explained Variance (EV):** refers to the extent of variability in the predicted \hat{y}_i that can be accounted for by the neural network. This resembles the coefficient of determination (R^2), which is predominantly employed in the context of linear regression.

$$EV = 1 - \frac{Var(y_i - \hat{y}_i)}{Var(y_i)}. \quad (2.30)$$

3. Results and discussion

3.1. Data-driven simulation for stochastic vaccine model

The COVID-19 data were procured from the Tennessee Health Department's official website. The temporal scope of the dataset employed in this study spans December 17, 2020, to May 16, 2020. Cumulative data about the number of individuals infected and recovered were extracted and processed. The susceptible data were obtained by subtracting the number of infected and recovered individuals from the total population, which is a known quantity. The data were subjected to preprocessing and scaling techniques to convert the values into a standardized range of 0 to 1. This scaling was done to facilitate the training process of the models. The primary analytical components used in the study were the cumulative counts of individuals who have recovered from and been infected with COVID-19. For reproducibility, we fixed values for the parameters $\beta = 0.18$, $\gamma = 0.13$, vaccination rate $v = 10\%$ and efficacy rate $\eta = 0.94$ [9]. Both the deterministic and stochastic models underwent training and

evaluation across varying noise levels (5%, 10%, 30%, 60%). We show the results for noise levels 5% and 60%. The remaining graphs will be shown in the appendix.

Figure 4 displays the actual COVID-19 data and the corresponding noisy data for the susceptible, infected and recovered groups. The data cover the period from December 17, 2020, to May 16, 2020, for the state of Tennessee. From the graph, the solid lines in each subplot represent the actual data, which are the true values of S , I and R at each time point. These values were obtained from reliable sources and serve as the ground truth for the COVID-19 dynamics in Tennessee. Different noise levels were introduced to the actual data to simulate real-world scenarios and account for measurement errors or uncertainties. The dashed lines in each subplot represent the noisy data. The noise levels depicted in the figure are 5%, 10%, 30% and 60%. At the lowest noise level (5%), the dotted line closely follows the dashed line, indicating that the noise has minimal impact on the data. As the noise level increases to 10%, 30%, 60%, the dotted lines deviate further from the dashed line, indicating greater discrepancies between the actual and noisy data. The noise in the data is introduced to mimic the challenges faced in real-world data collection, such as measurement errors, reporting inaccuracies or other sources of uncertainty. By incorporating these noisy data points, the SEINN algorithm can learn to account for such uncertainties and make more robust predictions.

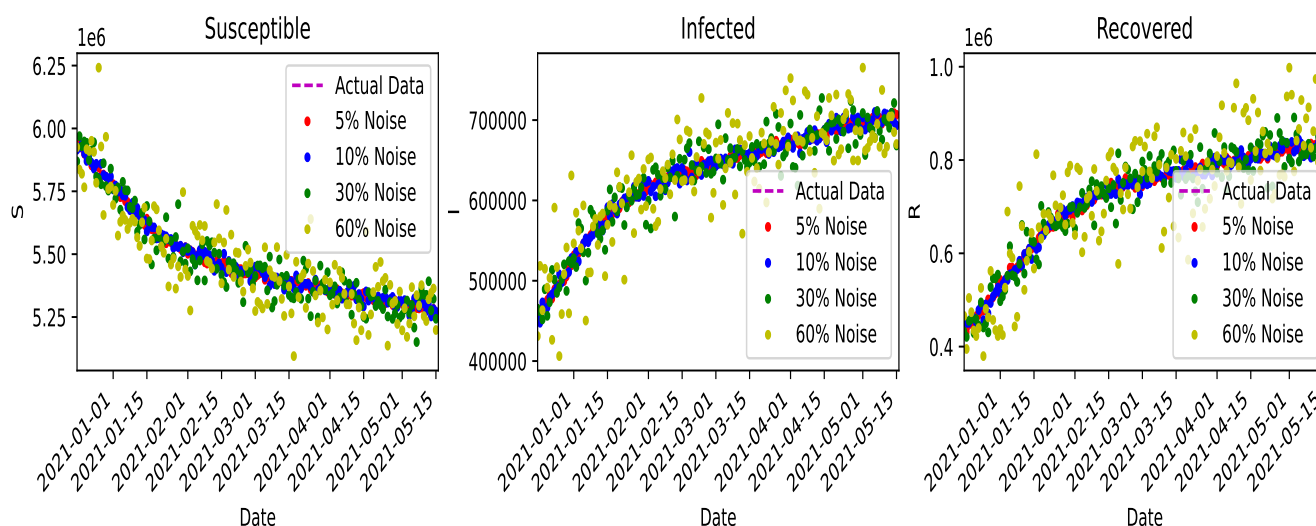


Figure 4. Actual COVID-19 data and noisy data for the state of Tennessee from December 17, 2020 to May 16, 2020. The left graph shows the susceptible group whilst the middle and right graphs show the infected and recovered groups, respectively.

Figure 5 compares the performance of the EINN and SEINN models for noisy data at a 5% noise level. The graph illustrates how SEINN outperforms EINN in terms of data fitting, specifically for noisy data. The plot consists of two sets of curves, each representing the predictions made by the EINN and SEINN models. Each graph's dotted line represents the noisy data, which serves as a reference for evaluating the performance of the two models. These noisy data points were obtained by introducing a 5% noise level to the actual COVID-19 data. Comparing the predictions made by EINN and SEINN, it is evident that SEINN performed better in terms of capturing the pattern and characteristics of the noisy data. The SEINN predictions, represented by the blue dashed lines, align more closely with the

dotted lines, indicating a better fit to the actual data.

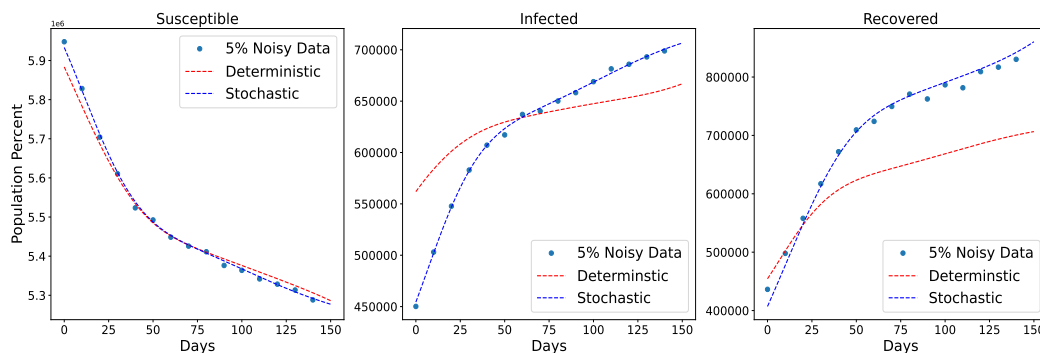


Figure 5. Comparing EINN and SEINN for noisy data at 5% noise level. The left figure shows the results for deterministic and stochastic models at 5% noise level for susceptible group. The middle and right figures show the results for infected and recovered groups, respectively.

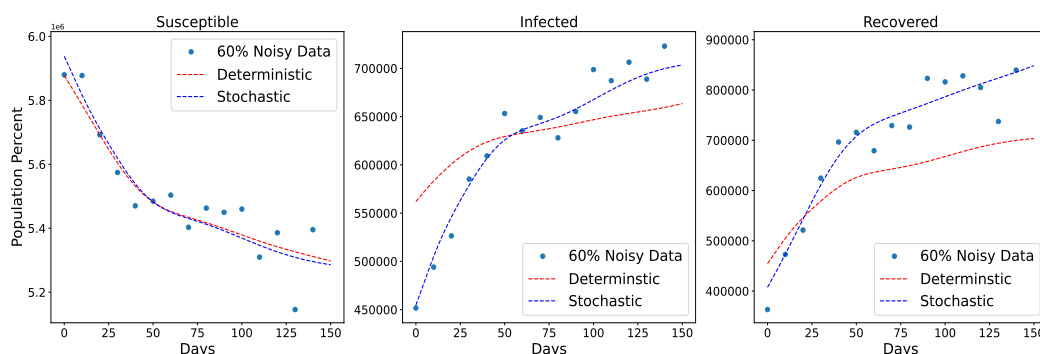


Figure 6. Comparing EINN and SEINN for noisy data at 60% noise level. The left figure shows the results for deterministic and stochastic models at 6% noise level for susceptible group. The middle and right figures show the results for infected and recovered groups, respectively.

Figure 6 compares the performance of the EINN and SEINN models for noisy data at a 60% noise level. This figure demonstrates how the two models handle and fit the data when the noise level is significantly higher. Comparing the predictions made by EINN and SEINN, it can be observed that both models struggled to accurately capture the pattern and characteristics of the noisy data. The EINN predictions, represented by the red dashed line, deviate significantly from the blue dotted line, indicating a poor fit to the actual data. Similarly, the SEINN predictions, represented by the blue dashed lines, also exhibit some deviations from the blue dotted line. The SEINN appears to slightly capture the pattern. The high noise level in the data introduced significant uncertainty and variability, making it challenging for both models to accurately capture the underlying dynamics. The deterministic EINN model, based on deterministic differential equations, is particularly limited in terms of handling high noise levels. It

fails to account for the stochastic nature of the data and does not effectively capture the variations and uncertainties present in the noisy data. On the other hand, the SEINN model, incorporating stochastic modeling techniques through SDEs (SDEs), attempts to capture the uncertainty in the data. However, even the SEINN predictions show deviations from the dotted lines, indicating that the noise level is substantial enough to pose challenges for both models.

Table 1 compares the EINN and SEINN models based on two error metrics, namely, the RMSE and MAPE. A comparison was performed for different noise levels under the assumption of a fixed efficacy rate of $\eta = 94\%$ and vaccination rate of $\nu = 10\%$. The table consists of four rows, each corresponding to a different noise level: 5%, 10%, 30%, and 60%. For each noise level, two columns represent the models being compared: the deterministic model (EINN) and the stochastic model (SEINN). Comparing the results, it is evident that the SEINN model consistently outperformed the EINN model in terms of both the RMSE and MAPE at all noise levels. For example, at a noise level of 5%, the RMSE for the deterministic model was 39,006, while the RMSE for the stochastic model was significantly lower at 3363. Similarly, the MAPE for the deterministic model was 0.0615, whereas the stochastic model achieved a much lower MAPE of 0.0053. As the noise level increases, the performance gap between the two models becomes more pronounced. At higher noise levels, such as 60%, the RMSE for the deterministic model is 53,960, whereas the stochastic model achieved a significantly lower RMSE of 37,557. The same trend is observed for the MAPE values, with the stochastic model consistently outperforming the deterministic model. These results highlight the superiority of the SEINN model in capturing the dynamics and uncertainties associated with noisy data. By incorporating stochastic modeling techniques, the SEINN model demonstrates improved accuracy and a better fit to the data than the deterministic EINN model. The SEINN model's ability to capture the inherent variability and uncertainty in the data makes it a more suitable choice for modeling and predicting complex systems affected by noise.

Table 1. Comparison of the EINN with SEINN in terms of RMSE and MAPE.

Model comparison for $\nu = 10\%$ and efficacy rate ($\eta = 94\%$)			
Noise Level	Model	RMSE	MAPE
5%	Deterministic	39,006	0.0615
	Stochastic	3363	0.0053
10%	Deterministic	386,696	0.0610
	Stochastic	6371	0.0101
30%	Deterministic	43,497	0.0686
	Stochastic	18,802	0.0297
60%	Deterministic	53,960	0.0850
	Stochastic	37,557	0.0592

Figure 7 visually compares the performance of the EINN and SEINN models based on two error metrics: RMSE and MAPE. The figure shows that for both the RMSE and MAPE, the SEINN model consistently outperformed the EINN model across all noise levels. This is indicated by the fact that

the line representing the SEINN model consistently lies below the line representing the EINN model. Lower RMSE values indicate better accuracy and a closer match between the predicted values and the actual data. Similarly, lower MAPE values indicate a smaller percentage difference between the predicted values and the actual data. The figure demonstrates that the SEINN model achieved lower RMSE and MAPE values than the EINN model, indicating better overall accuracy and data fitting performance. This consistent trend across different noise levels reinforces the stochastic EINN model's superiority in capturing the data's dynamics and patterns.

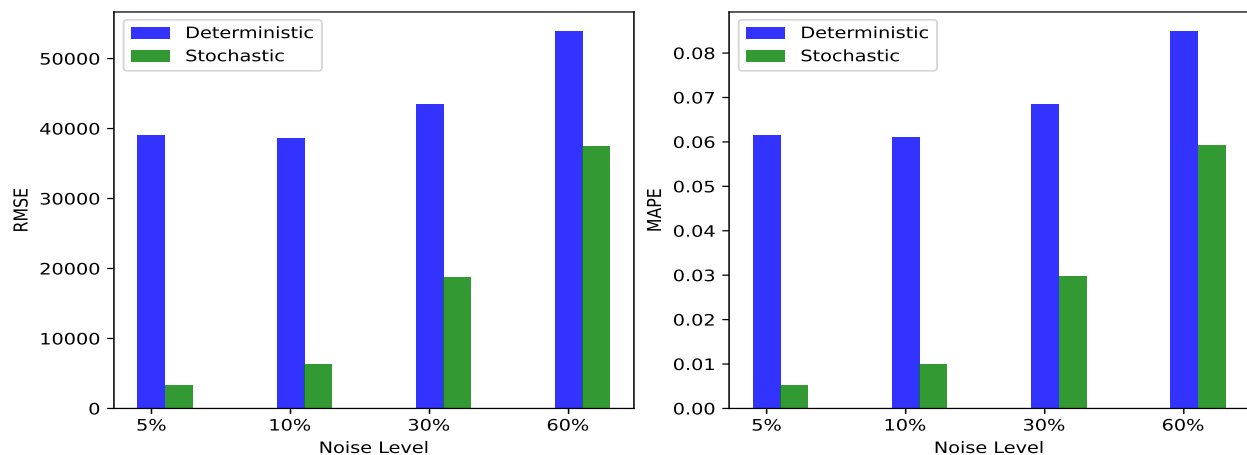


Figure 7. Comparison of the EINN with SEINN in terms of RMSE and MAPE. The left figure presents the RMSE results and the right figure shows the MAPE results for comparing deterministic and stochastic models.

3.2. Data-driven simulation for nonlinear incidence rate

The dataset used in this study to analyze COVID-19 data was obtained from the official website of the Tennessee Health Department. The dataset covers a temporal period from December 17, 2020 to May 16, 2020. It includes cumulative data on the number of individuals who were infected by and recovered from COVID-19. The number of infected and recovered individuals was subtracted from the total population, which was a known quantity, to obtain the susceptible data. This calculation provided the count of individuals who were still susceptible to the virus. Scaling techniques were applied to transform the data values into a standardized range of 0 to 1. This scaling process is commonly used in machine learning tasks to facilitate the training process of the models by ensuring that all features have similar ranges.

The data-driven simulations in this study were conducted by using a stochastic model with four different noise levels: 5%, 10%, 30% and 60%. The simulations aimed to explore the impact of noise and vaccination rates on the nonlinear incidence rate while considering fixed values for the parameters h , α and k that govern the nonlinear incidence rate. To learn the expected nonlinear incidence rate, we employed Algorithm 2, which is described in detail in the paper. The algorithm, referred to as the SEINN, leverages the power of deep neural networks to model and predict the nonlinear incidence rate. The simulations were performed for two vaccination rates: 1% and 10%. These rates represent the proportion of the population that received the vaccine. By varying the vaccination rate, we aimed to analyze its influence on the nonlinear incidence rate within the stochastic model. To conduct the simulations,

we utilized the stochastic SIR model described by 2.4 of the paper. The model parameters were set to specific values based on previous research [16]. The values chosen were: $b = 1$, $d = 0.1$, $\delta = 0.01$, $\mu = 0.05$, $k = 0.2$, $\alpha = 0.5$, $\gamma = 0.01$ and $h = 2$. These parameter values were used consistently across all simulation scenarios. Furthermore, the vaccine's efficacy rate was set to 0.94, indicating the vaccine's effectiveness in preventing infection. This efficacy rate represents the proportion of individuals who are protected from infection after receiving the vaccine. Several hyperparameters were specified to implement the SEINN for the purpose of modeling the nonlinear incidence rate. The model was trained for 30,000 epochs using a learning rate of 0.001. The neural network architecture consisted of 60 neural units, and a total of 2999 interpolation points were chosen from a pool of 150 available points. The regularization parameter $\varepsilon = 0.1$ was applied to the residual loss while $1 - \varepsilon$ was applied to the data loss. These choices of hyperparameters were made based on experimentation and optimization to achieve the best performance and accuracy in terms of capturing the dynamics of the nonlinear incidence rate.

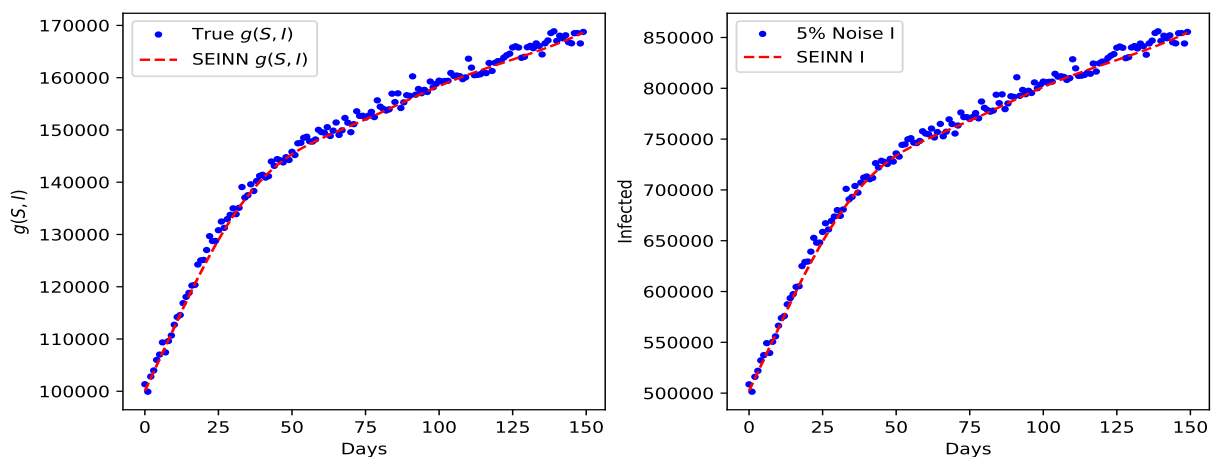


Figure 8. Nonlinear incidence rate for 5% noise level with 1% vaccination rate. The left figure shows how the true incidence rate (blue dotted line) compares with the predicted incidence rate (red broken line). The right figure shows the results for incidence rate at 5% noise level.

Figure 8 illustrates the impact of a small noise level and a small vaccination rate on the nonlinear incidence rate, denoted as $g(S, I)$. The figure showcases the data fitting performance of $g(S, I)$ and the infected group under these specific conditions. In this scenario, the noise level was set to 5%, indicating a relatively low level of uncertainty or variability in the data. Additionally, the vaccination rate was set to 1%, representing a small proportion of the population that has been vaccinated. By examining the figure, we can observe the relationship between the nonlinear incidence rate and the infected group. The figure showcases how well the nonlinear incidence rate, $g(S, I)$, aligns with the actual data of the infected group. The figure highlights that the data fitting for both the nonlinear incidence rate and the infected group is relatively better with a small noise level and a small vaccination rate. This suggests that the observed data points are in closer agreement with the model's predictions, indicating a more accurate representation of the underlying dynamics of the virus spread.

Figure 9 illustrates the impact of a higher noise level and a higher vaccination rate on the nonlinear

incidence rate, denoted as $g(S, I)$. The figure showcases the data fitting performance of $g(S, I)$ and the infected group under these specific conditions. In this case, the noise level was set to 60%, indicating a relatively high level of uncertainty or variability in the data. Additionally, the vaccination rate was set to 10%, representing a larger proportion of the population that has been vaccinated compared to the previous scenario. We can observe the relationship between the nonlinear incidence rate and the infected group by investigating the figure. The figure depicts how well the nonlinear incidence rate, $g(S, I)$, aligns with the actual data of the infected group. It is evident from the figure that with a higher noise level and a higher vaccination rate, the data fitting for both the nonlinear incidence rate and the infected group is poor. This indicates that the observed data points deviate significantly from the model's predictions, suggesting a lack of accuracy in terms of capturing the underlying dynamics of the virus spread. The figure emphasizes the importance of considering the combined effect of noise levels and vaccination rates on the accuracy of the nonlinear incidence rate. It highlights that higher levels of noise and higher vaccination rates can introduce more uncertainties and complexities into the modeling process, making it challenging to accurately capture the dynamics of infectious diseases such as COVID-19.

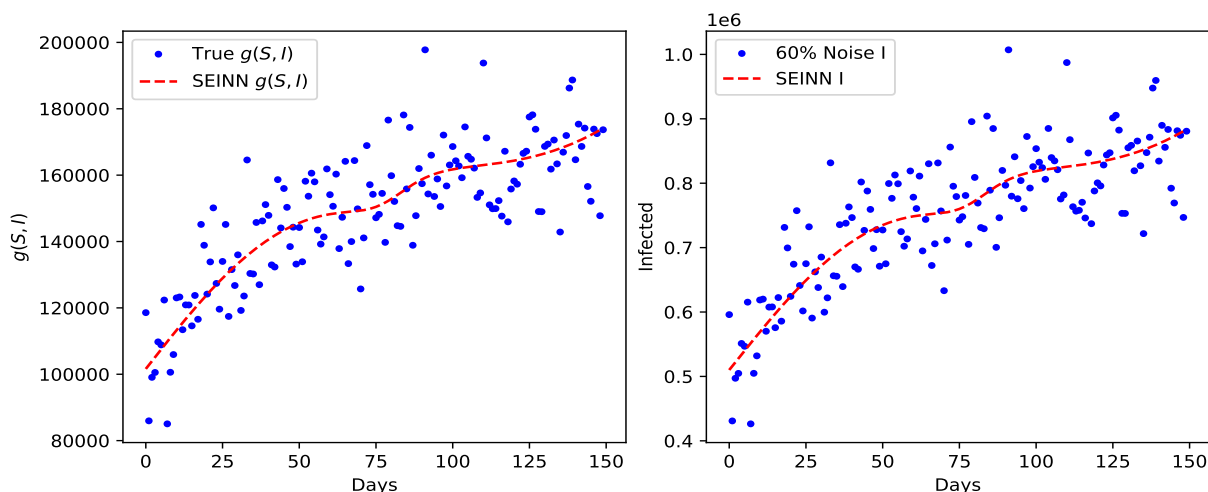


Figure 9. Nonlinear incidence rate for 60% noise level with 10% vaccination rate. The left figure shows how the true incidence rate (blue dotted line) compares with the predicted incidence rate (red broken line). The right figure shows the results for incidence rate at 60% noise level.

Table 2 presents the error metrics for the nonlinear incidence rate under different noise levels and vaccination rate combinations. The table provides insights into the accuracy and performance of the stochastic model in terms of capturing the dynamics of the nonlinear incidence rate under varying conditions. The noise levels considered in the table are 5%, 10%, 30% and 60%, representing different degrees of uncertainty or variability in the data. Additionally, two vaccination rates examined: 1% and 10%. These vaccination rates indicate the proportion of the population that has been vaccinated. For a noise level of 5%, both the 1% and 10% vaccination rates resulted in relatively low values of RMSE and MAPE. This suggests that the models performed reasonably well in terms of fitting the nonlinear incidence rate under these conditions. As the noise level increases to 10%, the RMSE and MAPE values

also increase for both vaccination rates. This indicates that the model's accuracy in terms of capturing the nonlinear incidence rate decreases as the noise level in the data increases. The trend continues as the noise level further increases to 30% and 60%. The RMSE and MAPE values became significantly higher, suggesting a larger discrepancy between the predicted values and the actual data. This indicates that the models struggled to accurately capture the nonlinear incidence rate under conditions of higher noise levels. Comparing the two vaccination rates, it can be observed that the higher vaccination rate of 10% generally results in slightly higher RMSE and MAPE values compared to the lower vaccination rate of 1%. This suggests that higher vaccination rates may introduce additional complexities or uncertainties into the modeling process, leading to slightly decreased accuracy in nonlinear incidence rate prediction.

Table 2. Error metrics for nonlinear incidence rate.

Model comparison for $v = 1\%$, $v = 10\%$ with efficacy rate ($\eta = 94\%$).			
Noise Level	Vaccination	RMSE	MAPE
5%	1%	6544	0.0087
	10%	5470	0.0073
10%	1%	102,266	0.0136
	10%	10,241	0.0136
30%	1%	29,846	0.0395
	10%	30,155	0.0395
60%	1%	59,618	0.0785
	10%	59,750	0.0787

3.3. Computational analysis of SEINN

This section thoroughly examines the impacts of perturbations in the model's parameters and the concern of overfitting in the proposed method. Through the implementation of sensitivity analysis, valuable insights can be obtained regarding the proposed method's robustness, generalization ability, and efficiency. The selection of suitable values for the regularization parameter ε is of utmost importance in guaranteeing the dependability and effectiveness of the model in practical scenarios. We consider the scenario in which the noise level is 10% and vaccination rate v is 1%.

3.3.1. 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 technique is a mechanism utilized to comprehend the actions of a model and evaluate its validity. In this work, we varied some parameters in the SEINN to observe the effects on the error metrics.

Table 3 displays the effects of altering the number of neurons and layers in an SEINN while maintaining a constant regularization parameter of $\varepsilon = 1e - 1$ for the error metrics. The presented tabular data display the numerical values of various performance metrics including the RMSE, MAPE, EV

and REL, for every possible combination of layers and neurons. Upon conducting a comparative analysis between the model containing 32 neurons and the model containing 64 neurons while keeping the number of layers constant, it was observed that the former exhibited superior performance. The optimal results achieved by a model comprising five layers and 32 neurons are presented in the fifth row of the table. The RMSE was computed to be 1912, the MAPE was determined to be 0.00254 and the REL was calculated to be 0.00131.

Table 3. Effect of $\varepsilon = 1e - 1$ on the SEINN by 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

Table 4. Effect of $\varepsilon = 1e - 3$ on the SEINN by 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

As shown in Table 4, reducing the regularization parameter to $\varepsilon = 1e - 3$ can potentially prevent overfitting and consequently improve the performance on error metrics. Alternatively, modifying the number of neurons or introducing an extra layer may serve as a countermeasure to the reduction in regularization and potentially amplify the overall efficacy. The results suggest that attaining maximum efficiency under the condition of the SEINN framework necessitates a careful equilibrium between the regularization parameter and the number of neurons and layers. The fifth row indicates that the optimal error metric values are achieved by combining four layers and 32 neurons.

Figure 10 illustrates the influence of the regularization parameter $\varepsilon = 1e - 1$ on the SEINN algorithm. The model's performance was assessed by using the RMSE and MAPE as the chosen perfor-

mance metrics. According to the RMSE graph, the most favorable arrangement of layers and neurons is attained by using five layers and 32 neurons, yielding a value of 1277. The configuration exhibiting the least favorable outcome comprises five layers, each comprising 64 neurons, resulting in a numerical output of 6371. Regarding the MAPE graph, it is evident that optimal performance is achieved by using a configuration consisting of 5 layers and 32 neurons, which yielded a MAPE value of 0.00160. The configuration demonstrating the least favorable outcome comprises five layers and 32 neural units, yielding a documented score of 0.00827.

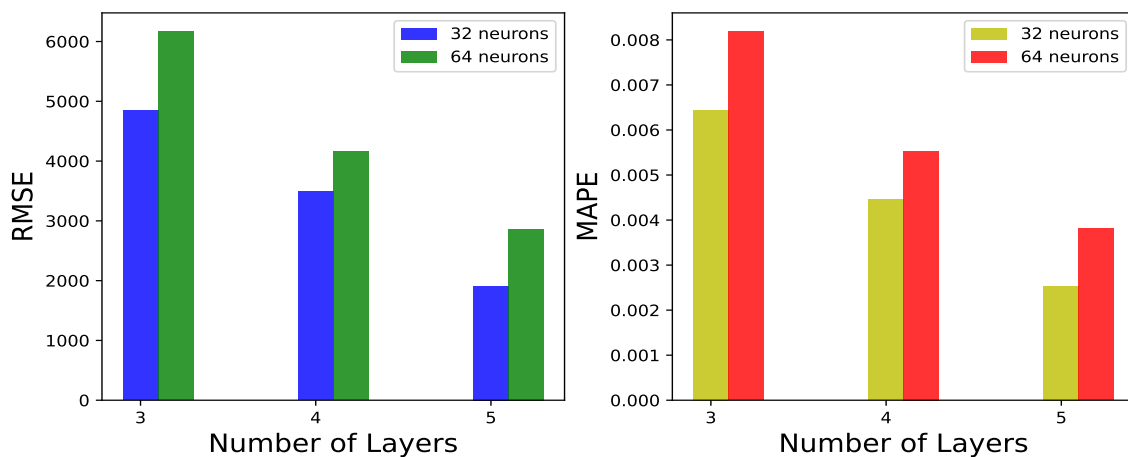


Figure 10. Graphical representation of effects of regularization parameter $\varepsilon = 1e - 1$ by using the RMSE and MAPE. The left figure depicts the effect of regularization on RMSE while the right figure shows the effect of regularization on MAPE.

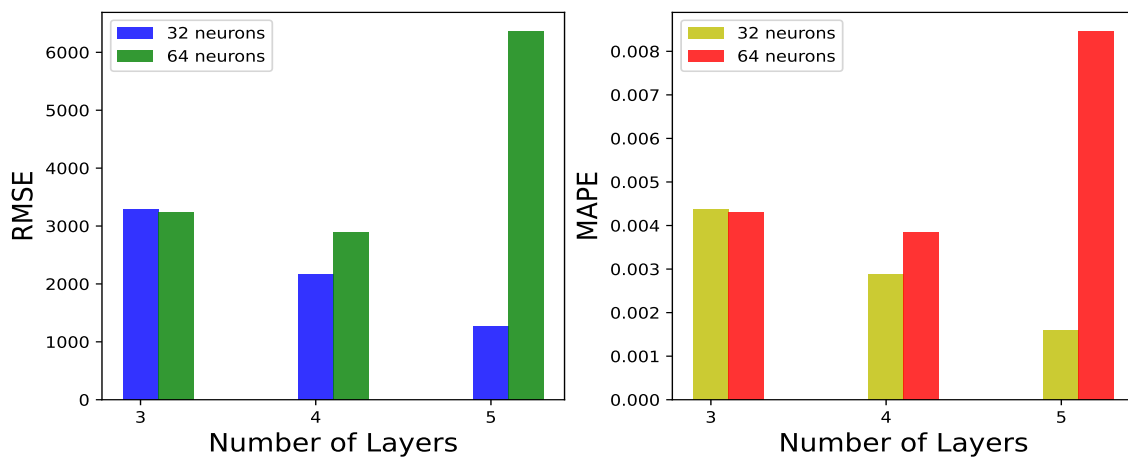


Figure 11. Graphical representation of effects of regularization parameter $\varepsilon = 1e - 3$ by using the RMSE and MAPE. The left figure depicts the effect of regularization on RMSE while the right figure shows the effect of regularization on MAPE.

The results depicted in Figure 11 indicate that a decrease in the regularization parameter value from $\varepsilon = 1e - 1$ to $\varepsilon = 1e - 3$ leads to notable enhancements in the performance of the SEINN model. The

graph showcases the most favorable configurations demonstrating exceptional performance. The configuration consists of four layers, each of which comprises 34 neurons. This result is consistent with the findings presented in Table 4. The MAPE graph provides a performance evaluation that is comparable to that of the RMSE graph. The diagram underscores the importance of selecting appropriate hyperparameters for the SEINN model to attain optimal performance and the need for the meticulous calibration of regularization parameters.

3.3.2. Overfitting analysis

Overfitting analysis refers to assessing a deep learning model's ability to perform on which it has not been trained. The aim was to determine whether the model demonstrates overfitting by assessing its performance on the training dataset compared to its performance on the validation or test dataset. The examination of overfitting can function as a method to guide the selection of appropriate hyperparameters for a given model and to assess the model's ability to generalize. The present study entailed assessing the influence of adjusting the regularization parameter ε and selecting appropriate hyperparameters on the phenomenon of overfitting. The input data for the SEINN was partitioned into a training set comprising 80% of the data and a validation set comprising 20% of the data. The number of layers was fixed at four, while the number of epochs varied between 30,000 and 60,000.

Figure 12 illustrates the impact of the regularization parameter $\varepsilon = 1e - 1$ on reducing overfitting in the SEINN model. The graph on the right-hand side compares the training and validation losses, where 30,000 epochs and four layers were utilized. Upon logarithmic transformation of the epoch count, it is evident that the training and validation losses exhibit dissimilarity in the initial stages, but eventually converge to a similar value after a certain number of epochs. In contrast, the graph on the left was generated by using 60,000 epochs and implementing four layers. Evidently, the training and validation losses exhibit identical values.

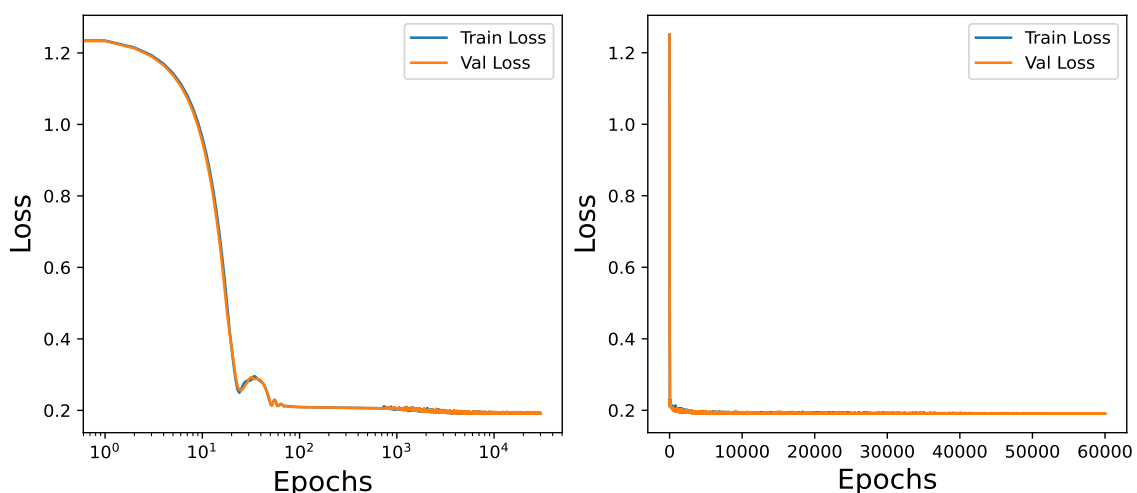


Figure 12. Graphical representation of effects of regularization parameter $\varepsilon = 1e - 1$ preventing overfitting using epochs of 30,000 and 60,000. The left figure shows the effect of regularization preventing overfitting using epoch of 30,000. The right figure shows the effect of using epochs 60,000 for $\varepsilon = 1e - 1$.

The impact of the regularization parameter $\varepsilon = 1e - 3$ in terms of mitigating overfitting in the SEINN model is depicted in Figure 13. The graph on the right-hand side compares the training and validation losses, which were obtained by utilizing 30,000 epochs and four layers. After applying a logarithmic transformation to the epoch count, it becomes apparent that the training and validation losses display dissimilarities initially but eventually converge to a comparable value following a specific number of epochs. By way of comparison, the graph depicted on the left was produced through the utilization of 60,000 epochs and the incorporation of four layers. The parity between the losses observed in the training and validation phases is apparent.

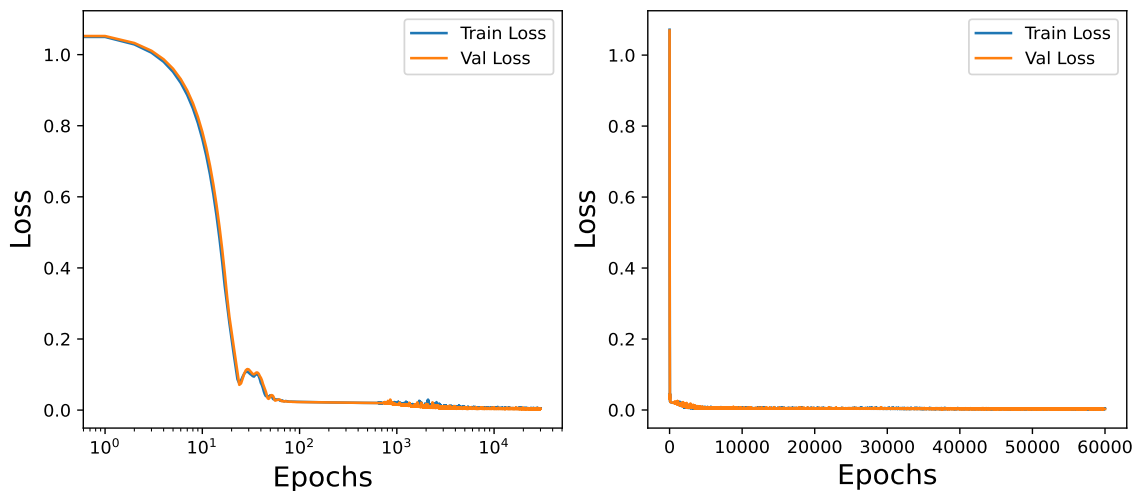


Figure 13. Graphical representation of effects of regularization parameter $\varepsilon = 1e - 3$ preventing overfitting using epochs of 30,000 and 60,000. The left figure shows the effect of regularization preventing overfitting using epoch of 30,000. The right figure shows the effect of using epochs 60,000 for $\varepsilon = 1e - 3$.

Table 5. Analysis of SEINN for different epochs: 30,000, 40,000 and 60,000.

Epochs	Epsilon ε	Loss
30,000	$1e - 1$	1.91×10^{-1}
30,000	$1e - 3$	4.21×10^{-3}
40,000	$1e - 1$	1.91×10^{-1}
40,000	$1e - 3$	2.68×10^{-3}
60,000	$1e - 1$	1.91×10^{-1}
60,000	$1e - 3$	3.40×10^{-3}

An analysis of data-driven simulations for training with varying numbers of epochs (30,000, 40,000, 60,000) and diverse values of the regularization parameter ε is presented in Table 5. The objective of the table was to assess the influence of altering the number of epochs on the loss function, which functions as a metric of the model's ability to conform to the data. The presented table elucidates

the influence of the factors above the loss function, which indicates the model's adequacy in terms of capturing the data. The results show an ideal quantity of epochs, after which additional training fails to enhance the model's efficacy substantially. In addition, selecting a suitable regularization parameter value holds significant importance in mitigating overfitting and improving the model's capacity for generalization.

3.4. Discussion

The present research examines environmental noise's effects on transmission and nonlinear incidence rates. The term "environmental noise" encompasses various external factors, including human mobility, societal factors, and demographic characteristics, that can impact the transmission of diseases. The initial phase of the study aimed to demonstrate the supremacy of stochastic models compared to deterministic models. According to the findings, both models' RMSE values demonstrate that the stochastic models perform better than the deterministic models.

The significance of examining models featuring nonlinear incidence rates within a vaccination regimen was considered. Recognizing the importance of complying with a vaccination regimen and the personal response to interventions aimed at reducing the spread of the COVID-19 pathogen is of utmost importance. The implementation of an effective vaccination program is of utmost importance in the management of the current pandemic and in the mitigation of its societal consequences. Vaccinations are a highly effective measure for mitigating infectious disease transmission, reducing symptom severity and lowering the likelihood of hospitalization and mortality. To effectively control the pandemic, it is necessary to undertake measures beyond implementing a vaccination regimen.

The model's efficacy in accurately predicting nonlinear incidence rates under varying parameter combinations and noise levels is demonstrated through data-driven simulations. This provides valuable insights into disease transmission dynamics and the influence of vaccination rates. The study's results demonstrate the model's effectiveness in accurately predicting nonlinear incidence rates under different parameter combinations and noise levels. The results above provide noteworthy insights into the mechanisms of disease propagation and the impact of immunization coverage.

The results of the computational analysis of the SEINN indicate that the method is characterized by robustness and effectiveness. By manipulating diverse components of the approach, we acquire a deeper understanding of the means to mitigate overfitting. Optimizing the selection of hyperparameters and regularization parameters can effectively mitigate the issue of overfitting. To avoid fluctuations in the performance of a model, it is necessary to exercise cautious intuition during experimentation to ascertain the extent to which the regularization parameter should be decreased.

The present study's research methodology examines environmental noise's impact on the nonlinear incidence rate. There are some limitations to the present study. The first limitation of the study is the key assumption of stochastic models that the host population is homogeneously mixed. This means that there is a need to consider contact heterogeneity in the population. Thus, it is more appropriate to formulate disease models in complex networks [33, 34]. Furthermore, the current work can be extended to network epidemic models [35]. The second limitation of the study is the exclusion of social determinants of health (such as poverty, employment, and access to health care), which are important factors in a comprehensive analysis of epidemic models.

4. Conclusions

A data-driven methodology has been considered for the construction of a model that examines the influence of stochasticity on the transmission of COVID-19. Inspired by the PINN, a SEINN was created to learn about epidemiological parameters and nonlinear incidence rates for four different noise levels in a vaccination regime. The deterministic model frequently fails to account for environmental noise, which encompasses factors such as human mobility, human response to viral transmission and demographic variables. Using data-driven simulations and error metrics has shown that stochastic models are better for analyzing epidemic models. This is because they can deal with the environmental noise that is part of the model. The proposed method has demonstrated the ability to learn diverse forms of nonlinear incidence rates under varied noise levels and vaccination rates. Additionally, we have illustrated the significance of computational analysis within deep learning models. The importance of selecting appropriate hyperparameters was demonstrated through the sensitivity analysis of the proposed methodology, and the use of regularization can aid in mitigating overfitting. An area of potential future research involves the integration of the supplementary time-varying parameters and social determinants of health that directly impact the transmission of infectious diseases. A possible extension of this work is to compare Bayesian and non-Bayesian methods to stochastic epidemic models.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Code and data availability

Code and data for the work can be found at **github**.

Acknowledgments

There was no source of funding for this article.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. S. Ghamizi, R. Rwemalika, M. Cordy, L. Veiber, T. F. Bissyandé, M. Papadakis, et al., Data-driven simulation and optimization for COVID-19 exit strategies, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, (2020), 3434–3442. <https://doi.org/10.1145/3394486.3412863>

2. N. Dalal, D. Greenhalgh, X. Mao, A stochastic model of AIDS and condom use, *J. Math. Anal. Appl.*, **325** (2007), 36–53. <https://doi.org/10.1016/j.jmaa.2006.01.055>
3. W. O. Kermack, A. G. McKendrick, A contribution to the mathematical theory of epidemics, in *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, **115** (1927), 700–721. <https://doi.org/10.1098/rspa.1927.0118>
4. T. A. Biala, Y. O. Afolabi, A. Q. M. Khaliq, How efficient is contact tracing in mitigating the spread of COVID-19? a mathematical modeling approach, *Appl. Math. Modell.*, **103** (2022), 714–730. <https://doi.org/10.1016/j.apm.2021.11.011>
5. K. M. Furati, I. O. Sarumi, A. Q. M. Khaliq, Fractional model for the spread of COVID-19 subject to government intervention and public perception, *Appl. Math. Modell.*, **95** (2021), 89–105. <https://doi.org/10.1016/j.apm.2021.02.006>
6. E. Kharazmi, M. Cai, X. Zheng, Z. Zhang, G. Lin, G. E. Karniadakis, Identifiability and predictability of integer- and fractional-order epidemiological models using physics-informed neural networks, *Nat. Comput. Sci.*, **1** (2021), 744–753.
7. J. Long, A. Q. M. Khaliq, K. M. Furati, Identification and prediction of time-varying parameters of COVID-19 model: a data-driven deep learning approach, *Int. J. Comput. Math.*, **98** (2021), 1617–1632. <https://doi.org/10.1080/00207160.2021.1929942>
8. K. D. Olumoyin, A. Q. M. Khaliq, K. M. Furati, Data-driven deep-learning algorithm for asymptomatic COVID-19 model with varying mitigation measures and transmission rate, *Epidemiologia*, **2** (2021), 471–489. <https://doi.org/10.3390/epidemiologia2040033>
9. T. K. Torku, A. Q. M. Khaliq, K. M. Furati, Deep-data-driven neural networks for COVID-19 vaccine efficacy, *Epidemiologia*, **2** (2021), 564–586. <https://doi.org/10.3390/epidemiologia2040039>
10. F. A. Rihan, U. Kandasamy, H. J. Alsakaji, N. Sottocornola, Dynamics of a fractional-order delayed model of COVID-19 with vaccination efficacy, *Vaccines*, **11** (2023), 758. <https://doi.org/10.3390/vaccines11040758>
11. M. Rafiq, A. Raza, M. U. Iqbal, Z. Butt, H. A. Naseem, M. A. Akram, et al., Numerical treatment of stochastic heroin epidemic model, *Adv. Differ. Equations*, **2019** (2019), 1–17.
12. Z. T. Win, M. A. Eissa, B. Tian, Stochastic epidemic model for COVID-19 transmission under intervention strategies in China, *Mathematics*, **10** (2022), 3119. <https://doi.org/10.3390/math10173119>
13. X. Mao, G. Marion, E. Renshaw, Environmental Brownian noise suppresses explosions in population dynamics, *Stochastic Processes Appl.*, **97** (2002), 95–110. [https://doi.org/10.1016/S0304-4149\(01\)00126-0](https://doi.org/10.1016/S0304-4149(01)00126-0)
14. A. Miao, X. Wang, T. Zhang, W. Wang, B. G. Sampath Aruna Pradeep, Dynamical analysis of a stochastic SIS epidemic model with nonlinear incidence rate and double epidemic hypothesis, *Adv. Differ. Equations*, **2017** (2017), 1–27.
15. D. J. Higham, An algorithmic introduction to numerical simulation of SDEs, *SIAM Rev.*, **43** (2001), 525–546. <https://doi.org/10.1137/S0036144500378302>

16. J. O’Leary, J. A. Paulson, A. Mesbah, Stochastic physics-informed neural networks (SPINN): A moment-matching framework for learning hidden physics within SDEs, preprint, arXiv:2109.01621.
17. Y. Cai, Y. Kang, W. Wang, A stochastic SIRS epidemic model with nonlinear incidence rate, *Appl. Math. Comput.*, **305** (2017), 221–240. <https://doi.org/10.1016/j.amc.2017.02.003>
18. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, Cambridge, 2016.
19. P. Ren, C. Rao, Y. Liu, J. X. Wang, H. Sun, PhyCRNet: Physics-informed convolutional-recurrent network for solving spatiotemporal PDEs, preprint, arXiv:2106.14103v1.
20. S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci.*, **113** (2016), 3932–3937. <https://doi.org/10.1073/pnas.1517384113>
21. M. Raissi, N. Ramezani, P. Seshaiyer, On parameter estimation approaches for predicting disease transmission through optimization, deep learning and statistical inference methods, *Lett. Biomath.*, **6** (2019), 1–26. <https://doi.org/10.30707/LiB6.2Raissi>
22. K. Hornik, M. Stinchcombe, H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *Neural Networks*, **3** (1990), 551–560. [https://doi.org/10.1016/0893-6080\(90\)90005-6](https://doi.org/10.1016/0893-6080(90)90005-6)
23. L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.*, **3** (2021), 218–229.
24. A. Zeroual, F. Harrou, A. Dairi, Y. Sun, Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study, *Chaos, Solitons Fractals*, **140** (2020), 110121. <https://doi.org/10.1016/j.chaos.2020.110121>
25. A. Dairi, F. Harrou, A. Zeroual, M. M. Hittawe, Y. Sun, Comparative study of machine learning methods for COVID-19 transmission forecasting, *J. Biomed. Inf.*, **118** (2021), 103791. <https://doi.org/10.1016/j.jbi.2021.103791>
26. L. Wang, T. Xu, T. Stoecker, Y. Jiang, K. Zhou, Machine learning spatiotemporal epidemiological model to evaluate Germany-county-level COVID-19 risk, *Mach. Learn. Sci. Technol.*, **2** (2021), 035031. <https://doi.org/10.1088/2632-2153/ac0314>
27. S. Han, L. Stelz, H. Stoecker, L. Wang, K. Zhou, Approaching epidemiological dynamics of COVID-19 with physics-informed neural networks, preprint, arXiv:2302.08796.
28. P. E. Kloeden, E. Platen, SDEs, in *Numerical Solution of SDEs*, Springer, (1992), 103–160.
29. G. Chowell, Fitting dynamic models to epidemic outbreaks with quantified uncertainty: A primer for parameter uncertainty, identifiability, and forecasts, *Infect. Dis. Modell.*, **2** (2017), 379–398. <https://doi.org/10.1016/j.idm.2017.08.001>
30. J. Kukačka, V. Golkov, D. Cremers, Regularization for deep learning: A taxonomy, preprint, arXiv:1710.10686.
31. E. Gobet, J. P. Lemor, X. Warin, A regression-based Monte Carlo method to solve backward SDEs, *Ann. Appl. Probab.*, **15** (2014) 2172–2202. <https://doi.org/10.1214/105051605000000412>
32. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, arXiv:1412.6980.

33. B. Alain, M. Barthelemy, V. Alessandro, *Dynamical Processes on Complex Networks*, Cambridge University Press, Cambridge, 2008.
34. Y. Wang, Z. Wei, J. Cao, Epidemic dynamics of influenza-like diseases spreading in complex networks, *Nonlinear Dyn.*, **101** (2020), 1801–1820. <https://doi.org/10.1007/s11071-020-05867-1>
35. B. Frank, N. Peter, Network epidemic models with two levels of mixing, *Math. Biosci.*, **212** (2008), 69–87. <https://doi.org/10.1016/j.mbs.2008.01.001>

Appendix

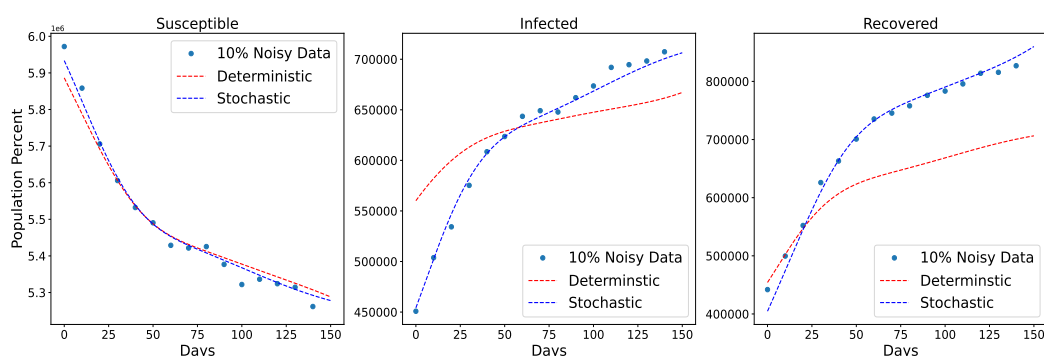


Figure A1. Comparison of EINN and SEINN for noisy data at 10% noise level. The left figure shows the results for deterministic and stochastic models at 10% noise level for susceptible group. The middle and right figures show the results for infected and recovered groups, respectively.

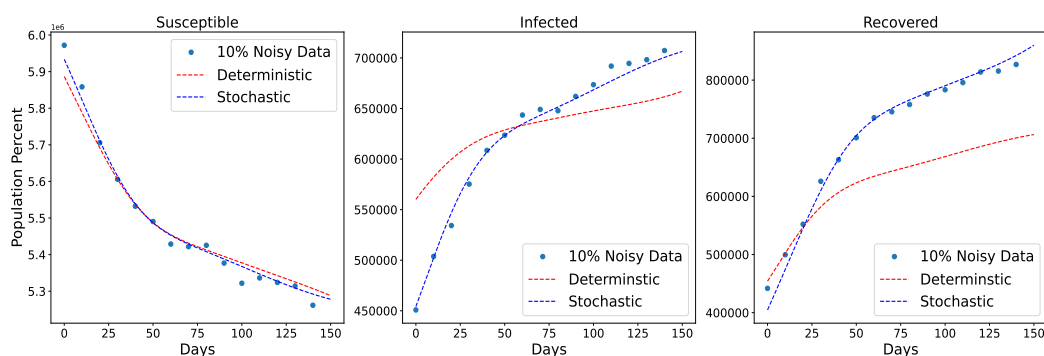


Figure A2. Comparison of EINN and SEINN for noisy data at 30% noise level. The left figure shows the results for deterministic and stochastic models at 30% noise level for susceptible group. The middle and right figures show the results for infected and recovered groups, respectively.

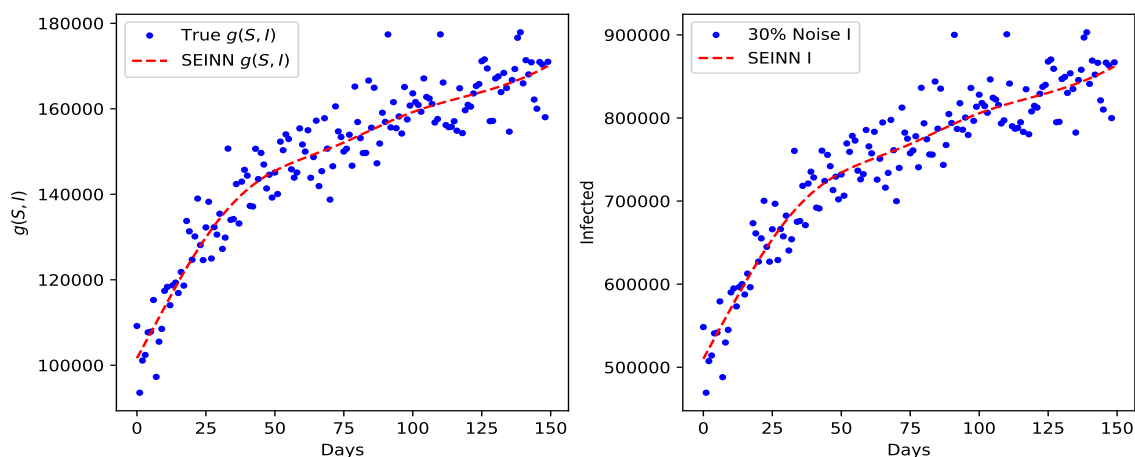


Figure A3. Nonlinear incidence rate for 30% noise level with 1% vaccination rate. The left figure shows how the true incidence rate (blue dotted line) compares with the predicted incidence rate (red broken line). The right figure shows the results for incidence rate at 30% noise level.



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)