Applied Computing and Intelligence

https://www.aimspress.com/journal/aci

Applied Computing and Intelligence
4(2): 195–208.
DOI: 10.3934/aci.2024012
Received: 24 August 2024
Revised: 11 November 2024
Accepted: 13 November 2024
Published: 18 November 2024

Research article

A physics-informed neural network model for social media user growth

Lingju Kong[1], **Ryan Z. Shi**[2] **and Min Wang**[3,*]

[1] Department of Mathematics, University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA

[2] Menaul School Qingdao, Qingdao 266000, China

[3] Department of Mathematics, Kennesaw State University, Marietta, GA 30060, USA

* **Correspondence:** Email: min.wang@kennesaw.edu.

Academic Editor: Pasi Fränti

Abstract: In this paper, a physics-informed neural network model is proposed to predict the growth of online social network users. The number of online social network users is modeled by a stochastic process and the associated Kolmogorov forward equation is derived. Then, a physics-informed neural network model is built based on the Kolmogorov forward equation and trained using real-world data. By combining mathematical modeling with machine learning, our approach provides a practical and interpretable methodology that harnesses the strengths of both physical laws and advancements in machine learning, while minimizing the opacity in machine learning models.

Keywords: physics-informed neural network; Kolmogorov forward equations; online social networks; compartment models; Markov process

1. Introduction

Social media, or online social networks (OSNs), have greatly transformed our lives by offering vital platforms for socialization, entertainment, and commerce [17,20,26]. Accurate predictive models for the number of OSN users are in high demand because the number of users is a key metric of an OSN's popularity, which directly influences its success, sustainability, and strategic planning. To meet this demand, various compartment models, which are dynamic equation models based on the compartment modeling approach, have been developed, see [5,6,11,14,16,18,23,24] and the references therein. The basic principle behind these models is to divide the total population into multiple compartments, such as current users and potential users, and use either differential or difference equations to describe the instantaneous changes in these compartments due to the interactions among them. This modeling approach is well-developed and has been extensively applied to study problems

in various areas, particularly in mathematical epidemiology, see for example [3, 7, 9, 10, 19, 25, 28]. While the compartment models are easy to interpret and analyze, they also have some limitations:

(1). The assumptions of the compartment model are often simplified so that the mathematical analysis can be conducted. The interactions among compartments are usually assumed to be either constants or functions with 'nice' properties to meet the needs for the qualitative analysis.
(2). It is difficult to interpret the predicted values of the deterministic compartment models. The compartment models formulated as deterministic dynamic equations are excellent for the qualitative analysis to reveal the dynamical trends of the underlying problem over time. However, it is challenging for the model to accurately predict specific values at individual time points.
(3). It may be difficult to calibrate the models with real data. Many existing models divide the total populations into multiple compartments so that they can interpret the role of each compartment and better capture its impact to the whole model. However, in practice, many of those compartments cannot be measured. Although some works simply assume that the initial values of those compartments are 0, it is arguable whether this assumption is sensible.

In this paper, we will leverage advancements in both mathematical modeling and machine learning (ML) to develop a new model for OSN user growth that addresses the aforementioned limitations.

A natural approach to address the aforementioned limitation 2 is the use of stochastic models. Stochastic differential equation (SDE) compartment models, which predict probability distributions rather than specific values at individual time points, offer a significant advantage in addressing limitation 2, see for example [1, 4, 17]. In these models, each compartment is viewed as a stochastic process and the probability of a specific state at a particular time is predicted, which is much more reasonable than a single deterministic value. However, the real-world applications of SDE models typically require calibrating the models to real data using the Maximum Likelihood Estimation (MLE). Previous experiments have revealed that fitting SDE models to real data using MLE typically requires significant computational resources and is very time-consuming, even for simplified models with constant coefficients. Various strategies have been explored to overcome this challenge, such as scanning all parameter values on a grid defined in a pre-set region [1], or searching the neighborhood around an optimal parameter set for the associated deterministic model [17]. When the number of model parameters increases, the task to calibrate the model efficiently becomes even more challenging. Clearly, this challenge is one of the barriers to resolving the aforementioned limitation 1. Since the model calibration process is essentially an optimization problem, we will explore using optimization techniques developed in the ML area to tackle this challenge.

ML, particularly neural networks (NNs), is known for its capability to approximate unknown functions from observed data by adjusting a large number of network parameters, such as weights and biases, through the training process. As a rapidly growing area, numerous technologies have been developed for the intensive computations in the training process of ML models. Since the training process involves solving an extremely high-dimensional optimization problem, ML training techniques are ideal solutions to address the aforementioned challenge.

The adoption of NN not only provides an efficient model calibration solution, but also allows us to incorporate more complex assumptions that better reflect reality. Hence, limitation 1 is addressed. Instead of using a black-box NN model to predict the OSN users, we will develop an SDE model for the OSN users and design an NN to implement this SDE model. By doing so, the NN architecture is based

on rigorous mathematical derivation. The only opaque components in our model will be two sub-NNs used to approximate the mathematical functions. In other words, NNs will serve the same purpose as classic numerical approximation methods in our model. Our approach primarily employs NN training algorithms as the optimization solvers. Therefore, our methodology offers a practical and interpretable solution that leverages the strengths of both mathematical modeling and ML advancements while minimizing the opacity typically associated with ML models. In the ML community, this methodology, which encodes mathematical equations models as a component of the NN itself, is known as physics-informed neural networks (PINNs). For more information and advancements in this area, the reader is referred to [8, 12, 21] and the references therein.

The remainder of the paper is organized as follows: in Section 2, we develop the OSN user models; the PINN architecture is presented in Section 3; a case study using real-world data is given as an application in Section 4; and finally, Section 5 contains the conclusions.

2. Stochastic OSN model

In this section, we develop the mathematical models for the users of an OSN product. To address limitation 3 mentioned in Section 1, we assume the entire world population, denoted by N, is divided into two compartments: the current OSN users, denoted by X, and the potential users, denoted by $N - X$. The historical data of both N and X are available online. Hence, the proposed model can be calibrated with real data. Since X is a subset of N, N and X have the same per capita birth and death rate, denoted by Λ and μ respectively. Let α be the probability for a potential user to join the OSN, and η be the probability a current user abandons the OSN product. Then, the relationship between N and X can be described by Figure 1. A summary of the notations used in the models is given in Table 1. All the notations are assumed to be nonnegative throughout this paper.

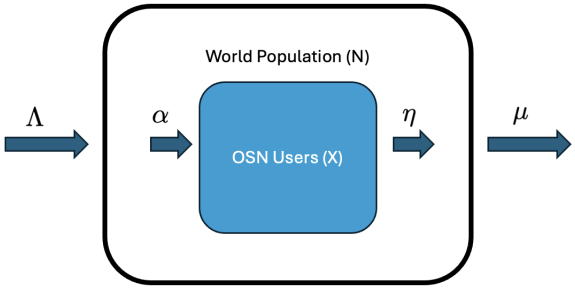


Figure 1. The relationship between the world population N and the OSN users X.

Table 1. List of notations.

Notations	Meaning
N	World population
X	Current OSN users
Λ	Per capita birth rate of N (and X)
μ	Per capita death rate of N (and X)
α	Probability at which potential users join OSN
η	OSN user abandonment probability

2.1. Population model

In this paper, we assume that the world population varies with respect to time t. Therefore, we first need to develop a model for N. By Figure 1 and Table 1, it is easy to see that the growth rate of N can be described by

$$\frac{dN}{dt} = \Lambda N - \mu N, \quad N(0) = N_0, \tag{2.1}$$

where $N_0 > 0$ is the world population at $t = 0$.

It is obvious that model (2.1) has a unique positive solution

$$N(t) = N_0 e^{(\Lambda-\mu)t}, \quad t \geq 0. \tag{2.2}$$

Hence, the values of Λ and μ can be determined by fitting (2.2) with real-world data.

2.2. Stochastic OSN model

Now, we are ready to develop the OSN user model. In the sequel, we assume the number of OSN users is a Markov process denoted by $X(t)$, $t \geq 0$. Due to the application needs, we only consider a finite horizon $0 \leq t \leq T$ in this paper. It is clear that for any $t \in [0, T]$, we have $0 \leq X(t) \leq N(t)$, where $N(t)$ is defined by Eq (2.2). The change in X is shown in Figure 2.

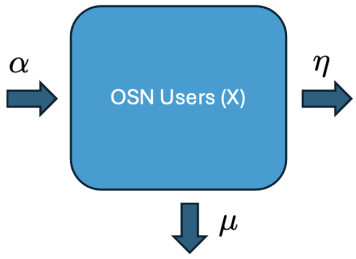


Figure 2. The diagram of the changes in the OSN users X.

For any $t \in [0, T]$ and $\Delta t > 0$, let $\Delta X(t) = X(t + \Delta t) - X(t)$. Then, the possible changes in X and the corresponding probabilities at time t for a short time step Δt are presented in Table 2, where $\lambda > 0$ is the unit change, and the probabilities p_1 and p_2 : $[0, T] \times \mathbb{R} \to \mathbb{R}^+$ are defined by

$$p_1(t, x) = \begin{cases} \alpha(t, x)\Delta t, & (t, x) \in [0, T] \times [0, N(t)], \\ 0, & otherwise, \end{cases} \tag{2.3}$$

and

$$p_2(t, x) = \begin{cases} \eta(t, x)\Delta t + \mu\Delta t, & (t, x) \in [0, T] \times [0, N(t)], \\ 0, & otherwise, \end{cases} \tag{2.4}$$

with α and η two nonegative functions on $\mathbb{R}^2$, and $\Delta t > 0$ sufficiently small such that $0 \leq p_1(t, x) + p_2(t, x) \leq 1$.

Table 2. Possible changes in X with the corresponding probabilities.

Change	Probability
$\Delta X(t) = \lambda$	$p_1(t, x)$
$\Delta X(t) = -\lambda$	$p_2(t, x)$
$\Delta X(t) = 0$	$1 - p_1(t, x) - p_2(t, x)$

We will apply a standard modeling process described in [1, Chapter 5] to establish the Kolmogorov forward equation (KFE) model for X. Let $p(t, x)$ be the probability for $X(t) = x$. By Table 2, it is clear that for any $t > 0$, $p(t + \Delta t, x) = 0$ when $x < 0$ or $x > N(t)$. When $0 \le x \le N(t)$, we have

$$p(t + \Delta t, x) = p(t, x)(1 - p_1(t, x) - p_2(t, x)) + p(t, x - \lambda)p_1(t, x - \lambda) + p(t, x + \lambda)p_2(t, x + \lambda). \quad (2.5)$$

When λ is sufficient small, we have

$$p(t, x - \lambda)p_1(t, x - \lambda) \simeq p(t, x)p_1(t, x) - \frac{\partial}{\partial x}[p(t, x)p_1(t, x)]\lambda + \frac{1}{2}\frac{\partial^2}{\partial x^2}[p(t, x)p_1(t, x)]\lambda^2, \quad (2.6)$$

and

$$p(t, x + \lambda)p_2(t, x + \lambda) \simeq p(t, x)p_2(t, x) + \frac{\partial}{\partial x}[p(t, x)p_2(t, x)]\lambda + \frac{1}{2}\frac{\partial^2}{\partial x^2}[p(t, x)p_2(t, x)]\lambda^2. \quad (2.7)$$

By (2.5)–(2.7), we have

$$\frac{p(t + \Delta t, x) - p(t, x)}{\Delta t} \simeq -\frac{\lambda}{\Delta t}\frac{\partial}{\partial x}[(p_1(t, x) - p_2(t, x))p(t, x)] + \frac{\lambda^2}{2\Delta t}\frac{\partial^2}{\partial x^2}[(p_1(t, x) + p_2(t, x))p(t, x)]. \quad (2.8)$$

By [1, Section 5.1], (2.3), and (2.4), when Δt and λ are sufficient small, the solution of (2.8) can be viewed as a discrete approximation of the continuous Markov process with the KFE

$$\frac{\partial p}{\partial t} = -\lambda\frac{\partial}{\partial x}\left[(\alpha(t, x) - \eta(t, x) - \mu)p\right] + \frac{\lambda^2}{2}\frac{\partial^2}{\partial x^2}\left[(\alpha(t, x) + \eta(t, x) + \mu)p\right], \ 0 < t < T, \ 0 < x < N(t). \quad (2.9)$$

By the problem assumption and Table 2, we propose the following initial and boundary conditions

$$\begin{cases} p(0, x) = \delta_{x_0}(x), & 0 < x < N_0, \\ p(t, N(t)) = 0, & \int_0^{N(t)} p(t, x)dx = 1, \quad 0 \le t \le T, \end{cases} \quad (2.10)$$

where N is defined by (2.2) with $N(0) = N_0$, $x_0 \in [0, N_0)$ is the number of OSN users at $t = 0$, and $\delta_{x_0}(x) = \delta(x - x_0)$ with δ the Dirac delta function.

In the sequel, we will use model (2.9), (2.10) as our OSN user model. Clearly, model (2.9), (2.10) is an initial-boundary value problem defined on a time-varying domain. The solution $p(t, x \mid x_0)$ represents the conditional probability distribution of the stochastic process X at time t given $X(0) = x_0$.

Remark 2.1. *We would like to make the following comments about our models:*

(1) The purpose of model (2.2) *is to define the upper bound of x. For simplicity, we assume the population N only depends on* Λ *and* μ *in a short period. Model* (2.2) *can be replaced by other more sophisticated population models if the assumptions change.*

(2) In the following sections, we will treat λ *in model* (2.9) *as a parameter and determine its value during the training process. This introduces an additional degree of freedom, which can enhance the model's performance.*

3. PINN OSN model

In this section, we design a PINN model to solve KFE models (2.9) and (2.10) and calibrate the KFE with real-world data.

3.1. Model architecture

The PINN model uses $(x_0, T, \Lambda, \mu, N_0)$ as the input, and outputs the conditional probability distribution $p(t, x \mid x_0)$, $0 \le x \le N(t)$, $0 < t \le T$. Instead of developing a black-box NN to directly generate the conditional probability distribution p, we will develop two sub-NNs, namely α-net and η-net, to approximate the functions α and η in model (2.9), respectively. Then, a PDE-Solver layer is used to solve models (2.9) and (2.10) and output p. The PINN architecture is shown in Figure 3.

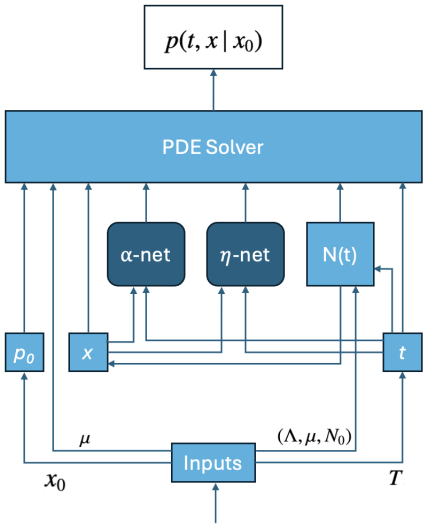


Figure 3. The PINN architecture.

The roles of the blocks shown in Figure 3 are summarized below:

- The p_0 block generates the initial distribution δ_{x_0} from the input x_0.
- The t block generates a discrete time interval $[0, T]_{\mathbb{Z}} := \{0, \Delta t, 2\Delta t, \cdots, T\}$ based on the preset Δt to discretize the time interval $[0, T]$.
- The $N(t)$ block calculates the world population by (2.2) for $t \in [0, T]_{\mathbb{Z}}$.
- The x block discretizes the spatial interval $[0, \bar{N}]$ as a discrete interval $[0, \bar{N}]_{\mathbb{Z}} := \{0, \Delta x, 2\Delta x, \cdots, \bar{N}\}$ based on Δx, where $\bar{N} = \max_{t \in [0,T]} N(t)$.
- For $(t, x) \in [0, T]_{\mathbb{Z}} \times [0, \bar{N}]_{\mathbb{Z}}$, the α-net and η-net approximate $\alpha(t, x)$ and $\eta(t, x)$ respectively.

- The PDE-Solver block numerically solves models (2.9) and (2.10) on the grid $[0, T]_{\mathbb{Z}} \times [0, \bar{N}]_{\mathbb{Z}}$ using the inputs from other blocks. The PDE-Solver block is implemented by the implicit finite difference method [15].

3.2. Forward propagation process

The forward propagation process of the PINN is to calculate the conditional probability distribution $p(t, x \mid x_0)$, $0 \le x \le N(t)$, $0 < t \le T$, by solving models (2.9) and (2.10) based on the input. The algorithm is summarized as follows:

Algorithm 1 Forward propagation.

1: (Optional) Determine values of Λ and μ by calibrating N defined by (2.2) with the real-world population data.

2: Input the initial OSN user number x_0, the initial world population N_0, the length of the time interval T, the population model parameters Λ and μ, the time step Δt, and the spatial step Δx. Initialize the α-net and η-net, and generate the grid $(t, x) \in [0, T]_{\mathbb{Z}} \times [0, \bar{N}]_{\mathbb{Z}}$ and the initial condition δ_{x_0}.

3: For $t = \Delta t, 2\Delta t, 3\Delta t, \ldots, T$:

Calculate $N(t)$, $\alpha(t, x)$, and $\eta(t, x)$, where $x \in [0, N(t)]_{\mathbb{Z}}$.
Calculate $p(t, x \mid x_0)$ by the PDE-Solver block for $x \in [0, N(t)]_{\mathbb{Z}}$ and set $p(t, x \mid x_0) = 0$, when $x \in [N(t) + \Delta x, \bar{N}]_{\mathbb{Z}}$.

4: Output $p(t, x \mid x_0)$, $(t, x) \in [0, T]_{\mathbb{Z}} \times [0, \bar{N}]_{\mathbb{Z}}$.

3.3. Training process

Once the output p is calculated, the PINN is trained by the MLE using the historical OSN user data. Let $\tilde{x}_i$ be the observed OSN user number at time $t_i \in [0, T]$, $i = 1, 2, \ldots, n$, and θ be the parameter vector that consists of the unit change factor λ in (2.9) and all the NN parameters in α-net and η-net. Then, the loss function L is defined by

$$L(\theta) = -1000\left(\sum_{i=1}^{n} \ln(p(t_i, \tilde{x}_i \mid x_0\ ;\ \theta) + 1)\right), \quad \theta \in \Theta, \tag{3.1}$$

where Θ denotes the set of all possible θ values. The goal of the training process is to find $\theta^* \in \Theta$ such that

$$L(\theta^*) = \min_{\theta \in \Theta} L(\theta).$$

This step will be performed by the ML backpropagation algorithms.

Remark 3.1. *The proposed PINN architecture is very flexible, allowing many blocks to be customized:*

(1). By Remark 2.1(a), the $N(t)$ block can use other population models instead of (2.1) *if the assumptions change.*

(2). The PDE-Solver block can be implemented using other algorithms to improve the speed, such as classic numerical methods like the finite element method or the spectral method, or emerging ML-based algorithms. Notably, when ML-based algorithms are used to solve the PDEs, the associated physics-informed loss needs to be added to the loss function defined by (3.1) *due to the opacity of the NNs. For classic numerical PDE solvers, the physics-informed loss is not needed, as the numerical errors are determined by the numerical scheme.*

(3). The α-net and η-net blocks can be implemented using various NN architectures and use other training algorithms to enhance the model performance.

Moreover, the modeling approach presented in Section 2 is suitable for a wide range of problems. The integration of the modeling approach and the approximation capability of NNs allows us to develop data-driven models with more complex assumptions that reflect realities. Therefore, the proposed PINN model has a great application potential in many areas due to the wide applications of compartment modeling.

4. Numerical experiments

In this section, we calibrate our PINN model with Facebook historical user data to demonstrate the application of our model. The OSN user dataset is obtained from [22], which recorded the average Facebook monthly active user (MAU) numbers from 2010 to 2019. The world population data N for the same period is taken from [27]. The Facebook MAU and world population are plotted in Figure 4.

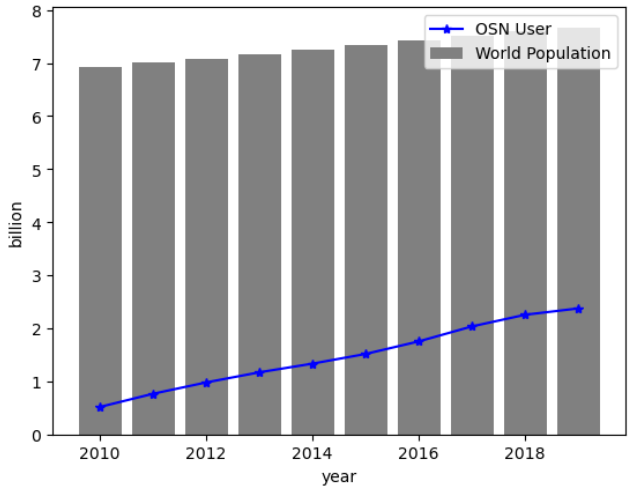


Figure 4. World population v.s. Facebook MAU.

The PINN model is developed following the architecture in Figure 3. The α-net and β-net therein use the same architecture shown in Figure 5. The triple $(t, x, N(t))$ is used as the input for both nets, which slightly differs from the assumption in the KFE (2.9). The inclusion of $N(t)$ allows us to consider not only the absolute value of x, but also its relative value.

The values of parameters Λ and μ are first obtained by calibrating (2.2) with the world population [27]. Then, the PINN model is implemented with PyTorch and trained with the Adam algorithm [13] and the Facebook MAU data [22] on the High Performance Computing Cluster hosted

at the Kennesaw State University [2]. The observations from the first 5 years (2010–2015) are used to train the model. The observations from 2016 to 2019 ($t = 6, \ldots, 9$) are used as the testing data. The model prediction result for 2011–2019 is compared with the real data as well. The experiment results are shown in Figures 6–9.

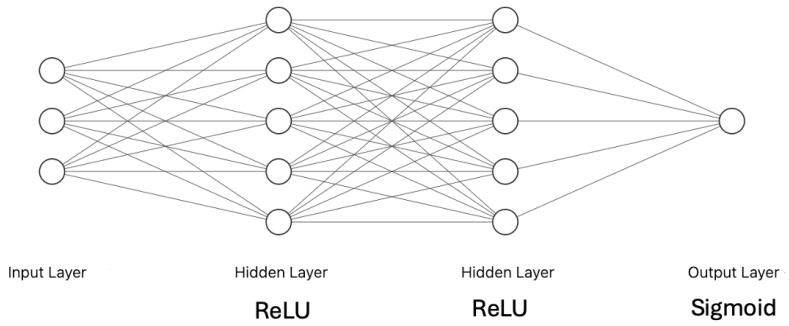


Figure 5. Architecture of α-net and η-net.

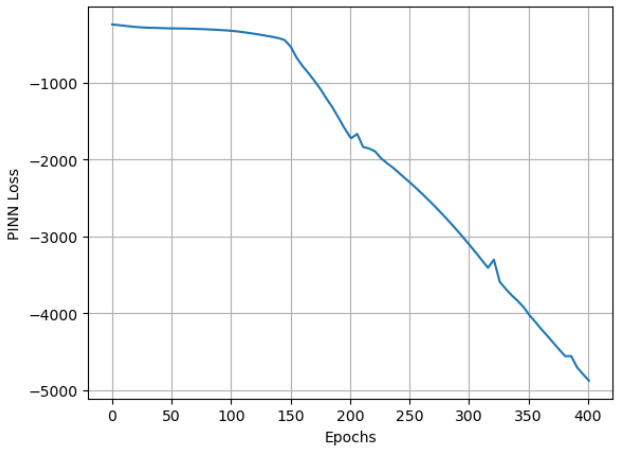


Figure 6. PINN Training Loss.

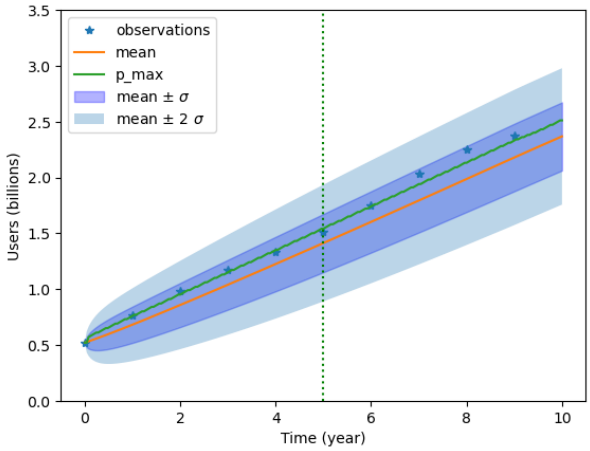


Figure 7. PINN calibration and prediction result.

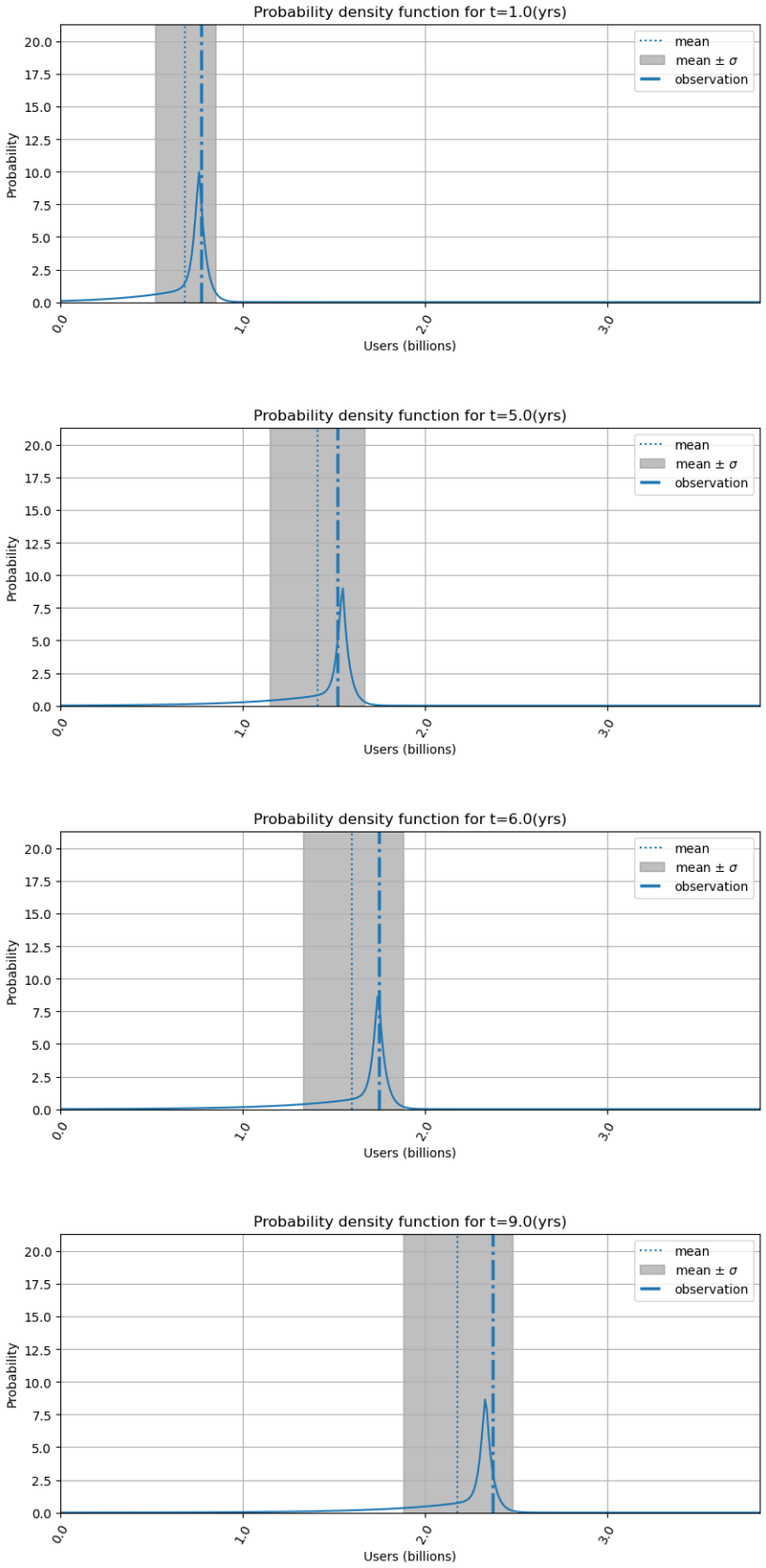
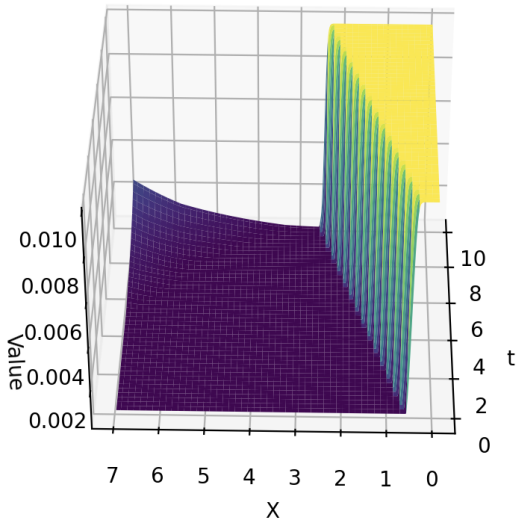
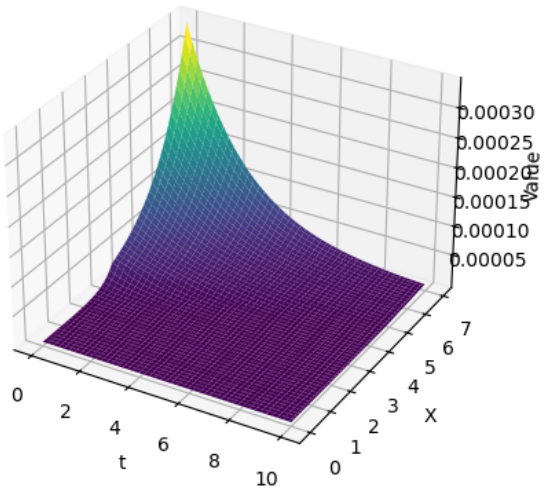


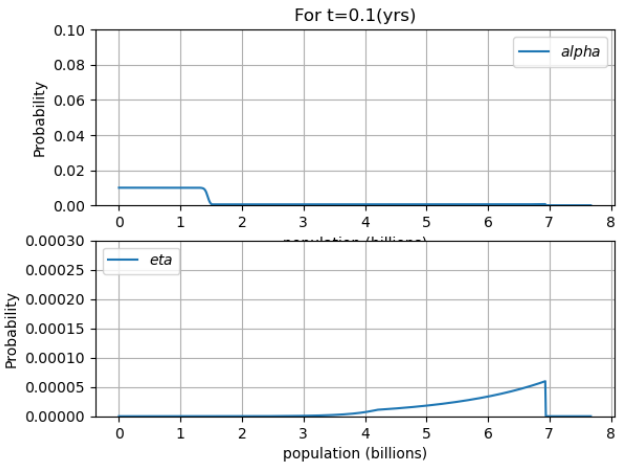
Figure 8. Predicted probability distributions at different time points.



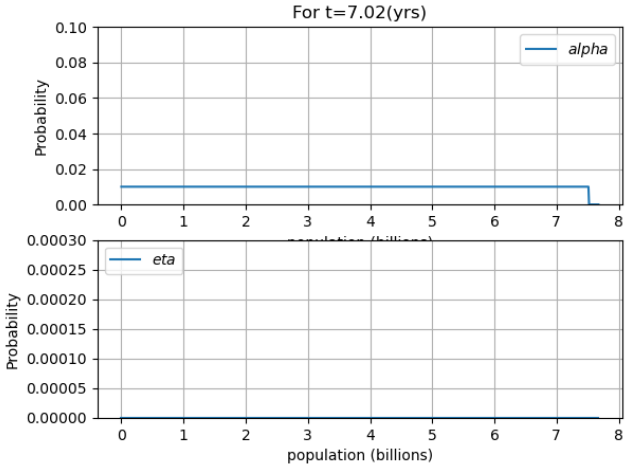
(a) Graph of α-net over $[0, T]_{\mathbb{Z}} \times [0, \bar{N}]_{\mathbb{Z}}$.



(b) Graph of η-net over $[0, T]_{\mathbb{Z}} \times [0, \bar{N}]_{\mathbb{Z}}$.



(c) Cross-sectional graphs of α-net and η-net at $t = 0.1$.



(d) Cross-sectional graphs of α-net and η-net at $t = 7.2$.

Figure 9. Graphs of α-net and η-net.

Figure 7 shows a comparison of the PINN model predictions with the historical MAU data. The observations either on or to the left of the vertical dashed line are used to train the model. At every time t, $p(t, x \mid x_0)$ is predicted by the PINN model. Then $p_{\max}(t) = \operatorname{argmax}_{x \in [0, \bar{N}]_{\mathbb{Z}}} p(t, x \mid x_0)$, along with the mean, mean $\pm\, \sigma$, and mean $\pm\, 2\sigma$ regions for the distribution $p(t, x \mid x_0)$ are plotted. All the observations are around the curve of $p_{\max}$, which is consistent with the expectation of the MLE. Moreover, all the observations fall into the mean $\pm\, \sigma$ region. This shows that our model can produce good prediction results.

Figure 8 further shows the predicted distributions $p(t, x \mid x_0)$ at four time points: $t = 1, 5, 6$, and 9. For the training data at $t = 1$ and $t = 5$, and the first testing data at $t = 6$, the real data are very close to the peaks of the corresponding distributions. The testing data at $t = 9$ is slightly away from the predicted distribution peak, but it is still covered by the mean $\pm\, \sigma$ region. The predicted distributions $p(t, x \mid x_0)$ allow us to consider various possible scenarios with their probabilities, as well as calculate

the statistical information such as the mean, standard deviation, and confidence intervals. This is the major advantage of our model compared to deterministic models.

Additionally, we plot the graphs of the α-net and η-net in Figure 9 to see if they behave as expected. Their graphs over the grid $[0, T]_{\mathbb{Z}} \times [0, \bar{N}]_{\mathbb{Z}}$ are shown in Figure 9a* and 9b. The cross-sectional graphs at $t = 0.1$ and t=7.2 are plotted in Figure 9c and 9d. All the graphs show reasonable patterns for the probabilities of the potential users who join the OSN (α) and the current users who abandon the OSN (η).

The numerical experiment results demonstrate that our work can provide an interpretable predictive model with a reliable performance. The probability outcomes are more practical than the pointwise prediction values from the deterministic models.

5. Conclusions

In this paper, we developed a stochastic PINN model to predict the growth of OSN users by harnessing the strengths of both mathematical modeling and ML advancements. The proposed PINN model architecture is based on rigorous mathematical formulations, with each model block clearly defined. As discussed in previous sections, the modeling approach addresses several well-known limitations in the existing compartment models, thus offering a great flexibility and application potential in many areas.

The experimental results reveal that the proposed PINN model demonstrates a superior performance and provides more useful predictive outcomes. Therefore, our approach offers a practical and interpretable methodology that leverages the strengths of both physical laws and ML advancements while minimizing the opacity typically associated with ML models.

Acknowledgments

M. Wang's research in this paper is supported by the 2024 Kennesaw State University Inspire Summer Scholars Program.

Conflict of interest

The authors declare no conflict of interest.

References

1. E. Allen, *Modeling with Itô stochastic differential equations*, Dordrecht: Springer, 2007. http://dx.doi.org/10.1007/978-1-4020-5953-7
2. T. Boyle, R. Aygun, Kennesaw State University HPC facilities and resources, *Digital Commons Training Materials*, **10** (2021), 1–3.
3. F. Brauer, Mathematical epidemiology: past, present, and future, *Infectious Disease Modelling*, **2** (2017), 113–127. http://dx.doi.org/10.1016/j.idm.2017.02.001

*The graph of α-net is rotated to get a better view.

4. C. Browne, M. Wang, G. F. Webb, A stochastic model of nosocomial epidemics in hospital intensive care units, *Electron. J. Qual. Theory Differ. Equ.*, **2017** (2017), 1–12. http://dx.doi.org/10.14232/ejqtde.2017.1.6
5. J. Cannarella, J. Spechler, Epidemiological modeling of online network dynamics, arXiv: 1401.4208. http://dx.doi.org/10.48550/arXiv.1401.4208
6. R. Chen, L. Kong, M. Wang, Stability analysis of an online social network model, *Rocky Mountain J. Math.*, **53** (2023), 1019–1041. http://dx.doi.org/10.1216/rmj.2023.53.1019
7. S. Chen, J. Shi, Z. Shuai, Y. Wu, Evolution of dispersal in advective patchy environments, *J. Nonlinear Sci.*, **33** (2023), 40. http://dx.doi.org/10.1007/s00332-023-09899-w
8. S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: where we are and what's next, *J. Sci. Comput.*, **92** (2022), 88. http://dx.doi.org/10.1007/s10915-022-01939-z
9. D. Gao, X. Yuan, A hybrid Lagrangian-Eulerian model for vector-borne diseases, *J. Math. Biol.*, **89** (2024), 16. http://dx.doi.org/10.1007/s00285-024-02109-5
10. J. R. Graef, S. Ho, L. Kong, M. Wang, A fractional differential equation model for bike share systems, *J. Nonlinear Funct. Anal.*, **2019** (2019), 23. http://dx.doi.org/10.23952/jnfa.2019.23
11. J. R. Graef, L. Kong, A. Ledoan, M. Wang, Stability analysis of a fractional online social network model, *Math. Comput. Simulat.*, **178** (2020), 625–645. http://dx.doi.org/10.1016/j.matcom.2020.07.012
12. Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, et al., Physics-informed machine learning: a survey on problems, methods and applications, arXiv: 2211.08064. http://dx.doi.org/10.48550/arXiv.2211.08064
13. D. P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv: 1412.6980. http://dx.doi.org/10.48550/arXiv.1412.6980
14. N. Kimmel, L. Kong, M. Wang, Modeling the dynamics of user adoption and abandonment in online social networks, *Math. Method. Appl. Sci.*, in press. http://dx.doi.org/10.1002/mma.10413
15. D. Kincaid, W. Cheney, *Numerical analysis: mathematics of scientific computing*, 3 Eds., Providence: American Mathematical Society, 2002.
16. L. Kong, Modelling the dynamics of product adoption and abandonment, *Proc. R. Soc. A.*, **480** (2024), 20240034. http://dx.doi.org/10.1098/rspa.2024.0034
17. L. Kong, M. Wang, Deterministic and stochastic online social network models with varying population size, *Dynamics of Continuous, Discrete and Impulsive Systems Series A: Mathematical Analysis*, **30** (2023), 253–275.
18. L. Kong, M. Wang, Optimal control for an ordinary differential equation online social network model, *Differ. Equat. Appl.*, **14** (2022), 205–214.
19. B. Ma, C. Li, J. Warner, Structured mathematical models to investigate the interactions between Plasmodium falciparum malaria parasites and host immune response, *Math. Biosci.*, **310** (2019), 65–75. http://dx.doi.org/10.1016/j.mbs.2019.02.005
20. *M. Mohsin, 10 social media statistics you need to know in 2024*, Oberlo, 2024. Available from: `https://www.oberlo.com/blog/social-media-marketing-statistics`.

21. K. Nath, X. Meng, D. J. Smith, G. Karniadakis, Physics-informed neural networks for predicting gas flow dynamics and unknown parameters in diesel engines, *Sci. Rep.*, **13** (2023), 13683. http://dx.doi.org/10.1038/s41598-023-39989-4

22. *E. Ortiz-Ospina, The rise of social media*, Our World In Data, 2019. Available from: `https://ourworldindata.org/rise-of-social-media`.

23. L. Wang, M. Wang, Stability and bifurcation analysis for an OSN model with delay, *Advances in the Theory of Nonlinear Analysis and its Application*, **7** (2023), 413–427. http://dx.doi.org/10.31197/atnaa.1152602

24. L. Wang, M. Wang, Bifurcation analysis for an OSN model with two delays, *Mathematics*, **12** (2024), 1321. http://dx.doi.org/10.3390/math12091321

25. G. Webb, X. E. Zhao, An epidemic model with infection age and vaccination age structure, *Infect. Dis. Rep.*, **16** (2024), 35–64. http://dx.doi.org/10.3390/idr16010004

26. *B. Wong, Top social media statistics and trends of 2024*, Forbes Media LLC., 2024. Available from: `https://www.forbes.com/advisor/business/social-media-statistics/`.

27. *World bank open data*, The World Bank Group, 2024. Available from: `https://data.worldbank.org/indicator/SP.POP.TOTL`.

28. N. Xiao, H. Xu, A. Morani, A. Shokri, H. Mukalazi, Exploring local and global stability of COVID-19 through numerical schemes, *Sci. Rep.*, **14** (2024), 7960. http://dx.doi.org/10.1038/s41598-024-56938-x

AIMS Press