

Physics-Informed Neural Networks for Bio-Nano Digital Twins: A Multi-Model Framework with IoBNT Integration

Mohammad (Behdad) Jamshidi, *Senior Member, IEEE*, Dinh Thai Hoang, *Senior Member, IEEE*, Diep N. Nguyen, *Senior Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, and Majid Ebrahimi Warkiani

Abstract—Digital Twin (DT) technology is transforming biological processes by enabling real-time predictive modeling and optimization. However, implementing DTs at the micro- and nano-scale presents significant challenges in data extraction, transmission, and computation due to the complexity of biological environments. To address these challenges, this work proposes a multi-model Physics-Informed Neural Network (PINN) framework, enhanced by the Internet of Bio-Nano Things (IoBNT), to improve real-time bioprocess monitoring and prediction. The proposed framework integrates IoBNT for efficient biological data acquisition and transmission with advanced deep neural networks (DNNs), including Fully Connected Neural Networks (FCNN), Residual Block Neural Networks (ResBlock NN), and Recurrent Neural Networks (RNN), to enhance predictive accuracy. By incorporating physics-based constraints into the learning process, the PINN architecture ensures biologically plausible and data-efficient predictions. Experimental results demonstrate the framework's effectiveness in monitoring microbial and substrate growth using Monod equations, achieving an average mean absolute error (MAE) of 0.0275 g/L, an R^2 value close to 1, and a root mean squared error (RMSE) of 0.0425 g/L across four of the five proposed architectures. Additionally, the integration of IoBNT enhances data reliability, reducing transmission errors by up to 98%. These findings highlight the potential of IoBNT-powered PINN frameworks for developing high-fidelity, self-adaptive DTs in digital biological and healthcare applications.

Keywords: Internet of Bio-Nano Things (IoBNT), Physics-Informed Neural Networks (PINN), Digital Twin (DT), Bioprocess Monitoring, Real-Time Data Transmission, and AI for IoBNT Systems.

I. INTRODUCTION

A. Motivations

The Digital Twin (DT) paradigm has emerged as a transformative technology for real-time monitoring and optimization across various domains, including bioprocessing, biomanufacturing, and healthcare [1], [2]. By creating virtual representations of systems, DTs enable predictive analytics and adaptive decision-making, improving operational efficiency and performance [1], [3]. However, implementing real-time DTs at the micro- and nano-scale presents critical challenges,

M. B. Jamshidi, D. T. Hoang, and D. N. Nguyen are with the School of Electrical and Data Engineering, University of Technology Sydney, NSW 2007, Australia (e-mail: mohammad.jamshidi@uts.edu.au).

D. Niyato is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798.

M. E. Warkiani is with the School of Biomedical Engineering, University of Technology Sydney, NSW 2007, Australia.

particularly in data extraction, transmission, and computation [4]. Traditional Internet of Things (IoT) frameworks struggle with the complexities of biological environments due to their limited spatial resolution, high latency, and constrained data transmission capabilities [5], [6].

To address these limitations, the Internet of Bio-Nano Things (IoBNT) has emerged as an advanced evolution of IoT, enabling seamless communication in micro- and nanoscale environments [7], [8]. Unlike conventional IoT sensors, IoBNT leverages molecular and nanoscale communication networks to facilitate real-time data collection from biological systems [9]. This high-resolution approach enhances bioprocess monitoring and predictive modeling, making it a key enabler for next-generation DTs in bioengineering and healthcare applications [10]. However, despite its potential, IoBNT data alone is insufficient for constructing high-fidelity DTs [11], as traditional data-driven models fail to incorporate the underlying physics governing bioprocesses.

A promising solution lies in Physics-Informed Neural Networks (PINNs), which integrate physical laws into deep learning models, ensuring biologically plausible and data-efficient predictions [12], [13]. PINNs bridge the gap between theoretical models and data-driven approaches, addressing key challenges in sensor noise, data sparsity, and transmission delays, issues critical to IoT-based bioprocess monitoring [14]. Hence, by incorporating fundamental physical principles, particularly in bioprocesses, PINNs enhance DTs in analyzing and predicting system behaviors across diverse conditions. Moreover, by addressing data gaps and ensuring reliable predictions, PINNs enhance DTs as an ideal platform for communication, monitoring, and continuous process optimization [15], [16]. However, constructing scalable and adaptive PINN architectures for real-time DT applications remains a major research challenge, particularly when balancing domain knowledge with real-time sensor data [17].

B. Related Work

DTs with IoT networks and devices of varying natures pose significant challenges for DT frameworks, often leading to inaccuracies and inefficiencies in capturing system dynamics. Consequently, addressing time synchronization, computational overhead, and real-time monitoring is critical for optimizing DT applications in IoT systems. For example, one study proposed a concurrent end-to-end time synchronization and multi-attribute data resampling scheme to enhance DT construction

accuracy and efficiency [1]. Specifically, digital clocks were established at the remote end, aligning time information for multiple attributes. Additionally, an edge-centric platform was developed to manage multi-dimensional data processing, thereby reducing temporal misalignment among sensing attributes [1]. Similarly, another study investigated DT-assisted task offloading in Unmanned Aerial Vehicle with Mobile Edge Computing (UAV-MEC) networks with energy harvesting [3]. In this work, a real-time simulated environment was leveraged to optimize UAV placement and IoT device association while considering MEC constraints. To achieve this, a branch-and-bound algorithm was employed to find optimal solutions, which was further supplemented by heuristic and convex penalty-based methods to reduce computational complexity [3]. Moreover, an additional study explored the application of human DTs in IoT healthcare systems, emphasizing high-fidelity virtual modeling and real-time status replication. In particular, Generative AI (GAI) was integrated to address data scarcity, bias, and noise [18]. Despite these advancements, existing approaches still suffer from costly testing, complexity, and limited real-time monitoring.

IoBNT integrates biological and nanoscale entities into digital frameworks, facilitating precise sensing, monitoring, and targeted therapeutic applications. By leveraging molecular communication (MC), IoBNT enables real-time data exchange, offering promising solutions for biomedical challenges such as drug delivery, health monitoring, and early disease detection. For example, a study developed a mathematical model and a stochastic particle-based simulator to analyze molecular communication across the blood-tissue barrier (BTB) [8]. Furthermore, another study proposed a novel M-ary concentration-shift keying (CSK) method, CSK with common detection thresholds (CSK-CTs), to improve the efficiency of molecular communication in IoBNT networks [9]. By utilizing optimized release concentrations and low common thresholds, CSK-CT achieved significantly lower error rates and reduced operational complexity. This advancement is particularly beneficial for IoBNT applications requiring efficient data gathering from multiple transmitters with computationally limited receivers [9]. Moreover, a DNA-based molecular communication system utilizing a track-hopper mechanism to enhance cargo transport precision was introduced [19]. This approach improved network reliability, minimized communication delays, and enabled directional control of molecular signals [19]. However, current gaps in computational physics, including difficulties in accurately modeling stochastic nanoscale processes, capturing multi-scale interactions, and enforcing physical constraints during real-time simulations, limit the predictive capabilities of IoBNT systems. These challenges highlight the need for integrated modeling approaches that incorporate physical laws directly into the learning process.

To address these issues, multi-model NN architectures present a sophisticated solution, effectively handling diverse scenarios by leveraging the strengths of specialized models tailored for distinct tasks. For instance, recurrent neural networks (RNNs) excel in processing time-series data extracted from systems in real time, capturing temporal dependencies

and dynamic behaviors [20]. Meanwhile, fully connected neural networks (FCNNs) exhibit exceptional adaptability when guided by the system's underlying physical equations, making them suitable for scenarios where theoretical models are pivotal [21]. Moreover, ensuring the highest levels of reliability and accuracy in DTs requires a well-structured PINN framework that incorporates flexibility. By doing so, the framework can integrate the most appropriate NN architecture based on the accuracy and characteristics of the data extracted from the system or process. Ultimately, this tailored approach facilitates the development of highly precise and dependable DTs.

C. Contributions

Through two distinct scenarios, we demonstrate that no single PINN architecture can universally address predicting challenges. In Scenario 1, we challenge the assumption that PINNs can inherently handle measurement inaccuracies or nonlinear complexities regardless of the network architecture. Scenario 2 further rejects the idea that one architecture can simultaneously address sparse data, dynamic system behavior, and strict physical constraints. These findings emphasize the importance of selecting architectures based on specific task demands. To address the aforementioned challenges, this paper introduces a novel multi-model PINN framework empowered by IoBNT to develop real-time, high-fidelity DTs of bioprocesses. The framework enhances system dynamics monitoring by leveraging IoBNT-enabled nanoscale data acquisition, ensuring precise and reliable data collection at the micro- and nano-scale. It integrates multiple deep neural network (DNN) architectures, including Fully Connected Neural Networks (FCNNs), Residual Block Neural Networks (ResBlock NNs), and Recurrent Neural Networks (RNNs), to enable adaptive, efficient processing tailored to the characteristics of bioprocess data.

Specifically, ResBlock-PINN performs best under noisy or limited data, RNN-PINN effectively captures dynamic system behavior when accurate measurements are available, and FCNN-PINN provides a balanced solution that maintains accuracy while moderately adhering to physical constraints. This flexible combination enables the framework to adapt across varying temporal and steady-state conditions. By systematically combining these architectures, the framework leverages the strengths of each model type to provide accurate, physically consistent digital representations of bioprocesses. Furthermore, the integration of IoBNT-enabled nanosensors facilitates high-resolution, low-latency data acquisition, enhancing the system's ability to deliver reliable, real-time insights. This multi-model strategy not only addresses the limitations of existing IoT-based DT solutions but also significantly improves the scalability, adaptability, and robustness of IoT services and applications, particularly in smart environments and precision biomanufacturing scenarios.

To validate its effectiveness, the proposed framework was implemented and tested in Microbial Growth Concentration (MGC) and Substrate Growth Concentration (SGC), described by Monod equations [22], to evaluate its performance under complex and nonlinear kinetics. Furthermore, it offers

key functionalities for extracting, transferring, preparing, and processing physics-based data to create reliable and real-time DTs. By compensating for data gaps and effectively handling nano-scale sensors, the IoBNT-PINN framework ensures trustworthy, data-driven decision-making. Key contributions and features of the framework are as follows.

- We propose a novel multi-model PINN framework integrated with IoBNT for DTs. This pioneering integration of PINNs and IoBNT revolutionizes DTs and, optimizes accuracy, data reliability, and scalability for complex biosystems.
- We design and advance IoBNT based on the Monod equations to optimize data acquisition and transmission processes. IoBNT specified for Monod equations, enables precise data acquisition and transmission, which enhances efficiency, reliability, and applicability.
- We develop three DNNs embedded within the PINN framework to address challenges related to varying physics and data reliability.
- We perform intensive and comprehensive simulations to evaluate the efficiency of our proposed framework.

The rest of the paper is organized as follows. Section II introduces the fundamentals and the framework architecture with four main layers. Section III presents the proposed methods. After that, Section IV describes experimental setups and Section V presents the performance metrics and evaluations based on established benchmarks. Finally, Section VII concludes the paper.

II. SYSTEM MODEL

This section presents the system architecture that integrates a multi-model PINN framework with IoBNT, aiming to enhance DT applications for complex, physics-driven processes. As illustrated in Fig. 1, the architecture is organized into four well-connected layers, i.e., the physical twin layer, the IoBNT communication layer, the data processing layer, and the application layer.

1) *Physical twin layer*: This layer models the real-world environment, capturing the intricate dynamics of physical phenomena through mathematical formulations. It serves as the foundation for the DT, ensuring that system behaviors are accurately represented and monitored.

2) *IoBNT Layer*: The IoBNT layer of the proposed framework functions as the communication backbone of the proposed framework, seamlessly connecting the physical and computational domains. It plays a vital role in transmitting real-world data from the physical twin layer to the computational layers, enabling effective analysis and decision-making. In addition, this layer is equipped with advanced bio-nano sensors deployed within the physical twin environment. These sensors collect high-resolution data. The data collection process is highly sensitive and precise, allowing for the observation of even the smallest fluctuations within the system [23]. In practical applications, the IoBNT layer can leverage specialized nano-scale nodes for communication, ensuring low latency and real-time data transfer [24]. These nodes are optimized to handle the challenges of nano-scale

environments, such as limited power availability and potential signal interference while maintaining robust connectivity [25]. This capability ensures uninterrupted monitoring, making the IoBNT layer indispensable for applications requiring stringent quality control, such as manufacturing or real-time monitoring. Overall, the IoBNT layer is central to the framework's functionality, providing granular, real-time insights that bridge the gap between the physical twin layer and computational analysis.

Unlike conventional IoT systems, IoBNT involves sensors that operate at molecular or nanoscale resolution and communicate via chemical signals or ligand binding [26]. This leads to sensing behaviors that are inherently noisy, delayed, and stochastic [27]. As such, the DT architecture must incorporate domain-specific knowledge, including biological kinetics and physical constraints, to compensate for data sparsity and enhance reliability. Our use of Monod kinetics and physics-informed loss functions within the multi-model PINN framework is particularly aligned with such needs. Furthermore, the integration of IoBNT supports future applications in real-time bioprocess monitoring, intra-body diagnostics, and biological manufacturing systems, where traditional digital communication may be infeasible [28]. Thus, IoBNT is not used as a generic data source but rather as a domain with properties that directly motivate and shape the framework's architecture.

3) *Data Processing Layer*: Data processing serves as the computational core of the proposed framework, integrating physics-based knowledge with data-driven methodologies. This hybrid approach enables accurate and reliable predictions of processes by leveraging both theoretical models and experimental data. This layer combines domain-specific mathematical models, such as Monod kinetics in this study, with NN architectures to define physics-based loss functions. These loss functions ensure that the predictions generated by the network are consistent with established scientific principles. For example, in microbial growth studies, Monod kinetics provide a foundational framework to describe the relationship between substrate concentration and microbial growth rates. By embedding such models into the loss functions, the PINN layer enforces physical consistency and aligns its outputs with known system behaviors. In parallel, the data-driven loss component of this layer minimizes the differences between model predictions and actual observations collected from the IoBNT layer. This dual-loss approach ensures robust performance, even when the input data is sparse, noisy, or incomplete. By balancing the contributions of physics-based and data-driven components, the PINN layer achieves a synergistic effect that enhances the overall predictive capability. This hybrid strategy enables the data processing layer to simulate complex phenomena with high fidelity.

4) *Application Layer*: The application layer serves as the user interface, providing stakeholders with an interactive, virtualized representation of the monitored system. It consolidates real-time sensor data and PINN-generated predictions, translating complex analytical outputs into intuitive visualizations. This layer supports decision-making in various IoT-enabled domains, such as smart environments, industrial automation, and real-time monitoring applications, ensuring

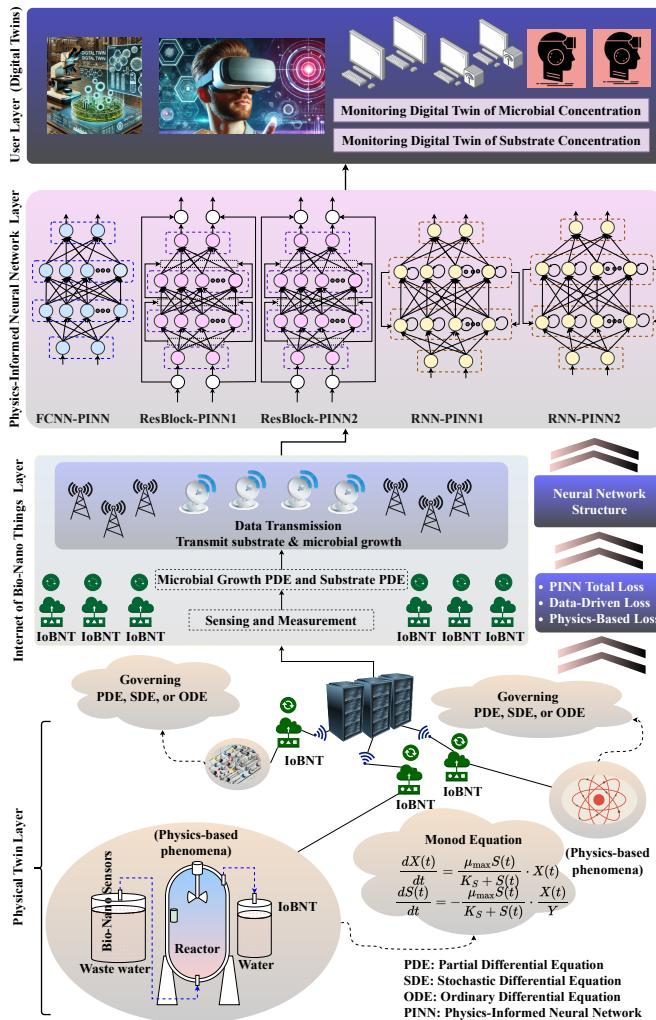


Fig. 1. The proposed system model integrates a multi-model PINN framework with IoBNT, organized into four layers: physical twin, IoBNT, data processing, and application layers.

actionable insights for effective system management.

In this work, the DT represents a computational surrogate of a biological system that integrates domain knowledge through Monod kinetics and dynamically reflects the current state of the physical system using sensor data. The architecture uses a physics-informed loss function to preserve biological realism, while the modularity of the neural network design (e.g., FCNN, ResBlock, and RNN) enables adaptation to diverse system behaviors. Over time, the DT can be updated by incorporating new data and re-optimizing the model parameters. This allows the framework to reflect changes in the physical world such as biological aging, environmental drift, or changes in experimental conditions.

The proposed DT framework can support a range of practical applications. To enhance interpretability and user interaction, the framework can integrate advanced visualization layers. For instance, VR platforms coupled with simulation outputs, as demonstrated in CFD-based fluid mechanics education, allow immersive exploration of system dynamics and data patterns [29]. Such visualization techniques, combining

platforms such as SteamVR, Oculus, and Ansys/EnSight, improve user engagement and conceptual understanding when tested in practical applications [29]. The application layer in the proposed framework can also be utilized for operator training, similar to the VR-based bioreactor simulator at the UK's National Horizons Center [30]. This VR-based bioreactor simulator is used to replicate dynamic bioprocess behavior and to provide real-time feedback [30]. It allows trainees to handle critical scenarios and supports skill validation, which addresses biomanufacturing training requirements in regulated environments [30]. A practical application demonstrating this potential is the use of Augmented Reality (AR) systems for environmental monitoring, where geo-referenced sensor data is visualized on-site to support real-time assessment of ecological conditions [31]. Following this concept, the application layer can be adapted for similar interactive monitoring tasks.

III. THE PROPOSED APPROACHES

A. IoBNT for Microbial Growth and Substrate Optimization

In this paper, the proposed IoBNT are designed to real-time monitoring of microbial growth and nutrient levels according to Monod kinetics. The nutrient concentration, denoted as $S(t)$, serves as the critical resource required for microbial proliferation, while microbial concentration is represented as $X(t)$. These interactions are governed by Monod kinetics, a widely recognized model that describes the relationship between microbial growth and nutrient availability [32].

1) *Substrate dynamics with Monod kinetics:* Substrate consumption adheres to Monod kinetics, which models the reduction in nutrient levels over time t as a result of microbial activity [32]. The dynamics of substrate utilization can be expressed through the following differential equation:

$$\frac{dS(t)}{dt} = -\frac{\mu_{\max}S(t)}{K_S + S(t)} \frac{X(t)}{Y}, \quad (1)$$

where μ_{\max} represents the maximum microbial growth rate achievable under conditions of abundant substrate availability, K_S is the Monod constant signifying the substrate concentration at which the growth rate reaches half of μ_{\max} , and Y denotes the yield coefficient, which measures the efficiency of converting substrate into biomass. The expression $\frac{\mu_{\max}S(t)}{K_S + S(t)}$ defines the specific growth rate, which asymptotically approaches μ_{\max} as the substrate concentration becomes very large ($S(t) \rightarrow \infty$). The negative sign in equation (1) indicates substrate depletion over time.

2) *Microbial growth dynamics:* The microbial concentration, $X(t)$, grows over time as the consumption of the substrate fuels microbial proliferation [33]. The rate of microbial growth is described by the following equation:

$$\frac{dX(t)}{dt} = \frac{\mu_{\max}S(t)}{K_S + S(t)} X(t). \quad (2)$$

Equation (2) demonstrates the direct relationship between microbial growth and substrate availability. Higher substrate concentrations result in faster microbial growth, while limited substrate availability leads to a reduction in growth.

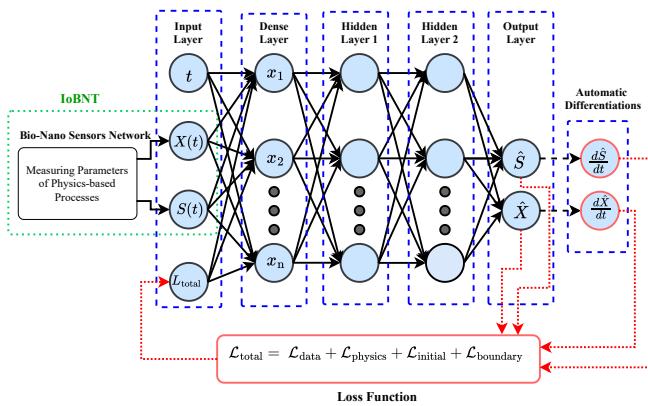


Fig. 2. Architecture of the PINN with FCNN.

B. Loss Functions

The PINN employs several components in its loss function to ensure it fits observed data accurately while also obeying Monod kinetics, which describe microbial growth. The data-driven loss $\mathcal{L}_{\text{data}}$ focuses on reducing the differences between observed and predicted values for $S(t)$ and $X(t)$:

$$\mathcal{L}_{\text{data}} = \frac{1}{N} \sum_{i=1}^N \left((S_i(t) - \hat{S}_i(t))^2 + (X_i(t) - \hat{X}_i(t))^2 \right), \quad (3)$$

where $S_i(t)$ and $X_i(t)$ represent the observed substrate and microbial concentrations, while $\hat{S}_i(t)$ and $\hat{X}_i(t)$ denote their corresponding predicted values. The physics-based loss $\mathcal{L}_{\text{physics}}$ ensures adherence to Monod kinetics by penalizing any deviations from the governing equations:

$$\begin{aligned} \mathcal{L}_{\text{physics}} = & \frac{1}{N} \sum_{i=1}^N \left(\left(\frac{d\hat{S}(t)}{dt} + \frac{1}{Y} \mu(\hat{S}(t)) \hat{X}(t) \right)^2 \right. \\ & \left. + \left(\frac{d\hat{X}(t)}{dt} - \mu(\hat{S}(t)) \hat{X}(t) \right)^2 \right), \end{aligned} \quad (4)$$

where $\mu(\hat{S}(t)) = \frac{\mu_{\max} \hat{S}(t)}{K_S + \hat{S}(t)}$ represents the Monod-based growth rate as a function of the predicted substrate concentration. The total loss function, $\mathcal{L}_{\text{total}}$, integrates all components, including terms for initial and boundary conditions, creating a comprehensive and robust optimization framework:

$$\begin{aligned} \mathcal{L}_{\text{total}} = & \lambda_{\text{data}} \mathcal{L}_{\text{data}} + \lambda_{\text{physics}} \mathcal{L}_{\text{physics}} \\ & + \lambda_{\text{initial}} \mathcal{L}_{\text{initial}} + \lambda_{\text{boundary}} \mathcal{L}_{\text{boundary}}. \end{aligned} \quad (5)$$

The weights λ_{data} , λ_{physics} , λ_{initial} , and $\lambda_{\text{boundary}}$ regulate the relative importance of each loss component.

C. The Proposed PINN's Models

The proposed multi-model PINN framework incorporates three types of NN architectures: FCNN, ResBlock NN, and RNN. These architectures are detailed in this section. For all the models, the same loss function, as defined in (2), is utilized to ensure consistency across the framework.

Algorithm 1 Learning process of FCNN within a PINN Framework

```

1: Initialize: Set learning rate, hidden layers, neurons/layer, and loss weights  $\lambda_{\text{data}}, \lambda_{\text{physics}}$ .
2: Input:  $x(t) = [S(t), X(t)]^\top$ .
3: Transformations:
4: Input layer:  $h^{(0)} = \tanh(W_{\text{input}}x(t) + b_{\text{input}})$ .
5: Hidden layers  $l = 1, \dots, n$ :  $h^{(l+1)} = \tanh(W^{(l)}h^{(l)} + b^{(l)})$ .
6: Output layer:  $[\hat{S}(t+1), \hat{X}(t+1)]^\top = W_{\text{output}}h^{(6)} + b_{\text{output}}$ .
7: Physics Constraints: Monod kinetics:

$$\frac{dS}{dt} = -\frac{\mu_{\max} S}{K_S + S} \frac{X}{Y}, \quad \frac{dX}{dt} = \frac{\mu_{\max} S}{K_S + S} X.$$

8: while training not converged do
9:   Combine losses:  $\mathcal{L}_{\text{total}} = \lambda_{\text{data}} \mathcal{L}_{\text{data}} + \lambda_{\text{physics}} \mathcal{L}_{\text{physics}}$ .
10:  Forward pass: Compute predictions  $\hat{S}(t+1), \hat{X}(t+1)$  and gradients.
11:  Backward pass: Update parameters  $\Theta^{(k+1)} = \Theta^{(k)} - \eta \nabla_{\Theta} \mathcal{L}_{\text{total}}$ .
12: end while
13: Evaluation: Compute RMSE, MAE, and  $R^2$ .

```

1) Architecture of FCNN-PINN: This section provides a detailed description of the FCNN architecture within the proposed framework, along with the associated training algorithm [34], [35]. Fig. 2 shows the architecture of the proposed PINN using FCNN. The input layer for each proposed architecture includes three primary variables obtained from the IoBNT system: time t , substrate concentration $S(t)$, and microbial concentration $X(t)$.

Algorithm 1 describes the learning process of FCNN integrated within the PINN framework. The output layer produces predictions for the substrate and microbial concentrations, denoted as $\hat{S}(t)$ and $\hat{X}(t)$, respectively. To enforce adherence to the underlying physical laws, the model uses automatic differentiation to compute derivatives, such as $\frac{d\hat{S}(t)}{dt}$ and $\frac{d\hat{X}(t)}{dt}$. These derivatives are critical for ensuring that the predicted dynamics align with the governing equations, maintaining both empirical accuracy and physical consistency. The FCNN processes input data $x(t) \in \mathbb{R}^2$, where $x(t) = [S(t), X(t)]^\top$, through hierarchical transformations in multiple layers. The input layer maps $x(t)$ into a higher-dimensional feature space [34], [36]:

$$h^{(0)} = \tanh(W_{\text{input}}x(t) + b_{\text{input}}), \quad (6)$$

where $W_{\text{input}} \in \mathbb{R}^{D_{\text{in}} \times D_h}$ and $b_{\text{input}} \in \mathbb{R}^{D_h}$ are the weights and biases of the input layer. The network includes two groups of three hidden layers, where each hidden layer l performs the transformation:

$$h^{(l+1)} = \tanh(W^{(l)}h^{(l)} + b^{(l)}), \quad l = 1, \dots, n, \quad (7)$$

where $W^{(l)} \in \mathbb{R}^{D_h \times D_h}$ and $b^{(l)} \in \mathbb{R}^{D_h}$. The nonlinear activation function \tanh is applied element-wise. Finally, the output layer maps the hidden representation to the predicted substrate and microbial concentrations:

$$\begin{bmatrix} \hat{S}(t+1) \\ \hat{X}(t+1) \end{bmatrix} = W_{\text{output}}h^{(6)} + b_{\text{output}}, \quad (8)$$

where $W_{\text{output}} \in \mathbb{R}^{D_h \times D_{\text{out}}}$ and $b_{\text{output}} \in \mathbb{R}^{D_{\text{out}}}$ are the output weights and biases, respectively.

Training algorithm: The training algorithm minimizes the total loss function $\mathcal{L}_{\text{total}}$, which has been previously defined,

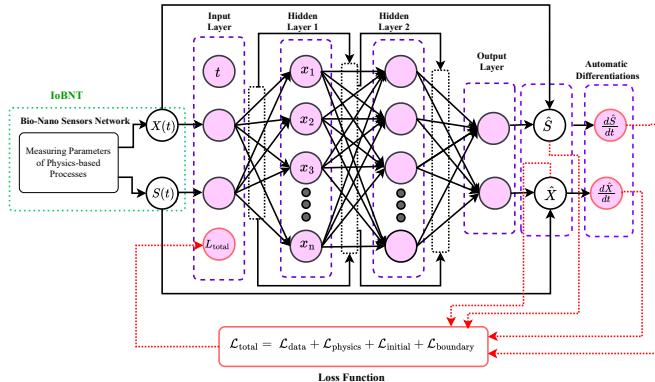


Fig. 3. Architecture of the proposed PINN-ResBlock.

by iteratively updating the model parameters. The key steps in the training process are as follows:

Forward pass: The forward pass propagates the input $x(t)$ through the network layers as defined by (6), (7), and (8), producing predictions $\hat{S}(t+1)$ and $\hat{X}(t+1)$. Automatic differentiation computes the predicted time derivatives:

$$\frac{d\hat{S}(t)}{dt} = \frac{\partial \hat{S}(t)}{\partial t}, \quad \frac{d\hat{X}(t)}{dt} = \frac{\partial \hat{X}(t)}{\partial t}. \quad (9)$$

Backward pass: The backward pass computes gradients of the total loss $\mathcal{L}_{\text{total}}$ with respect to the network parameters. For each layer, the gradients are calculated using backpropagation:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial W^{(l)}} = \delta^{(l)} h^{(l-1)\top}, \quad \frac{\partial \mathcal{L}_{\text{total}}}{\partial b^{(l)}} = \delta^{(l)}, \quad (10)$$

where $\delta^{(l)}$ is the error term propagated backward:

$$\delta^{(l)} = \left(W^{(l+1)\top} \delta^{(l+1)} \right) \odot \tanh'(z^{(l)}), \quad z^{(l)} = W^{(l)} h^{(l-1)} + b^{(l)} \quad (11)$$

Physics-informed constraints: The physics-based constraints, derived from Monod kinetics, are enforced during training:

$$\frac{dS(t)}{dt} = -\frac{\mu_{\max} S(t)}{K_S + S(t)} \frac{X(t)}{Y}, \quad \frac{dX(t)}{dt} = \frac{\mu_{\max} S(t)}{K_S + S(t)} X(t). \quad (12)$$

These constraints guide the network's predictions to remain consistent with the governing biochemical processes.

Parameter update: The model parameters $\Theta = \{W_{\text{input}}, W^{(l)}, W_{\text{output}}, b_{\text{input}}, b^{(l)}, b_{\text{output}}\}$ are updated using gradient-based optimization:

$$\Theta^{(k+1)} = \Theta^{(k)} - \eta \nabla_{\Theta} \mathcal{L}_{\text{total}}, \quad (13)$$

where η is the learning rate, and k is the iteration index.

Iterative training: The network is trained iteratively for a fixed number of epochs N_{epochs} . During each epoch, the total loss $\mathcal{L}_{\text{total}}$ is minimized:

$$\min_{\Theta} \mathcal{L}_{\text{total}}. \quad (14)$$

The training process ensures that the model accurately predicts microbial and substrate dynamics while adhering to the Monod kinetics.

Algorithm 2 Training algorithm for ResBlock in PINN Framework

```

1: Initialize: Set learning rate  $\eta$ , residual blocks  $L$ , loss weights  $\lambda_{\text{data}}, \lambda_{\text{physics}}$ , and input  $x_0 \in \mathbb{R}^d$ .
2: for  $i = 1$  to  $L$  do
3:   Compute  $x_{i+1} = x_i + F(x_i; \Theta_i)$ ,  $F(x_i; \Theta_i) = \sigma(W_{i,2}\sigma(W_{i,1}x_i + b_{i,1}) + b_{i,2})$ .
4: end for
5: Compute output:  $y = W_o x_L + b_o$ . Enforce Monod kinetics:
   
$$\frac{d\hat{S}(t)}{dt} = -\frac{\mu_{\max} \hat{S}(t)}{K_S + \hat{S}(t)} \frac{\hat{X}(t)}{Y}, \quad \frac{d\hat{X}(t)}{dt} = \frac{\mu_{\max} \hat{S}(t)}{K_S + \hat{S}(t)} \hat{X}(t).$$

6: for  $i = L$  to 1 do
7:   Compute gradients:
      
$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial W_{i,k}} = \sum_{t=1}^T \frac{\partial \mathcal{L}_{\text{total}}}{\partial x_{i+1}} \frac{\partial x_{i+1}}{\partial F(x_i; \Theta_i)} \frac{\partial F(x_i; \Theta_i)}{\partial W_{i,k}},$$

      update  $\frac{\partial \mathcal{L}_{\text{total}}}{\partial x_i} = \frac{\partial \mathcal{L}_{\text{total}}}{\partial x_{i+1}} \left( I + \frac{\partial F(x_i; \Theta_i)}{\partial x_i} \right)$ ,  $\frac{\partial F(x_i; \Theta_i)}{\partial x_i} = W_{i,1}^\top \sigma'(z_i)$ .
8: end for
9: Update parameters:  $\Theta^{(k+1)} = \Theta^{(k)} - \eta \nabla_{\Theta} \mathcal{L}_{\text{total}}$ .
10: Optimize  $\min_{\Theta} \mathcal{L}_{\text{total}}$  over  $N_{\text{epochs}}$ .
```

D. Architecture of ResBlock-PINN

The proposed ResBlock within the PINN framework introduces skip connections to alleviate the vanishing gradient problem and improve gradient flow during training [37]. Fig. 3 demonstrates the use of residual connections to enhance the learning of microbial and substrate dynamics. In addition, Algorithm 2 presents the training algorithm for ResBlock within the PINN framework. The output of the residual block is defined as [38]:

$$x_{i+1} = x_i + F(x_i; \Theta_i), \quad (15)$$

where $F(x_i; \Theta_i)$ represents the transformation applied by the dense layers in the i -th block, parameterized by weights and biases Θ_i . The transformation $F(x_i; \Theta_i)$ is given by:

$$F(x_i; \Theta_i) = \sigma(W_{i,2}\sigma(W_{i,1}x_i + b_{i,1}) + b_{i,2}), \quad (16)$$

where $W_{i,1}, W_{i,2} \in \mathbb{R}^{d \times d}$ are weight matrices, $b_{i,1}, b_{i,2} \in \mathbb{R}^d$ are biases, and σ is the activation function, such as tanh. The network output y for an input x is computed by sequentially passing the input through the residual blocks and a final output layer:

$$y = W_o x_L + b_o, \quad (17)$$

where x_L is the output of the last residual block, and $W_o \in \mathbb{R}^{d \times 2}, b_o \in \mathbb{R}^2$ are the weights and biases of the output layer.

Training algorithm for ResBlock in PINN framework:

The ResBlock NN is trained by minimizing the total loss $\mathcal{L}_{\text{total}}$, which combines data-driven, physics-based, initial condition, and boundary condition losses.

Forward pass and residual block computation:

During the forward pass, the input x is propagated through each residual block:

$$x_{i+1} = x_i + F(x_i; \Theta_i), \quad (18)$$

where $F(x_i; \Theta_i)$ is computed as in (16). The final network output is computed using (17). For physics-based constraints, automatic differentiation is used to compute the

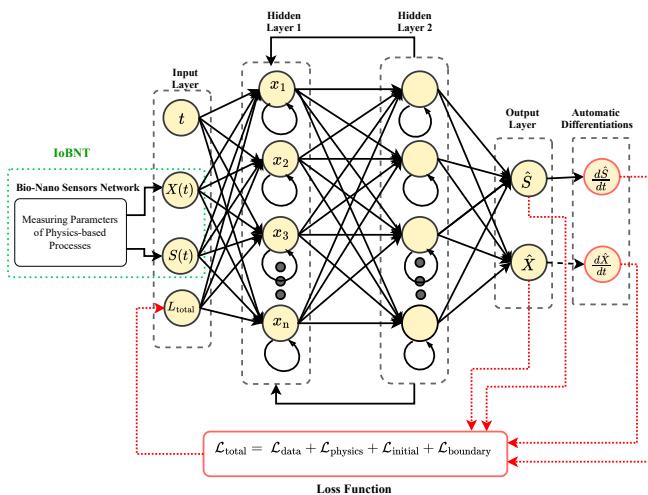


Fig. 4. Architecture of PINN-RNN.

time derivatives $\frac{d\hat{S}(t)}{dt}$ and $\frac{d\hat{X}(t)}{dt}$, ensuring consistency with Monod kinetics:

$$\frac{d\hat{S}(t)}{dt} = -\frac{\mu_{\max}\hat{S}(t)}{K_S + \hat{S}(t)} \frac{\hat{X}(t)}{Y}, \quad \frac{d\hat{X}(t)}{dt} = \frac{\mu_{\max}\hat{S}(t)}{K_S + \hat{S}(t)} \hat{X}(t). \quad (19)$$

Backward pass and gradient computation:

The gradients of the total loss $\mathcal{L}_{\text{total}}$ with respect to the network parameters are computed using backpropagation. The gradient of the loss with respect to the weights of the i -th residual block is given by:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial W_{i,k}} = \sum_{t=1}^T \frac{\partial \mathcal{L}_{\text{total}}}{\partial x_{i+1}} \frac{\partial x_{i+1}}{\partial F(x_i; \Theta_i)} \frac{\partial F(x_i; \Theta_i)}{\partial W_{i,k}}, \quad (20)$$

where $k = 1, 2$ denotes the layer index within the residual block.

The chain rule is used to compute the gradient with respect to x_i recursively:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial x_i} = \frac{\partial \mathcal{L}_{\text{total}}}{\partial x_{i+1}} \cdot \left(I + \frac{\partial F(x_i; \Theta_i)}{\partial x_i} \right), \quad (21)$$

where I is the identity matrix, and $\frac{\partial F(x_i; \Theta_i)}{\partial x_i}$ is computed as:

$$\frac{\partial F(x_i; \Theta_i)}{\partial x_i} = W_{i,1}^\top \sigma'(z_i), \quad (22)$$

with $z_i = W_{i,1}x_i + b_{i,1}$.

Parameter updates:

The parameters $\Theta = \{W_{i,k}, b_{i,k}, W_o, b_o\}$ are updated using gradient descent:

$$\Theta^{(k+1)} = \Theta^{(k)} - \eta \nabla_\Theta \mathcal{L}_{\text{total}}, \quad (23)$$

where η is the learning rate, and k is the iteration index.

Iterative training with physics constraints:

The ResBlock NN iteratively optimizes $\mathcal{L}_{\text{total}}$ while enforcing Monod kinetics (19). The algorithm repeats forward and backward passes for a predefined number of epochs N_{epochs} , ensuring convergence to the optimal solution:

$$\min_{\Theta} \mathcal{L}_{\text{total}}. \quad (24)$$

E. Architecture of RNN-PINN

The integration of RNNs within PINNs enables the modeling of dynamic systems by leveraging the temporal sequence learning capability of RNNs alongside the physics-based constraints of PINNs [39]. Fig. 4 illustrates the integration of sequential processing with physics-based modeling for predicting microbial and substrate dynamics. Furthermore, Algorithm 3 outlines the learning process of an RNN integrated within the PINN framework. An RNN processes input data sequentially by updating a hidden state $h(t)$ at each time step. The hidden state at time t is computed using the equation [40]:

$$h(t) = \sigma(W_h h(t-1) + W_x x(t) + b_h), \quad (25)$$

where $h(t) \in \mathbb{R}^{d_h}$ is the hidden state vector, $h(t-1)$ is the hidden state from the previous time step, $W_h \in \mathbb{R}^{d_h \times d_h}$ is the recurrent weight matrix, $W_x \in \mathbb{R}^{d_x \times d_h}$ is the input weight matrix, $b_h \in \mathbb{R}^{d_h}$ is the bias vector, and σ is the activation function, such as tanh. The input vector $x(t) = [S(t), X(t)]^\top$ represents substrate and microbial concentrations at time t , where $S(t)$ and $X(t)$ are the substrate and microbial concentrations, respectively. The output of the RNN, representing the predicted substrate and microbial concentrations at the next time step, is given by:

$$\begin{bmatrix} \hat{S}(t+1) \\ \hat{X}(t+1) \end{bmatrix} = W_o h(t) + b_o, \quad (26)$$

where $W_o \in \mathbb{R}^{d_h \times 2}$ is the output weight matrix, and $b_o \in \mathbb{R}^2$ is the output bias vector. The RNN sequentially propagates information through time, enabling it to model dynamic dependencies.

Training algorithm:

The training process minimizes the total loss $\mathcal{L}_{\text{total}}$, combining data-driven, physics-based, initial condition, and boundary condition losses. The algorithm involves forward propagation, gradient computation, and parameter updates.

Forward pass and hidden state propagation:

During the forward pass, the RNN computes hidden states for all time steps $t = 1, 2, \dots, T$ using the recurrence relation:

$$h(t) = \sigma(W_h h(t-1) + W_x x(t) + b_h). \quad (27)$$

The predicted substrate and microbial concentrations for the next time step are obtained as:

$$\begin{bmatrix} \hat{S}(t+1) \\ \hat{X}(t+1) \end{bmatrix} = W_o h(t) + b_o. \quad (28)$$

Automatic differentiation is employed to calculate the time derivatives $\frac{d\hat{S}(t)}{dt}$ and $\frac{d\hat{X}(t)}{dt}$, required for enforcing physics-based constraints [41], [42].

Backward pass and gradient computation:

The gradients of the total loss with respect to the hidden states are computed iteratively using the chain rule. For each time step t , the gradient is given by:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t)} = \frac{\partial \mathcal{L}_{\text{total}}}{\partial \hat{S}(t+1)} \frac{\partial \hat{S}(t+1)}{\partial h(t)} + \frac{\partial \mathcal{L}_{\text{total}}}{\partial \hat{X}(t+1)} \frac{\partial \hat{X}(t+1)}{\partial h(t)}, \quad (29)$$

Algorithm 3 Learning process of RNN within a PINN Framework

```

1: Initialize: Set learning rate  $\eta$ , hidden units  $d_h$ , and loss weights  $\lambda_{\text{data}}, \lambda_{\text{physics}}$ . Input sequential data  $x(t) = [S(t), X(t)]^\top$ .
2: for  $t = 1$  to  $T$  do
3:   Compute hidden state:  $h(t) = \sigma(W_h h(t-1) + W_x x(t) + b_h)$ .
4:   Predict concentrations:  $[\hat{S}(t+1), \hat{X}(t+1)]^\top = W_o h(t) + b_o$ .
5:   Compute time derivatives:  $\frac{d\hat{S}(t)}{dt}, \frac{d\hat{X}(t)}{dt}$ .
6: end for
7: Physics constraints: Enforce Monod kinetics:  $\frac{dS}{dt} = -\frac{\mu_{\max} S}{K_S + S} \frac{X}{Y}$ ,  $\frac{dX}{dt} = \frac{\mu_{\max} S}{K_S + S} X$ .
8: for  $t = T$  to 1 do
9:   Compute gradient:  $\frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t)} = \frac{\partial \mathcal{L}_{\text{total}}}{\partial \hat{S}(t+1)} \frac{\partial \hat{S}(t+1)}{\partial h(t)} + \frac{\partial \mathcal{L}_{\text{total}}}{\partial \hat{X}(t+1)} \frac{\partial \hat{X}(t+1)}{\partial h(t)}$ .
10:  Backpropagate:  $\frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t-1)} = \frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t)} \cdot \frac{\partial h(t)}{\partial h(t-1)}$ .
11: end for
12: Update Parameters: Compute gradients w.r.t.  $\Theta = \{W_h, W_x, W_o, b_h, b_o\}$  and update:  $\Theta^{(k+1)} = \Theta^{(k)} - \eta \nabla_{\Theta} \mathcal{L}_{\text{total}}$ .
13: while not converged do
14:   Minimize loss:  $\mathcal{L}_{\text{total}} = \lambda_{\text{data}} \mathcal{L}_{\text{data}} + \lambda_{\text{physics}} \mathcal{L}_{\text{physics}}$ .
15: end while
16: Evaluate: Compute RMSE, MAE, and  $R^2$ .

```

where $\frac{\partial \hat{S}(t+1)}{\partial h(t)} = W_o^{(S)}$ and $\frac{\partial \hat{X}(t+1)}{\partial h(t)} = W_o^{(X)}$, with $W_o^{(S)}$ and $W_o^{(X)}$ being rows of W_o corresponding to substrate and microbial predictions, respectively. Backpropagation Through Time (BPTT) is used to compute gradients for all time steps. The gradient of the hidden state at time $t-1$ is calculated recursively as:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t-1)} = \frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t)} \cdot \frac{\partial h(t)}{\partial h(t-1)}, \quad (30)$$

where:

$$\frac{\partial h(t)}{\partial h(t-1)} = \sigma'(z(t))W_h, \quad z(t) = W_h h(t-1) + W_x x(t) + b_h. \quad (31)$$

Parameter updates:

The gradients with respect to the RNN parameters $\Theta = \{W_h, W_x, W_o, b_h, b_o\}$ are computed as:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial W_h} = \sum_{t=1}^T \frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t)} \frac{\partial h(t)}{\partial W_h}, \quad \frac{\partial h(t)}{\partial W_h} = \sigma'(z(t))h(t-1)^\top, \quad (32)$$

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial W_x} = \sum_{t=1}^T \frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t)} \frac{\partial h(t)}{\partial W_x}, \quad \frac{\partial h(t)}{\partial W_x} = \sigma'(z(t))x(t)^\top, \quad (33)$$

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial W_o} = \sum_{t=1}^T \frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t)} h(t)^\top, \quad \frac{\partial \mathcal{L}_{\text{total}}}{\partial b_o} = \sum_{t=1}^T \frac{\partial \mathcal{L}_{\text{total}}}{\partial h(t)}. \quad (34)$$

The parameters are updated using gradient descent:

$$\Theta^{(k+1)} = \Theta^{(k)} - \eta \nabla_{\Theta} \mathcal{L}_{\text{total}}, \quad (35)$$

where η is the learning rate and k denotes the iteration index.

Integration with physics-informed constraints:

The physics-based constraints are enforced by ensuring predictions adhere to Monod kinetics:

$$\frac{dS}{dt} = -\frac{\mu_{\max} S}{K_S + S} \frac{X}{Y}, \quad \frac{dX}{dt} = \frac{\mu_{\max} S}{K_S + S} X. \quad (36)$$

The time derivatives $\frac{d\hat{S}(t)}{dt}$ and $\frac{d\hat{X}(t)}{dt}$ are computed using automatic differentiation:

$$\frac{d\hat{S}(t)}{dt} = \frac{\partial \hat{S}(t)}{\partial h(t)} \cdot \frac{\partial h(t)}{\partial t}, \quad \frac{d\hat{X}(t)}{dt} = \frac{\partial \hat{X}(t)}{\partial h(t)} \cdot \frac{\partial h(t)}{\partial t}. \quad (37)$$

Iterative optimization:

The forward and backward passes are repeated for a predefined number of epochs N_{epochs} , ensuring convergence to the optimal parameters:

$$\min_{\Theta} \mathcal{L}_{\text{total}}. \quad (38)$$

This training algorithm integrates the sequential modeling capabilities of RNNs with the physics-informed framework, ensuring temporal dependencies and physical consistency are simultaneously captured.

IV. EXPERIMENTAL SETUPS

1) *Scenario 1:* The hypothesis underpinning scenario 1 is that the type of NN integrated within the PINN framework can compensate for measurement inaccuracies or enhance adaptability to nonlinear systems, regardless of the specific NN architecture. Additionally, this scenario investigates whether the relative weights assigned to loss functions significantly impact the performance of different architectures for the same system. Each architecture is configured with distinct weight settings for its loss functions, reflecting different emphases on data accuracy versus the incorporation of physics-based constraints, as detailed in Table I.

To systematically explore these hypotheses, we designed Scenario 1 as a controlled experimental framework targeting two key aspects of PINN performance. First, we assessed how variations in NN architectures influence both the learning dynamics and predictive accuracy under physics-informed constraints. Second, we examined how different weighting strategies for the loss components affect the model's ability to balance data fidelity with the enforcement of governing equations. By applying structured combinations of architectures and loss weightings, this scenario enables a comparative evaluation of the trade-offs between expressiveness, regularization by physics, and robustness to data noise.

2) *Scenario 2:* In contrast to the hypothesis in the first scenario, where the focus was on demonstrating the ability of the PINN framework to compensate for weaknesses in measurements and physical complexities, scenario 2 emphasizes the importance of NN architecture in data analysis and processing. While PINNs are inherently capable of handling sparse or noisy data due to their integration of physics-based constraints, the choice of NN architecture significantly impacts performance. In scenario 2, the framework is extended to evaluate and challenge the most suitable NN architectures within the PINN framework. These include FCNN-PINN2-1, ResBlock-PINN2-1, ResBlock-PINN2-2, RNN-PINN2-1, and RNN-PINN2-1. Table II outlines the numerical settings and training configurations for scenario 2.

3) *Cross-Validated Grid Search for ANN Hyperparameter Optimization:* To objectively determine an appropriate ANN architecture for predicting microbial and substrate concentrations based on time, we applied a systematic hyperparameter tuning strategy. Specifically, we used a cross-validated grid search method [43]–[45] to explore multiple ANN configurations and identify the optimal combination of layer sizes, activation functions, learning rates, training epochs, and batch sizes.

TABLE I: Numerical settings and training configurations for the first scenario by PINN methods, including FCNN-PINN1-1 and RNN-PINN1-1

Method	Initial Conditions	Model Parameters	Loss Weights	Neural Network
FCNN-PINN1-1	$S_0 = 10.0, X_0 = 1.0$	$\mu_{\max} = 0.4, K_S = 0.5, Y = 0.6$	$\lambda_{\text{data}} = 1.0, \lambda_{\text{physics}} = 0.5, \lambda_{\text{initial}} = 0.5, \lambda_{\text{boundary}} = 0.5$	Input: 60 neurons (tanh), 2 groups of 3 hidden layers (60 neurons each)Output: 2 neurons
RNN-PINN1-1	$S_0 = 10.0, X_0 = 1.0$	$\mu_{\max} = 0.4, K_S = 0.5, Y = 0.6$	$\lambda_{\text{data}} = 1.0, \lambda_{\text{physics}} = 10.0, \lambda_{\text{initial}} = 10.0, \lambda_{\text{boundary}} = 40.0$	Single RNN layer (60 units, tanh)Output: 2 neurons

TABLE II: Numerical settings and training configurations for the second scenario through PINN methods, including FCNN-PINN2-1, ResBlock-PINN2-1, ResBlock-PINN2-2, RNN-PINN2-1, and RNN-PINN2-2

Method	Initial Conditions	Model Parameters	Loss Weights	Neural Network (NN)
FCNN-PINN2-1	$S_0 = 10.0, X_0 = 1.0$	$\mu_{\max} = 0.4, K_S = 0.5, Y = 0.6$	$\lambda_{\text{data}} = 1.0, \lambda_{\text{physics}} = 10.0, \lambda_{\text{initial}} = 10.0, \lambda_{\text{boundary}} = 40.0$	Input: 60 neurons (tanh), 2 groups of 3 hidden layers (60 neurons each)Output: 2 neurons
ResBlock-PINN2-1	$S_0 = 10.0, X_0 = 1.0$	$\mu_{\max} = 0.4, K_S = 0.5, Y = 0.6$	$\lambda_{\text{data}} = 1.0, \lambda_{\text{physics}} = 5.0, \lambda_{\text{initial}} = 5.0, \lambda_{\text{boundary}} = 10.0$	Input: 60 neurons (tanh), 2 residual blocks (2 layers of 60 neurons each)Output: 2 neurons with skip connections
ResBlock-PINN2-2	$S_0 = 10.0, X_0 = 1.0$	$\mu_{\max} = 0.4, K_S = 0.5, Y = 0.6$	$\lambda_{\text{data}} = 1.0, \lambda_{\text{physics}} = 10.0, \lambda_{\text{initial}} = 10.0, \lambda_{\text{boundary}} = 20.0$	Input: 60 neurons (tanh), 2 residual blocks (2 layers of 60 neurons each)Output: 2 neurons with skip connections
RNN-PINN2-1	$S_0 = 10.0, X_0 = 1.0$	$\mu_{\max} = 0.4, K_S = 0.5, Y = 0.6$	$\lambda_{\text{data}} = 1.0, \lambda_{\text{physics}} = 1.0, \lambda_{\text{initial}} = 1.0, \lambda_{\text{boundary}} = 1.0$	Single RNN layer (60 units, tanh)Output: 2 neurons
RNN-PINN2-2	$S_0 = 10.0, X_0 = 1.0$	$\mu_{\max} = 0.4, K_S = 0.5, Y = 0.6$	$\lambda_{\text{data}} = 1.0, \lambda_{\text{physics}} = 0.5, \lambda_{\text{initial}} = 0.5, \lambda_{\text{boundary}} = 0.5$	Single RNN layer (60 units, tanh)Output: 2 neurons

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ represent the dataset, where x_i denotes the time (in hours), and $y_i = [S_i, M_i]$ includes substrate and microbial concentrations in g/L. “Hours” indicate the biological time horizon of the simulated microbial growth process rather than computational training time. Our goal is to learn a mapping $f(x; \theta)$ parameterized by network weights θ to minimize the mean squared error (MSE) between predictions and true values over a 3-fold cross-validation:

$$\min_{\theta} \frac{1}{K} \sum_{k=1}^K \text{MSE}_k(\theta), \quad (39)$$

where $K = 3$ is the number of folds. The grid search was conducted over hidden layer sizes $[(15,), (30,), (60,), (15, 15), (30, 30), (60, 60), (15, 15, 15)]$, activation functions (\tanh , ReLU), learning rates ($10^{-3}, 10^{-2}$), maximum epochs (200 to 10,000), and batch sizes (auto, 32). Each configuration is evaluated using 3-fold cross-validation. The optimal setting minimizes the average validation MSE. As shown in Fig. 5, the best-performing configuration was a two-layer network with (60, 60) neurons and \tanh activation. Additionally, as illustrated in Fig. 6, longer training durations resulted in more stable and accurate convergence, with 10,000 training epochs yielding the best generalization.

4) *Architecture Transfer Strategy*: Based on the optimal FCNN configuration, we transferred the identified structure (60 neurons per layer and \tanh activation) to the ResBlock-PINN design. Specifically, each residual block is composed of two dense layers (60 neurons per layer). This preserves the representational capacity and provides stable performance across training runs. For RNN-PINNs, we empirically observed that deeper recurrent architectures (e.g., multiple stacked RNN layers) led to overfitting and higher variance. Therefore, we reduced the architecture to a single SimpleRNN layer with 60 neurons. This model not only converges faster but also yields superior test accuracy, justifying its selection. The same training settings, including 10,000 epochs and mini-batch size of 32, were used for all architectures to ensure consistency and fair comparison.

Data were generated by integrating the Monod equations at 100 uniformly spaced time points from 0 to 10 h. Each trajectory was split temporally into training (0-7 h, 70 points), validation (7-8.5 h, 15 points), and test (8.5-10 h, 15 points)

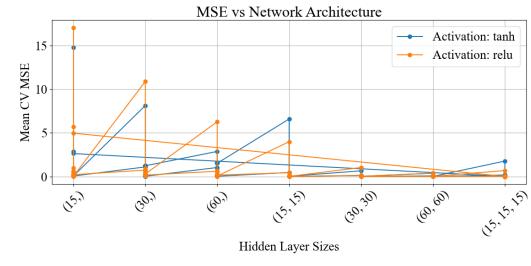


Fig. 5. Cross-validated MSE for different ANN architectures and activations. Best performance was achieved with (60, 60) using \tanh .

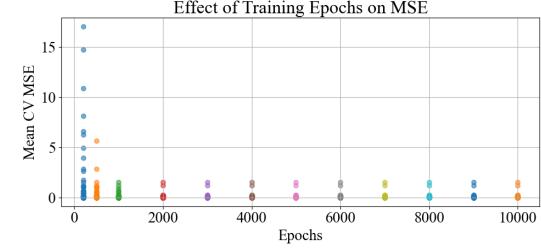


Fig. 6. Impact of training epochs on validation MSE. Longer training (10,000 epochs) achieved the best generalization.

sets without random sampling, preserving chronological order and assessing true forecasting performance.

V. PERFORMANCE EVALUATION

A. Performance Evaluation in Scenario 1

This section presents the simulation results for the first scenario, focusing on the performance of FCNN-PINN1-1 and RNN-PINN1-1 in estimating substrate and microbial concentrations compared to the physical model.

1) *Performance Analysis of Scenario 1*: Fig. 7(a) compares substrate concentration predicted by FCNN-PINN1-1 to the physical model, revealing unrealistic rapid depletion, including negative values. This reflects poor robustness, overfitting, and a failure to incorporate temporal dynamics. Similarly, Fig. 7(b) shows microbial concentration predictions by FCNN-PINN1-1, which exhibit sharp initial growth followed by incorrect saturation dynamics, further indicating inadequate

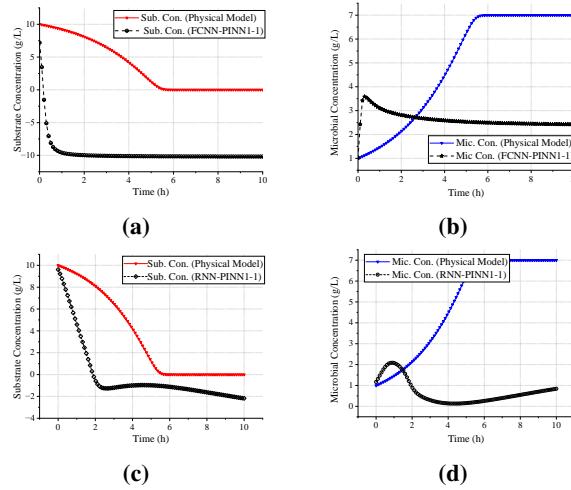


Fig. 7. The performance of FCNN-PINN1-1 and RNN-PINN1-1 compared to the physical model in scenario 1.

modeling. Fig. 7(c) illustrates substrate concentration predictions by RNN-PINN1-1. Additionally, Fig. 7(d) depicts microbial concentration predicted by RNN-PINN1-1, showing initial underestimation and later divergence from expected saturation behavior.

Fig. 8(a) presents the substrate concentration from the physical model, serving as the target behavior. Substrate decreases predictably due to microbial activity. Fig. 8(c) and Fig. 8(e) show predictions by FCNN-PINN1-1 and RNN-PINN1-1, respectively. FCNN-PINN1-1 fails to maintain expected trajectories, while RNN-PINN1-1 captures trends but introduces temporal inconsistencies. Similarly, Fig. 8(b) shows microbial concentration from the physical model, with exponential growth followed by stabilization. Tables III and IV highlight the limitations of FCNN-PINN1-1 and RNN-PINN1-1 in scenario 1. FCNN-PINN1-1 prioritizes system dynamics over measured data trends. However, its high final total loss (1.30×10^2) and data loss (9.68×10^1) indicate poor balance between data and equations. The results from scenario 1 contradict the hypothesis that NN architectures within the PINN framework can universally compensate for measurement inaccuracies or nonlinear system complexities, irrespective of the specific architecture. Both FCNN-PINN1-1 and RNN-PINN1-1 demonstrated significant limitations in capturing the underlying dynamics of the biochemical system, challenging the validity of this assumption.

To better understand the learning dynamics and performance limitations observed in Scenario 1, we provide the training loss curves for FCNN-PINN1-1 and RNN-PINN1-1 in Fig. 9. These curves show the evolution of total loss, data loss, physics loss, and initial condition loss over the training epochs, offering insight into how different loss components influenced the optimization process. For FCNN-PINN1-1, although moderate weights were assigned to the physics and initial condition terms ($\lambda_{\text{physics}} = 0.5$, $\lambda_{\text{initial}} = 0.5$), the data loss rapidly decreases and dominates the training objective. In contrast, the physics and initial losses remain relatively high throughout training. This imbalance results in the model primarily fitting the supervised data while underutilizing phys-

ical constraints, which leads to poor physical plausibility in the predicted dynamics, including unrealistic substrate and microbial concentration trajectories.

TABLE III: Loss components across methods in scenario 1 for estimating microbial and substrate concentrations

Metric	FCNN-PINN1-1	RNN-PINN1-1
Total Loss (Initial)	77.06	1053.41
Total Loss (Final)	1.30×10^2	6.52×10^1
Data Loss (Initial)	26.56	36.77
Data Loss (Final)	9.68×10^1	2.25×10^1
Physics Loss (Initial)	1.29×10^{-2}	6.643×10^{-1}
Physics Loss (Final)	5.96×10^1	4.09×10^0
Initial Loss (Initial)	101.00	101.00
Initial Loss (Final)	7.80×10^0	1.78×10^{-1}

TABLE IV: Summary of RMSE, MAE, and R^2 metrics in the first case scenario

Metric	FCNN-PINN1-1	RNN-PINN1-1
RMSE (Initial)	4.70	6.02
RMSE (Final)	9.84	4.74
RMSE (Average)	10.13	5.24
MAE (Initial)	3.78	5.34
MAE (Final)	7.99	3.98
MAE (Average)	8.39	4.50
R^2 (Initial)	-1.11	-2.45
R^2 (Final)	-8.23	-1.15
R^2 (Average)	-8.83	-1.64

For RNN-PINN1-1, higher weights were assigned to the physics and initial losses ($\lambda_{\text{physics}} = 10.0$, $\lambda_{\text{initial}} = 10.0$). However, the training progression shows that the physics loss remains low but flat, indicating weak gradient contributions from this component during optimization. The learning process is thus driven predominantly by the initial and data losses, leading to inadequate enforcement of the physics-based terms and deviations from the expected physical behavior. These results indicate that the degradation in model performance observed in Scenario 1 stems from imbalanced contributions of the loss components and the resulting ineffective training dynamics. This highlights the importance of carefully tuning loss function weights in PINN frameworks.

In Scenario 1, the FCNN-PINN1-1 model produced unphysical substrate values, including negative concentrations. This outcome is attributed to the underweighting of physics-based constraints during training. Although moderate weights were assigned to the physics and initial condition loss terms ($\lambda_{\text{physics}} = 0.5$, $\lambda_{\text{initial}} = 0.5$), the supervised data loss was weighted more heavily ($\lambda_{\text{data}} = 1.0$). As illustrated in the training curves (Fig. 9), this imbalance led the network to prioritize fitting the data over satisfying the physical equations. The elevated and persistent physics loss throughout training suggests that the governing Monod kinetics were not strongly enforced. As a result, the network was free to fit values that satisfied the data term, even if they violated physically valid bounds such as positivity constraints for concentrations. This finding underscores the necessity of carefully balancing loss components to prevent the emergence of physically implausible behaviors in PINN architectures.

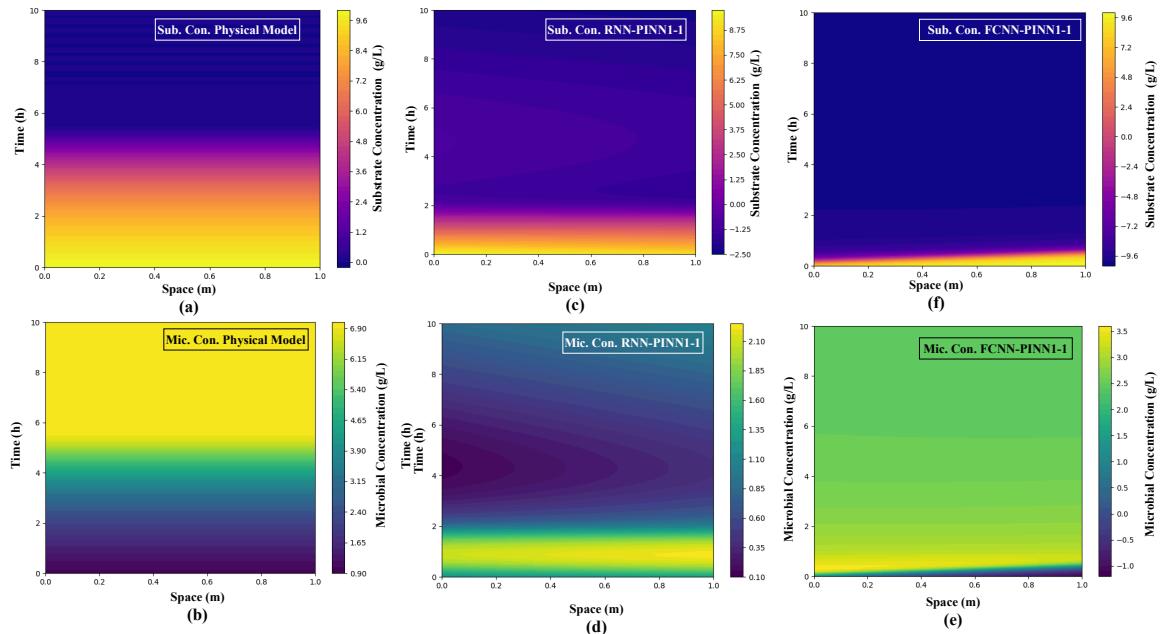


Fig. 8. 2D contour plots of substrate and microbial concentrations for scenario 1.

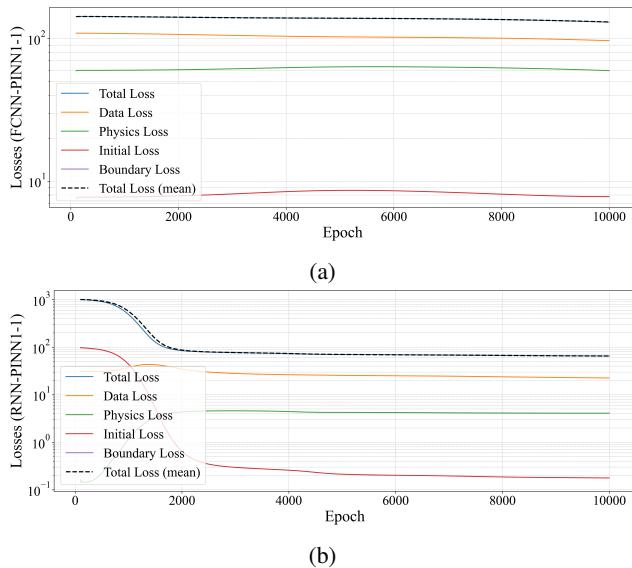


Fig. 9. Training loss curves for FCNN-PINN1-1 (top) and RNN-PINN1-1 (bottom), showing data loss, physics loss, initial condition loss, and total loss over training epochs.

B. Performance Evaluation in Scenario 2

1) *Performance and prediction evaluation of scenario 2:* Fig. 10 illustrates the results of scenario 2, comparing predictions for substrate and microbial concentrations across five PINN architectures: FCNN-PINN, ResBlock-PINN2-1, ResBlock-PINN2-2, RNN-PINN2-1, and RNN-PINN2-2. Each subfigure evaluates the impact of architectural configurations and loss weights on modeling dynamics. Fig. 10(a) demonstrates that the FCNN-PINN2-1 architecture accurately predicts substrate concentration, closely aligning with the physical model. This performance results from its fully con-

nected layers and higher loss weights for physical constraints ($\lambda_{\text{physics}} = 10.0$) and boundary conditions ($\lambda_{\text{boundary}} = 40.0$), ensuring strict adherence to governing equations. Similarly, Fig. 10(b) shows that microbial concentration predictions from FCNN-PINN2-1 maintain consistent alignment with the physical model, highlighting its robustness for applications requiring high physical fidelity.

ResBlock-PINN2-1 (Fig. 10(c)-(d)) demonstrates near-perfect agreement with the physical model in modeling substrate and microbial concentrations, benefiting from residual connections and moderately weighted loss terms ($\lambda_{\text{physics}} = 5.0, \lambda_{\text{boundary}} = 10.0$). ResBlock-PINN2-2 (Fig. 10(e)-(f)) achieves even greater precision due to higher loss weights, making it ideal for high-accuracy applications. RNN-PINN2-1 (Fig. 10(g)-(h)) efficiently captures temporal dynamics but deviates slightly from the physical model due to its lightweight design and lower loss weights ($\lambda_{\text{physics}} = 1.0, \lambda_{\text{boundary}} = 1.0$). RNN-PINN2-2 (Fig. 10(i)-(j)) emphasizes adaptability over precision, with very low loss weights. From these results, it becomes evident that ResBlock-PINN2-1 and ResBlock-PINN2-2 consistently deliver the best outputs.

Fig. 11 visualize substrate concentration dynamics within a reactor, comparing PINN predictions to the physical model. Fig. 11(a) represents the ground truth, showing smooth, uniform contour lines governed by physical equations. Fig. 11(b) shows FCNN-PINN2-1 predictions. Fig. 11(c) demonstrates ResBlock-PINN2-1, which aligns closely with the physical model, benefiting from residual connections that stabilize training and enhance detail resolution. Fig. 11(d) highlights ResBlock-PINN2-2's near-perfect alignment with the physical model, thanks to its high loss weights for physical and boundary constraints. Conversely, Fig. 11(e) depicts RNN-PINN2-1 predictions, capturing broad trends but failing in sharp transitions due to its lightweight design and lower loss weights. Fig. 11(f) illustrates RNN-PINN2-2, emphasizing

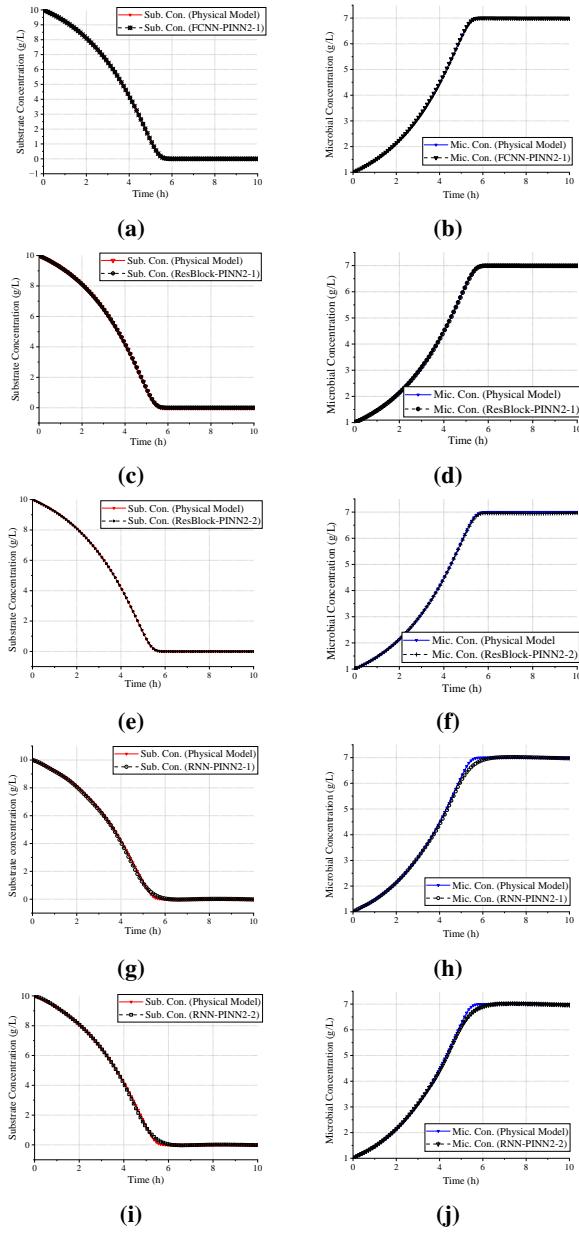


Fig. 10. Comparison of substrate and microbial concentration predictions across five PINN architecture.

adaptability but sacrificing precision, with very low loss weights.

Fig. 12 provides a comparison of microbial concentration predictions for the physical model and five PINN architectures. Fig. 12(a) represents the physical model, serving as the benchmark for evaluating the predictions from FCNN-PINN2-1, ResBlock-PINN2-1, ResBlock-PINN2-2, RNN-PINN2-1, and RNN-PINN2-2, shown in Fig. 12(b) through Fig. 12(f). Starting with Fig. 12(b), FCNN-PINN2-1 captures the overall trends in microbial concentration but shows slight deviations in areas with sharper transitions. This highlights the architecture's limited ability to resolve finer details due to the absence of residual connections. In contrast, Fig. 12(c) demonstrates ResBlock-PINN2-1's improved alignment with the physical model.

Fig. 12(d) showcases ResBlock-PINN2-2's exceptional precision, with contour lines closely matching those of the physical model. This is achieved through higher loss weights for physical constraints ($\lambda_{\text{physics}} = 10.0$) and boundary conditions ($\lambda_{\text{boundary}} = 20.0$), making it the most reliable architecture for high-fidelity modeling. Meanwhile, Fig. 12(e) shows that RNN-PINN2-1 captures broader trends but exhibits deviations in regions with steep transitions. Its lightweight design prioritizes computational efficiency, though it sacrifices some accuracy in contour alignment. Finally, Fig. 12(f) highlights RNN-PINN2-2's focus on adaptability.

Table V summarizes the initial and final values of total, data, physics, and initial loss across five PINN architectures, highlighting their convergence and optimization efficiency. FCNN-PINN2-1 and ResBlock-PINN2-2 start with identical total loss (1036.28), while ResBlock-PINN2-1 starts higher (2047.41). In contrast, RNN-PINN2-1 and RNN-PINN2-2 start lower at 131.64 and 80.27. ResBlock-PINN2-1 achieves the smallest final total loss (3.91×10^{-4}), followed by ResBlock-PINN2-2 (1.26×10^{-2}), with RNN models converging slower and less accurately. Table VII compares RMSE across architectures. ResBlock-PINN2-1 achieves perfect final RMSE (0.00), followed by ResBlock-PINN2-2 (0.03) and FCNN-PINN2-1 (0.025), showcasing strong predictive reliability.

2) *Performance analysis in scenario 2:* In summary, ResBlock-PINN2-1 emerges as the most accurate architecture, achieving the smallest final RMSE and MAE while maintaining the highest average R^2 . ResBlock-PINN2-2 also performs exceptionally well, striking a balance between accuracy and reliability. FCNN-PINN2-1 demonstrates strong performance but lags slightly behind the ResBlock architectures. RNN-PINN2-1 and RNN-PINN2-2 prioritize adaptability and computational efficiency, achieving satisfactory final metrics but showing higher initial errors and slower convergence. These results highlight the importance of choosing an appropriate architecture based on the trade-offs between precision and adaptability. Hence, scenario 2 rejects the assumption that a single NN architecture can universally address challenges of sparse data, nonlinear dynamics, and physical constraints. The results emphasize the critical role of architecture choice within the PINN framework, which demonstrates that each method exhibits unique strengths and limitations.

C. Computational Efficiency Comparison of PINN Architectures

Table VI summarizes the computational performance of the five PINN architectures tested under identical conditions. RNN-based models demonstrate the lowest complexity across all metrics: a reduction of over 82% in parameter count and FLOPs compared to FCNN-PINN2-1, along with training times under 500 seconds. Their memory footprint of just 0.01 MB makes them particularly suited for embedded IoT gateways where hardware is constrained. We calculated memory usage based on model weight sizes. The FCNN-PINN2-1 consumes approximately 0.08 MB of memory, ResBlock variants require about 0.06 MB, and the RNN-PINNs use

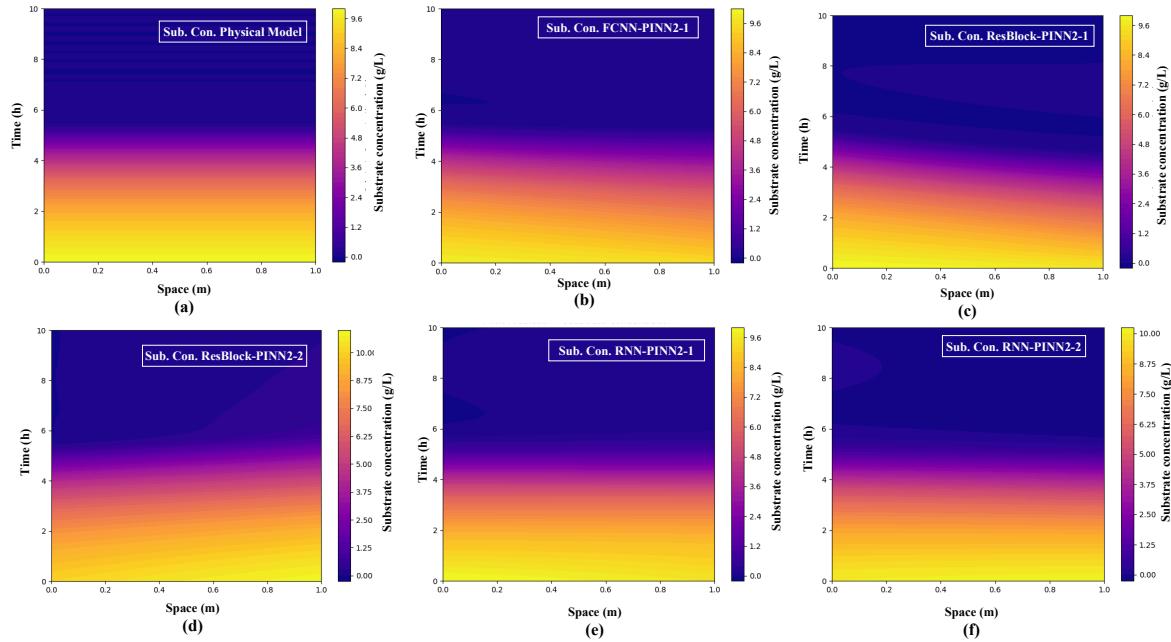


Fig. 11. The substrate concentration predictions across the physical model and PINN architectures.

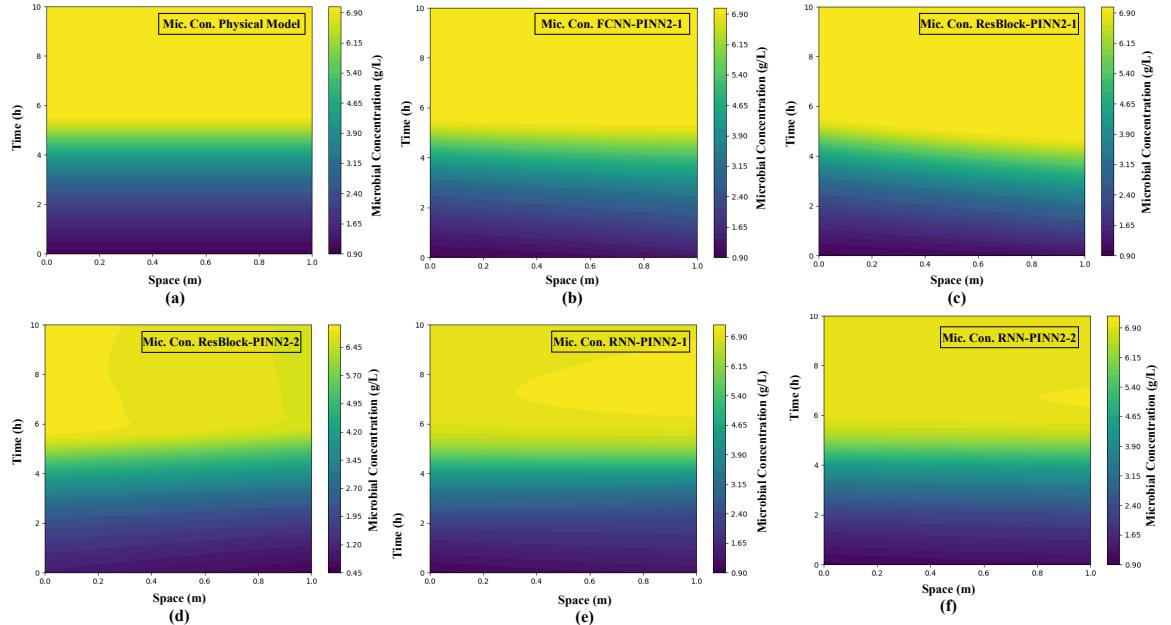


Fig. 12. The microbial concentration predictions across the physical model and PINN architectures.

TABLE V: Initial and final loss values for total, data, physics, and initial loss across five PINN architectures in scenario 2.

Metric	FCNN-PINN2-1	ResBlock-PINN2-1	ResBlock-PINN2-2	RNN-PINN2-1	RNN-PINN2-2
Total Loss (Initial)	1036.28	2047.41	1036.28	131.64	80.27
Total Loss (Final)	0.248	3.91×10^{-4}	1.26×10^{-2}	8.40×10^{-2}	4.53×10^{-2}
Data Loss (Initial)	26.00	27.24	26.00	30.34	25.61
Data Loss (Final)	6.1×10^{-4}	3.23×10^{-5}	1.19×10^{-3}	5.92×10^{-3}	4.03×10^{-3}
Physics Loss (Initial)	2.77×10^{-2}	8.10×10^{-3}	1.15×10^{-2}	3.00×10^{-1}	8.32×10^0
Physics Loss (Final)	2.48×10^{-2}	1.61×10^{-5}	1.14×10^{-3}	7.81×10^{-2}	8.24×10^{-2}
Initial Loss (Initial)	101.00	101.00	101.00	101.00	101.00
Initial Loss (Final)	1.24×10^{-5}	1.82×10^{-6}	2.44×10^{-6}	6.69×10^{-7}	4.45×10^{-6}

only 0.01 MB. This compactness of the RNNs permits direct deployment on microcontroller units without the need for quantization or pruning. Furthermore, the reduced model size minimizes bandwidth consumption during model updates in

federated setups and reduces inference delay for continuous physiological signal processing. Despite their simplicity, the RNN-PINNs retain the capability to approximate biological dynamics accurately, making them ideal candidates for effi-

TABLE VI: Comparison of PINN architectures in terms of parameters, FLOPs, training time, memory usage, and structure.

Method	Params	FLOPs	Time (s)	Mem. (MB)	Structure
FCNN-PINN2-1	22,262	44,102	729.19	0.08	6 Dense layers, 60 neurons per layer
ResBlock-PINN2-1	14,942	29,702	661.07	0.06	2 ResBlocks, 60 neurons per layer
ResBlock-PINN2-2	14,942	29,702	666.16	0.06	2 ResBlocks, 60 neurons per layer
RNN-PINN2-1	3,902	8,102	525.09	0.01	1 SimpleRNN layer, 60 neurons per layer
RNN-PINN2-2	3,902	8,102	482.89	0.01	1 SimpleRNN layer, 60 neurons per layer

TABLE VII: Performance metrics (RMSE, MAE, and R^2) across five PINN architectures at initial, final, and average stages.

Metric	FCNN-PINN2-1	ResBlock-PINN2-1	ResBlock-PINN2-2	RNN-PINN2-1	RNN-PINN2-2
RMSE (Initial)	4.86	4.92	4.92	5.47	5.07
RMSE (Final)	0.025	0.00	0.03	0.08	0.06
RMSE (Average)	4.70	0.60	1.86	0.51	0.52
MAE (Initial)	3.84	3.94	3.99	4.41	4.06
MAE (Final)	0.02	0.00	0.02	0.05	0.04
MAE (Average)	2.187	0.50	1.52	0.37	0.37
R^2 (Initial)	-1.25	-1.31	-1.31	-1.86	-1.45
R^2 (Final)	0.99	1.00	1.00	1.00	1.00
R^2 (Average)	-1.12	0.70	0.12	0.95	0.94

cient and deployable physics-informed modeling in biomedical applications.

VI. DISCUSSION

1) *Limitations and Future Extensions:* The IoBNT layer in the proposed framework functions as a nanoscale interface that enables real-time sensing and communication between the physical twin and the computational domain. IoBNT devices communicate via biochemical signaling pathways and molecular communication, exchanging information through molecules such as ions, hormones, or DNA. This fundamentally differs from traditional IoT systems, which rely solely on electromagnetic signals for data transmission [7]. While the full deployment of autonomous IoBNT systems in industrial bioprocess environments remains limited by technological constraints, several components of such architectures have been experimentally validated at the micro- and nanoscale. Recent advances have demonstrated functional prototypes involving DNA-based track-hopper systems for directional cargo transport, lipid vesicles for encapsulated sensing, and nanoparticle-assisted molecular communication [46]. These systems are capable of sensing biochemical variables such as substrate concentration and microbial density through enzymatic reactions or fluorescence-based mechanisms. Although the integration of these modules into a fully autonomous and scalable IoBNT infrastructure is still in progress, the proposed framework supports their use in controlled environments for simulation, modeling, and system design purposes.

Bio-NanoThings, as engineered biological entities, possess the inherent ability to reproduce and undergo genetic mutations over time. This natural tendency allows them to evolve selectively, potentially enhancing their performance or adapting to environmental changes [7]. Such autonomous evolution and mutation are intrinsic to biological systems and have no counterpart in conventional IoT devices. The IoBNT layer is designed with bio-nano sensors capable of detecting and quantifying critical bioprocess variables [10], particularly $S(t)$ and $X(t)$, as defined by the Monod equations. These sensors are designed to operate in continuous monitoring mode and are linked to molecular communication modules that employ

concentration-based modulation techniques. For instance, their operational performance demonstrates exceptionally low error rates, ensuring precise and reliable real-time biodata acquisition. This level of accuracy makes them not only suitable but indispensable for applications requiring rigorous tracking of bioprocess dynamics, enabling unprecedented levels of process control and predictive modeling in complex biological environments [9].

Additionally, similar to [8], [24], we consider that molecular communication channels that are latency-aware and exhibit limited signal degradation, as described in these works involving the blood-tissue barrier and intracellular environments. The use of threshold-optimized release mechanisms and simplified decoding protocols provides a realistic representation of near-future IoBNT performance under laboratory-controlled conditions. This simulation-driven modeling of the IoBNT layer supports a structured exploration of its impact on real-time DTs in biotechnology. By incorporating experimentally validated communication constraints and biological sensing mechanisms, our framework offers a forward-looking yet technically plausible pathway for deploying IoBNT-powered DTs. It also highlights the growing relevance of integrating physics-informed modeling with emerging bio-nano communication protocols to improve predictive accuracy and system responsiveness in bioprocess monitoring.

2) *Applicability Beyond IoBNT:* Although the proposed framework is designed with IoBNT characteristics in mind, its modularity allows it to be extended to other IoT domains. For instance, by replacing Monod kinetics with domain-relevant physical models, and adapting the neural network design to suit different signal structures, the framework can support applications in industrial sensing, environmental monitoring, and smart healthcare. However, such generalization requires careful modification, as each IoT application involves distinct noise characteristics, sensing dynamics, and communication constraints. We emphasize that while IoBNT serves as the core use case in this study, the methodology itself is extensible with proper domain adaptation.

3) *Model Updating and Aging:* The proposed framework supports continuous synchronization with the evolving phys-

ical system. The architecture can be updated through online learning or adaptive retraining as new sensor measurements become available. This capability enables the digital twin to reflect changes due to biological aging, operational drift, or external perturbations. As an example, if microbial growth slows over time due to environmental stress, the PINN structure can adjust its parameters while preserving consistency with the governing Monod kinetics.

4) *Uncertainty Quantification and Decision Support:* Although the current implementation focuses on deterministic predictions, the framework is extensible to support uncertainty quantification. By incorporating Monte Carlo PINNs, Bayesian PINNs, or ensemble-based PINNs, the system can capture both epistemic and aleatoric uncertainty [47]–[49]. These predictive distributions can be used to estimate confidence intervals and inform critical decision-making tasks, such as adjusting nutrient levels, scheduling maintenance, or predicting system failures.

VII. CONCLUSIONS

In this paper, we propose a novel IoT-enabled framework that seamlessly integrates physics-based modeling with data-driven intelligence to advance the deployment of DTs for micro- and nano-scale bioprocesses. Leveraging the IoBNT as the communication backbone, our architecture enables reliable, real-time sensing and data acquisition through embedded bio-nano sensors, addressing key IoT demands for low-latency, high-fidelity data transfer. The computational layer incorporates a multi-model PINN framework, comprising FCNNs, ResBlock NNs, and RNNs, to integrate domain-specific physical laws, such as Monod kinetics, into the learning process. This hybrid approach ensures physically consistent, scalable, and accurate modeling of complex biosystems, supporting IoT use cases such as smart environments and precision biomanufacturing. Experimental results demonstrate the framework's efficacy, achieving an average MAE of 0.0275, RMSE of 0.0425, and an R^2 score approaching 1 across four of the five tested network architectures. Additionally, the inclusion of IoBNT significantly enhances communication reliability, reducing data transmission errors by up to 98% under challenging conditions, thereby underscoring its potential for robust IoT applications involving real-time monitoring and control of bio-nano processes.

ACKNOWLEDGMENT

This research was supported by the Seatrium New Energy Laboratory, the Singapore Ministry of Education (MOE) Tier 1 grants (RT5/23 and RG24/24), the NTU Centre for Computational Technologies in Finance (NTU-CCTF), and the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award No. I2301E0026), administered by A*STAR.

APPENDIX A: TRAINING LOSS CURVES FOR SCENARIO 2

To support the interpretation of model performance across different architectures in Scenario 2, we present training loss progression plots for each configuration. Fig. 13(a)-(e) show

the evolution of four loss terms tracked during training: the total loss (sum of all weighted components), the supervised data loss (mean squared error between predicted and simulated values), the physics-informed loss (penalty on deviations from the Monod kinetics), and the initial condition loss (enforcing agreement with known initial values).

Each subplot highlights how these loss terms evolve depending on model architecture and assigned weightings. FCNN-PINN2-1 (Fig. 13(a)) demonstrates rapid reduction of the data loss but a relatively flat physics loss, suggesting a stronger emphasis on data fitting over physical consistency. ResBlock-PINN2-2 (Fig. 13(c)) shows simultaneous minimization of all loss terms, indicating more balanced optimization dynamics. In contrast, RNN-based architectures (Fig. 13(d) and Fig. 13(e)) exhibit slower and less stable convergence of the physics loss, likely due to their lighter structure and lower physics-related weightings. These visualizations support the conclusion that architectural form alone does not dictate performance. Rather, it is the interplay between architectural design and the configuration of loss weights that governs the balance between predictive accuracy and physical fidelity in PINN-based systems.

REFERENCES

- [1] P. Jia, X. Wang, and X. Shen, "Accurate and efficient digital twin construction using concurrent end-to-end synchronization and multi-attribute data resampling," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 4857–4870, 2023.
- [2] M. B. Jamshidi, S. Lotfi, H. Siahkamari, T. Blecha, J. Talla, and Z. Peroutka, "An intelligent digital twinning approach for complex circuits," *Applied Soft Computing*, vol. 154, p. 111327, 2024.
- [3] M. Basharat, M. Naeem, A. M. Khattak, and A. Anpalagan, "Digital-twin-assisted task offloading in uav-mec networks with energy harvesting for iot devices," *IEEE Internet of Things Journal*, vol. 11, no. 23, pp. 37550–37561, 2024.
- [4] B. N. Slautin, Y. Liu, H. Funakubo, R. K. Vasudevan, M. Ziatdinov, and S. V. Kalinin, "Bayesian conavigation: Dynamic designing of the material digital twins via active learning," *ACS nano*, vol. 18, no. 36, pp. 24 898–24 908, 2024.
- [5] K. R. Singh, V. Nayak, J. Singh, and R. P. Singh, "Nano-enabled wearable sensors for the internet of things (iot)," *Materials Letters*, vol. 304, p. 130614, 2021.
- [6] C. Li, J. Wang, S. Wang, and Y. Zhang, "A review of iot applications in healthcare," *Neurocomputing*, vol. 565, p. 127017, 2024.
- [7] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, and Y. Koucheryavy, "The internet of bio-nano things," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 32–40, 2015.
- [8] M. M. Al-Zubi, A. S. Mohan, P. Plapper, and S. H. Ling, "Intrabody molecular communication via blood-tissue barrier for internet of bio-nano things," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 802–21 810, 2022.
- [9] E. Shitiri and H.-S. Cho, "An m-ary concentration-shift keying with common detection thresholds for multitransmitter molecular communication," *IEEE Internet of Things Journal*, vol. 11, no. 10, pp. 17948–17959, 2024.
- [10] M. Jamshidi, D. T. Hoang, and D. N. Nguyen, "Cnn-fl for biotechnology industry empowered by internet-of-bionano things and digital twins," *IEEE Internet of Things Magazine*, vol. 7, no. 5, pp. 54–63, 2024.
- [11] A. Rana, D. Gautam, P. Kumar, and A. K. Das, "Architectures, benefits, security and privacy issues of internet of nano things: A comprehensive survey, opportunities and research challenges," *IEEE Communications Surveys & Tutorials*, 2024.
- [12] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
- [13] B. Shan, Y. Li, and S.-J. Huang, "Vi-pinn: Variance-involved physics-informed neural networks for fast and accurate prediction of partial differential equations," *Neurocomputing*, p. 129360, 2025.

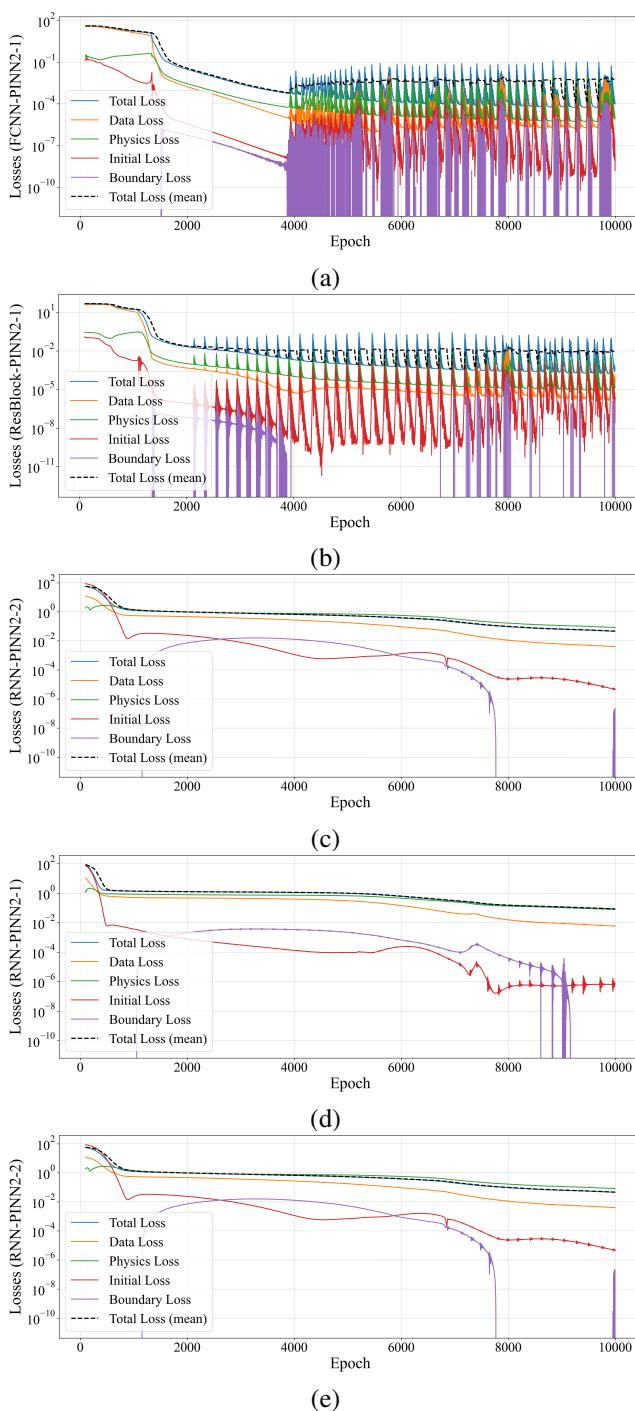


Fig. 13. Loss progression plots for all PINN architectures in Scenario 2: (a) FCNN-PINN2-1, (b) ResBlock-PINN2-1, (c) ResBlock-PINN2-2, (d) RNN-PINN2-1, and (e) RNN-PINN2-2.

- [14] Z. Meng, Q. Qian, M. Xu, B. Yu, A. R. Yildiz, and S. Mirjalili, "Pinn-form: a new physics-informed neural network for reliability analysis with partial differential equation," *Computer Methods in Applied Mechanics and Engineering*, vol. 414, p. 116172, 2023.
- [15] F. Stadtmann, E. R. Furevik, A. Rasheed, and T. Kvamsdal, "Physics-guided federated learning as an enabler for digital twins," *Expert Systems with Applications*, vol. 258, p. 125169, 2024.
- [16] X. Zhang, B. Mao, Y. Che, J. Kang, M. Luo, A. Qiao, Y. Liu, H. Anzai, M. Ohta, Y. Guo *et al.*, "Physics-informed neural networks (pinns) for

4d hemodynamics prediction: an investigation of optimal framework based on vascular morphology," *Computers in Biology and Medicine*, vol. 164, p. 107287, 2023.

- [17] E. A. Antonelo, E. Camponogara, L. O. Seman, J. P. Jordanou, E. R. de Souza, and J. F. Hübner, "Physics-informed neural nets for control of dynamical systems," *Neurocomputing*, vol. 579, p. 127419, 2024.
- [18] J. Chen, Y. Shi, C. Yi, H. Du, J. Kang, and D. Niyato, "Generative-ai-driven human digital twin in iot healthcare: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 34 749–34 773, 2024.
- [19] Y. Sun, W. Cheng, Q. Wang, K. Yang, and Y. Chen, "Advancing the internet of bio-nano things: A novel dna-based track-hopper system for enhanced efficiency and reliability," *IEEE Internet of Things Journal*, vol. 12, no. 4, pp. 4144–4157, 2025.
- [20] T. Muhammad and Z. Halim, "Employing artificial neural networks for constructing metadata-based model to automatically select an appropriate data visualization technique," *Applied Soft Computing*, vol. 49, pp. 365–384, 2016.
- [21] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, 2020.
- [22] A. Salakkam, N. Phukoethipim, P. Laopaiboon, and L. Laopaiboon, "Mathematical modeling of bioethanol production from sweet sorghum juice under high gravity fermentation: Applicability of monod-based, logistic, modified gompertz and weibull models," *Electronic Journal of Biotechnology*, vol. 64, pp. 18–26, 2023.
- [23] L. Huang, F. Liu, and L. Lin, "Molecular communication systems in cylindrical channels with non-newtonian fluid flows in iobnt," *IEEE Internet of Things Journal*, 2024.
- [24] H. Fouad, M. Hashem, and A. E. Youssef, "Retracted article: A nano-biosensors model with optimized bio-cyber communication system based on internet of bio-nano things for thrombosis prediction," *Journal of Nanoparticle Research*, vol. 22, pp. 1–17, 2020.
- [25] A. El-Fatyany, H. Wang, S. M. Abd El-atty, and M. Khan, "Biocyper interface-based privacy for internet of bio-nano things," *Wireless Personal Communications*, vol. 114, pp. 1465–1483, 2020.
- [26] I. F. Akyildiz, M. Ghovanloo, U. Guler, T. Ozkaya-Ahmadov, A. F. Sarioglu, and B. D. Unluturk, "Panacea: An internet of bio-nanothings application for early detection and mitigation of infectious diseases," *IEEE Access*, vol. 8, pp. 140 512–140 523, 2020.
- [27] C. Koca, M. Ozger, O. Cetinkaya, and O. B. Akan, "Information-theoretic lifetime maximization for iobnt-enabled sensing," *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, 2025.
- [28] S. Angerbauer, F. Enzenhofer, T. Pankratz, M. Hamidovic, A. Springer, and W. Haselmayr, "Novel nano-scale computing unit for the iobnt: Concept and practical considerations," *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, vol. 10, no. 4, pp. 549–565, 2024.
- [29] F. Salehi, J. Mohammadpour, R. Abbassi, S. Cheng, S. Diasinos, and R. Eaton, "Developing an interactive digital reality module for simulating physical laboratories in fluid mechanics," *Australasian Journal of Engineering Education*, vol. 27, no. 2, pp. 100–114, 2022.
- [30] M. Hassan, G. Montague, Z. Iqbal, and J. Fahey, "Virtual reality-based bioreactor digital twin for operator training," *Digital Chemical Engineering*, vol. 11, no. June, p. 100147, 2024.
- [31] J. Cao, X. Liu, X. Su, J. E. Hædahl, T. B. Fjellestad, D. Haziri, A. H.-A. Vu, J. Koskiaho, S. M. Karjalainen, A.-k. Ronkanen *et al.*, "Head-mounted display-based augmented reality for water quality visualisation," *Water Science and Engineering*, vol. 17, no. 3, pp. 236–248, 2024.
- [32] J. Alvarez-Ramirez, M. Meraz, and E. J. Vernon-Carter, "A theoretical derivation of the monod equation with a kinetics sense," *Biochemical Engineering Journal*, vol. 150, p. 107305, 2019.
- [33] Y. Liu, "Overview of some theoretical approaches for derivation of the monod equation," *Applied microbiology and biotechnology*, vol. 73, pp. 1241–1250, 2007.
- [34] H. Wang, H. Shi, K. Lin, C. Qin, L. Zhao, Y. Huang, and C. Liu, "A high-precision arrhythmia classification method based on dual fully connected neural network," *Biomedical Signal Processing and Control*, vol. 58, p. 101874, 2020.
- [35] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Neural window fully-connected crfs for monocular depth estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3916–3925.
- [36] P. Petersen and F. Voigtlaender, "Equivalence of approximation by convolutional neural networks and fully-connected networks," *Proceedings*

- of the American Mathematical Society, vol. 148, no. 4, pp. 1567–1581, 2020.
- [37] S. J. Anagnostopoulos, J. D. Toscano, N. Stergiopoulos, and G. E. Karniadakis, “Residual-based attention in physics-informed neural networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 421, p. 116805, 2024.
- [38] T. Shao, J. N. Shingala, A. Shyam, P. Yin, A. Suneja, S. P. Badya, A. Arora, and S. McCarthy, “Residual block fusion in low complexity neural network-based in-loop filtering for video compression,” in *2024 Data Compression Conference (DCC)*. IEEE, 2024, pp. 392–401.
- [39] Y. Tang, J. Fan, X. Li, J. Ma, M. Qi, C. Yu, and W. Gao, “Physics-informed recurrent neural network for time dynamics in optical resonances,” *Nature computational science*, vol. 2, no. 3, pp. 169–178, 2022.
- [40] T. M. Razakh, B. Wang, S. Jackson, R. K. Kalia, A. Nakano, K.-i. Nomura, and P. Vashishta, “Pnd: Physics-informed neural-network software for molecular dynamics applications,” *SoftwareX*, vol. 15, p. 100789, 2021.
- [41] P.-H. Chiu, J. C. Wong, C. Ooi, M. H. Dao, and Y.-S. Ong, “Can-pinn: A fast physics-informed neural network based on coupled-automatic-numerical differentiation method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 395, p. 114909, 2022.
- [42] L. Deng and Y. Pan, “Application of physics-informed neural networks for self-similar and transient solutions of spontaneous imbibition,” *Journal of Petroleum Science and Engineering*, vol. 203, p. 108644, 2021.
- [43] P. D. Adams, N. S. Pannu, R. J. Read, and A. T. Brünger, “Cross-validated maximum likelihood enhances crystallographic simulated annealing refinement,” *Proceedings of the National Academy of Sciences*, vol. 94, no. 10, pp. 5018–5023, 1997.
- [44] M. Adnan, A. A. S. Alarood, M. I. Uddin, and I. ur Rehman, “Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models,” *PeerJ Computer Science*, vol. 8, p. e803, 2022.
- [45] G. Ranjan, A. K. Verma, and S. Radhika, “K-nearest neighbors and grid search cv based real time fault monitoring system for industries,” in *2019 IEEE 5th international conference for convergence in technology (I2CT)*. IEEE, 2019, pp. 1–5.
- [46] Y. Sun, W. Cheng, Q. Wang, K. Yang, and Y. Chen, “Advancing the internet of bio-nano things: A novel dna-based track-hopper system for enhanced efficiency and reliability,” *IEEE Internet of Things Journal*, 2024.
- [47] S. Wang and G. E. Karniadakis, “Gmc-pinns: A new general monte carlo pinns method for solving fractional partial differential equations on irregular domains,” *Computer Methods in Applied Mechanics and Engineering*, vol. 429, p. 117189, 2024.
- [48] A. S. Nair, B. Jacob, A. A. Howard, J. Drgona, and P. Stinis, “E-pinns: Epistemic physics-informed neural networks,” *arXiv preprint arXiv:2503.19333*, 2025.
- [49] S. Lamichhane, M. Kavousi, and S. X.-D. Tan, “Bpinn-em: fast stochastic analysis of electromigration damage using bayesian physics-informed neural networks,” in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, 2024, pp. 1–9.



Mohammad (Behdad) Jamshidi (Senior Member, IEEE) received the Ph.D. degree in Engineering and Information Technology from the University of Technology Sydney (UTS), Sydney, Australia, in 2025. He is currently a Research Scientist with the School of Electrical and Data Engineering, UTS, and a member of the Radio Frequency and Communication Technologies (RFCT) Laboratory. Dr. Jamshidi has been listed among the world's top 2% of scientists by Stanford University in the field of Artificial Intelligence and Image Processing for four consecutive years (2022–2025). From 2017 to 2021, he made significant research contributions to aerospace component diagnostics of NASA's Prognostics Center of Excellence (PCoE) across multiple international institutions. His expertise spans artificial intelligence, neural networks, and digital twins. He has been recognized as an *Australian Global Talent in DigiTech* by the Australian Government since 2024 and as a *Highly Skilled Technology Developer* under the EU Blue Card Program since 2022. He has participated in numerous international research grants and scholarships and has authored more than 70 peer-reviewed publications in top-tier journals and IEEE conferences, several of which are highly cited or recognized as hot papers.

consecutive years (2022–2025). From 2017 to 2021, he made significant research contributions to aerospace component diagnostics of NASA's Prognostics Center of Excellence (PCoE) across multiple international institutions. His expertise spans artificial intelligence, neural networks, and digital twins. He has been recognized as an *Australian Global Talent in DigiTech* by the Australian Government since 2024 and as a *Highly Skilled Technology Developer* under the EU Blue Card Program since 2022. He has participated in numerous international research grants and scholarships and has authored more than 70 peer-reviewed publications in top-tier journals and IEEE conferences, several of which are highly cited or recognized as hot papers.



Dinh Thai Hoang (M'16, SM'22) is currently a faculty member at the School of Electrical and Data Engineering, University of Technology Sydney, Australia. He received his Ph.D. in Computer Science and Engineering from Nanyang Technological University, Singapore, in 2016. His research interests include emerging wireless communications and networking topics, especially machine learning applications in networking, edge computing, and cybersecurity. He has received several prestigious awards, including the Australian Research Council Discovery Early Career Researcher Award, IEEE TCSC Award for Excellence in Scalable Computing for Contributions on “Intelligent Mobile Edge Computing Systems” (Early Career Researcher), IEEE Asia-Pacific Board (APB) Outstanding Paper Award 2022, and IEEE Communications Society Best Survey Paper Award 2023. He is currently an Editor of IEEE TMC, IEEE TWC, IEEE TCOM, and IEEE TNSE.



Diep N. Nguyen (Senior Member, IEEE) received the M.E. degree in electrical and computer engineering from the University of California at San Diego (UCSD), La Jolla, CA, USA, in 2008, and the Ph.D. degree in electrical and computer engineering from The University of Arizona (UA), Tucson, AZ, USA, in 2013. He is currently the Head of 5G/6G Wireless Communications and Networking Lab, Director of Agile Communications and Computing group, Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Sydney, Australia. Before joining UTS, he was a DECRA Research Fellow with Macquarie University, Macquarie Park, NSW, Australia, and a Member of the Technical Staff with Broadcom Corporation, CA, USA, and ARCON Corporation, Boston, MA, USA, and consulting the Federal Administration of Aviation on turning detection of UAVs and aircraft, and the U.S. Air Force Research Laboratory, USA, on anti-jamming. His research interests include computer networking, wireless communications, and machine learning application, with emphasis on systems' performance and security/privacy. Dr. Nguyen received several awards from U.S. Congress, the U.S. National Science Foundation, and the Australian Research Council. He has served on the Editorial Boards of the IEEE Transactions on Mobile Computing, IEEE Communications Surveys & Tutorials (COMST), IEEE Open Journal of the Communications Society.



Dusit Niyato (Fellow, IEEE) is a professor in the College of Computing and Data Science, at Nanyang Technological University (NTU), Singapore. He received B.Eng. from King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand and Ph.D. in Electrical and Computer Engineering from the University of Manitoba, Canada. His research interests are in the areas of mobile generative AI, edge general intelligence, quantum computing and networking, and incentive mechanism design.



Majid E. Warkiani is a Professor in the School of Biomedical Engineering at the University of Technology Sydney (UTS). He received his PhD from Nanyang Technological University (Singapore) and completed postdoctoral training at MIT (USA). His multidisciplinary research integrates microfluidics, organ-on-chip systems, 3D microfabrication, and artificial intelligence (AI) for applications in cancer diagnostics, reproductive medicine, and regenerative biology. A recipient of the NHMRC and Cancer Institute NSW Fellowships, and awards such as the MIT Innovators Under 35 and Young Tall Poppy Science Award, Dr. Warkiani has authored over 270 peer-reviewed papers in top-tier journals including Nature Communications, Advanced Materials, and Lab on a Chip, with more than 16,000 citations and an h-index of 66 on Google Scholar.