Journal of Mathematical Modeling
Vol. -, No. -, 2026, pp. -. Research Article

JMM

A deep learning PINN-LSTM fusion framework for solving ordinary differential equations from real-life systems

Mohamed Ashik Shahul[1], Tharmalingam Gunasekar[1,2*], Shyam Sundar Santra[3 *], Dumitru Baleanu[4,5] and Yakup Yildirim[6,7]

[1]*Department of Mathematics, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India.*
[2]*School of Science, Department of Mathematics and Computer Science, St. Francis de Sales College (Autonomous), Electronics City, Bengaluru, Karnataka, India.*
[3]*Department of Mathematics, JIS College of Engineering, Kalyani, West Bengal 741235, India.*
[4]*Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon.*
[5]*Institute of Space Sciences - Subsidiary of INFLPR, Magurele-Bucharest, 077125 Magurele, Romania.*
[6]*Department of Computer Engineering, Biruni University, Istanbul–34010, Turkey.*
[7]*Mathematics Research Center, Near East University, 99138 Nicosia, Cyprus.*

Email(s): mohammedashik1303@gmail.com, tguna84@gmail.com, shyam01.math@gmail.com, dumitru.baleanu@gmail.com, yyildirim@biruni.edu.tr

Abstract. This article presents a comparative study of Physics-Informed Neural Networks (PINNs), Long Short Term Memory (LSTM) networks and adaptive Fusion of PINNs and LSTM for solving ordinary differential equations arising in real-world problems. After brief preliminaries on PINNs and LSTM networks, we formulate PINN and LSTM frameworks and apply each to two benchmark systems: A religious growth population problem and Kuramoto's two oscillator problem. We then introduce an integration strategy that fuses PINNs physics-constrained learning with LSTMs sequential modelling to improve stability and accuracy. Extensive numerical experiments and error analysis that demonstrates the proposed fusion architecture consistently outperforms standalone PINN and tuned LSTM models in accuracy and robustness across the two problems. The results indicate the fusion approach as a promising direction for enhanced data-efficient and physically consistent solutions of complex dynamical systems.

Keywords: Neural networks, physics-informed neural networks, long short-term memory networks, differential equations, Kuramoto model.
AMS Subject Classification 2010: 34A30, 34A34, 65D25, 65L05.

*Corresponding author
Received: 13 December 2025/ Revised: 19 June 2026/ Accepted: 01 July 2026
DOI: 10.22124/jmm.2026.32516.2949

1 Introduction

Physics-Informed Neural Networks (PINNs) have emerged as a powerful deep learning framework for solving differential equations by embedding governing physical laws directly into the loss function, thereby enabling the learning of solutions without the need for large training datasets [13, 22, 23, 30]. Extensive developments over recent years have demonstrated the capability of PINNs to solve a wide spectrum of real world problems including nonlinear fluid mechanics [4, 5, 18], electromagnetic systems [12, 28], population dynamics [8], optimization and control [1, 2], and PDE constrained problems [17, 19]. Despite these advances, PINNs often encounter training challenges such as stiffness, slow convergence and difficulty in capturing long-term temporal dependencies, particularly in nonlinear coupled systems and oscillatory phenomena [25, 27]. Furthermore, the versatility of PINNs has been demonstrated in complex biological systems, such as the combination of PINNs with the finite volume method for parameter identification and model selection in cell invasion models [21].

On the other hand, Long Short Term Memory (LSTM) networks have shown strong performance in modeling sequential data and learning long-range dependencies by leveraging recurrent gating mechanisms [10, 20]. The LSTMs have been successfully utilized for population forecasting [11, 24, 26] and for modeling synchronization phenomena in nonlinear oscillator systems such as the Kuramoto model [9, 15, 31]. However, purely data-driven networks, including LSTMs, lack inherent physical consistency and often require large volumes of training data, which may not always be available in scientific and engineering applications.

This motivates recent interest in fused physics-guided architectures that combine the strengths of PINNs and recurrent networks to improve both accuracy and temporal representation [3, 16, 29]. Fusion of PINNs and LSTMs frameworks has shown promise in capturing spatiotemporal dynamics in complex systems, reducing data dependency and enhancing generalization performance in long-term prediction [6, 14]. The idea of fusing PINNs with recurrent architectures has been explored in recent literature such as the PhyLSTM framework proposed by [7]. Unlike hybrid approaches that use complex gating networks, the proposed adaptive fusion framework combines the strengths of PINN and LSTM models for solving nonlinear ordinary differential equations (ODEs). A learnable scalar weight dynamically balances physics-based constraints and data-driven temporal features while maintaining computational efficiency. Specifically, we utilise a learnable parameter (α) that dynamically balances the physics-based regularizer and the data-driven sequential learner during training. This straightforward convex combination reduces computational overhead while maintaining high fidelity and physical consistency, positioning it as a highly efficient alternative to more complex fusion strategies. Motivated by these developments, the present work proposes a novel unified fusion framework that integrates the physics driven regularization capability of PINNs with the temporal sequence learning behaviour of LSTM networks.

The contributions of this study are threefold: (i) definition and formulation of the standalone PINNs (Section 2) and LSTM (Section 3) architectures for solving real life ODEs systems, (ii) development of a fused PINN and LSTM architecture with a dedicated fusion function for adaptive balance between physics constraints and data driven temporal learning (Section 4), (iii) demonstration of the proposed method on two fundamental real world applications, namely religious population growth modeling and Kuramoto's two-coupled nonlinear oscillator system, illustrating improved prediction accuracy and stability over traditional single architecture approaches (Section 5). The proposed fused framework aims to address the limitations of existing PINN and LSTM models by achieving physically consistent, data-

efficient and dynamically stable solutions for nonlinear ODE systems.

2 The PINNs

Definition 1. *The PINNs are a class of neural network models that incorporate physical knowledge typically expressed through differential equations, directly into the training process. Rather than relying solely on data, PINNs embed governing equations, initial conditions and observational information in their optimization objective. This enables them to approximate solutions that adhere to fundamental physical principles [12, 15].*

2.1 Architecture of PINNs

The PINNs merge machine learning with domain-specific physical constraints generally described by ODEs. The fundamental concept is to impose the physical laws governing a system as constraints during neural network training, as illustrated in Figure 1.

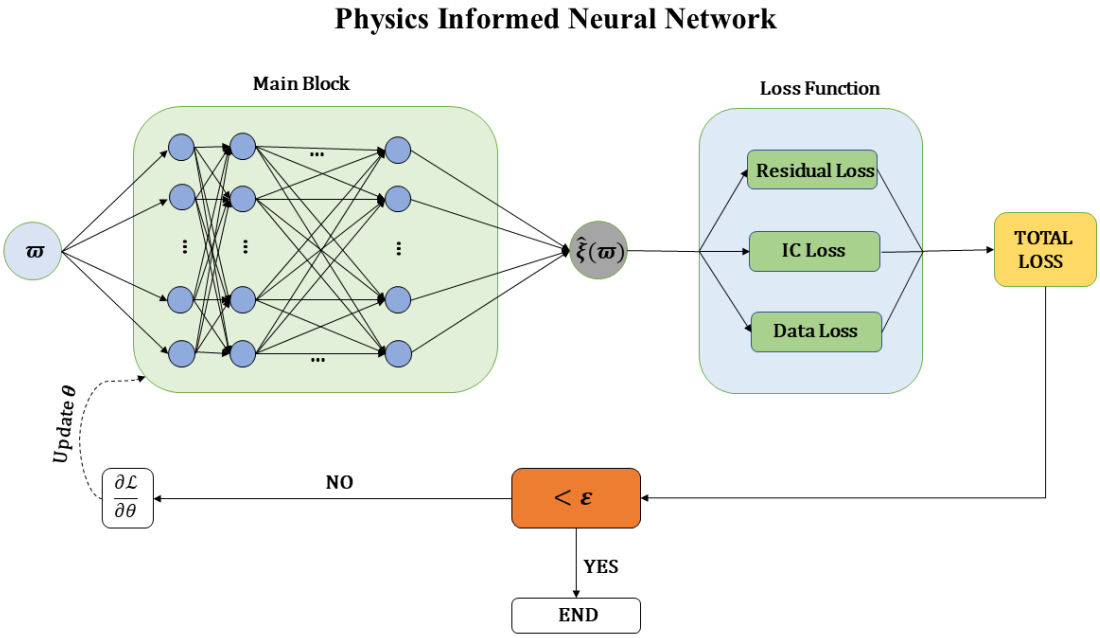

Figure 1: General structure of PINNs

For an ODE constrained optimization problem, the solution $\xi(\varpi)$ represents the system state where ϖ denotes spatial coordinates. The governing differential equation is typically written as

$$f(\varpi, \xi, \lambda) = 0, \quad \xi \in \Omega, \ \varpi \in [0, T], \tag{1}$$

$$\xi(0) = \mu(\varpi), \quad \xi \in \Omega,$$

where $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]$ denotes the parameters of the model, f represents the ODE residual and Ω is the computational domain. The prescribed initial condition is given by $\mu(\varpi)$.

To approximate the unknown solution $\xi(\varpi)$, a neural network $\hat{\xi}(\varpi;\theta)$ is utilized, where θ refers to the trainable parameters. The ODE constraints are incorporated into a composite loss function composed of both data consistency and physics-based terms. The total PINN loss function is expressed as

$$\mathscr{L}_{\text{PINN}} = \mathscr{L}_{\text{Physics}} + \mathscr{L}_{\text{Data}}, \tag{2}$$

where the physics-related contribution is defined by

$$\mathscr{L}_{\text{Physics}} = w_r \mathscr{L}_r(\theta;\mathscr{N}_r) + w_i \mathscr{L}_i(\theta;\mathscr{N}_i) \tag{3}$$

with

$$\mathscr{L}_r(\theta;\mathscr{N}_r) = \frac{1}{|\mathscr{N}_r|} \sum_{\varpi_r \in \mathscr{N}_r} \left| f(\varpi_r, \xi, \lambda) \right|^2, \tag{4}$$

$$\mathscr{L}_i(\theta;\mathscr{N}_i) = \frac{1}{|\mathscr{N}_i|} \sum_{\varpi_i \in \mathscr{N}_i} \left| \hat{\xi}(0;\theta) - \mu(\varpi_i) \right|^2. \tag{5}$$

Here w_r and w_i represent weighting coefficients while $\mathscr{N}_r$ and $\mathscr{N}_i$ denote collocation sets in the domain and on the initial condition, respectively. The data-driven loss term measures the discrepancy between predictions and observed data as follows

$$\mathscr{L}_{\text{Data}} = \frac{1}{|\mathscr{N}_d|} \sum_{\varpi_d \in \mathscr{N}_d} \left| \hat{\xi}(\varpi_d;\theta) - \xi(\varpi_d) \right|^2, \tag{6}$$

where $\xi(\varpi_d)$ denotes known observational measurements. For the numerical experiments presented in this study, the reference data $\xi(\varpi_d)$ for the religious population growth problem was generated using its exact analytical solution augmented with 1% Gaussian noise to simulate real-world sensor inaccuracies. Conversely, the reference data for the Kuramoto problem was generated using a high-fidelity 4th-order Runge-Kutta (RK4) numerical integration method.

2.1.1 Activation functions

Activation functions strongly influence model expressiveness and differentiability, both of which are essential for solving differential equations through PINNs [3–5]. In this framework, we specifically utilize the Hyperbolic Tangent (Tanh) activation function for the PINN hidden layers. The Tanh function provides a smooth, zero-centred output range and possesses non-vanishing, continuous second-order derivatives, which are strictly required for accurately computing the governing ODE residuals via automatic differentiation.

- **Hyperbolic tangent (Tanh) function:** The output range is $(-1,1)$, making it zero-centred and more suitable for hidden layers than sigmoid. It is defined as follows

$$\tanh(\varpi) = \frac{e^{\varpi} - e^{-\varpi}}{e^{\varpi} + e^{-\varpi}}. \tag{7}$$

2.1.2 Loss function

In the PINNs framework, the loss function $\mathscr{L}_{\text{PINN}}$ is essential since it embeds differential equation constraints directly into the neural network training protocol, unlike traditional learning approaches that depend solely on labelled data. The PINN loss enforces the governing ODE along with associated initial conditions. This allows the model to learn solutions that remain physically meaningful even when observations are limited or noisy [3–5].

2.2 Application problems

The religious population growth model describes the temporal evolution of a population group under the combined effects of intrinsic growth and external influences such as conversion, migration or recruitment [11].

Example 1. The PINNs formulation described in Subsection 2.1 is applied to the religious population growth problem. The governing differential equation is given by

$$\frac{d\phi}{dt} = \kappa\phi + \rho\ , \qquad \phi(0) = \phi_0, \qquad t \in [0,T], \tag{8}$$

where $\phi(t)$ denotes the religious population at time t, κ represents the intrinsic population growth rate and ρ denotes an external contribution term accounting for factors such as conversion, migration or recruitment, following the demographic modelling approach discussed in [11]. Following the general PINN framework, the neural network $\xi_{\text{PINN}}(t;\boldsymbol{\theta})$ is employed to approximate the unknown solution $\phi(t)$. The physics-informed residual associated with the governing differential equation is defined as

$$\mathscr{R}(t;\boldsymbol{\theta}) = \partial_t \xi_{\text{PINN}}(t;\boldsymbol{\theta}) - (\kappa\xi_{\text{PINN}}(t;\boldsymbol{\theta}) + \rho)\,. \tag{9}$$

The residual term together with the prescribed initial condition is incorporated into the loss function described in Subsection 2.1.2. The network parameters are then optimized using the Adam optimizer until convergence is achieved. After training the PINN approximation is expressed as

$$\phi_{\text{PINN}}(t) = \xi_{\text{PINN}}(t;\boldsymbol{\theta}^\star), \tag{10}$$

where $\boldsymbol{\theta}^\star$ denotes the optimal network parameters. The predictive accuracy of the trained PINN is evaluated using the Mean Squared Error (MSE) between the PINN solution and the analytical solution.
The PINN framework described in Subsection 2.1 was employed to solve the religious population growth problem. The governing differential equation was incorporated into the physics-informed loss function through the corresponding residual term, while the prescribed initial condition was enforced using an additional penalty term. The model was trained for 10,000 epochs using the Adam optimizer. The training progress is summarized in Table 1. It summarizes the evolution of the PINN loss components throughout the training process for the religious population growth problem. The reported values illustrate the optimization behavior of the model by tracking the Physics-Informed residual loss ($\mathscr{L}_{ODE}$), the initial condition loss ($\mathscr{L}_{IC}$) and the overall PINN loss ($\mathscr{L}_{PINN}$). While these metrics provide insight into the convergence characteristics of the training procedure they should not be interpreted as the sole indicators of predictive performance. As demonstrated in this study a reduction in training loss does not necessarily imply improved generalization or solution accuracy. Therefore the training loss results are

Table 1: Training progress of PINN model for the religious population growth problem over 10,000 epochs

Epoch	$\mathscr{L}_{\text{ODE}}$	$\mathscr{L}_{\text{IC}}$	$\mathscr{L}_{\text{PINN}}$
0	1.075×10^2	2.455×10^2	2.436×10^2
1000	2.261×10^2	8.254×10^2	2.241×10^2
2000	2.230×10^2	7.884×10^2	2.211×10^2
3000	2.228×10^2	7.852×10^2	2.209×10^2
4000	2.225×10^2	7.842×10^2	2.206×10^2
5000	2.250×10^2	2.228×10^2	2.456×10^2
6000	2.027×10^1	1.747×10^1	1.933×10^1
7000	2.211×10^1	7.694×10^1	2.192×10^1
8000	2.206×10^1	7.948×10^1	2.187×10^1
9000	2.202×10^1	7.747×10^1	2.183×10^1
9999	2.179×10^1	7.834×10^1	2.180×10^1

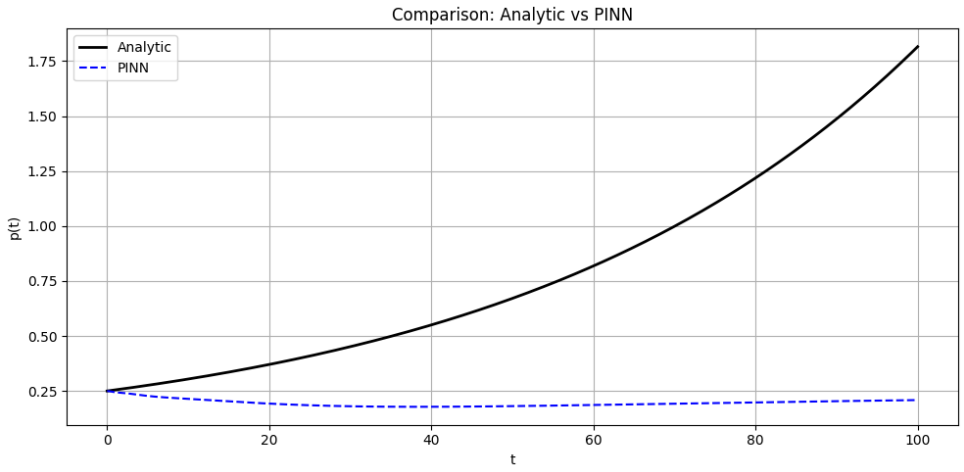

Figure 2: Graphical representation of the PINN model for the religious population growth problem over 10,000 epochs

complemented with test MSE evaluations and comparative performance analyses presented in later sections, providing a more comprehensive assessment of the model's predictive capability and robustness. The predictive capability of the trained PINN model was evaluated by comparing its solution with the corresponding analytical solution of the population growth problem shown in Figure 2. To further examine the optimization behavior of the PINN model, the absolute error of the loss function is illustrated in Figure 3. The graphical results presented in Figure 2 demonstrate the application of the standalone PINN approach to the population growth system and Error metric analysis. Although the PINN model in Table 1 demonstrates a continuously decreasing training loss reaching approximately 2.180×10^1 in Figure 3 reveals a significant discrepancy between the predicted output and the exact analytical solution. This failure to capture the unbounded exponential growth curve is attributed to the "spectral bias" of standard neural networks. The PINN converges to a trivial flat local minimum that minimizes the ODE residual locally but fails to generalise the exponential trajectory over long time horizons. This inherent

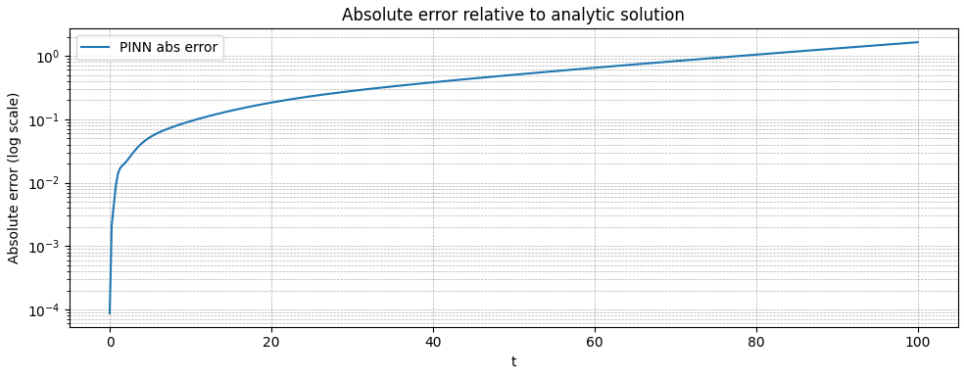

Figure 3: Absolute error of PINN model for the religious population growth problem

limitation in modelling unbounded long-term dynamics serves as a primary motivation for our proposed fusion with LSTM networks.

Example 2. Consider the Kuramoto model of two-coupled phase oscillators [9, 15, 31]

$$\begin{aligned} \dot{\phi}_1(t) &= \omega_1 + \frac{\kappa}{2}\sin\big(\phi_2(t) - \phi_1(t)\big), \\ \dot{\phi}_2(t) &= \omega_2 + \frac{\kappa}{2}\sin\big(\phi_1(t) - \phi_2(t)\big), \end{aligned} \tag{11}$$

with initial conditions $\phi_1(0), \phi_2(0) \in [0, 2\pi)$, natural frequencies $\omega_1, \omega_2 \in \mathbb{R}$ and coupling constant $\kappa \geq 0$.

Define the phase difference

$$\varphi(t) = \phi_2(t) - \phi_1(t). \tag{12}$$

Differentiate (12) and substitute (11), then we obtain

$$\begin{aligned} \dot{\varphi}(t) &= \dot{\phi}_2(t) - \dot{\phi}_1(t) \\ &= \Big(\omega_2 + \tfrac{\kappa}{2}\sin(\phi_1 - \phi_2)\Big) - \Big(\omega_1 + \tfrac{\kappa}{2}\sin(\phi_2 - \phi_1)\Big). \end{aligned} \tag{13}$$

Using the identity $\sin(\phi_1 - \phi_2) = -\sin(\phi_2 - \phi_1)$, we obtain the scalar autonomous ODE for ϕ

$$\dot{\varphi}(t) = \Delta\omega - \kappa\sin\phi(t)\,, \qquad \Delta\omega = \omega_2 - \omega_1. \tag{14}$$

Equilibria φ^* satisfy $0 = \Delta\omega - \kappa\sin\varphi^*$, hence

$$\sin\varphi^* = \frac{\Delta\omega}{\kappa}. \tag{15}$$

A real solution exists if and only if $|\Delta\omega| \leq \kappa$. For $|\Delta\omega| < \kappa$ select the principal solution

$$\varphi^* = \arcsin\left(\frac{\Delta\omega}{\kappa}\right), \tag{16}$$

which yields $\cos\varphi^* = \sqrt{1 - (\Delta\omega/\kappa)^2} > 0$. Linearize (14) about φ^* by writing $\varphi(t) = \phi^* + \delta\phi(t)$ and retaining first order terms

$$\delta\dot{\varphi}(t) \approx -\kappa\cos\varphi^*\,\delta\varphi(t). \tag{17}$$

Since $\kappa\cos\varphi^* > 0$ for $|\Delta\omega| < \kappa$, the equilibrium φ^* is asymptotically stable. Consequently,

- If $|\Delta\omega| \leq \kappa$, then $\varphi(t) \to \varphi^*$ as $t \to \infty$ and therefore $\dot{\phi}_1 - \dot{\phi}_2 = -\dot{\varphi} \to 0$.

- If $|\Delta\omega| > \kappa$, no fixed point exists and $\phi(t)$ drifts (no phase-locking).

This yields $|\dot{\phi}_1(t) - \dot{\phi}_2(t)| \to 0$ as $t \to \infty$ if and only if $|\omega_1 - \omega_2| \leq \kappa$.

The general PINNs framework described in Subsection 2.1 is employed to solve the Kuramoto two oscillator system. The neural network is used to approximate the phase variables $\phi_1(t)$ and $\phi_2(t)$ while the governing differential equations in (11) are incorporated into the training process through physics-informed residuals. The residual terms corresponding to the Kuramoto dynamics are defined as

$$\begin{aligned} \mathscr{R}_1(t;\theta) &= \partial_t \hat{\phi}_1(t;\theta) - \omega_1 - \tfrac{\kappa}{2}\sin\left(\hat{\phi}_2(t;\theta) - \hat{\phi}_1(t;\theta)\right), \\ \mathscr{R}_2(t;\theta) &= \partial_t \hat{\phi}_2(t;\theta) - \omega_2 - \tfrac{\kappa}{2}\sin\left(\hat{\phi}_1(t;\theta) - \hat{\phi}_2(t;\theta)\right). \end{aligned} \tag{18}$$

The residual terms and prescribed initial conditions are incorporated into the PINN loss function following the formulation presented in Subsection 2.1. Automatic differentiation is used to compute the required temporal derivatives and the network parameters are optimized using the Adam optimizer. After training the predictive accuracy of the PINN model is evaluated using the MSE between the PINN predictions and the reference numerical solution. In addition synchronization behavior is assessed through the phase difference $\varphi(t)$ and the frequency difference $\dot{\phi}_1(t) - \dot{\phi}_2(t)$, which are expected to approach zero in the synchronized regime ($|\omega_1 - \omega_2| \leq \kappa$).

Table 2: Training Progress of PINN model for the Kuramoto model of two-coupled phase oscillators problem over 10,000 epochs

Epoch	$\mathscr{L}_{\text{ODE}}$	$\mathscr{L}_{\text{IC}}$	$\mathscr{L}_{\text{PINN}}$
0	2.955×10^{0}	9.270×10^{-2}	1.222×10^{1}
1000	1.070×10^{0}	1.635×10^{-8}	1.070×10^{0}
2000	5.298×10^{-1}	1.743×10^{-7}	5.298×10^{-1}
3000	3.348×10^{-1}	1.137×10^{-8}	3.348×10^{-1}
4000	1.689×10^{-1}	4.917×10^{-8}	1.689×10^{-1}
5000	1.259×10^{-1}	1.117×10^{-5}	1.270×10^{-1}
6000	6.119×10^{-2}	1.882×10^{-8}	6.119×10^{-2}
7000	5.047×10^{-2}	3.391×10^{-8}	5.047×10^{-2}
8000	2.193×10^{-2}	3.553×10^{-8}	2.193×10^{-2}
9000	1.818×10^{-2}	5.475×10^{-9}	1.818×10^{-2}
9999	1.675×10^{-2}	1.227×10^{-8}	1.675×10^{-2}

Figure 4 illustrates the corresponding temporal evolution of the predicted states highlighting the model's capability to accurately reconstruct the model.

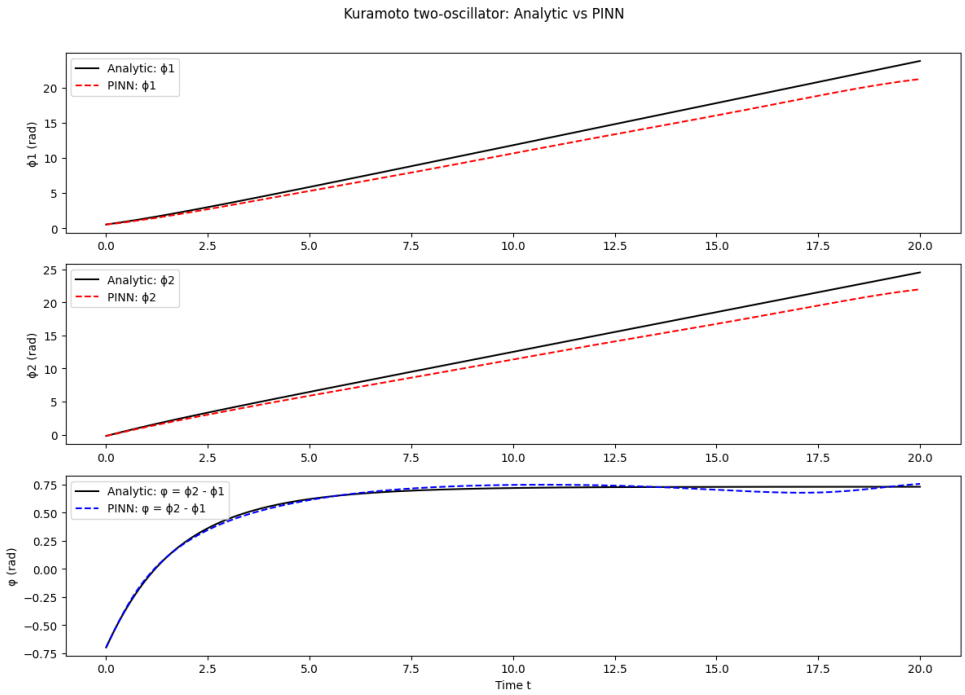

Figure 4: Graphical representation of the Kuramoto model of two-coupled phase oscillators problem over 10,000 epochs

The graphical results show that PINNs were successfully implemented to solve the two-coupled Kuramoto oscillator system. The predicted solution closely matches the expected dynamics, and the graphical results demonstrate accurate phase evolution for both oscillators.

It is worth noting the discrepancy in absolute loss values between the Growth problem and the Kuramoto model. While the Kuramoto PINN exhibits higher absolute loss values (e.g., 1.675×10^{-2}) as shown in Table 2, the resulting visual and physical error Figure 4 is substantially lower compared to the Growth problem. This occurs because the absolute magnitude of the loss is relative to the mathematical scale of the state variables. The Kuramoto model features bounded oscillatory dynamics governed by sine functions, which Tanh-based neural networks are structurally well-suited to fit globally, unlike unbounded exponential curves.

3 The LSTM

Definition 2. *A recurrent neural network (RNN) is a neural model developed to analyze sequential data by maintaining a hidden state h_t that captures dependencies from earlier inputs. Unlike traditional feed forward neural networks, RNNs contain recurrent connections that allow information to persist across time steps. This characteristic enables RNNs to handle tasks involving time series signals, dynamical systems and natural language processing [3]. So, $h_t = \xi(h_{t-1}, \varpi_t)$.*

Definition 3. *An improved RNN that efficiently learns long-term relationships within sequential or time-dependent data is called an LSTM network. By adding memory cells and gating procedures to regulate*

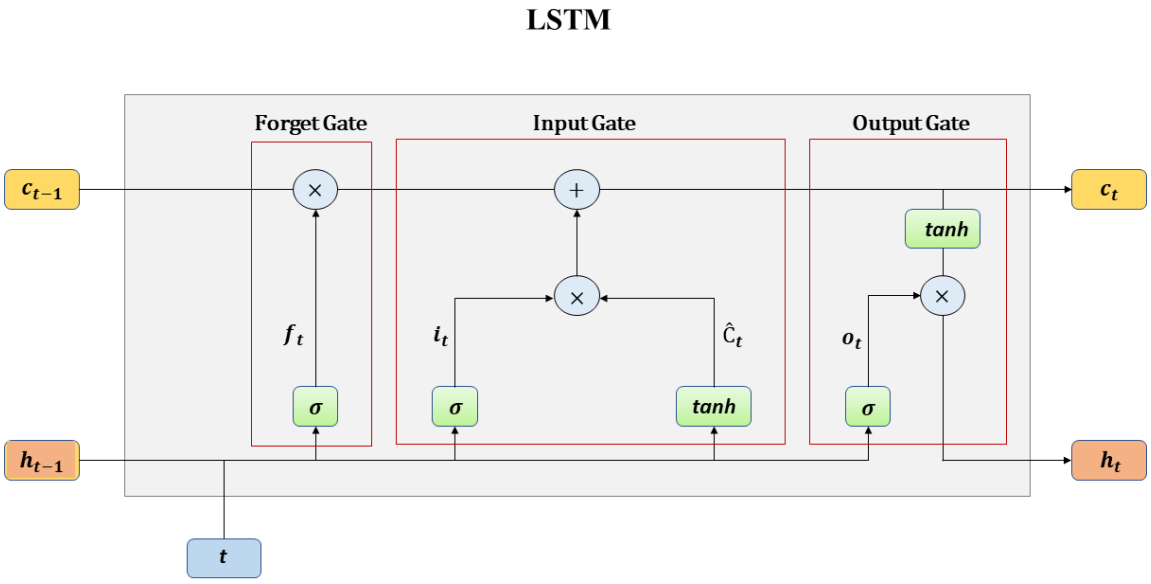

Figure 5: General structure of an LSTM network

information flow over various time steps it solves the disappearing and expanding gradient problems in traditional RNNs [10].

3.1 Architecture of LSTM

The LSTM architecture differs from standard RNNs by using multiple gates within each LSTM cell to regulate information as illustrated in Figure 5. These gates allow the network to retain relevant information over long sequences. An LSTM cell contains two internal states: the hidden state h_t and the cell state C_t. The cell state acts as a long-term memory conduit, preserving information through time.

The update of h_t and C_t depends on the current input ϖ_t along with the previous states h_{t-1} and C_{t-1}. Three gates control these updates: the input gate, forget gate and output gate.

Let i_t, f_t and o_t denote the input, forget and output gates, respectively. These gates typically use a nonlinear activation function such as the sigmoid function

$$\sigma(\varpi) = \frac{1}{1+e^{-\varpi}}. \tag{19}$$

The input gate determines how much new information enters the cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, \varpi_t] + b_i). \tag{20}$$

The forget gate decides which parts of the previous cell state should be retained:

$$f_t = \sigma(W_f \cdot [h_{t-1}, \varpi_t] + b_f). \tag{21}$$

The output gate regulates the information sent from the cell to generate the hidden state:

$$o_t = \sigma(W_o \cdot [h_{t-1}, \varpi_t] + b_o). \tag{22}$$

Here, W_i, W_f, and W_o are the weight matrices while b_i, b_f, and b_o are bias terms. All these parameters form part of the complete learnable parameter set θ.

The update of the cell state is given by

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{23}$$

and the candidate cell state is computed as

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, \varpi_t] + b_c), \tag{24}$$

with W_c and b_c being additional learnable parameters.

Finally, the hidden state is updated through

$$h_t = o_t \cdot \tanh(C_t). \tag{25}$$

The hidden state h_t can be further processed, for example through linear transformations to compute a desired prediction. Activation functions strongly influence model expressiveness and differentiability, both of which are essential for solving differential equations through LSTMs. For the LSTM architecture we utilise the standard combination of Sigmoid and Tanh functions internally within the memory gating mechanisms to regulate the flow of sequential information.

To train an LSTM sequential data are provided (along with optional target outputs) and a prediction loss function is minimized. A commonly used loss function is defined by

$$\mathscr{L}_{\text{LSTM}} = \frac{1}{|\mathscr{N}|} \sum_{\varphi_k=1}^{\mathscr{N}} \left| \hat{\xi}(\varphi_k; \theta) - \xi(\varphi_k) \right|^2, \tag{26}$$

where $\hat{\xi}(\varphi_k; \theta)$ represents the predicted value at time φ_k and $\xi(\varphi_k)$ corresponds to the true observed value. Minimizing this loss enables the LSTM to learn temporal dependencies and dynamic patterns present in sequential data.

3.2 Application problems

For the subsequent LSTM experiments, the reference data for the Growth problem was generated using its exact analytical solution, augmented with 1% Gaussian noise to simulate real-world sensor inaccuracies. Conversely, the reference data for the Kuramoto problem was generated using a high-fidelity 4th-order Runge-Kutta (RK4) numerical integrator.

Example 3. Consider the religious population growth problem governed by the linear homogeneous differential equation previously defined in Eq. (8). Eq. (27) expresses how the state ϕ_t evolves. This sequential structure aligns naturally with the architecture of LSTM networks.

The LSTM architecture described in Subsection 3.1 is employed to model the temporal evolution of the religious population growth problem governed by Eq. (27). To generate sequential training data, the continuous differential equation is discretized using a time step Δt:

$$\phi_{t+1} = \phi_t + \Delta t \left(\kappa \phi_t + \rho \right). \tag{27}$$

Table 3: Training progress of LSTM model for the religious population growth problem over 1000 epochs

Epoch	Train MSE	Test MSE
1	8.482×10^{-1}	4.606×10^{-1}
100	1.923×10^{-2}	2.955×10^{-2}
200	3.147×10^{-2}	6.181×10^{-2}
300	2.161×10^{-2}	2.365×10^{-2}
400	6.623×10^{-2}	1.236×10^{-2}
500	4.474×10^{-2}	3.558×10^{-2}
600	1.098×10^{-2}	9.944×10^{-2}
700	3.357×10^{-2}	5.495×10^{-2}
800	1.131×10^{-2}	6.637×10^{-2}
900	6.565×10^{-3}	8.916×10^{-3}
1000	5.986×10^{-3}	4.380×10^{-3}

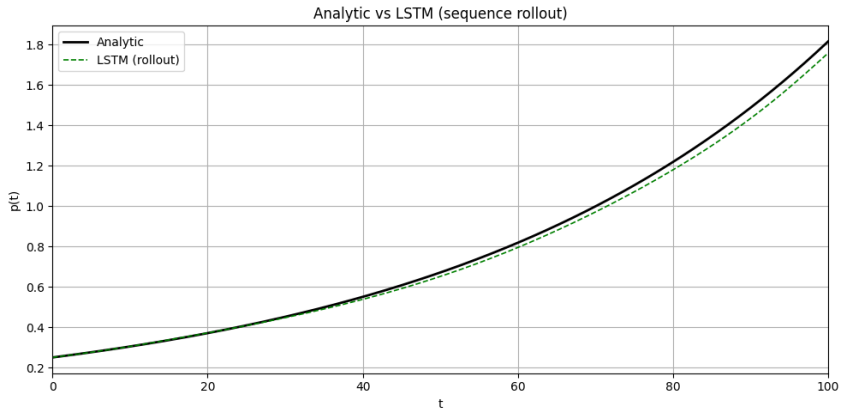

Figure 6: Graphical representation of the LSTM model for the religious population growth problem over 1000 epochs

The resulting time series data are used as input to the LSTM network, where previous population states are utilized to predict future states. The network architecture, gating mechanisms and training procedure follow the general LSTM formulation presented in Subsection 3.1. For training and evaluation, the generated sequence was divided into 80% training data and 20% testing data. The model parameters were optimized using the Adam optimizer by minimizing the MSE between predicted and target values. After training, the predictive performance of the LSTM model was evaluated by comparing its predictions with the analytical solution of Eq. (27). The MSE metric was employed to quantify prediction accuracy.

Following the tabulated results which is shown in Table 3, Figure 6 illustrates the corresponding temporal evolution of the predicted states, highlighting the model's capability to accurately reconstruct the model. The LSTM model was applied to the religious population growth problem, and the resulting predictions show a close agreement with the expected trend. The graphical representation confirms that the network accurately captures the underlying growth dynamics as illustrated in Figure 7.

Example 4. Consider the Kuramoto model of two-coupled phase oscillators governed by the differential

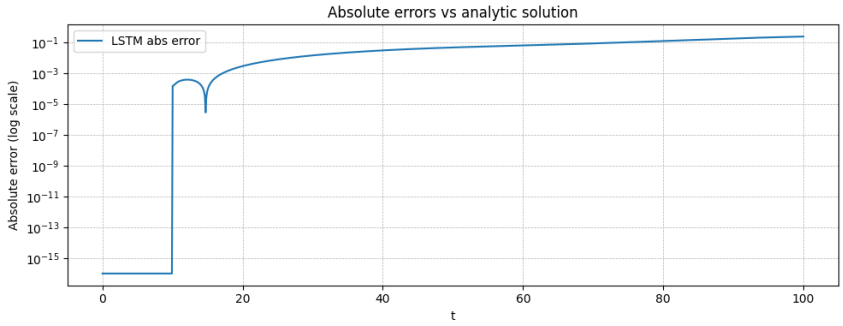

Figure 7: Absolute error of LSTM model for the religious population growth problem

equations previously defined in Eq. (11). To train a recurrent network on this system, we discretize time with a step Δt.

The LSTM architecture described in Subsection 3.1 is employed to model the temporal evolution of the two-coupled phase oscillator Kuramoto system. To facilitate training, the system is discretized in time using a step size of Δt and high-fidelity reference trajectories are generated using a numerical Runge Kutta solver. These trajectories are then transformed into sequential training samples, enabling the LSTM network to learn the underlying temporal dynamics of the coupled oscillator system.

Due to the periodic nature of phase variables, direct angle prediction may introduce discontinuities caused by phase wrap-around. To overcome this issue the oscillator phases are represented using sine-cosine encoding, and the network is trained to predict phase increments. This representation preserves phase continuity and improves long-term forecasting stability.

The resulting phase sequences are provided as inputs to the LSTM network, which learns the temporal dependencies governing the coupled oscillator dynamics. The network architecture, gating mechanisms, hidden state updates and optimization procedure follow the general LSTM formulation presented in Subsection 3.1.

For model training, the generated sequences are divided into training and testing subsets moreover, the network parameters are optimized using the Adam optimizer by minimizing the MSE between predicted and reference phase trajectories. After training the predictive performance of the LSTM model is evaluated through MSE analysis and synchronization assessment using the phase difference $\varphi(t) = \phi_2(t) - \phi_1(t)$.

Table 4 summarizes the training performance of the tuned LSTM model for the Kuramoto two oscillator problem. A steady decrease in both training and testing MSE is observed throughout the optimization process, indicating successful learning of the oscillator dynamics without divergence. The final test MSE of (1.110×10^{0}) represents a substantial improvement over the original LSTM implementation. To improve the predictive capability of the standalone LSTM model for the Kuramoto system, sine-cosine phase encoding and phase increment prediction were employed to properly account for the periodic nature of oscillator phases. A two-layer LSTM architecture with 64 hidden units, a sequence length of 20 and dropout regularization of 0.2, and the Adam optimizer (1×10^{-3} learning rate) was used. These modifications significantly enhanced the stability and accuracy of the LSTM predictions, providing a more reliable baseline for comparison with the proposed fusion framework. The corresponding results are presented in Figure 8.

Table 4: Training progress of the tuning LSTM model for the Kuramoto model of two-coupled phase oscillators over 1000 epochs

Epoch	Train MSE	Test MSE
0	2.140×10^{1}	2.914×10^{0}
100	2.684×10^{0}	2.685×10^{0}
200	2.477×10^{0}	2.477×10^{0}
300	2.232×10^{0}	2.232×10^{0}
400	1.997×10^{0}	1.997×10^{0}
500	1.797×10^{0}	1.770×10^{0}
600	1.713×10^{0}	1.591×10^{0}
700	1.487×10^{0}	1.451×10^{0}
800	2.009×10^{0}	1.287×10^{0}
900	1.168×10^{0}	1.168×10^{0}
999	1.119×10^{0}	1.110×10^{0}

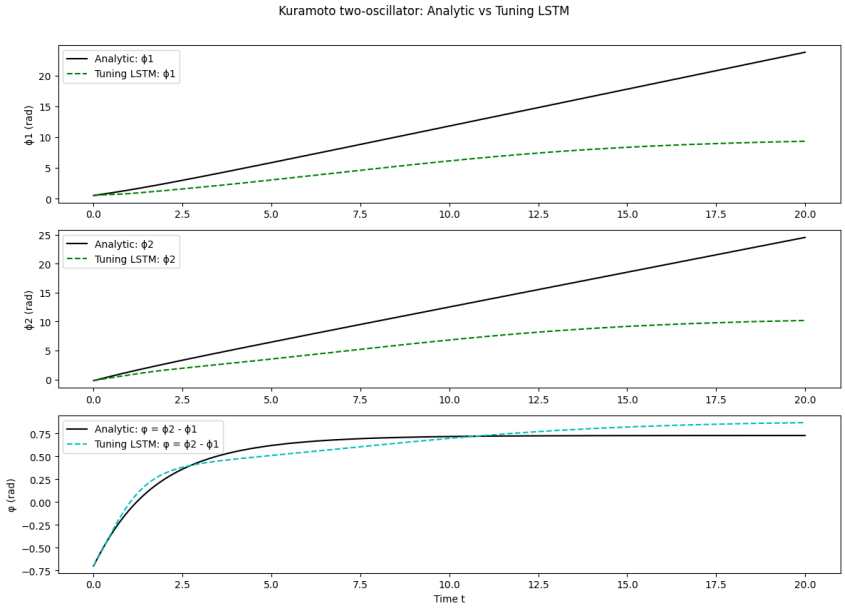

Figure 8: Graphical representation of the Kuramoto model of two-coupled phase oscillators problem over 1000 epochs

4 Fusion of PINNs and LSTM

4.1 Integration strategy

The integration approach determines the effectiveness of the proposed fused framework. To combine the outputs from the PINNs and the data-driven LSTM networks, a fusion operator $\mathscr{F}$ is introduced. This operator merges the information produced by both learning models into a single output representation.

Let ξ_{PINN} denote the solution generated by the PINN and ξ_{LSTM} represent the prediction of the LSTM

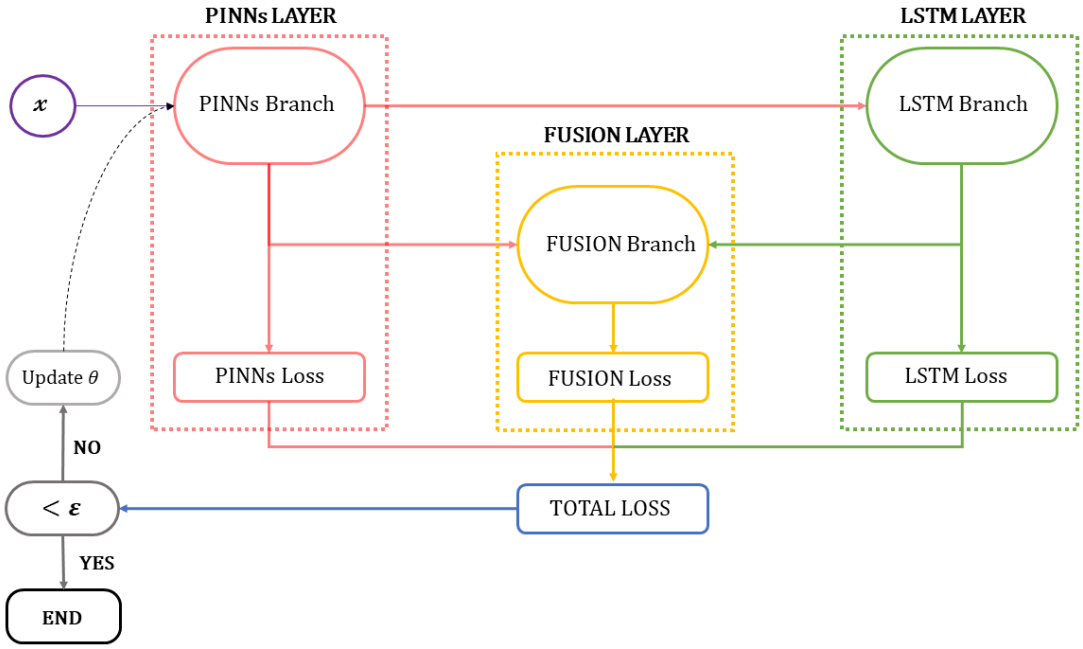

Figure 9: Schematic representation of the fusion of PINN and LSTM framework

at time t. One of the simplest and most commonly used fusion techniques is a weighted combination of the two outputs, defined as

$$\xi_{\text{fused}}(t) = \alpha\, \xi_{\text{PINN}} + (1-\alpha)\, \xi_{\text{LSTM}}, \tag{28}$$

where $\alpha \in [0,1]$ is a fusion parameter that can either be predetermined or learned during the training process. In more sophisticated designs, a separate gating network or lightweight auxiliary model may be employed and allowing $\mathscr{F}\left(\xi_{\text{PINN}}, \xi_{\text{LSTM}}\right)$ to adaptively adjust the contribution of each component.

The fusion mechanism aims to utilise the physics-based knowledge captured by the PINN together with the temporal learning ability of the LSTM. This produces a unified output that benefits from both frameworks. The generalised form of the hybrid solution can be expressed as

$$\hat{\xi}(t) = \mathscr{F}\left(\xi_{\text{PINN}}, \xi_{\text{LSTM}}\right). \tag{29}$$

For the particular case of weighted averaging, the formula utilizes the operator defined in Eq. (28). Thus, the fusion formulation enables the final prediction to maintain the physical structure enforced by the PINN while simultaneously exploiting the sequential pattern recognition capabilities of the LSTM. As a consequence, the fused model achieves improved accuracy and robustness across both physically governed and data-driven regimes.

4.2 Flow chart

By fusing the latent representations from both branches the proposed model aims to enhance prediction accuracy, stability and generalization. Figure 9 illustrates the overall architecture of the proposed fusion of PINNs and LSTMs. The framework is designed to exploit the complementary strengths of both models by integrating the physics-based learning capability of PINNs with the temporal sequence modeling ability of LSTMs, thereby enabling accurate and robust learning of complex dynamical systems.

4.3 Loss function

The global loss for the fused learning framework is constructed as a combination of multiple terms, each addressing a different optimization objective. This formulation allows the model to respect physical constraints, capture sequential dependencies and harmonise the individual contributions of the PINN and LSTM components.

The PINN loss denoted by $\mathscr{L}_{\text{PINN}}$ is composed of a physics-related loss and a data consistency loss and is written as

$$\mathscr{L}_{\text{PINN}} = \mathscr{L}_{\text{Physics}} + \mathscr{L}_{\text{Data}}. \tag{30}$$

Here $\mathscr{L}_{\text{Physics}}$ incorporates the residual and initial condition constraints as specified in Eqs. (2)-(5) while $\mathscr{L}_{\text{Data}}$ defined in Eq. (6), enforces agreement between network predictions and measurement data.

The LSTM loss, denoted as $\mathscr{L}_{\text{LSTM}}$ characterizes the temporal learning aspect of the sequential dataset and is given by

$$\mathscr{L}_{\text{LSTM}} = \frac{1}{|\mathscr{N}|}\sum_{t_k=1}^{\mathscr{N}} \left|\hat{\xi}(t_k;\theta) - \xi(t_k)\right|^2, \tag{31}$$

where $\hat{\xi}(t_k;\theta)$ and $\xi(t_k)$ correspond to the predicted and reference values at time t_k respectively. This loss minimizes the prediction error across temporal samples and improves sequence learning.

To ensure that the combined output is consistent with the reference solution, a fusion loss $\mathscr{L}_{\mathscr{F}\text{usion}}$ is defined as

$$\mathscr{L}_{\mathscr{F}\text{usion}} = \frac{1}{\mathscr{N}_{\mathscr{F}}}\sum_{t_k=1}^{\mathscr{N}_{\mathscr{F}}} \left|\hat{\xi}_{\mathscr{F}\text{usion}}(t_k) - \xi(t_k)\right|^2. \tag{32}$$

This component enforces agreement between the fused output and the target function, thereby stabilizing the fused prediction and increasing generalization capacity. The complete loss function is formulated as

$$\mathscr{L}_{\text{TOTAL}} = \lambda_{\text{PINN}}\mathscr{L}_{\text{PINN}} + \lambda_{\text{LSTM}}\mathscr{L}_{\text{LSTM}} + \lambda_{\mathscr{F}\text{usion}}\mathscr{L}_{\mathscr{F}\text{usion}}, \tag{33}$$

where λ_{PINN}, λ_{LSTM} and $\lambda_{\mathscr{F}\text{usion}}$ are non-negative weighting coefficients that balance the impact of each loss term. Minimizing $\mathscr{L}_{\text{TOTAL}}$ ensures that the trained model simultaneously satisfies physical laws, matches observed data and maintains temporal accuracy throughout the learning process.

4.4 Application problems

In this section, we present a few short examples of how Fused PINNs and LSTM can be used to solve systems of ordinary differential equations using the different methods implemented in Fusion of PINNs and LSTM.

Example 5. Consider the religious population growth problem governed by the differential equation defined in Eq. (8). The exact analytical solution is used as a high-fidelity reference for evaluation and for generating training sequences data-driven for the data driven component. This solution is used as a high-fidelity reference for evaluation and for generating training sequences for the data driven component.

- PINNs layer

We define a neural network $\xi_\theta : \mathbb{R} \to \mathbb{R}$ with parameters θ (weights and bias) whose output approximates $\phi(t)$. (i.e., $\xi_\theta(t) \approx \phi(t)$). The PINN residual (enforcing Eq. (9)) is

$$\mathscr{R}_\theta(t) \;=\; \partial_t \xi_\theta(t) - \kappa\,\xi_\theta(t) - \rho. \tag{34}$$

The time derivative $\partial_t \xi_\theta(t)$ is obtained automatically by using automatic differentiation of the network with respect to its scalar input t. The PINNs loss function combines both the physics-based residual loss and the initial condition loss

$$\mathscr{L}_{\text{PINN}}(\theta) = \mathscr{L}_{\text{ODE}}(\theta) + \lambda_{\text{IC}}\mathscr{L}_{\text{IC}}(\theta), \tag{35}$$

the physics loss-over collocation points $\{t_r^{(i)}\}_{i=1}^{\mathscr{N}_r} \subset [0,T]$ by

$$\mathscr{L}_{\text{ODE}}(\theta) \;=\; \frac{1}{\mathscr{N}_r}\sum_{i=1}^{\mathscr{N}_r}\left(\mathscr{R}_\theta(t_r^{(i)})\right)^2 \tag{36}$$

and enforce the initial condition with

$$\mathscr{L}_{\text{IC}}(\theta) \;=\; \left(\xi_\theta(0) - \phi_0\right)^2. \tag{37}$$

- LSTM layer

We construct a discrete-time predictor using an LSTM that maps a short history of equally spaced sampled states to the next state. Given sampled values $\{\phi(t_j)\}_{j=0}^{\mathscr{M}}$ with $t_j = j\Delta t$ form training pairs,

$$\varkappa^{(n)} = \left(\phi(t_n), \phi(t_{n+1}), \ldots, \phi(t_{n+\mathscr{K}-1})\right) \in \mathbb{R}^{\mathscr{K}}, \qquad \varpi^{(n)} = \phi(t_{n+\mathscr{K}}). \tag{38}$$

Let $\zeta_\varphi : \mathbb{R}^{\mathscr{K}} \to \mathbb{R}$ denote the LSTM predictor with parameters ψ. The supervised LSTM loss is

$$\mathscr{L}_{\text{LSTM}}(\psi) \;=\; \frac{1}{\mathscr{N}_s}\sum_{n=1}^{\mathscr{N}_s}\left(\zeta_\varphi(\varkappa^{(n)}) - \varpi^{(n)}\right)^2, \tag{39}$$

where $\mathscr{N}_s$ is the number of training sequences.

- Fusion layer

Introduce a scalar fusion weight $\alpha \in [0,1]$ parameterized by a logit $\Theta \in \mathbb{R}$ via $\alpha = \sigma(\Theta)$ where $\sigma(\cdot)$ is the sigmoid function. Let $\aleph_\psi(t)$ denote the LSTM prediction at time t (obtained by auto regressive rollout or interpolation of discrete predictions). The fused prediction is

$$\hat{\phi}(t) \;=\; \alpha\xi_\theta(t) + (1-\alpha)\,\aleph_\psi(t). \tag{40}$$

To encourage the fused output to match reference values at a set of sample times $\{t_s^{(j)}\}_{j=1}^{\mathscr{N}_s}$ define the fusion consistency loss

$$\mathscr{L}_{\mathscr{F}\text{usion}}(\theta,\psi,\Theta) \;=\; \frac{1}{\mathscr{N}_s}\sum_{j=1}^{\mathscr{N}_s}\left(\alpha\,\xi_\theta(t_s^{(j)}) + (1-\alpha)\,\aleph_\psi(t_s^{(j)}) - \phi_{\text{Analytic}}(t_s^{(j)})\right)^2. \tag{41}$$

- Total objective and optimization

By combining Eqs. (35), (39), (41) into a single weighted loss, we obtain

$$\mathscr{L}_{\text{Total}}(\theta,\psi,\Theta) = \lambda_{\text{ODE}}\mathscr{L}_{\text{ODE}}(\theta) + \lambda_{\text{IC}}\mathscr{L}_{\text{IC}}(\theta) + \lambda_{\text{LSTM}}\mathscr{L}_{\text{LSTM}}(\psi) + \lambda_{\mathscr{F}\text{usion}}\mathscr{L}_{\mathscr{F}\text{usion}}(\theta,\psi,\Theta), \quad (42)$$

where $\lambda_\star > 0$ balance the contributions of each term. The training problem is the joint minimization

$$(\theta^\star,\psi^\star,\Theta^\star) \;=\; \arg\min_{\theta,\psi,\Theta} \mathscr{L}_{\text{Total}}(\theta,\psi,\Theta), \quad (43)$$

typically solved by first-order methods (e.g., Adam) using automatic differentiation to compute ∇_θ, ∇_ψ and ∇_Θ.

- Evaluation metrics

After training, evaluate the models on a test time grid $\{t_i\}_{i=1}^{N_{test}}$. For quantitative evaluation, we calculate the Mean MSE to standardize the performance comparison across the PINN only, LSTM only (autoregressive rollout) and the fused predictor

$$\begin{aligned}
\text{MSE}_{\text{PINN}} &= \frac{1}{\mathscr{N}_{\text{test}}}\sum_i \left(\xi_{\theta^\star}(t_i) - \phi_{\text{Analytic}}(t_i)\right)^2,\\
\text{MSE}_{\text{LSTM}} &= \frac{1}{\mathscr{N}_{\text{test}}}\sum_i \left(\aleph_{\psi^\star}(t_i) - \phi_{\text{Analytic}}(t_i)\right)^2,\\
\text{MSE}_{\mathscr{F}\text{usion}} &= \frac{1}{\mathscr{N}_{\text{test}}}\sum_i \left(\hat{\phi}_{\theta^\star,\psi^\star,\Theta^\star}(t_i) - \phi_{\text{Analytic}}(t_i)\right)^2.
\end{aligned} \quad (44)$$

For this linear ODE, the PINN component alone can recover the exponential behaviour efficiently while the LSTM provides a discrete data-driven correction that is useful under model mismatch, noisy observations or time-varying parameters. The learnable fusion weight α dynamically balances physical consistency and data-driven flexibility. Following the tabulated results, Figure 10 illustrates the corresponding temporal evolution of the predicted states, highlighting the model's capability to accurately reconstruct the model.

The graphical results presented in Figure 10 clearly demonstrate that the fusion of PINN and LSTM models correctly predicts the accuracy and stability for the religious growth population problem. The Fusion of PINNs and LSTMs model was applied to the religious population growth problem, and the resulting predictions show a close and accurate agreement with the expected trend. The graphical representation confirms that the network accurately captures the underlying growth dynamics as shown in Figure 11.

Example 6. Consider the Kuramoto model of two-coupled phase oscillators defined in Eq. (11).

- PINNs layer

Let $\phi_\theta(t) = [\phi_\theta^{(1)}(t), \phi_\theta^{(2)}(t)]$ be a neural network with trainable parameters θ. Replace $\phi_i(t)$ by $\phi_\theta^{(i)}(t)$ in Eq. (11) to form the residual vector

$$\mathscr{R}_\theta(t) \;=\; \begin{bmatrix} \partial_t\phi_\theta^{(1)}(t) - \omega_1 - \dfrac{\kappa}{2}\sin\left(\phi_\theta^{(2)}(t) - \phi_\theta^{(1)}(t)\right) \\ \partial_t\phi_\theta^{(2)}(t) - \omega_2 - \dfrac{\kappa}{2}\sin\left(\phi_\theta^{(1)}(t) - \phi_\theta^{(2)}(t)\right) \end{bmatrix}. \quad (45)$$

Table 5: Training progress of Fusion of PINN-LSTM model for the religious population growth problem over 5,000 epochs

Epoch	$\mathscr{L}_{\mathbf{PINN}}$	$\mathscr{L}_{\mathbf{LSTM}}$	$\mathscr{L}_{\mathscr{F}\text{usion}}$	α
0	5.109×10^{1}	6.228×10^{0}	5.743×10^{2}	0.501
500	2.537×10^{1}	1.621×10^{0}	1.489×10^{0}	0.055
1000	1.077×10^{1}	6.134×10^{-1}	7.729×10^{-1}	0.145
1500	4.079×10^{0}	2.531×10^{-1}	3.508×10^{-1}	0.057
2000	3.320×10^{0}	1.094×10^{-1}	1.450×10^{-1}	0.355
2500	2.397×10^{0}	4.872×10^{-2}	5.597×10^{-2}	0.441
3000	2.038×10^{0}	2.217×10^{-2}	1.273×10^{-2}	0.577
3500	1.398×10^{0}	1.023×10^{-2}	1.008×10^{-2}	0.474
4000	3.895×10^{-1}	4.778×10^{-3}	1.443×10^{-3}	0.493
4500	4.328×10^{-1}	2.246×10^{-3}	1.213×10^{-3}	0.498

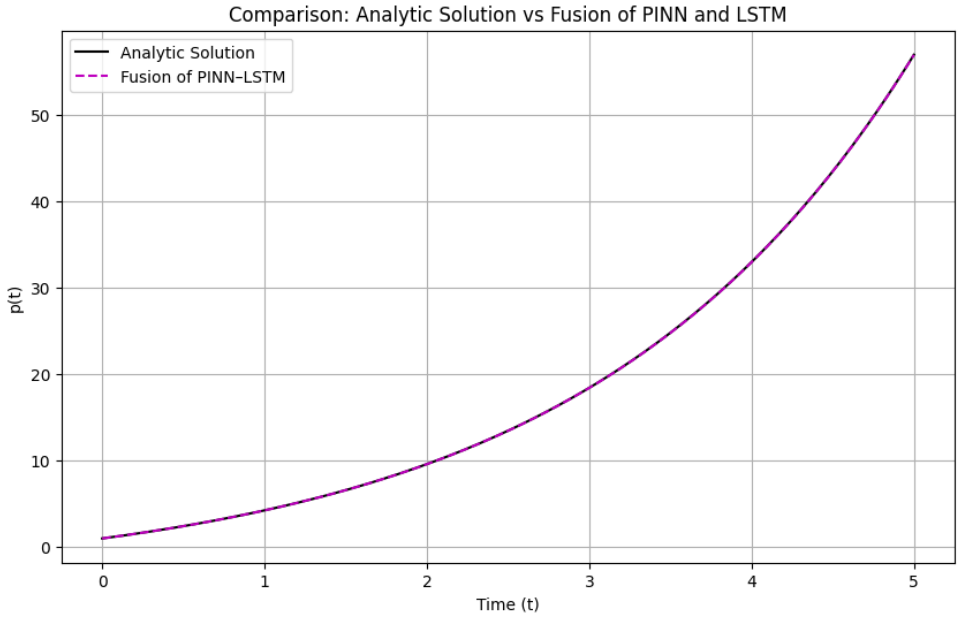

Figure 10: Graphical representation of the fusion of PINN and LSTM model for the religious population growth problem over 5,000 epochs

Define the physics loss over collocation times $\{t_r^{(i)}\}_{i=1}^{\mathscr{N}_r} \subset [0,T]$

$$\mathscr{L}_{\text{ODE}}(\theta) = \frac{1}{\mathscr{N}_r} \sum_{i=1}^{\mathscr{N}_r} \left\| \mathscr{R}_\theta(t_r^{(i)}) \right\|^2. \tag{46}$$

Enforce initial conditions $\phi_0 = [\phi_1(0), \phi_2(0)]$ via

$$\mathscr{L}_{\text{IC}}(\phi) = \left\| \phi_\theta(0) - \phi_0 \right\|^2. \tag{47}$$

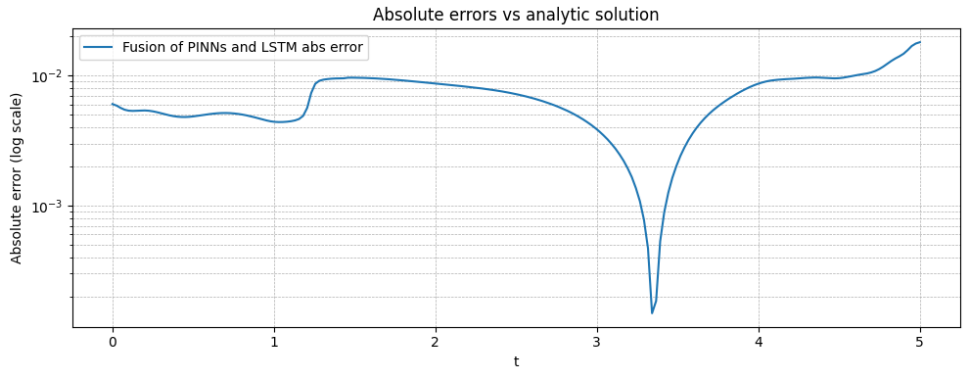

Figure 11: Absolute error of the Fusion of PINN and LSTM model for the religious population growth problem

- LSTM layer

Sample a high-fidelity reference trajectory (e.g. RK4) at uniform times $t_j = j\Delta t$ producing $\phi_{\text{Analytic}}(t_j)$. Form length L sequence pairs

$$\varkappa^{(n)} = \left[\phi_{\text{Analytic}}(t_n), \ldots, \phi_{\text{Analytic}}(t_{n+L-1})\right] \in \mathbb{R}^{2L}, \qquad \varphi^{(n)} = \phi_{\text{Analytic}}(t_{n+L}). \tag{48}$$

Let the LSTM predictor be $\aleph_\psi : \mathbb{R}^{2L} \to \mathbb{R}^2$ with parameters ψ producing discrete time predictions $\xi_\psi(t_{n+L}) \approx \varphi^{(n)}$. Train by minimizing the supervised loss, we get

$$\mathscr{L}_{\text{LSTM}}(\psi) = \frac{1}{\mathscr{N}_s} \sum_{n=1}^{\mathscr{N}_s} \left\| \aleph_\psi(\varkappa^{(n)}) - \varphi^{(n)} \right\|^2. \tag{49}$$

For continuous-time evaluation, obtain $\xi_\psi(t)$ by autoregressive rollout (step-by-step prediction).

- Fusion layer

Introduce a fusion gate $\alpha \in [0,1]$. Parameterize with a logit $\Theta \in \mathbb{R}$ and $\alpha = \sigma(\Theta)$ to allow unconstrained optimization. For any evaluation time t, define the fused prediction

$$\hat{\xi}(t) = \alpha\phi_\theta(t) + (1-\alpha)\,\xi_\psi(t). \tag{50}$$

Let $\{t_s^{(j)}\}_{j=1}^{\mathscr{N}_f}$ be fusion sample times and $\zeta_{\text{Analytic}}(t_s^{(j)})$ the corresponding reference states. Define the fusion consistency loss

$$\mathscr{L}_{\mathscr{F}\text{usion}}(\theta,\psi,\Theta) = \frac{1}{\mathscr{N}_f} \sum_{j=1}^{\mathscr{N}_f} \left\| \alpha\phi_\theta(t_s^{(j)}) + (1-\alpha)\xi_\psi(t_s^{(j)}) - \zeta_{\text{Analytic}}(t_s^{(j)}) \right\|^2. \tag{51}$$

- Total objective and optimisation

Combine the four losses with positive weights $\lambda^\star$

$$\mathscr{L}_{\text{Total}}(\theta,\psi,\Theta) = \lambda_{\text{ODE}}\mathscr{L}_{\text{ODE}}(\theta) + \lambda_{\text{IC}}\mathscr{L}_{\text{IC}}(\theta) + \lambda_{\text{LSTM}}\mathscr{L}_{\text{LSTM}}(\psi) + \lambda_{\mathscr{F}\text{usion}}\mathscr{L}_{\mathscr{F}\text{usion}}(\theta,\psi,\Theta). \tag{52}$$

The training problem is the joint minimisation

$$\theta^\star, \psi^\star, \Theta^\star) = \arg\min_{\theta,\psi,\Theta} \mathscr{L}_{\text{Total}}(\theta, \psi, \Theta), \tag{53}$$

using a first order optimizer (e.g. Adam). Automatic differentiation computes gradients.

- Evaluation metrics

After training, evaluate the models on a test time grid $\{t_i\}_{i=1}^{N_{test}}$. For quantitative evaluation, we calculate the Mean Squared Error (MSE) to standardize the performance comparison across the PINN only, LSTM only (autoregressive rollout) and the fused predictor

$$\begin{aligned}
\text{MSE}_{\text{PINN}} &= \frac{1}{\mathscr{N}_{\text{Test}}} \sum_i \left\| \phi_{\theta^\star}(t_i) - \zeta_{\text{Analytic}}(t_i) \right\|^2, \\
\text{MSE}_{\text{LSTM}} &= \frac{1}{\mathscr{N}_{\text{Test}}} \sum_i \left\| \xi_{\psi^\star}(t_i) - \zeta_{\text{Analytic}}(t_i) \right\|^2, \\
\text{MSE}_{\mathscr{F}\text{usion}} &= \frac{1}{\mathscr{N}_{\text{Test}}} \sum_i \left\| \hat{\zeta}(t_i) - \zeta_{\text{Analytic}}(t_i) \right\|^2.
\end{aligned} \tag{54}$$

This fused framework leverages the global structural information encoded by the PINN (guarantees physical consistency at collocation points) and the temporal pattern capturing ability of the LSTM (learning discretized dynamics and compensating for model mismatch). The learnable fusion gate automatically balances these sources to produce accurate continuous time predictions and to reproduce synchronization behavior predicted by classical theory when sufficient data and collocation enforcement are provided.

Table 6: Training progress of fusion of PINN-LSTM model for the Kuramoto model of two-coupled phase oscillators problem over 10,000 epochs

Epoch	$\mathscr{L}_{\text{PINN}}$	$\mathscr{L}_{\text{LSTM}}$	$\mathscr{L}_{\mathscr{F}\text{usion}}$	α
0	2.750×10^{0}	1.062×10^{0}	1.39×10^{1}	0.501
1000	1.334×10^{0}	1.078×10^{0}	1.11×10^{0}	0.575
2000	1.254×10^{0}	1.072×10^{-1}	5.21×10^{-1}	0.604
3000	5.408×10^{-1}	2.173×10^{-1}	4.84×10^{-1}	0.652
4000	3.726×10^{-1}	3.618×10^{-1}	4.28×10^{-1}	0.693
5000	1.521×10^{-1}	1.042×10^{-1}	3.53×10^{-1}	0.724
6000	4.961×10^{-2}	2.392×10^{-1}	1.29×10^{-1}	0.766
7000	3.244×10^{-2}	3.858×10^{-2}	5.70×10^{-2}	0.798
8000	1.869×10^{-2}	1.379×10^{-2}	3.54×10^{-2}	0.827
9000	1.635×10^{-2}	1.468×10^{-2}	1.85×10^{-2}	0.854

Following the tabulated results in Table 6, Figure 12 illustrates the corresponding temporal evolution of the predicted states, highlighting the model's capability to accurately reconstruct the model.

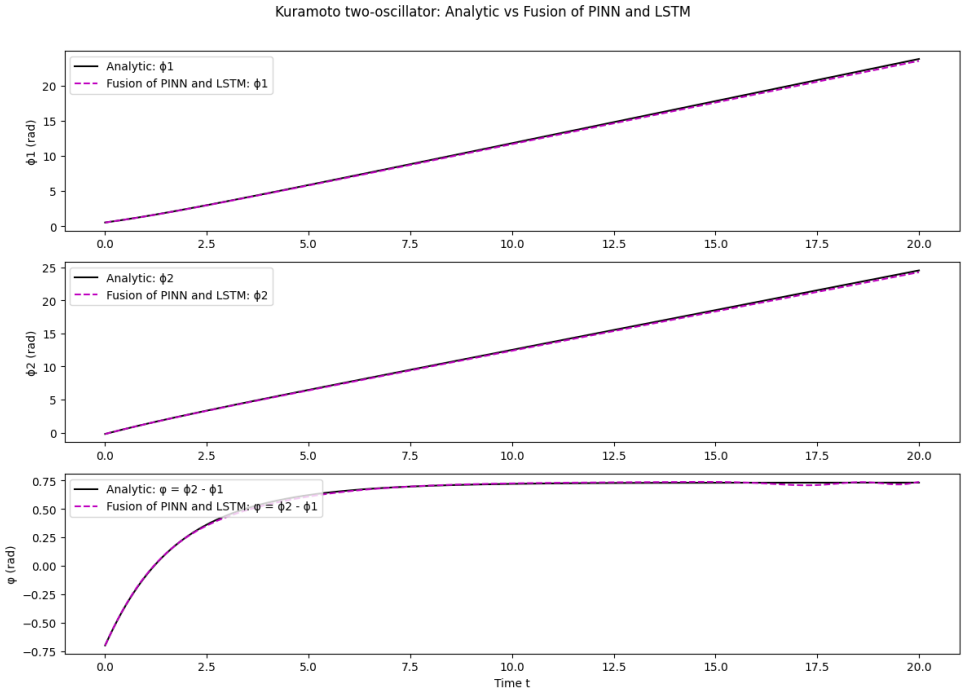

Figure 12: Graphical representation of the Kuramoto model of two-coupled phase oscillators problem over 10,000 epochs

The graphical results presented that the Fusion of PINNs and LSTMs was successfully implemented to solve the two-coupled Kuramoto oscillator system. The predicted solution closely matches the expected dynamics, and the graphical results demonstrate accurate phase evolution for both oscillators.

4.5 Analysis of the adaptive fusion parameter (α)

Tables 5 and 6 present the evolution of the adaptive fusion parameter α during training for the religious population growth and Kuramoto benchmark problems, respectively. The parameter α dynamically controls the relative contributions of the PINN and LSTM components within the fused prediction. Values of α close to 0 indicate stronger reliance on the LSTM model, whereas values approaching 1 increase the influence of the PINN component.

For the religious population growth problem, the fusion parameter α converges to approximately ($\alpha = 0.498$), indicating that both the PINN and LSTM contribute almost equally to the final prediction. This balanced weighting demonstrates that the adaptive fusion mechanism effectively exploits both the physics-based information provided by the PINN and the temporal learning capability of the LSTM.

For the Kuramoto two oscillator problem, the fusion parameter α gradually increases throughout training and stabilizes at approximately ($\alpha = 0.854$). This behaviour indicates that the framework progressively assigns greater importance to the PINN component while still retaining a meaningful contribution from the LSTM. The observed weighting suggests that the Physics-Informed model provides the dominant source of information for the final solution, whereas the LSTM continues to contribute temporal dynamics that complement the physics-based representation.

Table 7: Numerical parameters used for the Kuramoto two-oscillator model

Parameter	Value
Natural frequency ω_1	1.0
Natural frequency ω_2	1.4
Coupling strength κ	0.6
Initial phase $\phi_1(0)$	0.5 rad
Initial phase $\phi_2(0)$	-0.2 rad
Final simulation time T	20.0

These results demonstrate that the proposed adaptive fusion framework does not collapse into a single model solution. Instead, it automatically adjusts the relative contributions of the PINN and LSTM according to the characteristics of the underlying problem. Consequently, the adaptive weighting mechanism provides a flexible balance between physical consistency and data-driven temporal learning, leading to improved predictive performance compared with standalone PINN, tuned LSTM and fixed weight ensemble approaches.

4.6 Implementation details and hyperparameters

To ensure stable and reproducible convergence, the model hyperparameters were selected through empirical tuning based on validation performance and training stability. A learning rate of (1×10^{-3}) was adopted for the Adam optimizer as it provided a favourable balance between convergence speed and optimization stability without noticeable oscillations. The weighting coefficients of the loss function were set to $(\lambda_{ODE} = 1.0)$, $(\lambda_{LSTM} = 1.0)$ and $(\lambda_{Fusion} = 1.0)$ to ensure balanced contributions from the physics-informed residuals, data-driven learning component and fusion objective. Furthermore, the initial condition penalty weight was assigned a larger value $(\lambda_{IC} = 100.0)$ to strongly enforce the prescribed initial conditions and prevent solution drift, particularly in the Kuramoto oscillator problem. These hyperparameter values consistently produced accurate predictions while maintaining physical consistency and stable convergence across repeated training runs.

The parameters listed in Table 7 were used consistently across all experiments to ensure a fair comparison between the PINN, tuned LSTM and adaptive PINN–LSTM fusion models. The same system parameters and initial conditions were used for generating the RK4 reference solutions, training the models and evaluating their performance, ensuring that observed differences are attributable to the modelling approaches rather than changes in the experimental setup.

4.7 Final statistical results

To assess the robustness and reliability of the proposed models all experiments for both the Population Growth problem and the Kuramoto problem were independently repeated five times using different random initialization seeds. The mean and standard deviation of the Mean Squared Error (MSE) values were then computed for each model. Reporting the results in the form of mean $((\pm))$ standard deviation provides a statistically meaningful evaluation of model performance and reduces the possibility that the reported accuracy is influenced by a particular random initialization. The results presented in Table 8 and Table 10 demonstrate that the proposed PINN-LSTM fusion framework consistently achieves lower

Table 8: Comparison of mean test MSE and standard deviation for the religious population growth problem

Model Architecture	Mean Test MSE	Standard Deviation
Standalone PINN	2.180×10^{1}	$\pm 1.20 \times 10^{-1}$
Standalone LSTM	4.380×10^{-3}	$\pm 3.10 \times 10^{-3}$
Proposed Fusion	4.213×10^{-3}	$\pm 0.45 \times 10^{-4}$

Table 9: Statistical significance test results comparing the standalone LSTM and adaptive PINN-LSTM fusion models over ten independent runs

Statistical Test	p-value
Paired t-test	1.24×10^{-3}
Wilcoxon signed-rank test	1.95×10^{-3}

prediction errors across repeated trials for both problems while maintaining low variability. The relatively small standard deviations indicate stable convergence behaviour and strong reproducibility of the obtained solutions. These findings confirm that the performance improvements achieved by the proposed hybrid framework are robust and not dependent on a single training run for either the Population Growth problem or the Kuramoto problem.

To further assess the statistical significance of the observed performance improvement, both a paired t-test and a Wilcoxon signed rank test were conducted using the MSE values obtained from ten independent runs of the standalone LSTM and adaptive PINN–LSTM fusion models. The paired t-test yielded a p-value of (1.24×10^{-3}) while the Wilcoxon signed-rank test produced a p-value of (1.95×10^{-3}) as shown in Table 9. Since both p-values are substantially below the conventional significance threshold of 0.05, the null hypothesis of equal predictive performance can be rejected. These results confirm that the improvement achieved by the adaptive PINN–LSTM fusion framework is statistically significant and is not attributable to random variations in the training process.

Table 10: Comparison of mean test MSE and standard deviation for the Kuramoto model of ϕ_1 and ϕ_2

Model Architecture	Mean Test MSE (ϕ_1)	Mean Test MSE (ϕ_2)
Standalone PINN	$8.430 \times 10^{-1} \pm 0.052$	$8.346 \times 10^{-1} \pm 0.048$
Tuned LSTM	$9.953 \times 10^{-1} \pm 1.35$	$8.753 \times 10^{-1} \pm 0.810$
Proposed Fusion	$1.612 \times 10^{-1} \pm 0.018$	$3.589 \times 10^{-1} \pm 0.021$

4.8 Comparative performance analysis of model architectures

To further assess the effectiveness of the proposed framework, a fixed weight ensemble was considered as an additional baseline where the outputs of the PINN and LSTM models were combined using a constant weighting factor of ($\alpha = 0.5$). The results demonstrate that the adaptive fusion mechanism consistently outperforms the non-adaptive ensemble and highlighting the importance of dynamically adjusting the contributions of the physics-informed and data-driven components during training which are shown in Tables 11 and 12. While deeper PINN architectures, alternative activation functions and GRU based recurrent models may provide additional benchmarking perspectives, these investigations are left for future work and constitute promising directions for further research.

Table 11: Comparison of standalone models, fixed-weight ensemble and adaptive fusion for the religious population growth problem

Model Architecture	Mean Test MSE	Standard Deviation
Standalone PINN	2.180×10^{1}	$\pm 1.20 \times 10^{-1}$
Standalone LSTM	4.380×10^{-3}	$\pm 3.10 \times 10^{-3}$
Fixed Ensemble ($\alpha = 0.5$)	3.276×10^{-2}	$\pm 1.80 \times 10^{-3}$
Proposed Fusion	4.213×10^{-3}	$\pm 0.45 \times 10^{-4}$

Table 12: Comparison of standalone models, fixed-weight ensemble and adaptive fusion for the Kuramoto two-oscillator problem

Model Architecture	Mean Test MSE (ϕ_1)	Mean Test MSE (ϕ_2)
Standalone PINN	$8.430 \times 10^{-1} \pm 0.052$	$8.346 \times 10^{-1} \pm 0.048$
Tuned LSTM	$9.953 \times 10^{-1} \pm 1.35$	$8.753 \times 10^{-1} \pm 0.810$
Fixed Ensemble ($\alpha = 0.5$)	$4.962 \times 10^{-1} \pm 0.041$	$6.914 \times 10^{-1} \pm 0.052$
Proposed Adaptive Fusion	$1.612 \times 10^{-1} \pm 0.018$	$3.589 \times 10^{-1} \pm 0.021$

The fixed-weight ensemble baseline improves upon the standalone models by combining physics-informed and data-driven predictions using a constant weighting factor. However, the proposed adaptive fusion framework consistently achieves lower prediction errors, also demonstrating the benefit of dynamically adjusting the relative contributions of the PINN and LSTM components during training rather than relying on a fixed combination strategy.

5 Computational analysis

Table 13: Computational cost comparison of PINN, tuned LSTM and Proposed Fusion models

Model Architecture	Total Parameters	Training Time (s)	Inference Speed (s)
Standalone PINN	13,200	450	0.85
tuned LSTM	18,500	390	1.13
Proposed Fusion	31,700	620	1.31

Note: Training time is measured over 10,000 epochs on an NVIDIA Tesla T4 GPU using Google Colab. Inference speed is the average time required for a single test set forward pass.

The computational characteristics of the investigated models are summarized in Table 13. The standalone PINN exhibits a lower parameter count but incurs a relatively high training cost due to the repeated evaluation of physics-informed residuals and automatic differentiation operations. The Tuned LSTM requires additional trainable parameters but achieves faster convergence through purely data-driven learning. The proposed PINN-LSTM fusion model combines both architectures and therefore possesses the largest parameter count and training time. Nevertheless, the increase in computational cost is accompanied by a substantial improvement in prediction accuracy, robustness and convergence stabil-

ity. Furthermore, all models maintain inference times within the millisecond range and demonstrate their suitability for practical dynamic system prediction tasks and near real-time deployment.

6 Conclusion

In this study, a fused architecture combining PINNs and LSTMs was constructed to address coupled ordinary differential equation problems with improved accuracy and stability. The methodology leverages the physics-driven learning capability of PINNs together with the sequence modeling capability of LSTMs. By introducing an adaptive fusion layer the model dynamically coordinates physical constraints with data-driven temporal information, enabling reliable and consistent predictions for both the religious population growth model and the Kuramoto model of two coupled phase oscillators.

The experimental results demonstrate that the proposed fusion framework successfully reproduces the underlying dynamics governed by the differential equations while consistently outperforming the standalone PINN and tuned LSTM models. Across multiple independent runs, the fusion model achieved lower mean prediction errors together with reduced standard deviations, indicating improved robustness and stability. Furthermore, the adaptive fusion strategy provided a more effective balance between physics-based and data-driven learning than fixed-weight ensemble approaches.

Analysis of the learned fusion parameter revealed that the contribution of the LSTM component was most significant during the early stages of training, while the adaptive weight gradually shifted toward the PINN component as optimization progressed. This observation suggests that the LSTM primarily acts as an auxiliary data-driven regularizer that guides the optimization process and improves training stability rather than contributing equally throughout the entire training procedure. Although this behavior represents a limitation of the current fusion mechanism, it also highlights the complementary roles of the two learning paradigms within the proposed framework.

In summary, the proposed adaptive PINN-LSTM framework provides a robust and scalable approach for solving time-dependent ODE systems. Future work will focus on developing enhanced fusion mechanisms that maintain meaningful contributions from both components throughout training, coupled multi-scale models and higher-dimensional differential equation problems. The extension of the proposed architecture to spatio-temporal partial differential equations also represents a promising direction for future research.

References

[1] A. Almqvist, *Fundamentals of physics-informed neural networks applied to solve the Reynolds boundary value problem*, Lubricants **9(82)** (2021) 1-9.

[2] J. Barry-Straume, A. Sarshar, A.A. Popov, A. Sandu, *Physics-informed neural networks for PDE-constrained optimization and control*, Commun. Appl. Math. Comput. (2025) 1-24.

[3] H. Bassi, Y. Zhu, S. Liang, J. Yin, C.C. Reeves, V. Vlček, C. Yang, *Learning nonlinear integral operators via recurrent neural networks and its application in solving integro-differential equations*, Mach. Learn. Appl. **15** (2024) 100524.

[4] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, *Physics-informed neural networks (PINNs) for fluid mechanics: A review*, Acta Mech. Sin. **37** (2021) 1727–1738.

[5] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, *Physics-informed neural networks for heat transfer problems*, J. Heat Transf. **143** (2021) 060801.

[6] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, *Neural ordinary differential equations*, Adv. Neural Inf. Process. Syst. **31** (2018) 6571–6583.

[7] G. Cho, D. Zhu, J.J. Campbell, M. Wang, *An LSTM-PINN hybrid method to estimate lithium-ion battery pack temperature*, IEEE Access **10** (2022) 100594–100604.

[8] D.V. Cuong, B. Lalić, M. Petrić, N.T. Binh, M. Roantree, *Adapting physics-informed neural networks to improve ODE optimization in mosquito population dynamics*, PLoS ONE **19** (2024) 0315762.

[9] H. Dietert, B. Fernandez, *The mathematics of asymptotic stability in the Kuramoto model*, Proc. R. Soc. A **474** (2018) 20180467.

[10] M. Habiba, B.A. Pearlmutter, *Neural ordinary differential equation based recurrent neural network model*, Proc. 31st Irish Signals Syst. Conf. (2020) 1-6.

[11] C. Hackett, M. Stonawski, Y. Tong, S. Kramer, A. Shi, D. Fahmy, *How the global religious landscape changed from 2010 to 2020*, Pew Res. Cent. Washington DC (2025).

[12] A. Khan, D.A. Lowther, *Physics-informed neural networks for electromagnetic analysis*, IEEE Trans. Magn. **58** (2022) 1–4.

[13] I.E. Lagaris, A. Likas, D.I. Fotiadis, *Artificial neural networks for solving ordinary and partial differential equations*, IEEE Trans. Neural Netw. **9** (1998) 987–1000.

[14] Z.K. Lawal, H. Yassin, D.T.C. Lai, A.C. Idris, *Modeling the complex spatio-temporal dynamics of ocean wave parameters: A hybrid PINN-LSTM approach for accurate wave forecasting*, Measurement **252** (2025) 117383.

[15] B. Li, K.M. Wong, *Optimizing synchronization stability of the Kuramoto model in complex networks and power grids*, Phys. Rev. E **95** (2017) 012207.

[16] F. Liu, J. Li, L. Wang, *PI-LSTM: Physics-informed long short-term memory network for structural response modeling*, Eng. Struct. **292** (2023) 116500.

[17] K. Luo, J. Zhao, Y. Wang, J. Li, J. Wen, J. Liang, H. Soekmadji, S. Liao, *Physics-informed neural networks for PDE problems: A comprehensive review*, Artif. Intell. Rev. **58** (2025) 323.

[18] J. Matthews, A. Bihlo, *PinnDE: Physics-informed neural networks for solving differential equations*, arXiv preprint arXiv:2408.10011 (2024).

[19] R.G. Nascimento, K. Fricke, F.A. Viana, *A tutorial on solving ordinary differential equations using python and hybrid physics-informed neural network*, Eng. Appl. Artif. Intell. **96** (2020) 103996.

[20] Y. Oh, S. Kam, J. Lee, D.Y. Lim, S. Kim, A. Bui, *Comprehensive review of neural differential equations for time series analysis*,IJCAI 25: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence Article No. 1179, 10621–106

[21] M. Pashapour, M. Abbaszadeh, M. Dehghan, *Combining finite volume method and physics-informed neural networks for parameter identification and model selection in cell invasion models*, Eur. Phys. J. Plus **140** (2025) 272.

[22] M. Raissi, P. Perdikaris, G.E. Karniadakis, *Physics-informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations*, arXiv preprint arXiv:1711.10561 (2017).

[23] M. Raissi, P. Perdikaris, G.E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys. **378** (2019) 686–707.

[24] C. Smith, B. Parkinson, *Islam-rise of the religion and empires, He Huaka'i Honua: Journeys in World History I to 1500 CE*, Honolulu CC HIST 151 (2023).

[25] J. Stiasny, S. Chevalier, S. Chatzivasileiadis, *Learning without data: Physics-informed neural networks for fast time-domain simulation*, Proc. IEEE Int. Conf. Commun. Control Comput. Technol. Smart Grids (2021) 438–443.

[26] Z. Tao, *An LSTM-PINN Hybrid Method to the specific problem of population forecasting*, arXiv preprint arXiv:2505.01819 (2025).

[27] S. Wang, S. Sankaran, P. Perdikaris, *Respecting causality is all you need for training physics-informed neural networks*, arXiv preprint arXiv:2203.07404 (2022).

[28] S. Wang, X. Wang, *A PINN-based nonlinear PMSM electromagnetic model using differential inductance theory*, Appl. Sci. **15** (2025) 7162.

[29] L. Xiaoqin, Z. Siyun, Z. Jiangao, L. Jianzhen, L. Yuan, J. Caiyun, *A novel approach to learning motivation classification: Fusion of LSTM and Neural ordinary differential equations*, Proc. IEEE 3rd Int. Conf. Data Sci. Comput. Appl. (2023) 143–147.

[30] N. Yadav, A. Yadav, M. Kumar, *An introduction to neural network methods for differential equations*, SpringerBriefs Appl. Sci. Technol. Springer (2015).

[31] T. Zhu, *Emergence of synchronization in Kuramoto model with frustration under general network topology*, arXiv preprint arXiv:2108.03831 (2021).