Optimization of Neural Network Architectures for Dynamic System Prediction Using LSTM-Based NODE and Water Wave Optimization

Jiating Fang
School of Mathematics and Statistics, Baicheng Normal University, Baicheng, Jilin, 137000, China
E-mail: fang2358@126.com

Keywords: Dynamic system, Variational Differential Equation (VDE), time-series data, network architecture, Long Short Term Neural Ordinary Differential Water Wave Optimization (LST-NODE-WWO).

Scientific computing faces challenges in accurately predicting complex dynamic systems due to initial conditions and non-stationary behavior, with rational neural networks often requiring human tuning. To overcome these limitations, this research proposes a Long Short-Term Neural Ordinary Differential Water Wave Optimization (LST-NODE-WWO) for dynamic system prediction. The goal is to automatically discover the optimal network architecture that adapts to the dynamics of each system. The proposed methodology includes five steps: collection of time-series data (e.g., position and velocity), preprocessing with z-score normalization for temporal consistency, and multi-resolution feature extraction using the discrete wavelet transform (DWT). The hybrid model integrates LSTM for capturing temporal dependencies, NODE as a variational differential equation (VDE) for continuous-time modeling, and WWO as a metaheuristic strategy for structure and hyperparameter optimization. This fusion enables the system to learn both physical consistency and long-range temporal patterns. Experimental results on a coupled damped harmonic oscillator dataset show that the LST-NODE-WWO outperforms baseline models. Experimental evaluations using a coupled damped harmonic oscillator dataset exhibit that the suggested model outperforms baseline approaches. The LST-NODE-WWO achieves a test MAE of 0.0914 and RMSE of 0.1298 for Oscillator 1, and a test MAE of 0.0213 and RMSE of 0.0301 for Oscillator 2, significantly improving prediction accuracy. Hence, LST-NODE-WWO offers a robust solution for modeling nonlinear, time-varying dynamic systems with minimal manual intervention.

Povzetek: Raziskava predstavi hibridni model LST-NODE-WWO, ki združuje LSTM za učenje časovnih odvisnosti, Neural ODE za neprekinjeno modeliranje dinamike in Water Wave Optimization za samodejno izbiro arhitekture. Na sklopu dušene sklopljene nihalne dvojice doseže najnižje MAE/RMSE in znatno preseže QLSTM ter QGRU pri napovedovanju nelinearnih dinamičnih sistemov.

1 Introduction

Dynamical systems form the basis for understanding and modeling the physical universe since they describe how systems evolve with respect to internal and external factors [1]. Modeling and predicting these systems, i.e., determining their state at a future time or times, is significant in all aspects of life: traffic control, finance (financial markets), ecology, and biological systems [2]. These systems often exhibit nonlinear complex behaviors that make accurate prediction difficult, particularly when one doesn't know the governing equation(s) or only partially understands them. Time series modeling, or forecasting future behavior from timely observed data, has emerged as a technique to model such systems [3]. In physics and engineering, time series data occur from systems driven by DEs. However, due to the complexity of the system, hidden interactions, or measurement capabilities, it is frequently impossible to know the exact governing equations [4].

In consequence, data-driven methods have shifted into the limelight, allowing researchers to model system behavior directly from observations without using the true system equations [5]. Various data-driven approaches have been proposed; specifically, NN and DL-based models have demonstrated an impressive capability to capture temporal patterns and linear/nonlinear relationships in time series data. RNNs, LSTM networks, and derivatives are often used for time-series forecasting, as they appropriately preserve historical dependencies [6, 7]. Requirements, however, necessitate careful network design adjustment, which can significantly affect generalization and prediction performance [8].

To approach the structural selection and model-building problem, some research has recently introduced some novel mathematical approaches by including ODEs and optimization in the dynamic neural network-based approach [9]. One developing area involves incorporating the use of ODE for neural networks, which allows them to learn dynamics for continuous time and optimize their internal structure with good theoretical reasoning [10].

This allows for a more realistic portrayal of a system's behavior, and even improves stability with long-term predictions. In addition to common real-world model issues like incomplete and noisy observations across dynamic systems [11], wireless communications present unique challenges that further complicate inference due to issues of signal interference and/or congestion that can lead to observation inconsistencies or entirely missing information. In conjunction with greater dimensionality and more interacting variables, the ability to forecast pure dynamics reliably is almost impossible using standard forms of modeling [12]. NN models in conjunction with very adaptively trained variational optimization provide an opportunity to deal with the missing piece, and to also account for more complicated dependencies and entropies [13].

The purpose of this research is to create a highly adaptable and automated neural network model to predict complex dynamic systems accurately. By using a combination of the LSTM network, NODE, and structurally optimizing the model with WWO, this research aims to provide an alternative to conventional neural networks that typically require tuning by a human operator and rarely work for non-stationary data or noisy data. The LST-NODE-WWO model learns temporal patterns, captures the continuous-time behavior in the system, and can adjust its structure dynamically, resulting in a robust, generalizable methodology that satisfies a push-button solution for real-world applications for predicting dynamic systems.

The organization of the research is as follows: Part 2 gives the related works, Part 3 explains the methodology, Part 4 gives the results and discussion, and Part 5 concludes the research.

2 Related works

Recent advancements in neural networks have significantly improved the modeling and prediction of nonlinear dynamical systems. The research [14] compared RNN variants for predicting pore-water pressure in slope stability analysis, with LSTM and GRU showing higher robustness. To address uncertainty in runoff forecasting, a hybrid SVM-based model was introduced [15]. The research [16] connected dynamical system theory with neural training via KOT. The analysis [17] investigated adaptive behavior and predictive information in agent-environment systems. In physical forecasting, a DL-physics hybrid called PhICNet was proposed [18]. A transformer-based approach using continuous attention for spatiotemporal prediction was developed in [19], while SDPA-Net with novel dynamic attention mechanisms for sales forecasting was presented in [20]. LSTM-based data assimilation was improved in [21], and forecasting models were compared under pandemic conditions in [22]. Additionally, research [23] introduced a fuzzy recurrent neural model optimized by swarm intelligence, and improved learning was demonstrated by injecting partial physics knowledge into DNNs [24]. Table 1 illustrates the Existing studies with key methods and findings.

Table 1: Summary of related works with key methods and results

Ref	Model	Application Domain	Evaluation Metrics & Results	Key Contribution
Wei et al., [14]	RNN, LSTM, GRU vs MLP	PWP (Slope Stability)	LSTM & GRU achieved better robustness; MLP performed acceptably; standard RNN struggled with delay	LSTM & GRU gave more stable forecasts under lagged inputs
Feng et al., [15]	Hybrid SVM	Monthly Runoff Forecasting	Lower RMSE vs. conventional methods; exact values not stated	Handles nonstationary signals via decomposition
Hu et al., [16]	Non-deterministic WSO+ BST	Dynamic Economic Dispatch	RMSE (Clear): 2.68% vs 5.43% (LS-SVM)	Enhanced wind power forecasting accuracy across weather conditions and reduced scheduling time using a hybrid evolutionary optimization approach
Candadai and Izquierdo [17]	Random Dynamical System Model	Predictive Information in Agents	Conceptual; no metrics given	Shows that the prediction size varies with environmental dynamics

Saha et al., [18]	PhICNet (Conv-RNN + Physics)	Spatiotemporal Forecasting w/ Hidden Source	Accurate long-term prediction in spatiotemporal fields; quantitative metrics not specified	Integrates physics & deep learning for source identification
You et al., [19]	STNN (Spatiotemporal Transformer)	Short-Term Time Series Forecasting	Outperforms baselines; reconstructs phase space of systems; detailed metrics not given	Uses continuous attention for accurate forecasting
Li et al., [20]	SDPA-Net	Sales Forecasting	↑46% CORR, ↓39.5% RRSE, ↓41.5% RMAE over baseline	Combines change-sensitive loss, hierarchical attention, and regression
Cheng and Qiu [21]	LSTM for Covariance Learning	Data Assimilation in Dynamics	Better assimilation accuracy & lower computational cost than existing methods	Learns observational covariance directly from data
Oukhouya et al., [22]	ARDL, LSTM, XGBOOST	MASI (Financial Index Forecasting)	ARDL: MAPE = 26.7%, Processing time = 1 sec	ARDL captured trend-seasonality well in the pandemic context
Nasiri and Ebadzadeh [23]	MFRFNN + PSO	Chaotic Time Series + Real Datasets	RMSE significantly reduced vs. conventional NN	Combines fuzzy logic and recurrent feedback for precision
Robinson et al., [24]	Physics-Guided DNN	Nonlinear Dynamic Systems (Lorenz, etc.)	Enhanced convergence, reduced uncertainty, ↑ and accuracy with partial physics input	Improves learning by integrating physical priors

1.1 Problem statement

Dynamic systems, characterized by nonlinear, time-dependent behavior, are fundamental across domains such as hydrology, climate modeling, structural analysis, and financial forecasting. Traditional machine learning approaches often fail to capture temporal dependencies and nonlinearities effectively, prompting the rise of neural network-based methods. It has explored various architectures, including LSTM, GRU, transformer models, and hybrid physics-informed frameworks. For instance, LSTM and GRU have demonstrated robustness in slope stability analysis [14], while hybrid SVM models addressed uncertainty in runoff prediction [15]. Others, like PhICNet [18] and physics-guided DNNs [24], blend physical laws with deep learning for improved accuracy and interpretability. To overcome these challenges, the proposed LST-NODE-WWO model integrates LSTM's temporal learning, NODE's continuous-time dynamics, and WWO's global optimization. This hybrid approach adaptively discovers optimal architectures, enhances long-term prediction accuracy, and embeds dynamic system behavior, leading to improved generalization, interpretability, and robustness in complex nonlinear dynamic system forecasting.

3 Methodology

This methodology combines time-series data acquisition, z-score normalization, and DWT for effective preprocessing and feature extraction. It integrates LSTM for capturing temporal dependencies, Neural-ODEs for continuous dynamic modeling, and WWO for tuning architecture and parameters. Together, these components form a hybrid LST-NODE-WWO model for accurate nonlinear dynamic system prediction. Figure 1 gives the overall flow of the methodology.

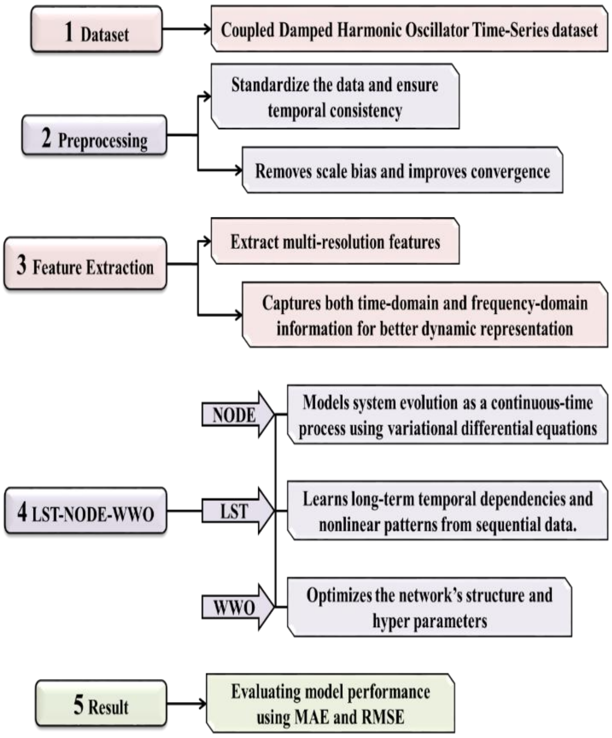

Figure 1: Overall flow of methodology

1.2 Dataset

The Coupled Damped Harmonic Oscillator Time-Series dataset represents the multivariate dynamics of two interacting masses connected through springs and dampers. Captured over 2763-time steps at uniform intervals of 0.01 seconds, the dataset includes detailed measurements of position (x1, x2), velocity (v1, v2), and acceleration (a1, a2) for both masses, offering a comprehensive view of their coupled motion. Each row corresponds to a specific point in time, and the system_id column distinguishes between different configurations or system simulations. This dataset is ideal for studying classical mechanical systems, analyzing second-order dynamic behavior, and applying multivariate time-series forecasting or physics-informed machine learning models. Source: https://www.kaggle.com/datasets/programmer3/coupled-damped-harmonic-oscillator-time-series

1.3 Pre-processing using z-score normalization

The normalization is applied to standardize time series input data, ensuring consistency across dynamic systems and improving model adaptability. It transforms feature values using their mean μ and standard deviation σ (Equation 1). This maps the data to a zero-centered scale, reducing the impact of outliers and improving the strength and accuracy of the proposed model during training and optimization.

$$u' = \frac{u-\mu}{\sigma} \quad (1)$$

Ensures that each feature has a mean of 0 and a standard deviation of 1, improving convergence during training. This normalization is particularly effective in handling the varying magnitudes of physical quantities (e.g., velocity vs. acceleration) and reducing model sensitivity to scale disparities.

1.4 Feature extraction using DWT

DWT is used to extract multi-resolution features from time-series data, offering better time-frequency localization than the STFT. In contrast to STFT, DWT offers high-frequency resolution for low frequencies and high time resolution for high frequencies, which is comparable to human hearing. The DWT uses a pair of filters; $h[]$ presents the high-pass filter to extract detail components. $g[]$ denotes the low-pass filter to extract approximation components. The decomposition is performed using Equations (2 and 3).

$$x_{high}[l] = \sum_m y[m]h[2l-m] \quad (2)$$

$$x_{low}[l] = \sum_m y[m]g[2l-m] \quad (3)$$

Where$x_{high}[l]$, and$x_{low}[l]$ are the outputs of high-pass (h) and low-pass (g) filters, respectively. $y[m]$ denotes the original time series input signal. $l$representsoutput sample index after down-sampling. m presents the input sample index. The process decreases dimensionality while retaining important temporal and frequency characteristics. DWT provides the LST-NODE-WWO model with the advantage of better learning dynamic behavior since it can capture short transients and longer trends.

1.5 Long short term neural ordinary differential water wave optimization (LST-NODE-WWO) for dynamic system prediction

The LST-NODE-WWO framework that was proposed combines the memory-oriented temporal learning of LSTMs with continuous modeling of physical consistency using NODE and the metaheuristics of WWO. The hybrid model provided adaptable architecture and hyperparameter selection with dynamically captured complex functions of time-varying behavior. As a result, it made accurate and generalizable predictions for nonlinear dynamic systems without user tuning, linking data-driven learning with the understanding of physical systems. WWO-LSTM-NODE Integration Strategy, detailing how WWO optimizes the LSTM-NODE model. After constructing the initial architecture, WWO explores hyperparameter configurations such as LSTM units, NODE depth, learning rate, and batch size. Using Propagation, Breaking, and Refraction, WWO generates candidate solutions, evaluates them by training the model, and selects the best based on validation performance. The final model is retrained using the optimal configuration, ensuring fully automated structure and training parameter optimization without manual tuning.

1.5.1 NODE

NODE is a type of DL model that represents system dynamics as a continuous-time process. NODE is similar to using traditional deep networks with distinct layers, but instead, NODE describes the transition of state in the system through DEs. The basic formulation is in Equation (4).

$$\dot{y}(t) = F\ (y(t), s) \tag{4}$$

Where $y(t)$ is the system state at time t; $\dot{y}(t)$ is the time derivative of $y(t)$; F is a neural network parameterized by time-dependent weights; θ represents the learnable parameters of the model; t is continuous time. Given an initial condition$y\,(0) = y_0$, an ODE solver integrates the equation to predict future system states y (t_k). This framework uses variational DEs to approximate the underlying physical laws of dynamic systems. Incorporating variational principles helps neural-ODEs generate smooth, accurate, and physics-consistent trajectories, making them especially suitable for modeling nonlinear, time-varying dynamic systems.

1.5.2 LSTM

LSTM networks improve upon RNNs as a sophisticated architecture that is capable of capturing information from data that is sequential with long-range temporal dependencies. The results of LSTMs can be used to predict within a dynamic system, capture periodic time-based behavior, and retain relevant information about the past over time. This makes LSTM highly capable in scenarios where standard RNNs often fall short, as follows in Equations (5 to 9).

$$f_s = \sigma(V_f \cdot [h_{s-1}, y_s] + a_f \tag{5}$$

$$j_s = \sigma(V_j \cdot [h_{s-1}, y_s] + a_j (6)$$

$$b_s = \tanh\,(V_b \cdot [h_{s-1}, y_s] + a_b \tag{7}$$

$$o_s = \sigma(V_o \cdot [h_{s-1}, y_s] + a_o \tag{8}$$

$$h_s = o_s \cdot \tanh\,(b_s) \tag{9}$$

Forget gate f_s determines what information from the previous state to discard. j_s input gate determines what new information to store. Candidate state b_s proposed new content to add to the cell state. The output gate o_s determines what part of the memory cell to output. Final hidden state h_s output of the LSTM for the current time step.

1.5.3 WWO

In the proposed LST-NODE-WWO model, WWO serves as a metaheuristic algorithm for automatically optimizing neural network structure and hyperparameters, eliminating the need for manual tuning. Inspired by the propagation of water waves, WWO represents each solution as a wave, where its position in the search space reflects a candidate solution, and its fitness indicates prediction accuracy.

Propagation adjusts each dimension c of the solution Y by a random displacement to explore the search space. The updated solution is given by Equation (10).

$$Y_c' = Y_c + rand\ (-1, 1) \cdot \lambda M_c \tag{10}$$

Where λ is the wave's **wavelength,**M_c is the range of the cth dimension, and rand(-1,1) is a uniform random number. Y_c: The value of the solution Y in the **dimension** (i.e., one specific parameter or component). Y_c': The **new (updated) value** after the operation is applied.

Breaking introduces local search behavior around promising solutions by generating variations using Gaussian perturbation (Equation 11). Where Gaussian (0,1) is a normally distributed random number (mean 0, standard deviation 1), and β is the breaking coefficient.

$$Y_c' = Y_c + Gaussian\ (0,1) \cdot \beta M_c \tag{11}$$

Refraction avoids stagnation by repositioning poor solutions based on the current best (Equation 12). Where $Y_{best,c}$ is the best solution's value in dimension c.

$$Y_c' = Gaussian\ \left(\frac{Y_{best,c} + Y_c}{2}, \frac{Y_{best,c} - Y_c}{2}\right) \tag{12}$$

After each operation, the wavelength is updated to balance exploration and exploitation (Equation 13).

$$\lambda' = \lambda \frac{f(Y)}{f(Y')} \tag{13}$$

The WWO algorithm is clearly defined and integrated with the LSTM-NODE model to automate hyperparameter tuning. Each solution Y is treated as a wave representing hyperparameters such as LSTM units, NODE depth, and learning rate. Propagation adjusts the dimension c as $Y_c' =$

$Y_c + rand\ (-1, 1) \cdot \lambda M_c$, where λ is the wavelength and M_c is the range of dimensions c. Breaking uses Gaussian noise: $Y_c' = Y_c + Gaussian\ (0,1) \cdot \beta M_c$, with β as the breaking coefficient. Refraction updates poor solutions via $Y_c' = Gaussian\ \left(\frac{Y_{best,c} + Y_c}{2}, \frac{Y_{best,c} - Y_c}{2}\right)$, where $Y_{best,c}$ is the best solution. Wavelength is updated with $\lambda' = \lambda \frac{f(Y)}{f(Y')}$, ensuring adaptive balance between exploration and exploitation in the optimization process.

Here,$f(Y)$is the current solution fitness, andϵ is the small constant to avoid division by zero. By embedding WWO into the LST-NODE-WWO framework, the model dynamically learns the most effective neural configurations, improving generalization and prediction accuracy for time-dependent, nonlinear dynamic systems. Algorithm 1 gives the pseudocode of the proposed LST-NODE-WWO method. Table 2 presents the hyperparameters of the proposed method.

Algorithm 1: LST-NODE-WWO

Input: Time-series dataset D (e.g., x1, x2, v1, v2, a1, a2)

Output: Optimized LST-NODE model for dynamic system prediction

1. Build Initial LST-NODE Model:

  a. Define an LSTM layer to capture long-range dependencies

  b. Integrate Neural ODE layer for continuous-time state evolution

  c. Combine into a single LST-NODE architecture

2. Train the Initial Model:

  a. Use time-series data to train the LST-NODE model

  b. Evaluate performance using MAE/RMSE on validation data

3. Optimize Model with Water Wave Optimization (WWO):

  a. Initialize a population of waves (each representing different hyperparameters)

  b. For each wave Y:

    i. Propagation: Y' = Y + rand(-1,1) * λ * Mc

    ii. Breaking (if promising): Y' = Y + Gaussian(0,1) * β * Mc

    iii. Refraction (if stagnant): Y' = Gaussian((Y_best + Y)/2, (Y_best - Y)/2)

    iv. Update wavelength: λ' = λ * (f(Y) / f(Y'))

    v. Evaluate fitness of each wave (i.e., retrain + validate)

  c. Select best-performing wave configuration Y_best

4. Retrain Final Model:

  a. Use the best parameters from Y_best to reconfigure LST-NODE

  b. Retrain using full training data

5. Prediction:

  a. Use the final trained model to forecast future dynamic states

Return: Final trained LST-NODE model with optimized structure

Table 2: The proposed LST-NODE-WWO model hyperparameters

Component	Parameter	Value
LSTM	Number of units	64
	Activation	tanh
	Optimizer	Adam
	Learning rate	0.001
NODE	ODE solver	Runge-Kutta (RK45)
	Time steps	0.01
	Tolerance	1e-5
WWO	Population size	30
	Number of iterations	100
	Breaking coefficient (β)	0.3
	Wavelength λ range	[0.1, 0.5]
	Fitness function	MAE on validation set

2 Result and discussion

The output behavior of the proposed LST-NODE-WWO model, including velocity, position, and acceleration plots, followed by quantitative performance comparisons, was presented in this section. Results demonstrate the model’s superior ability to capture and predict nonlinear dynamics with high accuracy and stability. Table 3 shows the experimental setup.

Table 3: Experimental setup

Component	Specification
Software	
Programming Language	Python 3.9
Deep Learning Framework	TensorFlow 2.11
Libraries Used	NumPy, SciPy, PyWavelets, Matplotlib, Scikit-learn
Hardware	
CPU	Intel Core i7-10750H
RAM	16 GB
GPU	NVIDIA GTX 1660 Ti (6 GB)
Operating System	Windows 10 / Ubuntu 22.0

2.1 Output phase

The velocities v1 and v2 of the two coupled masses are shown in Figure 2. Each signal has damped oscillatory motion, slowly lessening in amplitude due to loss of energy. The curves show phase separation that indicates the coupled interaction and the effectiveness of the proposed model to learn dynamic patterns.

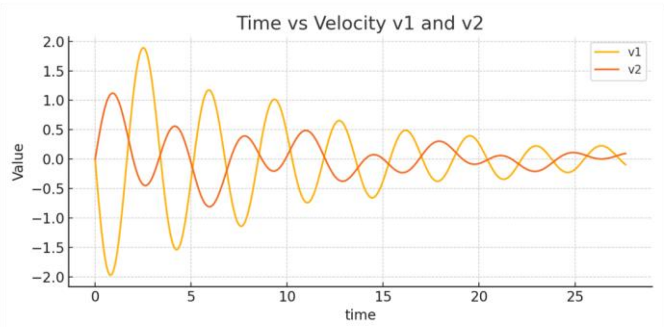

Figure 2:Velocity-time plot of oscillators v1 and v2, showing phase-shifted damped motion

Figure 3displays the position data x1and x2over time for both oscillators. The motion starts with high amplitude and converges as damping increases. The proposed model accurately captures the out-of-phase yet coupled behavior of the masses, demonstrating the system's energy dissipation and synchronization over time.

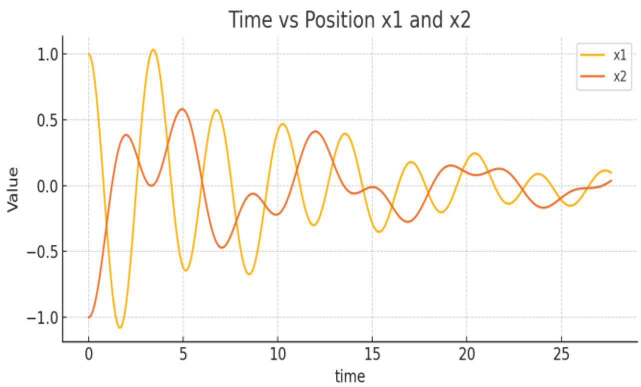

Figure 3: Position-time plot for x1 and x2, illustrating coupled damped oscillations.

Figures 4 and 5 show acceleration values a1 and a2, respectively. High-frequency oscillations are observed initially, gradually decaying as the system stabilizes. These results verify the model's capability to capture second-order dynamics and transient responses of the coupled system under damping and spring force interactions.

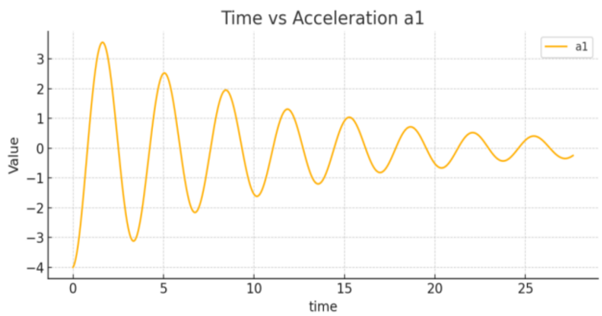

Figure 4: Acceleration-time plots for (a) a1, highlighting damped transient behavior.

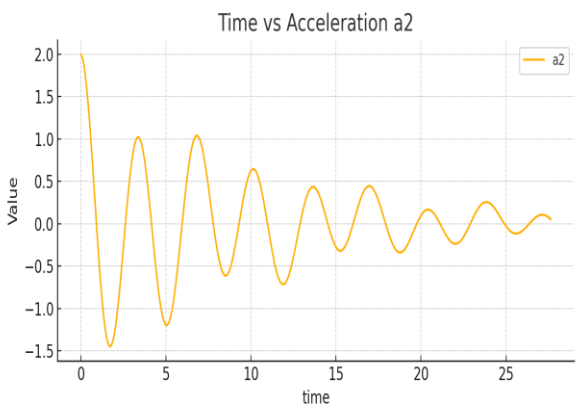

Figure 5: Acceleration-time plots for a2, highlighting damped transient behavior.

2.2 Comparison phase

The Two Coupled Damped Harmonic Oscillators system is a classical dynamic model used to simulate the interaction between two masses connected by springs and subjected to damping forces. It is widely used to test the effectiveness of time-series prediction models due to its nonlinear, oscillatory, and time-dependent behavior.

To assess the predictive accuracy of the proposed LST-NODE-WWO model, it was compared against two quantum-inspired baselines, QLSTM and QGRU [25], using MAE and RMSE on both training and testing sets.

Table 4: Error comparisons between Oscillator 1's train and test

Model	MAE		RMSE	
	Train	**Test**	**Train**	**Test**
QLSTM [25]	0.5284	0.4763	0.6987	0.5374
QGRU [25]	0.2585	0.2411	0.3587	0.2701
LST-NODE-WWO [Proposed]	0.1123	0.0914	0.1539	0.1298

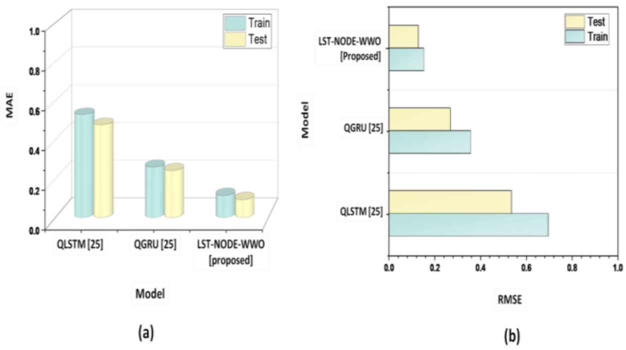

Figure 6: Performance comparison of models with (a) MAE and (b) RMSE

Table 5: Error comparisons between Oscillator 2's train and test

Model	MAE		RMSE	
	Train	**Test**	**Train**	**Test**
QLSTM [25]	0.1244	0.2591	0.1607	0.2885
QGRU [25]	0.0794	0.0482	0.0949	0.0602
LST-NODE-WWO [Proposed]	0.0276	0.0213	0.0365	0.0301

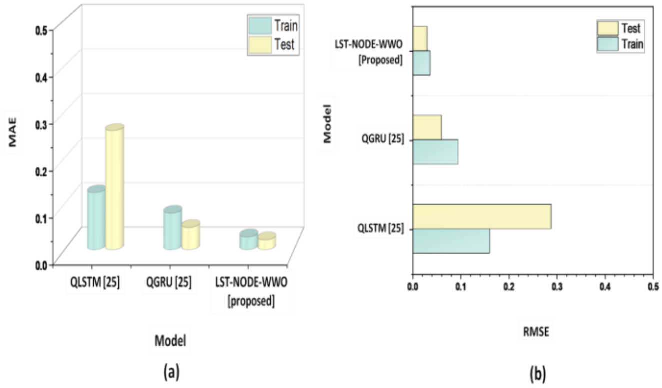

Figure 7: Performance comparison of models with (a) MAE and (b) RMSE

As shown in Tables 4 and 5, the proposed LST-NODE-WWO model outperforms both QLSTM and QGRU across all metrics for both oscillators. For Oscillator 1 (Figure 6 (a, b)), the LST-NODE-WWO achieves the lowest MAE and RMSE values on both training and testing sets, indicating high learning efficiency and generalization. Similarly, when looking at Oscillator 2 (Figure 7 (a, b)), the proposed model provides a significant reduction in prediction error, especially for the test phase, demonstrating its generalizability when forecasting unseen dynamic patterns.

Computational efficiency

Table 6 compares QLSTM, QGRU, and the proposed LST-NODE-WWO in terms of computational efficiency. LST-NODE-WWO requires the highest training time (42s), inference time (7.5ms), memory (3.2GB), and model size (4.7MB), reflecting its complex hybrid design. In contrast, QGRU is the most efficient. The trade-off suggests LST-NODE-WWO prioritizes accuracy over lightweight performance.

Table 6: Comparison of computational metrics across baseline and proposed prediction models.

Metric	**QLSTM [25]**	**QGRU [25]**	**LST-NODE-WWO (Proposed)**
Training Time	29 s	24 s	42 s
Total Training Time	24 min	20 min	35 min
Inference Time	5.1 ms	4.3 ms	7.5 ms
Memory Usage	2.1 GB	1.8 GB	3.2 GB
Model Size	3.2 MB	2.8 MB	4.7 MB

Existing methods, such as QLSTM [25] and QGRU [25] struggle to capture continuous dynamics and often require tedious manual tuning of hyperparameters, which limits their generalizability and accuracy for complex dynamical systems. The suggested LST-NODE-WWO is a hybrid approach that employs continuous time modelling, increases the time efficiency, and removes the need for explicit hyperparameter tuning, allowing for better adaptations to the wide range of possible nonlinear, time-variant behaviours that are naturally endowed to this domain, along with resulting in higher performance on dynamic system forecasting challenges. Overall, the findings support the efficacy of employing LSTM for temporal learning, NODE for continuous time modelling, and WWO for structural optimization. The proposed model functions quite well through combinations of these frameworks for accurate and stability-based forecasting of complex dynamic systems, demonstrating the plausibility of using a generalizable model to help quickly generate solutions to real-world time-series forecasting tasks.

3 Conclusion

The hybrid neural architecture called LST-NODE-WWO introduced a framework for dynamic system prediction that incorporates LSTM, NODE, and WWO. The LST-NODE-WWO model integrates both the temporal dependencies and continuous-time dynamics of the system while optimizing its architecture automatically. Applying LST-NODE-WWO to a coupled damped harmonic oscillator resulted in both smaller MAE (0.0213) and RMSE (0.0301) for Oscillator 2 relative to standard modelling approaches, demonstrating LST-NODE-WWO can be both accurate and robust. The model does not require any manual tuning and can adapt to the non-linear, time-varying behavior of real-world systems. The LST-NODE-WWO architecture, with its generalizability, could enable a variety of dynamic prediction problems across many fields in engineering and science. Furthermore, we clarified the **interaction mechanism** between LSTM, NODE, and WWO, showcasing how the model combines data-driven learning, differential modeling, and adaptive optimization. This comprehensive integration makes the proposed architecture suitable for modeling **nonlinear, time-varying systems** and opens avenues for applications in **real-time or high-dimensional dynamic environments**. Future work will focus on expanding the framework to handle more complex systems, leveraging hardware-accelerated solvers, and improving real-time deployment scalability.

Appendix

Abbreviation	Full form
NODE	Neural Ordinary Differential Equations
VDE	variational differential equation
WWO	Water Wave Optimization
DL	deep learning
DEs	differential equation
NN	Neural networks
RNN	recurrent NN
DNN	Deep neural networks
PWP	pore-water pressure
LSTM	long short-term memory
ANN	artificial NN
MLP	multi-layer perceptron
GRU	gated recurrent unit
KOT	Koopman Operator Theory
STNN	spatiotemporal transformer NN
SDPA	Spatiotemporal Dynamic Pattern Acquisition Mechanism
MASI	Moroccan All Shares Index
MAPE	Mean Absolute Percentage Error
MFRFNN	multi-functional recurrent fuzzy NN
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
QLSTM	Quantum LSTM
QGRU	Quantum GRU
PhICNet	Physics-Incorporated Convolutional Recurrent Neural Networks
ARDL	AutoRegressive Distributed Lag
XGBOOST	Extreme Gradient Boosting
STFT	short-time Fourier Transform
DWT	discrete wavelet transform
WSO	Whale Swarm Optimization
BST	Bee Species Transition

References

[1] PanahiS,& Lai YC (2024). Adaptable reservoir computing: A paradigm for model-free data-driven prediction of critical transitions in nonlinear dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 34*(5). https://doi.org/10.1063/5.0200898

[2] Yang, C., & Li, B. (2025). SDN-DRLTE Algorithm Based on DRL in Computer Network Traffic Control. *Informatica*, *49*(13). https://doi.org/10.31449/inf.v49i13.7576

[3] Uribarri G &Mindlin GB (2022). Dynamical time series embeddings in recurrent neural networks. *Chaos, Solitons& Fractals, 154*, 111612. https://doi.org/10.1016/j.chaos.2021.111612

[4] Chen N,&Majda AJ (2020). Predicting observed and hidden extreme events in complex nonlinear dynamical systems with partial observations and short training time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 30*(3). https://doi.org/10.1063/1.5122199

[5] Bertozzi O, Chamorro HR, Gomez-Diaz EO, Chong MS, & Ahmed S (2024). Application of data-driven methods in power systems analysis and control. *IET Energy Systems Integration*, *6*(3), 197–212. https://doi.org/10.1049/esi2.12122

[6] Song X, Liu Y, Xue L, Wang J, Zhang J, Wang J, Jiang L, & Cheng Z (2020). Time-series well performance prediction based on a Long Short-Term Memory (LSTM) neural network model. *Journal of Petroleum Science and Engineering*, *186*, 106682. https://doi.org/10.1016/j.petrol.2019.106682

[7] Islam BU,& Ahmed SF (2022). Short-term electrical load demand forecasting based on LSTM and RNN deep neural networks. *Mathematical Problems in Engineering*, *2022*(1), 2316474. https://doi.org/10.1155/2022/2316474

[8] Bandara K, Bergmeir C, &Smyl S (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, *140*, 112896. https://doi.org/10.1016/j.eswa.2019.112896

[9] Yang, Y., & Li, H. (2025). Neural ordinary differential equations for robust parameter estimation in dynamic systems with physical priors. *Applied Soft Computing*, *169*, 112649. https://doi.org/10.1016/j.asoc.2024.112649

[10] ZimmeringB,&Niggemann O (2024). Integrating continuous-time neural networks in engineering: Bridging machine learning and dynamical system modeling. *Machine Learning for Cyber Physical Systems*, 55.. DOI:10.24405/15313

[11] Liu, C., Quilliot, A., Toussaint, H., & Feillet, D. (2025). Dynamic Routing for Large-Scale Mobility On-Demand Transportation Systems. *Informatica*, *49*(1). https://doi.org/10.59615/ijie.3.1.55

[12] Cheng Y, Li G, Zhou X, & Ye S (2025). Research on time series forecasting models based on hybrid attention mechanism and graph neural networks. *Informatica*, *49*(21). https://doi.org/10.31449/inf.v49i21.7580

[13] Legaard C, Schranz T, Schweiger G, Drgoňa J, Falay B, Gomes C, Iosifidis A, Abkar, M, & Larsen P (2023). Constructing neural network-based models for simulating dynamical systems. *ACM Computing Surveys*, *55*(11), 1–34. https://doi.org/10.1145/3567591

[14] Wei X, Zhang L, Yang HQ, Zhang L, & Yao YP (2021). Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks. *Geoscience Frontiers*, *12*(1), 453–467. https://doi.org/10.1016/j.gsf.2020.04.011

[15] Feng Z K, Niu W J, Tang Z Y, Jiang ZQ, Xu Y, Liu Y, & Zhang HR (2020). Monthly runoff time series prediction by variational mode decomposition and support vector machine based on quantum-behaved particle swarm optimization. *Journal of Hydrology*, *583*, 124627. https://doi.org/10.1016/j.jhydrol.2020.124627

[16] Hu, Y., Yang, X., Chen, B., Gu, G., & Pan, L. (2025). Wind Power Prediction and Dynamic Economic Dispatch Strategy Optimization Based on BST-RGOA and NDO-WOA. *Informatica*, *49*(6). https://doi.org/10.31449/inf.v49i6.6940

[17] CandadaiM,&Izquierdo E J (2020). Sources of predictive information in dynamical neural networks. *Scientific Reports*, *10*(1), 16901. https://doi.org/10.1038/s41598-020-73380-x

[18] Saha P, Dash S, &Mukhopadhyay S (2021). Physics-incorporated convolutional recurrent neural networks for source identification and forecasting of dynamical systems. *Neural Networks*, *144*, 359–371. http://dx.doi.org/10.48550/arXiv.2004.06243

[19] You, Y, Zhang L, Tao P, Liu S, & Chen, L (2022). Spatiotemporal transformer neural network for time-series forecasting. *Entropy*, *24*(11), 1651. https://doi.org/10.3390/e24111651

[20] Li D, Lin K, Li X, Liao J, Du R, Chen D, & Madden A (2022). Improved sales time series predictions using deep neural networks with spatiotemporal dynamic pattern acquisition mechanism. *Information Processing & Management*, *59*(4), 102987. https://doi.org/10.1016/j.ipm.2022.102987

[21] Cheng S,&Qiu M (2022). Observation error covariance specification in dynamical systems for data assimilation using recurrent neural networks. *Neural Computing and Applications*, *34*(16), 13149–13167. https://doi.org/10.1007/s00521-021-06739-4

[22] Oukhouya MH, Angour N, Aboutabit N, &Hafidi I (2025). Comparative analysis of ARDL, LSTM, and XGBoost models for forecasting the Moroccan stock market during the COVID-19 pandemic. *Informatica*, *49*(14). https://doi.org/10.31449/inf.v49i14.5751

[23] NasiriH,&Ebadzadeh MM. (2022). MFRFNN: Multi-functional recurrent fuzzy neural network for chaotic time series prediction. *Neurocomputing*, *507*, 292–310. https://doi.org/10.1016/j.neucom.2022.08.032

[24] Robinson H, Pawar S, Rasheed A, & San O (2022). Physics guided neural networks for modelling of non-linear dynamics. *Neural*

Networks, *154*, 333–345. https://doi.org/10.1016/j.neunet.2022.07.023

[25] Chen Y,&Khaliq A (2024). Quantum recurrent neural networks: Predicting the dynamics of oscillatory and chaotic systems. *Algorithms*, *17*(4), 163. https://doi.org/10.3390/a17040163