Chinese Journal of Physics 103 (2026) 1466–1485

Contents lists available at ScienceDirect

Chinese Journal of Physics

journal homepage: www.sciencedirect.com/journal/chinese-journal-of-physics

Research Paper

Exploring spatiotemporal patterns in the Kuralay-II system using a neural network

Zeeshan Amjad [a,b], Aamir Farooq [a,c], H.W.A. Riaz [d], Wen-Xiu Ma [a,e,f,g,*]

[a] *Department of Mathematics, Zhejiang Normal University, 321004, PR China*
[b] *Center for Theoretical Physics, Khazar University 41 Mehseti Str., Baku, AZ1096, Azerbaijan*
[c] *Department of Applied Mathematics and Data Science, Asian University for Women, Chattogram, 4000, Bangladesh*
[d] *School of Civil and Hydraulic Engineering, Ningxia University, Yinchuan, 750021, PR China*
[e] *Research Center of Astrophysics and Cosmology, Khazar University 41 Mehseti Street, Baku, 1096, Azerbaijan*
[f] *Department of Mathematics and Statistics, University of South Florida Tampa, FL, 33620-5700, USA*
[g] *Material Science Innovation and Modelling, Department of Mathematical Sciences, North-West University Mafikeng Campus, Mmabatho, 2735, South Africa*

A R T I C L E I N F O

Keywords:
Kuralay system
Darboux transformation
Levenberg–Marquardt artificial neural network
Wave solutions
Monte Carlo simulations

A B S T R A C T

The Kuralay–II system is a coupled set of nonlinear partial differential equations with applications in nonlinear optics, ferromagnetic materials, and optical fibers. In this study, we investigated the spatiotemporal behavior of the system using a feedforward neural network trained on reference data compiled from traveling-wave reductions, where applicable, and high-fidelity numerical simulations across representative parameter ranges. The weights of the network are optimized with the Levenberg-Marquardt algorithm, and we assess the sensitivity to architectural choices, including the activation function family and the number and width of hidden layers. Robustness is evaluated through Monte Carlo trials with additive noise, and accuracy is summarized by a normalized Euclidean error over the space-time domain, complemented by contour and surface visualizations. The trained model reproduces the reference profiles with high fidelity and remains stable under noise, indicating that a simple neural network can serve as a fast tool for exploring spatiotemporal patterns in the Kuralay-II system and support downstream simulation and control tasks.

1. Introduction

Mathematical frameworks used to describe natural physical phenomena often rely on differential equations, which are particularly effective for analyzing dynamic systems. Nonlinear partial differential equations (NLPDEs) have been used to model nonlinear phenomena in natural and applied sciences, such as plasma and solid-state physics. These equations provide a deep understanding of the physical phenomena. Many scientists have made tremendous efforts to investigate various NLPDEs, exploring lucid insight of the behavior exhibited by physical phenomena such as the Navier-Stokes equations [1,2], Schrödinger equation [3,4], and Riemann wave equation [5]. The advancement of NLPDEs in the theory of solitary waves has significantly increased. Solitary waves have the characteristic feature of long-lasting propagation of energy, which is a stable localized solution of nonlinear evolution equations. Due

* Corresponding author.
E-mail addresses: z.amjadmath@zjnu.edu.cn (Z. Amjad), aamirf@zjnu.edu.cn (A. Farooq), wajahat@zjnu.edu.cn (H.W.A. Riaz), mawx@cas.usf.edu (W.-X. Ma).

https://doi.org/10.1016/j.cjph.2026.05.030
Received 27 October 2025; Received in revised form 17 May 2026; Accepted 26 May 2026
Available online 4 June 2026

to characteristic behaviour of solitary waves, these have been found novel applications in optical fibers [6,7], ferromagnetic materials [8,9], carbon nanomaterials [10,11].

This study focuses on the Kuralay-II system of equations (K-IIE), which is a type of Heisenberg ferromagnet equation with self-consistent potentials. The K-IIE stands out because of its ability to explain nonlinear dynamical properties over a wider range. In the study of nonlinear waves, dispersive effects, and solitons, K-IIE has become a cornerstone. In the regime of the ferromagnetic-type equation, K-IIE stands out as a benchmark, offering a significant contribution to the modeling of nonlinear phenomena in magnetic systems, especially the behavior of magnetic solitons and spin waves [12,13]. Its framework enables an in-depth analysis of the intricate interactions between material properties and magnetic fields, making it crucial for developing an understanding of how magnetic solitons behave in ferromagnetic materials under changing external conditions. The K-IIE has two principal forms: K-IIAE and K-IIBE, each providing specific deep insights into particular magnetic phenomena. Among these two forms, K-IIAE is a particular type of Heisenberg ferromagnet equation.

Various mathematical methods have been applied to investigate the deep structural complexity of the Kuralay system. Techniques such as the newly developed extended direct algebraic method have led to soliton solutions, revealing the stability characteristics and complex dynamics of soliton interactions [14]. Furthermore, perturbative techniques have been used to analyze small-amplitude solutions, shedding light on their impact and stability on the propagation of wave in magnetic systems [15,16], while Hirota method has been employed to obtain multi-soliton solutions [14]. Bifurcation theory was applied to analyze phase diagrams in magneto-optic waveguides, leading to the discovery of novel traveling wave solutions [17]. Through truncated M-fractional derivative using Jacobi Elliptic function expansion technique explored the exact solitary wave solutions of K-IIE [18]. Numerical results derived using the exponential cubic B-spline approach demonstrated the method's effectiveness in accurately capturing the dynamic behavior of the equation [19]. Analytical soliton solutions were presented using a new auxiliary equation method [20]. Furthermore, optical solitons were explored through a modified expansion of F and an extended version of the auxiliary equation method [21]. Recent developments have also explored variable-coefficient nonlinear wave equations, integrability structures, and machine-learning-assisted soliton modeling [22–24].

Numerous machine learning methodologieshave been developed to solve partial differential equations (PDEs). Among these, physics-informed neural networks (PINNs) incorporate governing physical laws into the loss function to efficiently approximate solutions [25]. In addition, several advanced neural-network-assisted analytical frameworks have recently emerged. These include the fractional sub-equation neural networks (fSENNs) method [26], bilinear neural network approaches [27], multimodal neurosymbolic reasoning algorithms [28], bilinear residual network methods [29], and neural-network-based symbolic computation techniques [30]. These methods provide new avenues for solving nonlinear evolution equations, discovering hidden structures, and enhancing interpretability in complex dynamical systems. Additional approaches include Deep Ritz methods [31], operator learning techniques such as DeepONets [32] and Fourier Neural Operators (FNOs) [33], and solvers based on reinforcement learning [34]. Furthermore, both supervised and unsupervised learning paradigms have been extensively employed. Supervised learning techniques utilize paired input-output data to train surrogate models for PDE solutions, facilitating the development of regression-based solvers [35–39]. Conversely, unsupervised learning techniques can approximate PDE solutions without labeled data, typically by directly minimizing the residuals or energy functionals [40–44]. Tian et al. [45,46] proposed advanced PINN variants capable of resolving solitons and rogue waves with high accuracy. In addition, physics-informed frameworks integrating variable-coefficient constraints and sparse-data reconstruction have been developed, including Res-KAN PDE discovery [47], VCINN inverse solvers [48], and self-adaptive equation-embedded networks for traffic-flow dynamics [49]. Collectively, these frameworks offer robust alternatives to traditional numerical solvers, providing efficiency, the ability to generalize high-dimensional systems, and the capacity to capture complex physical behaviors.

Building on these diverse methodologies, another significant class of approaches involves optimization-based training of neural networks. Notably, the Levenberg–Marquardt (LM) algorithm has emerged as a highly efficient and stable second-order method for the supervised training of feedforward neural networks [50]. Its capacity to merge the rapid convergence of Newton-type methods with the robustness of gradient descent renders it particularly suitable for nonlinear problems in scientific research. Recent studies have demonstrated the efficacy of the LM-artificial neural network (LM–ANN) in approximating soliton solutions of nonlinear evolution equations, achieving high accuracy and computational efficiency compared with conventional solvers. For example, LM–ANN has been successfully applied to approximate solitonic structures of the nonlocal short pulse equation, demonstrating accurate and robust performance under noise [51]. The combination of analytical soliton construction with LM–ANN has also been demonstrated for peakon and solitary wave solutions of the Lax equation [52]. Similarly, a hybrid framework combining the Darboux transformation (DT) with LM–ANN has been employed to study solitonic solutions of the reduced Maxwell–Bloch equations in nonlinear wave dynamics [53], while LM–ANN has also been utilized to investigate soliton solutions of the $(2+1)$ modified KdV equation with time-dependent coefficients [54]. Recently, Farooq et al. [55] applied the DT in combination with an LM–ANN to investigate nonlocal complex coupled dispersionless systems, demonstrating accurate analytical and numerical agreement. Their study revealed new symmetry-preserving and non-preserving wave structures, validated through relative L_2 error analysis on both clean and noisy datasets. Motivated by these theoretical convergence properties and demonstrated successes, this study employs the LM–ANN framework to construct accurate surrogate solutions for solitonic PDE models. In particular, we first derive explicit soliton and breather solutions of the K–IIE via DT, a powerful tool for constructing exact solutions, which, to the best of our knowledge, have not yet been reported for this model. These analytic solutions were then employed as a reference dataset to be approximated by an LM–ANN, providing an efficient numerical surrogate for the underlying dynamics. This framework not only uncovers new classes of solutions but also highlights their potential relevance to practical applications in nonlinear optics, plasmas, and fluid mechanics.

To construct the special structure of the K–IIE, we consider the AKNS spectral problem as

$$\Phi_x = E\Phi = (i\zeta\Lambda + Q)\Phi, \tag{1.1}$$

$$\Lambda = \begin{pmatrix} \alpha_1 I_m & 0 \\ 0 & \alpha_2 I_n \end{pmatrix}, Q = \begin{pmatrix} 0 & q \\ r & 0 \end{pmatrix},$$

where q and r are the two matrices given by

$$q = \left(q_{jk}\right)_{m\times n},\ r = \left(r_{kj}\right)_{n\times m}.$$

We formulate the Lax operator as follows

$$\Phi_t = F\Phi = \frac{1}{1-\alpha\zeta}W\Phi, \tag{1.2}$$

where $\alpha = \alpha_1 - \alpha_2$ and

$$W = W_1 + W_2, W_1 = -i\begin{bmatrix} v & 0 \\ 0 & w \end{bmatrix}, W_2 = \begin{bmatrix} 0 & -q_t \\ r_t & 0 \end{bmatrix}, \tag{1.3}$$

with v and w being two square matrices of orders m and n, respectively. The zero-curvature equation is

$$E_t - F_x + [E, F] = 0, \tag{1.4}$$

is equivalent to the following equation

$$(1-\alpha\zeta)E_t - W_x + [E, F] = 0, \tag{1.5}$$

which leads

$$\begin{aligned} -\alpha Q_t + i[\Lambda, W] &= 0, \\ Q_t - W_x + [Q, W] &= 0. \end{aligned} \tag{1.6}$$

The first part of (1.6) is satisfied, since we have

$$\left[\Lambda, W_1\right] = 0, \left[\Lambda, W_2\right] = -i\alpha Q_t. \tag{1.7}$$

Now, the second part of (1.6) leads

$$\begin{aligned} iq_t - q_{tx} + qw - vq &= 0, \\ ir_t + r_{tx} + rv - wr &= 0, \\ v_x + (qr)_t &= 0, \\ w_x - (rq)_t &= 0, \end{aligned} \tag{1.8}$$

which is the matrix generalization of K-IIE, belongs to the class of integrable coupled dispersionless systems. These Eq. (1.8) describe the nonlinear interactions between the primary fields q and r in a regime where the dispersive effects are negligible. In addition, the additional fields v and w arise naturally from the integrability structure of the system and are self-consistent potential fields, which can be evaluated through the constraints $v_x = -(qr)_t$ and $w_x = (rq)_t$. These fields are dynamically relevant and ensure the closure and integrability of the system. From a mathematical point of view, (1.8) admits the Lax pair representation and supports exact solutions, which motivates the study of its breather and quasi-Grammian solutions. Interestingly, (1.8) has no dependence on α. Even when $\alpha = 0$, the Lax pair (1.1)-(1.2) still generates Eq. (1.8). For $m = n = 1, w = -v$, we obtain

$$\begin{aligned} iq_t - q_{tx} - 2vq &= 0, \\ ir_t + r_{tx} + 2vr &= 0, \\ v_x + (qr)_t &= 0. \end{aligned} \tag{1.9}$$

Eq. (1.9) are presented as an integrable reduction of the matrix K-IIE (1.8), obtained by imposing suitable constraints on the associated potential fields. Although (1.9) has a simpler structure, it preserves the essential integrable properties of the original system and continues to describe coupled nonlinear wave dynamics in a dispersionless setting. This reduced form is particularly useful for analytical investigations and for constructing explicit exact solutions, while retaining the main physical and mathematical characteristics of the system.

The remainder of this paper is organized as follows. Section 2 outlines the overall methodology, including the DT in Section 2.1 and the LM–ANN in Section 2.2. Section 3 presents the application of these approaches: the DT is employed to derive analytical solutions in Section 3.1, whereas the LM–ANN is applied to approximate these solutions in Section 3.2. Finally, Section 4 summarizes the main findings and suggests possible directions for future research to be conducted.

2. Methods

This section summarizes the overall methodology for solving the system (1.9) by employing the DT to construct exact solutions and the LM–ANN algorithm to approximate them.

2.1. Darboux transformation

The DT is the strongest algebraic tool for deriving new exact solutions of NLPDEs. Originating from the theory of linear differential equations, it enables the iterative generation of multi-soliton solutions from a seed solution by systematically modifying the associated linear spectral problem (for details, see [56,57]). The DT can also be applied to continuous and discrete integrable systems [58]. In addition, the DT can be applied to supersymmetric integrable models to derive supersolitons [59]. Soliton and breather solutions are explored through DT for non-commutative K-IIE [60]. The matrix generalization and DT for K-IIE is studied in [61], but here we present DT for the system and express the solutions in terms of quasideterminants. Now, we define DT on matrix solution Φ as

$$\Phi[1] = D(\zeta)\Phi, \tag{2.1}$$

where $D(\zeta)$ is the Darboux matrix. Under the action of DT, new transformed solution $\Phi[1]$ has to satisfy the following linear system (1.1)- (1.2) as

$$\begin{aligned} \Phi_x[1] &= E[1]\Phi[1], \\ \Phi_t[1] &= F[1]\Phi[1], \end{aligned} \tag{2.2}$$

having $E[1]$ and $F[1]$ as,

$$\begin{aligned} E[1] &= (i\zeta\Lambda + Q[1]), \\ F[1] &= \frac{1}{1-\alpha\zeta}W[1]. \end{aligned} \tag{2.3}$$

Now, the Darboux matrix can be defined as

$$D(\zeta) = \zeta I - R, \tag{2.4}$$

where $R = \Pi\Upsilon\Pi^{-1}$ is 2×2 an auxiliary matrix and $I = diag(1,1)$. The distinct matrix solution Π can be defined as

$$\Pi = \left(\Phi(\zeta_1)|f\rangle_1, \Phi(\zeta_2|f\rangle_2)\right), \tag{2.5}$$

evaluated at

$$\Upsilon = \mathrm{diag}(\zeta_1, \zeta_2). \tag{2.6}$$

By using the expressions (2.5), (2.6), the linear system (1.1)-(1.2) can be written in the matrix form as

$$\Pi_x = i\Lambda\Pi\Upsilon + Q\Pi, \tag{2.7}$$

$$\Pi_t(I-\alpha\Upsilon) = W\Pi. \tag{2.8}$$

Based on the above discussion, the following theorem can be proved.

Theorem 2.1. *By operating DT (2.4), solution (2.3) has the same shape as that of W in (1.2), provided that the matrix R should have to satisfy these conditions*

$$W[1] = W + R_t, \tag{2.9}$$

$$R_t(I-R) = [W, R]. \tag{2.10}$$

Proof. The connection between $W[1]$ and W is developed and expressed in (2.9). So, we have to verify that $R = \Pi\Upsilon\Pi^{-1}$ satisfies the condition given by (2.10). Taking

$$\begin{aligned} R_t(I-R) &= \left(\partial_t\Pi\right)\Upsilon\Pi^{-1}(I-R) - R\left(\partial_t\Pi\right)\Pi^{-1}(I-R), \\ &= \left(W(I-R)^{-1}R - RW(I-R)^{-1}\right)(I-R), \\ &= \left(WR(I-R)^{-1}R^{-1}R - RW(I-R)^{-1}\right)(I-R), \\ &= [W, R]. \end{aligned}$$

□

Remark: Thus, the matrix $R = \Pi\Upsilon\Pi^{-1}$ proves to be the best choice, which satisfies all the conditions imposed by the DT. This is the preservation of the system; that is, if Φ, W are the solutions of the Lax pair (1.2), then $\Phi[1]$, $W[1]$ are also the solutions of the same equations.

By using (2.9), DT on dynamical solutions v and q can be written as

$$\begin{aligned} v[1] &= v + i\partial_t R_{11}, \\ q[1] &= q + iR_{12}, \end{aligned} \tag{2.11}$$

where we have used $r = -q^*$. Now, we express the solutions in terms of the quasideterminants. The one-fold DT (2.1) can be expressed by

$$\Phi[1] = \left(\zeta I - \Pi\Upsilon\Pi^{-1}\right)\Phi,$$

$$= \begin{vmatrix} \Pi & \Phi \\ \Pi\Upsilon & \boxed{\zeta I} \end{vmatrix}. \tag{2.12}$$

The K-fold DT is presented as

$$\Phi[K] = \begin{vmatrix} \Pi_1 & \Pi_2 & \cdots & \Pi_K & \Phi \\ \Pi_1\Upsilon_1 & \Pi_2\Upsilon_2 & \cdots & \Pi_K\Upsilon_K & \zeta\Phi \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Pi_1\Upsilon_1^{(K-1)} & \Pi_2\Upsilon_2^{(K-1)} & \cdots & \Pi_K\Upsilon_K^{(K-1)} & \zeta^{(K-1)}\Phi \\ \Pi_1\Upsilon_1^K & \Pi_2\Upsilon_2^K & \cdots & \Pi_K\Upsilon_K^K & \boxed{\zeta^K\Phi} \end{vmatrix}. \tag{2.13}$$

The expression (2.9) can also be expressed as

$$\begin{aligned} W[1] &= W - (O - \Pi\Upsilon\Pi^{-1})_t, \\ &= W - \begin{vmatrix} \Pi & I \\ \Pi\Upsilon & \boxed{O} \end{vmatrix}_t. \end{aligned} \tag{2.14}$$

The K-times DT on W can be written as

$$W[K] = W - \begin{vmatrix} \Pi_1 & \Pi_2 & \cdots & \Pi_K & O \\ \Pi_1\Upsilon_1 & \Pi_2\Upsilon_2 & \cdots & \Pi_K\Upsilon_K & O \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Pi_1\Upsilon_1^{(K-1)} & \Pi_2\Upsilon_2^{(K-1)} & \cdots & \Pi_K\Upsilon_K^{(K-1)} & I \\ \Pi_1\Upsilon_1^K & \Pi_2\Upsilon_2^K & \cdots & \Pi_K\Upsilon_K^K & \boxed{O} \end{vmatrix}_t. \tag{2.15}$$

Equation (2.15) is the K-fold quasideterminant expression of the matrix solution of the K-IIE.

2.2. Levenberg–Marquardt artificial neural network

This subsection presents the LM–ANN training procedure in a stepwise manner to provide a clear computational framework. The Levenberg–Marquardt algorithm, originally introduced by Kenneth Levenberg and Donald Marquardt, fuses the robustness of gradient descent with the fast convergence of Gauss–Newton [62–64]. The LM–ANN implemented here is a feedforward network designed to approximate nonlinear input–output mappings by minimizing the least-squares objective [65].

Approximating nonlinear PDE solution fields can be formulated as supervised regression on a compact domain, i.e., learning the nonlinear mapping $(x,t) \mapsto u(x,t)$ from sampled input–output pairs. Feedforward neural networks with nonlinear activation functions possess universal approximation capability, they can approximate continuous nonlinear functions on bounded domains to arbitrary accuracy provided sufficient representational capacity [66–68]. This theoretical foundation supports the use of ANNs as flexible surrogate approximators for complex wave patterns (e.g., localized solitons and oscillatory breathers) generated by nonlinear PDE systems. In the present study, the targets are analytically generated and densely sampled, so training reduces to minimizing a nonlinear least-squares objective under the MSE criterion. The Levenberg–Marquardt scheme is particularly suitable in this setting because it can be interpreted as a damped Gauss–Newton method that leverages Jacobian information to accelerate and stabilize convergence compared with purely first-order schemes for moderate-size networks [62–64,69].

For clarity, the complete LM–ANN implementation adopted in this study is summarized below in a sequential stepwise framework.

Step 1: Data generation and preprocessing.

The training dataset is defined as

$$\mathcal{T} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_m, \mathbf{y}_m)\},$$

where $\mathbf{x}_j \in \mathbb{R}^{d_x}$ is the input and $\mathbf{y}_j \in \mathbb{R}^{d_y}$ is the target.

In the present study, these input–output pairs were generated by discretizing the analytical DT-based solutions over the computational domain. The generated dataset was subsequently divided into training, validation, and testing subsets to ensure accurate generalization and reduce overfitting.

Step 2: Network architecture initialization.

A fully connected feedforward neural network was initialized with two input neurons corresponding to (x,t) and one output neuron corresponding to the target solution value. The hidden layers employed nonlinear activation functions, while the output layer remained linear. Different hidden-layer depths and activation functions were tested to determine the optimal architecture.

Given network parameters $\mathbf{w} \in \mathbb{R}^p$, the ANN outputs an approximation

$$\hat{\mathbf{y}}_j(\mathbf{w}),$$

where p is the total number of trainable weights.

Step 3: Forward propagation and residual computation.

The sample residual is

$$\boldsymbol{\varepsilon}_j(\ell) = \mathbf{y}_j - \hat{\mathbf{y}}_j\big(\mathbf{w}(\ell)\big),$$

and stacking them produces

$$\mathbf{r}(\ell) = \left[\boldsymbol{\varepsilon}_1(\ell)^\top, \dots, \boldsymbol{\varepsilon}_m(\ell)^\top\right]^\top \in \mathbb{R}^{md_y}.$$

At this stage, the network prediction is compared against the analytical reference solution to quantify the approximation error over the computational domain.

Step 4: Error evaluation.

The global mean-square error is computed as

$$\mathcal{E}(\ell) = \frac{1}{m}\sum_{j=1}^{m} \|\boldsymbol{\varepsilon}_j(\ell)\|_2^2 = \frac{1}{m}\|\mathbf{r}(\ell)\|_2^2.$$

The MSE objective function serves as the principal optimization criterion during network training and architecture selection.

Step 5: Jacobian and gradient computation.

The Jacobian of residuals with respect to the parameters is

$$\mathrm{J}(\ell) = \frac{\partial \mathbf{r}}{\partial \mathbf{w}}\Big|_{\mathbf{w}(\ell)} \in \mathbb{R}^{(md_y)\times p}.$$

Using the Gauss–Newton approximation,

$$H \approx \mathrm{J}(\ell)^\top \mathrm{J}(\ell), \qquad \nabla\mathcal{E}(\ell) \approx \frac{2}{m}\mathrm{J}(\ell)^\top \mathbf{r}(\ell).$$

The Jacobian matrix provides sensitivity information of the residuals with respect to the trainable parameters and plays a central role in accelerating convergence within the LM optimization framework.

Step 6: Parameter updating using LM optimization.

The network parameters are updated as

$$\mathbf{w}(\ell+1) = \mathbf{w}(\ell) - \left[\mathrm{J}(\ell)^\top \mathrm{J}(\ell) + \lambda I\right]^{-1} \mathrm{J}(\ell)^\top \mathbf{r}(\ell),$$

where $\lambda \geq 0$ is the damping coefficient.

As $\lambda \to 0$, the method approaches Gauss–Newton; for large λ, it behaves like gradient descent.

This adaptive damping strategy improves numerical stability while maintaining fast convergence during the optimization process.

Step 7: Validation and optimal architecture selection.

The candidate architectures were evaluated using validation-based hyperparameter tuning. The optimal architecture was selected by minimizing the validation MSE across different hidden-layer depths and activation functions while simultaneously monitoring robustness and convergence stability.

Step 8: Robustness analysis using Monte Carlo simulations.

To evaluate robustness, Gaussian noise was added to both the inputs and outputs under different noise levels. Monte Carlo simulations with 25 independent trials were performed to analyze the statistical stability and predictive reliability of the LM–ANN approximations under perturbations.

In summary, LM–ANN provides a stable and efficient mechanism for minimizing reconstruction error through a hybrid of gradient and Gauss–Newton updates, making it well-suited for medium-scale networks [69].

3. Application of the Darboux transformation and the Levenberg–Marquardt artificial neural network

In this section, we study the Kuralay-II system (1.9) using a hybrid approach combining DT and LM–ANN. Analytical reference profiles obtained from DT-applied reductions were used to train and evaluate the network, providing an efficient surrogate for exploring the spatiotemporal behavior of the system.

3.1. Application of Darboux transformation

In this subsection, we apply the DT to obtain analytical quasideterminant solutions of the system (1.9) under both zero- and non-zero seed conditions. In particular, we obtain one-soliton (bright and dark), two-soliton (bright and dark), and breather (bright and dark) solutions, which demonstrate the diverse dynamic behavior of the model. These analytical solutions were subsequently employed as reference data for approximation using LM–ANN, providing an efficient numerical surrogate and deeper insight into the system dynamics.

First we re-write the matrix $R[K]$ from (2.15) in a more precise shape as

$$R[K] = \begin{vmatrix} \Pi & \mathfrak{I}^{(K)} \\ \widetilde{\Pi} & \boxed{O} \end{vmatrix}, \tag{3.1}$$

where $\widetilde{\Pi}$, $\mathfrak{I}^{(K)}$, and Π are $N \times NK$, $NK \times N$, and $NK \times NK$ matrices, respectively. These matrices are given by

$$\widetilde{\Pi} = \begin{pmatrix} \Pi_1 \Upsilon_1^K & \Pi_2 \Upsilon_2^K & \cdots & \Pi_K \Upsilon_K^K \end{pmatrix},$$
$$\mathfrak{I}^{(K)} = \begin{pmatrix} I & O & \cdots & O \end{pmatrix}^T,$$

$$\Pi = \begin{pmatrix} \Pi_1 & \Pi_2 & \cdots & \Pi_K \\ \Pi_1\Upsilon_1 & \Pi_2\Upsilon_2 & \cdots & \Pi_K\Upsilon_K \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_1\Upsilon_1^{(K-1)} & \Pi_2\Upsilon_2^{(K-1)} & \cdots & \Pi_K\Upsilon_K^{(K-1)} \end{pmatrix}. \tag{3.2}$$

The elements of $R[K]$ can be written as

$$R_{ij}[K] = \left(\begin{vmatrix} \Pi & \mathfrak{F}^{(K)} \\ \widetilde{\Pi} & \boxed{O} \end{vmatrix} \right)_{ij} = \begin{vmatrix} \Pi & \mathfrak{F}_j^{(K)} \\ \left(\widetilde{\Pi}\right)_i & \boxed{0} \end{vmatrix},$$
$$= (-1)^{i+j}\frac{\det(\Pi)_{ij}}{\det(\Pi)}, \quad i,j = 1,2,\dots,K, \tag{3.3}$$

where $(\widetilde{\Pi})_i$ and $\mathfrak{F}_j^{(K)}$ representing the ith row and jth column of matrices, respectively. Now taking the simpler case $N = 2$. The matrix $R[K]$ can be expressed by

$$R[K] = \begin{vmatrix} \Pi & \mathfrak{F}^{(K)} \\ \widetilde{\Pi} & \boxed{O_2} \end{vmatrix},$$

where

$$R_{ij}[K] = \begin{vmatrix} \Pi & \mathfrak{F}_j^{(K)} \\ (\widetilde{\Pi})_i & \boxed{0} \end{vmatrix} = (-1)^{i+j}\frac{\det(\Pi)_{ij}}{\det(\Pi)}, \quad i,j = 1,2. \tag{3.4}$$

To derive an explicit expression, we take seed solution, i.e., $q = 0$, $v = 2c$ (c is nonzero constant), the solution of the Lax pair (1.1)-(1.2) $\Phi = \begin{pmatrix}\psi_k & \phi_k\end{pmatrix}$ given by

$$\psi_k = \exp\left(i\alpha\zeta_k x - \frac{2ic}{1-c\zeta_k}t\right),$$
$$\phi_k = \exp\left(i\alpha\zeta_k x + \frac{2ic}{1-c\zeta_k}t\right).$$

$$\Pi = \begin{pmatrix} \psi_1 & \psi_2 \\ \phi_1 & \phi_2 \end{pmatrix}. \tag{3.5}$$

Also,

$$\Upsilon = \begin{pmatrix} \zeta_1 & 0 \\ 0 & \zeta_2 \end{pmatrix}. \tag{3.6}$$

For $K = 1$, i.e., $\psi_{2K} = \overline{\phi}_{2K-1}$, $\phi_{2K} = -\overline{\psi}_{2K-1}$, and parametric values: $c = 1,\ \alpha_1 = 1,\ \alpha_2 = -1,\ \zeta_1 = i/2, \zeta_2 = -i/2$, we obtain the explicit expressions of dark and bright one soliton (2.11) solutions can be expressed as

$$v[1] = \frac{1 - 2\cosh(-2x+t) + 2\sinh(-2x+t) + \cosh(-4x+2t) - \sinh(-4x+2t)}{(\cosh(-2x+t) - \sinh(-2x+t) + 1)^2} \tag{3.7}$$

$$q[1] = -\frac{\cos\left(\frac{t}{2}\right) - i\sin\left(\frac{t}{2}\right)}{\cosh\left(-x + \frac{t}{2}\right)}. \tag{3.8}$$

Eqs. (3.7) and (3.8) present one-wave solutions, illustrated in Figs. 1 and 2.

For the two-wave case, we set $K = 2$ in (3.4), i.e., $\psi_{2k} = \overline{\phi}_{2k-1},\ \phi_{2k} = -\overline{\psi}_{2k-1},\ k = 1,\ 2$, we have

$$v[2] = 2c + i\left(\frac{p_2}{s_2}\right)_t, \tag{3.9}$$

$$q[2] = i\left(\frac{r_2}{s_2}\right), \tag{3.10}$$

where

$$p_2 = \Bigg(e^{\zeta_1 xi - \frac{cti}{2-4\zeta_1}} e^{\zeta_2 xi - \frac{cti}{2-4\zeta_2}} (\zeta_1 - \zeta_2)(\zeta_1 + \zeta_2)(\overline{\zeta_1} - \overline{\zeta_2}) e^{-\overline{\zeta_2} xi + \frac{cti}{2-4\overline{\zeta_2}}}$$
$$- e^{\overline{\zeta_2} xi - \frac{cti}{2-4\overline{\zeta_2}}} \left(-e^{\zeta_2 xi - \frac{cti}{2-4\zeta_2}} (\zeta_2 - \overline{\zeta_2})(\overline{\zeta_2} + \zeta_2)(\zeta_1 - \overline{\zeta_1}) e^{-\zeta_1 xi + \frac{cti}{2-4\zeta_1}}\right.$$
$$\left.\left. + e^{\zeta_1 xi - \frac{cti}{2-4\zeta_1}} e^{-\zeta_2 xi + \frac{cti}{2-4\zeta_2}} (\zeta_1 - \overline{\zeta_2})(\overline{\zeta_2} + \zeta_1)(\zeta_2 - \overline{\zeta_1}) \right)\right) e^{-\overline{\zeta_1} xi + \frac{cti}{2-4\overline{\zeta_1}}}$$

$$
\begin{aligned}
&+\Bigg(-\mathrm{e}^{\zeta_2 x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\zeta_2}}(\zeta_2-\overline{\zeta_1})(\overline{\zeta_1}+\zeta_2)(\zeta_1-\overline{\zeta_2})\mathrm{e}^{-\zeta_1 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_1}}\\
&\quad +\mathrm{e}^{\zeta_1 x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\zeta_1}}\mathrm{e}^{-\zeta_2 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_2}}(\zeta_1-\overline{\zeta_1})(\overline{\zeta_1}+\zeta_1)(\zeta_2-\overline{\zeta_2})\Bigg)\mathrm{e}^{-\overline{\zeta_2}x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\overline{\zeta_2}}}\\
&+\mathrm{e}^{-\zeta_1 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_1}}\mathrm{e}^{\overline{\zeta_2}x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\overline{\zeta_2}}}\mathrm{e}^{-\zeta_2 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_2}}(\overline{\zeta_1}-\overline{\zeta_2})(\overline{\zeta_1}+\overline{\zeta_2})(\zeta_1-\zeta_2)\mathrm{e}^{\overline{\zeta_1}x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\overline{\zeta_1}}}
\end{aligned}
$$

$$
\begin{aligned}
s_2=&\Bigg(\mathrm{e}^{\zeta_1 x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\zeta_1}}\mathrm{e}^{\zeta_2 x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\zeta_2}}(\overline{\zeta_1}-\overline{\zeta_2})(\zeta_1-\zeta_2)\mathrm{e}^{-\overline{\zeta_2}x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\overline{\zeta_2}}}\\
&-\Big(-\mathrm{e}^{\zeta_2 x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\zeta_2}}(\zeta_2-\overline{\zeta_2})(\zeta_1-\overline{\zeta_1})\mathrm{e}^{-\zeta_1 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_1}}\\
&\quad+\mathrm{e}^{\zeta_1 x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\zeta_1}}\mathrm{e}^{-\zeta_2 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_2}}(\zeta_2-\overline{\zeta_1})(\zeta_1-\overline{\zeta_2})\Big)\mathrm{e}^{\overline{\zeta_2}x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\overline{\zeta_2}}}\Bigg)\mathrm{e}^{-\overline{\zeta_1}x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\overline{\zeta_1}}}\\
&+\mathrm{e}^{\overline{\zeta_1}x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\overline{\zeta_1}}}\Bigg(\Big(-\mathrm{e}^{\zeta_2 x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\zeta_2}}(\zeta_2-\overline{\zeta_1})(\zeta_1-\overline{\zeta_2})\mathrm{e}^{-\zeta_1 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_1}}\\
&\quad+\mathrm{e}^{\zeta_1 x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\zeta_1}}\mathrm{e}^{-\zeta_2 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_2}}(\zeta_2-\overline{\zeta_2})(\zeta_1-\overline{\zeta_1})\Big)\mathrm{e}^{-\overline{\zeta_2}x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\overline{\zeta_2}}}\\
&\quad+\mathrm{e}^{\overline{\zeta_2}x\mathrm{i}-\frac{ct\mathrm{i}}{2-4\overline{\zeta_2}}}\mathrm{e}^{-\zeta_1 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_1}}\mathrm{e}^{-\zeta_2 x\mathrm{i}+\frac{ct\mathrm{i}}{2-4\zeta_2}}(\overline{\zeta_1}-\overline{\zeta_2})(\zeta_1-\zeta_2)\Bigg),
\end{aligned}
$$

$$
\begin{aligned}
r_2=&\Bigg(\mathrm{e}^{-\overline{\zeta_2}x\mathrm{i}+\frac{t\mathrm{i}}{2-4\overline{\zeta_2}}}(\zeta_1-\zeta_2)(\overline{\zeta_1}-\zeta_1)\mathrm{e}^{\zeta_1 x\mathrm{i}-\frac{t\mathrm{i}}{2-4\zeta_1}}\\
&+\mathrm{e}^{-\zeta_1 x\mathrm{i}+\frac{t\mathrm{i}}{2-4\zeta_1}}\mathrm{e}^{\overline{\zeta_2}x\mathrm{i}-\frac{t\mathrm{i}}{2-4\overline{\zeta_2}}}(\overline{\zeta_2}-\zeta_2)(\overline{\zeta_1}-\overline{\zeta_2})\Bigg)(\overline{\zeta_1}-\zeta_2)\mathrm{e}^{\overline{\zeta_1}x\mathrm{i}-\frac{t\mathrm{i}}{2-4\overline{\zeta_1}}}\\
&-\mathrm{e}^{\overline{\zeta_2}x\mathrm{i}-\frac{t\mathrm{i}}{2-4\overline{\zeta_2}}}\mathrm{e}^{-\overline{\zeta_1}x\mathrm{i}+\frac{t\mathrm{i}}{2-4\overline{\zeta_1}}}\mathrm{e}^{\zeta_1 x\mathrm{i}-\frac{t\mathrm{i}}{2-4\zeta_1}}(\zeta_1-\zeta_2)(\overline{\zeta_2}-\zeta_2)(\overline{\zeta_2}-\zeta_1)\\
&\times\mathrm{e}^{\zeta_2 x\mathrm{i}-\frac{t\mathrm{i}}{2-4\zeta_2}}\\
&-\mathrm{e}^{\overline{\zeta_2}x\mathrm{i}-\frac{t\mathrm{i}}{2-4\overline{\zeta_2}}}\mathrm{e}^{-\zeta_2 x\mathrm{i}+\frac{t\mathrm{i}}{2-4\zeta_2}}\mathrm{e}^{\overline{\zeta_1}x\mathrm{i}-\frac{t\mathrm{i}}{2-4\overline{\zeta_1}}}\mathrm{e}^{\zeta_1 x\mathrm{i}-\frac{t\mathrm{i}}{2-4\zeta_1}}(\overline{\zeta_2}-\zeta_1)(\overline{\zeta_1}-\zeta_1)(\overline{\zeta_1}-\overline{\zeta_2}).
\end{aligned}
$$

The analytical expressions given by $q[2]$ and $\upsilon[2]$ enable a deeper investigation of the mechanisms governing the interactions between localized wave patterns, particularly the redistribution of energy during collision events. These formulations support a nuanced analysis of the behavior of solitary waves when subjected to mutual interactions, shedding light on their inherent resilience and structural integrity across various physical media. As shown in Figs. 3a–6a, the visualizations capture the complex interplay between the two solitary waves structures: each waveform exhibits pronounced localization and propagates through space-time with only limited deformation. Notably, the transient interference fringes observed during the interaction indicated phase modulation, manifesting as temporary oscillations. These phase shifts serve as evidence of nonlinear interactions but do not compromise the core identity of the waveforms. The persistence of the shape and velocity post-interaction reinforces the classification of solitary waves as robust, particle-like entities within nonlinear systems. Together, the visual and analytical evidence highlights the stability and coherence of these wave patterns, even in the face of nontrivial interactions.

Bright solitons correspond to localized pulses of increased intensity, which can model pulse transmission in optical fibers where energy is concentrated and propagates without dispersion. Dark solitons represent localized dips or intensity voids in the wave profile, which are relevant to optical waveguides and spin-wave dynamics in ferromagnetic materials, illustrating regions of reduced amplitude propagating stably. Also breather waves describe localized oscillatory excitations, which can model periodic energy exchange between modes in nonlinear optical media or temporal oscillations of magnetization in ferromagnetic lattices.

3.2. Applications of the Levenberg–Marquardt artificial neural network

In this subsection, LM–ANN is employed to approximate several analytical solutions of the governing system (1.9). The considered cases include explicit one-soliton profiles (3.7), (3.8) and two-soliton solutions (3.9), (3.10). These cases capture distinct nonlinear wave structures governed by the same system as in (1.9).

Three sets of experiments were performed to examine the performance of LM–ANN. First, each analytical solution was approximated through a network, and the results were compared with the reference data through surface, contour, and error plots. For all solution types (i.e., $|q[1]|$–bright, $|q[2]|$–bright, $|q[2]|$–breather, $\mathrm{Re}(\upsilon[1])$–dark, $\mathrm{Re}(\upsilon[2])$–dark, and $\mathrm{Re}(\upsilon[2])$–breather), the accuracy is quantified using the relative L_2 error,

$$E_{L_2}^{(m)} = \frac{\|u^{(m)} - \hat{u}^{(m)}\|_2}{\|u^{(m)}\|_2} = \frac{\left(\sum_{i=1}^{N}\left(u_i^{(m)} - \hat{u}_i^{(m)}\right)^2\right)^{\frac{1}{2}}}{\left(\sum_{i=1}^{N}\left(u_i^{(m)}\right)^2\right)^{\frac{1}{2}}},$$

where $u^{(m)}$ denotes the analytical solution, and $\hat{u}^{(m)}$ denotes the LM–ANN approximation. Specifically, the mapping is given by

$$u^{(1)} = |q[1]|, \quad u^{(2)} = |q[2]|, \quad u^{(3)} = |q[2]|,$$
$$u^{(4)} = \mathrm{Re}(v[1]), \quad u^{(5)} = \mathrm{Re}(v[2]), \quad u^{(6)} = \mathrm{Re}(v[2]).$$

Second, the influence of network architecture and activation functions was examined through validation-based comparisons. Third, the robustness was investigated by adding Gaussian noise to both the inputs and outputs under three different noise configurations. Monte Carlo simulations with 25 independent runs were conducted to assess the statistical stability of the network under noise. All datasets were synthetically generated from the analytical solutions obtained through DT using different spatial and temporal discretizations for each case. No external data were utilized in this study. All computations, including data generation, LM–ANN training, and testing under clean and noisy conditions, were implemented in MATLAB R2023a on a workstation equipped with an Intel Core i7-13700H CPU, 16 GB RAM, and Windows 11 Pro-(64-bit).

For each case, the solution was taken as either $|q[1]|$, $|q[2]|$, $\mathfrak{R}(v[1])$, or $\mathfrak{R}(v[2])$, depending on the seed configuration and the type of wave pattern (bright soliton, dark soliton, or breather). Each solution was sampled on a uniform domain $(x,t) \in [-10,10]^2$, discretized into 500 points per direction, yielding $N = 250{,}000$ input–output pairs.

The LM–ANN architecture employed in this study was a fully connected feedforward neural network. The input layer contained two neurons corresponding to the spatial and temporal variables (x,t), while the output layer contained one neuron corresponding to the target solution value. The hidden part of the network consisted of three or four hidden layers, and each hidden layer contained 15 neurons. Thus, the depth of the network was represented by the number of hidden layers, whereas the width was represented by the number of neurons in each hidden layer.

The use of three and four hidden layers was motivated by the nonlinear and localized nature of the considered bright soliton, dark soliton, and breather profiles. These wave structures contain sharp gradients, localized amplitudes, and oscillatory spatiotemporal behavior; therefore, a shallow network may not provide sufficient representational capability. At the same time, excessively deep or wide networks increase the computational cost of the LM algorithm because it relies on Jacobian-based updates. Hence, the selected architecture provides a practical balance between approximation accuracy, numerical stability, and computational efficiency.

To systematically determine the optimal network architecture, an architecture selection criterion based on validation performance was employed. Let

$$\mathcal{A} = \{A_1, A_2, \ldots, A_K\},$$

denote the set of candidate architectures, where each architecture differs in hidden-layer depth and activation configuration. For every candidate architecture A_k, the validation mean-square error was computed as

$$\mathrm{MSE}_{\mathrm{val}}^{(k)} = \frac{1}{N_{\mathrm{val}}}\sum_{i=1}^{N_{\mathrm{val}}}\left(u_i - \hat{u}_i^{(k)}\right)^2,$$

where u_i represents the analytical reference solution and $\hat{u}_i^{(k)}$ denotes the LM–ANN prediction corresponding to the architecture A_k.

The optimal architecture was selected according to

$$A_{\mathrm{opt}} = \arg\min_{A_k \in \mathcal{A}} \mathrm{MSE}_{\mathrm{val}}^{(k)}.$$

Architecture selection was performed using validation-based hyperparameter tuning over the candidate depths $\{3,4\}$ with 15 neurons per hidden layer. For each candidate architecture, training was repeated using multiple random initializations, and the final architecture was selected by minimizing the mean validation MSE. Early stopping based on validation performance was used to reduce the possibility of overfitting. The width of 15 neurons was selected after preliminary numerical experiments, where smaller networks produced larger approximation errors for localized and oscillatory structures, whereas larger networks increased computational time without providing a significant reduction in validation error.

In addition to minimizing the validation error, the selected architecture was required to exhibit stable convergence behavior and robustness under Monte Carlo noisy trials. Based on these criteria, the architectures containing three or four hidden layers with 15 neurons per hidden layer provided the best balance between approximation accuracy, numerical stability, and computational efficiency for the considered nonlinear wave structures.

The hidden layers used the hyperbolic tangent sigmoid activation function (`tansig`), whereas the output layer was linear.

$$\mathtt{tansig}(z) = \frac{2}{1+e^{-2z}} - 1, \qquad z \in \mathbb{R},$$

was adopted in all hidden layers, where $\mathbb{R}$ denotes the set of real numbers. This activation function is bounded, differentiable, and effective in representing nonlinear and oscillatory behavior, making it suitable for the present soliton and breather approximations. The linear output layer was used because the task is a continuous regression problem.

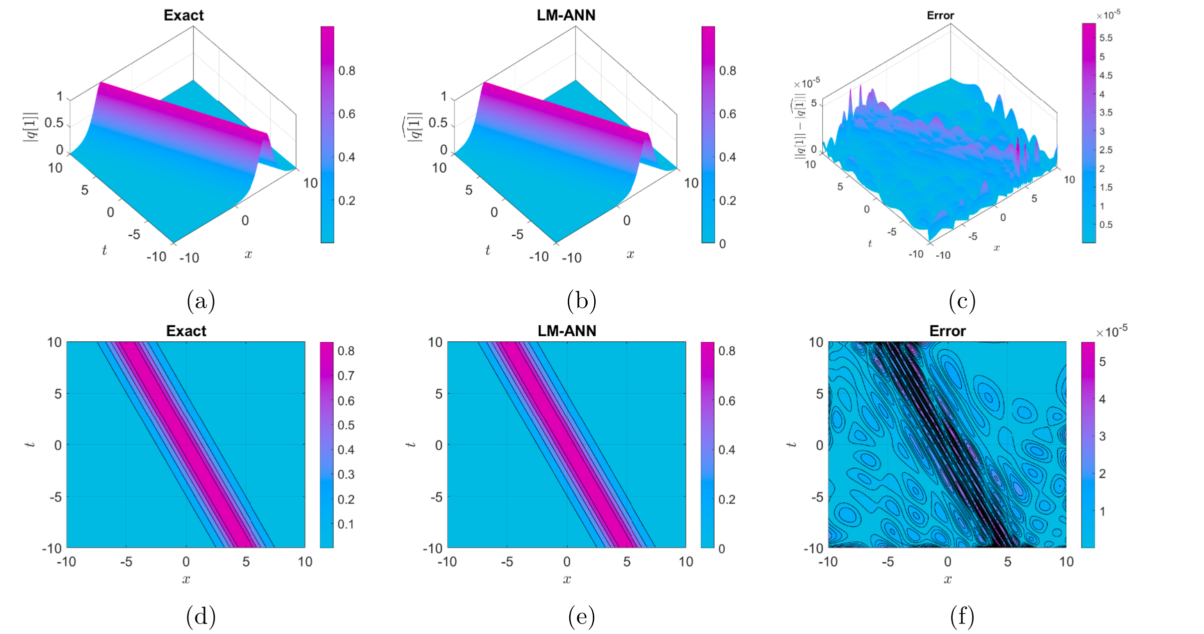

Fig. 1. The absolute plot of the $q[1]$. The figure consists of six subplots. (1a) exact solution $\left(|q[1]|\right)$, (1b) LM-ANN solution $\left(|\widehat{q[1]}|\right)$, (1c) error plot for 3D, (1d) contour plot of the exact solution $\left(|q[1]|\right)$, (1e) contour plot of the LM-ANN solution $\left(|\widehat{q[1]}|\right)$, and (1f) error plot for contour.

Table 1
$E^{(1)}_{L_2}$ for $u^{(1)}$ and the corresponding training time under different hidden layer configurations and activation functions.

Hidden Layers	tansig			logsig			radbas			Time (sec)
	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	
[3, 3, 3]	8.56488e-05	8.84821e-05	8.69096e-05	4.14819e-05	3.98316e-05	3.98291e-05	2.65314e-05	2.66491e-05	2.67686e-05	14.60 sec
[7, 7, 7]	1.25204e-05	1.22132e-05	1.32108e-05	3.04807e-05	2.99635e-05	3.03615e-05	4.52091e-05	4.72536e-05	4.53611e-05	18.27 sec
[10, 10, 10]	2.74856e-05	2.92218e-05	3.06902e-05	5.48880e-05	5.45277e-05	5.81577e-05	1.16540e-05	1.28063e-05	1.26795e-05	53.52 sec
[15, 15, 15]	1.24230e-05	1.31297e-05	1.34928e-05	4.79617e-05	5.20916e-05	4.99398e-05	2.32752e-05	2.39457e-05	2.42615e-05	124.29 sec
[3, 3, 3, 3]	3.48293e-04	3.47055e-04	3.68932e-04	1.05722e-04	1.10556e-04	1.08359e-04	4.66996e-05	4.60476e-05	4.78958e-05	31.32 sec
[7, 7, 7, 7]	2.52721e-05	2.57379e-05	2.48462e-05	5.26725e-05	5.57748e-05	5.23621e-05	7.98072e-06	8.51448e-06	8.34705e-06	46.44 sec
[10, 10, 10, 10]	1.03335e-05	9.91660e-06	1.05554e-05	3.60688e-05	3.69765e-05	3.83447e-05	5.19075e-05	5.90247e-05	5.27263e-05	36.23 sec
[15, 15, 15, 15]	1.97327e-05	2.17426e-05	2.35077e-05	5.84404e-05	6.57859e-05	6.09724e-05	3.10692e-05	3.59397e-05	3.36809e-05	78.24 sec

The analytical one-soliton bright solution $u^{(1)} = |q[1]|$ and its LM–ANN reconstruction are shown in Fig. 1. Fig. 1a–c present the exact, predicted, and error surfaces, while Fig. 1d–f depict the contour plots. The LM–ANN matched the analytical waveform with residuals of the order of 10^{-5}. Table 1 lists the relative L_2 error $E^{(1)}_{L_2}$ for training, validation, and testing across different hidden layers and activation functions, with `tansig` and `radbas` yielding the lowest errors. Fig. 7 shows the error evolution under the Monte Carlo noise tests (1%–3% input, output, and combined), and the summary in Table 7 confirms the stable performance of the soliton approximation.

The dark soliton $u^{(4)} = \mathrm{Re}(v[1])$ and its LM–ANN approximation are presented in Fig. 2, with Fig. 2a–c displaying the exact, predicted, and error surfaces, and Fig. 2d–f showing the contours. The LM–ANN reproduced the notch profile with small residuals. Table 2 reports $E^{(4)}_{L_2}$ under various hidden-layer settings, where `tansig` and `radbas` again minimized errors. Monte Carlo robustness results in Fig. 8 and statistics in Table 7 confirm reliable approximation of the solution.

The $q[2]$ bright soliton solution $u^{(2)} = |q[2]|$ is compared in Fig. 3, where Fig. 3a–b show the exact and predicted solutions, Fig. 3c presents the error surface, and Fig. 3d–f provide contour maps. The LM–ANN closely reproduces the soliton profile. Table 3 summarizes $E^{(2)}_{L_2}$ for different hidden-layer depths and activations, with deeper networks improving the accuracy at longer training times. Monte Carlo error trends in Fig. 9 demonstrate robustness under noisy input, output, and combined conditions, while Table 7 lists the corresponding statistical summary.

The dark soliton solution $u^{(5)} = \mathrm{Re}(v[2])$ is shown in Fig. 4, with Fig. 4a–c depicting the exact, LM–ANN, and error surfaces, and Fig. 4d–f containing contour maps. The LM–ANN effectively reproduced the solution with minimal residuals. Table 4 gives $E^{(5)}_{L_2}$ for various network structures, showing the lowest errors for `tansig` and `radbas`. Monte Carlo experiments in Fig. 10 and results in Table 7 confirm stability under noisy conditions.

Table 2
$E^{(4)}_{L_2}$ for $u^{(4)}$ and training time for different hidden layer configurations and activation functions.

Hidden Layers	tansig			logsig			radbas			Time (sec)
	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	
[3, 3, 3]	3.46620e-05	3.28925e-05	3.28221e-05	3.06262e-05	2.97295e-05	3.13892e-05	5.07849e-05	5.05468e-05	5.05005e-05	3.01 sec
[7, 7, 7]	5.39389e-05	5.58720e-05	5.95859e-05	3.08554e-05	3.20582e-05	2.77400e-05	5.35563e-05	5.54516e-05	5.25900e-05	6.34 sec
[10, 10, 10]	3.44316e-05	3.39814e-05	3.75157e-05	3.36015e-05	3.43845e-05	3.43882e-05	8.46951e-06	9.79784e-06	1.17431e-05	19.11 sec
[15, 15, 15]	1.19851e-05	1.24727e-05	1.20477e-05	5.63044e-05	6.32445e-05	5.93011e-05	1.94249e-05	2.22280e-05	2.53722e-05	52.60 sec
[3, 3, 3, 3]	5.64408e-05	5.74480e-05	5.46878e-05	6.05997e-05	5.81995e-05	5.82062e-05	2.87587e-01	3.04350e-01	2.95276e-01	9.83 sec
[7, 7, 7, 7]	2.46705e-05	2.41904e-05	2.50999e-05	6.55610e-05	6.70326e-05	6.66015e-05	9.34639e-06	1.19000e-05	9.50072e-06	14.30 sec
[10, 10, 10, 10]	5.06545e-05	5.18217e-05	5.42718e-05	2.23353e-05	2.32753e-05	2.40358e-05	1.18593e-05	1.17746e-05	1.17299e-05	48.32 sec
[15, 15, 15, 15]	6.08851e-05	6.86133e-05	7.01218e-05	6.68177e-05	7.40265e-05	7.00609e-05	2.14450e-05	2.34406e-05	2.40430e-05	78.17 sec

Table 3
$E^{(2)}_{L_2}$ for $u^{(2)}$ and training time for different hidden layer configurations and activation functions.

Hidden Layers	tansig			logsig			radbas			Time (sec)
	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	
[3, 3, 3]	5.07664e-02	5.47586e-02	5.47672e-02	6.13118e-02	5.95780e-02	6.18801e-02	5.81240e-02	5.80325e-02	5.58887e-02	5.09 sec
[7, 7, 7]	4.34568e-03	3.78198e-03	3.83920e-03	4.06126e-03	4.22793e-03	4.24343e-03	4.88393e-03	4.81458e-03	4.78103e-03	18.64 sec
[10, 10, 10]	1.52017e-03	1.72130e-03	2.61304e-03	2.01257e-03	1.95370e-03	1.87686e-03	2.60249e-03	2.84823e-03	2.93521e-03	38.67 sec
[15, 15, 15]	7.27033e-04	9.87043e-04	9.26144e-04	4.74051e-04	9.10897e-04	5.64578e-04	9.96099e-04	8.72016e-04	1.45434e-03	103.38 sec
[3, 3, 3, 3]	1.88074e-01	1.91152e-01	1.88874e-01	4.48539e-02	4.17308e-02	4.38763e-02	1.19673e-01	1.22593e-01	1.28132e-01	11.72 sec
[7, 7, 7, 7]	3.16291e-03	3.31565e-03	3.46077e-03	2.32828e-03	2.65611e-03	3.40861e-03	3.68377e-03	3.90179e-03	3.41856e-03	29.95 sec
[10, 10, 10, 10]	9.81257e-04	9.36908e-04	8.61271e-04	1.00680e-03	1.03003e-03	1.26217e-03	1.25695e-03	2.05181e-03	1.65328e-03	61.52 sec
[15, 15, 15, 15]	2.45336e-04	7.98583e-04	5.34452e-04	5.95490e-04	6.47076e-04	7.27573e-04	2.80998e-04	7.12193e-04	3.55571e-04	204.09 sec

Table 4
$E^{(5)}_{L_2}$ for $u^{(5)}$ and training time for different hidden layer configurations and activation functions.

Hidden Layers	tansig			logsig			radbas			Time (sec)
	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	
[3, 3, 3]	2.71756e-02	2.79706e-02	2.89163e-02	3.35237e-02	3.59781e-02	3.13541e-02	5.70787e-02	6.14370e-02	5.96670e-02	10.17 sec
[7, 7, 7]	8.18112e-04	8.50122e-04	8.86424e-04	2.17856e-03	2.32801e-03	2.23395e-03	3.88960e-03	3.85398e-03	3.86313e-03	54.03 sec
[10, 10, 10]	9.86757e-05	1.04146e-04	1.13874e-04	1.26853e-04	1.50406e-04	1.38808e-04	1.10883e-03	1.28326e-03	1.26299e-03	102.19 sec
[15, 15, 15]	3.59384e-05	4.01117e-05	4.10486e-05	5.35922e-05	6.49763e-05	6.17463e-05	1.27911e-04	1.52463e-04	1.56743e-04	239.09 sec
[3, 3, 3, 3]	2.85566e-02	2.99423e-02	3.05990e-02	2.22374e-02	2.24594e-02	2.13695e-02	2.11456e-01	2.05717e-01	2.18892e-01	40.24 sec
[7, 7, 7, 7]	4.47272e-04	4.58799e-04	4.98661e-04	4.00413e-04	4.34241e-04	4.64314e-04	7.13594e-04	8.33897e-04	7.71864e-04	93.56 sec
[10, 10, 10, 10]	6.20117e-05	6.99971e-05	6.44560e-05	1.24251e-04	1.51080e-04	1.28469e-04	1.25388e-04	1.46576e-04	1.39614e-04	172.33 sec
[15, 15, 15, 15]	1.53626e-05	1.81082e-05	1.76133e-05	2.52914e-05	2.64882e-05	2.83868e-05	3.34689e-05	4.34753e-05	4.19427e-05	386.44 sec

Table 5
$E^{(3)}_{L_2}$ for $u^{(3)}$ and training time for different hidden layer configurations and activation functions.

Hidden Layers	tansig			logsig			radbas			Time (sec)
	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	
[3, 3, 3]	4.42555e-01	4.32008e-01	4.32885e-01	4.14738e-01	4.39698e-01	4.13555e-01	4.14454e-01	4.09366e-01	4.33670e-01	11.92
[7, 7, 7]	9.28175e-02	9.80416e-02	9.41919e-02	2.78852e-01	2.89949e-01	2.78528e-01	7.91861e-02	8.29332e-02	8.18069e-02	66.80
[10, 10, 10]	5.48165e-02	6.38754e-02	6.39983e-02	4.77462e-02	4.81954e-02	4.99663e-02	5.70970e-02	6.20065e-02	6.06068e-02	114.14
[15, 15, 15]	1.69154e-02	1.98814e-02	1.95244e-02	1.52427e-02	1.81328e-02	2.15748e-02	1.87419e-02	2.46663e-02	2.34377e-02	249.14
[3, 3, 3, 3]	4.67386e-01	5.02498e-01	4.56063e-01	4.66961e-01	4.61426e-01	4.42097e-01	3.91928e-01	3.98029e-01	3.86217e-01	41.31
[7, 7, 7, 7]	7.02304e-02	7.06793e-02	7.17031e-02	6.25293e-02	7.17362e-02	6.69860e-02	6.68231e-02	7.14726e-02	7.35445e-02	77.04
[10, 10, 10, 10]	1.97646e-02	2.25492e-02	2.50925e-02	2.67599e-02	3.25899e-02	3.51142e-02	3.50422e-02	3.69128e-02	4.34822e-02	164.19
[15, 15, 15, 15]	8.17066e-03	1.21519e-02	1.68935e-02	6.53293e-03	8.88197e-03	9.57826e-03	1.03244e-02	1.59308e-02	1.58635e-02	386.66

Table 6
$E^{(6)}_{L_2}$ for $u^{(6)}$ and training time for different hidden layer configurations and activation functions.

Hidden Layers	tansig			logsig			radbas			Time (sec)
	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	Train RE	Val RE	Test RE	
[3, 3, 3]	2.11828e-01	2.13536e-01	2.14106e-01	2.14582e-01	2.25669e-01	2.05318e-01	2.32421e-01	2.32646e-01	2.23875e-01	5.02 sec
[7, 7, 7]	3.06744e-02	3.27275e-02	3.40280e-02	1.39869e-01	1.46605e-01	1.42376e-01	5.67417e-02	5.79935e-02	5.87653e-02	20.77 sec
[10, 10, 10]	4.70403e-03	5.21158e-03	5.24204e-03	5.99470e-03	6.74133e-03	6.59746e-03	1.72782e-02	1.87525e-02	1.85533e-02	36.31 sec
[15, 15, 15]	1.01131e-03	1.20882e-03	1.20048e-03	1.35775e-03	1.67381e-03	1.84543e-03	3.53696e-03	3.80385e-03	3.95747e-03	103.03 sec
[3, 3, 3, 3]	2.04275e-01	2.04508e-01	2.08278e-01	1.94239e-01	2.09881e-01	2.08650e-01	2.54780e-01	2.49237e-01	2.66004e-01	14.86 sec
[7, 7, 7, 7]	5.62424e-03	6.45496e-03	6.44865e-03	1.19345e-02	1.24941e-02	1.28703e-02	1.28317e-02	1.41461e-02	1.37648e-02	35.12 sec
[10, 10, 10, 10]	2.06649e-03	2.50137e-03	2.75770e-03	1.83989e-03	2.02825e-03	2.09176e-03	6.43907e-04	7.54606e-04	9.34765e-04	62.37 sec
[15, 15, 15, 15]	1.74309e-04	2.18425e-04	2.07749e-04	1.65360e-04	2.98363e-04	2.30593e-04	3.13508e-04	4.13462e-04	3.98367e-04	195.10 sec

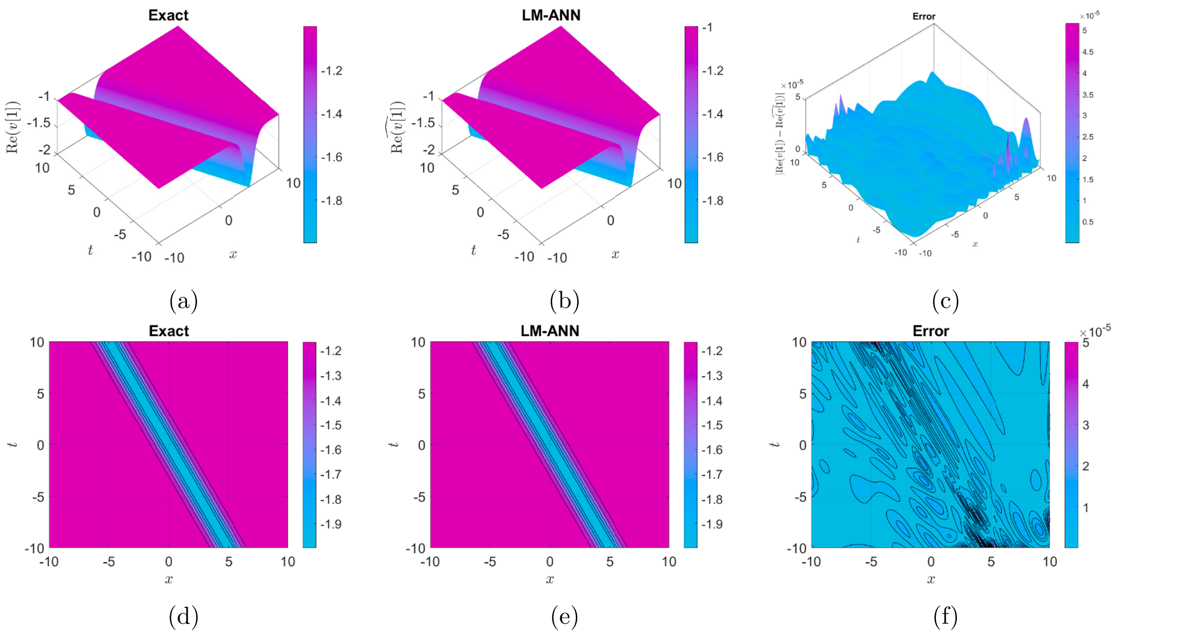

Fig. 2. The real plot of $v[1]$. The figure consists of six subplots: (2a) exact solution $\left(\mathrm{Re}(v[1])\right)$, (2b) LM-ANN solution $\left(\widehat{\mathrm{Re}(v[1])}\right)$, (2c) error plot for 3D, (2d) contour plot of the exact solution $\left(\mathrm{Re}(v[1])\right)$, (2e) contour plot of the LM-ANN solution $\left(\widehat{\mathrm{Re}(v[1])}\right)$, and (2f) contour plot of the error.

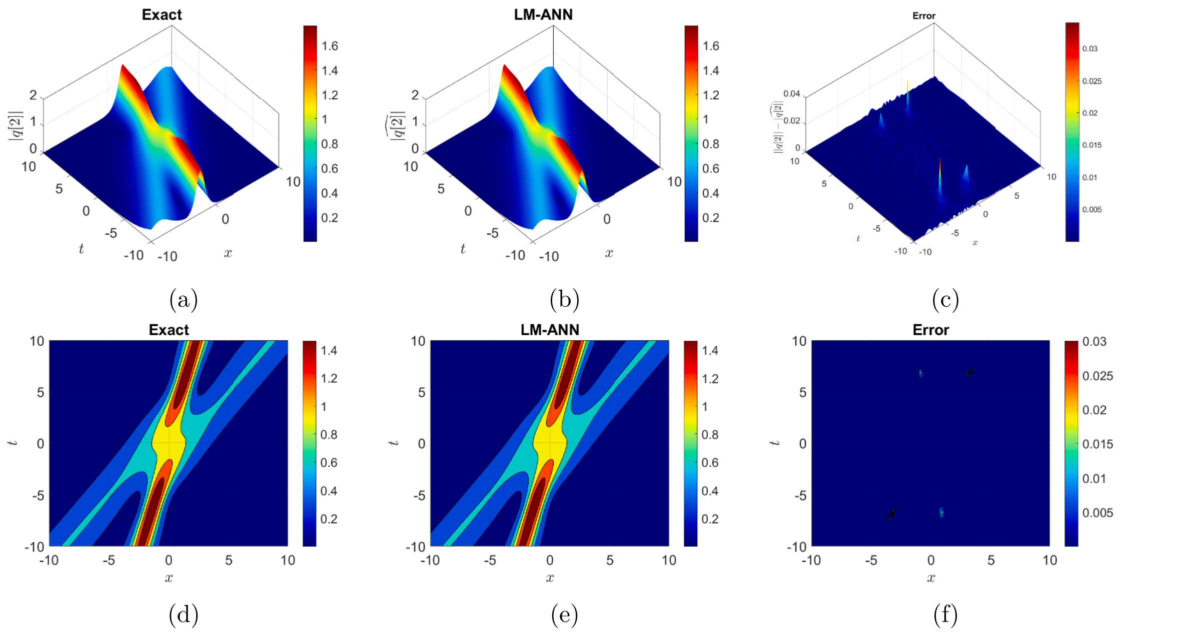

Fig. 3. The absolute plot of $q[2]$ soluion, with parameters $c = 1, \zeta_1 = 0.3i, \zeta_2 = -0.8i, \alpha_1 = -\alpha_2 = 1,\ c = 1$. The figure consists of six subplots: (3a) exact solution $\left(|q[2]|\right)$, (3b) LM–ANN solution $\left(|\widehat{q[2]}|\right)$, (3c) error plot for 3D, (3d) contour plot of the exact solution $\left(|q[2]|\right)$, (3e) contour plot of the LM–ANN solution $\left(|\widehat{q[2]}|\right)$, and (3f) contour plot of the error.

The breather solution $u^{(3)} = |q[2]|$ was obtained analytically by the DT and approximated with the LM–ANN, as shown in Fig. 5. Fig. 5a–d present the exact surfaces, whereas Fig. 5b–e show the LM–ANN results, and Fig. 5c–f display the error plots. Table 5 reports $E^{(3)}_{L_2}$ across hidden-layer and activation choices, where deeper networks improve accuracy. The robustness is illustrated in Fig. 11, which shows the error distributions under 25 noisy trials, with statistical results in Table 7, confirming stable predictions for breather dynamics.

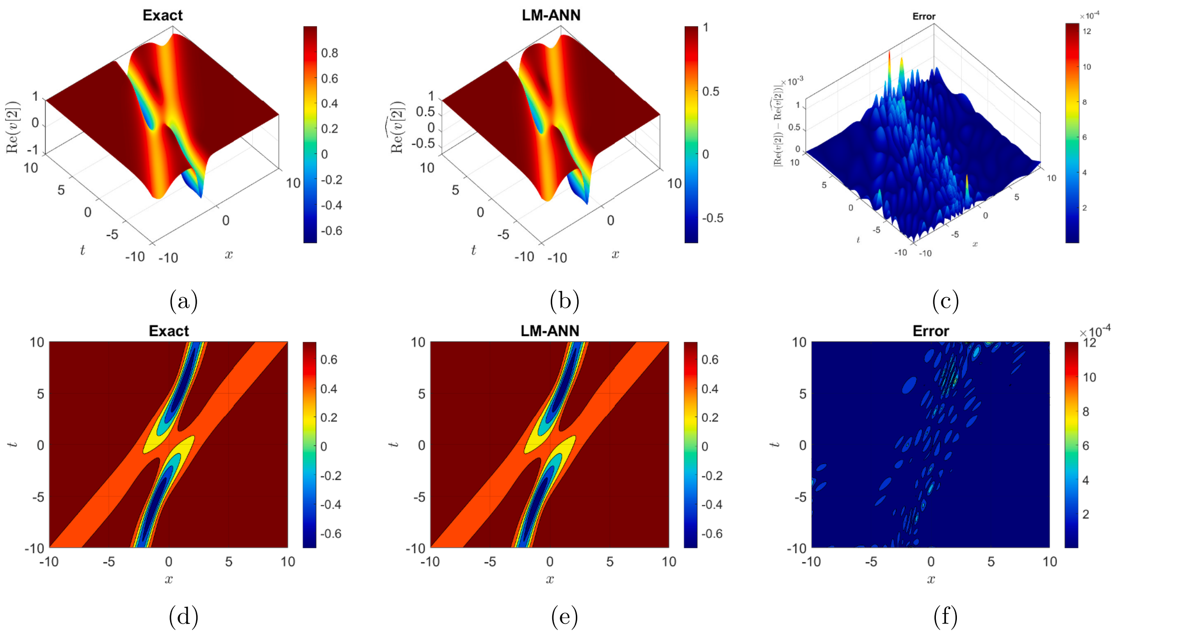

Fig. 4. Plot of the $\mathrm{Re}(v[2])$ soluion, is presented using the same parameters as in Fig. 3. The figure consists of six subplots: (4a) exact solution $\mathrm{Re}(v[2])$, (4b) LM–ANN solution $\left(\widehat{\mathrm{Re}(v[2])}\right)$, (4c) error plot for 3D, (4d) contour plot of the exact solution $\mathrm{Re}(v[2])$, (4e) contour plot of the LM-ANN solution $\left(\widehat{\mathrm{Re}(v[2])}\right)$, and (4f) contour plot of the error.

Table 7
$E_{L_2}^{(m)}$ (mean ± standard deviation) computed over 25 Monte Carlo trials for $u^{(1)}$, $u^{(2)}$, $u^{(3)}$, $u^{(4)}$, $u^{(5)}$ and $u^{(6)}$.

Solution Type	Noise (%)	Train (Clean)	Test (Noisy)	Train (Noisy)	Test (Clean)	Train (Noisy)	Test (Noisy)
	1	0.0571 ± 0.0261	0.1020 ± 0.0739	0.0045 ± 0.0010	0.0045 ± 0.0010	0.0527 ± 0.0209	0.1284 ± 0.0941
$\widehat{\lvert q[1]\rvert}$	2	0.1258 ± 0.0350	0.2174 ± 0.2568	0.0101 ± 0.0015	0.0101 ± 0.0015	0.1601 ± 0.0977	0.2632 ± 0.1893
	3	0.3171 ± 0.2938	0.3851 ± 0.2872	0.0224 ± 0.0294	0.0224 ± 0.0294	0.3111 ± 0.3068	0.3443 ± 0.1249
	1	0.0333 ± 0.0157	0.0602 ± 0.0299	0.0011 ± 0.0001	0.0011 ± 0.0001	0.0292 ± 0.0097	0.0611 ± 0.0233
$\widehat{Re(v[1])}$	2	0.1056 ± 0.0726	0.1830 ± 0.1491	0.0030 ± 0.0019	0.0030 ± 0.0019	0.0880 ± 0.0334	0.1141 ± 0.0623
	3	0.1823 ± 0.1122	0.2218 ± 0.1534	0.0039 ± 0.0009	0.0039 ± 0.0009	0.1556 ± 0.0943	0.1974 ± 0.1076
	1	0.1002 ± 0.1293	0.3352 ± 0.7558	0.0046 ± 0.0005	0.0046 ± 0.0005	0.0695 ± 0.0348	0.1751 ± 0.1173
$\widehat{\lvert q[2]\rvert}$	2	0.2352 ± 0.1012	0.3453 ± 0.1852	0.0084 ± 0.0005	0.0084 ± 0.0005	0.2514 ± 0.2158	0.3479 ± 0.2328
	3	0.3700 ± 0.1712	0.4554 ± 0.2305	0.0127 ± 0.0009	0.0127 ± 0.0009	0.3413 ± 0.1703	0.4979 ± 0.3036
	1	0.0720 ± 0.0824	0.1230 ± 0.1263	0.0017 ± 0.0002	0.0017 ± 0.0002	0.0542 ± 0.0246	0.1044 ± 0.0451
$\widehat{Re(v[2])}$	2	0.1765 ± 0.0917	0.2460 ± 0.1461	0.0036 ± 0.0006	0.0036 ± 0.0006	0.1847 ± 0.1212	0.2640 ± 0.1931
	3	0.3687 ± 0.3499	0.4378 ± 0.3463	0.0067 ± 0.0048	0.0067 ± 0.0048	0.3053 ± 0.2070	0.3518 ± 0.1950
	1	0.0731 ± 0.0094	0.1208 ± 0.0227	0.0609 ± 0.0090	0.0609 ± 0.0090	0.0790 ± 0.0140	0.1908 ± 0.1172
$\widehat{\lvert q[2]\rvert}$	2	0.1802 ± 0.0930	0.3012 ± 0.1909	0.0839 ± 0.1258	0.0839 ± 0.1258	0.1501 ± 0.0391	0.3094 ± 0.1201
	3	0.2964 ± 0.1203	0.4219 ± 0.2126	0.0629 ± 0.0075	0.0629 ± 0.0075	0.2706 ± 0.0875	0.4332 ± 0.1126
	1	0.0452 ± 0.0511	0.0808 ± 0.1048	0.0205 ± 0.0268	0.0205 ± 0.0268	0.0328 ± 0.0165	0.0690 ± 0.0345
$\widehat{Re(v[2])}$	2	0.0979 ± 0.0400	0.1638 ± 0.0971	0.0121 ± 0.0048	0.0121 ± 0.0048	0.1179 ± 0.0679	0.2255 ± 0.1567
	3	0.3079 ± 0.4254	0.3059 ± 0.2908	0.0202 ± 0.0265	0.0202 ± 0.0265	0.1957 ± 0.0834	0.2701 ± 0.1093

The breather solution $u^{(6)} = \mathrm{Re}(v[2])$ is also investigated. Fig. 6 compares the exact results (Fig. 6a, d), LM–ANN results (Fig. 6b, e), and error plots (Fig. 6c, f). Table 6 shows $E_{L_2}^{(6)}$ across hidden-layer depths and activation functions, confirming the improved accuracy with deeper networks at increased computational times. Robustness was assessed through Monte Carlo trials, as shown in Fig. 12, with the statistics in Table 7, supporting the reliability of LM–ANN for the breather solution.

Overall, across all six solution types, the LM–ANN framework provided accurate approximations with low relative L_2 errors, consistent performance across activation functions, and reliable robustness under noisy conditions. Tables 1–7 collectively confirm that deeper architectures improve accuracy at the expense of computation time, while the Monte Carlo trials demonstrate stable predictive ability across various solution dynamics.

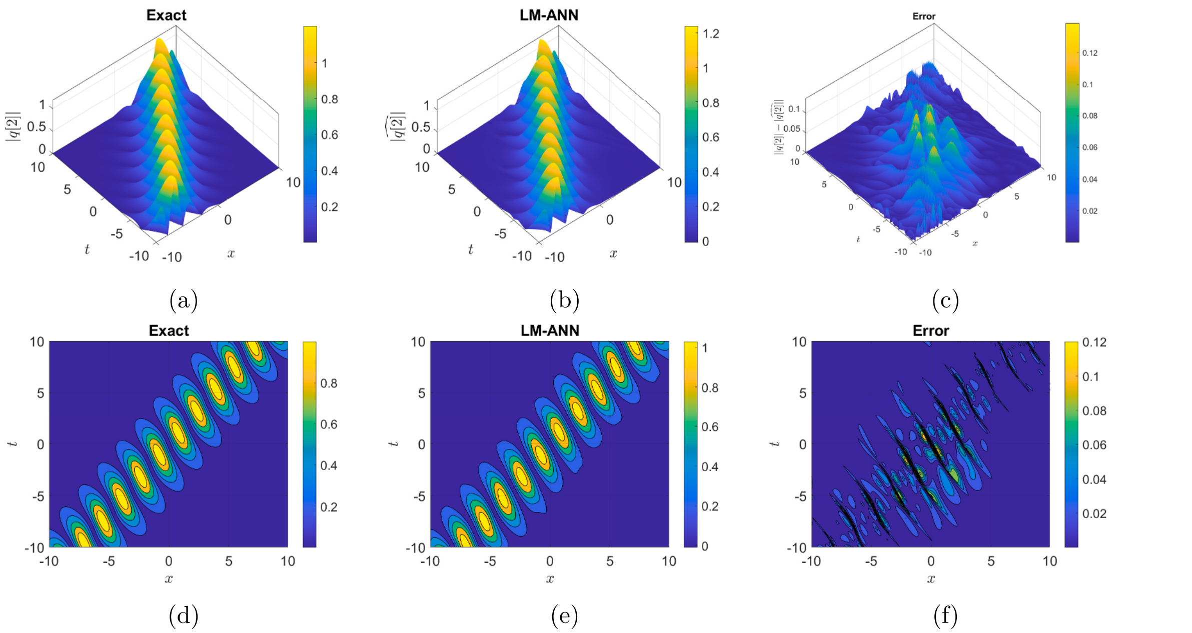

Fig. 5. The absolute plot of the $q[2]$, with parameters $c = 1,\ \zeta_1 = 1 + 0.3i,\ \zeta_2 = 1 - 0.3i,\ \alpha_1 = -\alpha_2 = 1$. The figure consists of six subplots: (5a) exact solution $(|q[2]|)$, (5b) LM–ANN solution $(\widehat{|q[2]|})$, (5c) error plot for 3D, (5d) contour plot of the exact solution $(|q[2]|)$, (5e) contour plot of the LM–ANN solution $(\widehat{|q[2]|})$, and (5f) contour plot of the error.

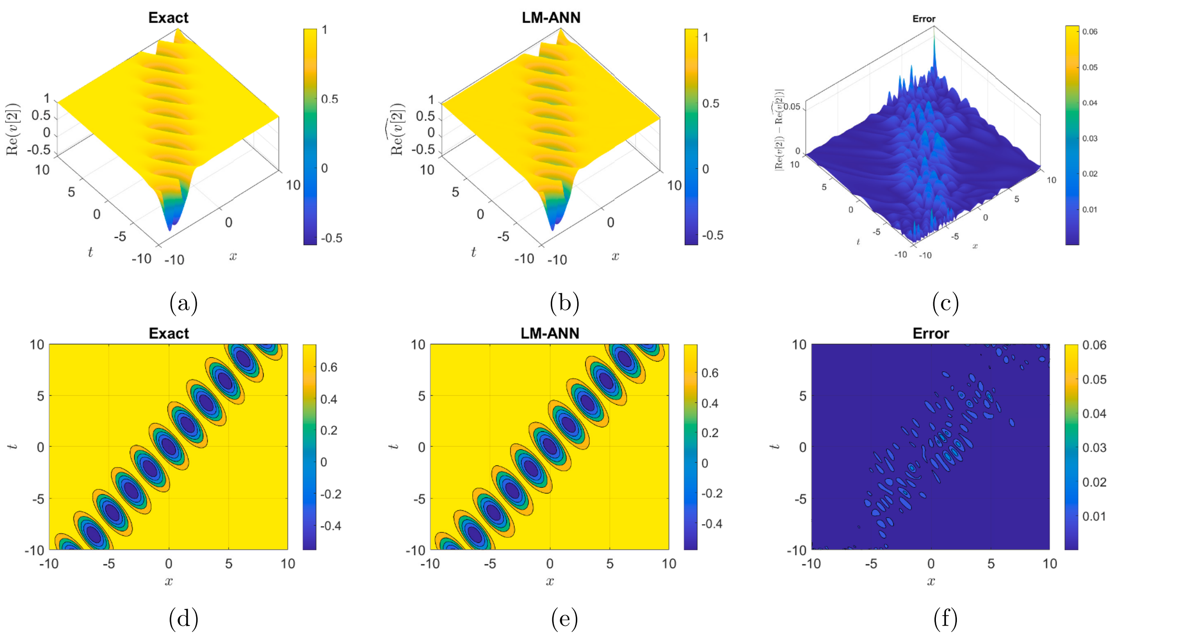

Fig. 6. Plot of the $v[2]$ soluion, using the same parameters as in Fig. 5. The figure consists of six subplots: (6a) exact solution, (6b) LM-ANN solution, (6c) error plot, (6d) contour plot of the exact solution, (6e) contour plot of the LM-ANN solution, and (6f) contour plot of the error.

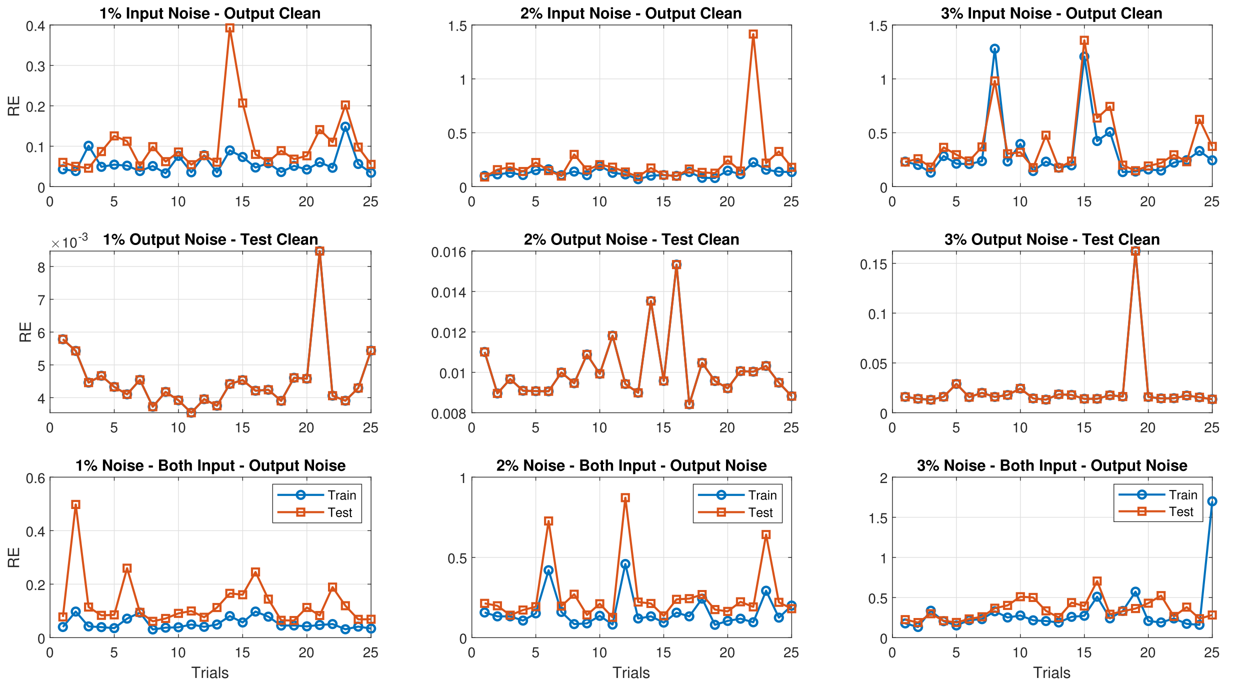

Fig. 7. $E_{L_2}^{(1)}$ across 25 Monte Carlo trials under different noise conditions for $\widehat{q[1]}$. Fig. 1 shows the corresponding analytical and predicted solution plots.

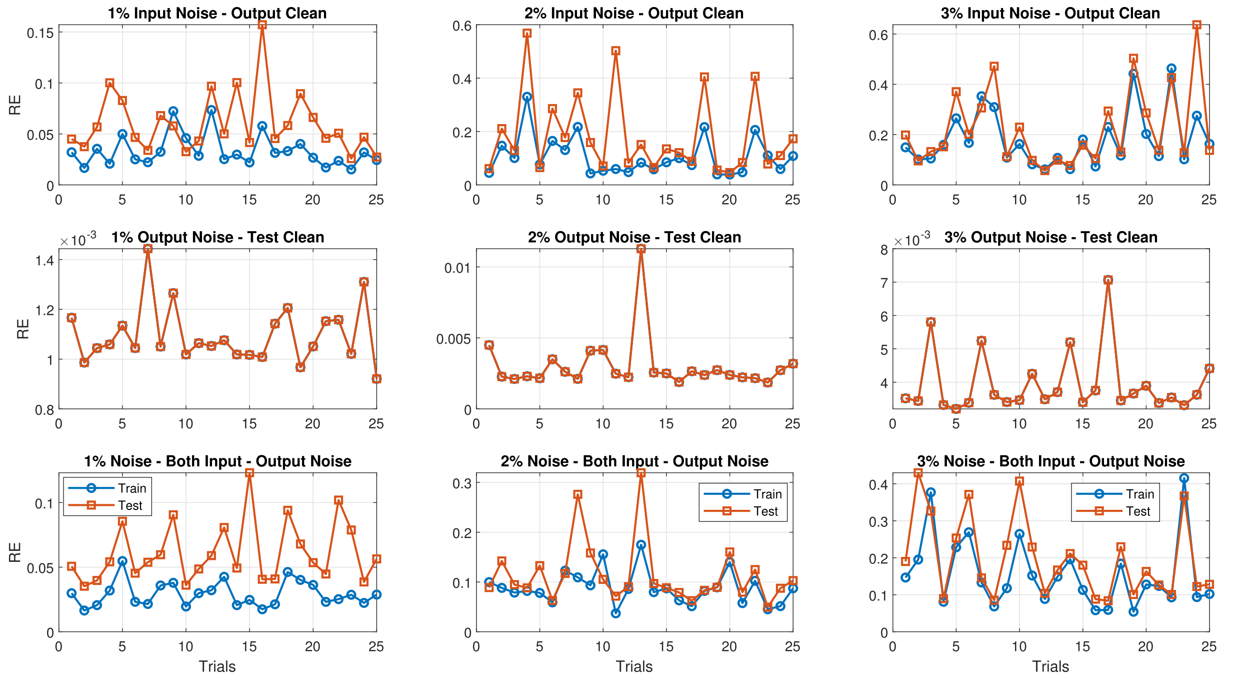

Fig. 8. $E_{L_2}^{(4)}$ across 25 Monte Carlo trials under different noise conditions for the $\widehat{\mathrm{Re}(v[1])}$. Refer to Fig. 2 for the corresponding exact and predicted solution plots.

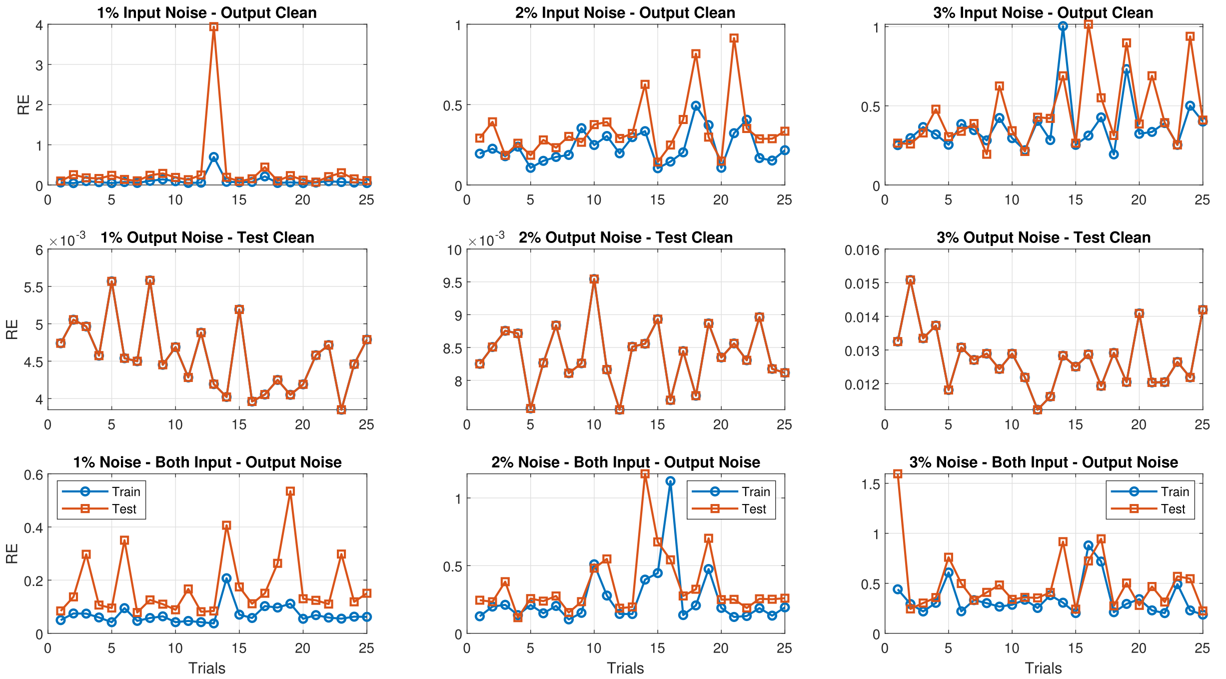

Fig. 9. $E^{(2)}_{L_2}$ across 25 Monte Carlo trials under different noise conditions for the q-2 $\widehat{|q[2]|}$. Refer to Fig. 3 for the corresponding exact and predicted solution plots.

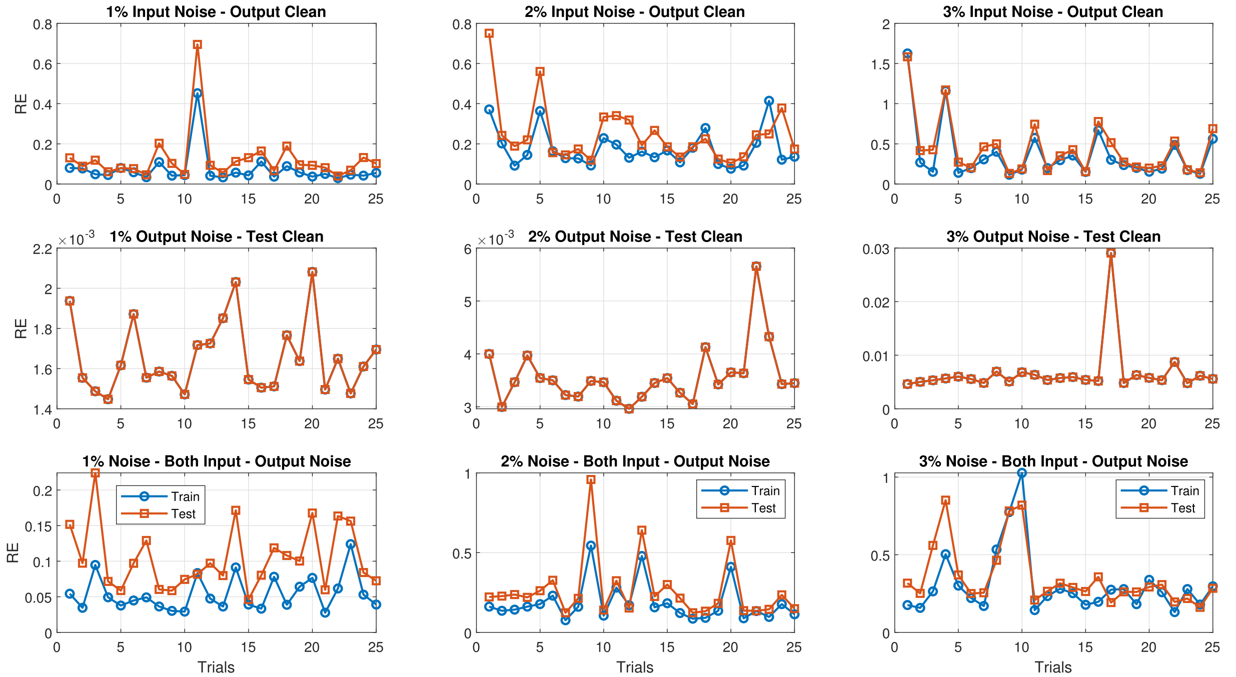

Fig. 10. $E^{(5)}_{L_2}$ across 25 Monte Carlo trials under different noise conditions for the v-2 $\widehat{\mathrm{Re}(v[2])}$. Refer to Fig. 4 for the corresponding exact and predicted solution plots.

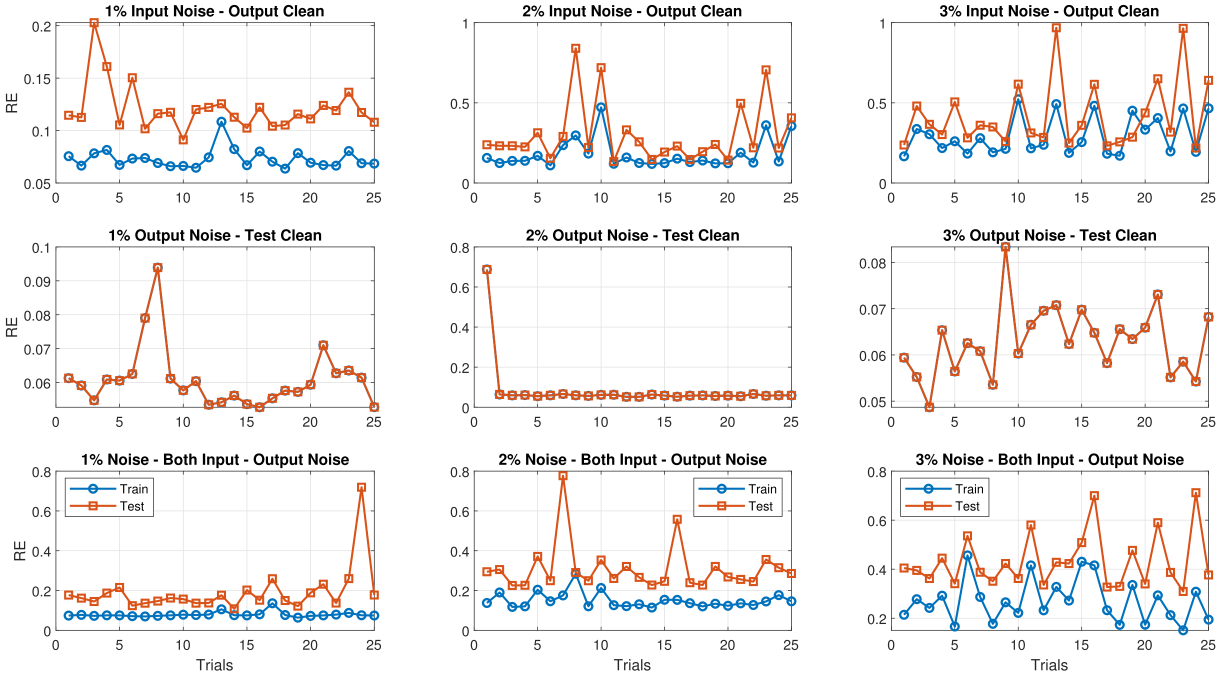

Fig. 11. $E_{L_2}^{(3)}$ across 25 Monte Carlo trials under different noise conditions for the $|\widehat{q[2]}|$. Refer to Fig. 5 for the corresponding exact and predicted solution plots.

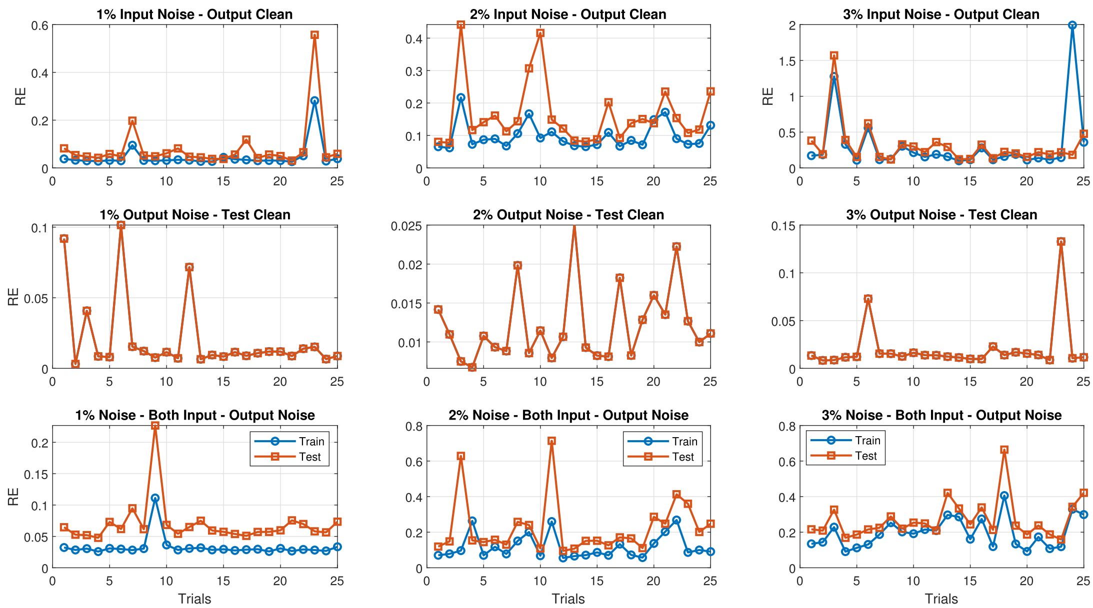

Fig. 12. $E_{L_2}^{(6)}$ across 25 Monte Carlo trials under different noise conditions for the v-2 $\widehat{\mathrm{Re}(v[2])}$. Refer to Fig. 6 for the corresponding exact and predicted solution plots.

4. Concluding remarks

In this study, we investigated the Kuralay-II system (1.9) using the DT, motivated by its applications in nonlinear optics, ferromagnetic materials, and optical fibers. We consider a matrix generalization of the K-IIE and via DT, and derive new classes of analytical solutions relevant to plasma dynamics and fluid mechanics. Using this approach, we generated dark and bright one- and two-solitary waves along with breather solutions and analyzed their dynamical profiles in detail.

To complement the analytical results, we employed an LM–ANN to approximate the DT-based solutions. The network accurately reconstructed the analytical wave profiles and time-periodic patterns with low relative L_2 errors. In addition, Monte Carlo experiments with noise in the inputs and outputs confirmed the robustness of the network, indicating that the approximations remained reliable under perturbations. These findings establish LM–ANN as a practical computational tool for handling complex nonlinear wave solutions when analytical evaluation alone is insufficient.

Our findings enhances the understanding of the model's integrability and clarify the physical importance of the computed bright and dark as well as breather wave solutions. These solutions demonstrate oscillatory excitations and localized energy pulses that can be used to study phenomena such as pulse propagation in optical fibers, temporal oscillations of magnetization in ferromagnetic materials, and intensity dips in nonlinear optical media. Thus, this study strengthens the link between soliton theory and its applications across various nonlinear wave systems.

In summary, the present LM–ANN results should be interpreted in light of standard approximation trade-offs: (i) the achievable accuracy depends on the network capacity (depth/width) relative to the complexity of the target solution, and wavefields with sharper gradients, stronger localization, or higher-frequency oscillations may require increased capacity and/or refined sampling; (ii) LM training can become computationally demanding for very large networks or substantially finer training grids due to Jacobian-based updates; and (iii) fitting difficulty may increase in extreme parameter regimes that generate highly stiff or rapidly varying dynamics, where insufficient capacity, suboptimal initialization, or inadequate sampling density can slow convergence or reduce generalization performance.

In future work, we will extend this study by applying physics-informed neural networks (PINNs). Although the present LM–ANN is effective for direct data-driven learning, PINNs embed the governing equations into training and are well-suited for inverse tasks such as parameter estimation and reconstruction from limited data. We anticipate that capturing time-periodic patterns within the PINN framework will be challenging and plan to address this, along with inverse formulations for system (1.9), to broaden our understanding of its nonlinear dynamics.

CRediT authorship contribution statement

Zeeshan Amjad: Writing – original draft, Methodology, Conceptualization; **Aamir Farooq:** Visualization, Software, Methodology; **H.W.A. Riaz:** Writing – original draft, Investigation, Data curation; **Wen-Xiu Ma:** Writing – review & editing, Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by NSFC under grants 12271488, 11975145, and 11972291 and the Ministry of Science and Technology of China under grants G2021016032L and G2023016011L.

Data availability

Data sharing does not apply to this article, as no datasets were generated or analyzed during the current study.

References

[1] E. Bruè, C. De Lellis, Anomalous dissipation for the forced 3D Navier–Stokes equations, Commun. Math. Phys. 400 (3) (2023) 1507–1533.
[2] T. Lange, Regularization by noise of an averaged version of the Navier-Stokes equations, J. Dyn. Differ. Equ. 36 (3) (2024) 3011–3036.
[3] J. Skipp, J. Laurie, S. Nazarenko, Hamiltonian derivation of the point vortex model from the two-dimensional nonlinear Schrödinger equation, Phys. Rev. E 107 (2) (2023) 025107.
[4] S.T.R. Rizvi, A.R. Seadawy, S. Ahmed, A. Bashir, Optical soliton solutions and various breathers lump interaction solutions with periodic wave for nonlinear Schrödinger equation with quadratic nonlinear susceptibility, Opt. Quantum Electron. 55 (3) (2023) 286.
[5] S.Z. Majid, W.A. Faridi, M.I. Asjad, M. Abd El-Rahman, S.M. Eldin, Explicit soliton structure formation for the Riemann wave equation and a sensitive demonstration, Fractal Fract. 7 (2) (2023) 102.
[6] J.-W. Yang, Y.-T. Gao, C.-Q. Su, D.-W. Zuo, Y.-J. Feng, Solitons and quasi-periodic behaviors in an inhomogeneous optical fiber, Commun. Nonlinear Sci. Numer. Simul. 42 (2017) 477–490.
[7] H. Triki, A. Biswas, S.P. Moshokoa, M. Belic, Optical solitons and conservation laws with quadratic-cubic nonlinearity, Optik 128 (2017) 63–70.
[8] J.-V. Kim, Role of nonlinear anisotropic damping in the magnetization dynamics of topological solitons, Phys. Rev. B 92 (1) (2015) 014418.
[9] H.T. Tchokouansi, V.K. Kuetche, T.C. Kofane, On the propagation of solitons in ferrites: the inverse scattering approach, Chaos Solit. Fractals 86 (2016) 64–74.
[10] M. Younis, S.T.R. Rizvi, Optical soliton like-pulses in ring-cavity fiber lasers of carbon nanotubes, J. Nanoelectron. Optoelectron. 11 (3) (2016) 276–279.

[11] Y.-L. Ma, B.-Q. Li, The wrinkle-like N-solitons for the thermophoretic motion equation through graphene sheets, Phys. A Stat. Mech. Appl. 494 (2018) 169–174.
[12] K. Yesmakhanova, G. Nugmanova, G. Shaikhova, G. Bekova, R. Myrzakulov, Coupled dispersionless and generalized Heisenberg ferromagnet equations with self-consistent sources: geometry and equivalence, Int. J. Geom. Methods Mod. Phys. 17 (07) (2020) 2050104.
[13] Z. Sagidullayeva, G. Nugmanova, R. Myrzakulov, N. Serikbayev, Integrable Kuralay equations: geometry, solutions and generalizations, Symmetry 14 (7) (2022) 1374.
[14] W.A. Faridi, Z. Myrzakulova, R. Myrzakulov, A. Akgül, M.S. Osman, The construction of exact solution and explicit propagating optical soliton waves of Kuralay equation by the new extended direct algebraic and Nucci's reduction techniques, Int. J. Model. Simul. 45 (6) (2024) 1–20.
[15] A. Hussain, T.F. Ibrahim, F.M.O. Birkea, B.R. Al-Sinan, Dynamical behavior of analytical soliton solutions to the Kuralay equations via symbolic computation, Nonlinear Dyn. 112 (22) (2024) 20231–20254.
[16] A. Zafar, M. Raheel, M.R. Ali, Z. Myrzakulova, A. Bekir, R. Myrzakulov, Exact solutions of M-fractional Kuralay equation via three analytical schemes, Symmetry 15 (10) (2023) 1862.
[17] J. Liu, Z. Li, Bifurcation analysis and soliton solutions to the Kuralay equation via dynamic system analysis method and complete discrimination system method, Qual. Theory Dyn. Syst. 23 (3) (2024) 126.
[18] A. Farooq, W.X. Ma, M.I. Khan, Exploring exact solitary wave solutions of Kuralay-II equation based on the truncated M-fractional derivative using the Jacobi Elliptic function expansion method, Opt. Quantum Electron. 56 (7) (2024) 1105.
[19] S.H. Alfalqi, M.M.A. Khater, Numerical solutions and analytical methods for the Kuralay equation: a path to understanding integrable systems, Opt. Quantum Electron. 56 (5) (2024) 756.
[20] W.A. Faridi, M.A. Bakar, Z. Myrzakulova, R. Myrzakulov, A. Akgül, S.M. El Din, The formation of solitary wave solutions and their propagation for Kuralay equation, Results Phys. 52 (2023) 106774.
[21] T. Mathanaranjan, Optical soliton, linear stability analysis and conservation laws via multipliers to the integrable Kuralay equation, Optik 290 (2023) 171266.
[22] Y. Wang, X. Lü, W.-X. Ma, Integrability characteristics and exact solutions of an extended variable-coefficient shallow water wave model, Nonlinear Dyn. 113 (2025) 21725–21741. https://doi.org/10.1007/s11071-025-XXXXX
[23] Y. Wang, X. Lü, Exact solutions and Bäcklund transformation for a generalized variable-coefficient Fokas-typed equation, Commun. Nonlinear Sci. Numer. Simul. 108567 (2025). https://doi.org/10.1016/j.cnsns.2024.108567
[24] Y. Zhang, X. Lü, Data-driven solutions and parameter discovery of the extended higher-order nonlinear Schrödinger equation in optical fibers, Phys. D 134284 (2024). https://doi.org/10.1016/j.physd.2024.134284
[25] 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.
[26] J. Wang, Y. Liu, L. Yan, K. Han, L. Feng, R. Zhang, Fractional sub-equation neural networks (fSENNs) method for exact solutions of space–time fractional partial differential equations, Chaos Interdiscip. J. Nonlinear Sci. 35 (4) (2025) 043110.
[27] R.-F. Zhang, S. Bilige, Bilinear neural network method to obtain the exact analytical solutions of nonlinear partial differential equations and its application to P-GBKP equation, Nonlinear Dyn. 95 (4) (2019) 3041–3048.
[28] H. Zhang, R. Zhang, Q. Liu, A novel multi-modal neurosymbolic reasoning intelligent algorithm for BLMP equation, Chin. Phys. Lett. 42 (10) (2025) 100002.
[29] R.-F. Zhang, M.-C. Li, Bilinear residual network method for solving the exactly explicit solutions of nonlinear evolution equations, Nonlinear Dyn. 108 (1) (2022) 521–531.
[30] X.-R. Xie, R.-F. Zhang, Neural network-based symbolic calculation approach for solving the Korteweg–de Vries equation, Chaos Solit. Fractals 194 (2025) 116232.
[31] W. E, B. Yu, The deep ritz method: a deep learning-based numerical algorithm for solving variational problems, Commun. Math. Stat. 6 (1) (2018) 1–12.
[32] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (3) (2021) 218–229.
[33] Z. Li, N.B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A.M. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, in: International Conference on Learning Representations (ICLR), 2021.
[34] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, Proc. Natl. Acad. Sci. 115 (34) (2018) 8505–8510.
[35] J. Sirignano, K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018) 1339–1364.
[36] X. Guo, W. Li, F. Iorio, Convolutional neural networks for steady flow approximation, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 481–490.
[37] R.K. Tripathy, I. Bilionis, Deep learning of dynamical systems from data: a deep neural network model for linear and nonlinear partial differential equations, J. Comput. Phys. 375 (2018) 565–588.
[38] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. 394 (2019) 56–81.
[39] H. Tang, L. Lu, G.E. Karniadakis, Deep learning of turbulent scalar mixing using physics-informed autoencoders, J. Fluid Mech. 900 (2020) A37.
[40] J. Berg, K. Nyström, A unified deep artificial neural network approach to partial differential equations in complex geometries, Neurocomputing 317 (2018) 28–41.
[41] L. Yang, X. Meng, G.E. Karniadakis, Unsupervised learning of PDEs with physics-informed neural networks, J. Comput. Phys. 425 (2021) 109913.
[42] H. Chen, K. Zhang, S. Cai, Physics-informed generative adversarial networks for stochastic solution of PDEs, J. Comput. Phys. 419 (2020) 109675.
[43] N. Geneva, N. Zabaras, Modeling the dynamics of PDE systems with unsupervised deep learning, J. Comput. Phys. 403 (2020) 109056.
[44] Z. Mao, A.D. Jagtap, G.E. Karniadakis, Physics-informed neural networks for high-speed flows, Comput. Methods Appl. Mech. Eng. 360 (2020) 112789.
[45] S. Tian, Y. Yu, B. Li, A data-driven adaptive deep learning method for nonlinear Schrödinger-type equation, Chin. Phys. Lett. (2025). https://doi.org/10.1088/0256-307X/42/12/120001
[46] S. Tian, B. Li, Z. Zhang, Data-driven AI- and bi-soliton of the cylindrical Korteweg–de -Vries equation via prior-information physics-informed neural networks, Chin. Phys. Lett. (2024). https://doi.org/10.1088/0256-307X/41/4/040201
[47] M.-H. Guo, X. Lü, Y.-X. Jin, Extraction and reconstruction of variable-coefficient governing equations using Res-KAN integrating sparse regression, Phys. D 134689 (2025). https://doi.org/10.1016/j.physd.2025.134689
[48] C. Han, X. Lü, Variable coefficient-informed neural network for PDE inverse problem in fluid dynamics, Phys. D 134362 (2024). https://doi.org/10.1016/j.physd.2024.134362
[49] Y.-B. Su, X. Lü, S.-K. Li, L.-X. Yang, Z. Gao, Self-adaptive equation embedded neural networks for traffic flow state estimation with sparse data, Phys. Fluids 36 (104127) (2024). https://doi.org/10.1063/5.0230757
[50] J. Bilski, J. Smołąg, B. Kowalczyk, K. Grzanek, I. Izonin, Fast computational approach to the Levenberg–Marquardt algorithm for training feedforward neural networks, J. Artif. Intell. Soft Comput. Res. 13 (2) (2023) 45–61.
[51] H. Riaz, A. Farooq, Hybrid analytical and neural-network approaches to the non-local short pulse equation, Eur. Phys. J. C 85 (8) (2025) 895.
[52] A. Waheed, M. Awais, M. Raja, M.Y. Malik, A.S. Alqahtani, Peakon and solitary wave solutions of the LAX equation: neuro computing procedure, Int. Commun. Heat Mass Transf. 152 (2024) 107321.
[53] H. Riaz, A. Farooq, Solitonic solutions for the reduced Maxwell-Bloch equations via the Darboux transformation and artificial neural network in nonlinear wave dynamics, Phys. Scr. 99 (12) (2024) 126010.
[54] H. Riaz, A. Farooq, A (2 + 1) modified KdV equation with time-dependent coefficients: exploring soliton solution via Darboux transformation and artificial neural network approach, Nonlinear Dyn. 113 (4) (2025) 3695–3711.
[55] A. Farooq, H.W.A. Riaz, W.X. Ma, On symmetry solutions of nonlocal complex coupled dispersionless system using Darboux transformation and artificial neural networks, Chin. J. Phys. 97 (2025) 933–947.
[56] V.B. Matveev, M.A. Salle, Darboux Transformations and Solitons, Springer Berlin Heidelberg, 1991.
[57] W.-M. Ma, Darboux transformations for a Lax integrable system in 2n dimensions, Lett. Math. Phys. 39 (1997) 33–49.

[58] Z. Amjad, B. Haider, W.-X. Ma, Integrable discretization and multi-soliton solutions of negative order AKNS equation, Qual. Theory Dyn. Syst. 23 (Suppl 1) (2024) 280.
[59] Z. Amjad, B. Haider, Darboux transformations of supersymmetric Heisenberg magnet model, J. Phys. Commun. 2 (3) (2018) 035019.
[60] H.W.A. Riaz, Quasi-Wronskian solitons in the non-commutative kuralay-IIA equation: analysis and simulations, Phys. Scr. 100 (2) (2025) 025239.
[61] W.-X. Ma, Matrix extension of the Kuralay–II equation and its associated Darboux transformation, Wave Motion (2025) 103644. .
[62] A. Fischer, A.F. Izmailov, M.V. Solodov, The Levenberg–Marquardt method: an overview of modern convergence theories and more, Comput. Optim. Appl. 89 (1) (2024) 33–67.
[63] X. Lv, Z. Yu, A modified Levenberg–Marquardt algorithm for low order-value optimization problem, J. Appl. Math. Comput. 70 (6) (2024) 5629–5644.
[64] H. Yu, B.M. Wilamowski, Levenberg–Marquardt training, in: Intelligent Systems, CRC Press, 2018, pp. 12–1.
[65] B.J. Erickson, P. Korfiatis, Z. Akkus, T.L. Kline, Machine learning for medical imaging, Radiographics 37 (2) (2017) 505–515.
[66] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (4) (1989) 303–314. https://doi.org/10.1007/BF02551274
[67] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Netw. 4 (2) (1991) 251–257. https://doi.org/10.1016/0893-6080(91)90009-T
[68] A. Pinkus, Approximation theory of the MLP model in neural networks, Acta Numer. 8 (1999) 143–195. https://doi.org/10.1017/S0962492900002919
[69] S. Roweis, Levenberg-marquardt optimization, Notes Univ. Tor. 52 (1996) 1027.