# Least Square Support Vector Machine Framework for Meshless and Accurate Solution of Higher Order Boundary Value Problems with Comparative Analysis of Machine Learning Techniques

Bhubaneswari Mishra and Snehashish Chakraverty*

Department of Mathematics, National Institute of Technology Rourkela, Rourkela, 769008, India


**INFORMATION**

**Keywords:**
Simulation-based engineering
LS-SVM
high-order BVP
meshless method
RBF kernel
computational mechanics
data-driven modelling

**DOI:** 10.23967/j.rimni.2025.10.72376

SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI in cooperation with CIMNE

# Least Square Support Vector Machine Framework for Meshless and Accurate Solution of Higher Order Boundary Value Problems with Comparative Analysis of Machine Learning Techniques

**Bhubaneswari Mishra and Snehashish Chakraverty***

Department of Mathematics, National Institute of Technology Rourkela, Rourkela, 769008, India

**ABSTRACT**

This paper presents an enhanced Least Squares Support Vector Machine (LS-SVM) approach for meshless and accurate solution of higher-order boundary value problems (BVPs) that commonly arise in structural mechanics, fluid dynamics, and other engineering fields. The discussed method formulates third- and fourth-order linear and nonlinear ordinary differential equations (ODEs) as data-driven optimization problems, eliminating the need for traditional mesh-based discretization. Leveraging a Radial Basis Function (RBF) kernel and regularization-based control of model complexity, the LS-SVM captures complex solution behaviour while maintaining stability and smoothness. The meshless nature of the model ensures geometry-independence, making it suitable for irregular or multi-point boundary conditions. A comparative analysis with established machine learning techniques, including Ridge Regression (RR), classical SVM, Random Forest (RF), and Extreme Gradient Boosting (XGB), demonstrates the competitive accuracy, robustness, and efficiency of LS-SVM. The results highlight its potential as a promising solver for nonlinear and multi-point problems where meshless methods are advantageous. The results highlight its potential as a promising solver for simulation-based workflows in computational mechanics and scientific computing, where adaptability, generalization, and reliability are critical.

**OPEN ACCESS**

**Received:** 25/08/2025

**Accepted:** 21/10/2025

**DOI**
10.23967/j.rimni.2025.10.72376

**Keywords:**
Simulation-based engineering
LS-SVM
high-order BVP
meshless method
RBF kernel
computational mechanics
data-driven modelling

## 1 Introduction

Boundary value problems (BVPs) governed by ODE are fundamental in modeling many natural and dynamic systems across biology, economics and applied engineering. Due to their wide relevance, finding accurate and efficient numerical methods to solve such problems has been a long standing problem in computational mathematics. Among the earliest methods, the finite difference method (FDM) [1] was established as a reliable tool to solve two-point BVPs for both linear and nonlinear higher order ODEs. Later the homotopy perturbation method [2,3] was introduced to solve fourth and sixth order cases. To generalize these methods for more complex scenarios including multi-point boundary conditions the optimal homotopy asymptotic method was introduced by Ali et al. [4] which

*Correspondance: Snehashish Chakraverty (sne_chak@yahoo.com). 

provides more accuracy and computational flexibility. Over the years many other methods have been developed. The Adomian Decomposition Method has been widely used due to its semi-analytical formulation and adaptability to nonlinear problems [5–7]. Several studies have further demonstrated its efficiency and accuracy across diverse applications [8–10]. Meanwhile wavelet based methods especially Haar and Shannon wavelet transforms [11,12] have gained popularity for their localized resolution and multiscale capabilities. Spectral methods are known for their high accuracy have also contributed a lot. Noteworthy among these is the work of Doha et al. [13], who applied spectral Galerkin techniques based on Jacobi polynomials for solving third- and fifth-order problems, and subsequently expanded this framework using Chebyshev polynomials of the third and fourth kinds to tackle more intricate higher-order differential systems [14]. The use of shifted Jacobi collocation techniques has further enhanced the handling of nonlinear multi-point BVPs [15]. Additionally, sinc-collocation methods have been effectively utilized in this context, as shown by Saadatmandi and Dehghan [16]. The variational iteration method has also proven to be a valuable strategy for both linear and nonlinear high-order BVPs, as explored in various investigations [17–19].

Despite the effectiveness of the aforementioned numerical methods, a persistent limitation lies in the discontinuity of the approximate solution's derivatives, which may compromise the stability and convergence of the solution, particularly in sensitive boundary value problems. Neural networks (NN), a subclass of machine intelligence techniques, have emerged as promising alternatives due to their universal function approximation capability [20–22]. These models generate smooth, analytically differentiable solutions in closed form, making them highly suitable for problems requiring continuous solution behavior. Neural network (NN) [23,24] based methodologies have been widely explored for solving a broad spectrum of differential equations. These include ODE, partial differential equations (PDEs) [25–27], fractional-order models [28–30] and integro-differential systems [31,32]. In one study, Chakraverty and Mall [33] designed a neural network model based on regression principles to address fourth-order linear ODEs with two-point boundary conditions. A related effort by Malek and Shekari Beidokhti [34] combined optimization techniques with feedforward neural networks to solve similar fourth-order BVPs. Additionally, Mai-Duy [35] utilized radial basis function neural networks for direct solution approximation in high-order boundary value problems. Several studies have explored advanced numerical and intelligent techniques for complex boundary value and dynamic problems. These include higher-order numerical formulations [36,37], analytical–approximate schemes for nonlinear wave equations [38], and neural-network-based approaches for fuzzy structural systems [39].

However, artificial neural networks are not without drawbacks. They often require the tuning of a large number of hyperparameters, including the number of hidden layers and neurons, which lacks a systematic selection criterion. Additionally, the training process is computationally intensive, prone to local minima, and often demands significant effort for convergence and generalization. An alternative to neural networks is the Support Vector Machine (SVM) algorithm, originally proposed by Vapnik within the framework of statistical learning theory [40]. SVM map input data into high-dimensional feature spaces via nonlinear kernel functions and solve a convex quadratic optimization problem to ensure global optimality. Their adoption of the structural risk minimization principle also enhances their generalization ability. A refinement of the classical SVM framework is the Least Squares Support Vector Machine (LS-SVM) algorithm, introduced to reduce computational complexity and improve numerical tractability [41]. In LS-SVM, the inequality constraints of the SVM are replaced with

equality constraints, and the loss function is reformulated as the sum of squared errors over the training dataset. As a result, the problem reduces to solving a linear system instead of a quadratic programming problem, significantly lowering training time while maintaining high approximation accuracy. Owing to their favorable characteristics, LS-SVM algorithms have found successful applications across diverse areas such as pattern classification [42], fault detection systems [43], and forecasting of time-dependent data [44,45]. In recent years, their utility has extended into the field of scientific computing, where LS-SVM has been effectively utilized to tackle differential equations [46,47], differential-algebraic systems [48,49]. Physics Informed LSSVM which is another advanced form of LSSVM is introduced in [50].

However, the application of LS-SVM methods to boundary value problems remains limited. Most existing works focus on second-order linear ODEs with two-point boundary conditions [46]. To date, there has been limited exploration of LS-SVM techniques for higher-order and nonlinear ODEs with two-point and multi-point boundary conditions. This gap highlights the need for further investigation into the potential of LS-SVM algorithms to serve as efficient and accurate solvers for a broader class of high-order boundary value problems.

## 2 Least Square Support Vector Machine

The training dataset is as follows $\{(x_m, y_n)\,|x_m \in R^n, y_m \in R\}_{m-1}^{N}, n = 1$, where the input data is given by $\{x_m\}_{m-1}^{N}$ and the output data is given by $\{y_m\}_{m-1}^{N}$. It is assumed that the corresponding functional relationship between the input and output variables can be represented by the following form:

$$y(x) = \omega^T \varphi(x) + b, \tag{1}$$

here $\omega$ and $b$ are the model parameters and $\varphi(x)$ is a nonlinear function that maps the input data into a higher-dimensional feature space. The idea is to find the best solution in this space by minimizing the difference between the model's predictions and the actual observed data [51]. Based on this, the primal form LS-SVM is defined by the following optimization problem [41,52]:

$$\min_{\omega,b,e_m} J(\omega, e) = \frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma e^T e, \tag{2}$$

subject to

$$y(x) = \omega^T \varphi(x) + b + e_m, \quad m = 1, 2, 3, \ldots N,$$

here $\gamma$ is a positive regularization parameter and $e_m$ represents the error associated with the $m'th$ training input. In this formulation, first term acts as a regularization component that controls the model complexity and second term penalizes the training errors. The resulting constrained optimization problem, as shown in Eq. (2), can be efficiently solved using Lagrange multipliers.

$$L(\omega, \alpha_m e_m) = \frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma e^T e - \sum_{m=1}^{N} \alpha_m \left[\omega^T \varphi(x_m) + b + e_m - y_m\right],$$

$$(m = 1, 2, 3, \ldots, N). \tag{3}$$

In this context, $\alpha_m$ are Lagrange multipliers which can take positive or negative values in the LS-SVM framework. By applying the Karush–Kuhn–Tucker (KKT) conditions, we obtain the following:

$$\begin{cases} \dfrac{\partial L}{\partial \omega} = \omega - \sum_{m=1}^{N} \alpha_m \varphi\left(x_m\right) \\ \dfrac{\partial L}{\partial b} = \sum_{m=1}^{N} \alpha_m = 0; \\ \dfrac{\partial L}{\partial e_m} = \alpha_m - \gamma e_m = 0; \\ \dfrac{\partial L}{\partial \alpha_m} = \omega^T \varphi\left(x_m\right) + b - \gamma_m + e_m = 0. \end{cases} \tag{4}$$

After eliminating $\omega$ and $e_m$ from the system of Eq. (4), the following linear system is obtained:

$$\begin{bmatrix} \Theta_{mn} + \gamma^{-1} E | I_{N-1}^T \\ \hline I_{N-1} | 0 \end{bmatrix} \begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix}, \tag{5}$$

where $\Theta_{mn} = K\left(x_m, x_n\right) = \varphi\left(x_m\right)^T \varphi\left(x_n\right) (m, n = 1, 2, \ldots, N)$ is the $mn$'th entry of the kernel matrix; $Y = [y_1, y_2, \ldots y_N]^T; \alpha = [\alpha_1, \alpha_2, \ldots \alpha_N]^T$ ; *and* $I_{N-1} = [1, 1, \ldots, 1]$.

Finally, the LS-SVM model can be expressed in its dual form as follows:

$$y\left(x\right) = \sum_{m=1}^{N} \alpha_m K\left(x_m, x\right) + b. \tag{6}$$

## 3 Overview and Mathematical Foundations of LS-SVM for ODE

This section provides a concise overview of the LS-SVM framework as applied to the solution of ODEs, along with the necessary definitions. Consider, for instance, a first-order linear ODE in the context of an initial value problem, expressed in the following general form [46]:

$$\begin{cases} \dfrac{dy}{dx} = a\left(x\right) y\left(x\right) + r\left(x\right), \quad x \in [a, c] \\ y\left(a\right) = A, \end{cases}. \tag{7}$$

As described in [46], the method begins by assuming a general form for the approximate solution:$y\left(x\right) = \omega^T \varphi\left(x\right) + b$, where $\omega$ and $b$ are the unknown parameters to be identified. The domain of the solution is divided into a finite number of collocation points with a suitable collocation technique [53]. At these points the parameters are optimized by solving a constrained optimization problem, as in [46]. To do this the method of Lagrange multipliers is used [54], so the optimization problem can be rewritten as a Lagrangian that combines the LS-SVM cost function with the constraints from the first order linear ODE and the initial condition, both enforced at the collocation points.

This is not limited to first order problems. It can be extended to higher order problems, second order boundary value problems and partial differential equations, as in [46–48]. By using Mercer's theorem [40], the derivative of the kernel function is defined as:

$$\nabla_{n,m}\left(K\left(x_m, x_n\right)\right) = \left.\frac{\partial^{i+j}\left(K\left(u, v\right)\right)}{\partial u^i \partial v^j}\right|_{u=x_m, v=x_n} = \varphi^{(i)}\left(x_m\right)^T \varphi^{(i)}\left(x_n\right) = \left[\Theta_{i,j}\right]_{m,n}, \tag{8}$$

here, RBF kernel is considered as kernel function and can be denoted as $K\left(u, v\right) = \exp\left(-\left(u - v\right)^2 / \sigma^2\right)$.

After this we have:

$$\nabla_{1,3}\left(K\left(x_m,x_n\right)\right)=\left.\frac{\partial^4\left(K\left(u,v\right)\right)}{\partial u\partial v^3}\right|_{u=x_m,v=x_n}=\varphi^{(1)}\left(x_m\right)^T\varphi^{(3)}\left(x_n\right)=\left[\Theta_{1,3}\right]_{m,n}$$
$$=-\left[\frac{12}{\sigma^4}-\frac{12}{\sigma^2}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^2+\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^4\right]K\left(x_m,x_n\right);$$
$$\nabla_{2,3}\left(K\left(x_m,x_n\right)\right)=\left.\frac{\partial^5\left(K\left(u,v\right)\right)}{\partial u^2\partial v^3}\right|_{u=x_m,v=x_n}=\varphi^{(2)}\left(x_m\right)^T\varphi^{(3)}\left(x_n\right)=\left[\Theta_{2,3}\right]_{m,n}$$
$$=-\left[\frac{60}{\sigma^4}-\frac{60}{\sigma^2}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^2+\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^4\right]\frac{2\left(x_m-x_n\right)}{\sigma^2}K\left(x_m,x_n\right);$$

$$\nabla_{3,3}\left(K\left(x_m,x_n\right)\right)=\left.\frac{\partial^6\left(K\left(u,v\right)\right)}{\partial u^3\partial v^3}\right|_{u=x_m,v=x_n}=\varphi^{(3)}\left(x_m\right)^T\varphi^{(3)}\left(x_n\right)=\left[\Theta_{3,3}\right]_{m,n}$$
$$=\left[\frac{120}{\sigma^6}-\frac{120}{\sigma^4}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^2+\frac{30}{\sigma^2}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^4-\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^6\right]K\left(x_m,x_n\right);$$
$$\nabla_{3,4}\left(K\left(x_m,x_n\right)\right)=\left.\frac{\partial^7\left(K\left(u,v\right)\right)}{\partial u^3\partial v^4}\right|_{u=x_m,v=x_n}=\varphi^{(3)}\left(x_m\right)^T\varphi^{(4)}\left(x_n\right)=\left[\Theta_{3,4}\right]_{m,n}$$
$$=\left[\begin{array}{l}\frac{840}{\sigma^6}-\frac{840}{\sigma^4}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^2+\frac{42}{\sigma^2}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^4\\-\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^6\end{array}\right]\frac{2\left(x_m-x_n\right)}{\sigma^2}K\left(x_m,x_n\right);$$
$$\nabla_{4,4}\left(K\left(x_m,x_n\right)\right)=\left.\frac{\partial^8\left(K\left(u,v\right)\right)}{\partial u^4\partial v^4}\right|_{u=x_m,v=x_n}=\varphi^{(4)}\left(x_m\right)^T\varphi^{(4)}\left(x_n\right)=\left[\Theta_{4,4}\right]_{m,n}$$
$$=\left[\begin{array}{l}\frac{1680}{\sigma^8}-\frac{3360}{\sigma^6}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^2+\frac{840}{\sigma^4}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^4\\-\frac{56}{\sigma^2}\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^6+\left[\frac{2\left(x_m-x_n\right)}{\sigma^2}\right]^8\end{array}\right]\frac{2\left(x_m-x_n\right)}{\sigma^2}K\left(x_m,x_n\right).$$

### *3.1 Higher Order ODE with Boundary Conditions*

This section presents a discussion on various forms of generalized boundary value problems associated with higher order ODE.

### *3.2 Higher Order Linear and Nonlinear ODE with Two Point Boundary Condition*

This section describes the application of the improved LS-SVM algorithm to two-point boundary value problems involving higher order linear and nonlinear ODE, emphasizing its accuracy and computational efficiency.

#### *3.2.1 Nonlinear ODE for Two-Point Boundary Value Problem*

The two-point boundary value problem for a nonlinear ordinary differential equation of order M can be formally stated as follows [15]:

$$\frac{d^My}{dx^M}+a_{M-1}\left(x\right)\frac{d^{M-1}y}{dx^{M-1}}+\ldots+a_1\left(x\right)\frac{dy}{dx}=f\left(x,y\right),\quad x\in\left[a,c\right].\tag{9}$$

The boundary conditions are:

$y^{(t)}(a) = p$
$y^{z}(c) = q$

The interval, which it represented as [a, c], is devided into a series of collocation points. Let us assume that a general approximate solution to Eq. (9) is given by Eq. (1).

To determine the best values for $\omega$ and $b$ the following optimization problem is solved:

$$\min_{\omega,b,e.,\xi,y_m} J(\omega, e, \xi) = \frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma e^T e + \frac{1}{2}\gamma \xi^T \xi \tag{10}$$

Subject to

$\omega^T \varphi^{(M)}(x_m) + \sum_{l=1}^{M-1} \omega^T a_l(x_m)\varphi^{(l)}(x_m) = f\left(x_m, y_j\right) + e_m; \quad m = 2, \ldots, N-1;$
$y_m = \omega^T \varphi(x_m) + b + \xi_m,$
$\omega^T \varphi(x_1) + b = p_0;$
$\omega^T \varphi(x_N) + b = q_0;$
$\omega^T \varphi^{(t)}(x_1) + b = p_t; \quad t = 1, 2, \ldots, T;$
$\omega^T \varphi^{(z)}(x_N) + b = q_z; \quad z = 1, 2, \ldots, Z;$

### 3.2.2 *Linear ODE for Two-Point Boundary Value Problem*

The two-point boundary value problem for a linear ordinary differential equation of order M can be formally stated as follows [16]:

$$\frac{d^M y}{dx^M} + a_{M-1}(x)\frac{d^{M-1}y}{dx^{M-1}} + \ldots + a_1(x)\frac{dy}{dx} + a_0(x)y = r(x), \quad x \in [a, c]. \tag{11}$$

The boundary conditions are:

$y^{(t)}(a) = p$
$y^{z}(c) = q$

Let us assume that Eq. (11) is solved by adopting the approximate solution form introduced in Eq. (1), enforced at collocation points with associated boundary conditions. This method discretizes the solution domain into a finite set of collocation points which act as evaluation nodes where both the differential equation and the associated boundary conditions must be satisfied. The optimal parameters are then computed by solving the following optimization problem:

$$\min_{\omega,b,e_m} J(\omega, e) = \frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma e^T e \tag{12}$$

Subject to

$\omega^T \varphi^{(M)}(x_m) + \sum_{l=0}^{M-1} \omega^T a_l(x_m)\varphi^{(l)}(x_m) + a_0(x_m) b = r(x_m) + e_m; \quad m = 2, \ldots, N-1;$
$\omega^T \varphi(x_1) + b = p_0;$
$\omega^T \varphi(x_N) + b = q_0;$
$\omega^T \varphi^{(t)}(x_1) = p_s; \quad t = 1, 2, \ldots, T;$
$\omega^T \varphi^{(z)}(x_N) = q_t; \quad z = 1, 2, \ldots, Z;$

### *3.3 Higher Order ODE with Multi Point Boundary Conditions*

This section presents a detailed application of LS-SVM algorithm to multi-point boundary value problems governed by higher-order linear and nonlinear ODEs. It highlights the algorithm's ability to accurately approximate complex solution behaviors while maintaining computational efficiency. Emphasis is placed on the method's robustness in handling multiple boundary constraints and its effectiveness in capturing the underlying dynamics of both linear and nonlinear systems across diverse test cases.

#### *3.3.1 Nonlinear ODE for Multi Point Boundary Value Problems*

The multi-point boundary value problem for a nonlinear ordinary differential equation of order M can be formally stated as follows [15]:

$$\frac{d^M y}{dx^M} + a_{M-1}(x)\frac{d^{M-1}y}{dx^{M-1}} + \ldots + a_1(x)\frac{dy}{dx} = f(x,y), \quad x \in [a,c]. \tag{13}$$

Subject to

$$\begin{aligned}
&y^{(q_0)}(a) = s_0\\
&y^{(q_n)}\left(x_{pn}\right) = s_r\\
&y^{(q_{M-1})}(c) = s_{M-1}\\
&x_{pn} \in [a,c]\\
&p_n \in Z,\ n = 1,2,\ldots,M-2\\
&0 \le q_0, q_1, \ldots, q_{M-1} \le M-1.
\end{aligned}$$

To facilitate the numerical solution, the interval [a, c] is discretized into a set of collocation points. The primal optimization problem can be described as follows:

$$\min_{\omega,b,e,\xi,y_m} J(\omega,e,\xi) = \frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma e^T e + \frac{1}{2}\gamma \xi^T \xi \tag{14}$$

Subject to

$$\begin{aligned}
&\omega^T\left[\varphi^{(M)}(x_m) + \textstyle\sum_{l=1}^{M-1}\omega^T a_l(x_m)\varphi^{(l)}(x_m)\right] = f(x_m,y_m) + e_m; \quad m = 2,\ldots,N-1;\\
&y_m = \omega^T\varphi(x_m) + b + \xi_m, \quad m = 2,\ldots,N-M;\\
&\omega^T\varphi^{(q0)}(x_1) + b^{(q0)} = s_0;\\
&\omega^T\varphi^{(qn)}\left(x_{pn}\right) + b^{(qn)} = s_n, \quad n = 1,2,\ldots,M-2;\\
&\omega^T\varphi^{(qM-1)}(x_N) + b^{(qM-1)} = s_{M-1};
\end{aligned}$$

#### *3.3.2 Linear ODE for Multi Point Boundary Value Problem*

The multi-point boundary value problem for a nonlinear ordinary differential equation of order M can be formally stated as follows [15]:

$$\frac{d^M y}{dx^M} + a_{M-1}(x)\frac{d^{M-1}y}{dx^{M-1}} + \ldots + a_1(x)\frac{dy}{dx} + a_0(x)\,y = r(x), \quad x \in [a,c]. \tag{15}$$

Subject to

$$y^{(q_0)}(a) = s_0$$
$$y^{(q_n)}\left(x_{pn}\right) = s_n$$
$$y^{(q_{M-1})}(c) = s_{M-1}$$
$$x_{pn} \in [a, c]$$
$$p_n \in Z,\ n = 1, 2, \ldots, M-2$$
$$0 \le q_0, q_1, \ldots, q_{M-1} \le M-1.$$

The interval $[a, c]$ is divided into a series of collocation points. The original optimization problem can be stated as follows:

$$\min_{\omega, b, e_m} J(\omega, e) = \frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma e^T e \tag{16}$$

Subject to

$$\omega^T\left[\varphi^{(M)}(x_m) + \sum_{l=0}^{M-1} a_l(x_i)\,\varphi^{(l)}(x_m)\right] = r(x_m) + e_m;\quad m = 2, 3, \ldots, p_1 - 1, p_1 + 1, N-1;$$
$$\omega^T\varphi^{(q0)}(x_1) + b^{(q0)} = s_0;$$
$$\omega^T\varphi^{(qn)}\left(x_{pn}\right) + b^{(qn)} = s_n,\ \ n = 1, 2, \ldots, M-2;$$
$$\omega^T\varphi^{(qM-1)}(x_N) + b^{(qM-1)} = s_{M-1};$$

## 4 Numerical Examples

This section presents a series of computational experiments to investigate the reliability and accuracy of the LS-SVM framework for solving high-order boundary value problems. The study encompasses both linear and nonlinear ordinary differential equations of third and fourth order, subject to two-point and multi-point boundary conditions. These problems are selected to reflect a range of structural complexities, from smooth linear systems to nonlinear and oscillatory solutions, thereby allowing for a thorough evaluation of the model's stability, generalization ability, and approximation precision. For all experiments, the LS-SVM algorithm is applied with a radial basis function (RBF) kernel of the form $K(x, x') = \exp\left(-\gamma\,(x - x')^2\right)$, where $\gamma$ controls the width of the kernel and $\lambda$ denotes the regularization parameter balancing the trade-off between the fitting accuracy and model smoothness.

In all experiments, collocation points are selected as uniformly distributed nodes over the problem domain, while boundary points are fixed at the prescribed conditions. The kernel width parameter $\sigma$ and the regularization parameter $\lambda$ are determined through a two-stage process: first, a coarse grid search is conducted, followed by fine-tuning using cross-validation. The resulting kernel matrices are occasionally ill-conditioned; to address this, a small diagonal loading is introduced as a stabilization strategy. All linear systems derived from the KKT conditions are solved using direct solvers available in standard numerical software. These steps ensure that the LS-SVM framework remains numerically stable and reproducible across different boundary-value problems.

To ensure consistent performance and to prevent ill-conditioning of the system matrix, the parameters are selected carefully through empirical validation. Specifically, for the third-order linear boundary value problem, $\gamma = 1.0$ and $\lambda = 10^{-4}$ are chosen; for the nonlinear fourth-order case, the same values of $\gamma = 1.0$ and $\lambda = 10^{-4}$ are retained. In the case of the fourth-order linear multi-point boundary value problem, a finer kernel width and stronger regularization are required, leading to the selection of $\gamma = 2.5$ and $\lambda = 10^{-6}$, while for the nonlinear third-order multi-point boundary condition

problem, the values $\gamma = 3.0$ and $\lambda = 10^{-6}$ are used to capture more localized features. To benchmark the effectiveness of LS-SVM, its results are compared with those from four prominent machine learning regressors: ridge regression, SVM, RF and XGB. All models are trained on uniformly spaced points over the domain [0, 1], and the performance is quantified using three standard error metrics. The mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) are defined as follows:

$$\begin{aligned} &MAE\colon \tfrac{1}{2}\sum_{i=1}^{N}\left|u_{exact,i}-u_{predicted,i}\right| \\ &MSE\colon \tfrac{1}{2}\sum_{i=1}^{N}\left(u_{exact,i}-u_{predicted,i}\right)^2 \\ &RMSE\colon \sqrt{MSE}=\sqrt{\tfrac{1}{2}\sum_{i=1}^{N}\left(u_{exact,i}-u_{predicted,i}\right)^2} \end{aligned} \tag{17}$$

These metrics are reported alongside graphical visualizations including prediction overlays, 3D surfaces, and error heatmaps to provide a comprehensive view of solver behavior under different modeling scenarios. The subsequent subsections present the detailed analysis for each problem.

To ensure reproducibility, each numerical example specifies the number and distribution of collocation and boundary points, along with the chosen values of $\sigma$ and $\lambda$. This transparency allows independent replication of the experiments without the need for explicit runtime or hardware specifications.

### *4.1 Example 1*

Here we begin with a smooth third-order linear BVP to assess baseline accuracy. Let's consider the 3rd order linear ODE:

$$y^3(x) = \cos(x), \quad x \in [0, 1]. \tag{18}$$

The boundary conditions are,

$$y(0) = 0, \quad y'(0) = 1,$$

$$y''(1) = -\sin(1).$$

The first test case involves a third-order linear boundary value problem defined on the interval [0, 1], characterized by trigonometric forcing and mixed boundary conditions. The differential structure is smooth but non-trivial, making it suitable for evaluating the stability and precision of numerical solvers. For this problem, 50 collocation points are uniformly distributed in the domain, and the parameters $\sigma = 1.0$, $\lambda = 10^{-4}$ are employed. As shown in Table 1, the LS-SVM method yields values in excellent agreement with the reference solution across the domain. Ensemble methods such as RF and XGB demonstrate competitive performance but display slight deviations at domain edges. In contrast, classical machine learning models like Ridge regression and SVM tend to produce systematic underfitting, particularly in regions with steep gradients.

**Table 1:** Solution comparison table (example 1)

| X | Exact | LS_SVM | Ridge | SVM | RF | XGB |
|---|---|---|---|---|---|---|
| 0.090909 | 0.1816 | 0.1809 | 0.2011 | 0.2316 | 0.1619 | 0.1903 |
| 0.19192 | 0.3826 | 0.3822 | 0.3883 | 0.4048 | 0.3276 | 0.3848 |
| 0.29293 | 0.5816 | 0.5821 | 0.5755 | 0.5780 | 0.4982 | 0.5526 |

(Continued)

**Table 1 (continued)**

| X | Exact | LS_SVM | Ridge | SVM | RF | XGB |
|---|---|---|---|---|---|---|
| 0.39394 | 0.7777 | 0.7783 | 0.7627 | 0.7512 | 0.8351 | 0.7765 |
| 0.49495 | 0.9699 | 0.9699 | 0.9499 | 0.9244 | 0.9708 | 0.9421 |
| 0.59596 | 1.1573 | 1.1566 | 1.1371 | 1.0977 | 1.1685 | 1.1319 |
| 0.69697 | 1.3389 | 1.3384 | 1.3243 | 1.2709 | 1.3547 | 1.2999 |
| 0.79798 | 1.5139 | 1.5144 | 1.5115 | 1.4441 | 1.4395 | 1.5094 |
| 0.89899 | 1.6817 | 1.6826 | 1.6987 | 1.6173 | 1.7413 | 1.6853 |
| 1 | 1.8415 | 1.8391 | 1.8859 | 1.7905 | 1.724 | 1.7956 |

Fig. 1 depicts the three dimensional surface of the LS-SVM solution, capturing the gradual increase in function values with high smoothness. The associated error surface, shown in Fig. 2, confirms that the LS-SVM maintains pointwise absolute errors predominantly below $10^{-3}$. Table 2 presents the absolute error at selected points for all models, highlighting that the LS-SVM consistently outperforms other approaches in localized accuracy. The RF and XGB models, while more expressive than linear models, exhibit occasional fluctuations due to their sensitivity to local data irregularities. Ridge and SVM, constrained by their linear or kernel-imposed bias, fail to capture finer solution variations.

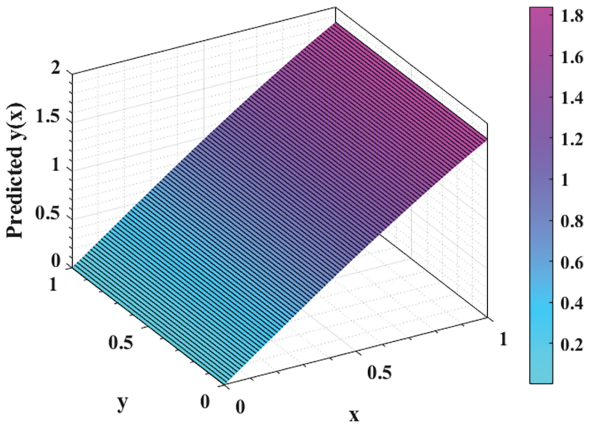


**Figure 1:** 3D surface plot of LSSVM solution (example 1)

Fig. 3 compares the predictions from all methods against the reference solution. The LS-SVM curve aligns closely throughout the domain, including in regions where other models diverge. Inset magnification illustrates this further, showing visible discrepancies in RF, Ridge, and SVM approximations. Quantitative evaluation is provided in Table 3, where LS-SVM achieves significantly lower error metrics MAE, MSE, and RMSE than the other techniques. These trends are concisely visualized in the error heatmap (Fig. 4), which underscores the superiority of the LS-SVM across all criteria. The results confirm the method's reliability in preserving higher-order solution characteristics, particularly under constraints involving derivative boundary conditions.

We observe that LS-SVM consistently achieves high accuracy across the entire domain, with errors remaining minimal even near boundary conditions. Compared to classical regressors and ensemble

methods, LS-SVM demonstrates superior stability and captures smooth variations without distortion, establishing a strong baseline for subsequent nonlinear and multi-point problems.

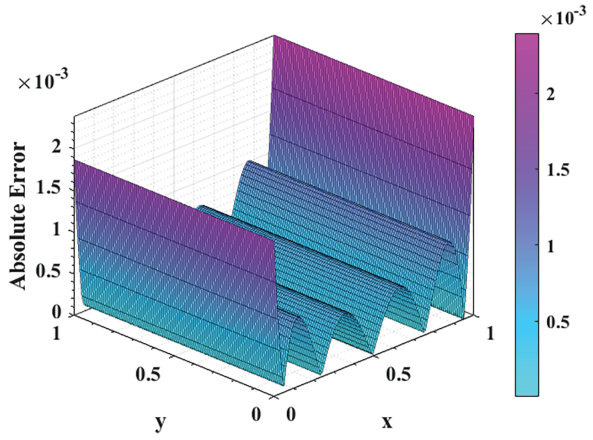


**Figure 2:** Error surface plot of LSSVM (example 1)

**Table 2:** Absolute error across comparing methods (example 1)

| X | Err_LS | Err_Ridge | Err_SVM | Err_RF | Err_XGB |
|---|---|---|---|---|---|
| 0.0909 | 0.00072171 | 0.019473 | 0.049907 | 0.019731 | 0.0086075 |
| 0.1912 | 0.00042947 | 0.0056979 | 0.022154 | 0.054979 | 0.0021381 |
| 0.2923 | 0.00045454 | 0.0061324 | 0.0036549 | 0.083429 | 0.029046 |
| 0.3934 | 0.00059463 | 0.015019 | 0.02652 | 0.057371 | 0.0012052 |
| 0.4945 | 2.6002e−05 | 0.019993 | 0.045472 | 0.0008583 | 0.027807 |
| 0.5956 | 0.00064812 | 0.020124 | 0.059582 | 0.011261 | 0.025372 |
| 0.6967 | 0.00048255 | 0.014534 | 0.06797 | 0.015798 | 0.039005 |
| 0.7978 | 0.00050057 | 0.0023995 | 0.069814 | 0.07438 | 0.0045511 |
| 0.8989 | 0.00091716 | 0.017033 | 0.064359 | 0.05966 | 0.0036153 |
| 1 | 0.002393 | 0.044446 | 0.050926 | 0.11751 | 0.045847 |

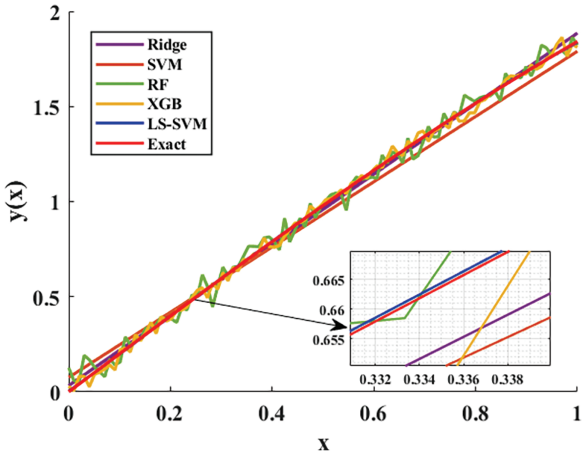


**Figure 3:** Solution comparison across ML methods (example 1)

**Table 3:** Error metrics across comparing methods (example 1)

| . | MAE | MSE | RMSE |
|---|---|---|---|
| LS-SVM | 0.000561 | 4.5939e−07 | 0.000677 |
| Ridge | 0.015602 | 0.000328 | 0.018126 |
| SVM | 0.047288 | 0.002693 | 0.051897 |
| RF | 0.043186 | 0.002845 | 0.05334 |
| XGB | 0.021601 | 0.000743 | 0.027276 |

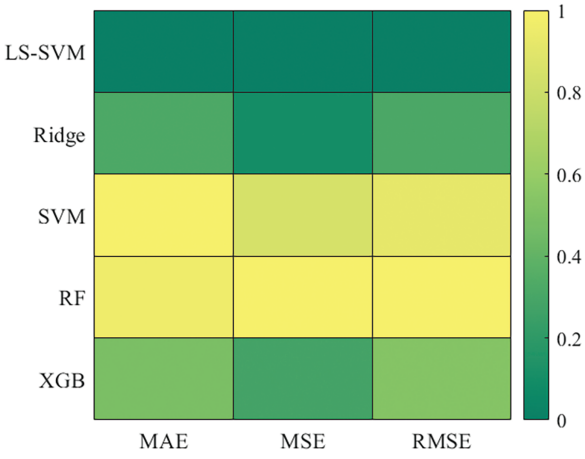


**Figure 4:** Error metrics heatmap (example 1)

### *4.2 Example 2*

Next, a fourth-order nonlinear ODE is considered to examine performance under nonlinearity.

$$y^4(x) + 12y(x)^2 = 0, x \in [0, 1]. \tag{19}$$

The two-point boundary conditions are given by,

$y(0) = 0, y'(0) = 0,$
$y(1) = 0, y'(1) = 0.$

The second example addresses a fourth-order boundary value problem incorporating nonlinearity through a quadratic term in the dependent variable. The solution is compactly supported and symmetric about the midpoint of the domain, making it a suitable candidate for evaluating the performance under smooth but nonlinear dynamics. A uniform grid of 60 collocation points is used, with $\sigma = 1.0$ and $\lambda = 10^{-4}$. As shown in Table 4, the LS-SVM model achieves strong agreement with the reference values, while other machine learning methods especially Ridge regression and SVM fail to capture the nonlinearity, reverting to nearly constant outputs. This is particularly evident in regions of curvature where both models completely miss the profile, likely due to their insufficient functional representation.

**Table 4:** Solution comparison table (example 2)

| X | Exact | LS_SVM | Ridge | SVM | RF | XGB |
|---|---|---|---|---|---|---|
| 0.010101 | 9.998e−05 | −0.000958 | 0.033 | 0.034655 | −0.054051 | 0.044726 |
| 0.12121 | 0.011346 | 0.011891 | 0.033 | 0.034655 | 0.046181 | −0.022391 |
| 0.23232 | 0.031808 | 0.031836 | 0.033 | 0.034655 | 0.025157 | 0.059035 |
| 0.33333 | 0.049383 | 0.049204 | 0.033 | 0.034655 | 0.0653 | 0.026545 |
| 0.44444 | 0.060966 | 0.060915 | 0.033 | 0.034655 | 0.035276 | 0.002535 |
| 0.55556 | 0.060966 | 0.060917 | 0.033 | 0.034655 | 0.018299 | 0.055084 |
| 0.66667 | 0.049383 | 0.049204 | 0.033 | 0.034655 | 0.025312 | 0.052693 |
| 0.76768 | 0.031808 | 0.031836 | 0.033 | 0.034655 | 0.182822 | 0.031995 |
| 0.87879 | 0.011346 | 0.011891 | 0.033 | 0.034655 | −0.036089 | −0.005792 |
| 0.9899 | 9.998e−05 | −0.000958 | 0.033 | 0.034655 | −0.009220 | 0.011384 |

Fig. 5 presents the LS-SVM solution surface, clearly reflecting the bell-shaped structure of the expected output. In Fig. 6, the error surface illustrates that the absolute deviation remains uniformly below $1.5 \times 10^{-3}$, with minimal boundary-layer discrepancies. Table 5 provides a detailed view of absolute error at representative points. LS-SVM maintains sub-milliscale errors across all coordinates, while the remaining models produce either overfitting (in the case of RF) or flat, misaligned approximations (as observed with Ridge and SVM). Notably, RF exhibits high-frequency oscillations and instability in central zones, indicative of poor generalization despite increased model flexibility.

Fig. 7 shows a direct overlay of predictions from all models. LS-SVM traces the reference curve closely, with nearly imperceptible deviation. In contrast, Ridge and SVM degenerate to flat baselines, and ensemble models fluctuate unpredictably, misrepresenting both amplitude and location of maxima. These qualitative differences are quantitatively confirmed in Table 6, where LS-SVM outperforms all alternatives across MAE, MSE, and RMSE by significant margins. The error metrics heatmap in Fig. 8 further consolidates this finding. LS-SVM occupies the lowest error region across all axes. This demonstrates the model's robustness in resolving smooth nonlinear structures, aided by its ability to implicitly encode differential constraints through the kernel system.

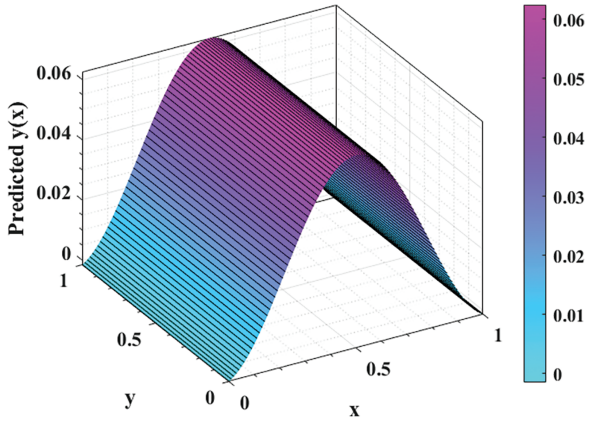


**Figure 5:** 3D surface plot of LSSVM solution (example 2)

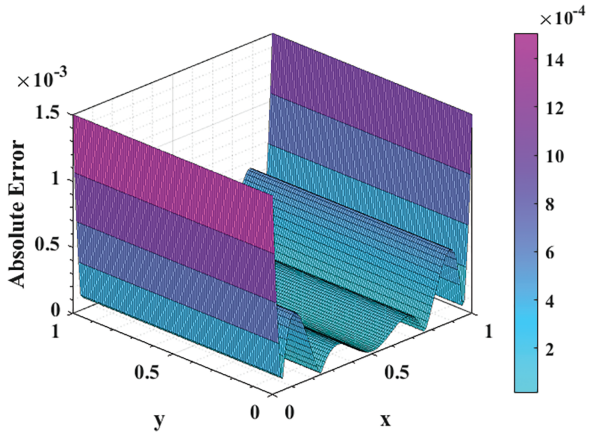


**Figure 6:** Error surface plot of LSSVM (example 2)

**Table 5:** Absolute error across comparing methods (example 2)

| X | Err_LS | Err_Ridge | Err_SVM | Err_RF | Err_XGB |
|---|---|---|---|---|---|
| 0.010101 | 0.001058 | 0.032970 | 0.034555 | 0.054151 | 0.044626 |
| 0.12121 | 0.000544 | 0.021654 | 0.023309 | 0.034835 | 0.033736 |
| 0.23232 | 2.779e−05 | 0.001191 | 0.002846 | 0.006651 | 0.027226 |
| 0.33333 | 0.000178 | 0.016383 | 0.014728 | 0.015917 | 0.022838 |
| 0.44444 | 5.627e−05 | 0.027966 | 0.026311 | 0.02569 | 0.058431 |
| 0.55556 | 5.627e−05 | 0.027966 | 0.026311 | 0.042668 | 0.005886 |
| 0.66667 | 0.000178 | 0.016383 | 0.014728 | 0.024071 | 0.003309 |
| 0.76768 | 2.7793e−05 | 0.001191 | 0.002846 | 0.151016 | 0.000181 |
| 0.87879 | 0.000544 | 0.021654 | 0.023309 | 0.047426 | 0.017139 |
| 0.9899 | 0.001058 | 0.0329 | 0.034555 | 0.009320 | 0.011284 |

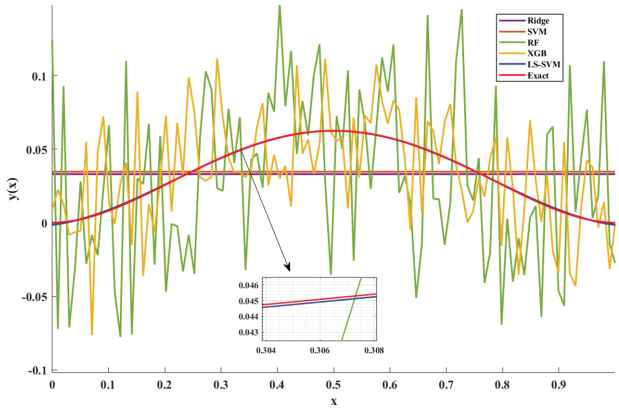


**Figure 7:** Solution comparison across ML methods (example 2)

**Table 6:** Error metrics across comparing methods (example 2)

| | MAE | MSE | RMSE |
|---|---|---|---|
| LS-SVM | 0.000258 | 1.4241e−07 | 0.000377 |
| Ridge | 0.019717 | 0.000482 | 0.021964 |
| SVM | 0.019688 | 0.000485 | 0.022027 |
| RF | 0.038482 | 0.002292 | 0.047875 |
| XGB | 0.024178 | 0.000913 | 0.030226 |

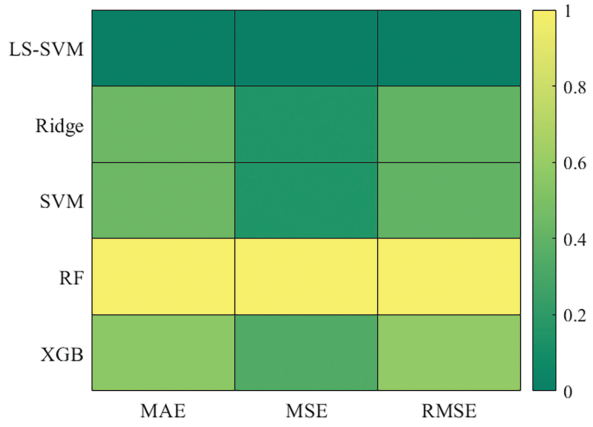


**Figure 8:** Error metrics heatmap (example 2)

### *4.3 Example 3*

We then analyze a linear fourth-order BVP with multi-point boundary conditions to test global consistency. Let's consider the fourth-order linear ODE:

$$y^4(x) = (2\pi)^4 \cos(2\pi x), \quad x \in [0,1]. \tag{20}$$

The multi-point boundary conditions are given by,

$$y(0) = y(1), \quad y''(0) = y''(1),$$
$$y(0.25) + y(0.75) = 0, y'(0.5) = 0.$$

This case involves a linear fourth-order boundary value problem defined over [0, 1] and constrained by a set of nonlocal multi-point boundary conditions. These conditions include values and derivatives at both endpoints and interior points, imposing a global structural requirement that challenges models relying purely on local fitting. The experiment is performed with 70 collocation points, using $\sigma = 2.5$, $\lambda = 10^{-6}$. As presented in Table 7, the LS-SVM model demonstrates excellent performance, producing consistent outputs aligned with expected oscillatory patterns. In contrast, Ridge regression and SVM degenerate to flat approximations, offering no variation across the domain, a clear indication of their inability to represent globally periodic behavior under complex constraints.

The LS-SVM output surface shown in Fig. 9 maintains smooth periodic variation, matching the expected behavior implied by the differential operator. The associated absolute error surface (Fig. 10) remains uniformly low throughout, with peak deviations limited to below $2 \times 10^{-4}$. Table 8 further illustrates the model-wise performance. LS-SVM achieves highly accurate pointwise predictions. Meanwhile, Ridge and SVM methods consistently produce large errors, often close to unity. Ensemble methods such as RF and XGB show improved tracking but are susceptible to minor distortions and phase mismatches, particularly in regions of inflection and extrema.

The LS-SVM prediction aligns precisely with the expected profile and retains structural integrity even at high curvature points, as evidenced in the magnified inset of Fig. 11. It provides a comparative visualization of all methods. On the contrary, Ridge and SVM provide near-constant outputs throughout, failing to satisfy the shape or periodicity requirements. Quantitative metrics in Table 9 confirm this observation: LS-SVM exhibits orders-of-magnitude lower MAE, MSE, and RMSE compared to all baselines. The heatmap in Fig. 12 summarizes the performance landscape, where LS-SVM distinctly dominates. This result highlights the necessity of integrating boundary and operator informed structures in numerical solvers when addressing high-order differential problems governed by nonlocal conditions.

**Table 7:** Solution comparison table (example 3)

| X | Exact | LS_SVM | Ridge | SVM | RF | XGB |
|---|---|---|---|---|---|---|
| 0 | 1 | 0.99981 | 0.01 | 0.015794 | 1.0333 | 0.97972 |
| 0.050505 | 0.95007 | 0.9502 | 0.01 | 0.015794 | 0.85798 | 0.91456 |
| 0.11111 | 0.76604 | 0.766 | 0.01 | 0.015794 | 0.7548 | 0.79537 |
| 0.16162 | 0.52723 | 0.52711 | 0.01 | 0.015794 | 0.55235 | 0.52909 |
| 0.22222 | 0.17365 | 0.17362 | 0.01 | 0.015794 | 0.15007 | 0.2353 |
| 0.77778 | 0.17365 | 0.17362 | 0.01 | 0.015794 | 0.20089 | 0.21897 |
| 0.83838 | 0.52723 | 0.52711 | 0.01 | 0.015794 | 0.52235 | 0.52296 |
| 0.88889 | 0.76604 | 0.766 | 0.01 | 0.015794 | 0.72241 | 0.77626 |
| 0.94949 | 0.95007 | 0.9502 | 0.01 | 0.015794 | 0.93577 | 0.95037 |
| 1 | 1 | 0.99981 | 0.01 | 0.015794 | 0.98222 | 0.99717 |

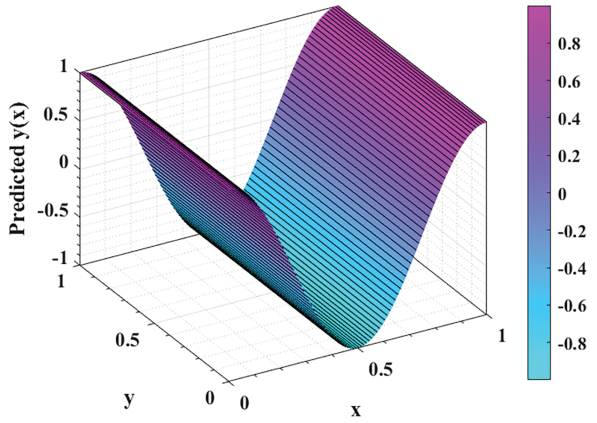


**Figure 9:** 3D surface plot of LSSVM solution (example 3)

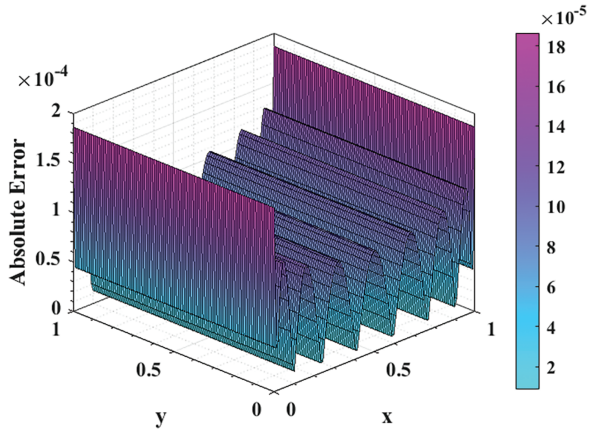


**Figure 10:** Error surface plot of LSSVM (example 3)

**Table 8:** Absolute error across comparing methods (example 3)

| X | Err_LS | Err_Ridge | Err_SVM | Err_RF | Err_XGB |
|---|---|---|---|---|---|
| 0 | 0.00018644 | 0.99 | 0.98421 | 0.033298 | 0.020277 |
| 0.050505 | 0.00012777 | 0.94007 | 0.93428 | 0.092094 | 0.035507 |
| 0.11111 | 4.3558e−05 | 0.75604 | 0.75025 | 0.011244 | 0.029328 |
| 0.16162 | 0.00011538 | 0.51723 | 0.51143 | 0.025122 | 0.001859 |
| 0.22222 | 3.2429e−05 | 0.16365 | 0.15785 | 0.023576 | 0.061656 |
| 0.77778 | 3.2429e−05 | 0.16365 | 0.15785 | 0.027245 | 0.045321 |
| 0.83838 | 0.00011538 | 0.51723 | 0.51143 | 0.004872 | 0.004269 |
| 0.88889 | 4.3558e−05 | 0.75604 | 0.75025 | 0.043634 | 0.010213 |
| 0.94949 | 0.00012777 | 0.94007 | 0.93428 | 0.014305 | 0.000302 |
| 1 | 0.00018644 | 0.99 | 0.98421 | 0.017776 | 0.002831 |

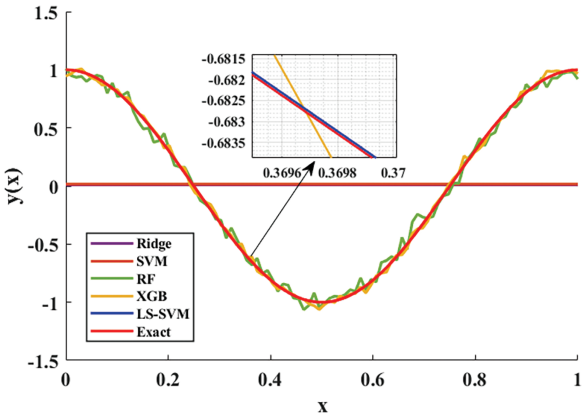


**Figure 11:** Solution comparison across ML methods (example 3)

**Table 9:** Error metrics across comparing methods (example 3)

| | MAE | MSE | RMSE |
|---|---|---|---|
| LS-SVM | 7.3675e−05 | 6.8549e−09 | 8.2794e−05 |
| Ridge | 0.64028 | 0.5049 | 0.71056 |
| SVM | 0.64028 | 0.50493 | 0.71059 |
| RF | 0.040614 | 0.0026972 | 0.051935 |
| XGB | 0.026205 | 0.001094 | 0.033076 |

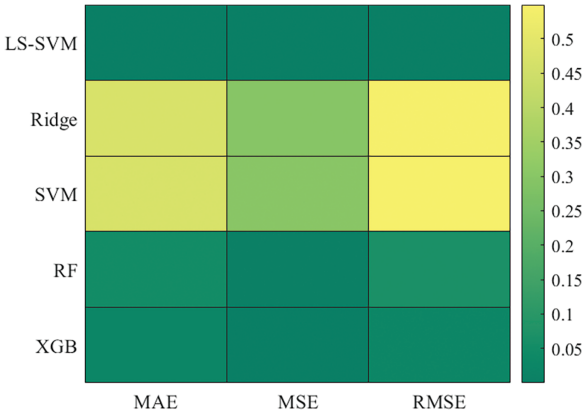


**Figure 12:** Error metrics heatmap (example 3)

### *4.4 Example 4*

Finally, a nonlinear third-order multi-point BVP is solved to evaluate robustness under oscillatory conditions. Let's consider the third-order nonlinear ODE as:

$$y^3(x) + 3x^2 y(x) = 0, x \in [-1, 1]. \tag{21}$$

The multi-point boundary conditions are,

$y(-1) = y(1)$,

$y(0) = 1$,

$y''(0) = -6\pi^2$.

This final case introduces a nonlinear third-order differential equation featuring multi-point boundary conditions distributed over a symmetric domain. The complexity arises not only from the nonlinear source term but also from the requirement to satisfy constraints at both endpoints and interior points, making the solution behavior sharp, oscillatory, and spatially decaying. Here, 80 collocation points are employed, and the parameters are $\sigma = 3.0$, $\lambda = 10^{-6}$. Table 10 shows that among all the considered models, LS-SVM is distinctly more capable of tracking the intricate shape and rapid variation present in the response. While ensemble models like XGBoost and RF manage to approximate the general profile, they exhibit occasional deviations in amplitude and phase. On the other hand, Ridge and SVM regressions fail dramatically, flattening the dynamics and providing outputs that lack structural resemblance to the true behavior.

**Table 10:** Solution comparison table (example 4)

| X | Exact | LS_SVM | Ridge | SVM | RF | XGB |
|---|---|---|---|---|---|---|
| 0 | 1 | 0.99298 | 0.099774 | 0.2224 | 0.95572 | 1.0106 |
| 0.050505 | 0.88657 | 0.89046 | 0.090862 | 0.20395 | 0.85387 | 0.88311 |
| 0.11111 | 0.49387 | 0.49433 | 0.080167 | 0.1818 | 0.5831 | 0.45851 |
| 0.16162 | 0.046355 | 0.043452 | 0.071255 | 0.16335 | −0.06111 | 0.053421 |
| 0.55556 | 0.36722 | 0.36529 | 0.001740 | 0.019417 | 0.33315 | 0.40561 |
| 0.60606 | 0.58265 | 0.58273 | −0.0071715 | 0.000964 | 0.59468 | 0.53302 |
| 0.66667 | 0.64118 | 0.64331 | −0.017866 | −0.021179 | 0.6808 | 0.64032 |
| 0.71717 | 0.53143 | 0.53334 | −0.026778 | −0.039632 | 0.48254 | 0.52564 |
| 0.77778 | 0.27305 | 0.27259 | −0.037473 | −0.061776 | 0.26095 | 0.25849 |
| 0.82828 | 0.02396 | 0.021916 | −0.046385 | −0.080229 | −0.01437 | −0.020573 |

The surface visualization in Fig. 13 highlights the LS-SVM model's competence in resolving the oscillatory decay of the solution. Fig. 14 further emphasizes its reliability, revealing a highly uniform error landscape where deviations remain bounded below $7 \times 10^{-3}$. Error concentrations near boundaries are noticeable yet controlled, hinting at the difficulty imposed by asymmetric derivative conditions. Absolute errors showed in Table 11 confirm that LS-SVM predictions are highly consistent across the domain. Even in regions with abrupt curvature, its performance remains robust. Comparatively, Ridge and SVM accumulate large deviations throughout, while RF introduces noticeable irregularities due to its stochastic nature.

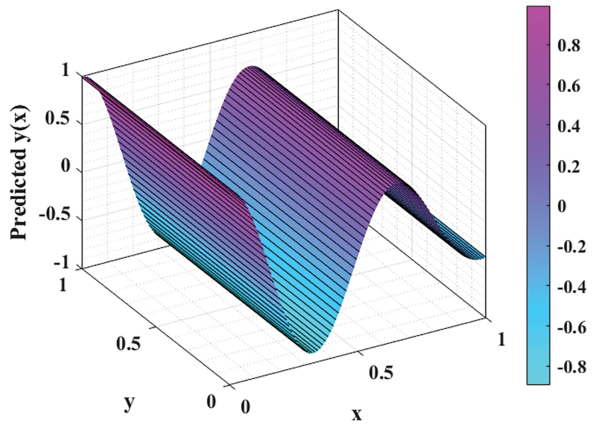


**Figure 13:** 3D surface plot of LSSVM solution (example 4)

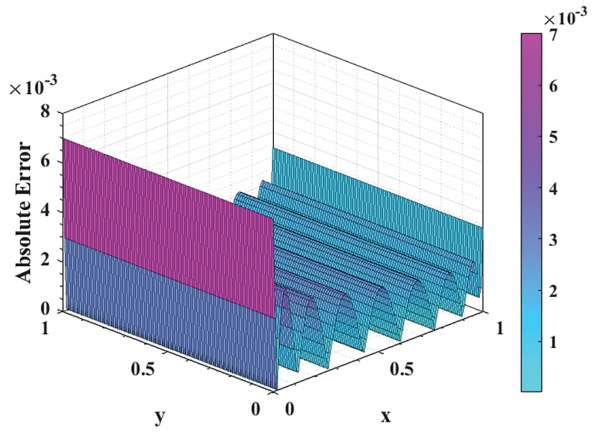


**Figure 14:** Error surface plot of LSSVM (example 4)

**Table 11:** Absolute error across comparing methods (example 4)

| X | Err_LS | Err_Ridge | Err_SVM | Err_RF | Err_XGB |
|---|---|---|---|---|---|
| 0 | 0.007015 | 0.90023 | 0.7776 | 0.044276 | 0.01057 |
| 0.050505 | 0.003891 | 0.79571 | 0.68262 | 0.032696 | 0.003454 |
| 0.11111 | 0.000468 | 0.4137 | 0.31206 | 0.089235 | 0.035354 |
| 0.16162 | 0.002903 | 0.0249 | 0.117 | 0.10747 | 0.007066 |
| 0.55556 | 0.001936 | 0.36548 | 0.3478 | 0.034076 | 0.038389 |
| 0.60606 | 7.7825e−05 | 0.58982 | 0.58168 | 0.012032 | 0.049633 |
| 0.66667 | 0.002127 | 0.65905 | 0.66236 | 0.039617 | 0.000856 |
| 0.71717 | 0.001905 | 0.55821 | 0.57107 | 0.04889 | 0.005788 |
| 0.77778 | 0.000460 | 0.31053 | 0.33483 | 0.012102 | 0.014563 |
| 0.82828 | 0.002044 | 0.070345 | 0.10419 | 0.03833 | 0.044533 |

A broader perspective emerges in Fig. 15, where solution curves from all models are juxtaposed. The LS-SVM output conforms remarkably well to the reference, retaining both the correct amplitude and phase. XGB follows closely, though with small under and overshooting. Meanwhile, Ridge and SVM are visibly misaligned, providing almost linear trends where strong curvature exists. The error metrics in Table 12 provide quantitative confirmation of LS-SVM which achieves significantly lower MAE, MSE, and RMSE values, reinforcing its effectiveness. The final heatmap in Fig. 16 summarizes this performance spectrum, where the LS-SVM model again occupies the optimal zone across all criteria. This example thus reinforces the importance of structure-preserving models in solving nonlinear, high-order systems governed by multi-point conditions.

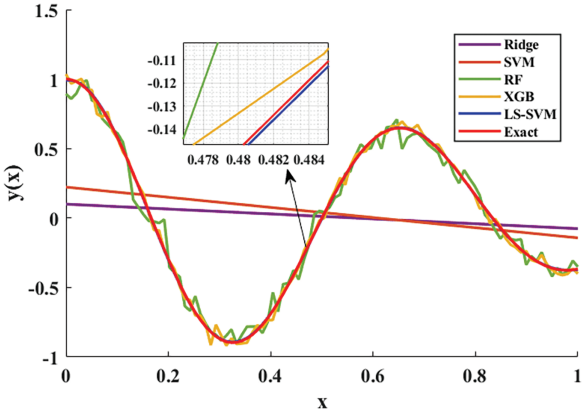


**Figure 15:** Solution comparison across ML methods (example 4)

**Table 12:** Error metrics across comparing methods (example 4)

| | MAE | MSE | RMSE |
|---|---|---|---|
| LS-SVM | 0.0017989 | 4.3955e−06 | 0.002096 |
| Ridge | 0.47211 | 0.29823 | 0.5461 |
| SVM | 0.46947 | 0.30206 | 0.5496 |
| RF | 0.055511 | 0.004783 | 0.06916 |
| XGB | 0.023039 | 0.000881 | 0.029683 |

## 5 Discussion

This study discusses a kernel-based Least Squares Support Vector Machine (LS-SVM) framework as a mesh-independent solver for high-order boundary value problems. The novelty of this approach lies in its seamless integration of machine learning principles with functional approximation theory to address complex differential models including linear or nonlinear, two-point or multi-point without requiring traditional meshing or basis construction. In contrast to conventional solvers, the LS-SVM model constructs solutions by minimizing a regularized functional directly in a reproducing kernel Hilbert space, allowing it to bypass the discretization and integration steps typically needed in finite or boundary element methods.

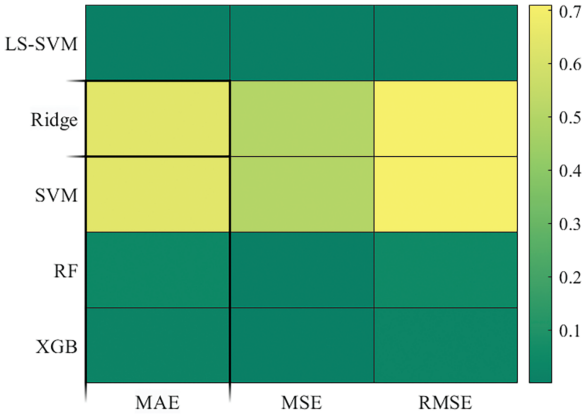


**Figure 16:** Error metrics heatmap (example 4)

The LS-SVM framework proves its adaptability through its application to four structurally distinct BVPs, encompassing both linear and two nonlinear cases. In each instance, the method consistently surpasses traditional regression-based models and ensemble learning algorithms in both predictive precision and qualitative accuracy. This is particularly evident in problems characterized by complex global interactions and abrupt local features, where LS-SVM maintains a high degree of fidelity in capturing the solution behavior. Its reliance on kernel functions effectively circumvents the need for spatial discretization, serving as a built-in mesh-reduction mechanism that delivers accurate solutions without being constrained by geometric intricacies. From a computational perspective, the LS-SVM offers a compelling alternative to conventional solver which is structurally agnostic, efficient, and readily scalable. The absence of meshing or bespoke shape functions makes it especially advantageous for scenarios involving irregular domains, heterogeneous boundary conditions, or nonlinear source terms. Moreover, the framework's capacity to encode high-order differential structures within a machine learning paradigm lays the foundation for integration with data-driven modeling strategies, uncertainty analysis, and real-time digital twin technologies. These features position LS-SVM not merely as a numerical tool, but as a building block for next-generation intelligent solvers that uphold both flexibility and analytical robustness.

## 6 Conclusion

This study has presented a comprehensive and systematic assessment of the Least Squares Support Vector Machine (LS-SVM) for solving high-order boundary value problems (BVPs) encompassing both linear and nonlinear cases. Through comparison with classical and modern machine learning models, including standard SVM, ensemble techniques, and regularized regression, LS-SVM has demonstrated superior performance in terms of solution accuracy, numerical robustness, and error convergence. The kernel-based formulation effectively transforms complex differential operators into a regularized least-squares optimization problem, thereby eliminating the need for meshing or basis-function selection. This property enables LS-SVM to efficiently handle nonlinear source terms, discontinuous boundary conditions, and irregular domains while maintaining smooth and physically consistent approximations even with sparse data. The method also exhibits high computational efficiency due to its closed-form dual solution, resulting in reduced memory requirements and faster

convergence. The discretization-free nature of LS-SVM makes it suitable for integration with real-time data streams and emerging applications such as digital twins, online monitoring, and physics-informed learning. Future research may extend this framework to partial differential equations, coupled multiphysics systems, and inverse problems under uncertainty, establishing LS-SVM as a powerful and flexible tool for next-generation intelligent scientific solvers.

**Acknowledgement:** Not applicable.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception, methodology design, computational implementation, analysis, manuscript drafting, and editing: Bhubaneswari Mishra; conceptualization, supervision, review, and visualization: Snehashish Chakraverty. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The datasets and codes generated or analyzed during the current study are part of the author's ongoing PhD research and are therefore not publicly available due to confidentiality agreements. However, they may be made available from the corresponding author upon reasonable request after the completion of the PhD work.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

1. Chawla MM, Katti CP. Finite difference methods for two-point boundary value problems involving high order differential equations. BIT Numer Math. 1979;19(1):27–33. doi:10.1007/BF01931218.
2. Mohyud-Din ST, Noor MA. Homotopy perturbation method for solving fourth-order boundary value problems. Math Probl Eng. 2007;2007:98602. doi:10.1155/2007/98602.
3. Noor MA, Mohyud-Din ST. Homotopy perturbation method for solving sixth-order boundary value problems. Comput Math Appl. 2008;55(12):2953–72. doi:10.1016/j.camwa.2007.11.026.
4. Ali J, Islam S, Islam S, Zaman G. The solution of multipoint boundary value problems by the optimal homotopy asymptotic method. Comput Math Appl. 2010;59(6):2000–6. doi:10.1016/j.camwa.2009.12.002.
5. Tatari M, Dehghan M. The use of the Adomian decomposition method for solving multipoint boundary value problems. Phys Scr. 2006;73(6):672–6. doi:10.1088/0031-8949/73/6/023.
6. Wazwaz AM. A new algorithm for calculating adomian polynomials for nonlinear operators. Appl Math Comput. 2000;111(1):33–51. doi:10.1016/s0096-3003(99)00063-6.
7. Wazwaz AM. The numerical solution of fifth-order boundary value problems by the decomposition method. J Comput Appl Math. 2001;136(1–2):259–70. doi:10.1016/s0377-0427(00)00618-x.
8. Wazwaz AM. The numerical solution of sixth-order boundary value problems by the modified decomposition method. Appl Math Comput. 2001;118(2–3):311–25. doi:10.1016/s0096-3003(99)00224-6.
9. Wazwaz AM. Approximate solutions to boundary value problems of higher order by the modified decomposition method. Comput Math Appl. 2000;40(6–7):679–91. doi:10.1016/s0898-1221(00)00187-5.
10. Wazwaz AM. The modified decomposition method for analytic treatment of differential equations. Appl Math Comput. 2006;173(1):165–76. doi:10.1016/j.amc.2005.02.048.

11. Aziz I, Siraj-ul-Islam, Nisar M. An efficient numerical algorithm based on Haar wavelet for solving a class of linear and nonlinear nonlocal boundary-value problems. Calcolo. 2016;53(4):621–33. doi:10.1007/s10092-015-0165-9.
12. Shi Z, Li F. Numerical solution of high-order differential equations by using periodized Shannon wavelets. Appl Math Model. 2014;38(7–8):2235–48. doi:10.1016/j.apm.2013.10.030.
13. Doha EH, Bhrawy AH, Hafez RM. A Jacobi-Jacobi dual-Petrov-Galerkin method for third- and fifth-order differential equations. Math Comput Model. 2011;53(9–10):1820–32. doi:10.1016/j.mcm.2011.01.002.
14. Doha EH, Abd-Elhameed WM, Bassuony MA. New algorithms for solving high even-order differential equations using third and fourth Chebyshev-Galerkin methods. J Comput Phys. 2013;236(176):563–79. doi:10.1016/j.jcp.2012.11.009.
15. Doha EH, Bhrawy AH, Hafez RM. On shifted Jacobi spectral method for high-order multi-point boundary value problems. Commun Nonlinear Sci Numer Simul. 2012;17(10):3802–10. doi:10.1016/j.cnsns.2012.02.027.
16. Saadatmandi A, Dehghan M. The use of Sinc-collocation method for solving multi-point boundary value problems. Commun Nonlinear Sci Numer Simul. 2012;17(2):593–601. doi:10.1016/j.cnsns.2011.06.018.
17. Liu C. Multi-task physics-informed learning for partial data inverse boundary value problems for the Nonlinear Schrödinger Equations [dissertation]. Shenyang, China: Northeastern University; 2025.
18. Apakov YP, Mamajonov SM. Boundary value problem for fourth order inhomogeneous equation with variable coefficients. J Math Sci. 2024;284(2):153–65. doi:10.1007/s10958-024-07340-5.
19. Zhou Z, Zhang H, Yang X. The compact difference scheme for the fourth-order nonlocal evolution equation with a weakly singular kernel. Math Methods Appl Sci. 2023;46(5):5422–47. doi:10.1002/mma.8842.
20. Hou M, Han X. The multidimensional function approximation based on constructive wavelet RBF neural network. Appl Soft Comput. 2011;11(2):2173–7. doi:10.1016/j.asoc.2010.07.016.
21. Hou M, Han X. Multivariate numerical approximation using constructive L2(R) RBF neural network. Neural Comput Appl. 2012;21(1):25–34. doi:10.1007/s00521-011-0604-8.
22. Hou M, Han X. Constructive approximation to multivariate function by decay RBF neural network. IEEE Trans Neural Netw. 2010;21(9):1517–23. doi:10.1109/TNN.2010.2055888.
23. Sahoo AK, Klein I. MoRPI-PINN: a physics-informed framework for mobile robot pure inertial navigation. arXiv:2507.18206. 2025.
24. Kumar S, Parc PARC, Sahoo AK, Chakraverty S, Parc PARC. Comparative study of Chebyshev and Legendre polynomial-based neural models for approximating multidimensional poverty for an Indian State. In: Polynomial paradigms. Bristol, UK: Polynomial Paradigms; 2022. p. 13–1–14. doi:10.1088/978-0-7503-5067-9ch13.
25. Wang J, Pang X, Yin F, Yao J. A deep neural network method for solving partial differential equations with complex boundary in groundwater seepage. J Petrol Sci Eng. 2022;209:109880. doi:10.1016/j.petrol.2021.109880.
26. Maslyaev M, Hvatov A, Kalyuzhnaya AV. Partial differential equations discovery with EPDE framework: application for real and synthetic data. J Comput Sci. 2021;53(2197):101345. doi:10.1016/j.jocs.2021.101345.
27. Yang Y, Hou M, Sun H, Zhang T, Weng F, Luo J. Neural network algorithm based on Legendre improved extreme learning machine for solving elliptic partial differential equations. Soft Comput. 2020;24(2):1083–96. doi:10.1007/s00500-019-03944-1.
28. Amilo D, Sadri K, Hincal E, Farman M, Nisar KS. Dual approach artificial neural networks-fractional order operator to enhancing *in vitro* fertilization predictions and success measure. Model Earth Syst Environ. 2025;11(4):281. doi:10.1007/s40808-025-02450-8.
29. Jafarian A, Rezaei R, Khalili Golmankhaneh A. On solving fractional higher-order equations via artificial neural networks. Iran J Sci Technol Trans A Sci. 2022;46(2):535–45. doi:10.1007/s40995-021-01254-6.

30. Zehra A, Farman M, Bonny T, Sambas A, Al Nassan W, Nisar KS, et al. Dynamics of 4D supply chain system with fractal fractional derivatives insight of stability analysis and ANN prediction. Sci Rep. 2025;15(1):30298. doi:10.1038/s41598-025-15706-1.
31. Saneifard R, Jafarian A, Ghalami N, Nia SM. Extended artificial neural networks approach for solving two-dimensional fractional-order Volterra-type integro-differential equations. Inf Sci. 2022;612(4):887–97. doi:10.1016/j.ins.2022.09.017.
32. Jafarian A, Saneifard R. Extended aartificial neural networks approach and fractional volterra integro-differential equations. Int J Indus Math. 2023;15(3):199.
33. Chakraverty S, Mall S. Regression-based weight generation algorithm in neural network for solution of initial and boundary value problems. Neural Comput Appl. 2014;25(3):585–94. doi:10.1007/s00521-013-1526-4.
34. Malek A, Shekari Beidokhti R. Numerical solution for high order differential equations using a hybrid neural network—optimization method. Appl Math Comput. 2006;183(1):260–71. doi:10.1016/j.amc.2006.05.068.
35. Mai-Duy N. Solving high order ordinary differential equations with radial basis function networks. Int J Numer Meth Eng. 2005;62(6):824–52. doi:10.1002/nme.1220.
36. El-Gamel M, Behiry SH, Hashish H. Numerical method for the solution of special nonlinear fourth-order boundary value problems. Appl Math Comput. 2003;145(2–3):717–34. doi:10.1016/s0096-3003(03)00269-8.
37. Nurmuhammad A, Muhammad M, Mori M, Sugihara M. Double exponential transformation in the Sinc-collocation method for a boundary value problem with fourth-order ordinary differential equation. J Comput Appl Math. 2005;182(1):32–50. doi:10.1016/j.cam.2004.09.061.
38. Karunakar P, Chakraverty S. Comparison of solutions of linear and non-linear shallow water wave equations using Homotopy perturbation method. Int J Numer Meth Heat Fluid Flow. 2017;27(9):2015–29. doi:10.1108/hff-09-2016-0329.
39. Jeswal SK, Chakraverty S. Fuzzy eigenvalue problems of structural dynamics using ANN. In: New paradigms in computational modeling and its applications. Amsterdam, The Netherlands: Elsevier; 2021. p. 145–61. doi:10.1016/b978-0-12-822133-4.00010-4.
40. Vapnik VN. The nature of statistical learning theory. 1st ed. Berlin/Heidelberg, Germany: Springer; 1995.
41. Suykens JAK, Vandewalle J. Least squares support vector machine classifiers. Neural Process Lett. 1999;9(3):293–300. doi:10.1023/A:1018628609742.
42. Yang Y, Tan M, Dai Y. An improved CS-LSSVM algorithm-based fault pattern recognition of ship power equipments. PLoS One. 2017;12(2):e0171246. doi:10.1371/journal.pone.0171246.
43. Liu X, Bo L, Luo H. Bearing faults diagnostics based on hybrid LS-SVM and EMD method. Measurement. 2015;59(8):145–66. doi:10.1016/j.measurement.2014.09.037.
44. Yu L, Chen H, Wang S, Lai KK. Evolving least squares support vector machines for stock market trend mining. IEEE Trans Evol Comput. 2009;13(1):87–102. doi:10.1109/TEVC.2008.928176.
45. Jung HC, Kim JS, Heo H. Prediction of building energy consumption using an improved real coded genetic algorithm based least squares support vector machine approach. Energy Build. 2015;90(4):76–84. doi:10.1016/j.enbuild.2014.12.029.
46. Mehrkanoon S, Falck T, Suykens JAK. Approximate solutions to ordinary differential equations using least squares support vector machines. IEEE Trans Neural Netw Learn Syst. 2012;23(9):1356–67. doi:10.1109/TNNLS.2012.2202126.
47. Mehrkanoon S, Suykens JAK. Learning solutions to partial differential equations using LS-SVM. Neurocomputing. 2015;159(1):105–16. doi:10.1016/j.neucom.2015.02.013.
48. Mehrkanoon S, Suykens JAK. LS-SVM approximate solution to linear time varying descriptor systems. Automatica. 2012;48(10):2502–11. doi:10.1016/j.automatica.2012.06.095.

49. Zhang G, Wang S, Wang Y, Liu W. LS-SVM approximate solution for affine nonlinear systems with partially unknown functions. J Ind Manag Optim. 2014;10(2):621–36. doi:10.3934/jimo.2014.10.621.
50. Mishra B, Chakraverty S. Kernel based physics-informed machine learning for approximating CEV model under nonlinear volatility regimes in real-world financial environments. Chaos Soliton Fract. 2025;200(3):117154. doi:10.1016/j.chaos.2025.117154.
51. Mehrkanoon S, Suykens JAK. Deep hybrid neural-kernel networks using random Fourier features. Neurocomputing. 2018;298(8):46–54. doi:10.1016/j.neucom.2017.12.065.
52. Suykens JAK, Gestel TV, Brabanter JD, Moor BD, Vandewalle J. Least squares support vector machines. Singapore: World Scientific; 2002.
53. Kincaid DR, Cheney EW. Numerical analysis—mathematics of scientific computing. Math Comput. 1992;59(199):297. doi:10.2307/2152998.
54. Arfken GB, Weber HJ. Mathematical methods for physicists. 4nd ed. Cambridge, MA, USA: Academic Press; 1995.