Journal of Mathematical Modeling
Vol. 12, No. 4, 2024, pp. 583–602. Research Article

JMM

A new approach to numerical solution of the time-fractional KdV-Burgers equations using least squares support vector regression

Abumoslem Mohammadi[†], **Abolfazl Tari**[*]

Department of Mathematics, Shahed University, Tehran, Iran
Email(s): abumoslem.mohammadi@shahed.ac.ir, tari@shahed.ac.ir

Abstract. The evolution of the waves on shallow water surfaces is described by a mathematical model given by nonlinear KdV and KdV-Burgers equations. These equations have many other applications and have been simulated by classical numerical methods in recent decades. In this paper, we develop a machine learning algorithm for the time-fractional KdV-Burgers equations. The proposed method implements a linearization of the problem and a time reduction by a Crank-Nicolson scheme. The least squares support vector regression (LS-SVR) is proposed to seek the approximate solution in a finite-dimensional polynomial kernel space. The Bernstein polynomials are used as the kernel of the proposed algorithm to handle the homogeneous boundary conditions easily in the framework of the Petrov-Galerkin spectral method. The proposed LS-SVR implements the orthogonal system of Bernstein-dual polynomials in the learning process, which gives quadratic programming in the primal form and provides a linear system of equations in dual variables with sparse positive definite matrices. It is shown that the involving mass and stiffness matrices are sparse. Some new theorems for the introduced basis are provided. Also, numerical results are presented to support the spectral convergence and accuracy of the method.

Keywords: Fractional KdV equation, machine learning, support vector machines, Petrov-Galerkin, least squares support vector regression.
AMS Subject Classification 2010: 35R11, 65M70, 65M22, 76M22.

1 Introduction

Machine learning algorithms are able to learn mathematical models from a set of training data and are used to predict new events that are not experienced before. As a supervised learning algorithm, support vector machines (SVM) are used for classification, regression, and function estimation for the analysis

[*]Corresponding author
Received: 11 February 2024 / Revised: 5 May 2024 / Accepted: 3 June 2024
DOI: 10.22124/jmm.2024.26733.2358

Table 1: A literature review for the classic numerical methods for the time-fractional differential equations on shallow waters. NR stands for *not reported*.

Problem	ref	method/basis	convergence
fractional KdV and KdV-Burgers, 2021	[14]	Collocation, Legendre	spectral
time-fractional KdV-Burgers, 2021	[29]	Petrov-Galerkin, Legendre	spectral
fractional combined KdV–mKdV	[12]	Collocation, RBF	NR
time-fractional KdV–Burgers, 2018	[5]	Petrov-Galerkin, B-Spline	spectral
modified KdV equation, 2022	[16]	Neural networks	NR
PDEs governing the shallow water waves, 2017	[3]	Radial basis functions (RBF)	spectral
KdV equations, 2023	[30]	Neural networks	NR

of big data sets. In recent decades, this technique has been used in various applications including but not limited to document classification in libraries, detection of hand-written manuscripts and classification of satellite data. The SVM for regression, which also known as SVR, has been implemented for other applications like the simulation of the differential equations and optimal control problems. For example, Mehrkanoon, et al. have developed a machine learning algorithm based on SVR for the numerical solution of the ordinary and partial differential equations as well as delay differential equations [20–22]. Also, the method and other neural networks have been implemented on some functional equations in recent works such as the problems on unbounded domains [24], Volterra integral equations and fractional Volterra's population models [26, 27], cognitive decision-making [6] and Fredholm integral equations [25]. In these works, the unknown solution was represented as a weighted summation of some known basis functions, and a residual function was minimized in some least squares sense. So, we refer to these works as LS-SVR [7, 28].

In this paper, we propose a new algorithm based on LS-SVR for the time-fractional KdV-Burgers equations. These equations are used to model the time evolution of the waves in shallow waters on a given spatial surface. They have important applications in acoustic waves, chaos theory, and Fermi-Ulam experiments [15]. For the simple models of these equations with integer derivatives, the exact solution is obtained via some linear transforms as moving solitons in a closed form. For the fractional problem in which the simultaneous change as a local operator is replaced with a memory-recording non-integer derivative, however, the exact solution is only derived as a convergent series solution that may suffer from slow convergence [13]. In order to provide a more general description of the physical problem, the time derivative in classic mathematical modeling is considered in the fractional sense to take into account and keep a record of the history of the simultaneous derivative from the very beginning. Due to these reasons, research on new efficient methods for these problems is still an ongoing effort. For instance, the time-fractional KdV-Burgers equations have been numerically simulated by the Petrov-Galerkin method with the B-spline basis functions [5] and the Legendre polynomials [29]. For other recently published papers implemented on KdV-related problems, see the literature review given in Table 1.

Here, a residual function is associated with the problem by considering a closed-form polynomial approximation of the unknown solution in terms of the Bernstein kernel, and it is minimized using a training dataset by implementing a Petrov-Galerkin technique in a support vector regression framework.

The novelty of this paper is the development of a new algorithm with LS-SVR for the dynamics of the evolution of water waves on shallow surfaces modeled as the time-fractional KdV-Burgers equations.

The fractional derivatives are discretized on the time steps using an L1 approximation and the nonlinear part of the problem is replaced with a linear approximation at each time step. The LS-SVR provides the unknown coefficients by minimizing the residual function with some training data.

The rest of this paper is organized as follows: Some preliminaries concerning the KdV-Burgers problem, fractional derivatives, and support vector machines are provided in Section 2. In Section 3, we present an LS-SVR for the time-fractional KdV-Burgers equations. Some numerical results are reported to support the efficiency of the proposed method in Section 4. Finally, some concluding remarks are provided at the end.

2 Preliminaries

In this section, we provide some basic definitions and concepts that are needed in the rest of the paper.

2.1 Time-fractional KdV-Burgers equation

The time-fractional KdV-Burgers equation given by [29]

$$\partial_t^\alpha u + \mu \partial_x^3 u + 2u\partial_x u - \gamma \partial_x^2 u = f, \quad (x,t) \in \Omega \times I, \tag{1}$$

equipped with the initial and boundary conditions

$$u(x,0) = g(x), \tag{2}$$

$$u(0,t) = u(1,t) = \partial_x u(1,t) = 0, \quad t \in I, \tag{3}$$

provides a mathematical description in terms of the nonlinear dispersive wave equation of the waves on shallow water surfaces and facilitates the prediction of the underlying dynamics. For the sake of the simplicity of the presentation, we assumed the homogeneous boundary conditions, otherwise by introducing a new wave variable the problem is converted to a similar problem with a new right hand side and homogeneous boundary conditions. Here, $u(x,t)$ is for the amplitude of the wave and illustrates the evolution of solitary waves on the water surface on the spatial domain $\Omega \subset \mathbb{R}$ and temporal domain $I = (0,T)$, μ and γ represent the dispersivity and diffusion coefficients, respectively. It is worth noting that in real applications $\Omega \subset \mathbb{R}^2$ and by angular symmetry assumptions, it is restricted to 1D. In (1), $\partial_t^\alpha u$ stands for the fractional derivative in the Caputo sense with order $0 < \alpha < 1$. The Caputo derivative is defined as

$${}_aD_t^{(\alpha)} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^{(n)}(s)}{(t-s)^{\alpha-n+1}} ds, \tag{4}$$

in which $n-1 \leq \alpha < n$ and $n \in \mathbb{N}$. The fractional derivative in (4) gives a weighted summation of the simultaneous changes in the function over the time domain from the beginning to the current time with the weight function $w(t,s) = \frac{1}{(t-s)^{\alpha-n+1}}$. Some researchers have investigated other weights, however the Caputo definition incorporates the initial conditions of its definition, which means that it takes into account the initial state of the system, making the modeling and solution process more intuitive [1, 2]. For the singularity issues at the initial time, throughout this work, we assume g to be smooth [10].

2.2 LS-SVR

As supervised machine learning algorithms, SVM and SVR are used to classify the data and predict the patterns in a large dataset, respectively, by an optimization problem with some inequality constraints. The least squares versions of LS-SVM and LS-SVR transform the inequalities to equalities and the resulting problem is then simply transformed into an equivalent system of linear equations. For more details, see [20, 24, 25].

Assume that a known data set (x_i, y_i), $i = 1, \ldots, N$, x_i's in R^{n_d} are independent variables, and real numbers y_i's depend on x_i's. Finding regression formulae for this data has great importance in many applied engineering problems. The LS-SVR provides an approximation in the form of $y(x) = w^T \phi(x) + b$ where ϕ_i's are known basis functions, $b \in \mathbb{R}$ is the bias, and w_i's and b are to be determined by the following primal problem:

$$\begin{aligned} &\min \frac{1}{2} w^T w + \frac{\gamma}{2} e^T e, \\ &\text{s.t.} \quad y_i = w^T \phi(x_i) + b + e_i, \quad i = 1, \ldots, N, \end{aligned} \tag{5}$$

where $w = [w_1, \ldots, w_N]^T$, $\phi = [\phi_1, \ldots, \phi_N]^T$ and $\gamma \in \mathbb{R}^+$ is the tuning parameter. Also, e_i as the primal variable, represents the error of the model at x_i (margin error). By using the Lagrangian as the sum of the objective function and a linear combination of the constraints with the dual variables α_i, it is easily verified that this quadratic programming (5) is converted to a dual problem with an equivalent linear system as follows [20, 25].

Theorem 1. *The primal problem given as a quadratic programming (5) is equivalent to*

$$\begin{bmatrix} \Omega + \frac{1}{\gamma} I_N & \mathbf{1}_N \\ \mathbf{1}_N^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix},$$

where $\Omega_{i,j} = \phi^T(x_i)\phi(x_j)$ is a positive definite matrix, I_N is the identity matrix, $\mathbf{1}_N^T = [1, \ldots, 1] \in \mathbb{R}^N$, $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ and $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$ is vector of the Lagrange coefficients. Now, the solution is given by

$$y(x) = \sum_{j=1}^{N} \alpha_j K(x, x_j) + b, \tag{6}$$

with the kernel function $K(x, x_j) = \phi^T(x)\phi(x_j)$.

First note that the kernel is symmetric, i.e. $K(x, y) = K(y, x)$. Also,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} K(x_i, x_j) c_i c_j = \sum_{i=1}^{N} \sum_{j=1}^{N} \phi^T(x_i)\phi(x_j) c_i c_j = \sum_{i=1}^{N} \phi^T(x_i) c_i \sum_{i=1}^{N} \phi(x_i) c_i \geq 0,$$

so it is a positive definite kernel.

So, the Mercer theorem guarantees the expansion of the kernel in terms of an orthonormal basis. It is used as a tool for the categorization of the symmetric positive definite kernels. For Mercer's theorem on the orthogonal polynomial kernels, see [23] and the references therein.

2.3 Kernel mapping and the basis functions

As we have seen in the previous section, the LS-SVR algorithm uses kernel mapping to represent a regression formula (6) for a given data set. For our purposes the dependent values y_i's are not known and we have only a partial differential equation governing the physical problem, so we use an alternative approach to provide the training data for the algorithm. This is well described in details in the next section. Since we use the polynomial kernel, for now, we present the definitions and some basic properties of a simple basis for the vector space of the finite-dimensional spaces of polynomials.

Let $\mathscr{P}_M$ be the polynomial of degree at most M on the unit interval $[0,1]$. For this set, we have the following Bernstein polynomials as a basis set:

$$B_{i,M}(x) = \binom{M}{i} x^i (1-x)^{M-i}, \quad 0 \le i \le M. \tag{7}$$

They have many advantages. For example, the easy computation and implementation of differential and integral functional equations. Also, they facilitate handling the homogeneous boundary conditions. However, they are not orthogonal and a corresponding set of polynomials is required to provide an biorthogonal system.

Lemma 1 ([8, 11]). *Consider the functions*

$$B^{\star}_{i,M}(x) = \sum_{j=0}^{M} d_{i,j} B_{j,M}(x), \quad 0 \le i \le M, \tag{8}$$

with

$$d_{i,j} = \frac{(-1)^{i+j}}{(b-a)\binom{M}{i}\binom{M}{j}} \sum_{r=0}^{min(i,j)} (2r+1) \binom{M+r+1}{M-i} \binom{M-r}{M-i} \binom{M+r+1}{M-j} \binom{M-r}{M-j}.$$

Then, we have the biorthogonal system

$$\int_a^b B_{i,M}(x) B^{\star}_{j,M}(x) dx = \delta_{ij}, \quad \forall i,j = 0,\ldots,M. \tag{9}$$

This makes the computations easier in the inner products arising in the variational formulation of the problem. As previously mentioned, Bernstein polynomials are easy to compute and have simple relations in boundaries that facilitate the implementation of boundary conditions in our algorithm by noticing the behavior at the boundaries

$$\begin{aligned} B_{i,M}(x) &= 0, \quad i = 1,\ldots,M-1, \\ B'_{i,M}(x) &= 0, \quad i = 2,\ldots,M-2, \end{aligned} \tag{10}$$

for $x = 0, 1$. Moreover, for these functions we have [4]

$$\int_0^1 B_{i,M}(x) dx \quad = \tfrac{1}{M+1}, \quad i = 0,\ldots,M. \tag{11}$$

3 A LS-SVR algorithm

In this section, we present the function spaces for the trial and test functions, also a variational form of the problem (1)-(3). We then introduce a supervised machine learning method for the given problem to train a function in the form of a polynomial kernel at each time step.

3.1 Time discretization

Consider the time-fractional KdV equation with the given initial and boundary conditions (1)-(3), which can be rewritten as

$$\begin{aligned} &\partial_t^\alpha u + \mathscr{L}u + \mathscr{N}u = f, \\ &\mathscr{L}u = \mu\partial_x^3 u - \gamma\partial_x^2 u, \quad \mathscr{N}(u) = 2u\partial_x u, \end{aligned} \tag{12}$$

in which the operator $\mathscr{L}$ is responsible for the spatial derivatives and $\mathscr{N}$ represents the nonlinear part of the equation. The initial condition gives $u = u^0$ at $t = t_0$. In the following, we present a quasi-linearization with a Bernstein Petrov-Galerkin LS-SVR to obtain an approximation for the solution at the next time steps $u^{n+1}, n = 0, 1, \ldots$. First, we write (12) at $t = t_{n+1}$ with the Crank-Nicolson scheme

$$\partial_t^\alpha u|_{t=t_{n+1}} + \theta(\mathscr{L}(u^{n+1}) + \mathscr{N}(u^{n+1}) - f^{n+1}) + (1-\theta)(\mathscr{L}(u^n) + \mathscr{N}(u^n) - f^n) = 0, \tag{13}$$

with $t_n = n\Delta t$ for a time step Δt. The time-fractional derivative $\partial_t^\alpha u$ is discretized at the time level $t = t_{n+1}$ as [17]

$$\begin{aligned} \partial_t^\alpha u|_{t=t_{n+1}} &= \frac{1}{\Gamma(1-\alpha)} \int_0^{t_{n+1}} \frac{\partial_s u(x,s)}{(t_{n+1}-s)^\alpha} ds \approx \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{n} \int_{t_j}^{t_{j+1}} \frac{u^{j+1}-u^j}{\Delta t_j (t_{n+1}-s)^\alpha} ds \\ &= \sum_{j=0}^{n} (u^{j+1}-u^j)\xi_{j,n} = \mu_{n+1}u^{n+1} + \sum_{j=0}^{n} \mu_j u^j, \end{aligned} \tag{14}$$

with $\xi_{j,n} = \frac{1}{\Gamma(1-\alpha)} \int_{t_j}^{t_{j+1}} \frac{1}{\Delta t_j (t_{n+1}-s)^\alpha} ds$, $\mu_{n+1} = \xi_{n,n}$, $\mu_j = \xi_{j-1,n} - \xi_{j,n}$, for $j = 0, \cdots, n$. For the sake of simplicity in this notation, we set $\xi_{-1,n} = 0$. Let us denote this approximation as $L_t^\alpha u(x, t_{n+1}) = \mu_{n+1}u^{n+1} + \sum_{j=0}^{n} \mu_j u^j$ and the error as $r_\tau^{n+1} = \partial_t^\alpha u|_{t=t_{n+1}} - L_t^\alpha u(x, t_{n+1})$. This is known as L1 approximation and can be further simplified by taking into account a uniform partition. However, to take advantage of an adaptive method, we use non-uniform meshes. The error of this approximation is given by $|r_\tau^{n+1}| \le c_u \tau^{2-\alpha}$ [17]. Using L1 approximation (14) in the problem (13), we get

$$\begin{aligned} &\mu_{n+1}u^{n+1} + \sum_{j=0}^{n} \mu_j u^j + \theta(\mathscr{L}(u^{n+1}) + \mathscr{N}(u^{n+1}) - f^{n+1}) \\ &\quad + (1-\theta)(\mathscr{L}(u^n) + \mathscr{N}(u^n) - f^n) = 0, \qquad x \in \Omega, \end{aligned} \tag{15}$$

that may be written in a simple form,

$$\mu_{n+1}u^{n+1} + \theta(\mathscr{L}(u^{n+1}) + \mathscr{N}(u^{n+1})) = \tilde{F}^{n+1}, \tag{16}$$

with $\tilde{F}^{n+1} = \theta f^{n+1} - (1-\theta)(\mathscr{L}(u^n) + \mathscr{N}(u^n) - f^n) - \sum_{j=0}^{n} \mu_j u^j$.

3.2 Quasi-linearization

Now, we use the following result for the linearization of $\mathscr{N}(u^{n+1})$ in the discretized equation (16). Suppose $\mathscr{N}(u) = 2uu_x$ as a function of u and u_x. Then, Taylor expansion about the points (u^n, u_x^n) gives

$$uu_x = u^n u_x^n + \frac{\partial \mathscr{N}}{\partial u}(u^n, u_x^n)(u - u^n) + \frac{\partial \mathscr{N}}{\partial u_x}(u^n, u_x^n)(u_x - u_x^n) + \cdots.$$

This immediately gives us a simple linear relation as

$$u^{n+1}u_x^{n+1} \approx u^{n+1}u_x^n + u^n u_x^{n+1} - u^n u_x^n,$$

in which, the upper bound of quasi-linearization error is $\left\|u^{n+1} - u^n\right\|_\infty \left\|u_x^{n+1} - u_x^n\right\|_\infty$, according to Taylor's reminder theorem. So by (16), we have

$$\mu_{n+1}u^{n+1} + \theta(\mathscr{L}(u^{n+1}) + 2u^{n+1}u_x^n + 2u^n u_x^{n+1}) = F^{n+1}, \quad x \in \Omega, \tag{17}$$

with $F^{n+1} = \tilde{F}^{n+1} + 2\theta u^n u_x^n$. The boundary conditions are derived from (3)

$$\begin{aligned} &u^{n+1}(x) = 0, \quad x \in \partial\Omega, \\ &\frac{d}{dx}u^{n+1}(1) = 0. \end{aligned} \tag{18}$$

The solution process starts with $u^0 = g(x)$ given by the initial condition (2).

3.3 Basis functions of the Petrov-Galerkin LS-SVR for (1)

Let $M, N \in \mathbb{N}$ stand for the number of training points in the spatial and the number of time steps in the temporal domain, respectively. To select a suitable basis for approximating the unknown functions in the spatial dimension by (7), set $\phi_i(x) = B_{i,M}(x), i = 0, \cdots, M$. These functions form a basis for $\mathscr{P}_M$, the vector space of polynomials with the degree at most M. Removing some elements from a linearly independent set is again a linearly independent set. So with a simple linear algebra argument, we get the following result.

Proposition 1. *The subset* $\{\phi_i(x), i = 1, \ldots, M-2\}$ *forms a basis for the vector space*

$$V_M^0 = \{p \in \mathscr{P}_M : p(0) = 0,\ p(1) = 0,\ p'(1) = 0\}. \tag{19}$$

This is the space in which the approximate solution of the problem is sought by an SVR training process.

On the other hand, we define the test space as

$$W_M^0 = \{p \in \mathscr{P}_M : p(0) = 0, p(1) = 0,\ p'(0) = 0\},$$

that is used for the least squares projection. To introduce a basis for this space, let $\tilde{\psi}_i(x) = B_{i,M}^\star(x)$ given in (8) for $i = 0, 1, \ldots, M$, and consider the coefficients in

$$\psi_i(x) = \sum_{k=i}^{i+3} a_{i,k}\tilde{\psi}_k(x), \tag{20}$$

with $a_{i,i} = 1$ such that $\psi_i(0) = \psi_i(1) = \psi_i'(0) = 0$. Since $\dim \mathscr{P}_M = M+1$, we have $\dim W_M^0 = M-2$. So we choose $\psi_i(x), i = 0, 1, \ldots, M-3$ as a basis for W_M^0. This gives a system of linear equations with three unknowns $a_{i,i+1},\ a_{i,i+2},\ a_{i,i+3}$ which are determined by solving the system before any implementation.

3.4 LS-SVR for the time-fractional KdV-Burgers equation

Before presenting the details of the SVR for the problem (17)-(18), we first provide the main ideas of the method for a general linear functional equation $\mathscr{L}u = f$ with a given source function f and some homogeneous boundary conditions $\mathscr{B}u = 0$. Following the works of [24, 25], for the numerical simulation, we trained a solution in the ansatz form

$$u = \sum_{j=0}^{M} \alpha_j K(x, x_j), \tag{21}$$

with the kernel function K. Then, we use the SVR to find the solution as $u = w^T \phi$ by solving

$$\begin{aligned} &\min \frac{1}{2} w^T w + \frac{\gamma}{2} e^T e \\ &\text{s.t. } (w^T \mathscr{L}\phi(x), \psi_i) = (f(x), \psi_i) + e_i, \quad i = 0, \dots, M, \end{aligned} \tag{22}$$

in which the functions ϕ_i are assumed to satisfy the homogeneous boundary conditions. In the constraints of (22), we have used the linearity of the operator and an biorthogonal projection. Then, in a similar argument as in Section 2.2, this problem is converted to a dual form as a linear system of algebraic equations. The quadratic programming (22) is easily written in terms of an equivalent linear system.

Theorem 2. *The dual variables α_i's in (21) satisfy*

$$(W + \frac{1}{\gamma} I_M)\boldsymbol{\alpha} = \boldsymbol{b}, \tag{23}$$

with

$$\begin{aligned} &b_i = (f, \psi_i), \quad i = 0, \dots, M, \\ &W_{k,i} = \sum_{j=0}^{M} (\mathscr{L}\phi_j, \psi_k)(\mathscr{L}\phi_j, \psi_i), \qquad k, i = 0, \dots, M. \end{aligned} \tag{24}$$

Proof. The Lagrangian of (22) is

$$L = \frac{1}{2} w^T w + \frac{\gamma}{2} e^T e - \sum_{i=0}^{M} \alpha_i ((w^T \mathscr{L}\phi, \psi_i) - (f, \psi_i) - e_i).$$

By the optimality conditions, we get

$$\frac{\partial L}{\partial w_k} = 0 \Rightarrow w_k = \sum_{i=0}^{M} \alpha_i (\mathscr{L}\phi_k, \psi_i), \tag{25}$$

$$\frac{\partial L}{\partial e_k} = 0 \Rightarrow \gamma e_k + \alpha_k = 0, \tag{26}$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \Rightarrow (w^T \mathscr{L}\phi, \psi_k) - (f, \psi_k) - e_k = 0, \tag{27}$$

for $k = 0, \ldots, M$. So, by (26) and (27), we have

$$\sum_{j=0}^{M} w_j(\mathscr{L}\phi_j, \psi_k) + \frac{1}{\gamma}\alpha_k = (f, \psi_k),$$

and by (25), we get

$$\sum_{j=0}^{M}\sum_{i=0}^{M} \alpha_i(\mathscr{L}\phi_j, \psi_i)(\mathscr{L}\phi_j, \psi_k) + \frac{1}{\gamma}\alpha_k = (f, \psi_k),$$

Using the Kronecker notation, this is written as

$$\sum_{i=0}^{M} \alpha_i(\sum_{j=0}^{M}(\mathscr{L}\phi_j, \psi_k)(\mathscr{L}\phi_j, \psi_i) + \frac{1}{\gamma}\alpha_k\delta_{ki}) = (f, \psi_k),$$

which is the desired result. □

Now, we turn to the main problem. The variational formulation of (17)-(18) is as follows

$$\mu_{n+1}(u^{n+1}, v) + \theta(\mu(\partial_x^3 u^{n+1}, v) - \gamma(\partial_x^2 u^{n+1}, v) + 2(u^{n+1}u_x^n, v) + 2(u^n u_x^{n+1}, v)) = (F^{n+1}, v),$$

for all $v \in L^2(\Omega)$. The initial condition $u^0 = g$ in F^{n+1} which is assumed as a smooth function, so the non-homogeneous problem avoids showing singularity at the initial condition [10].

The LS-SVR method in a variational form is to find the approximate solution in the vector space (19) at time step t_{n+1} as

$$u_M^{n+1} = \sum_{j=1}^{M-2} w_j^{n+1}\phi_j(x) = \boldsymbol{w}^T\boldsymbol{\phi}, \tag{28}$$

with the objective functions

$$\min \frac{1}{2}w^T w + \frac{\gamma}{2}e^T e$$

with undetermined weights $w_j := w_j^{n+1}$ in V_M^0 such that

$$\begin{aligned}\mu_{n+1}(u_M^{n+1}, v_M) + \theta(\mu(\partial_x^3 u_M^{n+1}, v_M) - \gamma(\partial_x^2 u^{n+1}, v_M) + 2(u^{n+1}u_x^n, v_M) + 2(u^n u_x^{n+1}, v_M)) \\ = (F^{n+1}, v_M) + e_M\end{aligned} \tag{29}$$

for all $v_M \in W_M^0$.

Remark 1. *For a basis as $v_M = \phi_i, i = 1, \ldots, M-2$, the linear system (29) is equivalent to the Galerkin LS-SVR, while for Dirac delta functions $v_M = \delta(x - x_i)$, with a partition of the interval $a < x_i < b$, we get the collocation LS-SVR. To avoid the ill-conditioning and dense matrices (with high computational cost), we use an alternative Petrov-Galerkin LS-SVR approach: $v_M = \psi_i, i = 0, \ldots, M-2$ given by (20) to make use of the biorthogonality of the Bernstein and the dual Bernstein polynomials.*

Substituting $v_M = \psi_i$ in (29), implies

$$A\boldsymbol{w}^{n+1} = \boldsymbol{f}^{n+1}, \qquad n = 0, 1, \ldots, \tag{30}$$

in which $A=[a_{i,j}]$, $i=0,\ldots,M-3$, $j=1,\ldots,M-2$ *is given by*

$$A=\mu_{n+1}R+\theta(-\mu Q+\gamma P+2G).$$

The method requires to compute of the following mass and stiffness matrices

$$R_{i,j}=(\phi_j,\psi_i),\ S_{i,j}=(\phi_j',\psi_i),$$
$$P_{i,j}=-(\phi_j',\psi_i'),\ Q_{i,j}=-(\phi_j'',\psi_i'), \tag{31}$$
$$G_{i,j}=(u_x^n\phi_j+u^n\phi_j',\psi_i). \tag{32}$$

Here, $R,S,P,$ *and* Q *are* $(2,1),(3,2),(4,3),$ *and* $(5,4)$*-band matrices, respectively. An* (m,n)*-band matrix is a sparse matrix whose nonzero entries are confined to a diagonal band, comprising the main diagonal and m lower and n upper diagonals.*

Lemma 2. *The mass matrix R has a band structure whose nonzero diagonals are given by*

$$R_{i,j}=\begin{cases}1, & i-j=-1,\\ a_{i,i+1}, & i-j=0,\\ a_{i,i+2}, & i-j=1,\\ a_{i,i+3}, & i-j=2,\end{cases}$$

for $i=0,\ldots,M-3$ *and* $j=1,\ldots,M-2$.

Proof. By Definition (7) and relation (20), we have

$$\begin{aligned}R_{i,j}=\ &(\phi_j,\psi_i)=\int_0^1 B_{j,M}(x)(\tilde{\psi}_i(x)+a_{i,i+1}\tilde{\psi}_{i+1}(x)+a_{i,i+2}\tilde{\psi}_{i+2}+a_{i,i+3}\tilde{\psi}_{i+3})dx\\ =\ &\delta_{i,j}+a_{i,i+1}\delta_{i+1,j}+a_{i,i+2}\delta_{i+2,j}+a_{i,i+3}\delta_{i+3,j},\end{aligned}$$

which gives the desired result noting that $i=0,\ldots,M-3$ and $j=1,\ldots,M-2$. □

To have an explicit representation of the entries of the stiffness matrix, we need the following result from [8].

Lemma 3. *The derivatives for the polynomials (20) are represented as*

$$\begin{aligned}\psi_i'(x)=\ &\alpha_{i,0}\tilde{\psi}_0(x)+(1-\delta_{i,1})i\tilde{\psi}_{i-1}(x)+(1-\delta_{i,0})(1-\delta_{i,M})(M-2i)\tilde{\psi}_i(x)\\ &-(1-\delta_{i,M-1})(M-i)\tilde{\psi}_{i+1}(x)-\alpha_{M-i,0}\tilde{\psi}_M(x),\end{aligned} \tag{33}$$

in which $\alpha_{i,0}:=-(-1)^i(M+1)\binom{M+1}{i+1}+M\delta_{i,0}+\delta_{i,1}$ *for* $0\le i\le M$, *with the convention* $\tilde{\psi}_i\equiv 0$ *for* $i<0$ *and* $i>M$.

The following lemma presents the structure of the stiffness matrix associated with the selected functions for trial and test spaces.

Lemma 4. *The stiffness matrix* $P=[p_{i,j},\ i=1,\ldots,M-2,\ j=0,\ldots,M-3]$ *is a (4,3)-band matrix whose entries are zero for* $|i-j|>4$.

Proof. First, we note that the derivatives of the basis functions are expressed as (see e.g., Theorem 2 in [9])

$$\phi_j'(x) = \sum_{k=j-1}^{j+1} b_{j,k}\phi_k(x),$$

with $b_{j,j-1} = N-j+1$, $b_{j,j} = -(N-2j)$, $b_{j,j+1} = -(j+1)$. Then

$$P_{i,j} = -(\phi_j', \psi_i') = -\psi_i(x) + a_i\psi_{i+1}(x) + b_i\psi_{i+2}(x) + c_i\psi_{i+3}(x).$$

From (20), we have

$$\psi_i' = \sum_{r=i}^{i+3} a_{i,r}\tilde{\psi}_r' = \sum_{r=i}^{i+3} a_{i,r}(\alpha_{r,0}\tilde{\psi}_0 + \sum_{k=r-1}^{r+1} c_{r,k}\tilde{\psi}_k - \alpha_{M-i,0}\tilde{\psi}_M).$$

And the biorthogonality (9), yields

$$\begin{aligned}
P_{i,j} &= \sum_{r=i}^{i+3} a_{i,r}(\alpha_{r,0} < \phi_j', \tilde{\psi}_0 > + \sum_{k=r-1}^{r+1} c_{r,k} < \phi_j', \tilde{\psi}_k > - \alpha_{M-i,0} < \phi_j', \tilde{\psi}_M >) \\
&= \sum_{r=i}^{i+3} a_{i,r}(\alpha_{r,0} \sum_{s=j-1}^{j+1} b_{j,s} < \phi_j, \tilde{\psi}_0 > + \sum_{k=r-1}^{r+1} c_{r,k} \sum_{s=j-1}^{j+1} b_{j,s} < \phi_j, \tilde{\psi}_k > - \alpha_{M-i,0} \sum_{s=j-1}^{j+1} b_{j,s} < \phi_j, \tilde{\psi}_M >) \\
&= \sum_{r=i}^{i+3} a_{i,r}(\alpha_{r,0} \sum_{s=j-1}^{j+1} b_{j,s}\delta_{j,0} + \sum_{k=r-1}^{r+1} c_{r,k} \sum_{s=j-1}^{j+1} b_{j,s}\delta_{j,k} - \alpha_{M-i,0} \sum_{s=j-1}^{j+1} b_{j,s}\delta_{j,M}),
\end{aligned}$$

in which δ denotes the Kronecker delta function. It is easy to see that this expression vanishes when $|i-j| > 4$. □

Other matrices in the formulation of the Petrov-Galerkin LS-SVR are obtained with a similar argument as in the proof of Lemmas 2 and 4.

It is seen that the proposed algorithm for the fractional nonlinear KdV-Burgers equation leads to the linear system (29) and its dual form is written in a matrix form as (30) at any time step to obtain the weights w's for the unknown solution u^{n+1} starting with $u^0 = g$, which is given by the initial condition.

Now, we discuss the error analysis of the proposed algorithm for the time-fractional KdV-Burgers equation in the weighted L2-norm. Let u_M^{n+1} be the approximate solution (28) of the problem (1))-(3), in which the exact solution is denoted by $u(x,t)$. Let $\tilde{A} = [w_1^{n+1}, \ldots, w_{M-2}^{n+1}]$ and $u_M^{n+1} = \phi^T\tilde{A}$ be the weights and the ansatz which are updated in the training process given in Section 3.

Let us denote the best approximation of u at $t = t_{n+1}$ by $\bar{u}(x, t_{n+1})$ in V_M^0 (19). Then,

$$\|u - u_M^{n+1}\|_2 \leq \|u - \bar{u}\|_2 + \|\bar{u} - u_M^{n+1}\|_2. \tag{34}$$

Let u have a continuous kth derivative, and let w_p be the modulus of continuity. An error bound for the first term is given by $\|u - \bar{u}\|_2 \leq C_k w_k(M^{-1})M^{-k}$ [18]. For the second term in (34), we proceed as follows

$$\|\bar{u} - u_M^{n+1}\|_2 = \|\phi^T\bar{A} - \phi^T\tilde{A}\|_2 \leq \|\phi\|_2\|\bar{A} - \tilde{A}\|_2.$$

On the other hand,

$$\|\phi\|_2^2 = \sum_{i=1}^{M-2} \int_0^1 \phi_i^2(x)dx = \sum_{i=1}^{M-2} \frac{\binom{M}{i}^2}{\binom{2M}{2i}} \int_0^1 \binom{2M}{2i} x^{2i}(1-x)^{2M-2i}dx$$
$$= \sum_{i=1}^{M-2} \frac{\binom{M}{i}^2}{\binom{2M}{2i}} \int_0^1 B_{2i,2M}(x)dx = \frac{1}{2M+1} \sum_{i=1}^{M-2} \frac{\binom{M}{i}^2}{\binom{2M}{2i}},$$

in which we have used the definition of the Bernstein polynomials (7) and the integral property (11). So, we get

$$\|\bar{u} - u_M^{n+1}\|_2 \leq (\frac{1}{2M+1} \sum_{i=1}^{M-2} \frac{\binom{M}{i}^2}{\binom{2M}{2i}})^{\frac{1}{2}} \|\bar{A} - \tilde{A}\|_2. \tag{35}$$

Now, from (34) and (35), we have

$$\|u - u_M^{n+1}\|_2 \leq C_k w_k(M^{-1})M^{-k} + C_M \|\bar{A} - \tilde{A}\|_2,$$

with $C_M = (\frac{1}{2M+1} \sum_{i=1}^{M-2} \frac{\binom{M}{i}^2}{\binom{2M}{2i}})^{\frac{1}{2}}$.

In the next section, some examples are considered to show the efficiency of the proposed machine learning algorithm, and the results are compared to some other methods in the literature. As we will see, it is verified that the order of convergence in time is $O(\tau^{2-\alpha})$ and the convergence in space exhibits a spectral behavior.

4 Numerical experiments

Here, we consider some test problems to illustrate the convergence of the proposed algorithm in time and space, discuss the structure of the involved matrices and make some comparisons with existing numerical results in the literature.

We first note that the matrices R, S, P, and Q in (31) have sparse and banded structures as proved in Lemma 2, 4. In Figure 1, the sparsity of the involving matrices (31) in our proposed method is illustrated. These matrices are visualized by *matrixplot* in Maple with $M = 20$ as 18×18 matrices.

The numerical errors at the final time T are computed using

$$L_\infty := \max_{x \in \Omega} |U(x,T) - u_M^N(x)| \approx \max_{0 \leq i \leq \mathscr{M}} |U(x_i,T) - u_M^N(x_i)|, \tag{36}$$

$$L_2 := (\int_\Omega |U(x,T) - u_M^N(x)|^2 dx)^{1/2} \approx (\frac{1}{\mathscr{M}} \sum_{i=1}^{\mathscr{M}} |U(x_i,T) - u_M^N(x_i)|^2)^{1/2}, \tag{37}$$

with $x_i = \frac{i}{\mathscr{M}}$, $i = 0, \ldots, \mathscr{M}$ and we set the number of test points, $\mathscr{M} = 100$ points. A numerical approximation of the order of convergence in time which is reported in numerical tables is obtained as

$$EOC_i = \frac{\ln(e_{i+1}/e_i)}{\ln(N_i/N_{i+1})}, \tag{38}$$

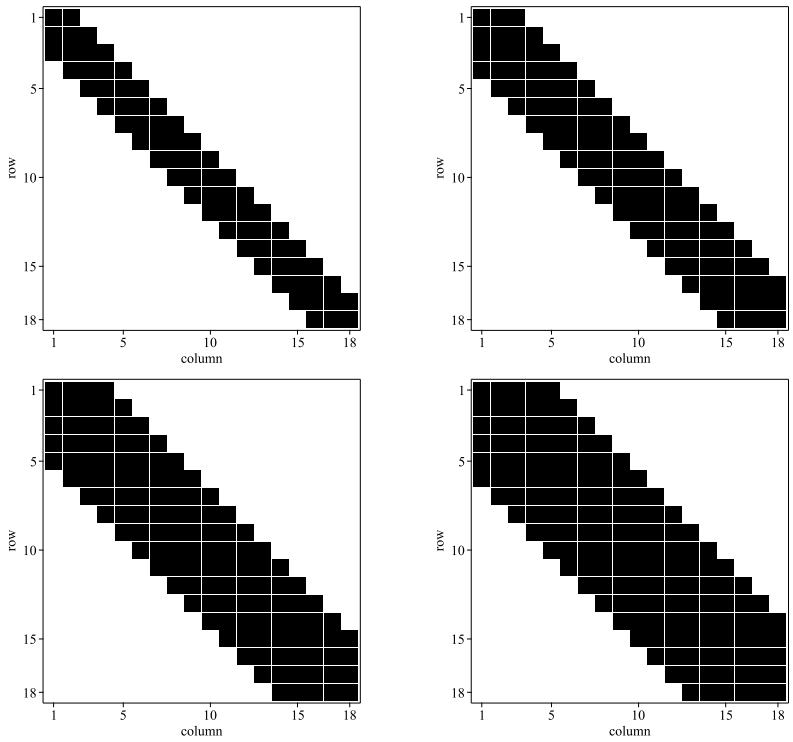

Figure 1: The structure of the matrices (top left R, right S, bottom left P, right Q).

where e_i stands for the error with the number of training points, N_i using $\Delta t = 1/N_i, i = 1,2,\ldots$ and a fixed number of spatial basis functions M. The experimental order of convergence in space is also computed similarly. It is worth noting that due to the physical understanding of the problem and the way the waves are interacting on the surface of water, the function u is assumed as an independent function of the angle and so a single spatial variable x is used as the distance to the center [29].

Example 1. As the first example, we consider the time-fractional KdV-Burgers equation (1) with the initial and boundary conditions given by (2) and (3), respectively. For the numerical simulation of the proposed LS-SVR method, we assume the exact solution as $u(x,t) = (1-x)\sin(\pi x)(\frac{2t}{T})^{\alpha/2+\theta}$. Note that for a simple presentation of the method, we assumed the homogenous boundary conditions, otherwise a change of variable is required at the beginning. We also set $\gamma = 1, \mu = 1$, and $T = 2$.

In Table 2, we report the errors in terms of the L_∞ norm as well as the convergence rates for some values of fractional orders $\alpha = 0.3, 0.5, 0.7$ and $\theta = 10$. We set $M = 8$, for the validation of the temporal convergence.

It is seen that the order of convergence in time is $O(\tau^{2-\alpha})$ supporting the theoretical results. In Figure 2, the convergence of the method is illustrated for three cases of fractional orders.

Example 2. Now, we consider the problem (1)-(3) with the exact solution

$$u(x,t) \quad = \quad (1-x)^2 \sin(2\pi x)\exp(-t).$$

Table 2: The numerical errors as L_∞ norm and the experimental order of convergence in time at $t = 1$ for some fractional orders.

	$\alpha = 0.3$		$\alpha = 0.5$		$\alpha = 0.7$	
N	Error	EOC	Error	EOC	Error	EOC
10	5.17E-03		6.29E-03		5.60E-03	
20	2.52E-03	1.0367	3.58E-03	0.8131	3.85E-03	0.5406
30	1.44E-03	1.3802	2.29E-03	1.1020	2.69E-03	0.8842
40	9.02E-04	1.6260	1.57E-03	1.3121	1.98E-03	1.0652
50	5.87E-04	1.9252	1.12E-03	1.5136	1.50E-03	1.2442

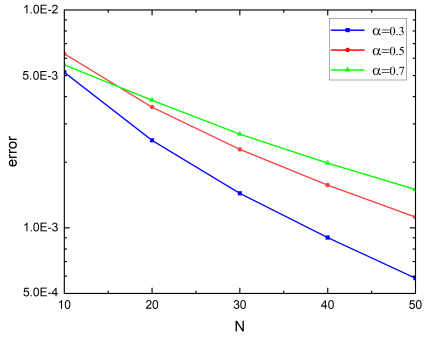

Figure 2: The temporal convergence of the LS-SVR method for the fractional KdV-Burgers equation with $M = 8$ and some fractional orders.

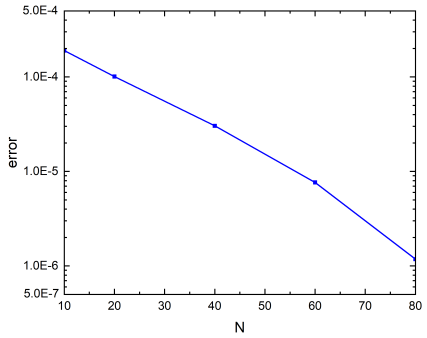

Figure 3: The temporal convergence of the LS-SVR method for the fractional KdV-Burgers equation with $M = 12$ and $\alpha = 0.5$.

For the proposed PGLS-SVR Galerkin approach, the numerical results are provided in Table 3. In this Table, the convergence both in space and time are reported as M and N increase, respectively.

Figures 3 and 4, illustrate the convergence of the PGLS-SVR for the problem in terms of The L_∞

Table 3: The error with different numbers of training points for the temporal and spatial accuracy at $t = 1$ for $\alpha = 0.5$. For the temporal results, we set $M = 12$ and for the spatial results, we set $N = 100$.

N	Error	M	Error
10	1.90E-04	4	3.48E-02
20	1.01E-04	6	1.76E-02
40	3.04E-05	8	1.50E-03
60	7.66E-06	10	7.07E-05
80	1.18E-06	12	1.30E-06

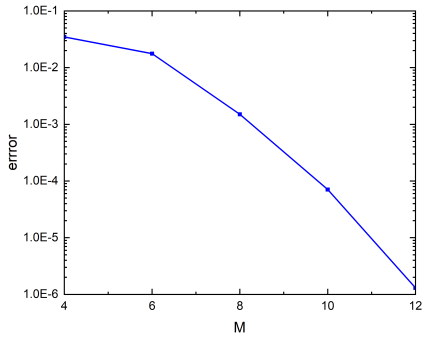

Figure 4: The spatial convergence of the LS-SVR method for the fractional KdV-Burgers equation with $N = 100$ and $\alpha = 0.5$.

norm for the various number of training functions in time and space, respectively. For the temporal results, we set $M = 12$ and for the spatial results, we set $N = 100$. It is seen that the spatial order of convergence is spectral since we set the vertical axis to be in logarithmic scale and the Fig. 4 behaves almost linearly as M increases.

In the next example, we consider the problem (1)-(3) with an exact solution representing a different behavior in comparison with the previous examples. It is written in terms of the Mittag-Leffler function that is usually implemented for the validation of the numerical schemes due to its applications in modeling relaxation of viscoelastic materials, see e.g., [19, 29].

Example 3. Now, we consider the time-fractional nonlinear KdV-Burgers equation (1) with the homogeneous boundary conditions (3) and the initial conditions (2) with the exact solution

$$u(x,t) \quad = \quad (1-x)^2 \sin(2\pi x)(\frac{2t}{T})^{\alpha} E_{1,\alpha/2+\theta+1}(\frac{2t}{T}),$$

where $E_{\bar{a},\bar{b}}$ is the Mittag-Leffler function given by

$$E_{\bar{a},\bar{b}}(z) = \sum_{\bar{k}=0}^{\infty} \frac{z^{\bar{k}}}{\Gamma(\bar{a}\bar{k}+\bar{b})}. \tag{39}$$

The parameters in (1) are chosen as $\gamma = 1, \mu = 1, T = 2$. In Tables 4 and 5, we report the errors in terms

Table 4: The temporal convergence of the LS-SVR scheme with the L_∞ measure as well as the experimental order of convergence at $t = 1$ for some fractional orders with $M = 12$.

	$\alpha = 0.3$		$\alpha = 0.5$		$\alpha = 0.7$	
N	Error	EOC	Error	EOC	Error	EOC
50	5.17E-07		5.58E-07		5.16E-07	
100	2.42E-07	1.0952	2.73E-07	1.0314	2.65E-07	0.9614
200	1.01E-07	1.2607	1.23E-07	1.1502	1.27E-07	1.0612
300	5.36E-08	1.5626	7.16E-08	1.3345	7.78E-08	1.2086
400	3.30E-08	1.6860	4.55E-08	1.5760	5.25E-08	1.3672

Table 5: The spatial convergence of the LS-SVR scheme with the L_∞ measure as well as the experimental order of convergence at $t = 1$ for some fractional orders with $N = 400$.

	$\alpha = 0.3$		$\alpha = 0.5$		$\alpha = 0.7$	
M	Error	EOC	Error	EOC	Error	EOC
4	7.25E-04		6.07E-04		5.08E-04	
6	3.63E-04	1.7061	3.03E-04	1.7136	2.53E-04	1.7192
8	3.12E-05	8.5302	2.61E-05	8.5226	2.18E-05	8.5215
10	1.48E-06	13.6611	1.25E-06	13.6181	1.05E-06	13.5927
12	3.30E-08	20.8603	4.55E-08	18.1722	5.25E-08	16.4310

of L_∞ norm as well as the convergence rates for some values of fractional orders $\alpha = 0.3, 0.5, 0.7$, and $\theta = 5$, to show the convergence in time and space, respectively.

To have a visual understanding of the convergence in time and space, we provide Tables 4, 5, and Figures 5 and 6. They confirm a spectral convergence as is expected in space and a polynomial convergence in time. Test problem 1 (of [29]) considers the problem (1)-(3) by means of the classic spectral Petrov-Galerkin method in which a Legendre basis has been used with a spectral convergence. The numerical results reported in Table 5 and Figure 6 show that our proposed algorithm also reach a spectral convergence.

It is seen that the spatial order of convergence for the proposed method is spectral since we set the vertical axis to be in logarithmic scale and Figure 6 behaves almost linearly as M increases verifying the spectral convergence. Moreover, the order of convergence in time is $O(\tau^{2-\alpha})$ supporting the theoretical results.

Table 6 considers the convergence in space and time for some fractional orders with $N = 100, M = 12$ by discrete L_∞ and L_2 norm given in (36)-(37).

The numerical results in Table 6 for the time fractional KdV problem (1)-(3) can be compared to the results given in [5] in which a B-Spline basis was utilized for the approximation of the unknown solution. The reported errors in Tables 4-6 in [5] for $\alpha = 0.5, 0.75, 1$ are between 10^{-2} to 10^{-5}.

The proposed learning algorithm can be expanded to address higher order nonlinear PDEs by incorporating quasi-linearization across all spatial dimensions and applying tensor product kernels, although additional research is needed in this area.

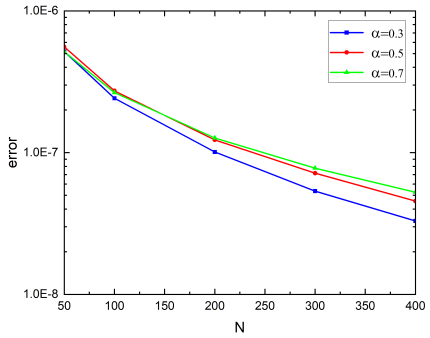

Figure 5: The temporal convergence of the LS-SVR method for the fractional KdV-Burgers equation with $M = 12$ and some fractional orders.

Table 6: L_2 and L_∞ norm errors for the fractional KdV-Burgers equation using Petrov-Galerkin LS-SVR method for some fractional orders.

	$\alpha = 1$		$\alpha = 0.75$		$\alpha = 0.5$	
t	L_2	L_∞	L_2	L_∞	L_2	L_∞
0.1	2.63e-10	5.59e-10	1.55e-11	3.29e-11	1.67e-11	3.55e-11
0.2	6.06e-09	1.29e-08	2.41e-10	5.14e-10	2.54e-10	5.40e-10
0.3	3.83e-08	8.14e-08	1.17e-09	2.48e-09	1.23e-09	2.61e-09
0.4	1.42e-07	3.03e-07	3.54e-09	7.52e-09	3.74e-09	7.96e-09
0.5	3.96e-07	8.43e-07	8.34e-09	1.77e-08	8.88e-09	1.89e-08
0.6	9.18e-07	1.95e-06	1.68e-08	3.57e-08	1.80e-08	3.82e-08
0.7	1.87e-06	3.99e-06	3.03e-08	6.44e-08	3.27e-08	6.93e-08
0.8	3.49e-06	7.42e-06	5.06e-08	1.07e-07	5.48e-08	1.16e-07
0.9	6.05e-06	1.29e-05	7.94e-08	1.68e-07	8.64e-08	1.83e-07
1.0	9.92e-06	2.11e-05	1.19e-07	2.52e-07	1.30e-07	2.75e-07

5 Conclusions

In this work, we presented a machine learning algorithm based on least squares support vector regression for the numerical simulation of the time-fractional KdV-Burgers equations. The nonlinearity and the non-local derivatives in the problem make it challenging for the numerical solvers. We first introduced a quasi-linearization as well as incorporating suitable trial and test spaces along with new polynomial kernel functions for the variational formulation to handle the challenges in the problem. We also presented a matrix formulation of the dual problem and investigated the resulting linear systems in a linear algebra framework. Finally, the numerical method was carried out for some test problems to support the theoretical results. From the numerical results, it is seen that the proposed method provides a fixed order convergence in time and a spectral convergence in space meaning the method gives the numerical results with higher accuracy than any method with polynomial order such as finite element and finite difference method. The method may be further developed to handle non-smooth data.

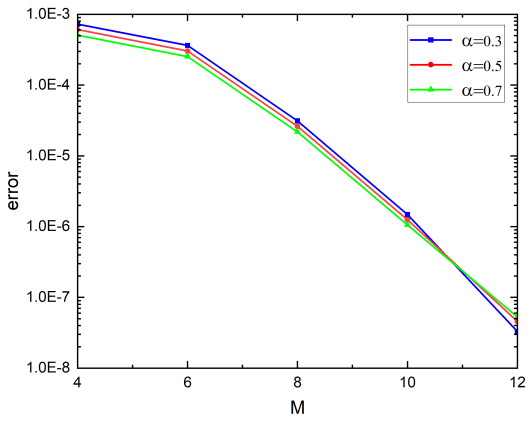

Figure 6: The spatial convergence of the LS-SVR method for the fractional KdV-Burgers equation with $N = 400$ and some fractional orders.

Acknowledgments

The authors are very grateful to the anonymous referees and the editor of this journal for carefully reading the paper and for their constructive comments and suggestions to improve the paper.

References

[1] E. Bas, R. Ozarslan, *Real world applications of fractional models by Atangana–Baleanu fractional derivative*, Chaos Solit. Fractals **116** (2018) 121–125.

[2] M. Caputo, M. Fabrizio, *A new definition of fractional derivative without singular kernel*, Prog. Fract. Differ. Appl. **1** (2015) 73–85.

[3] M. Dehghan, M. Abbaszadeh, *The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations,* J. Comput. Phys. **351** (2017) 478–510.

[4] R.T. Farouki, *The Bernstein polynomial basis: A centennial retrospective,* Comput. Aided Geom. Des. **29**, (2012) 379–419.

[5] A.K. Gupta, S.S. Ray, *On the solution of time-fractional KdV–Burgers equation using Petrov–Galerkin method for propagation of long wave in shallow water,* Chaos Solit. Fractals **116** (2018) 376–380.

[6] A.H. Hadian-Rasanan, N. Bajalan, K. Parand, J.A. Rad, *Simulation of nonlinear fractional dynamics arising in the modeling of cognitive decision making using a new fractional neural network,* Math. Methods Appl. Sci. **43** (2020) 1437–1466.

[7] A.H. Hadian-Rasanan, D. Rahmati, S. Gorgin, K. Parand, *A single layer fractional orthogonal neural network for solving various types of LaneEmden equation,* New Astron. **1** (2020) 101307.

[8] M. Jani, E. Babolian, S. Javadi, *Bernstein modal basis: Application to the spectral Petrov-Galerkin method for fractional partial differential equations,* Math. Methods Appl. Sci. **40** (2017) 7663–7672.

[9] M. Jani, E. Babolian, S. Javadi, D. Bhatta, *Banded operational matrices for Bernstein polynomials and application to the fractional advection-dispersion equation,* Numer. Algorithms **75** (2017) 1041–1063.

[10] B. Jin, R. Lazarov, Z. Zhou, *Two fully discrete schemes for fractional diffusion and diffusion-wave equations with nonsmooth data,* SIAM J. Sci. Comput. **38** (2016) A146–A170.

[11] B. Juttler, *The dual basis functions for the Bernstein polynomials,* Adv. Comput. Math. **8** (1998) 345–352.

[12] D. Kaya, S. Glbahar, A. Yokuş, M. Glbahar, *Solutions of the fractional combined KdV–mKdV equation with collocation method using radial basis function and their geometrical obstructions,* Adv. Differ. Equ. **1** (2018) 77.

[13] NR. Kevlahan, R. Khan, B. Protas, *On the convergence of data assimilation for the one-dimensional shallow water equations with sparse observations,* Adv. Comput. Math. **45** (2019) 3195–3216.

[14] M.M. Khader, K.M. Saad, Z. Hammouch, D. Baleanu, *A spectral collocation method for solving fractional KdV and KdV-Burgers equations with non-singular kernel derivatives,* Appl. Numer. Math. **161** (2021) 137–146.

[15] A. Kurganov, *Finite-volume schemes for shallow-water equations,* Acta Numer. **27** (2018) 289–351.

[16] J. Li, J. Chen, B. Li, *Gradient-optimized physics-informed neural networks (GOPINNs): a deep learning method for solving the complex modified KdV equation,* Nonlinear Dyn. **107** (2022) 781–792.

[17] Y. Lin, C. Xu, *Finite difference/spectral approximations for the time-fractional diffusion equation,* J. Comput. Phys. **225** (2007) 1533–1552.

[18] G.G. Lorentz, *Approximation of Functions*, American Mathematical Scociety, 2023.

[19] F. Mainardi, *Why the Mittag-Leffler function can be considered the Queen function of the Fractional Calculus?* Entropy, **22** (2020) 1359.

[20] S. Mehrkanoon, T. Falck, J.A. Suykens, *Approximate solutions to ordinary differential equations using least squares support vector machines*, IEEE Trans. Neural Netw. Learn. Syst. **23** (2012) 1356–1367.

[21] S. Mehrkanoon, S. Mehrkanoon, J.A. Suykens, *Parameter estimation of delay differential equations: an integration-free LS-SVM approach*, Commun. Nonlinear Sci. Numer. Simul. **19** (2014) 830–841.

[22] S. Mehrkanoon, J.A. Suykens, *Learning solutions to partial differential equations using LS-SVM*, Neurocomputing **159** (2015) 105–116.

[23] V.H. Moghaddam, J. Hamidzadeh, *New Hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier*, Pattern Recognit. **60** (2016) 921–935.

[24] A. Pakniyat, K. Parand, M. Jani, *Least squares support vector regression for differential equations on unbounded domains,* Chaos Solit. Fractals. **151** (2021) 111232.

[25] K. Parand, A.A. Aghaei, M. Jani, A. Ghodsi, *A new approach to the numerical solution of Fredholm integral equations using least squares-support vector regression*, Math. Comput. Simul. **180** (2021) 114–128.

[26] K. Parand, A.A. Aghaei, M. Jani, A. Ghodsi, *Parallel LS-SVM for the numerical simulation of fractional Volterra's population model*, Alex. Eng. J. **60** (2021) 5637–5647.

[27] K. Parand, M. Razzaghi, R. Sahleh, M. Jani, *Least squares support vector regression for solving Volterra integral equations*, Eng. Comput. **9** (2020) 789–796.

[28] J.A. Rad, K. Parand, S. Chakraverty, *Learning with Fractional Orthogonal Kernel Classifiers in Support Vector Machines: Theory, algorithms and applications,* Springer, 2023.

[29] Z. Yu, J. Sun, B. Wu, *A space–time spectral Petrov–Galerkin method for nonlinear time-fractional Korteweg–de Vries–Burgers equations,* Math. Methods Appl. Sci. **44** (2021) 4348–4365.

[30] H. Zhang, SJ. Cai, JY. Li, Y. Liu, ZY. Zhang, *Enforcing generalized conditional symmetry in physics-informed neural network for solving the KdV-like equation with Robin initial/boundary conditions,* Nonlinear Dyn.**111** (2023) 10381–10392.