

A NOVEL NUMERICAL METHOD FOR SOLVING FRACTIONAL DELAY INTEGRO-DIFFERENTIAL ALGEBRAIC EQUATIONS

HAITHAM M. KHAZEEM^{1*} and OSAMA H. MOHAMMED²

ABSTRACT. This paper introduces a new treatment of Least Squares Support Vector Machines (LS-SVMs) method for solving fractional order delay integro-differential algebraic equations (FDIDAEs). LS-SVMs are utilized to approximate solutions by transforming the FDIDAEs into a constrained optimization problem. The solutions are represented by weighted shifted Legendre polynomials with biases over the problem's domain. The solution procedure involves systematically determining the unknown weights and biases. Illustrative examples are provided to demonstrate the effectiveness of the proposed methodology. The study also presents a comparison with solutions obtained via fractional Physics-Informed Neural Networks Method.

Keywords. Fractional Order Delay Integro-Differential Algebraic Equations, Least Squares Support Vector Machines, Shifted Legendre Polynomials, Machine Learning

2020 Mathematics Subject Classification. Primary 34K37; Secondary 47G20

1. INTRODUCTION

Over the past few decades, fractional calculus, which involves non-integer-order derivatives and integrals of variable functions, has been demonstrated to be highly useful across various disciplines, including physics, chemistry, biology, economics, engineering, signal and image processing, and control theory [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Fractional-order differential equations (FDEs), fractional integral equations (FIEs), and fractional integro-differential equations (FIDEs) have become prominent research topics. Given the difficulty in obtaining analytical solutions for these equations, researchers focus on developing numerical and approximate solution methods, see [12, 13, 14, 15].

Fractional-order integro-differential algebraic equations (FIDAEs) constitute a subclass of fractional-order integro-differential equations, characterized by the presence of at least one equation within the system that lacks terms involving derivatives and integrals [16, 17].

Date: Received: Oct 7, 2025; Revised: Nov 5, 2026; Accepted: Nov 16, 2025.

* Corresponding author

© The Author(s) 2025. This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of the licence, visit <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

FIDAEs are extensively utilized in the modeling of dynamic processes over time or spatial domains, with applications spanning electric circuits, hydraulic systems, and chemical reaction dynamics [18].

When at least one term in a FIDAEs incorporates a fixed time delay $\tau > 0$, the equation is classified as a fractional-order delay integro-differential algebraic equation (FDIDAEs).

Least Squares Support Vector Machines (LS-SVMs) were introduced by J.A.K. Suykens in the late 1990s as a computationally efficient alternative to traditional Support Vector Machines (SVMs), originally developed by Vladimir Vapnik and colleagues in the early 1990s [19]. While SVMs brought significant advancements to machine learning through their strong theoretical foundations and effectiveness in classification and regression, their dependence on quadratic programming for optimization presented computational challenges, LS-SVMs given by substituting the hinge loss function in SVMs with a least squares loss, LS-SVMs reformulated the optimization problem, reducing it to solving a linear system instead of a quadratic one, training for regression and classification problems is subsequently performed by solving a system of linear equations [20].

Although classical neural networks possess desirable properties like universal approximation, they still face two persistent challenges. The first is the presence of numerous local minima solutions, and the second is determining the appropriate number of hidden units. SVMs are a strong tool for solving pattern recognition and function estimation problems [20].

In recent years, LS-SVMs has been employed in various studies, demonstrating its effectiveness in solving functional equations (see a timeline in [21]).

The structure of this paper is as follows: Section 2 presents preliminaries on fractional calculus. In sections 3 and 4 we explain the least squares support vector machines approach for solving FDIDAEs. Section 5 provides numerical experiments, along with a discussion.

2. PRELIMINARIES

This section provides essential background on fractional calculus, which plays a crucial role in our work [12, 3, 11, 22, 23].

Definition 2.1. The left Riemann-Liouville fractional-order integral of order α of a function u is defined as follows

$${}_0I_t^\alpha u(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} u(\tau) d\tau, t > 0, \alpha \in R^+$$

Definition 2.2. Let $u \in C[\alpha, m]$, $\alpha > 0$, the fractional order derivative of order α in the Caputo sense is

$${}_0^c D_t^\alpha u(t) = \begin{cases} \frac{1}{\Gamma(m - \alpha)} \int_0^t \frac{u^{(m)}(s)}{(t - s)^{\alpha+1-m}} ds, m - 1 < \alpha < m, s > 0, m \in N \\ \frac{d^m}{dt^m} u(t), \alpha = m \end{cases}$$

Remark 2.3. The following are key properties of the Riemann-Liouville (R-L) fractional integral and the Caputo fractional derivative.

- (1) ${}_0I_t^\alpha c t^\mu = c \frac{\Gamma(\mu+1)}{\Gamma(\mu+\alpha+1)} t^{\alpha+\mu}$, $\mu > -1$, $\alpha > 0$, $c \in \mathbf{R}$.
- (2) ${}_0^c D_t^\alpha c t^\mu = c \frac{\Gamma(\mu+1)}{\Gamma(\mu-\alpha+1)} t^{\mu-\alpha}$, $\mu > -1$, $\alpha > 0$, $c \in \mathbf{R}$.
- (3) ${}_0^c D_t^\alpha ({}_0I_t^\alpha u(t)) = u(t)$.
- (4) ${}_0I_t^\alpha ({}_0^c D_t^\alpha u(t)) = u(t) - \sum_{k=0}^{n-1} \frac{u^{(k)}(0)}{k!} t^k$, $t \geq 0$, $n-1 < \alpha < n$.

2.1. Shifted Legendre Polynomials and Function Approximation. In the interval $[-1, 1]$, Legendre polynomials are given by the following general form

$$P_{i+1}(t) = \frac{2i+1}{i+1} t P_i(t) - \frac{i}{i+1} P_{i-1}(t), \quad i = 1, 2, \dots$$

with $P_0(t) = 1$ and $P_1(t) = t$, [12].

The shifted Legendre polynomials P^* on the interval $[a, b]$ are given by the following relation

$$P_i^*(t) = P_i\left(\frac{2t-a-b}{b-a}\right)$$

On the interval $[0, 1]$ the Legendre polynomial of degree i in analytic form is [24]:

$$P_i^*(t) = \sum_{k=0}^i (-1)^{i+k} \frac{(i+k)!}{(i-k)!(k!)^2} t^k$$

Any function $u(t) \in L^2[0, 1]$ can be written as a linear combination of the shifted Legendre polynomials

$$u(t) = \sum_{i=0}^{\infty} w_i P_i^*(t) \quad \text{where} \quad P_i^*(t) = P_i(2t-1)$$

3. LEAST SQUARES SUPPORT VECTOR MACHINES

Least Squares Support Vector Machines [20] [25] is a machine learning-based method with both advantages and disadvantages. In terms of convergence, it exhibits exponential convergence for solving differential equations, regarding time complexity, it involves solving a positive definite linear system and requires a few to several parameters, in terms of scalability, it performs better than spectral methods but is less scalable than neural networks [21]. Let $\{t_i, u_i\}_{i=1}^N$ be a training set with input and output $t_i \in \mathbf{R}$ and $u_i \in \mathbf{R}$ here is one-dimensional input space, The objective of a regression problem is to construct a model in the form of $u(t) = w^T \varphi(t) + b$.

The primal LS-SVMs regression model can be expressed as follows

$$\begin{cases} \min_{w,b,e} \frac{1}{2} w^T w + \frac{\gamma}{2} e^T e \\ \text{s.t. } u_i = w^T \varphi(t_i) + b + e_i, \quad i = 1, 2, \dots, N \end{cases} \quad (3.1)$$

where $\gamma \in \mathbf{R}^+$, $b \in \mathbf{R}$, $w \in \mathbf{R}^h$. $\varphi(t) : \mathbf{R} \rightarrow \mathbf{R}^h$ is the feature map and h is the dimension of the feature space. The solution of problem (3.1) is given by the following system [25] :

$$\begin{bmatrix} \Omega + I_N/\gamma & \mathbf{1}_N \\ \mathbf{1}_N^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix} \quad (3.2)$$

where $\Omega_{ij} = K(t_i, t_j) = \varphi(t_i)^T \varphi(t_j)$, $\mathbf{1}_N = (1, 1, \dots, 1)^T \in \mathbf{R}^N$, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$, $u = (u_1, u_2, \dots, u_N)^T$ and I_N is $N \times N$ identity matrix. The model in the alternative representation can be expressed as: $u(t) = \sum_{i=1}^N \alpha_i K(t, t_i) + b$.

4. THE APPROACH

Consider the following FDIDAE

$${}_0^c D_t^{\alpha_i} u_i(t) = \int_0^t k_i(t, s, U(s)) ds + \psi_i(U(t), U(t - \tau)) + \phi_i(t), \quad (4.1)$$

$0 < \alpha_i < 1$, $i = 1, 2, \dots, n - 1$
subject to

$$g(U(t)) + q(t) = 0, \quad \tau > 0, \quad t > t_0$$

with the initial conditions

$$u_i(t) = \Psi_i(t), \quad i = 1, 2, \dots, n, \quad t \in [t_0 - \tau, t_0] \quad (4.2)$$

where $u_i(t) \in L^2[t_0, t_0 + \tau]$, $i = 1, 2, \dots, n$, $\tau > 0$ and $U(t)$ is a vector function defined by $U(t) = (u_i(t))_{i=1}^n$.

For the simplicity of the explanation of the methodology, let $n = 3$. To find the solution on the interval $[t_0, t_0 + \tau]$, substitute the initial conditions (4.2) into (4.1), then the FDIDAE will be transformed into FIDAE as follows:

$${}_0^c D_t^{\alpha_i} u_i(t) = \int_0^t k_i(t, s, U(s)) ds + \mathfrak{F}_i(U(t)) + \Phi_i(t), \quad 0 < \alpha_i < 1, \quad i = 1, 2 \quad (4.3)$$

subject to

$$g(U(t)) + q(t) = 0 \quad (4.4)$$

with initial conditions $u_i(t_0) = \Psi_i(t_0) = p_i$, $p_i \in \mathbf{R}$, $i = 1, 2, 3$. Now, let

$$\begin{aligned} \tilde{u}_1(t) &= w^T \varphi(t) + b_1 \\ \tilde{u}_2(t) &= v^T \varphi(t) + b_2 \\ \tilde{u}_3(t) &= r^T \varphi(t) + b_3 \end{aligned} \quad (4.5)$$

be the solution to equations (4.3)–(4.4), where $\varphi(t)$ is the h -dimensional vector of shifted Legendre polynomials on the domain of the problem, i.e., $\varphi(t) : [t_0, t_0 + \tau] \subset \mathbf{R} \rightarrow \mathbf{R}^h$.

Substituting equation (4.5) into equations (4.3)–(4.4), we can write the residuals of equations (4.3)–(4.4) along with those associated to initial conditions as follows:

$$\begin{cases} R_1 = \mathfrak{D}_1(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) - f_1 \\ R_2 = \mathfrak{D}_2(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) - f_2 \\ R_3 = \mathfrak{D}_3(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) - f_3 \\ R_4 = \mathfrak{D}_4(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) - f_4 \\ R_5 = \mathfrak{D}_5(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) - f_5 \\ R_6 = \mathfrak{D}_6(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) - f_6 \end{cases} \quad (4.6)$$

where $\mathfrak{D}_1(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) = {}^c_0D_t^{\alpha_1}\tilde{u}_1(t) - \int_0^t k_1(t, s, \tilde{U}(s))ds - \mathfrak{F}_1(\tilde{U}(t))$, $f_1 = \Phi_1(t)$, $\mathfrak{D}_2(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) = {}^c_0D_t^{\alpha_2}\tilde{u}_2(t) - \int_0^t k_2(t, s, \tilde{U}(s))ds - \mathfrak{F}_2(\tilde{U}(t))$, $f_2 = \Phi_2(t)$, $\mathfrak{D}_3(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) = g(\tilde{U}(t))$, $f_3 = -q(t)$, $\mathfrak{D}_4(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) = \tilde{u}_1(t_0)$, $f_4 = p_1$, $\mathfrak{D}_5(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) = \tilde{u}_2(t_0)$, $f_5 = p_2$, $\mathfrak{D}_6(\tilde{u}_1, \tilde{u}_2, \tilde{u}_3) = \tilde{u}_3(t_0)$, $f_6 = p_3$.

The LS-SVMs framework, applying the Galerkin technique [26], reformulates the minimization problem (3.1) in terms of an inner product. The main idea of the Galerkin method is to enforce the residuals (4.6) to be orthogonal to the basis functions $\varphi(t)$. The LS-SVMs framework, according to the Galerkin method, is formulated as follows:

$$\begin{cases} \min_{w,v,r,e,b} \frac{1}{2}w^Tw + \frac{1}{2}v^Tv + \frac{1}{2}r^Tr + \frac{\gamma}{2}e^Te \\ \text{s.t.} \begin{cases} \langle R_1, \varphi_i(t) \rangle = e_i \\ \langle R_2, \varphi_i(t) \rangle = e_{h+i} \\ \langle R_3, \varphi_i(t) \rangle = e_{2h+i} \\ \langle R_4, \varphi_i(t) \rangle = e_{3h+i} \\ \langle R_5, \varphi_i(t) \rangle = e_{4h+i} \\ \langle R_6, \varphi_i(t) \rangle = e_{5h+i} \end{cases} \end{cases} \quad (4.7)$$

, $i = 1, 2, \dots, h$, where $\langle \varphi_i(t), \varphi_j(t) \rangle = \int_{t_0}^{t_0+\tau} \varphi_i(t)\varphi_j(t)dt$

Assuming that $\mathfrak{D}_i$, for $i = 1, 2, \dots, 6$, are linear operators, the solution to (4.7) is provided by the following theorem.

Theorem 4.1. *Given $\varphi(t)$ the vector of first h shifted Legendre polynomials on $[t_0, t_0 + \tau]$ and regularization constant $\gamma \in \mathbf{R}^+$, the solution to (4.7) is obtained by solving the following dual problem*

$$\begin{bmatrix} \Omega + I/\gamma & \beta \\ \beta^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ b \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (4.8)$$

such that

$$w = \sum_{i=1}^6 A_i^T \lambda^{(i)}, \quad v = \sum_{i=1}^6 B_i^T \lambda^{(i)}, \quad r = \sum_{i=1}^6 C_i^T \lambda^{(i)}$$

where $\Omega_{6 \times 6}$ is given by $\Omega_{ij} = [A_i][A_j^T] + [B_i][B_j^T] + [C_i][C_j^T]$, $[A_k]_{ij} = \langle \mathfrak{D}_k(\varphi_j, 0, 0), \varphi_i \rangle$, $[B_k]_{ij} = \langle \mathfrak{D}_k(0, \varphi_j, 0), \varphi_i \rangle$, $[C_k]_{ij} = \langle \mathfrak{D}_k(0, 0, \varphi_j), \varphi_i \rangle$ and $\beta_{6 \times 3}$ is given by $\beta_{i1} = \langle \mathfrak{D}_i(1, 0, 0), \varphi \rangle$, $\beta_{i2} = \langle \mathfrak{D}_i(0, 1, 0), \varphi \rangle$, $\beta_{i3} = \langle \mathfrak{D}_i(0, 0, 1), \varphi \rangle$, $\lambda = (\lambda^{(1)} = (\lambda_1, \dots, \lambda_h)^T, \lambda^{(2)} = (\lambda_{h+1}, \dots, \lambda_{2h})^T, \dots, \lambda^{(6)} = (\lambda_{5h+1}, \dots, \lambda_{6h})^T)^T$, $b = (b_1, b_2, b_3)^T$, $F = (F_1, F_2, \dots, F_6)^T$, $F_i = \langle f_i, \varphi \rangle$.

Proof. Making use of the linearity of $\mathfrak{D}_i$, $i = 1, 2, \dots, 6$ and the inner product, we can rewrite the constraints of (4.7) as follows

$$\begin{cases} A_1w + B_1v + C_1r + b_1\beta_{11} + b_2\beta_{12} + b_3\beta_{13} - F_1 = E_1 \\ A_2w + B_2v + C_2r + b_1\beta_{21} + b_2\beta_{22} + b_3\beta_{23} - F_2 = E_2 \\ A_3w + B_3v + C_3r + b_1\beta_{31} + b_2\beta_{32} + b_3\beta_{33} - F_3 = E_3 \\ A_4w + B_4v + C_4r + b_1\beta_{41} + b_2\beta_{42} + b_3\beta_{43} - F_4 = E_4 \\ A_5w + B_5v + C_5r + b_1\beta_{51} + b_2\beta_{52} + b_3\beta_{53} - F_5 = E_5 \\ A_6w + B_6v + C_6r + b_1\beta_{61} + b_2\beta_{62} + b_3\beta_{63} - F_6 = E_6 \end{cases} \quad (4.9)$$

where $[A_k]_{ij} = \langle \mathfrak{D}_k(\varphi_j, 0, 0), \varphi_i \rangle$, $[B_k]_{ij} = \langle \mathfrak{D}_k(0, \varphi_j, 0), \varphi_i \rangle$, $[C_k]_{ij} = \langle \mathfrak{D}_k(0, 0, \varphi_j), \varphi_i \rangle$, $\beta_{k1} = \langle \mathfrak{D}_k(1, 0, 0), \varphi \rangle$, $\beta_{k2} = \langle \mathfrak{D}_k(0, 1, 0), \varphi \rangle$, $\beta_{k3} = \langle \mathfrak{D}_k(0, 0, 1), \varphi \rangle$, $F_k = \langle f_k, \varphi \rangle$, $E = (E_1 = (e_1, e_2, \dots, e_h)^T, E_2 = (e_{h+1}, e_{h+2}, \dots, e_{2h})^T, \dots, E_6 = (e_{5h+1}, e_{5h+2}, \dots, e_{6h})^T)^T$.

The Lagrangian function corresponding to equation (4.7), is formulated using expression (4.9) as follows:

$$L(w, v, r, E, \lambda, b) = \frac{1}{2}w^Tw + \frac{1}{2}v^Tv + \frac{1}{2}r^Tr + \frac{\gamma}{2}E^TE - (A_1w + B_1v + C_1r + b_1\beta_{11} + b_2\beta_{12} + b_3\beta_{13} - F_1 - E_1)^T\lambda^{(1)} - (A_2w + B_2v + C_2r + b_1\beta_{21} + b_2\beta_{22} + b_3\beta_{23} - F_2 - E_2)^T\lambda^{(2)} - (A_3w + B_3v + C_3r + b_1\beta_{31} + b_2\beta_{32} + b_3\beta_{33} - F_3 - E_3)^T\lambda^{(3)} - (A_4w + B_4v + C_4r + b_1\beta_{41} + b_2\beta_{42} + b_3\beta_{43} - F_4 - E_4)^T\lambda^{(4)} - (A_5w + B_5v + C_5r + b_1\beta_{51} + b_2\beta_{52} + b_3\beta_{53} - F_5 - E_5)^T\lambda^{(5)} - (A_6w + B_6v + C_6r + b_1\beta_{61} + b_2\beta_{62} + b_3\beta_{63} - F_6 - E_6)^T\lambda^{(6)}$$

The Karush-Khan-Tucker (KKT) optimality conditions are given by

$$\frac{\partial L}{\partial w} = 0 \rightarrow$$

$$w = A_1^T\lambda^{(1)} + A_2^T\lambda^{(2)} + A_3^T\lambda^{(3)} + A_4^T\lambda^{(4)} + A_5^T\lambda^{(5)} + A_6^T\lambda^{(6)} \quad (4.10)$$

$$\frac{\partial L}{\partial v} = 0 \rightarrow$$

$$v = B_1^T\lambda^{(1)} + B_2^T\lambda^{(2)} + B_3^T\lambda^{(3)} + B_4^T\lambda^{(4)} + B_5^T\lambda^{(5)} + B_6^T\lambda^{(6)} \quad (4.11)$$

$$\frac{\partial L}{\partial r} = 0 \rightarrow$$

$$r = C_1^T\lambda^{(1)} + C_2^T\lambda^{(2)} + C_3^T\lambda^{(3)} + C_4^T\lambda^{(4)} + C_5^T\lambda^{(5)} + C_6^T\lambda^{(6)} \quad (4.12)$$

$$\frac{\partial L}{\partial E_k} = \gamma E_k + \lambda^{(k)} = 0 \rightarrow$$

$$E_k = -\frac{\lambda^{(k)}}{\gamma}, \quad k = 1, 2, \dots, 6 \quad (4.13)$$

$$\frac{\partial L}{\partial \lambda^{(k)}} = 0 \rightarrow$$

$$A_kw + B_kv + C_kr + b_1\beta_{k1} + b_2\beta_{k2} + b_3\beta_{k3} - F_k - E_k = 0, \quad k = 1, 2, \dots, 6 \quad (4.14)$$

$$\frac{\partial L}{\partial b_1} = 0 \rightarrow$$

$$\beta_{11}\lambda^{(1)} + \beta_{21}\lambda^{(2)} + \beta_{31}\lambda^{(3)} + \beta_{41}\lambda^{(4)} + \beta_{51}\lambda^{(5)} + \beta_{61}\lambda^{(6)} = 0 \quad (4.15)$$

$$\frac{\partial L}{\partial b_2} = 0 \rightarrow \beta_{12}\lambda^{(1)} + \beta_{22}\lambda^{(2)} + \beta_{32}\lambda^{(3)} + \beta_{42}\lambda^{(4)} + \beta_{52}\lambda^{(5)} + \beta_{62}\lambda^{(6)} = 0 \quad (4.16)$$

$$\frac{\partial L}{\partial b_3} = 0 \rightarrow \beta_{13}\lambda^{(1)} + \beta_{23}\lambda^{(2)} + \beta_{33}\lambda^{(3)} + \beta_{43}\lambda^{(4)} + \beta_{53}\lambda^{(5)} + \beta_{63}\lambda^{(6)} = 0 \quad (4.17)$$

Substituting (4.10)–(4.13) into (4.14) in order to eliminate w , v , r and E_k , then, the system (4.14)–(4.17) directly gives (4.8), this completes the proof.

5. ILLUSTRATIVE EXAMPLES

In this section, illustrative examples are presented in order to demonstrate the efficiency of the method. All computations were performed using MATLAB R2024a on a computer equipped with a 12th Gen Intel Core i9-12900H processor and 20 CPUs.

Remark 5.1. It should be emphasized that, in contrast to the regression setting, the absence of target values implies the absence of noise. Accordingly, a sufficiently large regularization parameter γ is required to ensure that the error terms e_i , $i = 1, 2, \dots, 6h$ are effectively minimized, or equivalently, that the constraints are stringently satisfied [20]. Therefore, in all examples, we set $\gamma = 10^{15}$.

Example 5.2. Consider the following FDIDAE

$${}_0^c D_t^{0.5} u_1(t) = u_2(t) + \int_0^t u_1(s) ds - \sin(t) - u_1(t-1) + 1 + \sqrt{t} E_{1,1.5} \quad (5.1)$$

subject to

$$u_1(t) + u_2(t) - e^t - \sin(t) = 0 \quad (5.2)$$

$t \geq 0$, with initial conditions

$$u_1(t) = e^{t+1}, \quad t \in [-1, 0], \quad u_2(0) = 0 \quad (5.3)$$

Where $E_{1,1.5}$ is the Mittag-Leffler function. The closed form of the solution of the problem (5.1)–(5.3) is given by $u_1(t) = e^t$, $u_2(t) = \sin(t)$.

In Example 5.2, we chose $h = 8$, i.e, the first eight shifted Legendre polynomials on $[0, 1]$, subsequently, the training for the regression problem is performed by solving a linear system of thirty four equations, the elapsed time to obtain the results was 5.304572 seconds.

Figure 1 illustrates a comparison between the exact and approximate solutions in the $t - u$ plane with the relative errors for both functions. Additionally, Table 1 provides a selection of results, including the exact and approximate solutions for $u_1(t)$ and $u_2(t)$ along with their corresponding relative errors.

Table 2 represents a comparison between the proposed approach and the results obtained by using fractional Physics-Informed Neural Networks Method (fPINNs).

TABLE 1. The exact and approximate solutions u and $\tilde{u}$ and the relative error $\tilde{E}$ for Example 5.2.

t	u_1	$\tilde{u}_1$	$\tilde{E}$	u_2	$\tilde{u}_2$	$\tilde{E}$
0.22222	1.2488	1.2488	3.74e-09	0.2204	0.2204	2.25 e-08
0.44444	1.5596	1.5596	3.76e-07	0.42996	0.42996	1.36e-06
0.61616	1.8518	1.8518	3.10e-06	0.57791	0.57791	9.94e-06
0.88889	2.4324	2.4324	3.08e-05	0.77637	0.77645	9.66e-05
1	2.7183	2.7181	6.32e-05	0.84147	0.84164	2.04e-04

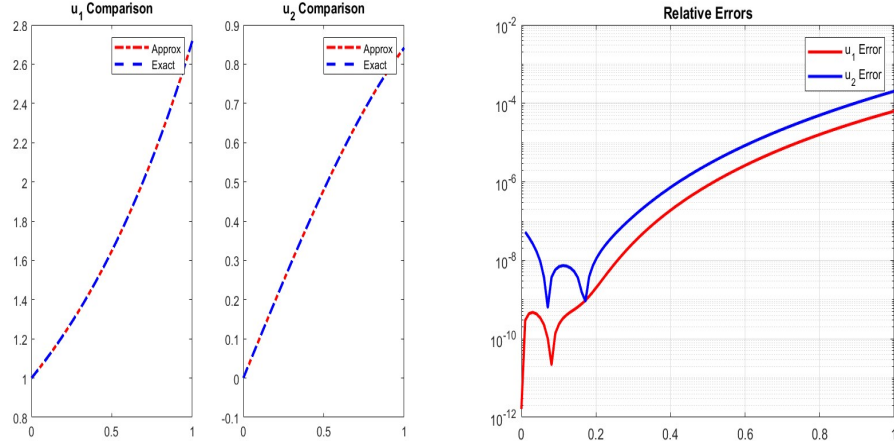
FIGURE 1. A comparison between the exact and approximate solutions with the relative errors $|u - \tilde{u}|/|u|$ of $u_1(t)$ and $u_2(t)$ for Example 5.2.

TABLE 2. Comparison of relative errors at selected points for Example 5.2.

t	fPINNs $\tilde{E}_1$	LS-SVMs $\tilde{E}_1$	fPINNs $\tilde{E}_2$	LS-SVMs $\tilde{E}_2$
0.44444	3.54e-05	3.76e-07	0.00214	1.3613e-06
0.67677	8.76e-05	5.63e-06	0.00066	1.7678e-05
0.86869	7.71e-05	2.6766e-05	0.00016	8.3569e-05

Example 5.3. Consider the following FIDAE

$${}_0^c D_t^{0.5} u_1(t) = u_3(t) + \int_0^t (t-s) u_2(s) ds + \sqrt{t} E_{1,1.5} - \sin(t) - \frac{t^4}{12} - \cos(t) \quad (5.4)$$

$${}_0^c D_t^{0.5} u_2(t) = \int_0^t (t-s) u_3(s) ds + \frac{\Gamma(3)}{\Gamma(2.5)} t^{1.5} - t - 1 + \cos(t) + \sin(t) \quad (5.5)$$

subject to

$$u_1(t) + u_2(t) + u_3(t) - e^t - t^2 - \cos(t) - \sin(t) = 0 \quad (5.6)$$

with $t \in [0, 1]$, $u_1(0) = u_3(0) = 1$, $u_2(0) = 0$, the exact solution of (5.4)–(5.6) is given by $u_1(t) = e^t$, $u_2(t) = t^2$, $u_3(t) = \cos(t) + \sin(t)$.

In example 5.3, we chose $h = 8$, i.e, the first eight shifted Legendre polynomials on $[0, 1]$, subsequently, the training for the regression problem is performed by solving a linear system of fifty one equations, the elapsed time to obtain the results was 10.267354 seconds.

Figure 2 illustrates a comparison between the exact and approximate solutions in the $t - u$ plane with the relative errors for both functions. Additionally, Table 3 provides a selection of results, including the exact and approximate solutions for $u_1(t)$, $u_2(t)$ and $u_3(t)$ along with their corresponding relative errors.

Table 4 represents a comparison between the proposed approach and the results obtained by using fractional Physics-Informed Neural Networks Method (fPINNs).

TABLE 3. The exact and approximate solutions u and $\tilde{u}$ and the relative error $\tilde{E}$ for Example 5.3.

t	u_1	$\tilde{u}_1$	$\tilde{E}$	u_2	$\tilde{u}_2$	$\tilde{E}$
0.22222	1.2488	1.2488	3.67e-09	0.049383	0.049383	8.53e-10
0.44444	1.5596	1.5596	3.72e-07	0.19753	0.19753	1.39e-09
0.66667	1.9477	1.9477	5.02e-06	0.44444	0.44444	3.61e-08
0.88889	2.4324	2.4324	3.00e-05	0.79012	0.79012	3.07e-07
1	2.7183	2.7181	6.13e-05	1	1	7.41e-07
	u_3	$\tilde{u}_3$	$\tilde{E}$			
		1.1958	1.1958	4.43e-09		
		1.3328	1.3328	4.34e-07		
		1.4043	1.4043	6.96e-06		
		1.4066	1.4067	5.18e-05		
		1.3818	1.3819	1.20e-04		

TABLE 4. Comparison of relative errors at selected points for Example 5.3.

t	fPINNs $\tilde{E}_1$	LS-SVMs $\tilde{E}_1$	fPINNs $\tilde{E}_2$	LS-SVMs $\tilde{E}_2$
0.44444	0.00022395	3.7232e-07	0.00080118	1.3992e-09
0.67677	0.00035811	5.524e-06	0.0011338	4.042e-08
0.86869	0.00027162	2.6087e-05	0.00030882	2.5844e-07
	fPINNs $\tilde{E}_3$		LS-SVMs $\tilde{E}_3$	
		0.0022385	4.3414e-07	
		0.0011763	7.7191e-06	
		0.0028003	4.3985e-05	

Example 5.4. Consider the following FIDAE

$${}_0^c D_t^{0.5} u_1(t) = \int_0^t u_2(s) ds - \frac{1}{2}(1 - \cos(t)) + \sum_{k=0}^{\infty} \frac{\Gamma(k+3)t^{k+1.5}}{k!\Gamma(k+2.5)} \quad (5.7)$$

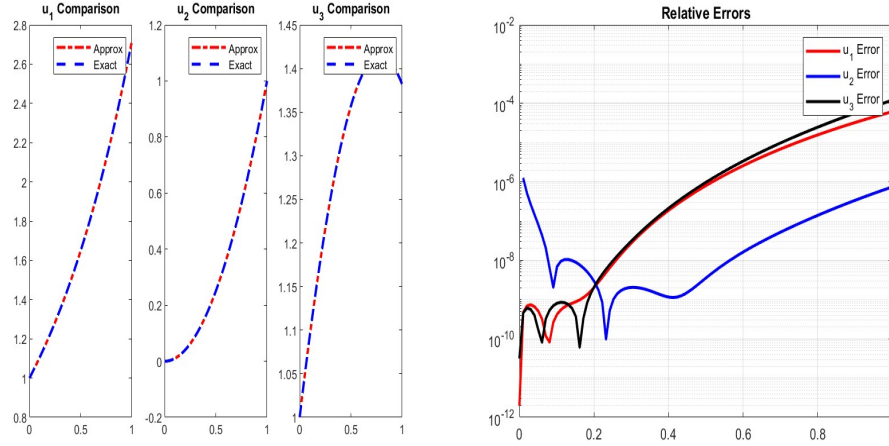


FIGURE 2. A comparison between the exact and approximate solutions with the relative errors $|u - \tilde{u}|/|u|$ of $u_1(t)$, $u_2(t)$ and $u_3(t)$ for Example 5.3.

$${}_0^c D_t^{0.5} u_3(t) = \int_0^t u_1(s) ds + \frac{\Gamma(2)}{\Gamma(1.5)} \sqrt{t} + \frac{\Gamma(3)}{\Gamma(2.5)} t \sqrt{t} - ((t^2 - 2t + 2)e^t - 2) \quad (5.8)$$

subject to

$$u_1(t) + u_2(t) + u_3(t) - t^2 e^t - t - t^2 - t \sin(t^2) = 0 \quad (5.9)$$

with $t \in [0, 1]$, $u_1(0) = u_2(0) = u_3(0) = 0$ and exact solution $u_1(t) = t^2 e^t$, $u_2(t) = t \sin(t^2)$, $u_3(t) = t + t^2$.

In Example 5.4, we chose $h = 8$, i.e, the first eight shifted Legendre polynomials on $[0, 1]$, subsequently, the training for the regression problem is performed by solving a linear system of fifty one equations, the elapsed time to obtain the results was 12.733701 seconds.

Figure 3 illustrates a comparison between the exact and approximate solutions in the $t - u$ plane with the relative errors for both functions. Additionally, Table 5 provides a selection of results, including the exact and approximate solutions for $u_1(t)$, $u_2(t)$ and $u_3(t)$ along with their corresponding relative errors.

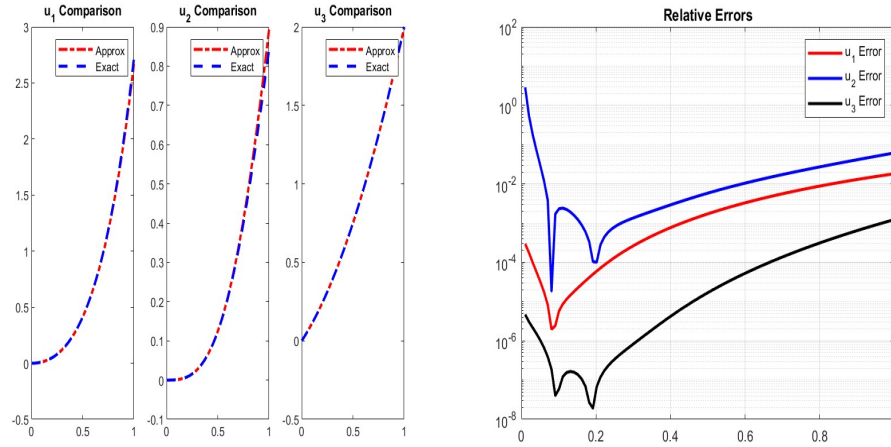
Table 6, represents a comparison between the proposed approach and the results obtained by using Fractional Physics-Informed Neural Networks Method (fPINNs).

6. CONCLUSIONS

In this study, we have explored the application of Least Squares Support Vector Machines to solve fractional order delay integro-differential algebraic equations, a class of equations notorious for their mathematical complexity and computational challenges. By taking advantage of the fundamental strengths of LS-SVMs and its kernel-based framework, flexibility, and computational efficiency, we presented a strong approach that successfully handles the combined challenges of fractional

TABLE 5. The exact and approximate solutions u and $\tilde{u}$ and the relative error $\tilde{E}$ for Example 5.4.

t	u_1	$\tilde{u}_1$	$\tilde{E}$	u_2	$\tilde{u}_2$	$\tilde{E}$
0.22222	0.061672	0.061666	8.50e-05	0.010969	0.010974	4.40e-04
0.44444	0.30807	0.30773	1.13e-03	0.087222	0.087571	4.00e-03
0.66667	0.86566	0.86157	4.73e-03	0.28664	0.29085	1.47e-02
0.88889	1.9219	1.8981	1.24e-02	0.6315	0.65629	3.93e-02
1	2.7183	2.6692	1.81e-02	0.84147	0.89298	6.12e-02
		u_3	$\tilde{u}_3$	$\tilde{E}$		
		0.2716	0.2716	1.72e-07		
		0.64198	0.64197	8.07e-06		
		1.1111	1.111	1.01e-04		
		1.679	1.678	5.95e-04		
		2	1.9975	1.23e-03		

FIGURE 3. A comparison between the exact and approximate solutions with the relative errors $|u - \tilde{u}|/|u|$ of $u_1(t)$, $u_2(t)$ and $u_3(t)$ for Example 5.4.

derivatives, delay terms, integral operators, and algebraic constraints. The proposed methodology make use of shifted Legendre polynomials to approximate the unknown solutions and reformulates FDIDAEs into a constrained optimization problem within the LS-SVMs framework. A linear system of unknown Lagrange multipliers and biases is obtained, which is easy to solve by any method. Using Galerkin technique decreases the number of Lagrange multipliers, subsequently decreases the number of equations to solve, instead of the direct substitution of the problem into the LS-SVMs framework with the collocation of the interval which is increases the number of equations to solve. The proposed method was tested and compared with the fractional Physics-Informed Neural Networks method, and the results demonstrate high accuracy and efficiency, even for equations with strong algebraic constraints.

TABLE 6. Comparison of relative errors at selected points for Example 5.4.

t	fPINNs $\tilde{E}_1$	LS-SVMs $\tilde{E}_1$	fPINNs $\tilde{E}_2$	LS-SVMs $\tilde{E}_2$
0.47475	0.0024379	0.0014284	0.019754	0.0049201
0.69697	0.0059882	0.0055031	0.019485	0.01699
0.87879	0.011942	0.011929	0.035277	0.037665
		fPINNs $\tilde{E}_3$	LS-SVMs $\tilde{E}_3$	
		0.00014358	1.2183e-05	
		0.0001507	0.00013289	
		0.00050309	0.00055434	

REFERENCES

1. HongGuang Sun and Yong Zhang and Dumitru Baleanu and Wen Chen and YangQuan Chen (2018). A new collection of real world applications of fractional calculus in science and engineering. Communications in Nonlinear Science and Numerical Simulation 64 (213–231). Elsevier. doi: [10.1016/j.cnsns.2018.04.019](https://doi.org/10.1016/j.cnsns.2018.04.019).
2. Mortezaee, Marzieh and Ghovatmand, Mehdi and Nazemi, Alireza (2020). Solving variable-order fractional differential algebraic equations via generalized fuzzy hyperbolic model with application in electric circuit modeling. Soft Computing (24) (22) (16745–16758). Springer. doi: [10.1007/s00500-020-04969-7](https://doi.org/10.1007/s00500-020-04969-7).
3. Dalir, Mehdi and Bashour, Majid (2010). Applications of fractional calculus. Applied Mathematical Sciences (4) (21) (1021–1032). Appl. Math. Sci., Vol. 4, 2010, no. 21-24, 1021-1032.
4. Hilfer, Rudolf (2000). Applications of fractional calculus in physics. World scientific. books.google.com.
5. Tenreiro Machado, JA and Silva, Manuel F and Barbosa, Ramiro S and Jesus, Isabel S and Reis, Cecilia M and Marcos, Maria G and Galhano, Alexandra F (2010). Some applications of fractional calculus in engineering. Mathematical problems in engineering (2010) (1) (639801). Wiley Online Library. doi: [10.1155/2010/639801](https://doi.org/10.1155/2010/639801).
6. Tarasov, Vasily E (2019). On history of mathematical economics: Application of fractional calculus. Mathematics (7) (6) (509). MDPI. doi: [10.3390/math7060509](https://doi.org/10.3390/math7060509).
7. Arora, Sugandha and Mathur, Trilok and Agarwal, Shivi and Tiwari, Kamlesh and Gupta, Phalguni (2022). Applications of fractional calculus in computer vision: a survey. Neurocomputing 489, 407–428. Elsevier. doi: [10.1016/j.neucom.2021.10.122](https://doi.org/10.1016/j.neucom.2021.10.122).
8. Singh, Abhaya Pal and Bingi, Kishore (2024). Applications of fractional-order calculus in robotics. Fractal and Fractional 8, 7, 403. MDPI. doi: [10.3390/fractalfract8070403](https://doi.org/10.3390/fractalfract8070403).
9. Miranda-Valdez, Isaac Y and Puente-Córdova, Jesús G and Rentería-Baltierrez, Flor Y and Fliri, Lukas and Hummel, Michael and Puisto, Antti and Koivisto, Juha and Alava, Mikko J (2024). Viscoelastic phenomena in methylcellulose aqueous systems: Application of fractional calculus. Food Hydrocolloids 147, 109334. Elsevier. doi: [10.1016/j.foodhyd.2023.109334](https://doi.org/10.1016/j.foodhyd.2023.109334).
10. Meng, Ruifan (2021). Application of fractional calculus to modeling the non-linear behaviors of ferroelectric polymer composites: viscoelasticity and dielectricity. Membranes 11, 6, 409. MDPI. doi: [10.3390/membranes11060409](https://doi.org/10.3390/membranes11060409).
11. Cheow, Yi Herng and Ng, Kok Haur and Phang, Chang and Ng, Kooi Huat (2024). The application of fractional calculus in economic growth modelling: An approach based on regression analysis. Heliyon, 10, 15. Elsevier. [10.1016/j.heliyon.2024.e35379](https://doi.org/10.1016/j.heliyon.2024.e35379).

12. Abdulhameed, Noor A and Mohammed, Osama H and Yousif, Ahmed A (2023). A hybrid technique for approximating the solution of fractional order integro differential equations. *Partial Differential Equations in Applied Mathematics* 8, 100552. Elsevier. doi: [10.1016/j.padiff.2023.100552](https://doi.org/10.1016/j.padiff.2023.100552).
13. Talib, Ahmed N and Mohammed, Osama H (2022). Artificial Neural Network Technique for Solving Variable Order Fractional Integro-Differential Algebraic Equations. *Al-Nahrain Journal of Science* 25, 3, 25-32. mail.anjs.edu.iq Vol. 25 No. 3 (2022): September.
14. Yousif, Ahmed A and AbdulKhaleq, Fajir A and Mohsin, Ahmed K and Mohammed, Osama H and Malik, Adyan M (2023). A developed technique of homotopy analysis method for solving nonlinear systems of Volterra integro-differential equations of fractional order. *Partial Differential Equations in Applied Mathematics* 8, 100548. Elsevier. doi: [10.1016/j.padiff.2023.100548](https://doi.org/10.1016/j.padiff.2023.100548).
15. Mohammed, Hasnaa F and Mohammed, Osama H (2024). A hybrid technique for solving fractional delay variational problems by the shifted Legendre polynomials. *Partial Differential Equations in Applied Mathematics* 9, 100635. Elsevier. doi: [10.1016/j.padiff.2024.100635](https://doi.org/10.1016/j.padiff.2024.100635).
16. Shiri, Babak and Baleanu, Dumitru (2019). System of fractional differential algebraic equations with applications. *Chaos, Solitons & Fractals* 120, 203-212. Elsevier. doi: [10.1016/j.chaos.2019.01.028](https://doi.org/10.1016/j.chaos.2019.01.028).
17. Xing, Baixue and Liu, Haixia and Liu, Hongliang and Tang, Xiao (2023). Chebyshev Neural Network Method for Solving Delay-Integro-Differential-Algebraic Equations Based on Variable Transformation. doi: [10.21203/rs.3.rs-3173404/v1](https://doi.org/10.21203/rs.3.rs-3173404/v1).
18. Yang, Wenqiang and Wu, Wenyuan and Reid, Greg (2023). Structural Analysis by Modified Signature Matrix for Integro-differential-algebraic Equations. arXiv preprint arXiv:2301.06986. doi: [10.48550/arXiv.2301.06986](https://doi.org/10.48550/arXiv.2301.06986).
19. Suykens, Johan AK and Vandewalle, Joos (1999). Least squares support vector machine classifiers. *Neural processing letters* 9, 3, 293-300. Springer. doi: [10.1023/A:1018628609742](https://doi.org/10.1023/A:1018628609742).
20. Mehrkanoon, Siamak and Falck, Tillmann and Suykens, Johan AK (2012). Approximate solutions to ordinary differential equations using least squares support vector machines. *IEEE transactions on neural networks and learning systems* 23, 9, 1356–1367. IEEE. doi: [10.1109/TNNLS.2012.2202126](https://doi.org/10.1109/TNNLS.2012.2202126).
21. Taheri, Tayebbeh and Aghaei, Alireza Afzal and Parand, Kourosh (2025). A new kernel-based approach for solving general fractional (integro)-differential-algebraic equations. *Engineering with Computers* 41, 2, 845–864. Springer. doi: [10.1007/s00366-024-02054-3](https://doi.org/10.1007/s00366-024-02054-3).
22. Farhood, Adnan Khalaf and Mohammed, Osama H and Taha, Bushra A (2023). Solving fractional time-delay diffusion equation with variable-order derivative based on shifted Legendre–Laguerre operational matrices. *Arabian Journal of Mathematics* 12, 3, 529–539. Springer. doi: [10.1007/s40065-022-00416-7](https://doi.org/10.1007/s40065-022-00416-7).
23. Mohammed, Osama H and Mohsin, Ahmed K (2021). Approximate methods for solving one-dimensional partial integro-differential equations of fractional order. *Italian Journal of Pure and Applied Mathematics* 205. *IJPAM-no-46-2021*.
24. Çerdik Yaslan, Handan and Mutlu, Ferdi (2020). Numerical solution of the conformable differential equations via shifted Legendre polynomials. *International Journal of Computer Mathematics* 97, 5, 1016–1028. Taylor & Francis. doi: [10.1080/00207160.2019.1605059](https://doi.org/10.1080/00207160.2019.1605059).
25. Suykens, Johan AK and Van Gestel, Tony and De Brabanter, Joseph and De Moor, Bart and Vandewalle, Joos PL (2002). Least squares support vector machines. World scientific. doi: [10.1142/5089](https://doi.org/10.1142/5089).
26. Parand, K and Razzaghi, M and Sahleh, R and Jani, M (2022). Least squares support vector regression for solving Volterra integral equations. *Engineering with Computers* 38, Suppl 1, 789–796. Springer. doi: [10.1007/s00366-020-01186-6](https://doi.org/10.1007/s00366-020-01186-6).

¹ DEPARTMENT OF MATHEMATICS AND COMPUTER APPLICATIONS, COLLEGE OF SCIENCE,
AL-NAHRAIN UNIVERSITY, BAGHDAD, IRAQ.

Email address: haithem.mms23@ced.nahrainuniv.edu.iq

² DEPARTMENT OF MATHEMATICS AND COMPUTER APPLICATIONS, COLLEGE OF SCIENCE,
AL-NAHRAIN UNIVERSITY, BAGHDAD, IRAQ.

Email address: osama.hameed@nahrainuniv.edu.iq