



## OPEN Legendre based neural networks integrated with heuristic algorithms for the analysis of Lorenz chaotic model: an intelligent and comparative study

Khalid Masood<sup>1</sup>✉, Muhammad Arshad<sup>2</sup>, Muhammad Abubakar<sup>1</sup>,  
Muhammad Naeem Aslam<sup>2</sup>, Mohammed A. Alghamdi<sup>3</sup> & Sultan H. Almotiri<sup>4</sup>

The optimal control methods for managing Lorenz model are achieved using an innovative intelligent computing framework that integrates artificial neural networks with stochastic unsupervised learning-based optimizers, specifically Firefly (FA) and Archimedes algorithm (AOA). The hyper parameters (weights & biases) of unsupervised neural networks are refined through Legendre polynomials Artificial Neural Networks (LENN) optimized with global search techniques, including FA-AOA collectively referred to as LENN-FA-AOA. This design approach is employed to Lorenz model across three (3) different scenarios using various step sizes and input intervals. The study's findings reveal that to minimize the computational cost to find the solution of nonlinear chaotic systems by intelligence strategies. The absolute errors values from LENN-FA-AOA with reference solution are being ranged from  $3.22 \times 10^{-5}$  to  $3.06 \times 10^{-7}$ ,  $4.56 \times 10^{-5}$  to  $7.27 \times 10^{-8}$  and  $5.17 \times 10^{-5}$  to  $2.11 \times 10^{-7}$ . Data validation through extensive graphical simulations confirms the effectiveness and robustness of the proposed intelligent solver. The LENN-FA-AOA solver is tested under different initial conditions of Lorenz model to assess its reliability, safety, and tolerance. Through this advanced LENN intelligent design framework, an objective/fitness optimization function is developed within a feedforward neural network. The hybrid FA-AOA optimization is also investigated to verify the LENN model accuracy and reliability. Mean square error (MSE) and TIC graphs are constructed to evaluate the proposed method integrity and efficiency.

**Keywords** Legendre neural networks, Chaotic system, Water cycle algorithm, Firefly algorithm, Hybrid approach

### Abbreviations

PDEs	Partial differential equations
ODEs	Ordinary differential equations
NLDEs	Nonlinear differential equations
ANNs	Artificial neural networks
PINNs	Physics-informed neural networks
$\rho, \sigma$ and R	Lorenz parameters
LeNN	Legendre neural networks
FA	Firefly algorithm
AOA	Archimedes optimization algorithm
MSE	Mean square error
TIC	Theil's inequality coefficient
LENN-FA-AOA	Hybrid Legendre neural networks with firefly algorithm and Archimedes optimization

<sup>1</sup>Department of Computer Science, Lahore Garrison University, Lahore, Pakistan. <sup>2</sup>Department of Mathematics, Lahore Garrison University, Lahore, Pakistan. <sup>3</sup>Department of Computer Science, Umm Al-Qura University, Makkah City, Saudi Arabia. <sup>4</sup>Department of Cybersecurity, College of Computing, Umm Al-Qura University, Makkah City, Saudi Arabia. ✉email: kmasoodk@gmail.com

algorithm  
5D, 6D and 7D      Five, six and seven dimensions

Chaos theory is a field of mathematics that analyze the effect of nonlinear dynamic systems, which are extremely sensitive to their initial conditions. This sensitivity results in dynamics that appear random, a phenomenon known as chaos. Chaotic systems have been studied in various domains such as mathematics, engineering's (chemical, biological and mechanical), physics, and other sciences due to its non-linear nature. Scientists and engineers have formulated several approaches to understand and control the chaotic dynamics, such as bifurcation theory. These methodologies have been utilized to various systems, like electrical, biological and mechanical. Chaotic systems are a very significant area of research in various domains, with the potential to enhance our understanding of nature and its applications.

Nonlinear differential equations (NLDEs) have become a standard model for representing chaotic systems. Kudryashov<sup>1</sup> explored analytical solutions to the Lorenz system that have been found and further classified all exact solutions. Algaba et al.<sup>2</sup> used Poincare sections to study the global connections developed by the subcritical Hopf bifurcation in the Lorenz system. Barrio and Serrano<sup>3</sup> conducted a theoretical and numerical investigation of the conventional Lorenz model, identifying the region where each positive semi-orbit converges to equilibrium and establishing constraints for the chaotic zone. Wu and Zhang<sup>4</sup> characterized all rational integrals and Darboux polynomials for Lorenz systems. Algaba et al.<sup>5</sup> conducted numerical analysis to find bifurcations in the Lorenz system. Eusebius Doedel et al. illustrated preturbulence in the three-dimensional space of the Lorenz system<sup>6</sup>. The Krishnan et al.<sup>7</sup> utilized the laplace homotopy method (LHM) to determine the solution of Lorenz differential equations (LDEs). Klöwer et al.<sup>8</sup> found that non-periodic simulations utilizing deterministic finite-precision result in periodic orbits.

The theory of nonlinear systems can be applied to solution of problems in various fields, including economics, astronomy, nerve physiology, chemistry, heartbeat control, cryptography, electronic circuits, and many others. Most systems in the modern world are by nature nonlinear<sup>9,10</sup>. Furthermore, many researchers have been interested in nonlinear oscillations because most vibration problems are nonlinear. Thus, the nonlinear differential equations (NLDEs) were very useful in understanding scientific and engineering problems, which often take the form of nonlinear types. The significance of mathematical computations was emphasized in numerous research studies and literature works that deal with NLDEs that arise in a variety of scientific and engineering fields<sup>11</sup>. While a large number of NLDEs can be numerically studied, very few of them can be solved directly. Several approximation schemes have been utilized in existing literature to investigate the correlation between the frequency and amplitude of the nonlinear oscillators (NOSs) such as Homotopy perturbation technique<sup>12,13</sup>, Variational iteration method<sup>14</sup>, energy balance approach and others<sup>15,16</sup>, Akbari Ganji method<sup>17</sup>, Adomain decomposition technique<sup>18</sup> and many others.

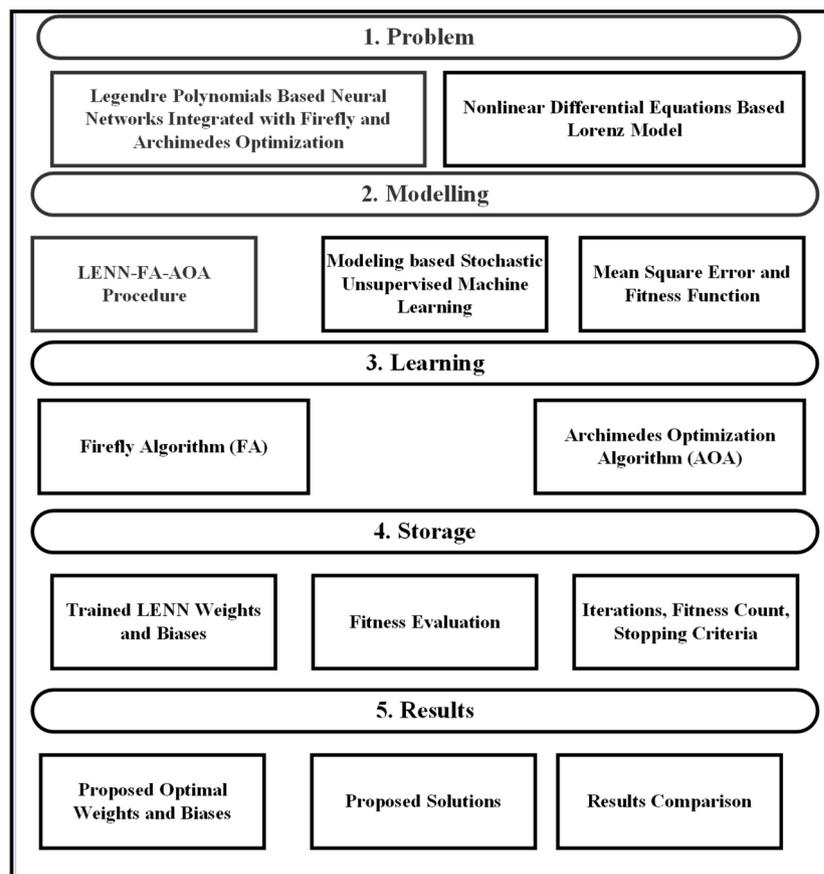
In recent hot area of research, there has also been a lot of research on the use of artificial neural networks (ANNs) to solve nonlinear differential equations. Yang et al.<sup>19</sup> proposed physics-informed generative adversarial networks as one data-driven method for handling stochastic differential equations (SDEs) using sparse observations. Raissi<sup>20</sup> investigated the application of deep learning methods for solving coupled SDEs. Mattheakis et al.<sup>21</sup> constructed neural networks based on physics phenomena to analyze the DEs that describe the dynamic behavior of systems. The NN integrates the Hamiltonian formulation utilizing a loss function, which ensures the energy efficiency of the solutions. The predictions made by the network are used to build this loss function in its entirety, no outside data is required. Piscopo et al.<sup>22</sup> studied for determining fully differentiable solutions for ordinary, partial (PDEs/ODEs) equations for analytical results. They investigated various network architectures and found that even very small networks achieved remarkable outcomes. Hagege et al.<sup>23</sup> introduced a system capable of estimating unknown functions or variable in iterated differential equations with suitable accuracy and sensitivity analysis for equations. This approach training the parameters of ANN for differential equation solvers at every time step, enabling faster backpropagation. Similarly, Mattheakis et al.<sup>24</sup> employed an artificial neural network (ANN) to solve nonlinear differential equations while ensuring energy conservation and velocity. They designed a symplectic neural network that adheres to energy conservation principles using the concept ANN. Raissi et al.<sup>25</sup> proposed the concept of physics-informed neural networks (PINNs), which are ANN trained to perform supervised learning tasks while complying with the nonlinear partial differential equations (PDEs/ODEs) dictated by physical laws. However, the challenge of addressing the “curse of dimensionality” has posed difficulties in developing algorithms for high-dimensional PDEs/ODEs. To address this, Han and Jentzen<sup>26</sup> presented a deep learning (DL) based method for solving high-dimensional PDEs/ODEs. Their approach involves transforming PDEs into stochastic differential equations (SDEs) and using neural networks to approximate the gradient of unknown solutions, a method comparable to deep reinforcement learning where the gradient acts as the policy function. In recent years, swarm intelligence (SI) has emerged as a popular method for solving optimization problems. Buhl et al.<sup>27</sup> highlighted how agents in SI interact with one another and their environment. Additionally, research by Blum and Li<sup>28</sup> demonstrated that many machine learning algorithms draw inspiration from the collective behavior of natural systems, such as ant colonies, animal herds, and bird flocks. The firefly algorithm (FA) is a global algorithm developed by Yang and Karamanoglu<sup>29</sup> that was inspired by the bioluminescent communication and flickering motion of fireflies. Yang<sup>30</sup> created the Firefly Algorithm based on these concepts, fireflies are unisexual, meaning that they become attracted to one another independent of sex, and the strength of this attraction is directly correlated with the brightness of each firefly's lights. However, the attraction reduces as the two fireflies get farther apart. When two fireflies are equally bright, they will fly in opposing directions. Note that the firefly intensity should correspond to the objective function of the task. Because of their attraction, fireflies can divide into smaller groups, and each group eventually settles on a single optimum solution. This feature, according to Apostolopoulos and Vlachos<sup>31</sup>, makes the Firefly Algorithm especially helpful for optimization challenges. The mathematical and logical

processes that make up an algorithm must be as simple as feasible in the field of applied mathematics. Due to its natural simplicity, FA behavior is ideally suitable for the solution of continuous mathematical functions. Based on statistical analyses utilizing simple stochastic test functions<sup>32</sup>, FA outperforms other widely used algorithms, gaining plaudits for its effectiveness. To solve optimization problems in complicated systems<sup>33</sup>, researchers have employed the FA technique.

The traditional approach fails to find solutions for nonlinear systems. To address this issue, a more effective method is needed. In this research, we use the Legendre polynomial as an activation function to handle nonlinearity and achieve an optimal solution for the Lorenz model. The modelling and prediction of intricate patterns that would be challenging to investigate using traditional methods, is made possible by ANNs ability to capture the complicated relationships and nonlinearity found in these problems. The ability of artificial neural networks (ANNs) to capture complicated nonlinear interactions within a data set inspired them to tackle the challenges of nonlinear oscillator modelling. In this paper, we use LENN-FA-AOA to optimize parameters, improve accuracy and speed up simulations for nonlinear oscillators. This study aims to investigate how the accuracy of ANN models can be enhanced by combining the heuristic algorithms. The design methodology represents in Fig. 1.

### Key features of this study include

- Application of the LENN-FA-AOA approach to analyze the chaotic Lorenz model.
- Reliable combined outcomes from FA-AOA and the Adams numerical method as a reference solution indicate the proposed approach accuracy.
- Validation of the LENN-FA-AOA model statistical performance across multiple runs for nonlinear dynamics using various statistical metrics.
- Comparison with reference solutions for Lorenz model shows that the LENN-FA-AOA results are accurate, stable, precise, and convergent.
- Further validation of the LENN-FA-AOA model performance is provided by quantitative metrics, including mean square error (MSE) and Theil's inequality coefficient (TIC). The use of LENN offers a logical sequence of solutions across the entire training and learning region, making the proposed stochastic intelligent computational solver efficient and stable.



**Fig. 1.** Design Methodology of the proposed LENN-FA-AOA.

### Problem formulation

The non-linear chaotic system consists of three nonlinear differential equations that characterize the behavior of a dynamical system. The system of nonlinear Lorenz equation expressed as follow<sup>34</sup>:

$$\begin{aligned} \frac{dx}{dt} &= \rho (y(t) - x(t)) \\ \frac{dy}{dt} &= \sigma x(t) - y(t) - x(t)z(t) \\ \frac{dz}{dt} &= x(t)y(t) - Rz(t) \\ t &\in [0, T] \end{aligned} \tag{1}$$

where  $t$  represents time and  $T$  denotes the final value of time and initial conditions are given as,

$$x(0) = a_1, y(0) = a_2, z(0) = a_3$$

In this given Lorenz model, the variables  $x(t)$ ,  $y(t)$  and  $z(t)$  are considered as three-dimensional coordinates. The dynamic of the chaotic system is analyze using these parameters,  $\rho$ ,  $\sigma$  and  $R$ , which demonstrate the intensity of non-linear interactions across the variables.

### Methodology: Legendre neural network

There are two stages to the suggested approach for resolving the governing mathematical model of Lorenz differential equation. The initial stage involves designing Legendre neural networks (LeNN) based on polynomials. The second phase demonstrate the optimizing unknown weights in the LeNN structure using the built network to determine the fitness value in an unsupervised manner.

### Design of Le-NN model

The architecture of a single-layer Legendre Neural Network (LeNN), which is made up of a functional expansion block based on Legendre polynomials as well as an input and one output layer. To eliminate the hidden layer, the input pattern is transformed to a higher dimension using Legendre polynomials<sup>35</sup>. They are orthogonal across  $[-1, 1]$  and belong to the set of orthogonal polynomials. The first ten Legendre polynomials are listed in Table 1.

To construct high-order Legendre polynomials, use the recursive formula given Eq. (2):

$$L_{n+1}(\xi) = \frac{1}{n+1} [(2n+1)\xi L_n(\xi) - nL_{n-1}(\xi)]. \tag{2}$$

The problem’s mathematical model is represented by the series solution, which includes input, hidden, and output layers.

The solution  $f(\xi)$  and its higher derivatives can be expressed as Eqs. (3), (4), (5), (6), (7).

$$\hat{f}(t) = \sum_{p=1}^q \phi_p L(\omega_p t + \beta_p), \tag{3}$$

$$\hat{f}'(t) = \sum_{p=1}^q \phi_p L'(\omega_p t + \beta_p), \tag{4}$$

$$\hat{f}''(t) = \sum_{p=1}^q \phi_p L''(\omega_p t + \beta_p), \tag{5}$$

$n$	$L_n(t)$
0	1
1	$t$
2	$\frac{1}{2}(3t^2 - 1)$
3	$\frac{1}{2}(5t^3 - 3t)$
4	$\frac{1}{8}(35t^4 - 30t^2 + 3)$
5	$\frac{1}{8}(63t^5 - 70t^3 + 15t)$
6	$\frac{1}{16}(231t^6 - 315t^4 + 105t^2 - 5)$
7	$\frac{1}{16}(429t^7 - 693t^5 + 315t^3 - 35t)$
8	$\frac{1}{128}(643t^8 - 12,012t^6 + 6930t^4 - 1260t^2 + 35)$
9	$\frac{1}{128}(12155t^9 - 25,740t^7 + 18,018t^5 - 4620t^3 + 315t)$
10	$\frac{1}{256}(46,189t^{10} - 109,395t^8 + 90,090t^6 - 30,030t^4 + 3465t^2)$

**Table 1.** Legendre polynomials.

$$\widehat{f}'''(t) = \sum_{p=1}^q \phi_p L'''(\omega_p t + \beta_p), \quad (6)$$

$$\widehat{f}^{iv}(t) = \sum_{p=1}^q \phi_p L^{iv}(\omega_p t + \beta_p). \quad (7)$$

$\phi = [\phi_1, \phi_2, \phi_3, \dots, \phi_n]$ ,  $\omega = [\omega_1, \omega_2, \omega_3, \dots, \omega_n]$  and  $\beta = [\beta_1, \beta_2, \beta_3, \dots, \beta_n]$  are represented the real-valued vectors and bounded. Where  $L$  denotes the Legendre polynomials,  $n$  represents the order of polynomial and  $p$  demonstrates the number of neurons in the proposed structure. Table 1 presents the formulation of Legendre polynomials.

### LENN based fitness function

The fitness function is designed for the proposed problem with initial conditions in term of mean square error (MSE) are given in Eq. (8).

$$E = e_1 + e_2 + e_3 + e_4 \quad (8)$$

The fitness functions for the non-linear Lorenz model are in Eqs. (9), (10), (11), (12),

$$e_1 = \frac{1}{Q} \sum_{P=1}^Q \left( \frac{d\widehat{x}}{dt} - (\rho\widehat{y}(t) + \widehat{x}(t)) \right)^2 \quad (9)$$

$$e_2 = \frac{1}{Q} \sum_{p=1}^Q \left( \frac{d\widehat{y}}{dt} - \sigma\widehat{x}(t) + \widehat{y}(t) + \widehat{x}(t)\widehat{z}(t) \right)^2 \quad (10)$$

$$e_3 = \frac{1}{Q} \sum_{p=1}^Q \left( \frac{d\widehat{z}}{dt} - \widehat{x}(t)\widehat{y}(t) + R\widehat{z}(t) \right)^2 \quad (11)$$

$$e_4 = \frac{1}{3} (x(u) - a)^2 + y(v) - b)^2 + z(w) - c)^2 \quad (12)$$

The purpose of developing fitness functions for Lorenz attractor analysis is to find weights in the LENN structure that minimize error. The proposed technique accurately approximates the exact solution for the given problem when the fitness function value approaches zero.

### Meta-heuristic optimization algorithms

Heuristic schemes have played a crucial role in handling a broad variety of optimization problems, both unconstrained and constrained, across diverse engineering fields. These schemes were developed to provide approximate/optimal solutions to problems that typical optimization approaches cannot solve due to the difficulty or nonlinearity of the function involved. In the past years, researchers have designed various heuristics algorithms, with unique methodology. Several algorithms have made major contributions to engineering, including genetic algorithm GA, PSO, FA, WCA, GGO<sup>36,37</sup>, ABC, and many others.

#### Firefly algorithm

Fireflies (FA) are bioluminescent insects emitting light from their abdomen without ultraviolet frequencies. They use this light to attract mates, lure prey, and warn predators, aiding in protection. The attraction of firefly  $j$  to firefly  $i$  is determined by two factors: the firefly brightness  $i$  and the distance across firefly  $j$  and firefly  $i$ , as shown in Eq. (13).

$$I(r) = \frac{I_s}{r^2} \quad (13)$$

For  $n$  fireflies,  $x_i$  represents each firefly's solution, with brightness  $I$  determined by the objective function  $f(x_i)$ , reflecting its fitness value

$$I_i = f(z_i) \quad (14)$$

Fireflies reduced brightness are attracted to brighter ones, each having a unique attraction rating  $\beta$ , which varies with distance, as defined by the Eq. (15).

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (15)$$

where  $\beta_0$  indicates the initial attraction value of a firefly at  $r = 0$ , and  $\gamma$  is the coefficient of absorption of light in the medium that surrounds it. The following equation to analyze how a firefly ( $i$ ) at position  $z_i$  transitions to a brighter firefly ( $j$ ) at position  $z_j$ .

$$z_i(t+1) = z_i(t) + \beta_0 e^{-\gamma r^2} (z_j - z_i) + \alpha \epsilon_i \quad (16)$$

where  $\beta_0 e^{-\gamma r^2}$  denotes the attraction of firefly  $z_j$  and  $\alpha \epsilon_i$  indicates the random parameters. When  $\beta_0 = 0$ , the movement becomes a random process. The algorithm relocates the firefly if the new position has higher attractiveness; otherwise, it stays put. The termination condition for the firefly algorithm can be based on iterations or fitness values. The equation for the randomized movement of the brightest firefly is in Eq. (17).

$$z_i(t+1) = z_i(t) + \alpha \epsilon_i \quad (17)$$

### Archimedes optimization algorithm

Hashim et al.<sup>38</sup> proposed the new Archimedes Optimization Algorithm (AOA) that, inspired by Archimedes' principles of buoyancy and floating bodies. The heuristic algorithm combines local search and global exploration to refine potential solutions based on their fitness values<sup>39,40</sup>. By simulating buoyancy, it assigns higher weights to more effective solutions, enhancing optimization efficiency. The proposed optimization algorithm continuously improves potential solutions through the combination of local search techniques such as gradient descent or global exploration approach like crossover. This technique aims to find a balance across exploitation and exploration, utilizing both local and global knowledge to rapidly identify optimal or nearly perfect solutions. The Archimedes method has successfully solved optimization problems in various fields, particularly in engineering, data analysis, and logistics. This study aims to evaluate the effectiveness of the Archimedes optimization approach for a specific optimization problem. We aim to analyze its efficacy, advantages, and challenges, and compare it to other meta-heuristic methods. Our study aims to contribute to the understanding of optimization strategies and their applications. Suppose there are multiple objects, each attempting to reach a state of equilibrium. The acceleration of submerged things varies based on their density and volume. If the buoyant force  $F_b$  and weight  $W_0$  are equal, the object is in equilibrium.

$$F_b = W_0 \quad (18)$$

$$P_b \nu_b a_b = P_0 \nu_0 a_0 \quad (19)$$

In the equations above, subscripts  $b$  and  $o$  represent fluid and submerged items.  $P$  indicates the density,  $a$  denotes the gravity/acceleration, and  $\nu$  is the volume. If the item is influenced by another force, like a collision with a neighboring object  $r$ , the equilibrium state will change that represents in Eqs. (20), (21).

$$F_b - W_r = W_0 \quad (20)$$

$$P_b \nu_b a_b - P_r \nu_r a_r = P_0 \nu_0 a_0 \quad (21)$$

The AOA technique randomly populates the search space by random selected volumes, accelerations, and densities. After evaluating the initial population's fitness, AOA iteratively updates each item's density, volume, and speed until the termination condition is met, determining new locations based on these properties. Here is a mathematical description of the proposed algorithm phases.

$$O_j = LB_J + rand^*(UB_J - LB_J), j = 1, 2, 3, \dots, M \quad (22)$$

where  $O_j$  denotes the  $j^{th}$  object between  $N$  total items.  $LB_J$  and  $UB_J$  indicate the lower and upper limitation of the search space. Initializes volume and density for every  $j^{th}$  item using

$$den_j = rand, vol_j = rand \quad (23)$$

$rand$  is a  $D$ -dimensional vector that creates random numbers between 0 and 1. Initialize the acceleration of the  $j^{th}$  item using Eq. (24).

$$acc_j = LB_J + rand^*(UB_J - LB_J), j = 1, 2, 3, \dots, M \quad (24)$$

Evaluate the initial population and select the item with the higher fitness value. Specify the order of  $x_{best}$ ,  $den_{best}$ ,  $vol_{best}$ , and  $accel_{best}$ . The density and volume of item  $J$  are adjusted for iteration  $t + 1$  using the following Eqs. (25), (26):

$$den_j^{t+1} = den_j^t + rand^*(den_{best} - den_j^t) \quad (25)$$

$$vol_j^{t+1} = vol_j^t + rand^*(vol_{best} - vol_j^t) \quad (26)$$

The terms  $vol_{best}$  and  $den_{best}$  refer to the volume and density of the best item identified. The  $rand$  variable indicates a uniformly distributed random number. Objects collide at first, but eventually strive for equilibrium. The proposed AOA algorithm uses the transfer operator (TF) to move the search from exploration to exploitation. The transfer operator transforms search behaviour, enhancing convergence to optimal solutions.

$$T_F = \exp\left(\frac{t - t_{max}}{t_{max}}\right) \quad (27)$$

In this case, the transfer  $T_F$  increases eventually unless it reaches 1. The variables  $t$  and  $t_{max}$  indicate the iteration and maximum number, respectively. Furthermore, the density decreasing parameter  $d$  helps balance across the global and local searches. Equation (11) shows a decrease over time.

$$d^{t+1} = \exp\left[\left(\frac{t - t_{max}}{t_{max}}\right) - \frac{t}{t_{max}}\right] \quad (28)$$

The algorithm can converge to an already determined promising region when the value of  $d^{t+1}$  reduces over time. A collision across objects is indicated by a transfer operator TF of 0.5 or less. To update an object's acceleration for iteration  $t + 1$ , a random material  $mr$  is selected and a specific technique in Eq. (29).

$$acc_j^{t+1} = \frac{den_{mr} + vol_{mr} acc_{mr}}{den_j^{t+1} vol_j^{t+1}} \quad (29)$$

where  $den_{mr}$ ,  $vol_{mr}$ , and  $acc_{mr}$  indicate the density, volume and acceleration of the random material. The TF value greater than 0.5 implies no collision across items. In that case, the item's acceleration is adjusted for the next iteration ( $t + 1$ ) utilizing a specific technique or formula.

$$acc_j^{t+1} = \frac{den_{best} + vol_{best} acc_{best}}{den_j^{t+1} vol_j^{t+1}} \quad (30)$$

To determine the percentage of change, divide the acceleration by a reference value.

$$acc_{j-norm}^{t+1} = u \left( \frac{acc_j^{t+1} - \min(acc)}{\max(acc) - \min(acc)} \right) + l \quad (31)$$

The normalization ranges ( $u$  and  $l$ ) are set at 0.9 and 0.1. Where  $acc_{j-norm}^{t+1}$  specifies the percentage of each agent's step change. If an object is placed distant from the global optimum, it will receive a greater acceleration value, showing it is in the exploration phase. If the item is close to the global optimum, it will receive a smaller acceleration value, demonstrating it is in the exploitation phase. This change from exploration to exploitation illustrates the way the search process evolves. When TF is below or equal to 0.5, it represents the exploration phase. The location of the  $j^{th}$  item for the next iteration ( $t + 1$ ) is computed utilizing the following Eqs. (32), (33).

$$x_j^{i+1} = x_j^i + C_1 \cdot rand \cdot acc_{j-norm}^{t+1} \cdot d \cdot (x_{rand} - x_j^i) \quad (32)$$

When TF exceeds 0.5, showing the exploitation phase, the items update their locations with a constant value of  $C_1$  is equal to zero.

$$x_j^{i+1} = x_{best}^t + F \cdot C_2 \cdot rand \cdot acc_{j-norm}^{t+1} \cdot d \cdot (T x_{best} - x_j^{i+1}) \quad (33)$$

where  $C_1$  is equal to 6, the variable  $T$  is increasing over time and is proportional to the transfer operator. Particularly,  $T$  can be expressed as  $T = C_3 T_F T$ , where  $C_1$  indicates a constant value and  $T$  continuously increases with time over the range of  $[C_3 \times 0.3, 1]$ . Initially,  $T$  selects a percentage from the best location. The parameter  $G$  is a flag utilized to change the direction of motion.

$$G = \begin{cases} 1, & \text{if } P \leq 0.5 \\ -1, & \text{if } P > 0.5 \end{cases} \quad (34)$$

Here  $P = 2rand - C_4$ . We apply the objective function  $f$  to evaluate each object and track the optimal solution identified so far.

### Performance indices

In this section, we presented the statistical technique for MSE and TIC to the nonlinear Lorenz model. The mathematical indices consist of MSE and TIC are demonstrated as given,

$$\begin{bmatrix} MSE_x \\ MSE_y \\ MSE_z \end{bmatrix} = \begin{bmatrix} \sum_{p=1}^q (x_p - \hat{x}_p)^2 \\ \sum_{p=1}^q (y_p - \hat{y}_p)^2 \\ \sum_{p=1}^q (z_p - \hat{z}_p)^2 \end{bmatrix} \quad (35)$$

$$\begin{bmatrix} TIC_x \\ TIC_y \\ TIC_z \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{\frac{1}{q} \sum_{p=1}^q (x_p - \hat{x}_p)^2}}{\sqrt{\frac{1}{q} \sum_{p=1}^q (x_p)^2 + \frac{1}{q} \sum_{p=1}^q (\hat{x}_p)^2}} \\ \frac{\sqrt{\frac{1}{q} \sum_{p=1}^q (y_p - \hat{y}_p)^2}}{\sqrt{\frac{1}{q} \sum_{p=1}^q (y_p)^2 + \frac{1}{q} \sum_{p=1}^q (\hat{y}_p)^2}} \\ \frac{\sqrt{\frac{1}{q} \sum_{p=1}^q (z_p - \hat{z}_p)^2}}{\sqrt{\frac{1}{q} \sum_{p=1}^q (z_p)^2 + \frac{1}{q} \sum_{p=1}^q (\hat{z}_p)^2}} \end{bmatrix} \tag{37}$$

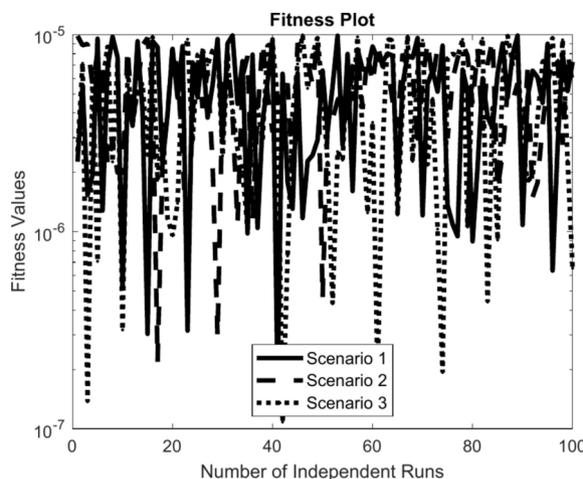
### Result and discussion

Traditional methods for solving the Lorenz differential equations, such as analytical and numerical often face several limitations. These approaches require small time steps to maintain stability and accuracy, making them computationally intensive and time-consuming, especially for highly sensitive chaotic systems like the Lorenz equations. Furthermore, traditional methods may struggle to capture long-term dynamics due to the nonlinear chaotic systems of these differential equations, which can lead to cumulative errors that grow over time, distorting predictions. Intelligent based approaches especially physics informed machine learning to handle these issues by learning the dynamics more efficiently and generalizing across different scenarios.

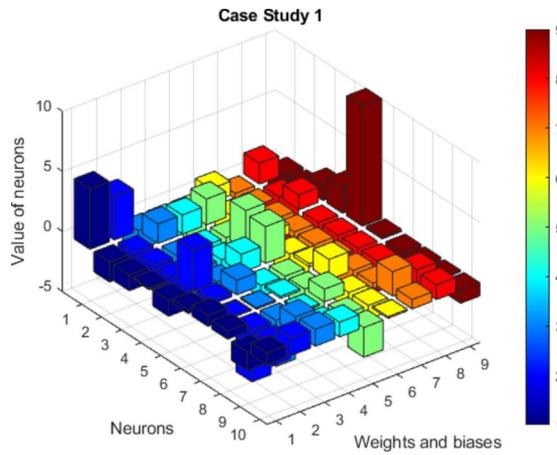
This section presents a detailed discussion of the Lorenz model results. Statistical metrics used the LENNs based results to validate the reliability and suitability of the proposed approach. The suggested findings for addressing the Lorenz model with LENNs-FA-AOA are displayed through relevant graphical and numerical representations to assess convergence and accuracy. The feed-forward LENN variables for 10 neurons are determined by applying FA and AOA coding across multiple runs in the Lorenz model. We are tested the proposed LENN-FA-AOA intelligent method for three (03) distinct scenarios. The fitness/ error function are evaluated through one hundred cycles to check the reliability convergence; the best values of each runs are presented in Fig. 2 for each scenario.

Figures 3, 4, 5 illustrates the optimal LENN parameters  $[w_x, w_y, w_z, \beta_x, \beta_y, \beta_z, \phi_x, \phi_y, \phi_z]$  generated by FA-AOA within the range -10 to 10. All of which are used in the fitness value to represent quantitative measurements for  $x(t)$ ,  $y(t)$  and  $z(t)$  among ten neurons to generate the outcomes of the Lorenz model. Figures 6, 7, 8 present the graphical representations of reference solution that obtained from Adam numerical approach and best and mean results based on LENN-FA-AOA across 100 independent runs, incorporating 10 neurons.

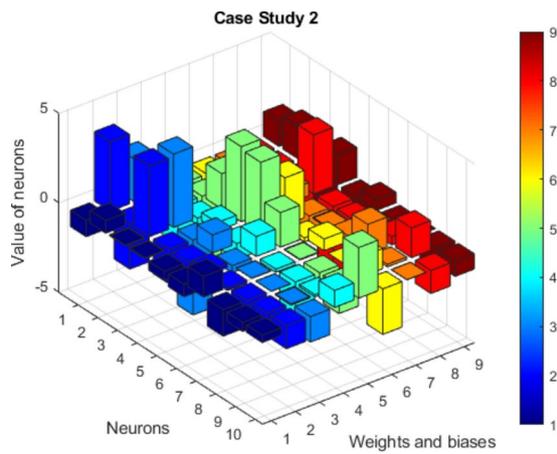
Tables 2, 3, 4 provide absolute error (AE) that compares reference results with LENN-FA-AOA for  $x(t)$ ,  $y(t)$  and  $z(t)$ . In scenario 1, 2 and 3 the initial condition and constant of the Lorenz model are  $[\rho = 0.1, \sigma = 0.2, R = -0.3, x(0) = 0.5, y(0) = 0.5, z(0) = 0.5], [\rho = -0.4, \sigma = 0.2, R = 0.3, x(0) = 0.4, y(0) = 0.4, z(0) = 0.4]$  and  $[\rho = -0.4, \sigma = 0.2, R = 0.3, x(0) = 0.3, y(0) = 0.3, z(0) = 0.3]$ . In scenario 1, the AE values are suitable for all class of model  $(x(t), y(t)$  and  $z(t))$  range between  $3.22 \times 10^{-5}$  to  $3.97 \times 10^{-6}$ ,  $2.47 \times 10^{-5}$  to  $3.06 \times 10^{-7}$  and  $4.24 \times 10^{-5}$  to  $6.82 \times 10^{-7}$ . The AE values for scenario 2 are ranging between  $1.26 \times 10^{-5}$  to  $7.27 \times 10^{-7}$ ,  $1.26 \times 10^{-5}$  to  $7.27 \times 10^{-8}$  and  $4.56 \times 10^{-5}$  to  $1.04 \times 10^{-6}$ . The AE are ranging between  $1.75 \times 10^{-5}$  to  $2.11 \times 10^{-7}$ ,  $5.17 \times 10^{-5}$  to  $2.59 \times 10^{-6}$  and  $3.83 \times 10^{-5}$  to  $4.63 \times 10^{-7}$  for Lorenz model scenario 3. The results obtained from LENN-FA-AOA that showed the proposed approach is convergence and efficient (Figs. 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26).



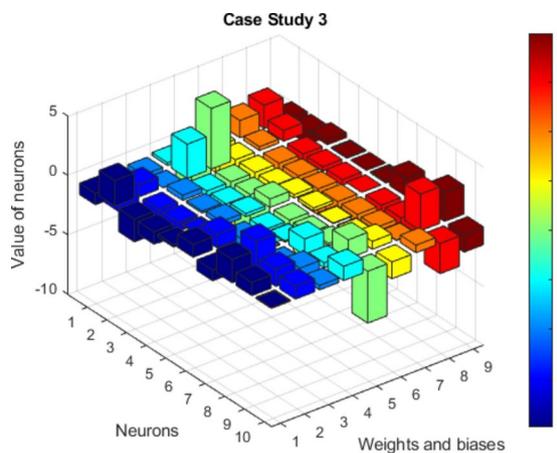
**Fig. 2.** Convergence analysis of fitness function for three different scenarios with 10 neurons of LENN-FA-AOA.



**Fig. 3.** Optimized hyper parameters of LENN through FA-AOA for scenario 1.



**Fig. 4.** Optimized hyper parameters of LENN through FA-AOA for scenario 2.



**Fig. 5.** Optimized hyper parameters of LENN through FA-AOA for scenario 3.

Furthermore, to check the deep understanding of the convergence and efficiency for proposed LENN-FA-AOA methodology to compute mean square error (MSE) and Theil's inequality coefficient (TIC). Figures 9, 15, 21 show the convergence results of MSE for all model dependent parameters using LENN-FA-AOA. For convergence, MSE values range approximately between  $10^{-6}$  and  $10^{-9}$  for  $x(t)$ ,  $10^{-6}$  and  $10^{-9}$  for  $y(t)$ ,  $10^{-6}$

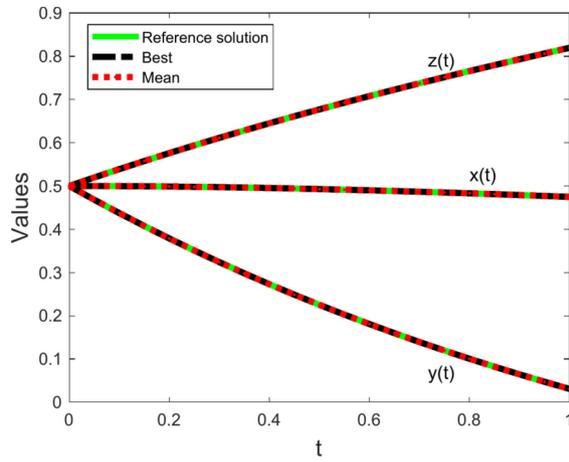


Fig. 6. Comparative analysis between the reference and LENN-FA-AOA solutions for scenario 1.

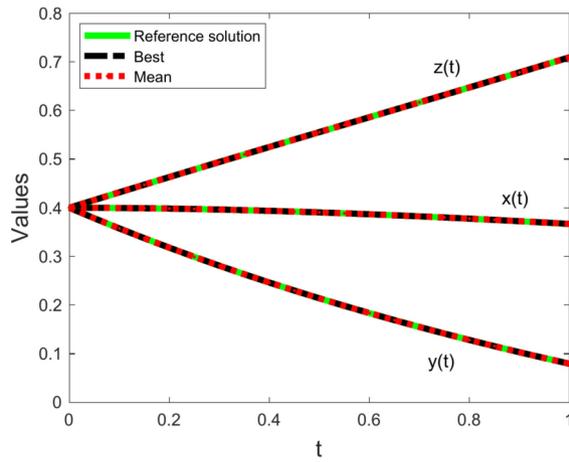


Fig. 7. Comparative analysis between the reference and LENN-FA-AOA solutions for scenario 2.

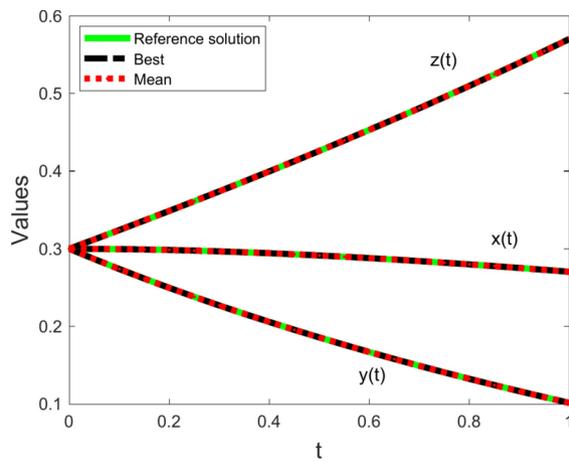


Fig. 8. Comparative analysis between the reference and LENN-FA-AOA solutions for scenario 3.

$t$	$x(t)$	$y(t)$	$z(t)$	$\hat{x}(t)$	$\hat{y}(t)$	$\hat{z}(t)$	$AE_{x(t)}$	$AE_{y(t)}$	$AE_{z(t)}$
0	0.5	0.5	0.5	0.499992	0.499983	0.500002	7.71E-06	1.68E-05	1.95E-06
0.1	0.499683	0.437197	0.538997	0.49969	0.437194	0.539015	6.24E-06	2.68E-06	1.76E-05
0.2	0.498766	0.378583	0.576076	0.498773	0.378577	0.576093	6.43E-06	5.91E-06	1.7E-05
0.3	0.497295	0.323875	0.611371	0.497316	0.323852	0.611368	2.09E-05	2.25E-05	2.84E-06
0.4	0.495312	0.272811	0.645015	0.495303	0.272791	0.645019	8.57E-06	1.99E-05	4.37E-06
0.5	0.492858	0.225152	0.677141	0.492847	0.22515	0.677172	1.02E-05	1.22E-06	3.04E-05
0.6	0.48997	0.180677	0.70788	0.489993	0.180683	0.707912	2.31E-05	6.52E-06	3.15E-05
0.7	0.486683	0.139181	0.737362	0.486711	0.139173	0.737361	2.78E-05	8.78E-06	6.82E-07
0.8	0.48303	0.100474	0.765711	0.483022	0.10045	0.765692	8.4E-06	2.47E-05	1.87E-05
0.9	0.479042	0.064378	0.793051	0.479031	0.064368	0.79307	1.04E-05	1.02E-05	1.88E-05
1	0.474746	0.030726	0.819501	0.474768	0.030728	0.819537	2.16E-05	2.49E-06	3.59E-05

**Table 2.** Comparative analysis with absolute error of  $x(t)$ ,  $y(t)$  and  $z(t)$  between reference and LENN-FA-AOA solution for scenario 1.

$t$	$x(t)$	$y(t)$	$z(t)$	$\hat{x}(t)$	$\hat{y}(t)$	$\hat{z}(t)$	$AE_{x(t)}$	$AE_{y(t)}$	$AE_{z(t)}$
0	0.4	0.4	0.4	0.400001	0.399997	0.399992	1.23E-06	3.21E-06	7.73E-06
0.1	0.399573	0.357514	0.431772	0.399579	0.357509	0.431791	6.28E-06	4.62E-06	1.95E-05
0.2	0.398342	0.317882	0.463138	0.398361	0.317881	0.463137	1.94E-05	8.59E-07	1.04E-06
0.3	0.396377	0.280865	0.494175	0.396391	0.280858	0.494162	1.44E-05	6.76E-06	1.27E-05
0.4	0.393742	0.246248	0.524965	0.393741	0.246238	0.52498	7.23E-07	1.03E-05	1.49E-05
0.5	0.390496	0.213839	0.555592	0.390488	0.213834	0.555624	8.42E-06	4.6E-06	3.25E-05
0.6	0.386693	0.183464	0.586142	0.386692	0.183464	0.586152	7.04E-07	7.27E-08	1.03E-05
0.7	3.82E-01	1.55E-01	6.17E-01	3.82E-01	1.55E-01	6.17E-01	1.42E-05	5.36E-06	2.27E-05
0.8	0.377611	0.128206	0.647359	0.37763	0.128193	0.647341	1.86E-05	1.26E-05	1.76E-05
0.9	0.37242	0.103052	0.678202	0.372424	0.103044	0.678231	4.05E-06	7.25E-06	2.92E-05
1	0.366849	0.079388	0.70932	0.366847	0.079386	0.709349	2.14E-06	2.52E-06	2.89E-05

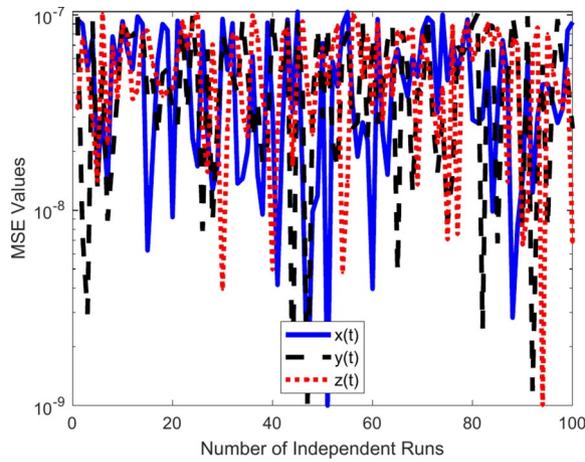
**Table 3.** Comparative analysis with absolute error of  $x(t)$ ,  $y(t)$  and  $z(t)$  between reference and LENN-FA-AOA solution for scenario 2.

$t$	$x(t)$	$y(t)$	$z(t)$	$\hat{x}(t)$	$\hat{y}(t)$	$\hat{z}(t)$	$AE_{x(t)}$	$AE_{y(t)}$	$AE_{z(t)}$
0	0.3	0.3	0.3	0.299994	0.299992	0.3	6.32E-06	8.31E-06	4.63E-07
0.1	0.299609	0.273955	0.324204	0.29961	0.273924	0.324208	1.3E-06	3.1E-05	3.25E-06
0.2	0.298486	0.249687	0.348856	0.298487	0.249635	0.348859	1.1E-06	5.17E-05	3.13E-06
0.3	0.296703	0.227014	0.374017	0.296694	0.226993	0.374003	9.17E-06	2.05E-05	1.32E-05
0.4	0.294325	0.205775	0.399751	0.294314	0.205781	0.399737	1.11E-05	6.19E-06	1.39E-05
0.5	0.291408	0.185829	0.426128	0.291407	0.185826	0.426134	1.8E-06	2.59E-06	6.48E-06
0.6	0.288006	0.167047	0.453219	0.288012	0.167021	0.453231	5.54E-06	2.66E-05	1.19E-05
0.7	0.284166	0.149318	0.481097	0.284166	0.149287	0.481081	2.11E-07	3.15E-05	1.63E-05
0.8	0.279929	0.132543	0.509841	0.279916	0.132529	0.509802	1.28E-05	1.36E-05	3.83E-05
0.9	0.275335	0.116633	0.539529	0.275317	0.116628	0.539526	1.75E-05	4.47E-06	3.25E-06
1	0.270418	0.101509	0.570244	0.270411	0.101495	0.570263	6.89E-06	1.39E-05	1.84E-05

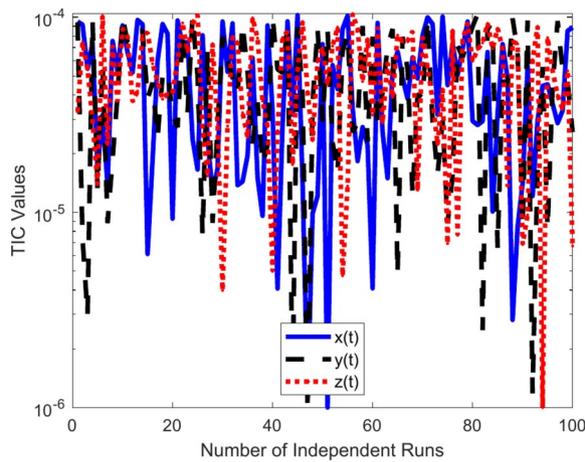
**Table 4.** Comparative analysis with absolute error of  $x(t)$ ,  $y(t)$  and  $z(t)$  between reference and LENN-FA-AOA solution for scenario 3.

and  $10^{-9}$  for  $z(t)$ , respectively. The MSE values across multiple cycles/runs are illustrated in Figs. 11, 17, 23 for different scenarios and the ranging of MSE normal distributions are from  $10^{-6}$  to  $10^{-7}$  approximately and are represented in Figs. 12, 18, 24 for all scenarios.

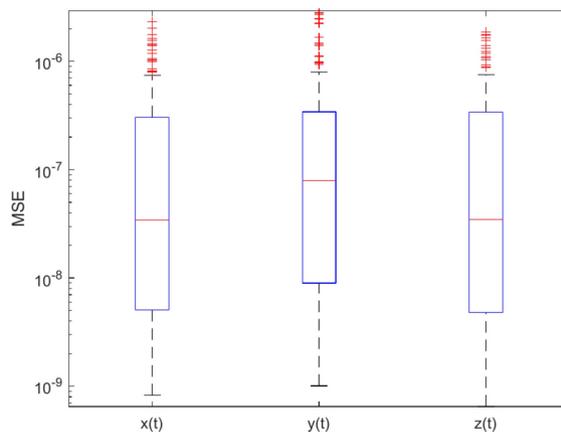
Figures 10, 16, 22 show the convergence results of TIC for all model classes using LENN-FA-AOA. For convergence, TIC values range approximately between  $10^{-4}$  and  $10^{-6}$  for  $x(t)$ ,  $10^{-4}$  and  $10^{-6}$  for  $y(t)$ ,  $10^{-4}$  and  $10^{-6}$  for  $z(t)$ , respectively. The TIC values across multiple cycles/runs are illustrated in Figs. 12, 18, 24 for different scenarios and the ranging of MSE normal distributions are from  $10^{-3}$  to  $10^{-4}$  approximately and are



**Fig. 9.** Convergence analysis of MSE for scenarios 1 with 10 neurons of LENN-FA-AOA.

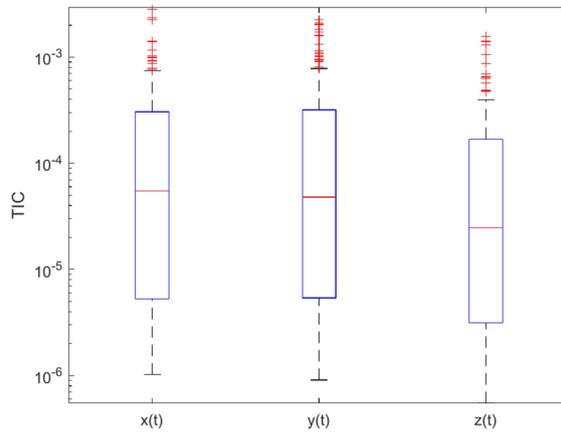


**Fig. 10.** Convergence analysis of TIC for scenarios 1 with 10 neurons of LENN-FA-AOA.

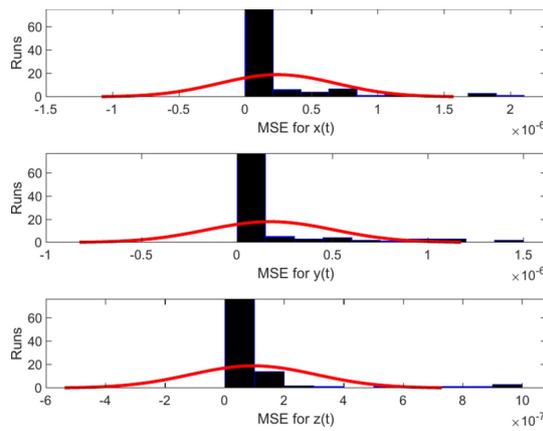


**Fig. 11.** Convergence value of MSE for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 1.

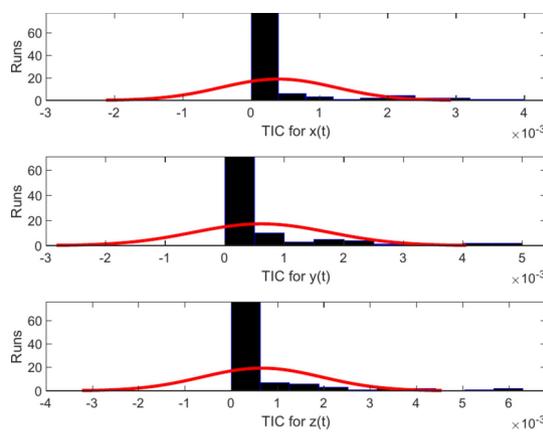
represented in Figs. 13, 19, 25 for all scenarios. The intelligent unsupervised learning approach LENN-FA-AOA is used to train the weights of LENN through a FA-AOA hybrid scheme, assessing time of execution, generations/ iterations, and function counts in the Lorenz model solution using 10 neurons. Figures 14, 20, 26 present the convergence value of TIC using histograms for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA for all



**Fig. 12.** Convergence value of TIC for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 1.

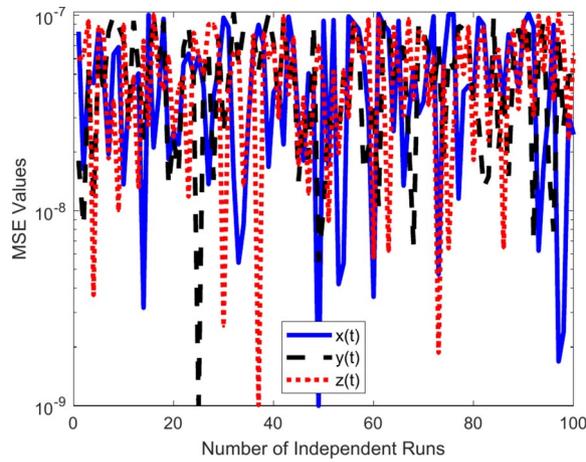


**Fig. 13.** Convergence value of MSE through histogram across multiple runs for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 1.

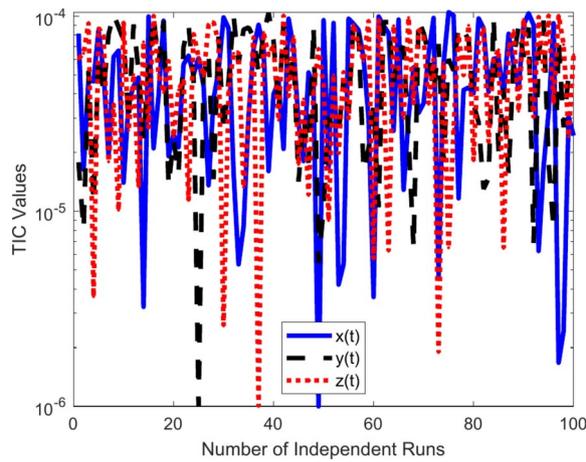


**Fig. 14.** Convergence value of TIC through histogram across multiple runs for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 1.

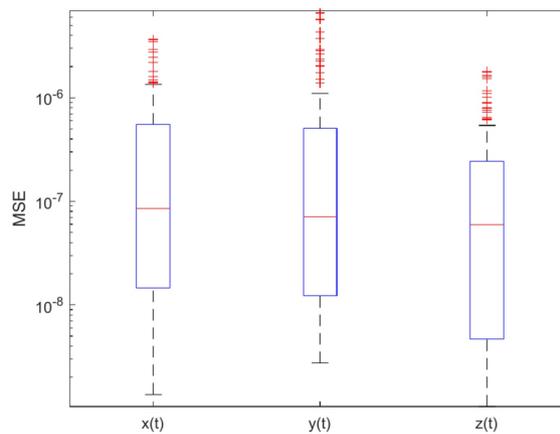
three scenarios. The statistical analysis was conducted across multiple runs to evaluate the maximum, average, minimum, and standard deviation values, as tabulated in Tables 5, 6, 7, for  $x(t)$ ,  $y(t)$  and  $z(t)$  for scenario 1. This analysis provides a comprehensive assessment of the variability and performance trends across different conditions. Furthermore, the optimization of LeNN hyperparameters, specifically the weights and biases, was



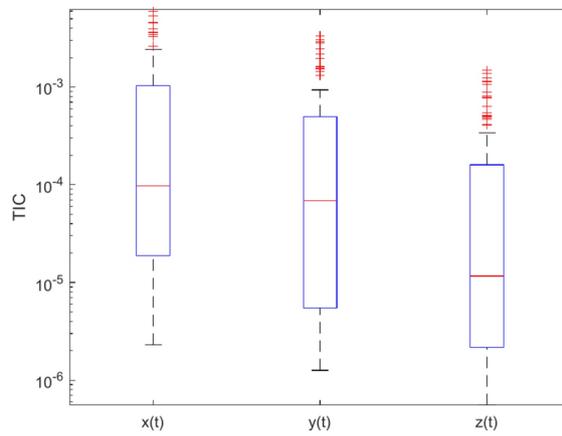
**Fig. 15.** Convergence analysis of MSE for scenarios 2 with 10 neurons of LENN-FA-AOA.



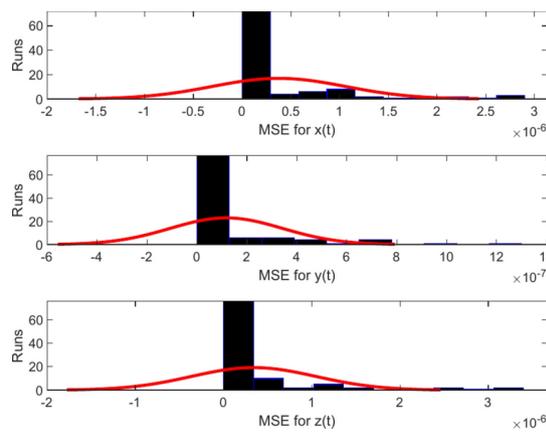
**Fig. 16.** Convergence analysis of TIC for scenarios 2 with 10 neurons of LENN-FA-AOA.



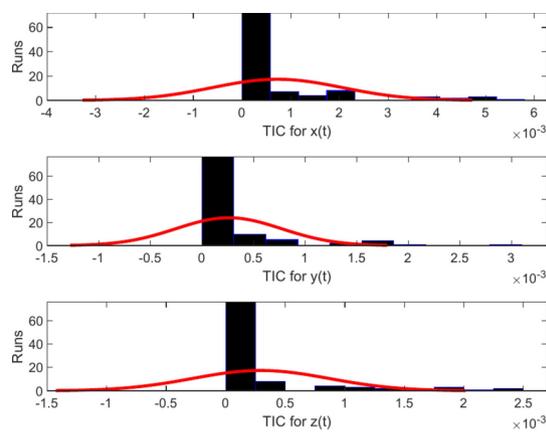
**Fig. 17.** Convergence value of MSE for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 2.



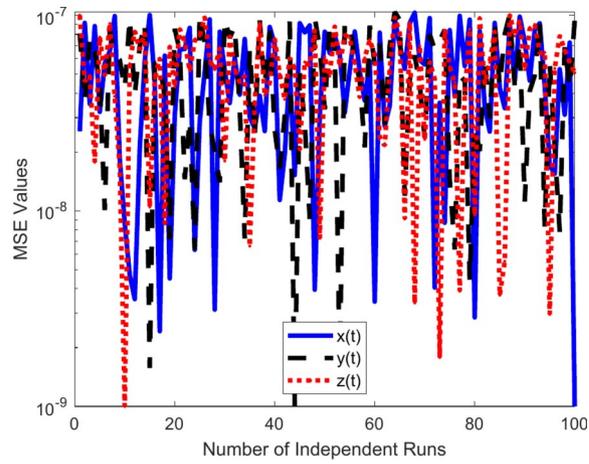
**Fig. 18.** Convergence value of TIC for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 2.



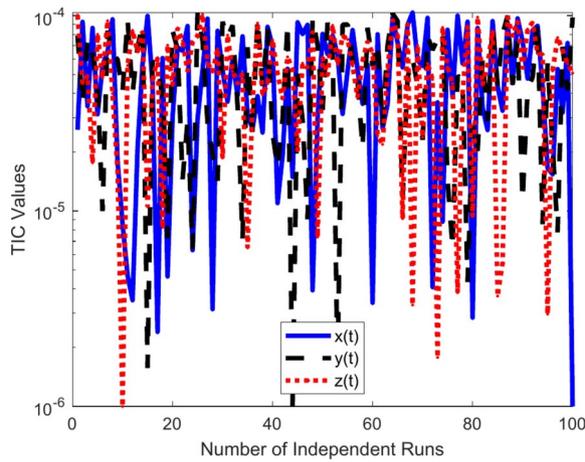
**Fig. 19.** Convergence value of MSE through histogram across multiple runs for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 2.



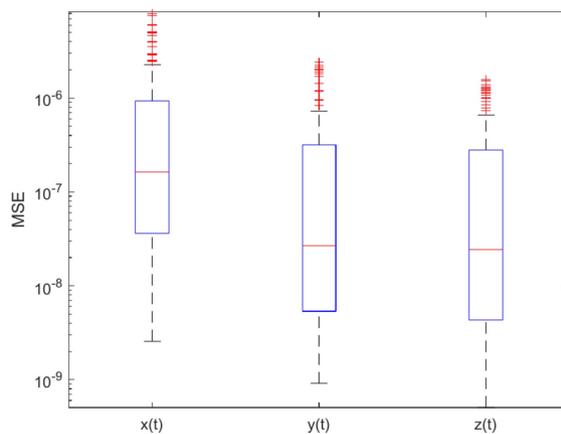
**Fig. 20.** Convergence value of TIC through histogram across multiple runs for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 2.



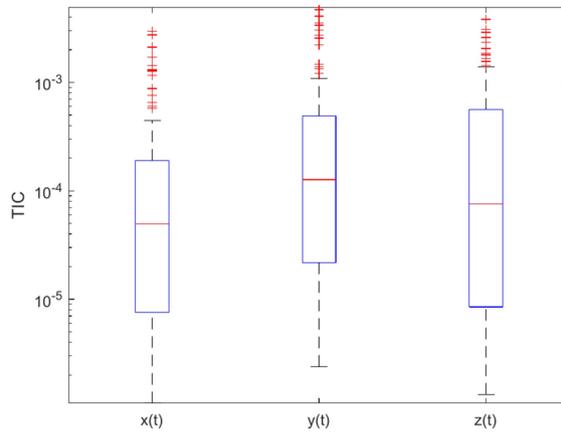
**Fig. 21.** Convergence analysis of MSE for scenarios 3 with 10 neurons of LENN-FA-AOA.



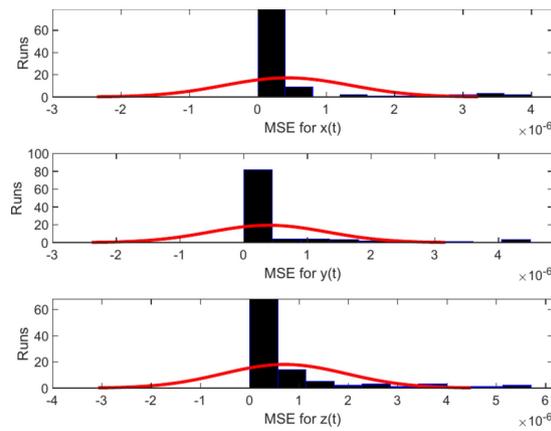
**Fig. 22.** Convergence analysis of TIC for scenarios 3 with 10 neurons of LENN-FA-AOA.



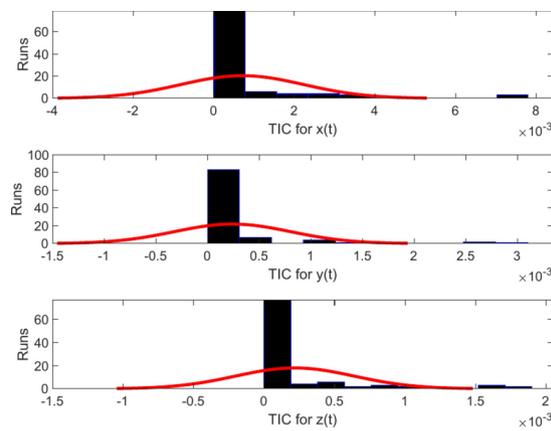
**Fig. 23.** Convergence value of MSE for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 3.



**Fig. 24.** Convergence value of TIC for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 3.



**Fig. 25.** Convergence value of MSE through histogram across multiple runs for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 3.



**Fig. 26.** Convergence value of TIC through histogram across multiple runs for  $x(t)$ ,  $y(t)$  and  $z(t)$  with 10 neurons of LENN-FA-AOA of scenario 3.

$t$	<i>min</i>	<i>max</i>	<i>average</i>	<i>S.D</i>
0	1.69E-08	1.53E-04	7.67E-05	1.67E-04
0.1	2.93E-08	6.08E-04	3.04E-04	8.31E-04
0.2	8.04E-08	3.03E-04	1.51E-04	2.76E-04
0.3	3.86E-08	1.61E-04	8.05E-05	6.86E-04
0.4	6.17E-08	7.88E-04	3.94E-04	2.07E-04
0.5	5.58E-08	8.26E-04	4.13E-04	8.08E-04
0.6	3.41E-08	4.22E-04	2.11E-04	3.72E-04
0.7	2.93E-08	6.55E-04	3.28E-04	4.69E-04
0.8	2.79E-08	1.41E-04	7.07E-05	5.65E-04
0.9	4.57E-08	2.26E-04	1.13E-04	9.18E-04
1	9.45E-08	5.35E-04	2.68E-04	4.48E-05

**Table 5.** Statistical analysis of absolute error of  $x(t)$  between reference and LENN-FA-AOA solution for scenario 1 across multiple runs.

$t$	<i>min</i>	<i>max</i>	<i>average</i>	<i>S.D</i>
0	2.36E-08	8.96E-04	4.48E-04	9.50E-04
0.1	3.97E-08	3.92E-04	1.96E-04	3.59E-04
0.2	5.35E-08	7.55E-04	3.78E-04	7.45E-04
0.3	5.17E-08	3.08E-04	1.54E-04	3.08E-04
0.4	8.88E-08	7.81E-04	3.91E-04	9.93E-04
0.5	8.96E-08	9.10E-04	4.55E-04	2.11E-04
0.6	2.47E-09	6.34E-04	3.17E-04	9.43E-04
0.7	4.06E-08	2.94E-04	1.47E-04	1.62E-04
0.8	3.56E-08	7.23E-04	3.61E-05	1.08E-05
0.9	5.95E-08	9.30E-04	4.65E-04	3.47E-04
1	4.48E-08	8.14E-04	4.29E-07	8.51E-04

**Table 6.** Statistical analysis of absolute error of  $y(t)$  between reference and LENN-FA-AOA solution for scenario 1 across multiple runs.

$t$	<i>min</i>	<i>max</i>	<i>average</i>	<i>S.D</i>
0	5.68E-10	8.98E-04	4.49E-04	2.07E-04
0.1	8.91E-08	5.96E-04	2.98E-04	6.93E-05
0.2	3.99E-08	5.71E-04	2.86E-04	5.57E-04
0.3	4.76E-08	8.85E-04	4.43E-04	8.00E-04
0.4	4.25E-08	5.39E-04	2.69E-04	1.84E-04
0.5	9.90E-08	6.99E-04	3.50E-04	4.03E-04
0.6	8.59E-09	3.79E-04	1.90E-04	2.23E-04
0.7	4.57E-08	7.85E-04	3.93E-04	9.25E-04
0.8	3.15E-09	8.44E-04	4.22E-05	9.59E-04
0.9	5.68E-08	8.64E-04	4.32E-04	2.82E-04
1	4.85E-08	9.38E-04	4.69E-04	8.85E-04

**Table 7.** Statistical analysis of absolute error of  $z(t)$  between reference and LENN-FA-AOA solution for scenario 1 across multiple runs.

carried out using various optimizers and comparative evaluation of these optimizers, highlighting their tuning effectiveness, is presented in Table 8.

## Conclusion

LENN-FA-AOA is an innovative stochastic numerical solution based on a Legendre polynomials artificial neural network, enhanced by heuristic optimization algorithms. This proposed LENN-FA-AOA approach converges quickly and is capable of addressing a wide range of chaotic systems, including linear/nonlinear, singular/nonsingular, and stiff systems. The quantitative effectiveness of LENN-FA-AOA is validated through

Algorithm	Scenario 1		Scenario 2		Scenario 3	
	Fitness values	MSE	Fitness values	MSE	Fitness values	MSE
FA	8.11E-05	2.22E-07	1.32E-05	4.34E-07	3.54E-05	6.76E-07
AOA	2.23E-05	3.23E-07	3.54E-05	5.53E-07	4.37E-05	3.78E-07
PSO	4.01E-05	6.56E-07	7.27E-05	6.78E-07	3.34E-05	8.13E-07
WCA	1.14E-05	7.64E-07	6.97E-05	5.45E-07	1.22E-05	6.49E-07
GA	2.23E-05	8.76E-07	7.63E-05	8.98E-07	5.54E-05	6.09E-07
ABC	2.82E-05	5.45E-07	3.15E-05	3.34E-07	3.45E-05	2.05E-07
FA-AOA	1.01E-06	2.12E-09	1.67E-06	2.43E-09	3.69E-06	1.90E-09

**Table 8.** Comparison of different optimizers.

numerical testing, achieving a high level of precision that confirms the robustness of the proposed mathematical framework. By combining FA optimization hybrid with AOA used for the model predictive accuracy is enhanced, providing an effective tool for analysing the Lorenz model. Utilized neural networks with ten neurons, the proposed LENN-FA-AOA statistical approach efficiently examined the performance of the proposed method. The absolute errors values obtained from LENN-FA-AOA with reference solution ranging from  $3.22 \times 10^{-5}$  to  $3.06 \times 10^{-7}$ ,  $4.56 \times 10^{-5}$  to  $7.27 \times 10^{-8}$  and  $5.17 \times 10^{-5}$  to  $2.11 \times 10^{-7}$ . In one hundred independent trials/runs, statistical measures including the absolute error, mean square mean and TIC values further validate the reliability of the LENN-FA-AOA method. The limitation of the proposed approach is reduced prediction accuracy when the chaotic model has a high dimensionality like 5D, 6D and 7D.

### Future work

In figure the proposed hybrid algorithm will be used for cyber security analysis based on differential equations. Also, various neuro-evolutionary intelligent computing paradigms will be developed to analyze fractional differential order based Lorenz system. Additionally to be used the quantum algorithms to enhance the accuracy of the Lorenz model.

### Data availability

All data generated or analysed during this study are included in this published article.

Received: 3 January 2025; Accepted: 24 February 2025

Published online: 24 April 2025

### References

- Kudryashov, N. A. Analytical solutions of the Lorenz system. *Regul. Chaotic Dyn.* **20**, 123–133 (2015).
- Algaba, A., Fernández-Sánchez, F., Merino, M. & Rodríguez-Luis, A. J. Analysis of the T-point-Hopf bifurcation in the Lorenz system. *Commun. Nonlinear Sci. Numer. Simul.* **22**(1–3), 676–691 (2015).
- Barrio, R. & Serrano, S. Bounds for the chaotic region in the Lorenz model. *Phys. D* **238**(16), 1615–1624 (2009).
- Wu, K. & Zhang, X. Darboux polynomials and rational first integrals of the generalized Lorenz systems. *Bull. Sci. Math.* **136**(3), 291–308 (2012).
- Algaba, A., Domínguez-Moreno, M. C., Merino, M. & Rodríguez-Luis, A. J. Double-zero degeneracy and heteroclinic cycles in a perturbation of the Lorenz system. *Commun. Nonlinear Sci. Numer. Simul.* **111**, 106482 (2022).
- Doedel, E. J., Krauskopf, B. & Osinga, H. M. Global invariant manifolds in the transition to preturbulence in the Lorenz system. *Indag. Math.* **22**(3–4), 222–240 (2011).
- Malathy R, Saratha SR. Solving lorenz system of equation by Laplace homotopy analysis method. In *Proceedings of the First International Conference on Combinatorial and Optimization, ICCAP 2021, December 7–8 2021, Chennai, India.* (2021).
- Klöwer, M., Coveney, P. V., Paxton, E. A. & Palmer, T. N. Periodic orbits in chaotic systems simulated at low precision. *Sci. Rep.* **13**(1), 11410 (2023).
- Brown, P. N. & Saad, Y. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.* **11**(3), 450–481 (1990).
- Drazin, P. G. *Nonlinear Systems (No. 10)* (Cambridge University Press, 1992).
- Kumar, D., Singh, J. & Baleanu, D. A hybrid computational approach for Klein-Gordon equations on Cantor sets. *Nonlinear Dyn.* **87**, 511–517 (2017).
- He, J. H. A coupling method of a homotopy technique and a perturbation technique for non-linear problems. *Int. J. Non-linear Mech.* **35**(1), 37–43 (2000).
- Biazar, J. & Eslami, M. A new homotopy perturbation method for solving systems of partial differential equations. *Comput. Math. Appl.* **62**(1), 225–234 (2011).
- Amir, M., Ashraf, A. & Haider, J. A. The variational iteration method for a pendulum with a combined translational and rotational system. *Acta Mech. Autom.* **18**(1), 48–54 (2024).
- Koochi, A. & Goharimanesh, M. Nonlinear oscillations of CNT nano-resonator based on nonlocal elasticity: The energy balance method. *Rep. Mech. Eng.* **2**(1), 41–50 (2021).
- Ahmed, M., Gamal, M., Ismail, I. M. & El-Din, E. H. An AI-based system for predicting renewable energy power output using advanced optimization algorithms. *J. Artif. Intell. Metaheuristics* **8**, 1–8 (2024).
- Samadi, H., Mohammadi, N. S., Shamoushaki, M., Asadi, Z. & Ganji, D. D. An analytical investigation and comparison of oscillating systems with nonlinear behavior using AGM and HPM. *Alex. Eng. J.* **61**(11), 8987–8996 (2022).
- Adomian G. *Solving frontier problems of physics: the decomposition method.* Springer Science & Business Media. (2013).
- Yang, L., Zhang, D. & Karniadakis, G. E. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM J. Sci. Comput.* **42**(1), A292–A317 (2020).

20. Raissi, M. Forward–backward stochastic neural networks: deep learning of high-dimensional partial differential equations. In *Peter Carr Gedenkschrift: Research Advances in Mathematical Finance* 637–655 (World Scientific, 2024).
21. Mattheakis, M., Sondak, D., Dogra, A. S. & Protopapas, P. Hamiltonian neural networks for solving equations of motion. *Phys. Rev. E* **105**(6), 065305 (2022).
22. Piscopo, M. L., Spannowsky, M. & Waite, P. Solving differential equations with neural networks: Applications to the calculation of cosmological phase transitions. *Phys. Rev. D* **100**(1), 016002 (2019).
23. Hagge T, Stinis P, Yeung E, Tartakovsky AM. Solving differential equations with unknown constitutive relations as recurrent neural networks. [arXiv:1710.02242](https://arxiv.org/abs/1710.02242). (2017).
24. Mattheakis M, Protopapas P, Sondak D, Di Giovanni M, Kaxiras E. Physical symmetries embedded in neural networks. [arXiv:1904.08991](https://arxiv.org/abs/1904.08991). (2019).
25. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
26. Han, J., Jentzen, A. & E, W., Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci.* **115**(34), 8505–8510 (2018).
27. Buhl, J. et al. As predicted by statistical models swarms of locusts undergo rapid transitions from disordered to ordered collective motion as their density increases. *Science* **6**, 1402–1406 (2006).
28. Blum, C. & Li, X. Swarm intelligence in optimization. In *Swarm Intelligence: Introduction and Applications* 43–85 (Springer, 2008).
29. Aslam, M. N., Riaz, A., Shaukat, N., Aslam, M. W. & Alhamzi, G. Machine learning analysis of heat transfer and electroosmotic effects on multiphase wavy flow: A numerical approach. *Int. J. Numer. Methods Heat Fluid Flow* **34**(1), 150–177 (2024).
30. Yang XS. Metaheuristic optimization: algorithm analysis and open problems. In *International symposium on experimental algorithms*. Springer Berlin Heidelberg. (2011).
31. Apostolopoulos, T. & Vlachos, A. Application of the firefly algorithm for solving the economic emissions load dispatch problem. *Int. J. Comb.* **2011**(1), 523806 (2011).
32. Yang, X. S. Firefly algorithms for multimodal optimization. In *International symposium on Stochastic Algorithms* 169–178 (Springer, 2009).
33. Fister, I., Fister, I. Jr., Yang, X. S. & Brest, J. A comprehensive review of firefly algorithms. *Swarm Evol. Comput.* **13**, 34–46 (2013).
34. Aslam, M. N. et al. Neuro-computing solution for Lorenz differential equations through artificial neural networks integrated with PSO-NNA hybrid meta-heuristic algorithms: a comparative study. *Sci. Rep.* **14**(1), 7518 (2024).
35. Rajchakit, G., Agarwal, P. & Ramalingam, S. *Stability Analysis of Neural Networks* 1–404 (Springer, 2021).
36. El-kenawy, E.-S.M. et al. Greylag Goose Optimization: Nature-inspired optimization algorithm. *Expert Syst. Appl.* **238**(122147), 122147 (2024).
37. El-Sayed, E., Eid, M. M. & Abualigah, L. Machine learning in public health forecasting and monitoring the zika virus. *Metaheuristic Optim. Rev.* **1**, 1–11 (2024).
38. Hashim, F. A., Hussain, K., Houssein, E. H., Mabrouk, M. S. & Al-Atabany, W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **51**, 1531–1551 (2021).
39. Chauhan, S., Vashishtha, G., Zimroz, R. & Kumar, R. A crayfish optimised wavelet filter and its application to fault diagnosis of machine components. *Int. J. Adv. Manuf. Technol.* **135**(3–4), 1825–1837. <https://doi.org/10.1007/s00170-024-14626-0> (2024).
40. Chauhan, S., Vashishtha, G., Zimroz, R., Kumar, R. & Kumar Gupta, M. Optimal filter design using mountain gazelle optimizer driven by novel sparsity index and its application to fault diagnosis. *Appl. Acoustics* **225**(110200), 110200. <https://doi.org/10.1016/j.apacoust.2024.110200> (2024).

## Author contributions

Khalid Masood and Muhammed Naeen: contributed to the design and writing of the article, Muhammed Arshad: data collection and Analysis of the data, Abu Bakkar: performed simulations and experiments, Mohammed A. Alghamdi and Sultan H. Almotiri participated in the data analysis and in the writing, Khalid Masood: responsible for the revision of the full text. All authors reviewed the manuscript.

## Funding

This work was funded by Deputyship of Research and Innovation, Ministry of Education in Saudi Arabia, through the project number IFP22UQU4250002DSR216.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to K.M.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.