Applied Soft Computing Journal 151 (2024) 111163

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

F-TLBO-ID: Fuzzy fed teaching learning based optimisation algorithm to predict the number of k-barriers for intrusion detection

Abhilash Singh [a,*], Seyed Muhammad Hossein Mousavi [b,1], Jaiprakash Nagar [c,2]

[a] Fluvial Geomorphology and Remote Sensing Laboratory, Indian Institute of Science Education and Research Bhopal, India
[b] Independent Scientist, Tehran, Iran
[c] Subir Chowdhury School of Quality and Reliability, Indian Institute of Technology Kharagpur, India

ARTICLE INFO

Dataset link: https://abhilashsingh.net/codes.html

Keywords:
Intrusion detection
Nature-inspired regression
Monte-Carlo simulations
Barrier coverage
Synthetic datasets

ABSTRACT

Ensuring fast and efficient Intrusion Detection and Prevention (IDP) at international borders is crucial for maintaining security and safeguarding nations. In this study, we propose an innovative approach that harnesses the power of machine learning and Wireless Sensor Networks (WSNs) to achieve faster and more accurate IDP. Our novel Fuzzy fed Teaching Learning Based Optimisation regression algorithm (F-TLBO-ID) revolutionises the prediction of the required number of k-barriers for rapid IDP. To develop and validate our approach, we synthetically generated pertinent features using Monte-Carlo simulations. These features encompass essential parameters such as the concerned region's area, effective transmission range, effective sensing range, number of sensor nodes, and the fading parameter. Training the F-TLBO-ID algorithm with these features yielded exceptional results, accurately predicting the required number of k-barriers with an impressive correlation coefficient (R = 0.99), minimal Root Mean Square Error (RMSE = 11.32), and negligible bias (−3.66). To benchmark the performance of our F-TLBO-ID algorithm, we conducted comprehensive comparisons with fine-tuned benchmark algorithms, including AutoML, GPR, GRNN, RF, RNN, SVM, and ANN. Additionally, we evaluated the algorithm against 11 different variants of nature-inspired algorithms. Remarkably, our F-TLBO-ID algorithm outperformed all these methods in terms of accuracy, firmly establishing its superiority. Finally, we validated the performance of the F-TLBO-ID algorithm using publicly available datasets. The results were highly satisfactory, exhibiting a strong correlation coefficient (R = 0.84), acceptable RMSE (36.24), and minimal bias (−7.17). This study offers a robust and reliable algorithm to predict the required barriers for fast IDP, surpassing the accuracy of existing benchmark algorithms. By implementing our proposed algorithm, the efficiency of IDP systems at international borders can be significantly improved, ultimately enhancing security and facilitating smooth border operations.

1. Introduction

Surveillance, intrusion detection, and prevention along international boundaries are significant challenges for most countries. The international boundaries of a country with the neighbouring states may stretch from several hundred to thousands of kilometers, where it is impossible to station military personnel at every position. Consequently, boundary regions are vulnerable to intrusion and other unauthorised entries [1]. The problem at hand can be solved with the help of Wireless Sensor Networks (WSNs).

A WSN is made of many sensors, which can operate independently without the need for any pre-installed infrastructure. Further, these networks communicate in a single or multi-hop manner over wireless channels. They can be installed on the fly easily in remote and/or inaccessible terrain and emergency or hazardous scenarios [2,3]. Consequently, WSNs have a colossal number of civilian as well as military applications such as industrial and environmental monitoring, precision agriculture, patient health monitoring, smart homes, air quality monitoring, battlefield surveillance, coverage mapping, node localisation, reconnaissance, and intrusion detection [4–7].

Intrusion detection is one of the significant applications of WSNs. The various studies conducted on intrusion detection can be categorised into two main categories. In the first type, the studies consider intrusion

* Corresponding author.
E-mail addresses: abhilash.iiserb@gmail.com, sabhilash@iiserb.ac.in (A. Singh), mosavi.a.i.buali@gmail.com, seyed.mousavi@supsi.ch (S.M.H. Mousavi), jpnagar@iitkgp.ac.in (J. Nagar).
[1] Now at Department of innovative technologies, SUPSI, Switzerland.
[2] Now at Eurecom, SophiaTech, Saint Antipolish, 06410, Biot, France.

https://doi.org/10.1016/j.asoc.2023.111163
Received 30 June 2022; Received in revised form 21 November 2023; Accepted 12 December 2023
Available online 19 December 2023

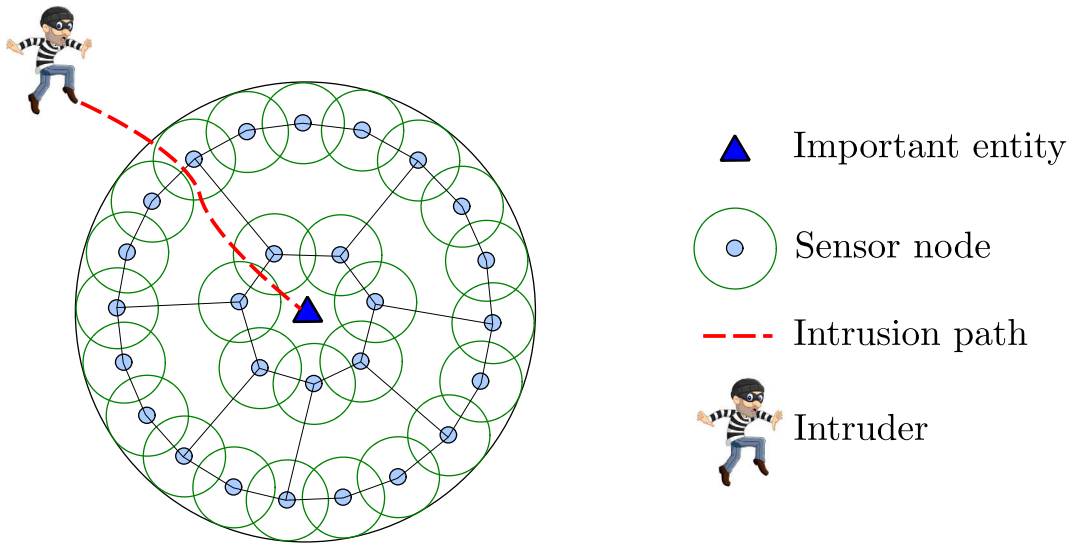

Fig. 1. Illustration of 2-barrier coverage in WSNs.

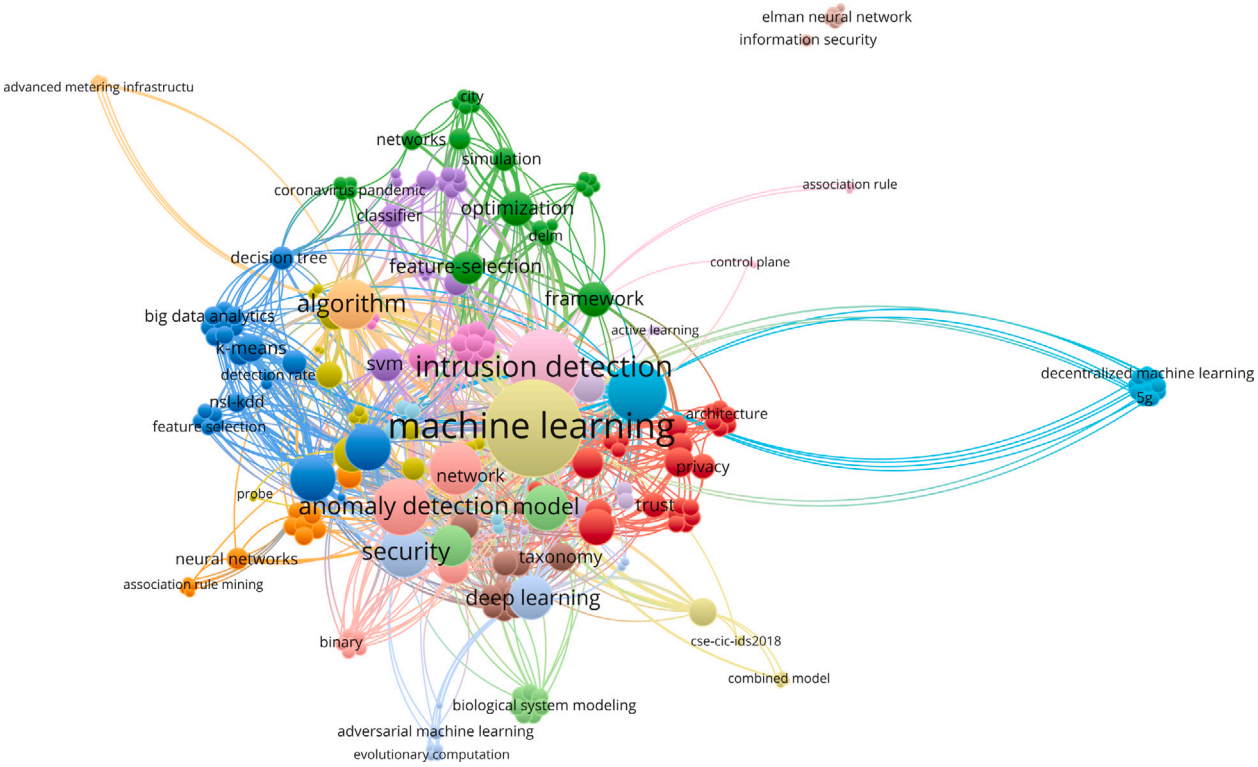

Fig. 2. Bibliometric analysis of the frequently used keywords on the intrusion detection based studies concerning machine learning algorithms. We considered 1573 research articles published in the Web of Science (WoS) database with 512 publications from the year 2021.

detection as one of the desired functions of the system, where it can identify vulnerabilities or compromised sensors and guarantee the desired network response with minimum false alarms. In the second type, the studies consider it as a surveillance and monitoring set-up capable of detecting suspicious activity like intrusion and unauthorised activity in a given Region of Interest (RoI) [8]. The paper in hand focuses on the second category. This study assumes that an important entity is in the centre of a circular RoI. To detect and prevent any possible movement towards the target, sensor barriers are formed for any possible path leading the intruder to the centre of the RoI. The aim is to achieve k-barrier coverage for any intrusion path. Fig. 1 shows such a scenario, where sensors form 2-barrier coverage for all the possible paths from the point of intrusion to the centre of the circular RoI.

Earlier researchers have employed WSNs for surveillance and intrusion detection purposes and found that WSNs are very effective and render a cost-effective solution for intrusion detection and prevention [9–14]. In Keung et al. [15], the authors have developed an analytical framework using mobile sensor networks for intrusion detection and surveillance at border regions. With the help of the developed framework, they examined the impact of various systems variables such as sensor density, sensing range, and Sensor to Intruder Velocity Ratio (SIVR) on the k-barrier coverage probability against the moving intruder. Furthermore, they found that the same k-barrier coverage

performance can be achieved by using fewer mobile sensors than static sensors. In Luo and Zou [9], authors have proposed a barrier constructing algorithm for intrusion detection using WSNs. The proposed algorithm is capable of identifying an illegal as well as a legal intruder in a circular region. In another work [11], authors have presented a sensor scheduling algorithm in which barrier-forming sensors can stay in a sleeping state to save energy consumption. The proposed scheme delivers an energy balance and maximal network lifetime. Further, Ghosh et al. [10] have proposed two routing schemes that render a cost and energy-efficient solution, thus, prolonging the lifetime of WSNs deployed at unattended border regions and other hazardous places. In Huang et al. [12], the authors have formulated an analytical framework that considers sensors' and intruders' mobility patterns for intrusion detection. The derived expression provides a better k-barrier coverage probability than other approaches. Recently, He et al. [13] have proposed a fault-tolerant intrusion detection algorithm to deal with the false information communicated by the faulty sensors. The proposed algorithm can identify almost all the faulty sensors and achieves a lower false alarm rate. Chang et al. [16] proposed a novel approach to achieve high-energy-efficient barrier coverage in underwater WSNs. The approach is based on node alliances, considering the detection and energy consumption modes of underwater WSNs. By investigating the relationship between the number of virtual sensors and energy consumption, they formulated the problem of achieving high energy efficiency as an optimisation problem. Through solving this optimisation problem, they determined the optimal number of physical sensors required to form virtual fences. The construction of barrier coverage in their approach takes into account both high detection probability and low energy consumption. They demonstrated the effectiveness of their proposed strategy in providing high-energy-efficient barrier coverage in underwater WSNs through analytical and simulation studies. More recently, a paper by Fan and Zhai [17] introduced a distributed multi-layer ring barrier coverage algorithm to mitigate the security risks associated with single-layer coverage. The authors devised a distributed adjustment mechanism that operates between multiple layers of barriers, which was integrated with the single-layer ring barrier coverage algorithm to create a novel distributed multi-layer ring barrier coverage algorithm. The effectiveness of the proposed approach was evaluated through numerical simulations conducted by the authors. It is necessary to mention that the above-discussed works provide a great deal of knowledge towards understanding intrusion detection problems, major challenges, issues, and possible solutions. However, validating analytical models is computationally expensive and needs huge financial support and time. Thus, it is essential to devise new methods and approaches that are quick, cheap, and computationally inexpensive for their validation. The problem at hand can be resolved with the help of ML techniques that extract only valuable information by discarding redundant ones; in this way, ML techniques can reduce the computational time from hours to seconds.

We performed bibliometric analysis on the research articles published in the Web of Science (WoS) database that relate machine learning with intrusion detection (see Fig. 2). The frequently used keywords have been marked in Fig. 2 from 1573 research articles. Each circle represents an independent keyword, and the circle's scale (*i.e.,* diameter of the circle) represents the frequency of its occurrence [18]. We can clearly visualise the name of some of the frequently used machine learning algorithms, such as Neural Network (NN), adversarial machine learning, Elman NN, and decentralised machine learning. We observed an exponential increase in the publications concerning the use of the data-driven approach for accurate IDP, with a maximum of 512 research publications from 2021.

The rest of the manuscript has been divided into six sections. Section 2 provides a comprehensive review of the state-of-the-art works relevant to the present study. In Section 3, the system model is presented, and key terms are discussed. Section 4 focuses on the machine learning model, covering aspects such as feature generation, importance calculation, the influence of each feature on the target variable, and the proposed algorithm. The obtained results are presented in Section 5. Section 6 compares these results with benchmark algorithms in terms of accuracy and time-complexity. Additionally, the performance of the proposed algorithm on publicly available datasets is evaluated, and the limitations of the study are highlighted. Finally, the findings of the study are summarised and concluded in Section 7.

2. Related works

Machine learning methodologies have garnered considerable interest within the research community for their efficacy in tackling the time-complexity and computation cost challenges inherent in traditional intrusion detection methods. A plethora of studies has been undertaken to precisely predict the k-barriers, advancing the field of fast intrusion detection. Singh et al. [19] have proposed three different ML algorithms based on Gaussian Process Regression (GPR) to accurately predict the k-coverage probability for intrusion detection using WSNs over a rectangular RoI. Based upon the feature scaling, they have proposed Scale GPR (S-GPR), Centre-mean GPR (C-GPR), and Not standarised (NS-GPR) algorithms. They extracted six predictors: the number of sensors, sensing range, SIVR, Mobile to Static Node Ratio (MSNR), angle of the intrusion path, and the required k synthetically through Monte-Carlo simulations. They performed the predictor importance analysis through the regression tree ensemble approach and found the number of nodes to be the most relevant and the intrusion path angle to be the least relevant feature in predicting the k-coverage probability. They have trained these three algorithms over the extracted datasets. They have reported that the NS-GPR performs better than the S-GPR and C-GPR (with R = 0.85 and RMSE = 0.095). Also, they found that the NS-GPR outperforms all the corresponding variants of Support Vector Regression (*i.e.,* S-SVR, C-SVR, and NS-SVR). In another study, Singh et al. [20] proposed an efficient algorithm based on predictor transformation and scaling. They have proposed an LT-FS-ID algorithm to precisely predict the k-barriers count required for fast IDP using WSNs over a rectangular RoI. They have considered the area of the RoI, number of sensors, transmission range, and sensing range as the potential predictors to predict the number of k-barriers. They extracted these features synthetically through Monte-Carlo simulations considering the binary sensing model. They found the area to be the least relevant feature and the other remaining features (*i.e.,* the number of sensors, transmission range, and sensing range) to be the most pertinent features in predicting the k-barriers. Further, they reported that the proposed algorithms precisely predict the number of k-barriers (with R = 0.98 and RMSE = 6.47). Also, the LT-FS-ID outperforms several benchmark algorithms (*i.e.,* GPR, GRNN, ANN, and RF). However, LT-FS-ID fails to accurately predict k-barriers if any of the input features is not a real positive number. To overcome this limitation, Singh et al. [1] proposed an automated machine learning algorithm, AutoML-ID, to predict the number of k-barriers over a rectangular RoI considering Gaussian distribution node deployment. They only considered the explainable machine learning algorithms (*i.e.,* SVR, GPR, binary decision tree, bagging ensemble learning, boosting ensemble learning, kernel regression, and linear regression model) for executing the automated machine learning module. Similar to the LT-FS-ID case, they have synthetically extracted four features; area of the RoI, number of sensors, transmission range, and sensing range using Monte Carlo simulations considering binary sensing model to train and validate the proposed algorithm. They reported that the transmission range has the highest and the sensing range has the lowest relevancy in predicting the k-barriers for accurate and fast intrusion detection and prevention. They found that the AutoML-ID algorithm performs exceptionally well (with R = 1, RMSE = 0.007, and bias = −0.006) over the unseen datasets. Also, it outperforms several benchmark algorithms

such as Feed-Forward Neural Networks (FFNNs), Recurrent Neural Networks (RNNs), Radial Basis Functions Neural Networks (RBNN), Exact RBNN (ERBNN), and GRNN. In their study, de Campos Souza et al. [21] proposed an explainable Evolving Fuzzy Neural Network (EFNN) model specifically designed to predict the presence of k-barriers accurately. They conducted an evaluation of the proposed model using the LT-FS-ID datasets [20] and reported a root mean square error (RMSE) value of 11.16. Building upon this work, the authors aim to enhance the capabilities of the EFNN model by incorporating the dataset generated from the prediction task. This combined approach not only seeks to achieve precise regression results but also aims to provide valuable insights into the underlying data.

Apart from accurately predicting the k-barriers, researchers have also proposed various solutions to predict the k-barriers coverage probability for fast IDP. Recently, Arora and Pal [22] proposed a deep learning based architecture based on ANN to predict the k-coverage probability for a circular RoI accurately. They also extracted synthetic features through Monte Carlo simulations. They reported that the proposed architecture accurately predicts the k-coverage probability with R = 0.98 and RMSE = 0.07. More recently, Nagar et al. [23] proposed a machine learning model based on Generalised Regression Neural Network (GRNN) to predict the k-coverage probability for a rectangular RoI by considering boundary effects (BEs) and shadowing effects (SEs). They reported that the proposed model accurately mapped the k-coverage probability with R = 0.78 and RMSE = 0.14. These results highlight the effectiveness of their model in accurately estimating the k-coverage probability while considering the influences of boundary effects and shadowing effects.

The existing literature in intrusion detection using machine learning and WSNs has predominantly focused on either utilising explainable machine learning models or black box models. These studies have explored the trade-off between model interpretability and prediction accuracy in the context of intrusion detection. While some researchers have prioritised the development of models that offer explainability and insights into the detection process, others have leaned towards leveraging the predictive power of more complex but less interpretable black box models. However, it is worth noting that none of the aforementioned studies have explored the potential of harnessing the power of coupling a Fuzzy Inference System (FIS) with a nature-inspired algorithm for the efficient prediction of k-barriers in the context of fast intrusion detection and prediction. By combining the interpretability of FIS with the optimisation capabilities of nature-inspired algorithms, there is a promising opportunity to enhance the accuracy and effectiveness of predicting the presence of barriers, thereby improving the overall performance of intrusion detection systems. This unexplored approach presents an exciting direction for future research in the field. In addition, none of the above-mentioned studies have considered the log-normal shadow fading model and circular RoI. Considering the research gap and limitations of the previous studies, the study in hand has proposed a novel evolutionary regression algorithm based on FIS and TLBO to predict the number of k-barriers for intrusion detection employing WSNs considering log-normal shadow fading model and circular RoI concurrently. The choice of TLBO as the meta-heuristic algorithm for this study was primarily driven by its well-established reputation and widespread usage in various real-world problems [24,25]. TLBO has consistently demonstrated competitive performance across diverse engineering, economics, and computer science domains. Its simplicity, efficiency, and ease of implementation have made it an attractive option for researchers and practitioners. It should be noted that the selection of an algorithm depends on the specific problem and research objectives at hand. While newer algorithms may exhibit improved performance in certain benchmark functions or specific scenarios, they might not be universally suitable or yield optimal results for every problem domain, particularly in regression tasks. On the other hand, TLBO has showcased effectiveness in a wide range of problem types, encompassing single-objective, multi-objective, and constrained optimisation problems [26]. The main contributions of this study are as follows;

- A robust framework is presented, consistently generating synthetic datasets to offer an economical and efficient solution with high reliability.
- Through the utilisation of the RReliefF algorithm, a rigorous analysis is conducted to accurately identify the most relevant features for intrusion detection and prevention.
- Comprehensive insights are gained by performing an extensive sensitivity analysis of all the identified features. This analysis is facilitated by employing advanced techniques such as a two-dimensional Partial Dependency plot (PDP) and regression tree ensemble learning.
- A pioneering evolutionary-based regression algorithm is proposed, specifically designed for accurate intrusion detection and prevention. The algorithm represents a significant advancement in this domain, pushing the boundaries of existing approaches.

3. System model

This section discusses the system model consisting of the sensor distribution model, the sensing range model, and some important terms and definitions that are crucial for the work presented in this study.

3.1. Sensor distribution model

It is assumed that a finite N number of sensors are distributed uniformly and randomly (*i.e.,* the probability that a given sensor would lie at an arbitrary point inside the region is the same for each sensor) inside a finite circular region of radius R meters and area $A = \pi R^2$. Each sensor is assumed to have identical hardware and software components, resulting in similar computational and processing capabilities. Mathematically, the probability that a sensor would lie at a random point denoted by (x, y) inside the circular region is given by Eq. (1)

$$f(x, y) = \frac{1}{A} \tag{1}$$

3.2. Log-normal shadow fading model

Since a WSN may be deployed in regions full of impediments that represent a realistic environment and may cause an abrupt change in the received signal power. This abrupt change in the received signal power is known as shadowing effects (SEs). Due to these SEs, the sensing range of sensors is not uniform in all directions. Therefore, earlier assumptions of uniform sensing range in all directions are not true and cannot be used in realistic environments. This work considers a more realistic and practical sensing range model, namely the log-normal shadowing path-loss model that incorporates the influence of SEs as well as the asymmetry in the sensing capability of sensors. The probability that the sensor would detect a target located at a distance r from the sensor is given by Eq. (2) [27].

$$P_{det}(r) = \phi\left(\frac{10\xi \log_{10}(r/\bar{r})}{\sigma}\right) \tag{2}$$

where, $\phi(\psi) = \frac{1}{\sqrt{2\pi}} \int_{\psi}^{\infty} \exp\left(\frac{-\chi_{\sigma}^2}{2}\right) d\chi_{\sigma}$; ξ, σ and $\bar{r}$ represent the signal power decay factor, standard deviation of SEs, and the expected sensing range of sensors, respectively.

3.3. Barrier and barrier path

We are familiar with the traditional method of protecting a specific region or entity from any kind of intrusion or demolition by fencing around it. Nowadays, the same approach has been extended using electric fences and WSNs. Currently, WSNs are being deployed to form barriers around circular regions and at borders to detect and prevent any kind of intrusion from the enemy side. Border surveillance should make sure that no enemy moves or illegal immigration take place across the state boundaries. In order to achieve the same, every path existing

Table 1
Simulation parameters used to synthetically extract the potential features.

Parameters	Values
Simulator	NS-2.35
Network Region	Circular RoI
Network Area (m^2)	5000, 9375, 15 000, 21 875, 30 000, 39 375, 50 000
Number of sensors (N)	100 to 400
Effective Sensing range (R_s)	6 to 35 m
Effective Transmission range (R_{tx})	12 to 70 m
Fading Parameter σ	2 to 12 dB
Sensor's deployment type	Uniform Distribution
Sensing model	Log-normal Shadowing model

from one end of the boundary to another must be monitored by at least one sensor, *i.e.*, the deployed sensors must form at least one barrier in such a way that every path across the boundaries must be covered by at least one sensor, this is known as barrier-coverage [15,28]. Similarly, if every possible intrusion path is covered by at least k distinct sensors of the deployed network, then the network is said to render k-barrier coverage. It is imperative to mention that the number of barriers for a given potential intrusion path depends on the shape and size of the region, the number of sensors deployed, and the sensing and transmission capabilities of sensors [29,30].

4. Machine learning model

4.1. Datasets generation

The performance and proficiency of any data-driven approach solely depend on the dataset's quality on which the corresponding model or algorithm is trained. These datasets can be either real data (*i.e.,* collected or measured on the field using sensors) or synthetic data (*i.e.,* derived from simple rules, models, or simulations). The process of collecting real data is complex and requires huge capital investments. In contrast, obtaining synthetic data is simple but requires high computational facilities. From the last few years, the use of synthetic data has increased drastically, and a study by Gartner predicts that its uses will increase exponentially and overshadow real data by 2030 [31]. We can find the use of synthetic data for machine learning in various applications such as healthcare [32,33], WSNs [34], and intrusion detection [19,20,35].

In this study, we have synthetically generated the whole dataset for training and testing purposes using network simulator NS-2.35, and the range of various parameters used in the simulation are given in Table 1. It is assumed that a given number of sensors (N) are spread randomly and uniformly inside a finite circular RoI for simulation results. All the sensors are considered to be homogeneous, *i.e.,* they have identical sensing, transmission, and computational capabilities. Note that the sensing and transmission range of sensors follows one of the most widely used log-normal shadow fading models represented by Eq. (2). The sensor's effective sensing and transmission range denotes the distances up to which a given sensor can sense and transmit the sensed information to the intended receiver, respectively. Further, it is assumed that two given sensors in the network can converse with each other if the effective transmission range is at least equal to twice the effective sensing range, *i.e.,* $R_{tx} \geq 2R_s$, where, R_s and R_{tx} denote the effective sensing and transmission range, respectively. Note that k-barrier coverage makes sure that a given random point inside the circular region is detected by at least k distinct sensors making the network robust against sensor failure. Further, k-barrier coverage for an intrusion path indicates that every possible path is covered by k distinct sensor while moving from the circumference to the centre of the circular RoI.

4.2. Feature importance

In this study, we have used RReliefF (ReliefF for regression problems) algorithm to evaluate the relative feature importance weights [36]. It ranks the features with k nearest neighbour. In this study, we considered a k of 10 based on the stability of the ranking. It penalises those features that assign different values to neighbours with the same target values and simultaneously rewards those that assign different values to neighbours with different target values. It computes the final importance weight by using the intermediary weights.

Let w_{dy}, w_{dj}, and $w_{dy\wedge dj}$ be the weights of having different values for the target (y), features (F_j), target and features, respectively. All these values are initialised to zero, including all the features (*i.e.,* $w_j = 0$). Any observation (say x_r) is randomly selected in iterative mode and searches the corresponding k-nearest observations to update the weights for each neighbour (say x_q).

$$w^i_{dy} = w^{i-1}_{dy} + \Delta_y(x_r, x_q) \cdot d_{rq} \tag{3}$$

$$w^i_{dj} = w^{i-1}_{dj} + \Delta_j(x_r, x_q) \cdot d_{rq} \tag{4}$$

$$w^i_{dy\wedge dj} = w^{i-1}_{dy\wedge dj} + \Delta_y(x_r, x_q) \cdot \Delta_j(x_r, x_q) \cdot d_{rq} \tag{5}$$

where $\Delta_y(x_r, x_q)$ is given by

$$\Delta_y(x_r, x_q) = \frac{|y_r - y_q|}{max(y) - min(y)} \tag{6}$$

where y_r and y_q are the target values for the observations x_r and x_q, respectively. The $\Delta_j(x_r, x_q)$ and d_{rq} are given by Eqs. (7) and (8) [37].

$$\Delta_j(x_r, x_q) = \frac{|x_{rj} - y_{qj}|}{max(Fj) - min(Fj)} \tag{7}$$

$$d_{rq} = \frac{\tilde{d_{rq}}}{\sum_{l=1}^{k} \tilde{d_{rl}}} \tag{8}$$

where $\tilde{d_{rq}}$ is given by

$$\tilde{d_{rq}} = e^{-\left(\frac{rank(r,q)}{sigma}\right)^2} \tag{9}$$

where $rank(r, q)$ is the position of the q^{th} observation's position among the r^{th} observation's, sorted by k distance. $sigma$ is the scaling factor. The final relative feature importance weight value is computed after the completion of the updating of all the intermediate weights.

$$w_j = \frac{w_{dy\wedge dj}}{w_{dy}} - \frac{w_{dj} - w_{dy\wedge dj}}{m - w_{dy}} \tag{10}$$

where m is the number of iterations.

4.3. Feature sensitivity

We can only estimate the relevancy of the features by estimating the feature importance. We conducted the sensitivity analysis of each feature by using a Partial Dependence Plot (PDP) through a regression tree ensemble for detailed impact analysis [38–43]. Through PDP, we can estimate the partial dependence of the target or response variable

either on a single feature or on a pair of features. In this study, we estimated the PDP between the target variable and pair of features by marginalising the impact of all the remaining features.

Let F^s be a subset given by $F^s = \{f_1, f_2\}$ of F_j given by $F_j = \{f_1, f_2, \cdots, f_j\}$. F^c represent the complementary set of F_j. The target response, $R(\boldsymbol{F})$, depends on all the elements present in F_j.

$$R(\boldsymbol{F}) = R(\boldsymbol{F^s}, \boldsymbol{F^c}) \tag{11}$$

The partial dependence of the target variable on $\boldsymbol{F^s}$ is estimated by taking the expectation of the response with respect to $\boldsymbol{F^c}$ [38].

$$R^s(\boldsymbol{F^s}) = E_c[R(\boldsymbol{F^s}, \boldsymbol{F^c})] \tag{12}$$

$$= \int R(\boldsymbol{F^s}, \boldsymbol{F^c}) \cdot pm_c(\boldsymbol{F^c}) \cdot d\boldsymbol{F^c} \tag{13}$$

where $pm_c(\boldsymbol{F^c})$ denotes the marginal probability of F^c.

$$pm_c(\boldsymbol{F^c}) \approx \int p(\boldsymbol{F^s}, \boldsymbol{F^c}) \cdot d\boldsymbol{F^s} \tag{14}$$

The final PDP for the feature set F^s is given by

$$R^s(\boldsymbol{F^s}) \approx \frac{1}{N_T} \sum_{i=1}^{N_t} R(\boldsymbol{F^s}, \boldsymbol{F_i^c}) \tag{15}$$

where N_T represent the total number of observations and $\boldsymbol{F_i} = (\boldsymbol{F_i^s}, \boldsymbol{F_i^c})$ represents the i^{th} observation.

4.4. Model setup

Algorithm 1: F-TLBO-ID regression algorithm

Input: Fuzzy Sets and Rules
Output: RMSE, R, and bias (Performance metrics)

```
Initialise the population size N and number of generations
while number of generations is not reached do
    Teacher Phase:
    Find the mean of each design variable, x_mean
    Identify the best solution as teacher
     [xteacher → x with f(x)max]
    for range(1,n) do
        Calculate T_F,i = round[1 + rand(0,1){2 − 1}]
        x_(new, i) = x_i + rand(0,1)[x_teacher - T_F,i · x_(mean)]
        Calculate f(x_new,i) for x_new,i
        if f(x_new,i) < f(x_i) then
            x_i = x_new,i
        End of Teacher Phase
        Student Phase:
        Select a learner randomly x_j such that j ≠ i
        if f(x_i) < f(x_j) then
            x_new,i = x_old,i + rand_i(x_i − x_j)
            else
                x_new,i = x_old,i + rand_i(x_j − x_i)
        if f(x_new,i) < f(x_i) then
            x_i = x_new,i
        End of Student Phase
return RMSE, R, and bias
```

TLBO is one of the widely used and robust bio-inspired optimisation algorithms [44]. It has been successfully applied in many diverse fields for optimisation tasks such as electrical engineering [45–48], thermal engineering [49], civil engineering [50] and many more [26,51–54]. Besides solving purely optimisation tasks, its potential is also evaluated for forecasting problems. Das and Padhy [55] have proposed a hybrid regression algorithm based on Support Vector Machine and TLBO (*i.e.,* SVM–TLBO) for commodity futures index forecasting.

In this investigation, we augmented TLBO with clustering techniques using Fuzzy C-mean clustering [56], a fuzzy counterpart of K-means or Lloyd's clustering algorithm [57]. The initial step involves clustering the input data, organising it in an optimised fashion for subsequent training. While an increased number of clusters enhances accuracy, it concurrently leads to heightened computational time-complexity. The objective is to fine-tune the fuzzy base parameters by leveraging TLBO to address modelling errors, yielding the optimal values as the conclusive outcome. Designating p_i^* as the ultimate optimised value for regression, two of its parameters, namely x_i and p_i^o, are determined through the collaborative efforts of TLBO and Fuzzy logic. The process commences by segregating the data (comprising inputs and targets) passing through the fuzzy system into training and testing sets, maintaining a ratio of 70:30, respectively. The second step involves defining linguistic variables and constructing membership functions, sets, and rules. Subsequently, the crisp feature matrix (inputs and target) is converted to a fuzzy model through fuzzification, resulting in an initial fuzzy model prepared for training using the TLBO algorithm. The fuzzy part of the system employs the "Sugeno" inference system, which has demonstrated superior performance compared to "Mamdani". Each input corresponds to a single feature, resulting in a total of five inputs (each possessing three membership functions). Furthermore, there are three rules employing the "and" operator, culminating in an output that contains the target values for this particular step.

Afterward, the output of the fuzzy model is passed as input to the TLBO algorithm to adjust the basic fuzzy parameters through its nature-inspired behaviour based on the x_i value. This adjustment affects the membership functions, modifying the Gaussian curve by adjusting the range and variance according to the fittest solution identified by the TLBO algorithm. Similar to other bio-inspired algorithms, the performance of the TLBO algorithm is influenced by the number of populations and iterations. In this case, the values of population and iteration are set to 11 and 2000, respectively. The number of decision variables is also set to 10, with lower and upper bounds of −10 and 10, respectively. The TLBO algorithm consists of two main phases: the teacher and learner phases. The teacher phase begins by finding the mean of each designed variable and identifying the best solution within the population, which becomes the teacher. In the learner phase, the selected teacher from the previous step shares its knowledge with the students to enhance the overall knowledge of the class. Consequently, the fuzzy input model is improved by incorporating the computations performed by the TLBO algorithm on its membership functions and parameters. By employing a fuzzy inference engine, the trained data (both train and test) is evaluated using the fuzzy TLBO model. In order to calculate the errors, the fuzzy data needs to be transformed back to its original crisp mode through a process known as defuzzification. This enables comparing the observed and predicted values, providing system error metrics such as RMSE and bias. For a detailed algorithm description, please refer to Algorithm 1. The entire methodology is depicted in Fig. 3.

5. Results

5.1. Feature importance and sensitivity

We evaluated the relative feature importance weight of each feature and plotted the feature importance graph (Fig. 4). We found that the number of sensor nodes emerges as the most relevant feature (with a feature weight of 0.0281) in predicting the number of k-barriers. The effective sensing and transmission range carries an equal importance weight of 0.0275. Interestingly, we found that the fading parameter is the least relevant feature (with a feature importance weight of 0.002). We have not found any outliers in any of the features.

In addition, we carried out the sensitivity analysis of all the features by using PDP analysis (Fig. 5). We observed that the area of the circular RoI and the effective sensing range has a negative impact on the target

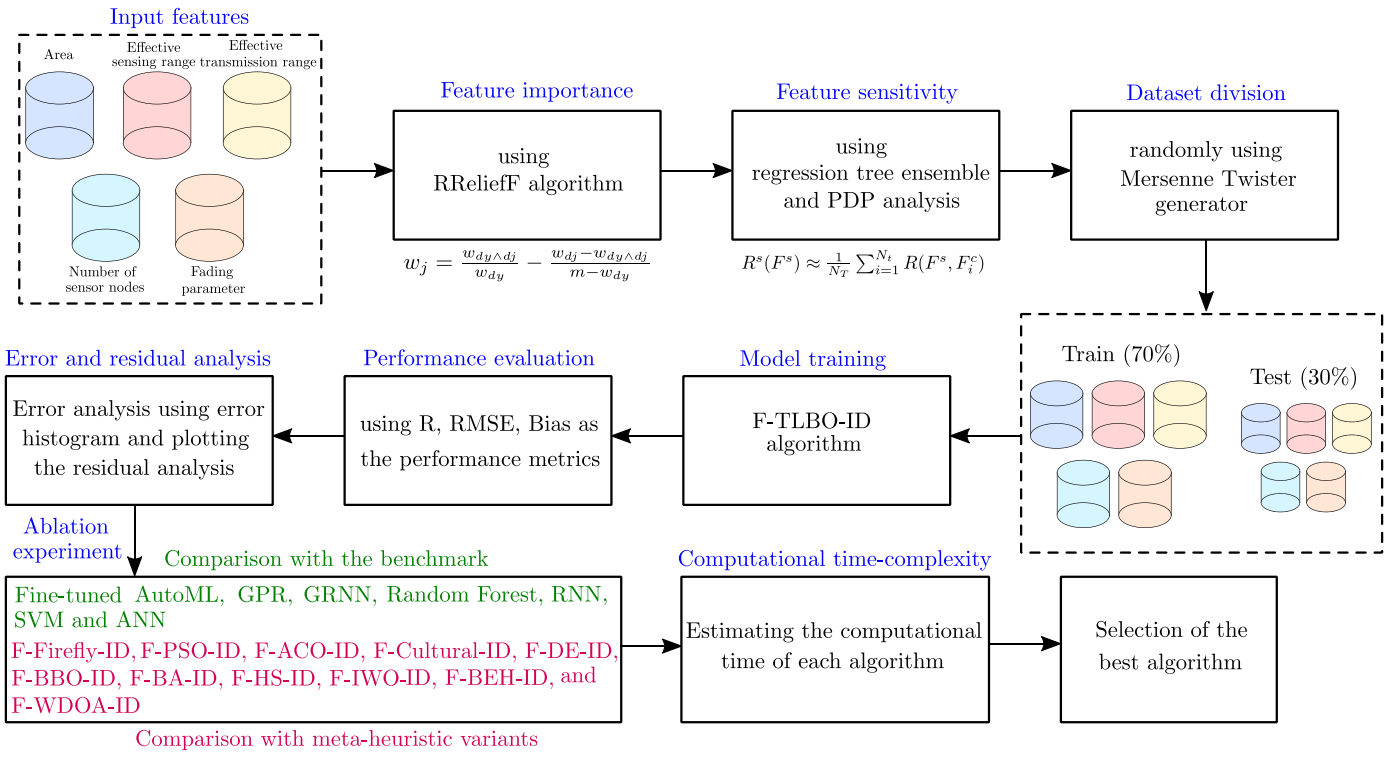

Fig. 3. Detailed flowchart of the proposed methodology.

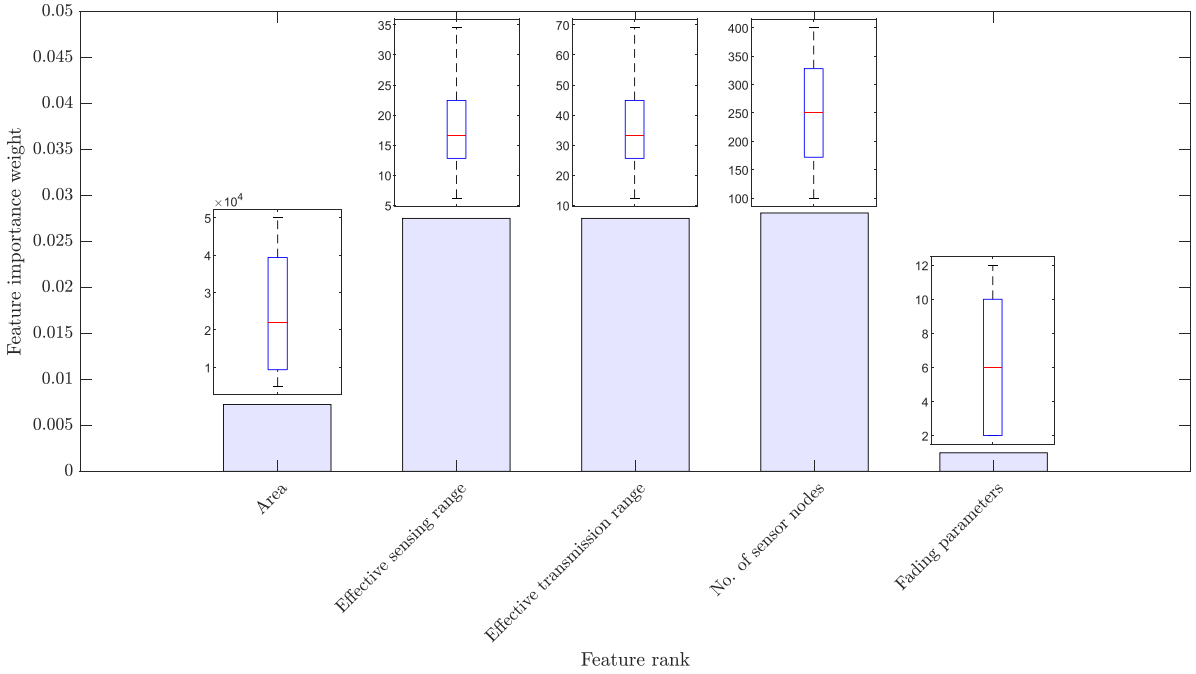

Fig. 4. Bar diagram illustrating the relative feature importance weight or score. The box plot over the bar shows the summary for the corresponding feature.

variable. Surprisingly, the effective transmission range does not have any effect of its own. The number of sensor nodes has a fluctuating positive impact on the target variable, whereas the fading parameter has no clear trend.

5.2. Model performance

We trained the fuzzy fed TLBO algorithm using the training datasets. To evaluate how well the model has been trained using the training datasets, we estimated the training accuracy by evaluating the model performance over the training datasets itself. We found the trained model performs well with R = 1, RMSE = 4.89, and bias = −0.02. However, we need to evaluate the model performance over unseen datasets for an unbiased evaluation. To do so, we evaluate the performance of the proposed algorithm over the testing datasets. We fed the testing datasets into the model input and plotted a linear fit between the predicted and the observed number of barriers (Fig. 6). We found

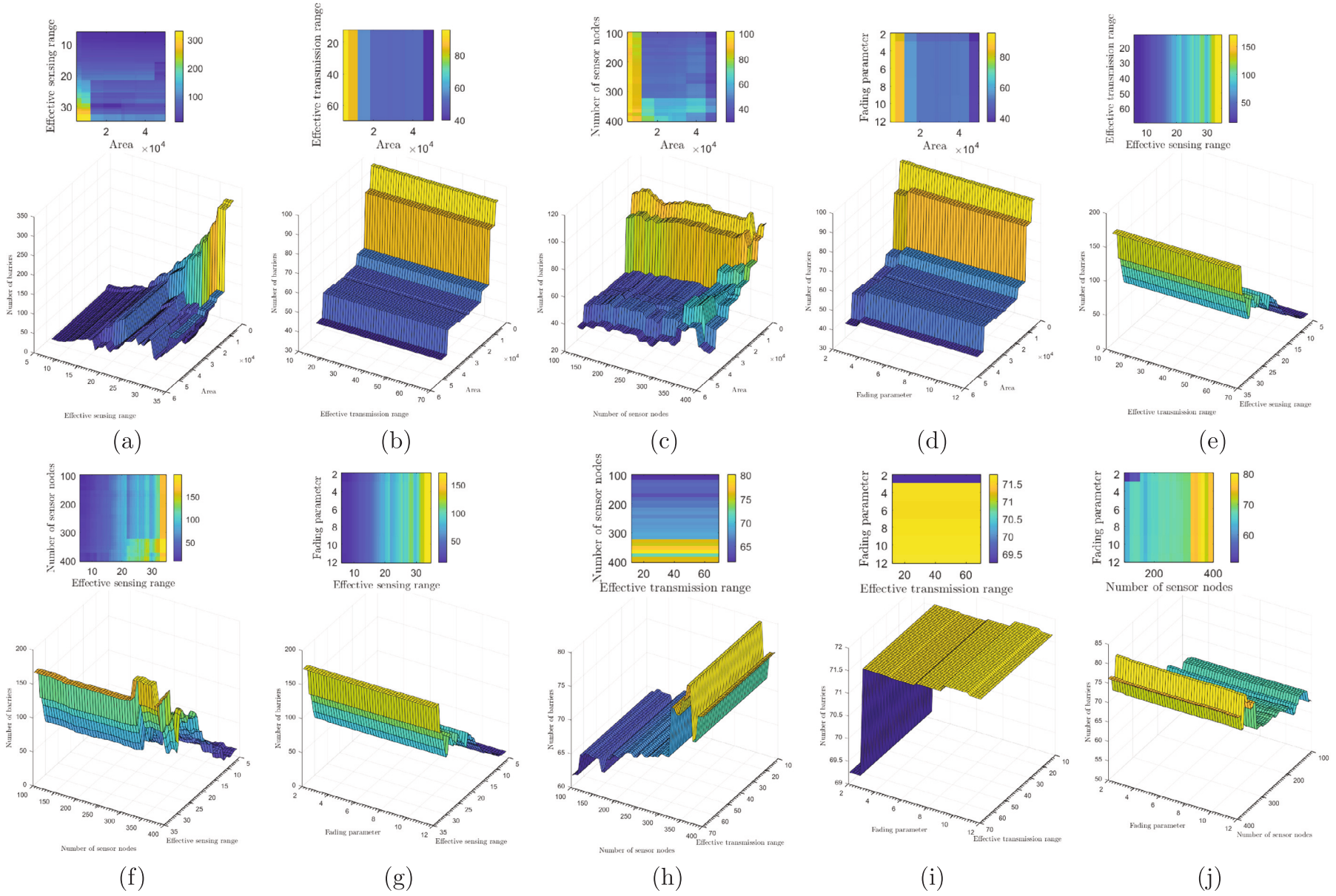

Fig. 5. Sensitivity analysis of the features by using the PDP. We have considered two features simultaneously and evaluated all possible (*i.e.,* ten) combinations illustrated in Figure a-f. We plotted the 2-D (top) and 3-D (bottom) variation profiles for each pair.

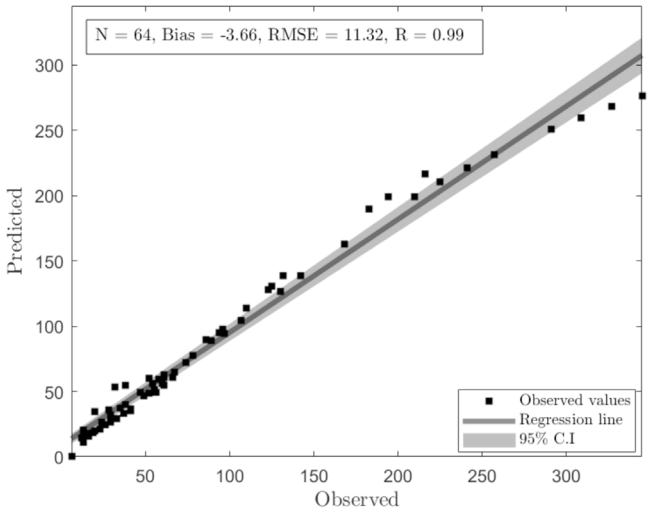

Fig. 6. Linear fit between the predicted and observed number of barriers. The solid line represents the best-fit line. The grey shade represents the 95% confidence interval.

that the model performs well over the testing datasets with R = 0.99, RMSE = 11.32, and bias = −3.66.

Further, to assess the appropriateness of the proposed algorithm, we performed the residual analysis. We plotted the time-series of the observed and predicted number of barriers with the 95% confidence interval (Fig. 7). We calculated the residual by subtracting the fitted values from the observed values and plotted the residual plot. The residuals are randomly oriented and do not form any specific pattern, indicating a good fit. Furthermore, we performed error analysis to know the distribution of the error by using an error histogram (Fig. 8). To do so, we plotted the error histogram with 10 bins. The error ranges from −64.02 (leftmost bin) to 17.05 (rightmost bin). The region on the left and right of the zero-error line shows the underestimated and overestimated regions, respectively. Interestingly, the peak of the error histogram coincides with the zero-error line that indicates a good fit.

6. Discussion

6.1. Ablation experiment

We conducted an input ablation study on the proposed algorithm by systematically removing or modifying parts of the input features to gain insights into the model's prediction process. To accomplish this, we selected various pairs of input features based on their relative feature importance scores. We examined a total of 11 input pairs to obtain a better understanding of the model's behaviour. We evaluated the performance of the model on all 11 input pairs under two scenarios: one utilising only the FIS (Fuzzy Inference System) and the other F-TLBO-ID (which combines the FIS with the TLBO algorithm). The results are presented in Table 2.

We observed that the F-TLBO-ID system consistently outperformed the FIS-only system for all input feature pairs. Furthermore, we discovered that the highest accuracy was achieved when all five input features were considered. This result can be attributed to the integration of the TLBO algorithm with the FIS, creating an evolutionary fuzzy system. This coupling enables the F-TLBO-ID system to adapt and evolve over time, thereby enhancing its overall performance. By utilising evolutionary algorithms, these systems optimise their fuzzy rules, membership functions, and control parameters to better align with the specific problem at hand. This adaptability empowers them to navigate changing or dynamic environments effectively. Additionally, evolutionary fuzzy systems exhibit enhanced robustness by leveraging fuzzy logic, which enables the representation of linguistic variables and rules to handle uncertainties and imprecise information. When combined with evolutionary algorithms, these systems further optimise their fuzzy rules to withstand noise, outliers, and variations in input data. As a result, they prove invaluable in real-world applications where data may be incomplete or noisy. Moreover, evolutionary fuzzy systems automate knowledge acquisition from data, eliminating the need for explicit programming or manual rule creation. This automated knowledge acquisition proves particularly beneficial in domains where human experts may possess an incomplete understanding of the underlying system. Furthermore, evolutionary fuzzy systems excel at modelling and controlling real-world problems that exhibit non-linear and complex behaviour. Traditional approaches often struggle in such scenarios. However, by combining the capacity of fuzzy logic to handle linguistic uncertainty with the aptitude of evolutionary algorithms for exploring complex solution spaces, these systems can more efficiently tackle non-linearities and complexities [58,59].

6.2. Comparison with fine-tuned benchmark algorithms and other meta-heuristic algorithms

For a robust analysis, we have compared the performance of the proposed algorithm with the benchmark machine learning algorithms (Table 3), namely Automated Machine Learning (AutoML), GPR, Generalised Regression Neural Network (GRNN), Random Forest, Recurrent Neural Network (RNN), Support Vector Machine (SVM), and Artificial Neural Network (ANN) [60–66]. We have selected these algorithms based on their generalisation capabilities in various applications domains of science and engineering, including intrusion detection [19,67–71]. Instead of using the standalone variants of these algorithms, we fine-tuned (*i.e.,* applied transfer learning to) all the algorithms, including AutoML, GPR, GRNN, RF, RNN, SVM, and ANN, to ensure a fair and comprehensive comparison. To achieve this, we utilised the initial learning parameters provided by Singh et al. [72] for GPR, GRNN, RF, RNN, SVM, and ANN, and Singh et al. [1] for AutoML. By incorporating the fine-tuning process and utilising the initial learning parameters provided by Singh et al. [1,72], we aimed to improve the performance and accuracy of these algorithms. Subsequently, we trained these fine-tuned algorithms over the same training datasets and evaluated their performance using the test datasets. We found that the proposed algorithm outperforms all the benchmark algorithms (with R = 0.99, RMSE = 11.32, and bias = −3.66). In terms of RMSE, GRNN emerged as the second-best algorithm with R = 0.91, RMSE = 12.03, and bias = 31.81. Notably, RNN performed the worst among the benchmark algorithms. It is worth mentioning that we observed a significant improvement in the accuracy of these fine-tuned algorithms compared to their native standalone variants. By employing the approach of fine-tuning the algorithms with carefully selected initial learning parameters from previous studies, we established a fair and rigorous comparison, considering the best possible performance of each algorithm.

To ensure a fair evaluation, we have compared the potential of TLBO for solving regression tasks with 11 different meta-heuristic algorithms. We carefully selected a balanced combination of established and newly proposed algorithms, taking into account their relevance to the regression task of predicting k-barriers for intrusion detection. In doing so, we selected Firefly algorithm [73], Particle swarm optimisation [74], Ant colony optimisation [75], Cultural algorithm [76], Differential evolution [77], Biogeography-based optimisation [78], Bees algorithm [79], Harmony search [80], Bee-Eater hunting strategy algorithm [81], and Weevil damage optimisation algorithm [82]. To evaluate the performance of these algorithms, we developed corresponding variants of F-TLBO-ID, such as F-Firefly-ID, F-PSO-ID, and so on, incorporating the specific optimisation techniques from each algorithm. These variants were trained on the same dataset, and we assessed their efficiency using widely recognised performance metrics for regression tasks, including R, RMSE, and bias. The results of our evaluation can be found in Table 4.

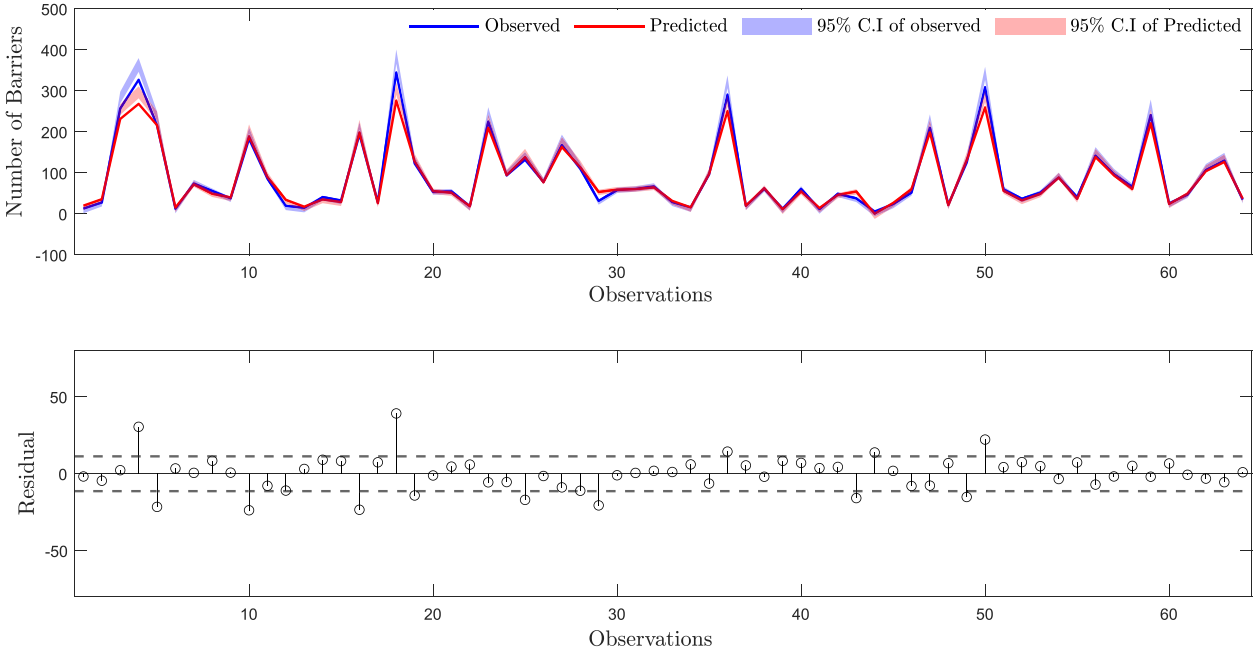

Fig. 7. Residual analysis of the proposed algorithm. The dotted line represents the RMSE value.

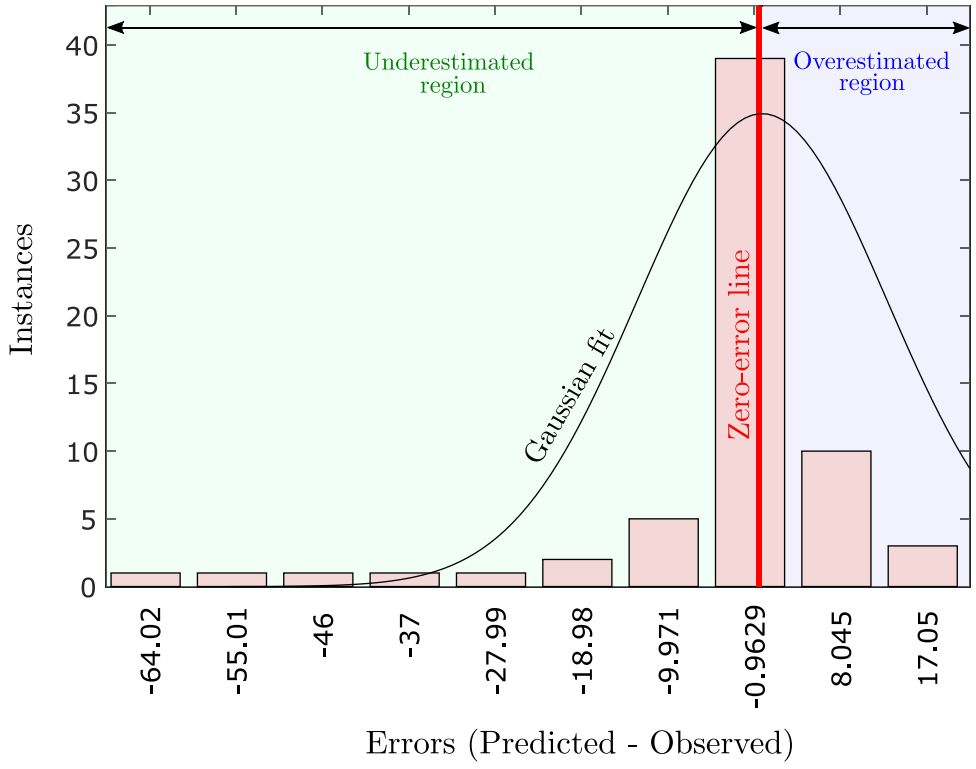

Fig. 8. Image illustrating the error histogram with 10 bins. The heights of each bin show the number of instances of the corresponding error. The vertical line (in red) represents the zero-error line. The black line shows the Gaussian fit.

After careful analysis, we found that F-TLBO-ID consistently outperformed the other algorithms across the selected performance metrics. Following F-TLBO-ID, F-PSO-ID emerged as the second-best algorithm in terms of performance metrics. This finding provides strong evidence for the potential and effectiveness of TLBO in solving the regression task of predicting k-barriers for intrusion detection. We want to emphasise that in our evaluation, we ensured a proper balance between established and newly proposed algorithms. By including both well-established algorithms with a strong theoretical foundation and newer algorithms that have shown promise in recent research, we aimed to provide a comprehensive comparison that takes into account the advancements in the field while considering the reliability and established performance of established algorithms.

Furthermore, to assess stability and convergence, we employed an ANOVA multiple comparison statistical test, which is widely recognised for its ability to compare means across multiple algorithms. Initially, we calculated the model errors for all 19 algorithms considered in our study. For the fine-tuned benchmark algorithms (AutoML, GPR,

Table 2
Ablation experiment on the input features. SN denotes the number of sensor nodes, ESR denotes the effective sensing range, ETR denotes the effective transmission range, and FP denotes fading parameter.

Input feature ablation	Only FIS			F-TLBO-ID (*i.e.,* FIS with TLBO)		
	R	RMSE	Bias	R	RMSE	Bias
SN	0.58	37.5038	−2.77	0.58	37.5038	−2.77
SN+ESR	0.84	26.47	0.84	0.85	27.69	3.51
SN+ETR	0.83	31.05	−2.5	0.84	30.87	−1.76
SN+Area	0.82	33.69	−0.46	0.96	14.21	−2.61
SN+FP	0.59	37.39	−0.86	0.61	36.11	−2.08
ESR+ETR	0.71	44.26	−6.99	0.79	36.99	−3.71
SN+ESR+ETR	0.84	23	23.27	0.89	19.18	21.89
SN+ESR+Area	0.91	22.8	2.27	0.97	13.45	1.78
SN+ETR+Area	0.92	21.48	−4.76	0.97	13.01	−1.54
SN+ESR+ETR+Area	0.83	74.63	16.54	0.89	27.38	1.28
SN+ESR+ETR+Area+FP	0.95	16.89	10.87	0.99	11.32	−3.66

Table 3
Comparison of the proposed algorithm with the fine-tuned machine learning algorithms.

Performance metrics	Algorithms							
	F-TLBO-ID	AutoML	GPR	GRNN	Random Forest	RNN	SVM	ANN
R	0.99	0.80	0.81	0.91	0.83	0.60	0.75	0.95
RMSE	11.32	2.79	82.37	12.03	41.27	148.93	14.02	19.34
Bias	−3.66	42.10	47.22	31.81	39.32	132.61	24.98	−1.39

Note: The values marked in blue and red represents the best and worst observations, respectively.

Table 4
Comparison of the proposed algorithm with its corresponding variants of nature-inspired algorithms. PSO denotes particle swarm optimisation, ACO denotes ant colony optimisation, DE denotes differential evolution, BBO denotes biogeography-based optimisation, BA denotes bees algorithm, HS denotes harmony search, BEH denotes bee-eater hunting strategy algorithm, and WDOA denotes Weevil damage optimisation algorithm.

Performance metrics	F-Firefly-ID	F-PSO-ID	F-ACO-ID	F-Cultural-ID	F-DE-ID	F-BBO-ID	F-BA-ID	F-HS-ID	F-IWO-ID	F-BEH-ID	F-WDOA-ID	F-TLBO-ID (This study)
R	0.78	0.94	0.88	0.63	0.85	0.92	0.86	0.79	0.93	0.77	0.47	0.99
RMSE	46.65	26.62	26.70	53.49	44.47	20.17	30.03	47.71	21.44	25.31	103.71	11.32
Bias	15.82	0.55	2.71	17.53	−22.27	−0.42	9.72	0.96	−1.96	5.09	−21.21	−3.66

Note: The values marked in blue and red represents the best and worst observations, respectively.

GRNN, RF, RNN, SVM, and ANN), we determined the model error by averaging the results of 30 independent runs for each algorithm. Conversely, for the meta-heuristic algorithms (F-Firefly-ID, F-PSO-ID, F-ACO-ID, F-Cultural-ID, F-DE-ID, F-BBO-ID, F-BA-ID, F-HS-ID, F-IWO-ID, F-BEH-ID, F-WDOA-ID, and F-TLBO-ID), we conducted multiple runs of the algorithms, and each run was executed for a sufficient number of iterations to ensure algorithm convergence. To achieve convergence, we executed each of the 12 algorithms independently 30 times, allowing a maximum of 2000 iterations per run. This resulted in a total of 720,000 iterations. We rigorously monitored and verified that each algorithm consistently converged to near-optimal solutions.

To guarantee the validity of our analysis, we computed the average model error for each algorithm to a Kolmogorov–Smirnov normality test. The outcomes unequivocally affirmed that the model error distributions for all 19 algorithms adhered to a normal distribution. Based on this confirmation, we proceeded with a one-way ANOVA multiple comparison test, employing Tukey's Honestly Significant Difference (HSD) method as a robust post hoc analysis technique. By utilising Tukey's HSD method, we were able to compare the means of the model errors across different algorithms and derive adjusted p-values. These adjusted p-values served as a measure of significance for pairwise comparisons, enabling us to assess the differences between algorithm performance with greater confidence and reliability.

To visually depict the error distribution observed in the ANOVA analysis, we generated a highly informative box plot of the ANOVA model errors. Fig. 9 showcases this visual representation, offering a clear and intuitive understanding of the range and variability of the model errors associated with each algorithm. This graphical representation enhances the overall comprehension of the analysis results. Furthermore, we presented the pairwise comparisons in Table 5, which offers detailed insights into the statistical significance of our findings. In this table, we included the adjusted p-values, which serve as reliable indicators of the significance of the observed differences between algorithm performances.

To enhance the clarity of our research findings, we have taken an additional step by providing an illustrative graphical representation of the pairwise comparisons. The visual representation, shown in Fig. 10, offers a compelling visualisation that clearly demonstrates the superior performance of our proposed F-TLBO-ID algorithm (marked in blue) compared to the other algorithms considered. Within this graphical depiction, it is important to note that models exhibiting overlap with the blue patch indicate a good model fit, suggesting similar performance levels. Conversely, models depicted with green lines outperform those represented by red lines, emphasising the notable advantages and higher efficacy of our proposed algorithm.

6.3. Comparison of the computational time-complexity

Comparison of the proposed algorithm with a benchmark algorithm, whether it is an explainable or black-box model or a nature-inspired approach, should not solely rely on performance metrics. Disregarding the computational time-complexity can lead to biased interpretations. To ensure a fair and comprehensive evaluation, we considered each algorithm's computational time and employed a bubble diagram to better visualise and interpret the results (see Fig. 11).

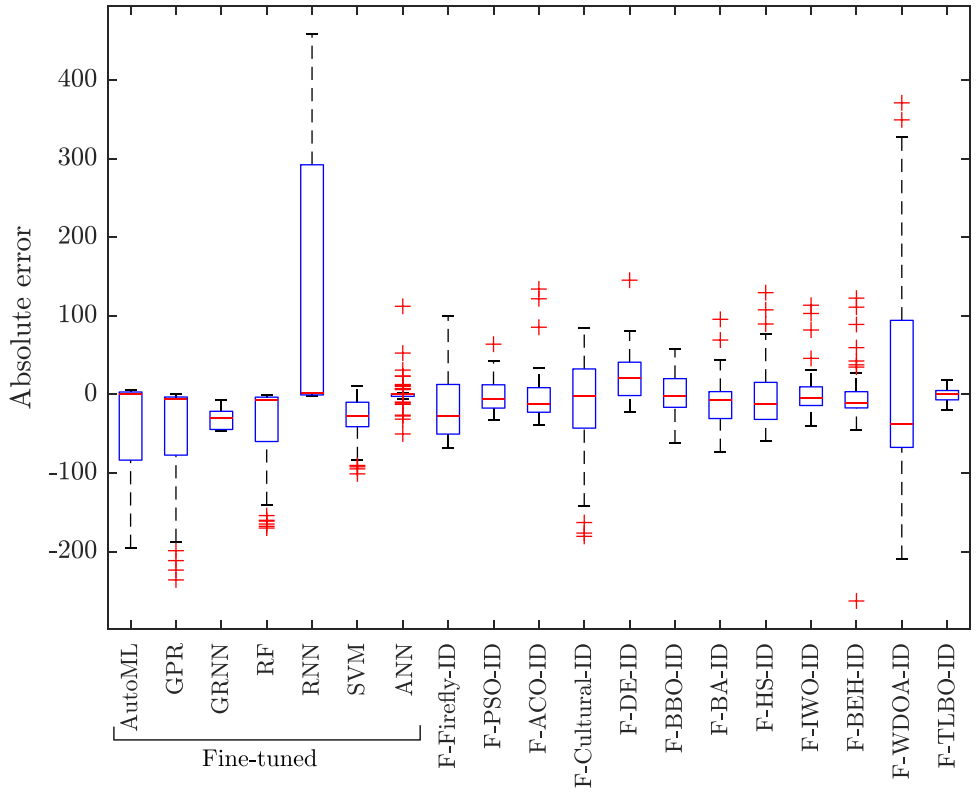

Fig. 9. Boxplot representation of the model error analysed using ANOVA. The tops and bottoms of each "box" represent the samples' upper and lower quartiles (the 75th and 25th percentiles). The line in the middle of each box represents the sample median.

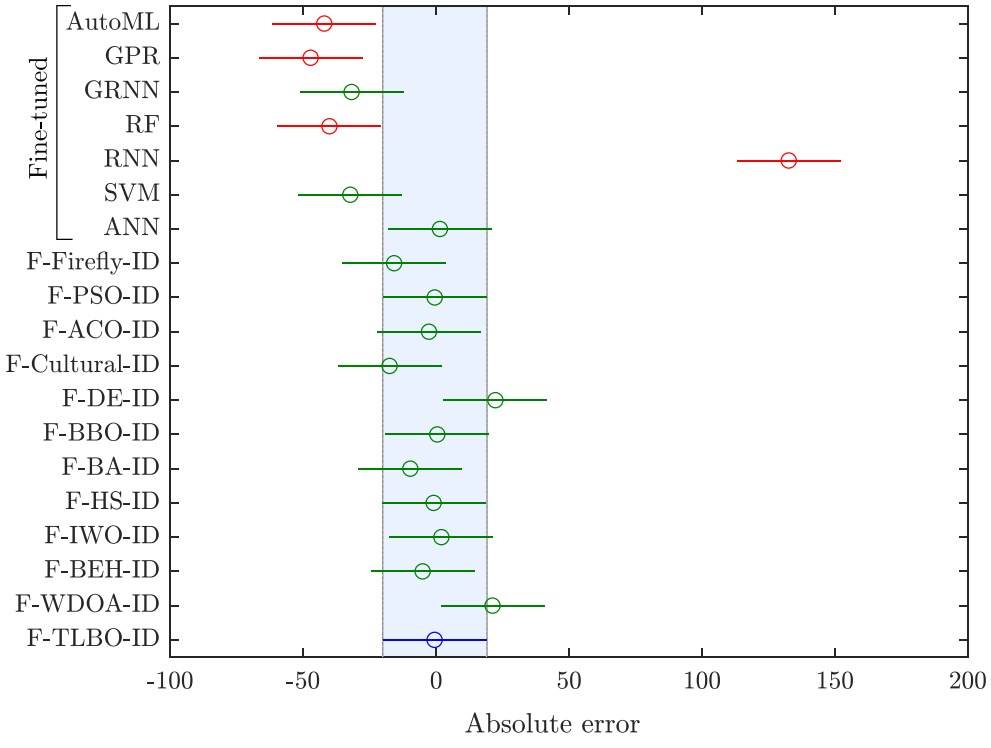

Fig. 10. ANOVA multiple comparison test using Tukey's Honestly Significant Difference (HSD) approach.

Table 5
Model-wise comparison results of the post hoc tests.

Group A	Group B	Lower limit	A–B	Upper limit	p-value
Fine-tuned AutoML	Fine-tuned GPR	−33.899399	5.129741	44.158882	0.999999
Fine-tuned AutoML	Fine-tuned GRNN	−49.311904	−10.282763	28.746378	0.999990
Fine-tuned AutoML	Fine-tuned RF	−40.973266	−1.944125	37.085016	0.999999
Fine-tuned AutoML	Fine-tuned RNN	−213.742504	−174.713363	−135.684222	0.000000
Fine-tuned AutoML	Fine-tuned SVM	−48.824075	−9.794934	29.234207	0.999995
Fine-tuned AutoML	Fine-tuned ANN	−82.534016	−43.504875	−4.475734	0.011984
Fine-tuned AutoML	F-Firefly-ID	−65.306034	−26.276893	12.752248	0.660023
Fine-tuned AutoML	F-PSO-ID	−80.576557	−41.547416	−2.518275	0.023032
Fine-tuned AutoML	F-ACO-ID	−78.418153	−39.389012	−0.359871	0.044967
Fine-tuned AutoML	F-Cultural-ID	−63.596350	−24.567209	14.461932	0.769595
Fine-tuned AutoML	F-DE-ID	−103.396907	−64.367766	−25.338625	0.000001
Fine-tuned AutoML	F-BBO-ID	−81.540048	−42.510907	−3.481766	0.016789
Fine-tuned AutoML	F-BA-ID	−71.398861	−32.369720	6.659421	0.260992
Fine-tuned AutoML	F-HS-ID	−80.161767	−41.132626	−2.103485	0.026304
Fine-tuned AutoML	F-IWO-ID	−83.084124	−44.054983	−5.025842	0.009898
Fine-tuned AutoML	F-BEH-ID	−76.033217	−37.004076	2.025065	0.088096
Fine-tuned AutoML	F-WDOA-ID	−102.336683	−63.307542	−24.278401	0.000002
Fine-tuned AutoML	F-TLBO-ID	−80.537343	−41.508202	−2.479061	0.023325
Fine-tuned GPR	Fine-tuned GRNN	−54.441646	−15.412505	23.616636	0.997325
Fine-tuned GPR	Fine-tuned RF	−46.103008	−7.073867	31.955274	0.999999
Fine-tuned GPR	Fine-tuned RNN	−218.872245	−179.843104	−140.813963	0.000000
Fine-tuned GPR	Fine-tuned SVM	−53.953816	−14.924675	24.104466	0.998211
Fine-tuned GPR	Fine-tuned ANN	−87.663757	−48.634616	−9.605475	0.001781
Fine-tuned GPR	F-Firefly-ID	−70.435776	−31.406635	7.622506	0.314351
Fine-tuned GPR	F-PSO-ID	−85.706298	−46.677157	−7.648016	0.003806
Fine-tuned GPR	F-ACO-ID	−83.547895	−44.518754	−5.489613	0.008403
Fine-tuned GPR	F-Cultural-ID	−68.726092	−29.696951	9.332190	0.421950
Fine-tuned GPR	F-DE-ID	−108.526648	−69.497507	−30.468366	0.000000
Fine-tuned GPR	F-BBO-ID	−86.669789	−47.640648	−8.611507	0.002632
Fine-tuned GPR	F-BA-ID	−76.528602	−37.499461	1.529680	0.077073
Fine-tuned GPR	F-HS-ID	−85.291508	−46.262367	−7.233226	0.004448
Fine-tuned GPR	F-IWO-ID	−88.213865	−49.184724	−10.155583	0.001429
Fine-tuned GPR	F-BEH-ID	−81.162959	−42.133818	−3.104677	0.019024
Fine-tuned GPR	F-WDOA-ID	−107.466425	−68.437284	−29.408143	0.000000
Fine-tuned GPR	F-TLBO-ID	−85.667085	−46.637944	−7.608803	0.003863
Fine-tuned GRNN	Fine-tuned RF	−30.690503	8.338638	47.367779	0.999999
Fine-tuned GRNN	Fine-tuned RNN	−203.459740	−164.430599	−125.401458	0.000000
Fine-tuned GRNN	Fine-tuned SVM	−38.541311	0.487830	39.516971	0.999999
Fine-tuned GRNN	Fine-tuned ANN	−72.251252	−33.222111	5.807030	0.218775
Fine-tuned GRNN	F-Firefly-ID	−55.023271	−15.994130	23.035011	0.995800
Fine-tuned GRNN	F-PSO-ID	−70.293793	−31.264652	7.764489	0.322698
Fine-tuned GRNN	F-ACO-ID	−68.135390	−29.106249	9.922892	0.462070
Fine-tuned GRNN	F-Cultural-ID	−53.313587	−14.284446	24.744695	0.998982
Fine-tuned GRNN	F-DE-ID	−93.114143	−54.085002	−15.055861	0.000177
Fine-tuned GRNN	F-BBO-ID	−71.257285	−32.228144	6.800997	0.268468
Fine-tuned GRNN	F-BA-ID	−61.116097	−22.086956	16.942185	0.891890
Fine-tuned GRNN	F-HS-ID	−69.879003	−30.849862	8.179279	0.347743
Fine-tuned GRNN	F-IWO-ID	−72.801360	−33.772219	5.256922	0.194107
Fine-tuned GRNN	F-BEH-ID	−65.750454	−26.721313	12.307828	0.629445
Fine-tuned GRNN	F-WDOA-ID	−92.053920	−53.024779	−13.995638	0.000283
Fine-tuned GRNN	F-TLBO-ID	−70.254580	−31.225439	7.803702	0.325024
Fine-tuned RF	Fine-tuned RNN	−211.798378	−172.769237	−133.740096	0.000000
Fine-tuned RF	Fine-tuned SVM	−46.879949	−7.850808	31.178333	0.999999
Fine-tuned RF	Fine-tuned ANN	−80.589890	−41.560749	−2.531608	0.022933
Fine-tuned RF	F-Firefly-ID	−63.361909	−24.332768	14.696373	0.783257
Fine-tuned RF	F-PSO-ID	−78.632431	−39.603290	−0.574149	0.042183
Fine-tuned RF	F-ACO-ID	−76.474028	−37.444887	1.584254	0.078228
Fine-tuned RF	F-Cultural-ID	−61.652225	−22.623084	16.406057	0.869870
Fine-tuned RF	F-DE-ID	−101.452781	−62.423640	−23.394499	0.000003
Fine-tuned RF	F-BBO-ID	−79.595922	−40.566781	−1.537640	0.031429
Fine-tuned RF	F-BA-ID	−69.454735	−30.425594	8.603547	0.374319
Fine-tuned RF	F-HS-ID	−78.217641	−39.188500	−0.159359	0.047714
Fine-tuned RF	F-IWO-ID	−81.139998	−42.110857	−3.081716	0.019168
Fine-tuned RF	F-BEH-ID	−74.089092	−35.059951	3.969190	0.144193
Fine-tuned RF	F-WDOA-ID	−100.392558	−61.363417	−22.334276	0.000005
Fine-tuned RF	F-TLBO-ID	−78.593218	−39.564077	−0.534936	0.042681
Fine-tuned RNN	Fine-tuned SVM	125.889288	164.918429	203.947570	0.000000
Fine-tuned RNN	Fine-tuned ANN	92.179347	131.208488	170.237629	0.000000
Fine-tuned RNN	F-Firefly-ID	109.407329	148.436470	187.465611	0.000000
Fine-tuned RNN	F-PSO-ID	94.136806	133.165947	172.195088	0.000000
Fine-tuned RNN	F-ACO-ID	96.295210	135.324351	174.353492	0.000000
Fine-tuned RNN	F-Cultural-ID	111.117012	150.146153	189.175294	0.000000
Fine-tuned RNN	F-DE-ID	71.316456	110.345597	149.374738	0.000000

(*continued on next page*)

Table 5 (*continued*).

Group A	Group B	Lower limit	A–B	Upper limit	p-value
Fine-tuned RNN	F-BBO-ID	93.173315	132.202456	171.231597	0.000000
Fine-tuned RNN	F-BA-ID	103.314502	142.343643	181.372784	0.000000
Fine-tuned RNN	F-HS-ID	94.551596	133.580737	172.609878	0.000000
Fine-tuned RNN	F-IWO-ID	91.629239	130.658380	169.687521	0.000000
Fine-tuned RNN	F-BEH-ID	98.680145	137.709286	176.738427	0.000000
Fine-tuned RNN	F-WDOA-ID	72.376679	111.405820	150.434961	0.000000
Fine-tuned RNN	F-TLBO-ID	94.176020	133.205161	172.234302	0.000000
Fine-tuned SVM	Fine-tuned ANN	−72.739082	−33.709941	5.319200	0.196798
Fine-tuned SVM	F-Firefly-ID	−55.511100	−16.481959	22.547182	0.994009
Fine-tuned SVM	F-PSO-ID	−70.781623	−31.752482	7.276659	0.294523
Fine-tuned SVM	F-ACO-ID	−68.623219	−29.594078	9.435063	0.428851
Fine-tuned SVM	F-Cultural-ID	−53.801417	−14.772276	24.256865	0.998430
Fine-tuned SVM	F-DE-ID	−93.601973	−54.572832	−15.543691	0.000142
Fine-tuned SVM	F-BBO-ID	−71.745114	−32.715973	6.313168	0.243261
Fine-tuned SVM	F-BA-ID	−61.603927	−22.574786	16.454355	0.871958
Fine-tuned SVM	F-HS-ID	−70.366833	−31.337692	7.691449	0.318389
Fine-tuned SVM	F-IWO-ID	−73.289190	−34.260049	4.769092	0.173922
Fine-tuned SVM	F-BEH-ID	−66.238284	−27.209143	11.819998	0.595331
Fine-tuned SVM	F-WDOA-ID	−92.541750	−53.512609	−14.483468	0.000228
Fine-tuned SVM	F-TLBO-ID	−70.742409	−31.713268	7.315873	0.296735
Fine-tuned ANN	F-Firefly-ID	−21.801159	17.227982	56.257123	0.990066
Fine-tuned ANN	F-PSO-ID	−37.071682	1.957459	40.986600	0.999999
Fine-tuned ANN	F-ACO-ID	−34.913278	4.115863	43.145004	0.999999
Fine-tuned ANN	F-Cultural-ID	−20.091476	18.937665	57.966806	0.972888
Fine-tuned ANN	F-DE-ID	−59.892032	−20.862891	18.166250	0.932722
Fine-tuned ANN	F-BBO-ID	−38.035173	0.993968	40.023109	0.999999
Fine-tuned ANN	F-BA-ID	−27.893986	11.135155	50.164296	0.999968
Fine-tuned ANN	F-HS-ID	−36.656892	2.372249	41.401390	0.999999
Fine-tuned ANN	F-IWO-ID	−39.579249	−0.550108	38.479033	0.999999
Fine-tuned ANN	F-BEH-ID	−32.528343	6.500798	45.529939	0.999999
Fine-tuned ANN	F-WDOA-ID	−58.831809	−19.802668	19.226473	0.958116
Fine-tuned ANN	F-TLBO-ID	−37.032468	1.996673	41.025814	0.999999
F-Firefly-ID	F-PSO-ID	−54.299664	−15.270523	23.758618	0.997615
F-Firefly-ID	F-ACO-ID	−52.141260	−13.112119	25.917022	0.999677
F-Firefly-ID	F-Cultural-ID	−37.319457	1.709684	40.738825	0.999999
F-Firefly-ID	F-DE-ID	−77.120014	−38.090873	0.938268	0.065430
F-Firefly-ID	F-BBO-ID	−55.263155	−16.234014	22.795127	0.994986
F-Firefly-ID	F-BA-ID	−45.121968	−6.092827	32.936314	0.999999
F-Firefly-ID	F-HS-ID	−53.884874	−14.855733	24.173408	0.998313
F-Firefly-ID	F-IWO-ID	−56.807231	−17.778090	21.251051	0.985967
F-Firefly-ID	F-BEH-ID	−49.756324	−10.727183	28.301958	0.999982
F-Firefly-ID	F-WDOA-ID	−76.059790	−37.030649	1.998492	0.087474
F-Firefly-ID	F-TLBO-ID	−54.260450	−15.231309	23.797832	0.997690
F-PSO-ID	F-ACO-ID	−36.870737	2.158404	41.187545	0.999999
F-PSO-ID	F-Cultural-ID	−22.048935	16.980206	56.009347	0.991561
F-PSO-ID	F-DE-ID	−61.849491	−22.820350	16.208791	0.861131
F-PSO-ID	F-BBO-ID	−39.992632	−0.963491	38.065650	0.999999
F-PSO-ID	F-BA-ID	−29.851445	9.177696	48.206837	0.999998
F-PSO-ID	F-HS-ID	−38.614351	0.414790	39.443931	0.999999
F-PSO-ID	F-IWO-ID	−41.536708	−2.507567	36.521574	0.999999
F-PSO-ID	F-BEH-ID	−34.485802	4.543339	43.572480	0.999999
F-PSO-ID	F-WDOA-ID	−60.789268	−21.760127	17.269014	0.904066
F-PSO-ID	F-TLBO-ID	−38.989927	0.039214	39.068355	0.999999
F-ACO-ID	F-Cultural-ID	−24.207338	14.821803	53.850944	0.998361
F-ACO-ID	F-DE-ID	−64.007895	−24.978754	14.050387	0.744711
F-ACO-ID	F-BBO-ID	−42.151036	−3.121895	35.907246	0.999999
F-ACO-ID	F-BA-ID	−32.009849	7.019292	46.048433	0.999999
F-ACO-ID	F-HS-ID	−40.772755	−1.743614	37.285527	0.999999
F-ACO-ID	F-IWO-ID	−43.695112	−4.665971	34.363170	0.999999
F-ACO-ID	F-BEH-ID	−36.644205	2.384936	41.414077	0.999999
F-ACO-ID	F-WDOA-ID	−62.947671	−23.918530	15.110611	0.806412
F-ACO-ID	F-TLBO-ID	−41.148331	−2.119190	36.909951	0.999999
F-Cultural-ID	F-DE-ID	−78.829697	−39.800556	−0.771415	0.039752
F-Cultural-ID	F-BBO-ID	−56.972839	−17.943698	21.085443	0.984495
F-Cultural-ID	F-BA-ID	−46.831651	−7.802510	31.226631	0.999999
F-Cultural-ID	F-HS-ID	−55.594557	−16.565416	22.463725	0.993646
F-Cultural-ID	F-IWO-ID	−58.516914	−19.487773	19.541368	0.964061
F-Cultural-ID	F-BEH-ID	−51.466008	−12.436867	26.592274	0.999845
F-Cultural-ID	F-WDOA-ID	−77.769474	−38.740333	0.288808	0.054379
F-Cultural-ID	F-TLBO-ID	−55.970134	−16.940993	22.088148	0.991780
F-DE-ID	F-BBO-ID	−17.172282	21.856859	60.886000	0.900561
F-DE-ID	F-BA-ID	−7.031095	31.998046	71.027187	0.280893
F-DE-ID	F-HS-ID	−15.794001	23.235140	62.264281	0.841652
F-DE-ID	F-IWO-ID	−18.716358	20.312783	59.341924	0.946979

(*continued on next page*)

Table 5 (*continued*).

Group A	Group B	Lower limit	A–B	Upper limit	p-value
F-DE-ID	F-BEH-ID	−11.665452	27.363689	66.392830	0.584445
F-DE-ID	F-WDOA-ID	−37.968917	1.060223	40.089364	0.999999
F-DE-ID	F-TLBO-ID	−16.169577	22.859564	61.888705	0.859353
F-BBO-ID	F-BA-ID	−28.887954	10.141187	49.170328	0.999992
F-BBO-ID	F-HS-ID	−37.650860	1.378281	40.407422	0.999999
F-BBO-ID	F-IWO-ID	−40.573217	−1.544076	37.485065	0.999999
F-BBO-ID	F-BEH-ID	−33.522310	5.506831	44.535972	0.999999
F-BBO-ID	F-WDOA-ID	−59.825776	−20.796635	18.232506	0.934568
F-BBO-ID	F-TLBO-ID	−38.026436	1.002705	40.031846	0.999999
F-BA-ID	F-HS-ID	−47.792047	−8.762906	30.266235	0.999998
F-BA-ID	F-IWO-ID	−50.714404	−11.685263	27.343878	0.999936
F-BA-ID	F-BEH-ID	−43.663498	−4.634357	34.394784	0.999999
F-BA-ID	F-WDOA-ID	−69.966964	−30.937823	8.091318	0.342352
F-BA-ID	F-TLBO-ID	−48.167623	−9.138483	29.890658	0.999998
F-HS-ID	F-IWO-ID	−41.951498	−2.922357	36.106784	0.999999
F-HS-ID	F-BEH-ID	−34.900592	4.128549	43.157690	0.999999
F-HS-ID	F-WDOA-ID	−61.204058	−22.174917	16.854224	0.888452
F-HS-ID	F-TLBO-ID	−39.404717	−0.375576	38.653565	0.999999
F-IWO-ID	F-BEH-ID	−31.978235	7.050906	46.080047	0.999999
F-IWO-ID	F-WDOA-ID	−58.281701	−19.252560	19.776581	0.968068
F-IWO-ID	F-TLBO-ID	−36.482360	2.546781	41.575922	0.999999
F-BEH-ID	F-WDOA-ID	−65.332607	−26.303466	12.725675	0.658212
F-BEH-ID	F-TLBO-ID	−43.533267	−4.504126	34.525015	0.999999
F-WDOA-ID	F-TLBO-ID	−17.229801	21.799340	60.828481	0.902655

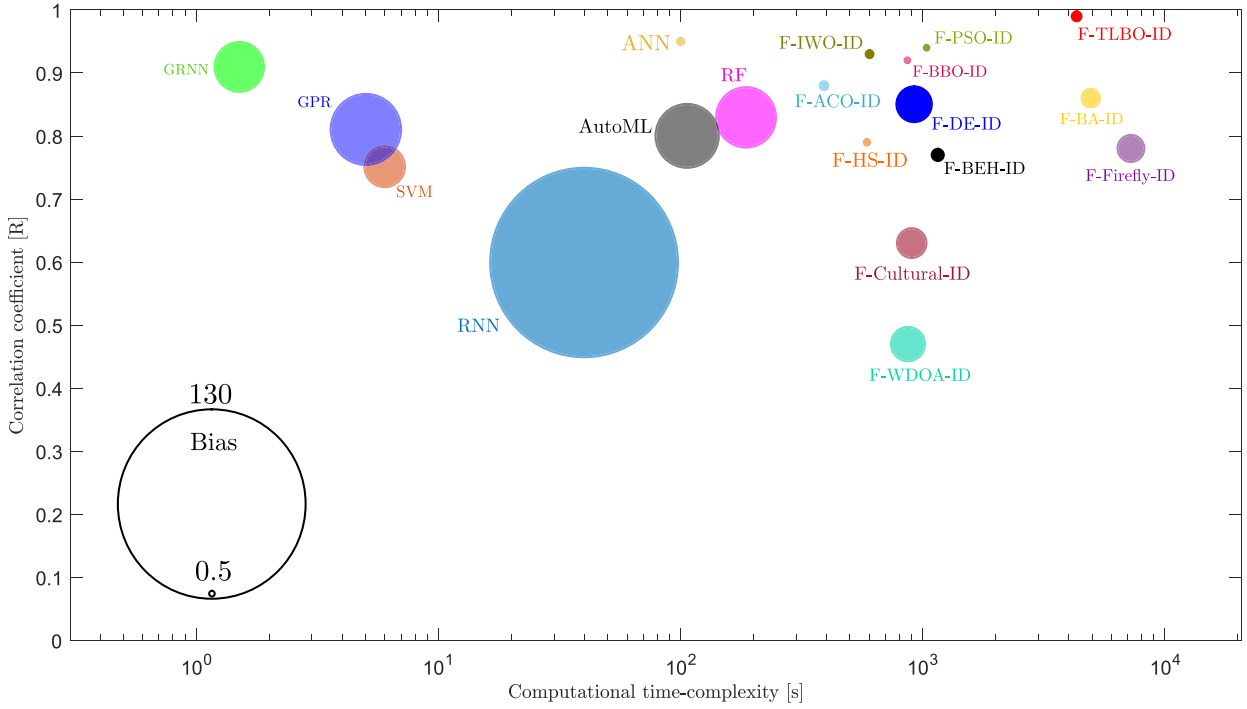

Fig. 11. Computational time-complexity analysis of the benchmark algorithms. The radius of the circle indicates the magnitude of the bias value.

Our analysis revealed interesting findings regarding the time-complexity of the algorithms under consideration. F-Firefly-ID exhibited the highest time-complexity among all the algorithms, followed by F-BA-ID. Surprisingly, despite its exceptional performance, the proposed F-TLBO-ID algorithm ranked third in terms of time-complexity. Notably, a distinct segregation was observed between fine-tuned machine learning algorithms and nature-inspired algorithms in terms of their time-complexity. In general, nature-inspired algorithms exhibited higher time-complexity compared to fine-tuned machine learning algorithms. This observation highlights the trade-off between the computational cost and the performance achieved by these different algorithmic paradigms. It is worth mentioning that, among all the algorithms compared, the GRNN algorithm demonstrated the least time-complexity while maintaining reasonable performance. By considering both performance and computational time-complexity, we can gain a more comprehensive understanding of the strengths and limitations of each algorithm, leading to a more informed and unbiased interpretation of the results.

6.4. Performance over benchmark datasets

We also evaluated the performance of the proposed algorithm over publicly available datasets to predict the number of barriers. We considered the datasets of Singh et al. [20] downloaded from UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/715/lt+fs+id+intrusion+detection+in+wsns). This dataset consists of four features (*i.e.,* area, sensing range, transmission range, and sensors) to

predict the number of barriers. They proposed the LT-FS-ID algorithm over these datasets and reported the performance metrics of R = 0.98, RMSE = 6.47, and bias = 12.35. We applied the F-TLBO-ID algorithm over this data and found that it performs well with R = 0.84, RMSE = 36.24, and bias = −7.17. Hence, the proposed algorithm has good generalisation capabilities and can be extended to other intrusion detection problems.

6.5. Limitations

Although F-TLBO-ID outperforms the benchmark algorithms in terms of accuracy, it has relatively higher computational time-complexity. However, once the model is trained, then the prediction of the barriers takes little time, which is essential for fast IDP. Further, the aging effect of the sensors may result in a performance mismatch. This can be easily avoided by periodic retraining of the proposed model or through routine maintenance of the deployed sensors.

7. Conclusion

This study presents F-TLBO-ID, a novel fuzzy-fed TLBO regression algorithm for fast IDP using WSNs. We considered synthetic features (*i.e.,* area of the RoI, effective sensing range, effective transmission range, number of sensor nodes, and fading parameter) as the potential features to map the number of k-barriers using the proposed algorithm. Based on our intensive mechanism following conclusion can be drawn;

- The number of sensor nodes turns out to be the most relevant feature in predicting the number of k-barriers for fast intrusion detection and prevention. It is followed by effective sensing and transmission range with equal weightage.
- The number of sensor nodes has a fluctuating positive impact on the k-barriers and area of the RoI, and the effective sensing range has a negative impact.
- The proposed F-TLBO-ID regression algorithm performs exceptionally well in terms of accuracy.
- F-TLBO-ID outperforms various benchmark and nature-inspired algorithms in accurately predicting the k-barriers.
- Relatively, F-TLBO-ID has high computational time-complexity.

Further, to assess the generalisation capability of the F-TLBO-ID algorithms, we evaluated its performance over the publicly available datasets and found satisfactory performance. This study is a step towards an evolutionary regression-based solution for fast IDP using WSNs. The proposed scheme can be implemented for near-real-time surveillance applications.

CRediT authorship contribution statement

Abhilash Singh: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Seyed Muhammad Hossein Mousavi:** Data curation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Jaiprakash Nagar:** Conceptualization, Data curation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

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

Code and data availability

The code and the relevant data are available for download at https://abhilashsingh.net/codes.html.

Acknowledgements

The authors would like to acknowledge IISER Bhopal and IIT Kharagpur for providing institutional support. They would like to thank to the editor and all the five anonymous reviewers for providing helpful comments and suggestions.

References

[1] A. Singh, J. Amutha, J. Nagar, S. Sharma, C.-C. Lee, AutoML-ID: automated machine learning model for intrusion detection using wireless sensor network, Sci. Rep. 12 (1) (2022) 1–14.
[2] J. Nagar, S.K. Chaturvedi, S. Soh, An analytical model to estimate the performance metrics of a finite multihop network deployed in a rectangular region, J. Netw. Comput. Appl. 149 (2020) 102466.
[3] A. Singh, S. Sharma, J. Singh, Nature-inspired algorithms for wireless sensor networks: A comprehensive survey, Comp. Sci. Rev. 39 (2021) 100342.
[4] E. Felemban, Advanced Border Intrusion Detection and Surveillance Using Wireless Sensor Network Technology, Scientific Research Publishing, 2013.
[5] A. Singh, S. Sharma, J. Singh, R. Kumar, Mathematical modelling for reducing the sensing of redundant information in WSNs based on biologically inspired techniques, J. Intell. Fuzzy Systems 37 (5) (2019) 6829–6839.
[6] D. Kandris, C. Nakas, D. Vomvas, G. Koulouras, Applications of wireless sensor networks: an up-to-date survey, Appl. Syst. Innov. 3 (1) (2020) 14.
[7] V. Kotiyal, A. Singh, S. Sharma, J. Nagar, C.-C. Lee, ECS-NL: An enhanced cuckoo search algorithm for node localisation in wireless sensor networks, Sensors 21 (11) (2021) 3576.
[8] Y. Wang, W. Fu, D.P. Agrawal, Gaussian versus uniform distribution for intrusion detection in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 24 (2) (2012) 342–355.
[9] J. Luo, S. Zou, Strong k-barrier coverage for one-way intruders detection in wireless sensor networks, Int. J. Distrib. Sens. Netw. 12 (6) (2016) 3807824.
[10] K. Ghosh, S. Neogy, P.K. Das, M. Mehta, Intrusion detection at international borders and large military barracks with multi-sink wireless sensor networks: An energy efficient solution, Wirel. Pers. Commun. 98 (1) (2018) 1083–1101.
[11] C.-I. Weng, C.-Y. Chang, C.-Y. Hsiao, C.-T. Chang, H. Chen, On-supporting energy balanced k-barrier coverage in wireless sensor networks, IEEE Access 6 (2018) 13261–13274.
[12] H. Huang, T. Gong, R. Zhang, L.-L. Yang, J. Zhang, F. Xiao, Intrusion detection based on k-coverage in mobile sensor networks with empowered intruders, IEEE Trans. Veh. Technol. 67 (12) (2018) 12109–12123.
[13] S. He, J. Chen, Y. Shu, X. Cui, K. Shi, C. Wei, Z. Shi, Efficient fault-tolerant information barrier coverage in internet of things, IEEE Trans. Wireless Commun. 20 (12) (2021) 7963–7976.
[14] T. Sood, S. Prakash, S. Sharma, A. Singh, H. Choubey, Intrusion detection system in wireless sensor network using conditional generative adversarial network, Wirel. Pers. Commun. (2022) 1–21.
[15] G.Y. Keung, B. Li, Q. Zhang, The intrusion detection in mobile sensor network, IEEE/ACM Trans. Netw. 20 (4) (2012) 1152–1161.
[16] J. Chang, X. Shen, W. Bai, X. Li, Energy-efficient barrier coverage based on nodes alliance for intrusion detection in underwater sensor networks, IEEE Sens. J. 22 (4) (2022) 3766–3776.
[17] P. Fan, C. Zhai, Distributed control strategy for layered barrier coverage of multi-agent systems in uncertain environments, 2023, arXiv preprint arXiv:2301.02061.
[18] A. Singh, K. Gaurav, G.K. Sonkar, C.-C. Lee, Strategies to measure soil moisture using traditional methods, automated sensors, remote sensing, and machine learning techniques: review, bibliometric analysis, applications, research findings, and future directions, IEEE Access (2023).
[19] A. Singh, J. Nagar, S. Sharma, V. Kotiyal, A Gaussian process regression approach to predict the k-barrier coverage probability for intrusion detection in wireless sensor networks, Expert Syst. Appl. 172 (2021) 114603.
[20] A. Singh, J. Amutha, J. Nagar, S. Sharma, C.-C. Lee, LT-FS-ID: Log-transformed feature learning and feature-scaling-based machine learning algorithms to predict the k-barriers for intrusion detection using wireless sensor network, Sensors 22 (3) (2022) URL: https://www.mdpi.com/1424-8220/22/3/1070.
[21] P.V. de Campos Souza, E. Lughofer, H. Rodrigues Batista, An explainable evolving fuzzy neural network to predict the k barriers for intrusion detection using a wireless sensor network, Sensors 22 (14) (2022) 5446.
[22] M. Arora, A. Pal, A deep learning approach to accurately predict the k-coverage probability in wireless sensor networks, Wirel. Pers. Commun. (2022) 1–21.

[23] J. Nagar, S.K. Chaturvedi, S. Soh, A. Singh, A machine learning approach to predict the k-coverage probability of wireless multihop networks considering boundary and shadowing effects, Expert Syst. Appl. 226 (2023) 120160.

[24] A. Taheri, K. RahimiZadeh, R.V. Rao, An efficient balanced teaching-learning-based optimization algorithm with individual restarting strategy for solving global optimization problems, Inform. Sci. 576 (2021) 68–104.

[25] Y. Xu, Z. Yang, X. Li, H. Kang, X. Yang, Dynamic opposite learning enhanced teaching–learning-based optimization, Knowl.-Based Syst. 188 (2020) 104966.

[26] F. Zou, D. Chen, Q. Xu, A survey of teaching–learning-based optimization, Neurocomputing 335 (2019) 366–383.

[27] Y.-R. Tsai, Sensing coverage for randomly distributed wireless sensor networks in shadowed environments, IEEE Trans. Veh. Technol. 57 (1) (2008) 556–564.

[28] D. Kim, H. Kim, D. Li, S.-S. Kwon, A.O. Tokuta, J.A. Cobb, Maximum lifetime dependable barrier-coverage in wireless sensor networks, Ad Hoc Netw. 36 (2016) 296–307.

[29] J. Amutha, J. Nagar, S. Sharma, A distributed border surveillance (dbs) system for rectangular and circular region of interest with wireless sensor networks in shadowed environments, Wirel. Pers. Commun. 117 (3) (2021) 2135–2155.

[30] S. Sharma, J. Nagar, Intrusion detection in mobile sensor networks: A case study for different intrusion paths, Wirel. Pers. Commun. 115 (3) (2020) 2569–2589.

[31] L. Ramos, J. Subramanyam, Maverick Research: Forget About Your Real Data — Synthetic Data Is the Future of AI, Gartner, 2021.

[32] R.J. Chen, M.Y. Lu, T.Y. Chen, D.F. Williamson, F. Mahmood, Synthetic data in machine learning for medicine and healthcare, Nat. Biomed. Eng. 5 (6) (2021) 493–497.

[33] D. Rankin, M. Black, R. Bond, J. Wallace, M. Mulvenna, G. Epelde, et al., Reliability of supervised machine learning using synthetic data in health care: Model to preserve privacy for data sharing, JMIR Med. Inf. 8 (7) (2020) e18910.

[34] A. Singh, V. Kotiyal, S. Sharma, J. Nagar, C.-C. Lee, A machine learning approach to predict the average localization error with applications to wireless sensor networks, IEEE Access 8 (2020) 208253–208263.

[35] A. Singh, J. Nagar, J. Amutha, S. Sharma, P2CA-GAM-ID: Coupling of probabilistic principal components analysis with generalised additive model to predict the k- barriers for intrusion detection, Eng. Appl. Artif. Intell. 126 (2023) 107137.

[36] R.J. Urbanowicz, M. Meeker, W. La Cava, R.S. Olson, J.H. Moore, Relief-based feature selection: Introduction and review, J. Biomed. Inf. 85 (2018) 189–203.

[37] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (1) (2003) 23–69.

[38] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.

[39] T. Hastie, R. Tibshirani, J.H. Friedman, J.H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Vol. 2, Springer, 2009.

[40] A. Goldstein, A. Kapelner, J. Bleich, E. Pitkin, Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation, J. Comput. Graph. Stat. 24 (1) (2015) 44–65.

[41] A. Singh, K. Gaurav, Deep learning and data fusion to estimate surface soil moisture from multi-sensor satellite images, Sci. Rep. 13 (1) (2023) 2251.

[42] A. Singh, M. Mehra, A. Kumar, M. Niranjannaik, D. Priya, K. Gaurav, Leveraging hybrid machine learning and data fusion for accurate mapping of malaria cases using meteorological variables in western India, Intell. Syst. Appl. 17 (2023) 200164.

[43] A. Singh, S. Patel, V. Bhadani, V. Kumar, K. Gaurav, AutoML-GWL: Automated machine learning model for the prediction of groundwater level, Eng. Appl. Artif. Intell. 127 (2024) 107405.

[44] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput.-Aided Des. 43 (3) (2011) 303–315.

[45] T.K. Pati, J.R. Nayak, B.K. Sahu, Application of TLBO algorithm to study the performance of automatic generation control of a two-area multi-units interconnected power system, in: 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, SPICES, IEEE, 2015, pp. 1–5.

[46] M. Singh, B. Panigrahi, A. Abhyankar, Optimal coordination of directional over-current relays using teaching learning-based optimization (TLBO) algorithm, Int. J. Electr. Power Energy Syst. 50 (2013) 33–41.

[47] T. Niknam, R. Azizipanah-Abarghooee, M.R. Narimani, A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems, Eng. Appl. Artif. Intell. 25 (8) (2012) 1577–1588.

[48] B.K. Sahu, T.K. Pati, J.R. Nayak, S. Panda, S.K. Kar, A novel hybrid LUS–TLBO optimized fuzzy-PID controller for load frequency control of multi-source power system, Int. J. Electr. Power Energy Syst. 74 (2016) 58–69.

[49] B.D. Raja, R. Jhala, V. Patel, Multi-objective optimization of a rotary regenerator using tutorial training and self-learning inspired teaching-learning based optimization algorithm (TS-TLBO), Appl. Therm. Eng. 93 (2016) 456–467.

[50] V. Toğan, M.A. Eirgash, Time-cost trade-off optimization of construction projects using teaching learning based optimization, KSCE J. Civ. Eng. 23 (1) (2019) 10–20.

[51] R.V. Rao, Teaching-learning-based optimization algorithm, in: Teaching Learning Based Optimization Algorithm, Springer, 2016, pp. 9–39.

[52] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems, Inf. Sci. 183 (1) (2012) 1–15.

[53] M. Črepinšek, S.-H. Liu, L. Mernik, A note on teaching–learning-based optimization algorithm, Inform. Sci. 212 (2012) 79–93.

[54] R.V. Rao, V. Savsani, J. Balic, Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems, Eng. Optim. 44 (12) (2012) 1447–1462.

[55] S.P. Das, S. Padhy, A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting, Int. J. Mach. Learn. Cybern. 9 (1) (2018) 97–111.

[56] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203.

[57] A. Likas, N. Vlassis, J.J. Verbeek, The global k-means clustering algorithm, Pattern Recognit. 36 (2) (2003) 451–461.

[58] A. Zhang, W. Shi, Mining significant fuzzy association rules with differential evolution algorithm, Appl. Soft Comput. 97 (2020) 105518.

[59] A. Telikani, A. Tahmassebi, W. Banzhaf, A.H. Gandomi, Evolutionary machine learning: A survey, ACM Comput. Surv. 54 (8) (2021) 1–35.

[60] J. Quinonero-Candela, C.E. Rasmussen, A unifying view of sparse approximate Gaussian process regression, J. Mach. Learn. Res. 6 (2005) 1939–1959.

[61] C.K. Williams, C.E. Rasmussen, Gaussian Processes for Machine Learning, Vol. 2, No. 3, MIT press Cambridge, MA, 2006.

[62] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133.

[63] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[64] J.J. Hopfield, Hopfield network, Scholarpedia 2 (5) (2007) 1977.

[65] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[66] P. Benardos, G.-C. Vosniakos, Optimizing feedforward artificial neural network architecture, Eng. Appl. Artif. Intell. 20 (3) (2007) 365–382.

[67] S.M.H. Mousavi, V. Charles, T. Gherman, An evolutionary pentagon support vector finder method, Expert Syst. Appl. 150 (2020) 113284.

[68] M.A. Khan, HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system, Processes 9 (5) (2021) 834.

[69] C. Liu, Z. Gu, J. Wang, A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning, IEEE Access 9 (2021) 75729–75740.

[70] C. Huang, Forest management and resource monitoring based on AMI intrusion detection algorithm and artificial intelligence, J. Ambient Intell. Humaniz. Comput. (2021) 1–15.

[71] A. Singh, K. Gaurav, A.K. Rai, Z. Beg, Machine learning to estimate surface roughness from satellite images, Remote Sens. 13 (19) (2021) 3794.

[72] A. Singh, J. Amutha, J. Nagar, S. Sharma, A deep learning approach to predict the number of k-barriers for intrusion detection over a circular region using wireless sensor networks, Expert Syst. Appl. 211 (2023) 118588.

[73] X.-S. Yang, Firefly algorithms for multimodal optimization, in: International Symposium on Stochastic Algorithms, Springer, 2009, pp. 169–178.

[74] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[75] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.

[76] R.G. Reynolds, An introduction to cultural algorithms, in: Proceedings of the 3rd Annual Conference on Evolutionary Programming, World Scientific Publishing, World Scientific, 1994, pp. 131–139.

[77] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.

[78] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.

[79] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The Bees Algorithm, Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005, pp. 44–48.

[80] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.

[81] S.M.H. Mousavi, Introducing bee-eater hunting strategy algorithm for IoT-based green house monitoring and analysis, in: 2022 Sixth International Conference on Smart Cities, Internet of Things and Applications, SCIoT, IEEE, 2022, pp. 1–6.

[82] S. Mousavi, S. Mirinezhad, Weevil damage optimization algorithm and its applications, J. Future Sustain. 2 (4) (2022) 133–144.