Journal of Soft Computing and Computer Applications

Volume 1 | Issue 1 Article 1001

2024

FOXANN: A Method for Boosting Neural Network Performance

Mahmood A. Jumaah
Department of Computer Science, University of Technology - Iraq, Baghdad, Iraq,
cs.22.27@grad.uotechnology.edu.iq

Yossra H. Ali
Department of Computer Science, University of Technology - Iraq, Baghdad, Iraq,
Yossra.H.Ali@uotechnology.edu.iq

Tarik A. Rashid
Department of Computer Science and Engineering, University of Kurdistan Hewlêr, Erbil, Iraq,
tarik.ahmed@ukh.edu.krd

S. Vimal
Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, India,
vimal@ritrjpm.ac.in

Recommended Citation

Jumaah, Mahmood A.; Ali, Yossra H.; Rashid, Tarik A.; and Vimal, S. (2024) "FOXANN: A Method for Boosting Neural Network Performance," *Journal of Soft Computing and Computer Applications*: Vol. 1: Iss. 1, Article 1001.
DOI: https://doi.org/10.70403/3008-1084.1001

JOURNAL OF SOFT COMPUTING AND COMPUTER APPLICATIONS 2024;1:1001

ORIGINAL STUDY

FOXANN: A Method for Boosting Neural Network Performance

Mahmood A. Jumaah [a,*], Yossra H. Ali [a], Tarik A. Rashid [b], S. Vimal [c]

[a] Department of Computer Science, University of Technology – Iraq, Baghdad, Iraq
[b] Department of Computer Science and Engineering, University of Kurdistan Hewlêr, Erbil, Iraq
[c] Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, India

ABSTRACT

Artificial neural networks play a crucial role in machine learning and there is a need to improve their performance. This paper presents FOXANN, a novel classification model that combines the recently developed Fox optimizer with ANN to solve ML problems. Fox optimizer replaces the backpropagation algorithm in ANN; optimizes synaptic weights; and achieves high classification accuracy with a minimum loss, improved model generalization, and interpretability. The performance of FOXANN is evaluated on three standard datasets: Iris Flower, Breast Cancer Wisconsin, and Wine. The results presented in this paper are derived from 100 epochs using 10-fold cross-validation, ensuring that all dataset samples are involved in both the training and validation stages. Moreover, the results show that FOXANN outperforms traditional ANN and logistic regression methods as well as other models proposed in the literature such as ABC-ANN, ABC-MNN, CROANN, and PSO-DNN, achieving a higher accuracy of 0.9969 and a lower validation loss of 0.0028. These results demonstrate that FOXANN is more effective than traditional methods and other proposed models across standard datasets. Thus, FOXANN effectively addresses the challenges in ML algorithms and improves classification performance.

Keywords: Artificial neural network, Classification, FOX, Machine learning, Optimization

1. Introduction

Machine learning (ML) consist of algorithms and techniques that aim to train machines (computers) to solve real-world problems, increase performance, increase production, enhance efficiency, and reduce errors caused by humans or traditional techniques [1]. Supervised learning is a subset of the ML field. It is termed *supervised* because the machine is provided with the inputs and outputs (target) during the training process. This type of training lets the algorithm know the relationships between the inputs and output to achieve optimal results in the prediction process when it has been fed with inputs only [2].

Received 26 March 2024; accepted 31 May 2024.
Available online 27 June 2024

* Corresponding author.
E-mail addresses: cs.22.27@grad.uotechnology.edu.iq (M. A. Jumaah), Yossra.H.Ali@uotechnology.edu.iq (Y. H. Ali), tarik.ahmed@ukh.edu.krd (T. A. Rashid), vimal@ritrjpm.ac.in (S. Vimal).

https://doi.org/10.70403/3008-1084.1001

Artificial neural networks (ANNs) are considered the most used supervised classifiers and have solved problems effectively since they were developed [3]. Furthermore, ANNs simulate the human brain's work by modeling it mathematically; for example, it mimics the connections among neurons, the functions of neurons, and the memory of the neurons [4, 5]. Research is continuing to improve the field of machine learning in general and neural networks in particular because ANNs play an essential role in most machine learning algorithms and problems such as convolutional neural networks (CNNs), deep learning, transfer learning, natural language processing (NLP), and generative adversarial network (GAN) [6, 7].

Improving the neural network is necessary due to increases in data and its complexity, computer resources, cloud solutions, cybersecurity threats, and generative AI [8]. However, optimization is widely applied in ANN such as swarm algorithms or evolutionary algorithms, i.e., artificial bee colony (ABC), ant colony optimization (ACO), particle swarm optimization (PSO), genetic algorithms (GA), and differential evolution (EA). Moreover, some researchers have examined the tuning of ANN hyperparameters, such as learning rate, initial weight, momentum, and epochs, while others have attempted to choose the best structure for an ANN (i.e., the number of hidden layers, and the size of the hidden layer) [9, 10].

This paper proposes an improved ANN based on the Fox optimizer (FOX), a state-of-the-art metaheuristic optimization algorithm that simulates the behavior of red foxes hunting prey [11]. This novel classification model, called FOXANN, aimed to increase accuracy and reduce loss in classification problems by eliminating the backpropagation algorithm from the ANN and employing FOX to optimize the synaptic weights. Moreover, the authors utilized FOX because it uses a static balance between exploitation and exploration that leverages the diversity in the search space to reduce the probability of entrapment in local optima. In contrast, the backpropagation algorithm uses gradient descent methods that focus on exploitation only, enabling the process of training ANN entrapped with local optima, especially when the number of samples is insufficient.

Three standard datasets have been used to evaluate the proposed FOXANN. First, the Iris Flower dataset consists of 150 instances and contains four features, with three classes: Setosa, Versicolor, and Virginica. Second, the Wine dataset consists of 178 instances and 13 features, with three classes determining the type of Wine. Lastly, Breast Cancer Wisconsin dataset consists of 569 instances and 30 features with two classes: Malignant and Benign [12].

The main contributions of this paper are highlighted as follows:

1. FOXANN utilizes the FOX algorithm instead of the backpropagation algorithm in an ANN to avoid the problem of local optima.
2. This replacement improves accuracy, robustness against local optima, and interpretability due to FOX's superior optimization capabilities compared to other optimization techniques.
3. FOXANN provides a deeper understanding of decision-making processes by optimizing synaptic weights more effectively than other techniques, and thus, FOXANN represents a significant advancement in the field of ML algorithms.

The rest of this paper is organized as follows: Section 2 presents the latest studies that have used optimization methods to improve ANNs. Section 3 presents the materials and methods used in this paper. Section 4 and Section 5 list and discuss the results with comparisons. Finally, Section 6 presents conclusions and suggestions for future directions in ANN improvement.

2. Related work

This section reviews the related literature on ANN improvement using different optimization techniques. In recent years, many optimizers, such as ABC, GA, EA, and PSO, have been used to improve ANN, and these improvements have been crucial for solving ML problems.

The study by Awan et al. [13] proposed a hybrid model that utilized ABC and an ANN to efficiently optimize a set of neuron connection weights for an ANN, producing an alternative scheme to traditional ANN training used for short-term electric load prediction. The results of the proposed model showed an improved version of the ANN with higher accuracy and a faster convergence rate with prediction tasks [13]. A recently proposed model that adopts ABC used modular neural networks (MNNs) to optimize the weights. The results of ABC-MNN showed generalization performance with higher training accuracy on the Iris Flower dataset [14]. Moreover, a model called CROANN, which uses chemical reaction optimization (CRO), was proposed to design the structure and tune the weights of an ANN. CROANN was evaluated using the Iris Flower and Breast Cancer Wisconsin datasets, and the results showed that CRO is superior to many EA strategies with ANN [15]. Furthermore, six hybrid ANN models based on adaptive EA were proposed to optimize ANN parameters and feature selection evaluated on two meteorological datasets as prediction tasks. The results showed that these proposed models can be used as generic models with improved forecasting accuracy [16]. A multi-objective optimizer called GA-ANN-GA, was presented by using GA before training to tune the weight of an ANN automatically. Then, the GA was used again after the ANN training phase; this model was used to optimize heat transfer in rectangular perforated plate fins and the results show that optimized fin geometry can be used in many heat transfer problems with less effort [17]. Another novel hybrid algorithm called BRKGA-NN that used GA with ANN was proposed to determine the connection weights, bias values of the hidden neurons, and the number of hidden neurons in an ANN. The BRKGA-NN was evaluated on time-series datasets, and the results showed that BRKGA-NN provided more effective predictions than support vector regression (SVR), BPANN, and autoregressive integrated moving average (ARIMA) [18]. PSO was integrated with deep neural networks (DNNs) in a recent study [19] that proposed a new method called PSO-DNN to optimize the number of hidden layer nodes (neurons). An evaluation of PSO-DNN on digital modulation classification tasks confirmed that the proposed method is effective in optimizing DNN [19, 20]. A recent study suggested a CNN hyperparameter optimization method using linearly decreasing weight PSO (LDWPSO). The proposed LDWPSO-CNN method was evaluated on MNIST and CIFAR-10 datasets. The results showed increased accuracy by 4% from a baseline CNN on the MNIST dataset and by 41% on the CIFAR-10 dataset for 10 epochs [21, 22].

Table 1 summarizes related work, including the proposed models, datasets, evaluation tasks, and key findings. Furthermore, several studies have contributed to improving ANNs based on various optimization methods, which were reviewed in this paper [23].

3. Materials and methods

This section explains the proposed method by presenting a detailed overview of the working principles and definitions of ANN, FOX, and the proposed FOXANN.

Table 1. Summary of hybrid ANN models based on optimization techniques.

Study	Model	Dataset	Task	Results
[13]	ABC-ANN	Electric Load	Prediction	Improved ANN accuracy and faster convergence rate.
[14]	ABC-MNN	Iris Flower	Classification	Higher training accuracy with generalization performance.
[15]	CROANN	Iris Flower, Breast Cancer Wisconsin	Classification	Superior performance compared to many EA strategies with ANN.
[16]	Adaptive EA-ANN	Meteorological	Prediction	Improved prediction accuracy.
[17]	GA-ANN-GA	Heat Transfer	Optimization	Optimized fin geometry with less effort for heat transfer problems.
[18]	BRKGA-NN	Time-Series	Prediction	More effective predictions compared to SVR, BPANN, and ARIMA.
[19]	PSO-DNN	Digital Modulation	Classification	Effective optimization of DNN for digital modulation classification tasks.
[22]	LDWPSO-CNN	MNIST, CIFAR-10	Classification	Increased accuracy compared to CNN on MNIST and CIFAR-10 datasets.

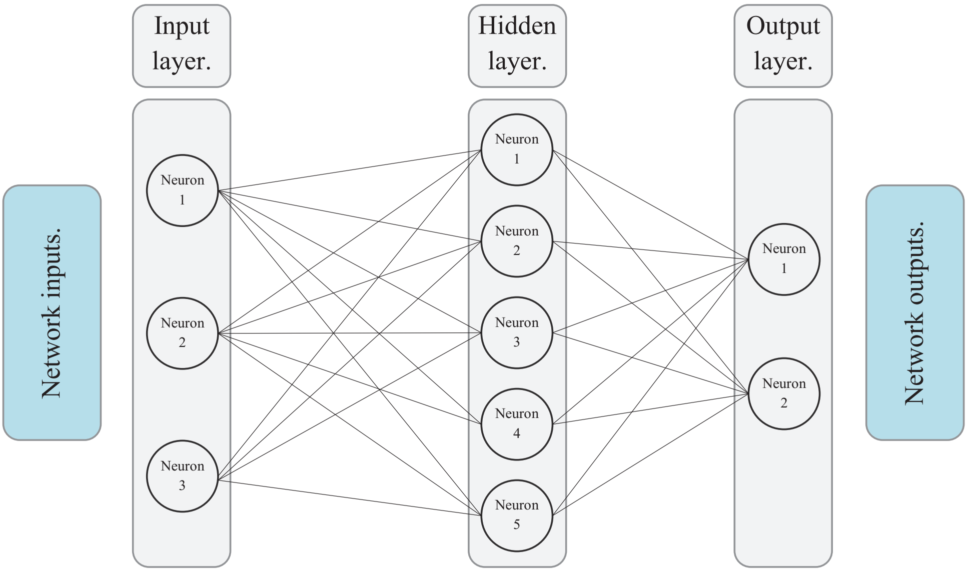

Fig. 1. Artificial neural network architecture [27].

3.1. Artificial neural networks

The ANN consists of three main layers: the input, hidden, and output layers; each layer is a cluster of multiple nodes (neurons). The input layer accepts and feeds the input data to the hidden layer. Then, the output layer computes the last result (see Fig. 1) [24]. The direction of data from the input to the output is called a feed-forward neural network. ANNs can be used for several tasks, including classification, clustering, and pattern recognition. The input data are multiplied by the value of the weights and passed to the hidden layer neurons, each of which has an activation function to compute the neuron's output as seen in Equation (1) [25]:

$$z_j = f\left(\sum_{i=1}^{n} (x_i \times w_{ij}) + b_j\right) \tag{1}$$

and

$$f = \frac{1}{1 + e^{-x}} \tag{2}$$

where x_i is the data input, w_{ij} is the weight between the i^{th} input neuron and the j^{th} hidden neuron, b_i is the bias of the j^{th} hidden neuron, and z_j represents the output of the j^{th} neuron in the hidden layer. Moreover, the neuron's output (z_j) is fed into f activation function (i.e., sigmoid Equation (2)). Many activation functions exist, such as sigmoid, tanh, RELU, and softmax. Each function is problem-dependent (i.e., the sigmoid function, Equation (2), is employed when the output range is expected to be between zero and one, in contrast, the tanh function is employed when the output is between −1 and 1) [26].

The input of ANN must be standardized to reduce the variance between data. Moreover, if the output is text data, it must be converted into numerical form to be accepted by the neural network functions. ANNs require standardized input to mitigate the variance between data and provide optimal performance. Additionally, if the output (label) is text data, it must be converted into a numerical format compatible with the neural network's functions. This conversion and standardization enable the network to process and analyze the data effectively [28].

$$\mathrm{E} = \frac{1}{2}\sum_{i=1}^{n}(y_i - y_i')^2 \tag{3}$$

$$\delta_j = (y_j - y_j') \cdot f'(z_j) \tag{4}$$

$$w_{ij} = w_{ij} + \eta \cdot x_i \cdot \delta_j \tag{5}$$

In the above equations, E is the error between the expected (y_i) and actual (y_i') outputs, δ_j is the error gradient for the unit y_j, $f'(z_j)$ is the derived activation function, w_{ij} is the weight connecting unit i to j, updated with learning rate η and input x_i [29, 30].

The most important algorithm in an ANN is backpropagation, which involves feeding the output to the network input again to update the weights. Backpropagation has four main steps: the first is the computation of the error between the expected and the actual output (target) using error functions like mean squared error (MSE), as seen in Equation (3). The second is deriving the error gradient from outputs, which are used to update the weights for the last layer in the network using Equation (4). The third is error backpropagation, in which the error propagates to other layers. Lastly, the weights are updated based on the derived error and learning rate by Equation (5) [31].

3.2. Fox optimizer algorithm

FOX is an optimization method motivated by red foxes' hunting behavior. It searches for the best position (solution) using static exploration and exploitation. Through exploration, FOX uses a random walk policy, aided by its ability to detect ultrasound to find prey. Upon detecting prey, the FOX agent enters the exploitation phase and waits for the sound of the prey's ultrasound. The FOX agent evaluates the time required to catch its prey based on the sound's travel time and then jumps [32]. The FOX algorithm requires two inputs: an objective function and bounds. The objective function calculates the fitness value, while the bounds determine the range of values for each variable in the optimization problem. Algorithm 1 below introduces a visual representation of how the FOX is used to find the best position; further details can be found in [11].

Algorithm 1. FOX optimizer [11].

Input: Objective function, Problem bounds

```
Initialize the red fox population (X)
While iter < Max_iter
    Initialize variables
    Calculate the fitness of each search agent
    Select Best_Position and Best_Fitness among the population
    If $fitness_{i+1}$ < $fitness_i$
        Best_Fitness = $fitness_{i+1}$
        Best_Position = $X_{(i)}$
    EndIf
      If r >= 0.5
          Initialize time randomly;
          Calculate Distance_Sound_travels
          Calculate Sp_S
          Calculate the distance from fox to prey
          Tt = average time;
          T = Tt/2;
          Calculate jump
          If p > 0.18
              Find $X_{(i+1)}$ using Equation (6)
          Elseif p <= 0.18
              Find $X_{(i+1)}$ using Equation (7)
          EndIf
      else
          Find MinT
          Explore $X_{(i+1)}$ using Equation (8)
      EndIf
      Clip the position if it goes beyond the limits
      Evaluate search agents by fitness
      Update Best_Position
      iter = iter + 1
  EndWhile
  return Best_Position & Best_Fitness
```

Output: Best Solution, Best Fitness

FOX employs a static trade-off between exploration and exploitation (50% for each). The algorithm uses random walks to find the red fox's prey during exploration. During exploitation, the algorithm calculates the distance to the prey, jump height, and new positions.

$$X_{(i+1)} = \mathrm{DF}P_i * \mathrm{Jump}_i * c_1 \tag{6}$$

$$X_{(i+1)} = \mathrm{DF}P_i * \mathrm{Jump}_i * c_2 \tag{7}$$

The two constants, c_1 and c_2, have been fixed at 0.18 and 0.82, respectively. These constants are derived from observations made while studying the jumping behavior of a red FOX. It has been observed that the FOX's jumps are either directed toward the northeast or the opposite direction. The FOX explores the Red Fox's surroundings by using the following equation to calculate its new position (this is considered exploration) [33]:

$$X_{(i+1)} = \mathrm{Best}\mathrm{Position} * \mathrm{rand}(1, dimension) * MinT * a \tag{8}$$

and

$$tt = \frac{sum(Time_{st_{it}})}{dimension}, \quad MinT = Min(tt) \tag{9}$$

$$\mathrm{a} = 2\left(iter - \frac{1}{max(iter)}\right) \tag{10}$$

where tt is the time average equal to the summation of the time variable divided by the dimension of the problem, $iter$ is the current iteration, and $max(iter)$ is maximum iterations. Calculating both the $MinT$ and a variable has a vital effect on the search phase to move toward a solution that is close to the best solution. Using a random function $rand(1, dimension)$ ensures that the fox walks stochastically to explore the prey [11, 34].

3.3. Fox artificial neural network

The proposed FOXANN method is inspired by the use of FOX to tune the hyperparameter automatically in Q-learning [35] and by FOX's superior performance in solving common optimization problems. This section presents an AAN based on FOX intended to solve classification tasks in standard datasets, which are expected to improve the ANN's ability to solve ML problems. The ANN structure in the proposed method, FOXANN, remains unchanged, and the essential improvement lies in the backpropagation algorithm. Since this step was eliminated, the authors use FOX to improve the weights based on the minimization of MSE in Equation (3).

The proposed model's architecture is simple, as shown in Fig. 2. Once the data have been processed for input, they are moved directly into the ANN. The feed-forward process is then performed, where FOX iteratively optimizes the weights with minimum loss and returns them to the ANN. The ANN uses the optimized weights and activation function to compute the final output based on Equation (1).

3.4. Materials

The Iris Flower, a standard benchmark dataset in ML, contains 150 objects and four features. These features represent the sepals and petals of the iris flower, with three distinct classes: Setosa, Versicolor, and Virginica [36]. Furthermore, a well-known Breast Cancer Wisconsin dataset offers insights into breast cancer diagnosis with 569 instances and 30 features. Features are extracted from fine needle aspirates of breast tissue. This

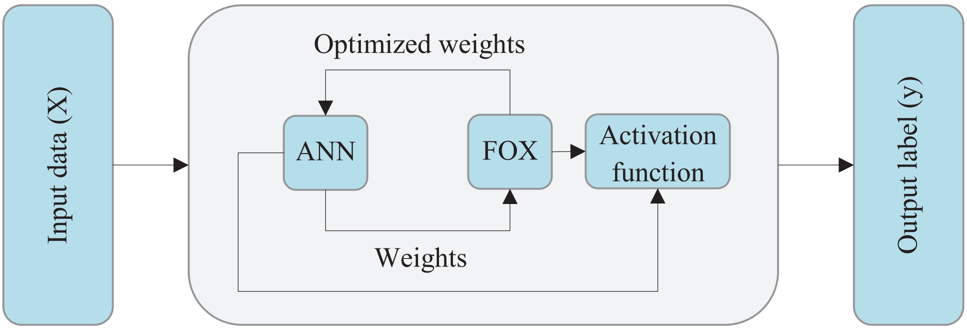

Fig. 2. FOXANN architecture.

Table 2. Datasets' characteristics.

Dataset	Instances	Features	Classes
Iris Flower	150	4	3
Breast Cancer Wisconsin	569	30	2
Wine	178	13	3

dataset provides valuable information for classifying malignant and benign tumors [37]. Moreover, the Wine dataset, which contains 178 instances and 13 features, including attributes such as alcohol content, acidity levels, and color intensity, is used to classify wines into one of three classes [21]. These datasets are explored in Table 2 and are basic for classification tasks, helping researchers to explore and evaluate various ML algorithms.

All the datasets have been preprocessed using min-max normalization from Equation (11) to ensure that different feature scales are processed in the same range, typically between 0 and 1. Moreover, the preprocessing using min-max normalization helps to reduce variations in feature scales, ensuring that the model remains robust against outliers.

$$X_{normalized} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{11}$$

In the above equation, X is the feature vector used by the models as input, $X_{\min}$ is the minimum value in the vector, and $X_{\max}$ is the maximum value in the vector.

4. Results

The performance of the FOXANN model was evaluated using the Iris Flower, Wine, and Breast Cancer Wisconsin datasets, considering standard evaluation metrics such as validation loss, accuracy, recall, precision, and F-score. The ANN topology has been designed based on a previous study [38] that recommends using two hidden layers with a small-to-median dataset. The first layer was twice the size of the input layer, and the second layer was half the size of the input layer to better generalization. Additionally, the initial weights of the ANN range [−3, 3] are based on a trial-and-error strategy. In this paper, many experiments have been conducted to evaluate the proposed model's performance against traditional ANN and logistic regression (LR) classifiers. Furthermore, it compares the most important results from the literature with the proposed model FOXANN. The figures and tables below show the results obtained after 100 epochs using the cross-validation method with 10 folds, which ensures that all data are involved in the training process for each classifier [39]. In ML, visualizing the results is a crucial step when analyzing and comparing the performances of different methods. The performance of each classifier's training process was measured by computing the validation loss using the MSE. Fig. 3(A), (B), and (C) show the validation loss on the Iris Follower dataset, Breast Cancer Wisconsin dataset, and Wine dataset, respectively. Furthermore, Fig. 3(D) presents the average validation loss for all datasets.

Table 3 shows the results of the evaluations performed on three different datasets using the proposed FOXANN, ANN, and LR. Firstly, the Iris Flower dataset was evaluated using FOXANN and reached an accuracy of 0.9776 and a validation loss of 0.0107. The ANN achieved an accuracy of 0.9754 and a validation loss of 0.0296, and the LR produced an accuracy of 0.8555 with a validation loss of 0.1193. The Breast Cancer Wisconsin dataset was then evaluated using FOXAN and achieved an accuracy of 0.9749 with a validation loss of 0.0075, while the ANN achieved an accuracy of 0.9654 with a validation loss of

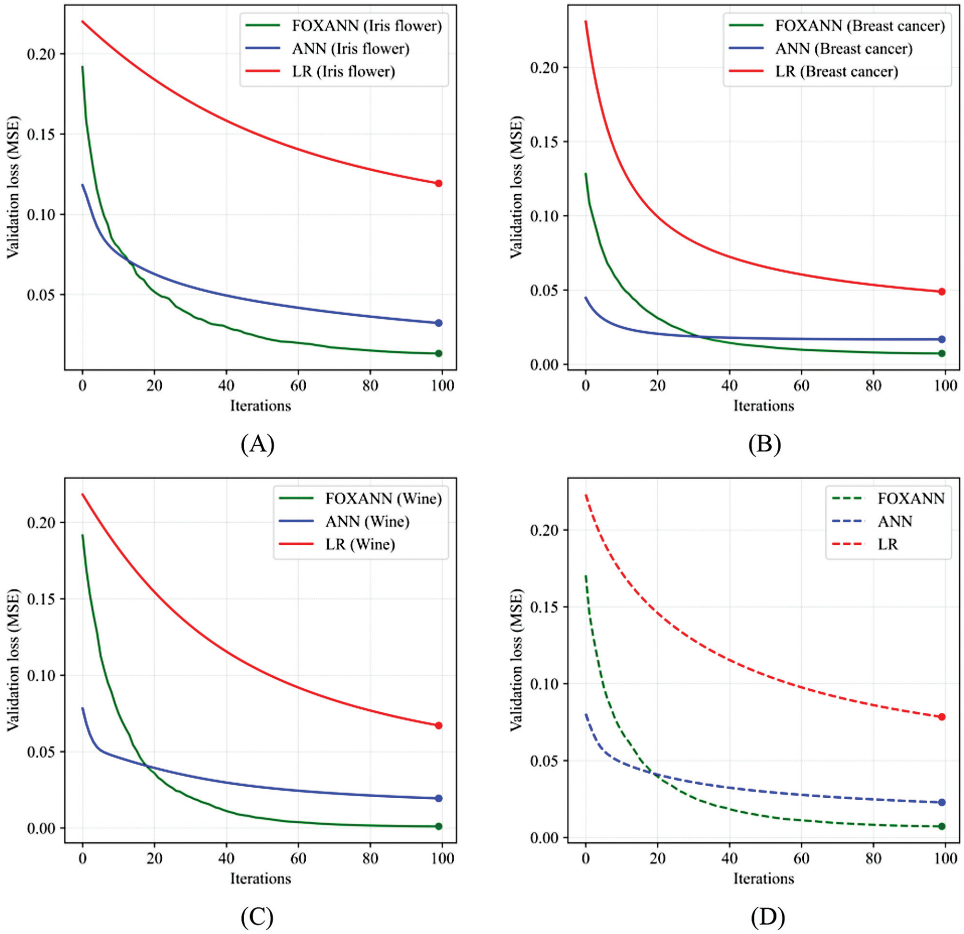

Fig. 3. Validation loss on the dataset: (A) Iris Flower, (B) Breast Cancer Wisconsin, (C) Wine, and (D) average validation loss.

Table 3. FOXANN Performance results.

Dataset	Model	Accuracy	Loss	Precision	Recall	F-Score
Iris Flower	FOXANN	0.9776	0.0107	0.9664	0.9664	0.9664
	ANN	0.9754	0.0296	0.9664	0.9598	0.9631
	LR	0.8555	0.1193	0.9891	0.5733	0.7239
Breast Cancer Wisconsin	FOXANN	0.9749	0.0075	0.9749	0.9749	0.9749
	ANN	0.9654	0.0201	0.9654	0.9654	0.9654
	LR	0.9596	0.0489	0.9596	0.9596	0.9596
Wine	FOXANN	0.9969	0.0028	0.9834	0.9875	0.9902
	ANN	0.9844	0.0137	0.9844	0.9788	0.9816
	LR	0.9421	0.0671	0.9684	0.854	0.9076
Average results	FOXANN	0.9831	0.0070	0.9749	0.9763	0.9772
	ANN	0.9751	0.0211	0.9721	0.968	0.9700
	LR	0.9191	0.0784	0.9724	0.7956	0.8637

0.0201, and LR produced an accuracy of 0.9596 with a validation loss of 0.0489. Lastly, the evaluation of the Wine dataset using FOXANN achieved an accuracy of 0.9969 with a validation loss of 0.0028, while the ANN achieved an accuracy of 0.9844 with a validation loss of 0.0137, and LR produced an accuracy of 0.9421 with a validation loss of 0.0671.

Furthermore, Table 3 shows that the average accuracy and average validation loss using FOXAN, ANN, and LR, were 0.9831, 0.9751, and 0.9191, respectively. It also presents important metrics, such as precision, recall, and F-score.

5. Discussion

The results in Table 3 and Fig. 3 show that FOXANN performs well compared with ANN and RL methods, and it outperformed both the ANN and LR for the Iris Flower dataset, achieving the highest accuracy of 0.9776. This indicates that FOXANN effectively learned the patterns in the datasets and produced accurate predictions. It also achieved the lowest validation loss of 0.0107, indicating better generalization ability compared to ANN and LR. In the Breast Cancer Wisconsin dataset, FOXANN performed effectively, with an accuracy of 0.9749 and the lowest validation loss of 0.0075. These results suggest that FOXANN could recognize the fundamental patterns within the datasets effectively, leading to accurate predictions and performance. The ANN performed well, with an accuracy of 0.9654 and a validation loss of 0.0201, and LR performed the worst, with an accuracy of 0.9596 and the highest validation loss of 0.0489.

The FOXANN performed better than the ANN and LR in evaluating the Wine dataset, achieving the highest accuracy of 0.9969 and the lowest validation loss of 0.0028. This result shows that FOXANN can effectively identify the patterns in the Wine dataset, resulting in superior performance. The average accuracies for FOXANN, ANN, and LR were 0.9831, 0.9751, and 0.9191, respectively; the validation losses were 0.007, 0.0211, and 0.0784, respectively.

Finally, FOXANN surpassed both the ANN and LR algorithms due to its unique capabilities based on the static balance between exploration and exploitation, which effectively mitigates the local optima trap in the backpropagation algorithm. Table 4 highlights FOXANN's dominance over the proposed models in the literature. It outperformed the ANN-based optimizers, such as ABC, CRO, and RFC, in terms of accuracy. Additionally, FOXANN surpassed GA, PSO, EA, and LDWPSO in terms of interpretability and generality, with a significantly low error rate of 0.007 compared to 0.01 (GA), 0.009 (PSO and

Table 4. Comparison between FOXANN and other models.

Model	Reference	Dataset	Accuracy
FOXANN	Proposed	Iris Flower	0.9776
ABC-NN	[14]		0.9666
ABC-MNN	[14]		0.9722
CROANN	[15]		0.9656
FOXANN	Proposed	Breast Cancer Wisconsin	0.9749
Decision tree (DT)	[20]		0.9400
DT (PCA transformation)	[20]		0.9600
CROANN	[15]		0.9611
FOXANN	Proposed	Wine	0.9969
RFC	[21]		0.8979
KNN	[21]		0.8666

LDWPSO), and 0.022 (EA). These results emphasize FOXANN's broader applicability and superior interpretability compared to ML methods that operate based on best optimizers.

6. Conclusion

This paper presents a novel model called FOXANN, which combines the FOX optimization algorithm with an ANN algorithm to effectively increase performance in solving machine learning problems. The backpropagation algorithm is replaced with FOX to optimize the weights, improve classification performance, and avoid the local optima that may be caused by the backpropagation algorithm. Experimental results on standard datasets (Iris Flower, Breast Cancer Wisconsin, and Wine) show that FOXANN achieves higher accuracy and lower validation loss than traditional ANN and LR methods, as well as other methods proposed in the literature. Future studies may focus on enhancing the FOX optimizer and integrating it with more complex ML models, such as deep learning models, while considering imbalanced or imagery datasets. Furthermore, the FOX algorithm might be used to present optimal ANN or CNN structure designs based on the problem features to reduce the model's complexity.

References

1. I. H. Sarker, "Machine learning: algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, p. 160, Mar. 2021.
2. J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, Feb. 2020.
3. S. S. Jasim, A. K. Abdul Hassan, and S. Turner, "Optimizing artificial neural networks using levy-chaotic mapping on wolf pack optimization algorithm for detect driving sleepiness," *Iraqi Journal of Computers, Communications, Control & Systems Engineering*, vol. 22, no. 3, pp. 128–136, Sep. 2022.
4. A. H. Alrubaie, M. A. A. Khodher, and A. T. Abdulameer, "Object detection using deep learning methods: a review," *Iraqi Journal of Computers, Communications, Control & Systems Engineering*, vol. 23, no. 2, pp. 136–152, Jun. 2023.
5. X. Yao, "Evolving artificial neural networks," in *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, Sep. 1999.
6. S. Khatir, S. Tiachacht, C. Le Thanh, E. Ghandourah, S. Mirjalili, and M. Abdel Wahab, "An improved artificial neural network using arithmetic optimization algorithm for damage assessment in FGM composite plates," *Composite Structures*, vol. 273, p. 114287, Oct. 2021.
7. A. H. Ahmed, H. B. Alwan, and M. Çakmak, "Convolutional neural network-based lung cancer nodule detection based on computer tomography," *in Proceedings of Data Analytics and Management: ICDAM*, Springer Nature Singapore, pp. 89–102, Mar. 2023.
8. A. R. Kareem and H. S. Abdullah, "A survey on business intelligence approach based on deep learning," *Iraqi Journal of Computers, Communications, Control & Systems Engineering*, vol. 23, no. 4, pp. 58–70, Dec. 2023.
9. Z. K. Abdalrdha, A. M. Al-Bakry, and A. K. Farhan, "A hybrid CNN-LSTM and XGBoost approach for crime detection in tweets using an intelligent dictionary", *Revue d'Intelligence Artificielle*, vol. 37, no. 6, pp. 1651–1661, Dec. 2023.
10. C. J. Haug and J. M. Drazen, "Artificial intelligence and machine learning in clinical medicine, 2023," *New England Journal of Medicine*, vol. 388, no. 13, pp. 1201–1208, Mar. 2023.
11. H. Mohammed and T. Rashid, "FOX: a FOX-inspired optimization algorithm," *Applied Intelligence*, vol. 53, no. 1, pp. 1030–1050, Apr. 2022.
12. A. A. Badr and A. K. Abdul-Hassan, "Estimating age in short utterances based on multi-class classification approach," *Computers, Materials & Continua*, vol. 68, no. 2, pp. 1713–1729, Jan. 2021.
13. S. M. Awan, M. Aslam, Z. A. Khan, and H. Saeed, "An efficient model based on artificial bee colony optimization algorithm with neural networks for electric load forecasting," *Neural Computing and Applications*, vol. 25, pp. 1967–1978, Aug. 2014.
14. C. Zhuo-Ming, W. Yun-Xia, L. Wei-Xin, X. Zhen, and X. Han-Lin-Wei, "Artificial bee colony algorithm for modular neural network," *Advances in Neural Networks-ISNN 2013*, Springer Berlin Heidelberg, pp. 350–356, Jul. 2013.

15. J. J. Q. Yu, A. Y. S. Lam, and V. O. K. Li, “Evolutionary artificial neural network based on chemical reaction optimization,” in *2011 IEEE Congress of Evolutionary Computation (CEC)*, New Orleans, LA, USA, 2011, pp. 2083–2090, Jun. 2011.
16. F. Kılıç, İ. H. Yılmaz, and Ö. Kaya, “Adaptive co-optimization of artificial neural networks using evolutionary algorithm for global radiation forecasting,” *Renewable Energy*, vol. 171, pp. 176–190, Jun. 2021.
17. B. Chidambaram, M. Ravichandran, A. Seshadri, and V. Muniyandi, “Computational heat transfer analysis and genetic algorithm-artificial neural network-genetic algorithm-based multiobjective optimization of rectangular perforated plate fins,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, no. 2, pp. 208–216, Feb. 2017.
18. Z. I. Erzurum Cicek and Z. Kamisli Ozturk, “Optimizing the artificial neural network parameters using a biased random key genetic algorithm for time series forecasting,” *Applied Soft Computing*, vol. 102, pp. 107091, Apr. 2021.
19. W. Shi, D. Liu, X. Cheng, Y. Li, and Y. Zhao, “Particle swarm optimization-based deep neural network for digital modulation recognition,” *IEEE Access*, vol. 7, pp. 104591–104600, Dec. 2019.
20. R. Hazra, M. Banerjee, and L. Badia, “Machine learning for breast cancer classification with ANN and decision tree,” in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada, 2020, pp. 0522–0527, Nov. 2020.
21. P. Dhaliwal, S. Sharma, and L. Chauhan, “Detailed study of wine dataset and its optimization,” *International Journal of Intelligent Systems and Applications*, vol. 14, no. 5, pp. 35–46, Oct. 2022.
22. T. Serizawa and H. Fujita, “Optimization of convolutional neural network using the linearly decreasing weight particle swarm optimization,” *arXiv*, Mar. 2023.
23. M. G. M. Abdolrasol *et al.*, “Artificial neural networks based optimization techniques: a review,” *Electronics*, vol. 10, no. 21, pp. 2689, Nov. 2021.
24. O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: a survey,” *Heliyon*, vol. 4, no. 11, p. e00938, Nov. 2018.
25. Y.-c. Wu and J.-w. Feng, “Development and application of artificial neural network,” *Wireless Personal Communications*, vol. 102, no. 2, pp. 1645–1656, Dec. 2017.
26. A. D. Rasamoelina, F. Adjailia, and P. Sincak, “A review of activation function for artificial neural network,” *in 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herlany, Slovakia, 2020, pp. 281–286, Jan. 2020.
27. Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, “A survey on neural network interpretability,” in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, Oct. 2021.
28. A. Carré *et al.*, “Standardization of brain MR images across machines and protocols: bridging the gap for MRI-based radiomics,” *Scientific Reports*, vol. 10, no. 1, p. 12340, Jul. 2020.
29. T. O. Hodson, “Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not,” *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022.
30. M. A. Jumaah, A. I. Shihab, and A. A. Farhan, “Epileptic seizures detection using DCT-II and KNN classifier in long-term EEG signals,” *Iraqi Journal of Science*, pp. 2687–2694, Oct. 2020.
31. N. A. Mohammed and A. Al-Bazi, “An adaptive backpropagation algorithm for long-term electricity load forecasting,” *Neural Computing and Applications*, vol. 34, no. 1, pp. 477–491, Aug. 2021.
32. D. Połap and M. Woźniak, “Red fox optimization algorithm,” *Expert Systems with Applications*, vol. 166, pp. 114107, Mar. 2021.
33. M. H. Nadimi-Shahraki, H. Zamani, Z. A. Varzaneh, and S. Mirjalili, “A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations,” *Archives of Computational Methods in Engineering*, vol. 30, no. 7, pp. 4113–4159, May 2023.
34. O. O. Akinola, A. E. Ezugwu, J. O. Agushaka, R. A. Zitar, and L. Abualigah, “Multiclass feature selection with metaheuristic optimization algorithms: a review,” *Neural Computing and Applications*, vol. 34, no. 22, pp. 19751–19790, Aug. 2022.
35. M. A. Jumaah, Y. H. Ali, and T. A. Rashid, “Q-FOX learning: breaking tradition in reinforcement learning,” *arXiv*, Mar. 2024.
36. L. Omelina, J. Goga, J. Pavlovicova, M. Oravec, and B. Jansen, “A survey of iris datasets,” *Image and Vision Computing*, vol. 108, pp. 104109, Apr. 2021.
37. K. M. M. Uddin, N. Biswas, S. T. Rikta, and S. K. Dey, “Machine learning-based diagnosis of breast cancer utilizing feature optimization technique,” *Computer Methods and Programs in Biomedicine Update*, vol. 3, pp. 100098, Jan. 2023.
38. D. Hunter, H. Yu, M. S. Pukish, J. Kolbusz, and B. M. Wilamowski, “Selection of proper neural network sizes and architectures—a comparative study,” *in IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 228–240, May 2012.
39. B. G. Marcot and A. M. Hanea, “What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?,” *Computational Statistics*, vol. 36, no. 3, pp. 2009–2031, Jun. 2020.