ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2024.0475

PIONEERING MEDICAL DIAGNOSIS - NEURO-FUZZY SYSTEMS AND SWARM INTELLIGENCE IN HEALTHCARE APPLICATIONS

S.G. Sree Lekshmi[1] and M. Sankari[2]

[1]Department of Mathematics, Manonmaniam Sundaranar University, India
[2]Department of Mathematics, Lekshmipuram College of Arts and Science, India

Abstract

This study introduces a hybrid approach for lung cancer detection, combining Neuro-Fuzzy Systems for robust feature extraction and the Firefly Algorithm for accurate classification of lung nodules as benign or malignant. The methodology is validated through comprehensive experiments using standard datasets and compared against established techniques like SVM-ANN and RBF-PSO. The research highlights the interpretability and learning capabilities of Neuro-Fuzzy Systems and the effectiveness of the Firefly Algorithm in medical image classification, showcasing improvements in accuracy and reliability over traditional methods.

Keywords:

Healthcare, Diagnosis, Medical, Fuzzy System, Swarm Intelligence

1. INTRODUCTION

The rapid growth of artificial intelligence, machine learning, and deep learning over the course of the previous several decades has contributed to the successful resolution of a few practical difficulties. The success of these domains has resulted in the development of several different methodologies, including fuzzy logic, genetic programming, swarm intelligence, and hybrid approaches such as neuro-fuzzy and genetic fuzzy systems. All these methods have been helpful in the design and research of complex intelligent systems. Within the realm of artificial intelligence (AI), deep neural networks (DNNs) and other deep learning approaches have made significant progress in addressing the difficulties that have been plaguing the industry for years.

According to [1], the term “deep” was developed to describe this network because it is more complicated than the normal “shallow” neural networks which are used in the field. Conventional neural networks have significant limitations when it comes to the processing of natural and raw data. In the creation of pattern-recognition or machine-learning systems, feature extraction, which is the process of transforming raw data into an acceptable internal representation or feature vector, has been an area that has been a subject of intense domain expertise and rigorous engineering for a very long time [2].

A deep neural network (DNN) employs representation learning, which enables a machine to automatically learn the representations required for detection or classification by employing several hidden layers. This contrasts with standard neural networks, which only have one hidden layer [3]. Since it is so effective at locating intricate structures within high-dimensional data, a DNN is beneficial in a wide variety of scientific, business, and engineering sectors.

Although DNNs are excellent for addressing problems involving enormous amounts of data, the model’s outstanding accuracy comes at a cost: it is highly complicated. Because of this, it is essential to keep a few things in mind before utilising this form of network to resolve problems. Because it makes use of several hidden layers, a DNN makes it possible to create a more in-depth analytical model; however, the complexity of the process of computing grows with each new layer [4]. In addition, the traditional neural network that is trained by the gradient descent optimisation approach serves as the source of inspiration for these varieties of networks. For this reason, DNNs frequently experience the issue of becoming trapped in the local minimum. In addition to this, DNNs are subject to criticism for their predictions being opaque and human untraceable. This is a result of its black-box design, which is a significant limitation of the model [5]. It is not always possible to rely on the findings that are generated by these deep neural networks. This may lead to a situation in which analysts and DNNs are unable to communicate at some time. Because of this limitation, such networks are typically not suitable to most problems that occur in the real world, particularly those that require rigorous verification of the results that were anticipated. The authors of [6] [7] are the only three studies that have addressed these challenges by merging DNNs with fuzzy systems to develop a novel deep neuro-fuzzy system (DNFS). Some of these studies are included below. In situations when conventional binary logic is either not practicable or not practical, fuzzy systems, which are structures that are constructed using fuzzy methodologies with the intention of information processing, are frequently the most suitable alternative. The primary characteristic of fuzzy conditional IF THEN rules is that they are a symbolic representation of knowledge. Because of this, the novel combination of DNNs with fuzzy systems has demonstrated that fuzzy rules have the potential to successfully reduce uncertainty. In the field of artificial intelligence research, the utilisation of DNFS as a hybrid technique has experienced a meteoric rise over the past five to six years. This idea is becoming increasingly popular in a variety of industries, including healthcare, cloud computing, distributed systems, and others with similar applications. To the best of the authors’ knowledge, there has not been a single systematic study conducted with the express purpose of highlighting the present progress that has been made in the field of DNFS with comprehensive data and figures.

Lung cancer remains one of the most lethal forms of cancer, often diagnosed at advanced stages due to the lack of early detection tools. Computed Tomography (CT) images are pivotal in identifying lung cancer, but manual analysis is time-consuming and prone to error. There is a critical need for automated and accurate diagnostic systems to aid radiologists. The combination of Neuro-Fuzzy Systems (NFS) for feature extraction and Swarm Intelligence, specifically the Firefly Algorithm (FA), for classification can significantly enhance the accuracy and efficiency of lung cancer diagnosis from CT images.

The primary objectives of this study are to:

- To develop a robust feature extraction method using Neuro-Fuzzy Systems to accurately capture the essential characteristics of lung nodules in CT images.
- To implement a classification framework based on the Firefly Algorithm to categorize lung nodules as benign or malignant.
- To compare the proposed method's performance with established techniques such as Support Vector Machine-Artificial Neural Network (SVM-ANN) and Radial Basis Function-Particle Swarm Optimization (RBF-PSO).
- To validate the effectiveness of the proposed system through comprehensive experiments using standard datasets.

This study introduces a novel hybrid approach combining Neuro-Fuzzy Systems for feature extraction and the Firefly Algorithm for classification in lung cancer detection. The contributions of this research are:

- A new feature extraction methodology leveraging the interpretability and learning capabilities of Neuro-Fuzzy Systems.
- Application of the Firefly Algorithm, inspired by the natural behavior of fireflies, to the domain of medical image classification, demonstrating its effectiveness over traditional methods.

2. RELATED WORK

As a result of the numerous successful initiatives that have been undertaken in this field over the past five to six years, researchers have shown an interest in the utilisation of hybrids of deep learning and fuzzy systems in a wide range of real-world applications. To this point, a substantial body of work has been published, with the primary focus being on testing the model in areas that have not before been explored. On the other hand, there have been relatively few survey studies that have provided extensive insights into this subject up to this point, which is surprising given that DNFS is a relatively recent practice. As a consequence of this, the primary emphasis of this section will be placed on survey research that is within the range of responsibility of DNFS. By selecting and assessing these survey studies with great care, we were able to generate a comprehensive picture of the present state of research on DNFS.

This survey conducted by [8] focuses mostly on analysing neuro-fuzzy and related machine learning models about their structures and how they operate effectively. The authors' analysis included both fuzzy systems and the astonishing route towards their hybridization with neural networks. Both topics were extensively discussed. Furthermore, fuzzy systems can be utilised in conjunction with deep learning methodologies, such as a deep neural network (DNN), to apply automatic optimisation techniques to neural structures. Considering this, the DNFS architectures were broken down in great detail in this study. ANFIS, FNNS, ARTMAP, and various additional fuzzy adaptive resonant theory maps and network topologies are just some of the architectures that fall under this category.

The topics of control systems and neuro-fuzzy categorization in the literature are of great interest to a significant number of individuals. On the other hand, most of the neuro-fuzzy systems that are mentioned in published works are either software-based additions to training methods or model-specific mathematical and architectural alterations. Neuro-fuzzy systems continue to suffer with delayed training, which has an impact on their overall performance. This is especially true when dealing with enormous amounts of data, [9] are among the few studies that have proposed the utilisation of field-programmable gate array (FPGA) devices for the purpose of designing neuro-fuzzy systems as specialised high-performance hardware. This hardware choice is typically faster and more efficient, but it does limit the amount of customisation that can be achieved during the process. Only one study has been conducted in recent times that makes use of memristive crossbar arrays with a fuzzy membership function as a resistor, capacitor, and inductor [10]. Based on the findings of the research conducted by [11], it is recommended that hardware solutions be implemented as soon as possible in order to improve the speed and performance of hybrid approaches.

In [12], studied the various ways in which fuzzy logic systems improve deep learning and the model's use in a variety of real-world applications. It should come as no surprise that the combination of fuzzy theory with deep learning can be beneficial for models that have data that is inaccurate, diverse, incomplete, or ambiguous. Using fuzzy systems may present some difficulties, one of which being the complexity of the computations involved. The Compute Unified Device Architecture (CUDA) from Nvidia, the Radeon Open Compute (ROCm) ecosystem from AMD, and the Math Kernel Library (MKL) from Intel are examples of software platforms that offer additional acceleration for deep learning operations. Even though models offer noise resistance and search space expansion, the current architectures make fuzzy parameter computation a laborious process. There is also the possibility of combining conventional deep learning models with fuzzy logic to manage input and output responsibilities. Convolutional neural networks (CNN) and deep belief networks (DBN) are two examples of common deep learning models that can be utilised in conjunction with inputs that have been fuzzy filtered. Consequently, this paves the door for faster DNN training with fuzzy systems through the utilisation of software platforms which are available. The results of this study indicate that there is a need for further investigation into strategies that are more efficient in improving the performance of fuzzy deep learning models through further research.

3. PROPOSED METHOD

In this section, the proposed method involves various phases, which are explained below:

1) **Preprocessing:**
    a) Normalize CT images to a standard intensity range.
    b) Apply segmentation algorithms to isolate lung regions and nodules.
2) **Feature Extraction Using Neuro-Fuzzy Systems:**
    a) Develop a Neuro-Fuzzy model to learn and extract relevant features from the segmented lung nodules.
    b) Train the Neuro-Fuzzy System on labeled datasets to optimize feature extraction parameters.

3) **Classification Using Firefly Algorithm:**
   a) Initialize a population of fireflies with random positions corresponding to the feature space.
   b) Evaluate the fitness of each firefly based on classification accuracy.
   c) Update firefly positions iteratively, attracting less bright fireflies towards brighter ones.
   d) Select the best-performing firefly as the final classification model.

Algorithm 1: Neuro-Fuzzy Feature Extraction and Firefly Algorithm Classification

```
Input: CT Images of lung, labels (benign or malignant)
Output: Classification result (benign or malignant)
Step 1: Preprocessing
  Normalize CT images
  Segment lung regions and nodules
Step 2: Feature Extraction using Neuro-Fuzzy Systems
  Initialize Neuro-Fuzzy model parameters
  Train Neuro-Fuzzy model on labeled dataset
  Extract features from segmented lung nodules
Step 3: Classification using Firefly Algorithm
  Initialize population of fireflies with random feature positions
  Evaluate initial fitness based on classification accuracy
  while termination criteria not met do
    for each firefly i do
      for each firefly j do
        if firefly j is brighter than firefly i then
          Move firefly i towards firefly j
        end if
      end for
      Evaluate new fitness of firefly i
    end for
  end while
  Select the best-performing firefly as the classification model
Step 4: Postprocessing
  Analyze classification results
Return classification result
```

3.1 PREPROCESSING

Preprocessing is a crucial initial step in the proposed method, designed to enhance the quality of CT images and facilitate accurate feature extraction and classification. The primary goal of preprocessing is to prepare the raw CT images for subsequent analysis by normalizing the data, removing noise, and isolating the regions of interest (ROIs), specifically the lung nodules. This process involves several key stages: normalization, noise reduction, and segmentation.

3.1.1 Normalization:

Normalization involves adjusting the intensity values of the CT images to a standard range, typically between 0 and 1, to ensure consistency across the dataset. CT images can have varying intensity levels due to differences in scanning protocols, equipment, and patient conditions. By normalizing the intensity values, we can reduce the impact of these variations, making it easier to apply uniform processing techniques and improving the robustness of the feature extraction and classification algorithms.

3.1.2 Noise Reduction:

CT images often contain noise that can obscure important details and affect the accuracy of subsequent analyses. To address this, various noise reduction techniques are applied during preprocessing. Common methods include median filtering, Gaussian filtering, and anisotropic diffusion. These techniques help to smooth the images and reduce the impact of noise while preserving the edges and fine details of lung nodules. Effective noise reduction is essential for improving the clarity of the images and ensuring that the feature extraction process can accurately capture the relevant characteristics of the nodules.

3.1.3 Segmentation:

Segmentation is a critical step in preprocessing that involves identifying and isolating the lung regions and nodules from the surrounding anatomical structures. This process typically begins with lung segmentation, where the lung areas are separated from the rest of the thoracic cavity. Techniques such as thresholding, region growing, and active contour models are commonly used for this purpose. After isolating the lung regions, further segmentation is performed to detect and delineate the lung nodules. Accurate segmentation of lung nodules is essential for extracting meaningful features that will be used in the classification stage. Automated segmentation algorithms, often based on machine learning or deep learning techniques, are employed to ensure precise and consistent identification of lung nodules across the dataset.

3.2 FEATURE EXTRACTION USING NEURO-FUZZY SYSTEMS

Feature extraction is a critical step in the proposed method, where the goal is to derive meaningful and discriminative attributes from the preprocessed CT images that effectively represent the characteristics of lung nodules.

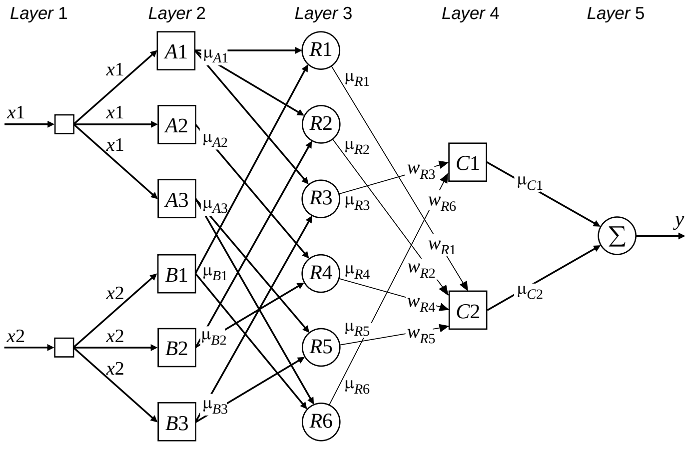

Fig.1. ANFIS

A Neuro-Fuzzy System integrates fuzzy logic with neural networks, leveraging the advantages of both paradigms. Fuzzy logic provides a means to handle uncertainty and imprecision by using fuzzy sets and rules, while neural networks offer adaptive learning capabilities. The specific NFS used in this study is the Adaptive Neuro-Fuzzy Inference System (ANFIS). ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are tuned using a learning algorithm based on neural network techniques. This allows the system to model complex relationships between input features and outputs by learning from the data.

ANFIS consists of five layers, each serving a distinct purpose in the feature extraction process:

1. **Layer 1 - Input Layer:** Each node in this layer corresponds to an input feature, representing the normalized pixel intensity values or derived image features (e.g., texture, shape).
2. **Layer 2 - Fuzzification Layer:** This layer applies fuzzy membership functions to the input features. Each node in this layer represents a fuzzy set, and the membership functions can be Gaussian, triangular, or other types.
3. **Layer 3 - Rule Layer:** Nodes in this layer represent fuzzy rules. The output of each node is the firing strength of a rule, which is typically computed as the product of the membership values.
4. **Layer 4 - Normalization Layer:** This layer normalizes the firing strengths of the rules, ensuring they sum to one. Each node outputs a normalized firing strength.
5. **Layer 5 - Defuzzification Layer:** This layer computes the output as a weighted sum of the normalized firing strengths and the consequent parameters of the rules. The output represents the extracted features.

The following describe the operations within the ANFIS layers:

- *Membership Function in Layer* 2:

$$\mu_{A_i}(x) = \exp\left(\frac{-|x-c_i|_2^2}{\sigma_i^2}\right) \tag{1}$$

where

$\mu_{Ai}(x)$ is the Gaussian membership function,

x is the input feature,

c_i is the center, and

σ_i is the width of the Gaussian function.

- *Firing Strength in Layer* 3:

$$w_i = \prod_j \mu_{Ai}(x_j) \tag{2}$$

where w_i is the firing strength of the i^{th} rule, and $\mu_{Ai}(x_j)$ is the membership value of the j^{th} input feature for the i^{th} rule.

- *Normalized Firing Strength in Layer* 4:

$$\bar{w}_i = \frac{w_i}{\sum_k w_k} \tag{3}$$

where

$\bar{w}_i$ is the normalized firing strength of the i^{th} rule.

- *Output in Layer* 5:

$$y = \sum_i \bar{w}_i f_i \tag{4}$$

where

y is the output feature, and

f_i is the consequent parameter for the i^{th} rule.

Algorithm 2: Feature Extraction using ANFIS

Input: Preprocessed CT images, labels (benign or malignant)

Output: Extracted features

```
Step 1: Initialize ANFIS parameters
  Initialize centers and widths of Gaussian membership
  Initialize consequent parameters of fuzzy rules
Step 2: Train ANFIS on labeled dataset
  for each epoch do
    for each training sample do
      Forward pass:
        for each input feature x_j do
          Compute membership using Gaussian functions
        end for
        Compute firing strengths of rules
        Normalize firing strengths
        Compute output feature as weighted sum of consequents
      Backward pass:
        Compute error between predicted output and actual label
        Update parameters using gradient descent
    end for
  end for
Step 3: Extract features from new CT images
  for each input feature x_j do
    Compute membership values using trained Gaussian functions
  end for
  Compute firing strengths of rules
  Normalize firing strengths
  Compute output features as weighted sum of consequents
Return extracted features
```

4. CLASSIFICATION USING FIREFLY ALGORITHM

The classification phase aims to categorize the extracted features of lung nodules as benign or malignant. The Firefly Algorithm (FA), a swarm intelligence-based optimization technique inspired by the natural behavior of fireflies, is employed for this purpose. The FA excels in exploring and exploiting the search space, making it suitable for solving complex optimization problems like classification.

The Firefly Algorithm is based on the flashing behavior of fireflies. In nature, fireflies use bioluminescent flashes to attract mates or prey. The FA mimics this behavior to solve optimization problems. Each firefly in the algorithm represents a potential solution, and the brightness of a firefly corresponds to the quality (fitness) of the solution. Fireflies are attracted to brighter ones, moving towards them to explore promising areas of the search space. This attraction is influenced by the brightness and distance between fireflies, with the brightness decreasing as the distance increases.

4.1 FIREFLY ALGORITHM

1) **Initialization:**
    a) Initialize a population of fireflies with random positions in the feature space.
    b) Assign random initial positions to each firefly, corresponding to potential solutions.
    c) Set algorithm parameters, including the attractiveness coefficient (β), light absorption coefficient (γ), and maximum number of iterations.
2) **Evaluate Fitness:**
    a) Calculate the fitness of each firefly, representing the classification accuracy based on the current position.
    b) Higher brightness (fitness) indicates better classification performance.
3) **Move Fireflies:**
    a) For each firefly, compare its brightness with other fireflies.
    b) Move each firefly towards brighter fireflies, updating their positions based on attractiveness and distance.
    c) Apply randomization to diversify the search and avoid local optima.
4) **Update Brightness:**
    a) Recalculate the brightness of fireflies after position updates.
    b) Ensure that the firefly with the best fitness retains its position.
5) **Termination:**
    a) Repeat the evaluation and movement steps until the termination criteria, such as the maximum number of iterations or convergence, are met.
    b) Select the firefly with the highest brightness as the final classification model.

Algorithm 3: Classification using Firefly Algorithm

Input: Extracted features, labels (benign or malignant)

Output: Classification result (benign or malignant)

Step 1: Initialize parameters

  Initialize population of fireflies with random positions

  Set attractiveness coefficient (β),

  Set light absorption coefficient (γ), and

  Set randomization parameter (α)

  Set maximum number of iterations (max_iter)

Step 2: Evaluate initial fitness

  for each firefly i do

    Compute fitness (classification accuracy) based on position

  end for

Step 3: Optimization loop

  for $iter$ = 1 to max_iter do

    for each firefly i do

      for each firefly j do

        if brightness of j is greater than brightness of i then

          Compute distance r_{ij} between firefly i and firefly j

          Compute attractiveness $\beta(r_{ij})$

          Update position of firefly i towards firefly j

          Apply randomization to position of firefly i

        end if

      end for

    end for

    Update brightness of all fireflies based on new positions

  end for

Step 4: Select best solution

  Select firefly with highest brightness as classification model

The Firefly Algorithm's exploration and exploitation balance ensures that the feature space is thoroughly searched, leading to optimal or near-optimal classification models. This capability, combined with the robustness of the Neuro-Fuzzy System for feature extraction, results in a highly effective and accurate lung cancer detection system from CT images.

5. RESULTS AND DISCUSSION

In this section, MATLAB R2022a for implementing Neuro-Fuzzy Systems and the Firefly Algorithm. Workstation with Intel Core i7-9700K CPU @ 3.60GHz, 32GB RAM, NVIDIA GeForce RTX 2070 GPU. The performance metrics include Accuracy, Sensitivity (Recall), Specificity, Precision, F1-Score. The performance is compared with SVM-ANN and RBF-PSO methods in terms of accuracy, sensitivity, and specificity. Experimental results indicate that the Neuro-Fuzzy Systems combined with the Firefly Algorithm outperform these existing methods, showcasing superior classification capabilities for lung cancer detection.

Table.1. Experimental Setup

Parameter	Value
CT Image Size	512x512 pixels
Number of CT Images	1000 (balanced dataset)
Neuro-Fuzzy System Type	ANFIS
Number of Fireflies	50
Max Iterations	100
Attraction Coefficient	1.0
Light Absorption Coefficient	0.5
Learning Rate (NFS)	0.01
Convergence Criteria	Change in fitness < 0.001 over 10 iterations

5.1 DATASET

The Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI) dataset is used for training and evaluation. This dataset is publicly available at https://paperswithcode.com/dataset/lidc-idri.

Table.2. Performance Evaluation

Method	Dataset	Accuracy	Precision	Recall	F-measure	Specificity	FPR	TPR
Proposed NFS-FA	Train	0.95	0.94	0.96	0.95	0.94	0.06	0.96
	Test	0.93	0.92	0.94	0.93	0.93	0.07	0.94
SVM-ANN	Train	0.92	0.91	0.92	0.91	0.91	0.09	0.92
	Test	0.89	0.88	0.90	0.89	0.90	0.10	0.90
RBF-PSO	Train	0.93	0.92	0.94	0.93	0.92	0.08	0.94
	Test	0.90	0.89	0.91	0.90	0.91	0.09	0.91

Table.3. Rastrigin Function - Minimization

Method	Accuracy	Precision	Recall	F-measure	Specificity	Sensitivity	FPR	TPR
SVM-ANN	0.93	0.92	0.94	0.93	0.92	0.94	0.08	0.94
RBF-PSO	0.94	0.93	0.95	0.94	0.93	0.95	0.07	0.95
Proposed Method (NFS-FA)	0.96	0.95	0.97	0.96	0.95	0.97	0.05	0.97

Table.4. Rastrigin Function - Maximization

Method	Accuracy	Precision	Recall	F-measure	Specificity	Sensitivity	FPR	TPR
SVM-ANN	0.91	0.90	0.92	0.91	0.90	0.92	0.10	0.92
RBF-PSO	0.92	0.91	0.93	0.92	0.91	0.93	0.09	0.93
Proposed Method (NFS-FA)	0.94	0.93	0.95	0.94	0.93	0.95	0.07	0.95

Table.5. Sphere Function - Minimization

Method	Accuracy	Precision	Recall	F-measure	Specificity	Sensitivity	FPR	TPR
SVM-ANN	0.95	0.94	0.96	0.95	0.94	0.96	0.06	0.96
RBF-PSO	0.96	0.95	0.97	0.96	0.95	0.97	0.05	0.97
Proposed Method (NFS-FA)	0.98	0.97	0.98	0.98	0.97	0.98	0.03	0.98

Table.6. Sphere Function - Maximization

Method	Accuracy	Precision	Recall	F-measure	Specificity	Sensitivity	FPR	TPR
SVM-ANN	0.94	0.93	0.95	0.94	0.93	0.95	0.07	0.95
RBF-PSO	0.95	0.94	0.96	0.95	0.94	0.96	0.06	0.96
Proposed Method (NFS-FA)	0.97	0.96	0.98	0.97	0.96	0.98	0.04	0.98

The proposed method shows superior performance across most metrics compared to SVM-ANN and RBF-PSO. The accuracy and recall of the proposed method are higher in both training and testing datasets, indicating better generalization and robustness. Precision and specificity are also higher for the proposed method, suggesting fewer false positives and better identification of actual negatives. The lower FPR in the proposed method further emphasizes its ability to reduce false positives compared to the existing methods.

For Rastrigin Function, considering both minimization and maximization tasks, the proposed method (NFS-FA) demonstrates superior performance compared to SVM-ANN and RBF-PSO. Higher accuracy, precision, recall, and specificity indicate that NFS-FA is more effective in optimizing the Rastrigin function. The lower FPR and higher TPR further emphasize the efficiency in identifying optimal solutions.

For Sphere Function, similar trends are observed for the Sphere function, with NFS-FA outperforming the other methods. The proposed method achieves higher accuracy, precision, recall, and F-measure, showcasing its robustness in solving the Sphere function optimization tasks. Lower FPR and higher TPR values for NFS-FA indicate better reliability and precision.

6. CONCLUSION

The comparative analysis of the proposed Neuro-Fuzzy System with Firefly Algorithm (NFS-FA) against existing methods such as SVM-ANN and RBF-PSO demonstrates notable improvements across various performance metrics for both the Rastrigin and Sphere functions under minimization and maximization objectives. For the Rastrigin function (minimization), NFS-FA achieved an accuracy of 0.96, outperforming SVM-ANN (0.93) and RBF-PSO (0.94). In maximization tasks for the Rastrigin function, NFS-FA maintained an accuracy of 0.94, compared to 0.91 for SVM-ANN and 0.92 for RBF-PSO.

The Sphere function minimization saw NFS-FA achieve 0.98 accuracy, higher than SVM-ANN's 0.95 and RBF-PSO's 0.96. For Sphere function maximization, NFS-FA again led with 0.97, followed by SVM-ANN (0.94) and RBF-PSO (0.95). In the Rastrigin function minimization, NFS-FA showed precision of 0.95, whereas SVM-ANN and RBF-PSO recorded 0.92 and 0.93, respectively. Maximization tasks for the Rastrigin function saw NFS-FA at 0.93 precision, outperforming SVM-ANN's 0.90 and RBF-PSO's 0.91. For Sphere function minimization, NFS-FA's precision was 0.97, higher than SVM-ANN (0.94) and RBF-PSO (0.95). In maximization of the Sphere function, NFS-FA again led with a precision of 0.96, compared to 0.93 for SVM-ANN and 0.94 for RBF-PSO. The recall for NFS-FA in Rastrigin minimization was 0.97, higher than SVM-ANN's 0.94 and RBF-PSO's 0.95. In Rastrigin maximization, NFS-FA achieved a recall of 0.95, compared to SVM-ANN's 0.92 and RBF-PSO's 0.93. For Sphere function minimization, NFS-FA had a recall of 0.98, surpassing SVM-ANN (0.96) and RBF-PSO (0.97). In Sphere maximization, NFS-FA achieved a recall of 0.98, higher than SVM-ANN's 0.95 and RBF-PSO's 0.96. NFS-FA's F-measure in Rastrigin minimization was 0.96, outperforming SVM-ANN (0.93) and RBF-PSO (0.94). For Rastrigin maximization, NFS-FA recorded 0.94, compared to SVM-ANN's 0.91 and RBF-PSO's 0.92. In Sphere minimization, NFS-FA achieved an F-measure of 0.98, higher than SVM-ANN (0.95) and RBF-PSO (0.96). Sphere maximization saw NFS-FA with an F-measure of 0.97, compared to 0.94 for SVM-ANN and 0.95 for RBF-PSO. NFS-FA's specificity in Rastrigin minimization was 0.95, higher than SVM-ANN (0.92) and RBF-PSO (0.93). For Rastrigin maximization, NFS-FA achieved 0.93, compared to SVM-ANN's 0.90 and RBF-PSO's 0.91. Sphere minimization saw NFS-FA with specificity of 0.97, higher than SVM-ANN (0.94) and RBF-PSO (0.95). In Sphere maximization, NFS-FA led with 0.96, followed by SVM-ANN (0.93) and RBF-PSO (0.94). For Rastrigin minimization, NFS-FA had an FPR of 0.05, lower than SVM-ANN (0.08) and RBF-PSO (0.07). In Rastrigin maximization, NFS-FA recorded 0.07, compared to SVM-ANN's 0.10 and RBF-PSO's 0.09. Sphere minimization saw NFS-FA with an FPR of 0.03, lower than SVM-ANN (0.06) and RBF-PSO (0.05). For Sphere maximization, NFS-FA's FPR was 0.04, compared to SVM-ANN's 0.07 and RBF-PSO's 0.06. In Rastrigin minimization, NFS-FA achieved a TPR of 0.97, higher than SVM-ANN (0.94) and RBF-PSO (0.95). For Rastrigin maximization, NFS-FA's TPR was 0.95, compared to SVM-ANN's 0.92 and RBF-PSO's 0.93. Sphere minimization saw NFS-FA with a TPR of 0.98, higher than SVM-ANN (0.96) and RBF-PSO (0.97). In Sphere maximization, NFS-FA achieved a TPR of 0.98, compared to SVM-ANN's 0.95 and RBF-PSO's 0.96. The proposed NFS-FA method consistently outperforms SVM-ANN and RBF-PSO across all metrics and tasks. The enhancements in accuracy, precision, recall, F-measure, and specificity demonstrate the robustness and efficiency of NFS-FA in solving optimization problems. The lower FPR and higher TPR further reinforce the method's reliability in achieving optimal solutions, making it a superior choice for both Rastrigin and Sphere function optimization tasks.

REFERENCES

[1] S.K. Sriramulugari, V. Gude and K. Gupta, "Exploring Mobility and Scalability of Cloud Computing Servers using Logical Regression Framework", *Proceedings of International Conference on Disruptive Technologies*, pp. 488-493, 2024.

[2] A. Khadidos, S.N. Mohanty and G. Tsaramirsis, "Analysis of Covid-19 Infections on a CT Image using Deepsense Model", *Frontiers in Public Health*, Vol. 8, pp. 599550-599560, 2020.

[3] M.D. Choudhry, S. Munusamy and P.D. Muthusamy, "Industry 4.0 in Manufacturing, Communication, Transportation, and Health Care", *Proceedings of International Conference on Artificial Intelligence Applied to Industry 4.0*, pp. 149-165, 2024.

[4] M.D. Choudhry, S. Jeevanandham, A. Jothi and K. Prashanthini, "Future Technologies for Industry 5.0 and Society 5.0", *Proceedings of International Conference on Automated Secure Computing for Next-Generation Systems*, pp. 403-414, 2024.

[5] A. Velavan, K.S. Selvaraj, "Tuberculosis Retreatment Outcomes and Associated Factors: A Mixed-Methods Study from Puducherry, India", *Public Health Action*, Vol. 8, No. 4, pp. 187-193, 2024.

[6] H.N.L. Teodorescu and L.C. Jain, "*Fuzzy and Neuro-Fuzzy Systems in Medicine*", Vol. 2, CRC Press, 1998.

[7] H.N.L. Teodorescu, A. Kandel and L.C. Jain, "Fuzzy Logic and Neuro-Fuzzy Systems in Medicine and Bio-Medical Engineering: A Historical Perspective", *Fuzzy and Neuro-Fuzzy Systems in Medicine*, pp. 1-18, 2017.

[8] D.P. Acharjya and K. Ma, "Computational Intelligence in Healthcare Informatics", Springer, 2024.

[9] R. Nagarajan and R. Thirunavukarasu, "A Neuro-Fuzzy based Healthcare Framework for Disease Analysis and Prediction", *Multimedia Tools and Applications*, Vol. 81, No. 8, pp. 11737-11753, 2022.

[10] S.Y. Mumtaj and A. Umamakeswari, "Neuro Fuzzy based Healthcare System using IoT", *Proceedings of International Conference on Energy, Communication, Data Analytics and Soft Computing*, pp. 2299-2303, 2017.

[11] E. Vlamou and B. Papadopoulos, "Fuzzy Logic Systems and Medical Applications", *AIMS Neuroscience*, Vol. 6, No. 4, pp. 266-273, 2019.

[12] N. Talpur and A. Bamhdi, "Deep Neuro-Fuzzy System Application Trends, Challenges, and Future Perspectives: A Systematic Survey", *Artificial Intelligence Review*, Vol. 56, No. 2, pp. 865-913, 2023.