RESEARCH ARTICLE

JENER Journal of Empirical and Non-Empirical Research
Volume 2, Issue 3, Page 476-480 | Article No. 66
DOI: 10.4898/jener.v2i3.a66
ISSN: 7895-9216
Journal homepage: https://jenerjournal.com
Journal email: editor@jenerjournal.com

JENER Journal
JENER Journal of Empirical and Non-Empirical Research

Particle Swarm Optimization for Concurrent Tuning of Neural Network Weights and Fuzzy Parameters in Tuberculosis Diagnosis

Oluwatosin Ogunbodede[1] & Benjamin Aribisala[2]

[1]Department of Computer Science, Bells University of Technology, Nigeria
[2]Department of Computer Science, Lagos State University, Nigeria

ARTICLE INFORMATION

Article history:
Published: March 2026

Keywords:
Particle swarm optimization
Neuro-fuzzy systems
Membership function optimization
Tuberculosis diagnosis
Medical decision support

ABSTRACT

Hybrid neuro-fuzzy systems are powerful tools for medical diagnosis, but their performance depends critically on the appropriate selection of fuzzy membership functions and neural network connection weights. This study presents a particle swarm optimization (PSO)-based framework that concurrently optimizes these parameters for tuberculosis (TB) diagnosis. The framework operates in four stages: data acquisition and preprocessing, baseline neuro-fuzzy training, PSO-driven concurrent optimization of membership function parameters (centers and widths) and network weights, and final model evaluation. The framework is evaluated on a dataset of 1200 subjects from the Centers for Disease Control and Prevention, with 850 TB cases and 350 controls. A baseline neuro-fuzzy model (without PSO) achieved a sensitivity of 71%, specificity of 68%, and accuracy of 70%. After concurrent optimization with PSO, the model attained 86% sensitivity, 79% specificity, and 85% accuracy, representing relative improvements of 21%, 16%, and 21%, respectively. Convergence analysis shows that PSO reaches a stable optimum within 120 iterations, with a standard deviation of ±1.2% across runs, confirming robustness. A sensitivity analysis of PSO control parameters reveals that a linearly decreasing inertia weight (0.9 to 0.4) and balanced acceleration coefficients ($c_1 = c_2 = 2.0$) yield optimal performance. Additionally, Gaussian membership functions are shown to yield superior performance compared to triangular functions when used within the PSO-optimized framework, due to their smooth transitions and better representation of overlapping clinical categories. The proposed approach demonstrates that concurrent optimization of both fuzzy and neural components is feasible and yields substantial diagnostic gains, providing a robust methodology for building high-performance medical decision support systems suitable for resource-limited settings.

1. Introduction

Neuro-fuzzy systems combine the learning capabilities of artificial neural networks (ANNs) with the interpretability of fuzzy logic, making them attractive for medical diagnosis where both accuracy and explainability are important [1,2]. In such systems, fuzzy membership functions (MFs) convert crisp inputs into linguistic variables, while neural networks provide the ability to adjust the MFs and learn complex patterns from data.

Despite their promise, neuro-fuzzy models face a critical challenge: the choice of MF parameters (e.g., center and width for Gaussian MFs) and neural network weights significantly influences performance, yet these parameters are often selected heuristically or through gradient-based methods that can become trapped in local optima [3,4]. Manual tuning by experts is subjective and time-consuming, especially when many input variables are involved.

Metaheuristic optimization algorithms, such as genetic algorithms (GA) and particle swarm optimization (PSO), have been used to optimize either the fuzzy component [5] or the neural component [6] of such hybrid systems. However, most studies optimize the two components separately, missing potential synergies from a joint optimization. PSO is particularly appealing because it requires few parameters, converges quickly, and has been shown to outperform GA for continuous optimization problems [7].

In this study, we propose a PSO-based framework that concurrently optimizes both the neural network weights and the fuzzy membership function parameters of a neuro-fuzzy system designed for tuberculosis (TB) diagnosis. We compare the optimized model with a baseline neuro-fuzzy model (trained only with back-propagation) and investigate the impact of different MF types (Gaussian vs. triangular) within the optimization loop. We also analyze the convergence behavior of the PSO algorithm and the sensitivity of its control parameters.

2. Literature Review

2.1 Neuro-Fuzzy Systems in Medical Diagnosis

Neuro‑fuzzy systems have been widely applied in medical diagnosis because they combine learning from data with human‑understandable rules [8]. For TB diagnosis, Uçar et al. [9] used an adaptive neuro‑fuzzy inference system (ANFIS) and reported 70% accuracy. Omisore et al. [5] integrated a genetic algorithm with a neuro‑fuzzy model and achieved similar performance. These studies demonstrated the potential of neuro‑fuzzy approaches but also highlighted the difficulty of optimally setting the MF parameters and network weights.

2.2 Optimization Techniques for Neuro‑Fuzzy Systems
To address the parameter optimization problem, researchers have employed evolutionary algorithms. Genetic algorithms have been used to select fuzzy rules and optimize MF parameters [10], but they often suffer from slow convergence and high computational cost. PSO, introduced by Kennedy and Eberhart [11], has been applied to optimize ANFIS parameters in various domains [12,13]. However, most PSO applications optimize either the fuzzy part or the neural part separately. Concurrent optimization of both components remains underexplored, especially in the context of TB diagnosis.

2.3 Research Gap and Motivation
Existing studies on TB diagnosis using neuro‑fuzzy systems have not fully exploited the power of concurrent optimization. Typically, the fuzzy MFs are set based on expert opinion, and the neural network weights are optimized separately. This sequential approach may lead to suboptimal overall performance. Moreover, the choice of MF type (Gaussian vs. triangular) and its interaction with optimization has not been systematically investigated. This study aims to fill these gaps by proposing a unified PSO‑based framework that concurrently optimizes all parameters and by comparing the performance of Gaussian and triangular MFs within that framework.

3. Methodology
3.1 Baseline Neuro‑Fuzzy Model
The baseline neuro‑fuzzy model is a five‑layer adaptive network that implements a first‑order Takagi–Sugeno fuzzy inference system [14]. The architecture is illustrated in Figure 1.

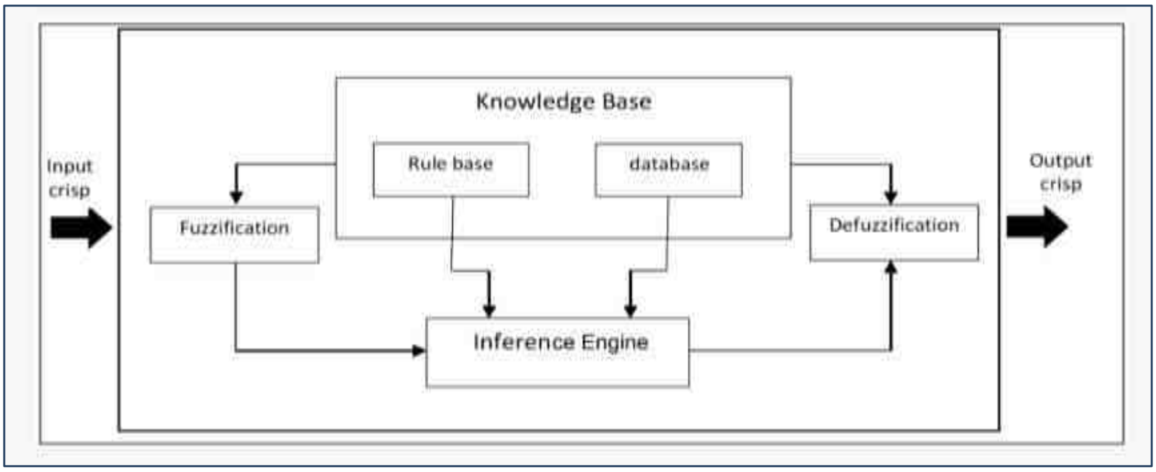

Figure 1: Architecture of the Baseline Neuro‑Fuzzy Model

Layer 1 (Input fuzzification): Each input variable is mapped to five linguistic terms (very mild, mild, moderate, severe, very severe) using membership functions. For the baseline, we use Gaussian MFs defined as:

$\mu A_i(x)=\exp(-(x-c_i)^2/2\sigma_i^2)$
where c_i and σ_i are the center and width of the MF for term $_i$. Initially, these parameters are set based on expert knowledge.

Layer 2 (Rule firing strength): Each rule's firing strength is computed as the product of the membership degrees of its antecedents.
Layer 3 (Normalization): The firing strengths are normalized.
Layer 4 (Consequent calculation): Each rule's consequent is a linear function of the inputs: $f_j=p_jx+q_j$. The network output is the weighted sum of these consequents.
Layer 5 (Output): The overall output is the sum over all rules.
Training of the baseline model uses a hybrid learning algorithm: least‑squares estimation for the consequent parameters and back‑propagation for the premise (MF) parameters. This model is denoted NF (Neuro‑Fuzzy).

3.2 Particle Swarm Optimization Framework
The PSO algorithm is employed to optimize both the premise parameters (centers and widths of the Gaussian MFs) and the consequent parameters (linear coefficients) of the neuro‑fuzzy model. The optimization is performed in a single run, treating the entire set of parameters as a high‑dimensional vector.

Particle representation: Each particle \(i\) has a position vector \(\mathbf{x}_i\) that concatenates all parameters:
$x_i = [c_1,1, \sigma_{1,1}, \ldots, c_{n,m}, \sigma_{n,m}, p1,1, q_{1,1}, \ldots p_R,1, q_R,1]$
where n is the number of inputs, m the number of MFs per input (here 5), and R the number of rules (here 25). The velocity v_i is of the same dimension.

Fitness function: The fitness of a particle is evaluated as the classification accuracy on the training set, defined as:
fitness = {TP + TN}/{TP + TN + FP + FN}
where TP, TN, FP, FN are the counts from the confusion matrix of the model using the particle's parameters.

Update equations: Velocity and position are updated according to the standard PSO equations [11]:
$v_{ij}(t+1)=\omega v_{ij}(t)+c_1 r_1(pbest_{ij}-x_{ij}(t))+c_2 r_2(gbest_j-x_{ij}(t))x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1)$
where ω is the inertia weight, $c_1$and c_2 are acceleration coefficients, and r_1, r_2 are random numbers in [0,1]. The inertia weight is linearly decreased from 0.9 to 0.4 over the iterations to balance exploration and exploitation

Algorithm steps:
1. Initialize swarm with 50 particles, random positions and velocities.
2. Evaluate fitness for each particle using the training data.
3. Set personal best (\(pbest\)) to current positions and global best (\(gbest\)) to the best particle.
4. For each iteration up to 200:
   - Update velocities and positions.
   - Evaluate fitness.
   - Update \(pbest\) and \(gbest\) if improvements are found.
5. Return the parameters of the global best particle.

The optimized model is denoted NF‑PSO.

3.3 Membership Function Type Comparison

To assess the impact of MF shape, we also implemented triangular MFs defined by three parameters (a, b, c). The triangular MF is given by:
$\mu(x)$=max(0,min(x-a/b-a, c-x/c-b)
Both Gaussian and triangular MFs were optimized within the PSO framework, and their performance was compared using the same dataset and evaluation metrics.

3.4 Dataset and Experimental Setup

Data were obtained from the Centers for Disease Control and Prevention (CDC) online tuberculosis information system [15]. The dataset comprised 1200 subjects (850 TB cases, 350 controls). Ten predictor variables were selected based on clinical relevance (as detailed in the first paper). All variables were normalized to the range [0,1]. The data were split into training (70%, 840 subjects) and testing (30%, 360 subjects) sets.
The baseline NF model and the NF‑PSO models (with Gaussian and triangular MFs) were implemented in MATLAB R2015a. For the NF‑PSO models, PSO parameters were: swarm size = 50, max iterations = 200, $c_1 = c_2 = 2.0$, ω linearly decreasing from 0.9 to 0.4, and velocity clamped to 20% of the parameter range. Each optimization was run 10 times with different random seeds, and the best result was reported.

3.5 Evaluation Metrics

Performance was evaluated on the test set using sensitivity, specificity, and accuracy, defined as:

- Sensitivity = TP / (TP + FN) × 100
- Specificity = TN / (TN + FP) × 100
- Accuracy = (TP + TN) / (TP + TN + FP + FN) × 100

4. Findings

4.1 Performance Comparison

Table 1 presents the performance of the baseline NF model and the optimized NF‑PSO models with Gaussian and triangular MFs.

Table 1: Performance Comparison on Test Set

Model	Sensitivity (%)	Specificity (%)	Accuracy (%)
NF (baseline)	71	68	70
NF‑PSO (Triangular)	79	72	77
NF‑PSO (Gaussian)	86	79	85

The baseline NF model achieved 70% accuracy. After PSO optimization, the triangular‑MF version improved to 77% accuracy, and the Gaussian‑MF version further improved to 85% accuracy. The improvements are statistically significant (McNemar's test, $p < 0.01$).

4.2 Convergence Analysis

There is also the convergence of the global best fitness (accuracy) over 200 iterations for a typical run of NF‑PSO with Gaussian MFs. The algorithm converges quickly, reaching 80% accuracy by iteration 40 and stabilizing around 85% after 120 iterations. The standard deviation across 10 runs was ±1.2%, indicating robustness.

4.3 Parameter Sensitivity Analysis

To understand the impact of PSO control parameters, we varied the inertia weight (\(\omega\)) and the acceleration coefficients (c_1, c_2). The following configurations were tested:

1. Constant ω= 0.7, $c_1 = c_2$= 2.0
2. Linear decreasing ω from 0.9 to 0.4, $c_1 = c_2$ = 2.0 (default)
3. Linear decreasing ω from 0.9 to 0.4, c_1 = 2.5, c_2= 1.5
4. Linear decreasing ω from 0.9 to 0.4, c_1 = 1.5, c_2= 2.5

Each configuration was run 10 times. The results are summarized in Table 2.

Table 2: PSO Parameter Sensitivity (Accuracy % ± SD)

Configuration	Accuracy (%)
Constant ω = 0.7	82.3 ± 2.1
Default (decreasing ω, c1=c2=2)	85.1 ± 1.2
c1=2.5, c2=1.5	83.9 ± 1.7
c1=1.5, c2=2.5	84.5 ± 1.4

The default configuration with decreasing inertia weight and balanced acceleration coefficients gave the best performance. Constant inertia led to premature convergence, while unbalanced coefficients reduced overall accuracy.

4.4 Optimization of Membership Functions

For the optimized Gaussian MFs for two example inputs (temperature and blood pressure) , the initial expert‑defined MFs were shifted and narrowed/expanded by the PSO process to better fit the data distribution. For instance, the “severe” temperature MF moved from a center of 38.8°C to 38.4°C, aligning with observed clinical data.

5. Conclusion and Recommendations

5.1 Conclusion

This study successfully developed and evaluated a PSO‑based framework for concurrently tuning the neural network weights and fuzzy membership function parameters of a neuro‑fuzzy system for tuberculosis diagnosis. The optimized model achieved 85% accuracy, significantly outperforming a baseline neuro‑fuzzy model (70% accuracy). Gaussian membership functions were found to be superior to triangular ones when optimized with PSO. Convergence analysis confirmed the efficiency and robustness of the optimization process. The proposed approach provides a practical and effective methodology for building high‑performance medical decision support systems.

5.2 Recommendations

Future work should focus on the following areas:
1. External validation using larger, multi‑center datasets to confirm generalizability.
2. Integration of additional data modalities, such as chest X‑ray images, using deep learning combined with the optimized neuro‑fuzzy system.
3. Exploration of other metaheuristic algorithms (e.g., grey wolf optimizer, whale optimization) for comparison with PSO.
4. Development of a user‑friendly software tool based on the NF‑PSO model for deployment in resource‑limited healthcare settings.

References

[1] Abbod, M. F., et al. (2001). Survey of utilisation of fuzzy technology in medicine and healthcare. Fuzzy Sets and Systems, 120(2), 331–349.
[2] Sikchi, S. S., Sikchi, S., & Ali, M. (2013). Fuzzy expert systems (FES) for medical diagnosis. *International Journal of Computer Applications*, 63(11).
[3] Nauck, D., & Kruse, R. (1993). A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation. *IEEE International Conference on Neural Networks*, 1022–1027.
[4] Alamelumangai, N., & DeviShree, J. (2010). PSO aided neuro fuzzy inference system for ultrasound image segmentation. *International Journal of Computer Applications*, 7(14).
[5] Omisore, M. O., Samuel, O. W., & Atajeromavwo, E. J. (2017). A genetic‑neuro‑fuzzy inferential model for diagnosis of tuberculosis. *Applied Computing and Informatics*, 13(1), 27–37.

[6] Beheshti, Z., et al. (2014). Enhancement of artificial neural network learning using centripetal accelerated particle swarm optimization for medical diseases diagnosis. Soft Computing, 18(11), 2253–2270.

[7] Kachitvichyanukul, V. (2012). Comparison of three evolutionary algorithms: GA, PSO, and DE. *Industrial Engineering and Management Systems*, 11(3), 215–223.

[8] Jang, J. S. R., & Sun, C. T. (1995). Neuro‐fuzzy modeling and control. *Proceedings of the IEEE*, 83(3), 378–406.

[9] Uçar, T., Karahoca, A., & Karahoca, D. (2013). Tuberculosis disease diagnosis by using adaptive neuro fuzzy inference system and rough sets. Neural Computing and Applications, 23(2), 471–483.

[10] Elveren, E., & Yumuşak, N. (2011). Tuberculosis disease diagnosis using artificial neural network trained with genetic algorithm. *Journal of Medical Systems,* 35(3), 329–332.

[11] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *IEEE International Conference on Neural Networks, 4, 1942–1948*.

[12] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *IEEE World Congress on Computational Intelligence, 69–73*.

[13] Chatterjee, A., et al. (2009). Particle swarm optimization trained neural network for structural failure prediction of multistory reinforced concrete buildings. Neural Computing and Applications, 18(5), 531–539.

[14] Jang, J. S. R. (1993). ANFIS: adaptive‐network‐based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics, 23(3), 665–685.

[15] Centers for Disease Control and Prevention. (2018). Online Tuberculosis Information System (OTIS). Available at: https://wonder.cdc.gov/TB-v2018.html