



Electric Fish Optimizer-Based Hyperparameter Tuning of MLP for Fault Diagnosis in Grid-Connected Photovoltaic Systems

Muhammed A. ADAWUDI^{1*}, Danlami MALIKI², Ibrahim M. ABDULLAHI³

^{1*}Department of Electrical/Electronic Engineering, Kogi State Polytechnic, Lokoja, Kogi State, Nigeria

^{2,3}Department of Computer Engineering, Federal University of Technology, Minna, Nigeria

^{1*}m.adawudi@gmail.com, ²danlami.maliki@futminna.edu.ng, ³imabdullahi@futminna.edu.ng

Abstract

This paper presents an optimized multilayer perceptron (MLP) model for fault diagnosis in grid-connected photovoltaic (PV) systems using the Electric Fish Optimizer (EFO). A real-world dataset comprising five fault classes was preprocessed and used to train a baseline MLP model. The EFO algorithm was employed to optimize key hyperparameters of the network, and a systematic random search was further conducted to determine optimal EFO configurations. The optimized model achieved an accuracy of 99.06%, improving upon the baseline accuracy of 98.63%, with enhanced per-class F1-scores across all fault categories. The results demonstrate that EFO is an effective tool for improving neural network performance in PV fault diagnosis, particularly when applied to large and imbalanced datasets.

Keywords: *Electric Fish Optimizer, Fault diagnosis, Metaheuristic optimization, Multilayer perceptron, Photovoltaic systems.*

1.0 Introduction

The total available installed capacity of grid-connected solar power globally is estimated at 1,177 GW by the end of 2022, having increased massively since 2000 [1]. Available projections also indicate that grid-connected solar systems are expected to generate significant power worldwide by 2040 [2]. Grid-connected photovoltaic systems play a significant role in modern power systems as an alternative source of clean and sustainable energy in residential and office buildings. However, most PV systems are vulnerable to various faults such as short circuits, open circuits, partial shading, degradation, and inverter-related faults [3]. Effective fault detection and diagnosis (FDD) techniques are therefore highly essential for ensuring the reliability, safety, and optimal operation of grid-connected PV systems.

1.1 Background

The approaches applied in traditional fault detection techniques often rely on fixed thresholds, analytical models, or hardware-based protection schemes and they often lack robustness under dynamic environmental conditions such as fluctuating irradiance and temperature [4] [5].

Consequently, recent data-driven techniques especially artificial neural networks - are in vogue due to their capability to learn the complex nonlinear relationships between measured system variables and fault conditions [6].

Multilayer perceptrons (MLPs) are widely adopted due to their simplicity and ability to model non-linear relationships [7]. Although, the performance of MLPs is dependent on the choice of hyperparameters such as the number of hidden layers, neurons per layer, dropout rate, and learning rate; manual tuning could be time-consuming and may not generate optimal configurations, especially when dealing with large, imbalanced datasets typical of real-world PV systems [8], [9].

1.2 Problem

Unresolved electrical faults in photovoltaic systems lead to performance degradation, safety hazards, and equipment damage which could be expensive to remedy [3], [10]. Manual tuning is often inadequate to exploit the full capacity of MLP models due to its inherent inefficiency when dealing with large and unbalanced datasets [2], [9].

1.3 Related Work

The Electric Fish Optimization algorithm was introduced by [11] as a novel metaheuristic inspired by the electrolocation behaviour of electric fish. Their foundational work compared EFO against established algorithms including Simulated Annealing, Vortex Search, Genetic Algorithms, Differential Evolution, Particle Swarm Optimization, and Artificial Bee Colony across 80 mathematical benchmark functions and several real-world

constrained design problems. The results demonstrated that EFO outperformed or was highly competitive with all comparators, establishing its potential as a powerful optimization tool. The study acknowledged that interaction effects between EFO's parameters were not explicitly investigated, leaving room for further refinement of the algorithm's meta-parameters.

To address one of EFO's limitations [12] proposed an Adaptive EFO based on Standstill Label and Levy Flight (SLLF-EFO). The modified algorithm successfully mitigated standard EFO's tendency to become trapped in local optima, improving both search speed and accuracy on benchmark functions. While this theoretical advancement strengthened the algorithm's mathematical foundations, the research was confined to benchmark function testing and lacked application to real-world fault diagnosis challenges or neural network training contexts.

The Electric Fish-Based Arithmetic Optimization Algorithm was created by [13] through merging EFO with Arithmetic Optimization. This hybrid approach was specifically designed to improve exploration capabilities for feature selection problems. Although the algorithm demonstrated enhanced performance, its application remained general rather than being tailored to Multilayer Perceptrons or PV fault diagnosis specifically.

In another application of the EFO algorithm, [14] combined the EFAOA algorithm with a discrete convolution-based vision transformer to automate hyperparameter tuning for deep neural network classifiers. The approach successfully tuned model parameters based on validation dataset performance. However, the optimization target was the Vision Transformer architecture rather than a Multilayer Perceptron, leaving the application of EFO variants to MLP optimization unexplored.

To solve power system optimization problems, [15] applied the standard EFO algorithm to solve the Economic Load Dispatch problem in power systems. Their implementation achieved lower-cost generation scheduling with favourable convergence characteristics. While demonstrating EFO's utility in power system optimization, this application focused entirely on operational scheduling rather than fault diagnosis or neural network training. Notably, the authors explicitly recommended that future studies should implement EFO on neural network applications, a gap directly addressed by this paper.

A review study done by [7] compared classical physical models against metaheuristic algorithms (PSO, GA, ACO) and hybrid machine learning and deep learning models for PV fault diagnosis. Their analysis revealed that hybrid methodologies combining metaheuristics for tuning with ML/DL classifiers dominate the field, consistently achieving classification accuracy exceeding 95% with superior robustness compared to standalone approaches. However, the review identified significant barriers to practical deployment including stochastic variability, data scarcity, lack of model interpretability, and high computational costs, particularly for resource-constrained platforms. Notably, the study did not consider EFO optimization.

1.4 Gap

While optimization has been proven to be effective in overcoming the limitations of neural networks when applied in PV fault diagnosis; evidenced by the application of various metaheuristic algorithms in enhancing neural networks [7] [16], there is no reviewed study which has applied the Electric Fish Optimizer (EFO) specifically to train or tune a Multilayer Perceptron for photovoltaic fault diagnosis despite its exceptional exploration and exploitation capabilities. Furthermore, existing optimization approaches often rely on dated metaheuristics, lack rigorous statistical validation, and are seldom validated on authentic operational data [17], [18], [19]. EFO has demonstrated rapid convergence, the ability to jump out of local optima, and high stability [12] and has proven to be highly effective in tuning deep neural networks for complex classification tasks and solving power system scheduling problems. [15]

1.5 Contributions

The main contributions of this paper are:

- (i) Application of the Electric Fish Optimizer for hyperparameter tuning of an MLP in PV fault diagnosis,
- (ii) Evaluation of its performance on a real-world PV dataset with five fault classes,
- (iii) Systematic optimization of EFO parameters, and
- (iv) Validation of performance improvements over baseline models.

2.0 Materials and Methods

This paper employs a metaheuristic algorithm, the Electric Fish Optimizer (EFO) to tune the hyperparameters of the MLP and enhance its classification accuracy and robustness. Its performance is then compared against the baseline MLP. The details of the dataset used, preprocessing steps, MLP architecture, the EFO algorithm, the experimental setup, and the metrics used in assessing the performance of the classifier are presented.

2.1 Materials

(i) Software: Cloud-Based Model Development & Analysis: Kaggle notebooks with dual Tesla T4 GPUs (16 GB memory each). The software environment included TensorFlow 2.19, Python 3.12, and standard libraries (pandas, numpy, matplotlib, seaborn, scikit-learn).

(ii) Hardware: Local Simulation Machine: Intel Pentium Processor (@ 2.90 GHz), 4 GB RAM.

2.2 Overall Framework

The methodological framework involves optimizing a baseline MLP model for PV fault diagnosis using Electric Fish Optimization, as illustrated in Figures 1 and 2. The key steps are:

1. Acquisition, preprocessing and normalization of the dataset.
2. Design and testing of a baseline MLP neural network for classification.
3. Training and testing of the MLP on the full dataset with all metrics computed and stored.
4. Optimization of the MLP hyperparameters using EFO to improve performance.
5. The optimized MLP is trained and tested on the full dataset.

To investigate the optimal EFO parameters for MLP optimization the keys steps are:

6. Defining the search space containing 400 possible configurations of EFO parameters.
7. Evaluation of 25 random configurations to find the best EFO configuration based on the highest fitness (accuracy) attained.
8. From the best EFO configuration, extract the optimal MLP hyperparameters.
9. The optimized MLP is trained on the full training set and evaluated on the held-out test set and performance is also recorded.
10. Comprehensive performance evaluation and analysis.

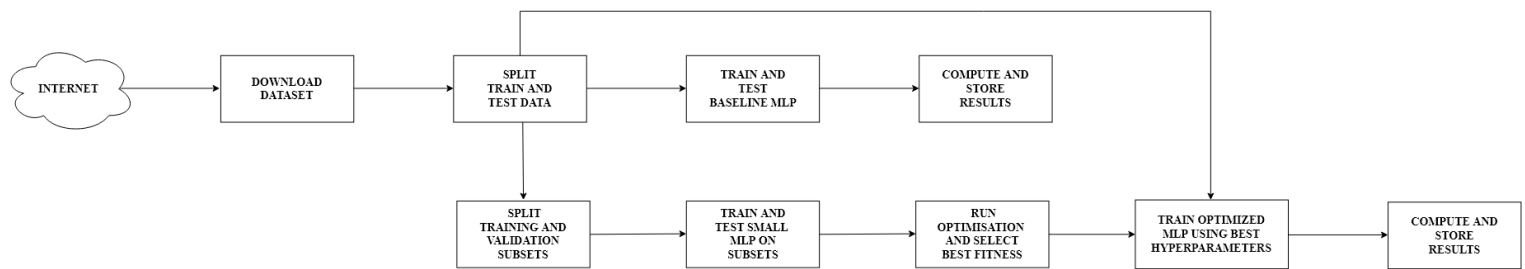


Figure 1: Workflow for Baseline MLP Optimization

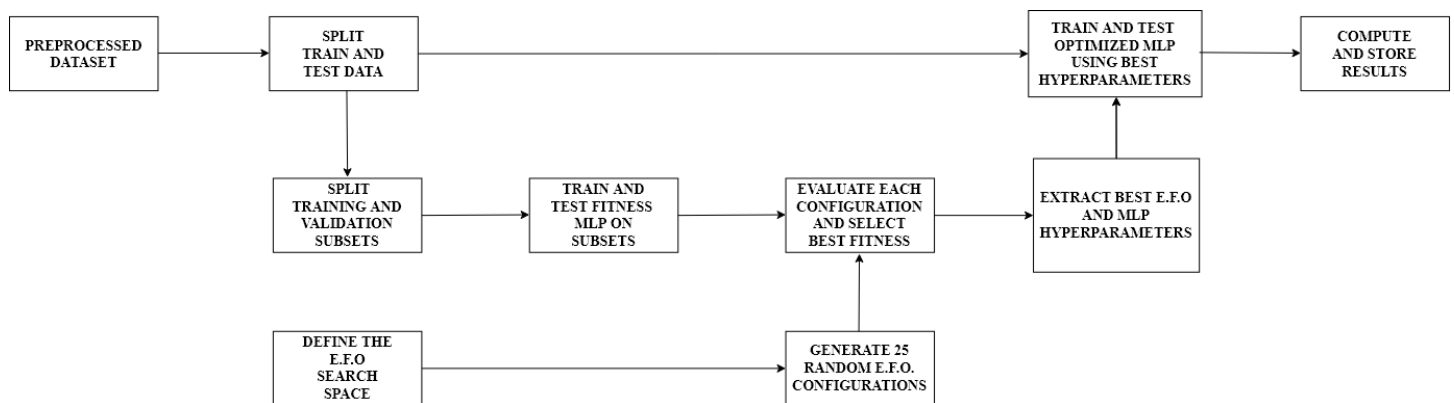


Figure 2: Workflow for EFO Optimization using random search

2.3 Dataset

The dataset used was sourced from from GitHub via https://github.com/clayton-h-costa/pv_fault_dataset and was from a study conducted by [20].

This dataset contains 16 days of operational data from a 5.28 kWp grid-tied PV plant, comprising both normal and faulty conditions. The dataset contains over 1 million data points with electrical data (String voltages (vdc1, vdc2) and currents (idc1, idc2) and ambient and label data (Irradiance (irr), module temperature (pvt), and fault labels (f_nv). The fault classes are labeled as “0” (Normal), “1” (Short-Circuit), “2” (Degradation), “3” (Open-Circuit), “4” (Shadowing).

The dataset was preprocessed using the following steps:

- (i) Conversion and integration: The dataset files were converted from .mat to .csv format and multiple files were integrated into a single file.
- (ii) Feature selection: The relevant features (voltage, current, power, irradiance, temperature) needed for training the MLP were selected to make a feature vector.
- (iii) Min-max normalization: The numerical values of the features in the dataset were scaled between 0 and 1 using min-max normalization technique.
- (iv) Stratified data splitting: The dataset was split into training and testing subsets along the ratio of 70% for training and 30% for testing. The training subset was further split using an 80/20 ratio to create training and validation sets to be used during optimization.

2.4 Baseline MLP Model

A baseline MLP was designed with two hidden layers, each containing 128 neurons, ReLU activation, batch normalisation, and dropout (rate = 0.3). The output layer uses softmax activation to produce class probabilities. The model was compiled with the adam optimizer (learning rate = 0.001) and sparse categorical cross-entropy loss. Training was performed for up to 150 epochs with early stopping (patience = 10) based on validation accuracy.

2.5 Electric Fish Optimizer (EFO)

The Electric Fish Optimizer is a metaheuristic algorithm inspired by the electrolocation behaviour of electric fish. It balances exploration (active electrolocation) and exploitation (passive electrolocation) to search for optimal solutions. EFO is used to optimize the following MLP hyperparameters:

- Number of neurons in the first hidden layer (n1 in [32, 256])
- Number of neurons in the second hidden layer (n2 in [16, 128])
- Dropout rate (d in [0.0, 0.5])
- Learning rate (η in [10⁻⁴, 10⁻²])

Each candidate solution (fish) represents a set of these four hyperparameters. The fitness of a solution is the validation accuracy obtained after training a reduced MLP with a single hidden layer for a limited number of epochs on a subsample of the training data. This lightweight evaluation keeps the optimization computationally feasible. The solution with the best fitness represents the optimal hyperparameters which can then be used to train the MLP and improve its performance.

The EFO algorithm operates in the following steps:

1. Initialisation: A population of fish is randomly initialised within the bounds of the search space.
2. Active electrolocation (exploration): A fish moves towards a better performing fish selected randomly, with a step size that decreases over generations.
3. Passive electrolocation (exploitation): A fish moves towards a weighted combination of the best fish and another random fish, adding a small perturbation.
4. Selection: If the new position yields higher fitness, it replaces the old one.
5. Elite preservation: The best fish is retained across generations.

2.6 EFO Optimization

The baseline MLP is optimized by using the EFO to search for the best set of hyperparameters which are then used to train the MLP. After testing and observing the results obtained, further optimization was carried out to find the best set of EFO parameters which would generate the best fitness value. The EFO hyperparameters which are population size, number of generations, active ratio, and step size were tuned via a random search over 25 configurations, each evaluated on a reduced validation set. The best combination was then used for the final optimization run on the full training data.

The optimization process consisted of two phases:

- (i) Phase 1 (best EFO hyperparameter search): 25 random configurations of EFO parameters were evaluated. Each configuration ran a lightweight EFO optimization using with fitness evaluated on a 3000-sample subset over 8 epochs. The best EFO configuration was selected based on the highest fitness achieved.

(ii) Phase 2 (Final optimization): The optimal MLP hyperparameters were obtained from the best EFO configuration, and the optimized MLP was trained on the full training set (without any subsampling) for up to 150 epochs with early stopping, and evaluated on the held-out test set.

2.7 Experimental Setup

All the experiments were conducted on Kaggle notebooks with dual Tesla T4 GPUs (16 GB memory each). The software environment included TensorFlow 2.19, Python 3.12, and standard libraries (pandas, numpy, matplotlib, seaborn, scikit-learn). Random seeds were fixed to ensure reproducibility.

2.8 Evaluation Metrics

Given the multiclass nature of the fault classification task, the following metrics were computed on the test set:

(i) Accuracy: Overall proportion of correct predictions.

$$Accuracy = \frac{1}{N} \sum_{i=1}^C TP_i \quad (2.1)$$

where N = total number of samples, C = number of classes, TP_i = true positives for class i

(ii) Weighted precision: Average of per-class precision weighted by class support.

$$\text{for class } i, \quad Precision, P_i = \frac{TP_i}{TP_i + FP_i} \quad (2.2)$$

$$\text{weighted Precision} = \frac{1}{N} \sum_{i=1}^C support_i \times Precision_i \quad (2.3)$$

where $support_i$ = number of true instances of class i , N = total number of samples, C = number of classes, FP_i = false positives for class i

(iii) Weighted recall: Average of per-class recall weighted by class support.

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (2.4)$$

$$\text{weighted Recall} = \frac{1}{N} \sum_{i=1}^C support_i \times Recall_i \quad (2.5)$$

where $support_i$ = number of true instances of class i , N = total number of samples, C = number of classes, FN_i = false negatives for class i

(iv) Weighted F1-score: Harmonic mean of precision and recall, weighted by class support.

$$F1\ score_i = 2 \cdot \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (2.6)$$

$$\text{weighted F1 score} = \frac{1}{N} \sum_{i=1}^C support_i \times F1_i \quad (2.7)$$

(v) Confusion matrix: Visual representation of correct and misclassified instances per class.

(vi) Per-class precision, recall, and F1-score: Detailed performance for each fault type.

(vii) Paired t-test: Statistical comparison of per-class F1-scores between baseline and optimized models to determine significance of improvement.

3.0 Results and Discussion

The results obtained and the relevant discussions are presented in this section.

3.1 Baseline MLP and Optimized MLP Performance

The baseline MLP, with fixed architecture (128-128 neurons, dropout 0.3, learning rate 0.001), was optimized using default parameters. The convergence curve for the EFO optimizer in Figure 3 shows a sharp rise in the convergence after 5 generations showing that a high fitness value is attained quickly during the optimization.

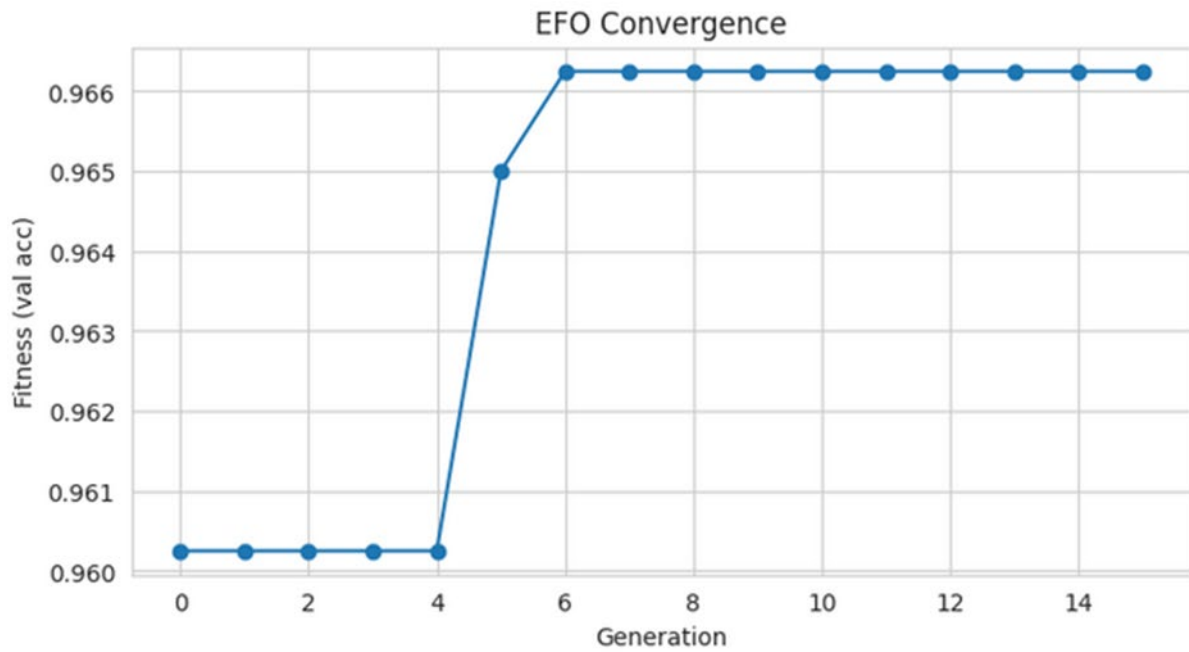


Figure 3: Convergence Curve of EFO-MLP

Figure 4 shows the confusion matrices for the Baseline MLP and EFO MLP. Both matrices indicate strong classification accuracy as shown by the diagonal values. There is also a decrease in the misclassifications (439 to 372) particularly for fault class ‘4’ (partial shading fault) indicating an improvement in accuracy after optimization.

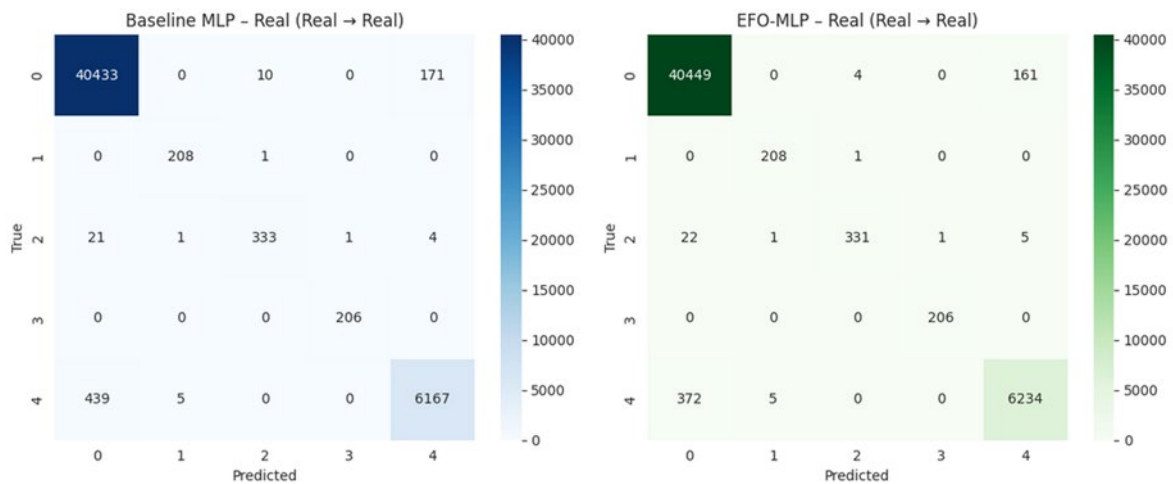


Figure 4: Confusion Matrices for Baseline MLP and EFO-MLP

3.2 Random Search for Optimal EFO Parameters

To fine tune the optimization, a random sample of 25 configurations of EFO parameters out of 400 was drawn and each configuration was assessed to find the optimal EFO meta parameters for the subsequent optimization of the final MLP classifier. Table 1 shows the best EFO configuration obtained which had a validation fitness of 0.9273. Table 2 shows the optimal MLP hyperparameters used to train the MLP classifier on the full dataset.

Table 1: Best EFO Hyperparameters from Random Search

Outcome	Parameter	Value
Best EFO hyperparameters	Population size	6.0
	Maximum generations	12.0
	Active ratio	0.5
	Step size	0.2
	Best fitness	0.9273

Table 2: Optimal MLP Hyperparameters from Best Fitness Value

Outcome	Parameter	Value
Optimal MLP hyperparameters	1 st hidden layer neurons	226
	2 nd hidden layer neurons	108
	Dropout	0.441
	Learning rate	0.00788

The optimized model was retrained on the full dataset and evaluated on the test set, yielding an accuracy of 0.9906. Incidentally all four metrics in Table 3 are nearly identical to four decimal places, indicating that the model is extremely well-balanced and makes very few errors. Figure 5 shows the training accuracy and training loss curves. The training and validation accuracy curves show near perfect convergence and plateau early and the training and validation loss curves decline sharply in tandem indicating that the model learns and generalizes well from the data shown to it. The confusion matrix (Figure 6) shows near-perfect classification, with only a handful of misclassifications in fault class ‘4’.

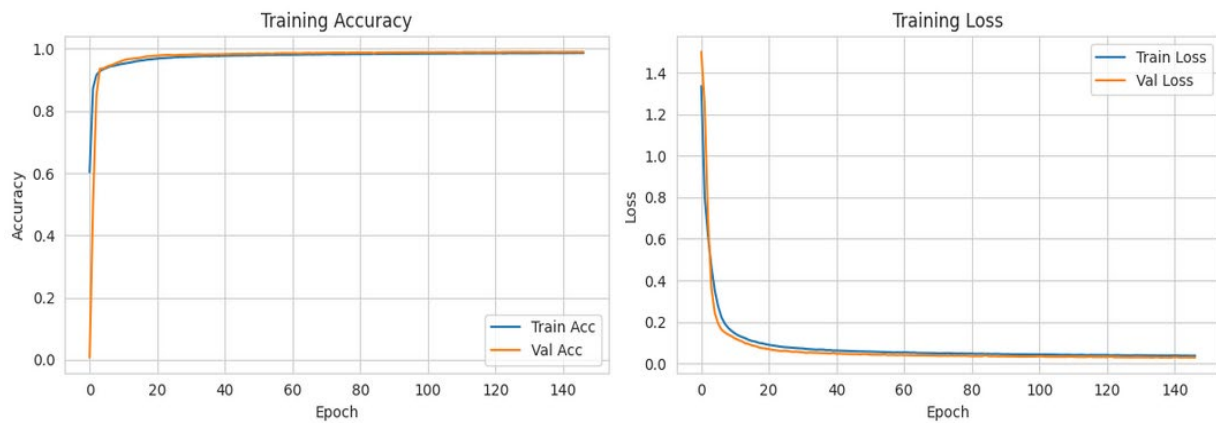


Figure 5: Training Accuracy and Training Loss Curve of Optimized MLP



Table 3: Metrics of EFO Random Search Optimized MLP

Figure 6: Confusion Matrix of Optimized MLP

Metric	Value
Accuracy	0.9906
Precision	0.9905
Recall	0.9906
F1 Score	0.9905

Tables 4 and 5 show the comparison between the metrics of the baseline and the re-optimized EFO. In table 4 an improvement of about 0.42% is observed in the F1 score and accuracy of the EFO MLP compared to the previous optimization which yielded 0.17% improvement in F1 score and accuracy indicating that the optimal hyperparameters obtained from the random search are quite effective in improving the overall performance of the diagnostic model. Table 5 shows a comparison of the per class F1 scores obtained from the baseline MLP and the two optimized versions (EFO and EFO-II) showing a clear upward trend in F1 score for all classes of faults. EFO-II is able to improve the F1 score for classes 1 and 3 which were not affected by the previous EFO optimization demonstrating higher performance and superiority.

In summary, the baseline MLP achieved an accuracy of 98.63%, while the optimized model achieved 99.06%, indicating improved classification performance. In addition to overall accuracy, the optimized model demonstrated improved per-class F1-scores across all fault categories, highlighting its effectiveness in handling class imbalance. The confusion matrix further confirmed improved classification consistency across fault types. The results also showed improved convergence behavior and reduced training error, indicating that the EFO effectively guided the search for optimal hyperparameters.

Although the absolute accuracy improvement is modest, the consistent improvement across all evaluation metrics and fault classes demonstrates enhanced robustness and generalization capability.

Table 4: Baseline and Optimized MLP F1 and Accuracy Comparison

Base Accuracy	Base F1	EFO-II Accuracy	EFO-II F1	Improved Accuracy (%)	Improved F1 (%)
0.986396	0.986277	0.9906	0.9905	0.4261	0.4281

Table 5: Baseline and Optimized MLP per class F1 Comparison

CLASS	BASELINE F1	EFO F1	EFO-II F1
0	0.992136	0.993137	0.994572
1	0.983452	0.983452	0.986717
2	0.946023	0.951149	0.964652
3	0.997579	0.997579	0.998058
4	0.952212	0.958266	0.966868

4.0 Conclusion

The optimized MLP achieved an accuracy of 99.06% compared to 98.63% for the baseline model, indicating improved classification performance. In addition, per-class F1-scores increased across all fault categories, demonstrating enhanced robustness in handling imbalanced data. The results confirm that EFO effectively improves hyperparameter selection, leading to better generalization and classification stability.

Overall, the results indicate that metaheuristic optimization can significantly enhance MLP-based fault diagnosis. The EFO was able to explore the hyperparameter search space to find a configuration that balances model capacity (226 neurons in the first layer) with strong regularization (dropout 0.441). The relatively high learning rate (0.00788) allowed faster convergence without causing instability.

This paper presented an EFO-optimized MLP model for fault diagnosis in grid-connected PV systems. The implementation of this approach improved classification accuracy and robustness compared to a baseline model. The findings demonstrate the effectiveness of metaheuristic optimization in enhancing ANN-based fault diagnosis systems. Future work may explore comparisons with other optimization algorithms and real-time deployment.

References

- [1] G. Nassreddine, A. El Arid, M. Nassereddine, and O. Al Khatib, "Fault Detection and Classification for Photovoltaic Panel System Using Machine Learning Techniques," *Applied AI Letters*, vol. 6, no. 2, p. e115, Apr. 2025, doi: 10.1002/ail2.115.
- [2] U. Hijjawi, S. Lakshminarayana, T. Xu, G. Piero Malfense Fierro, and M. Rahman, "A review of automated solar photovoltaic defect detection systems: Approaches, challenges, and future orientations," *Solar Energy*, vol. 266, p. 112186, Dec. 2023, doi: 10.1016/j.solener.2023.112186.
- [3] S. A. Memon, Q. Javed, W.-G. Kim, Z. Mahmood, U. Khan, and M. Shahzad, "A Machine-Learning-Based Robust Classification Method for PV Panel Faults," *Sensors*, vol. 22, no. 21, p. 8515, Nov. 2022, doi: 10.3390/s22218515.
- [4] C. B. Lema, S. Perabi Ngoffe, F. E. Ndi, G. Abessolo Ondoua, and S. Ndjakomo Essiane, "Conventional KPCA Approach Applied to Detect Simulated Faults in PV Systems Using Simulated Data," *International Journal of Photoenergy*, vol. 2024, no. 1, p. 5517822, Jan. 2024, doi: 10.1155/2024/5517822.

- [5] A. Freej, A. S. Sabik, and I. A. Nassar, "Performance Improvement of Photovoltaic Panels Through Advanced Fault Detection Techniques," *Processes*, vol. 13, no. 12, p. 3831, Nov. 2025, doi: 10.3390/pr13123831.
- [6] Z. Asimiyu, "A Novel Method for Real-Time Detection of Faults and Partial Shading in PV Arrays," Oct. 2024.
- [7] L. A. I. Carrera, G. Alfonso-Francia, C. D. Constantino-Robles, J. Terven, E. A. Chávez-Urbiola, and J. Rodríguez-Reséndiz, "Advances and Optimization Trends in Photovoltaic Systems: A Systematic Review," 2025.
- [8] M. Bougoffa, S. Benmoussa, M. Djeziri, and O. Palais, "Hybrid Deep Learning for Fault Diagnosis in Photovoltaic Systems," *Machines*, vol. 13, no. 5, p. 378, Apr. 2025, doi: 10.3390/machines13050378.
- [9] A. Teta, B. Korich, D. Bakria, N. Hadroug, A. Rabehi, M. Alsharif, M. Bajaj, I. Zaitsev, and S. S. M. Ghoneim, "Fault detection and diagnosis of grid-connected photovoltaic systems using energy valley optimizer based lightweight CNN and wavelet transform," *Sci Rep*, vol. 14, no. 1, p. 18907, Aug. 2024, doi: 10.1038/s41598-024-69890-7.
- [10] J. L. Da Silva, D. Z. Ñaupari Huatuco, and Y. P. Molina Rodriguez, "Fault Classification in Photovoltaic Power Plants Using Machine Learning," *Energies*, vol. 18, no. 17, p. 4681, Sep. 2025, doi: 10.3390/en18174681.
- [11] S. Yilmaz and S. Sen, "Electric fish optimization: a new heuristic algorithm inspired by electrolocation," *Neural Comput & Applic*, vol. 32, no. 15, pp. 11543–11578, Aug. 2020, doi: 10.1007/s00521-019-04641-8.
- [12] W. Luo, H. Wu, and J. Peng, "Improvement of Electric Fish Optimization Algorithm for Standstill Label Combined with Levy Flight Strategy," *Biomimetics*, vol. 9, no. 11, p. 677, Nov. 2024, doi: 10.3390/biomimetics9110677.
- [13] R. A. Ibrahim, L. Abualigah, A. A. Ewees, M. A. A. Al-qaness, D. Yousri, S. Alshathri, and M. Abd Elaziz, "An Electric Fish-Based Arithmetic Optimization Algorithm for Feature Selection," *Entropy*, vol. 23, no. 9, p. 1189, Sep. 2021, doi: 10.3390/e23091189.
- [14] B. Chitradevi, P. Mathiyalagan, A. Ramachandran, R. Dhanapal, K. Sheikdavood, and S. Gnanamurugan, "Conv-ViT: An improved discrete convolution-based vision transformer for diabetic retinopathy detection," *Franklin Open*, vol. 14, p. 100477, Mar. 2026, doi: 10.1016/j.fraope.2025.100477.
- [15] Y. Arikan Yildiz, Ö. P. Akkaş, M. Saka, M. Çoban, and İ. Eke, "ELECTRIC FISH OPTIMIZATION FOR ECONOMIC LOAD DISPATCH PROBLEM," *NÖHÜ Müh. Bilim. Derg.*, Mar. 2024, doi: 10.28948/ngumuh.1390037.
- [16] O. Alqaraghuli and A. Ibrahim, "Optimizing Photovoltaic System Diagnostics: Integrating Machine Learning and DBFLA for Advanced Fault Detection and Classification," *Electronics*, vol. 14, no. 8, p. 1495, Apr. 2025, doi: 10.3390/electronics14081495.
- [17] M. Arif, F. Ur Rehman, L. Sekanina, and A. S. Malik, "A comprehensive survey of evolutionary algorithms and metaheuristics in brain EEG-based applications," *J. Neural Eng.*, vol. 21, no. 5, p. 051002, Oct. 2024, doi: 10.1088/1741-2552/ad7f8e.
- [18] A. Nedaei, A. Eskandari, and M. Aghaei, "Photovoltaic fault detection and classification: Reconsideration of classic machine learning and dataset shrinkage techniques for simplification," *Results in Engineering*, vol. 27, p. 106356, Sep. 2025, doi: 10.1016/j.rineng.2025.106356.
- [19] Z. Yuan, G. Xiong, and X. Fu, "Artificial Neural Network for Fault Diagnosis of Solar Photovoltaic Systems: A Survey," *Energies*, vol. 15, no. 22, p. 8693, Nov. 2022, doi: 10.3390/en15228693.
- [20] A. E. Lazzaretti, C. H. D. Costa, M. P. Rodrigues, G. D. Yamada, G. Lexinoski, G. L. Moritz, E. Oroski, R. E. D. Goes, R. R. Linhares, P. C. Stadzisz, J. S. Omori, and R. B. D. Santos, "A Monitoring System for Online Fault Detection and Classification in Photovoltaic Plants," *Sensors*, vol. 20, no. 17, p. 4688, Aug. 2020, doi: 10.3390/s20174688.