International Journal Of
Industrial Electronics Control and Optimization
Print ISSN: 3517-2645
Online ISSN: 3568-2645
Homepage: https://ieco.usb.ac.ir/

Photovoltaic Fault Diagnosis with CNN-ResNet and Adaptive Learning Rate Scheduling

Sepehr Shakibi | Amir Mohammad Farahani | Mohsen Hamzeh

School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran.
Corresponding author's email: mohsenhamzeh@ut.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: 04-April-2025
Received in revised form: 17-June-2025
Accepted: 30-June-2025
Published online: 22-June-2026

Keywords:
Deep learning,
Fault detection,
Learning rate schedule,
Photovoltaic systems,
Renewable energy reliability.

ABSTRACT

Photovoltaic (PV) systems are a backbone of the infrastructure of renewable energy, with their usage growing significantly. Early fault detection of these systems is essential to enhance their reliability and efficiency. Despite the development of fault diagnosis methods of PV system promoted by machine learning models such as ensemble learning, support vector machine and neural networks, challenges in achieving high accuracy and generalization persist. In this paper, propose a deep learning method based on a ResNet architecture for reliable and efficient fault detection, including the following categories: Normal Operation, Short-Circuit, Degradation, Open Circuit, and Shadowing. Also devise a new learning rate schedule(LRS), which considerably improves the training dynamics and enables a 63% improvement in model performance. The suggested method achieves excellent performance, reaching 99.8% accuracy throughout the training, validation and testing phases. The results obtained showcase the potential of ResNet-based architectures, in addition to prowess in adaptive learning rate strategies, at enhancing the reliability of photovoltaic systems through scalable and precise fault diagnosis.

I. Introduction

Photovoltaic (PV) systems are utilized in various applications, and analyzing fault occurrences can significantly enhance their efficiency and operation. Faults in photovoltaic systems can be caused by reduced output power and damage to healthy modules[1]. Therefore, the automatic monitoring of faults in installed PV arrays is essential. Increasing user awareness about faults and their effects is critical, as this awareness can reduce photovoltaic system losses by up to 15%[2]. Consequently, detecting faults using rapid and accurate approaches is of great importance.

Numerous fault detection and diagnosis (FDD) methods have been proposed, which can be classified into two main categories: visual and thermal methods, and electrical methods. Electrical methods can be further divided into five subcategories, one of which includes artificial intelligence techniques [3]. Due to the characteristics of deep learning (DL) methods, intelligent FDD has become more automated and effective in the context of big data [4].

In the literature, various deep DL methods have explored customized architectures rooted in artificial neural networks, such as multi-layer perceptrons (MLP), convolutional neural networks (CNN), and long short-term memory (LSTM) networks, which are widely employed for fault diagnosis in photovoltaic systems. For instance, [5], introduces an artificial neural network (ANN) that processes three inputs through a single hidden layer of nine neurons, yielding six outputs: normal operation, one, two, or three open diodes, and one or two short-circuited diodes. Another proposed architecture is a dual-stream neural network for diagnosing five fault conditions and normal operation. This architecture incorporates LSTM and 2D-CNN with an attention mechanism. It combines multimodal data, including the I-V curves of PV strings and transformed 2D images, to deeply extract features under various PV shading conditions, ensuring accurate fault diagnosis [6]. In [7], a customized autoencoder-based semi-supervised deep learning method is followed by a hybrid SVM-LR model to classify four types of faults. In [8], a method for fault detection in PV systems using Siamese networks based on MobileNet (SiamMN) and

panoramic I-V features (PIVF) is presented. SiamMN consists of two parameter-sharing MobileNetv3 subnetworks that compute the similarity between two input PIVF features. These PIVFs are extracted from corrected and normalized I-V curves, and random pairing is employed to enhance the model's generalization capability. In [9], the Compact Convolutional Transformer (CCT) is used. This architecture combines CNNs with transformer mechanisms, allowing for efficient processing of image data while preserving spatial information. The proposed hybrid deep learning model for fault detection and classification in grid-connected PV systems integrates Wavelet Packet Transform (WPT) for preprocessing PV voltage signals, Stacked Autoencoders (SAE) for automatic feature extraction, the Equilibrium Optimizer Algorithm (EOA) for optimal feature selection, and LSTM for classification. This architecture enables the model to effectively detect and classify various faults, achieving an impressive accuracy of 99.93% in noiseless conditions while maintaining robustness against noise. The outputs include precise fault classifications and enhanced operational reliability, significantly improving maintenance scheduling and safety in PV systems [10]. Additionally, [11] introduce a hierarchical classification (HC) approach combined with machine learning techniques to detect and classify Line-Line (LL) and Line-Ground (LG) faults under challenging conditions like high impedance and low mismatch levels, achieving experimental accuracies of 96.66% and 91.66% for LL and LG faults, respectively, with reduced dataset requirements. In [12], PV-YOLO—a lightweight YOLO-based model enhanced with a transformer-based PVTv2 backbone and CBAM attention mechanism—targets fault detection in drone-captured infrared images, reaching a mean Average Precision (mAP) of 92.56% and excelling at identifying small occlusions. Furthermore, [13] transforms I-V and P-V curves into graphical features using the Gramian Angular Difference Field (GADF), employing a CNN with CBAM modules to diagnose faults in dust-affected conditions, attaining accuracies up to 99.17% even with noisy data and proving robust across varied PV configurations. Additionally, [14] propose a Stacked Autoencoder (SAE)-based approach to detect degradation in series resistance (Rs) in PV systems, utilizing Mean Squared Error (MSE) to quantify reconstruction errors, achieving robust fault detection under varying radiation and temperature conditions, with MSE analysis revealing high sensitivity to fault-induced changes. [15] introduce a deep learning framework for crack segmentation in PV cells using ensemble learning with U-Net, LinkNet, FPN, and Attention U-Net models, achieving a mean Intersection-over-Union (mIOU) of 54.19% and linking crack orientation and size to power efficiency losses, offering a comprehensive approach to both detection and impact assessment. [16] present a lightweight CNN optimized by the Energy Valley Optimizer (EVO) with Continuous Wavelet Transform (CWT)-generated scalograms as input, achieving test accuracies of 100%, 99.86%, and 93.75% across balanced, unbalanced, and noisy datasets, respectively, demonstrating superior performance and efficiency for deployment on low-cost edge devices. Additionally, [17] proposes a hybrid DL approach blending CNN with Bidirectional Gated Recurrent Unit (Bi-GRU) to detect and classify faults like open circuits, short circuits, and partial shading in PV systems, harnessing dynamic maximum power point outputs to deliver near-perfect accuracies validated with real-world data. [18] introduces a DL method utilizing an Autoencoder neural network for feature extraction, paired with an ANN for classifying short-circuit faults in a 9.54 kW PV system in Algeria, achieving flawless accuracy in noiseless settings and robust performance with noisy data. In addition to deep learning-based methods, other model-based and sensor-reduction techniques have shown promise. For example, an explicit piecewise quadratic model has been introduced for accurate I-V curve modeling in PV systems, offering linearity in parameters and low computational cost while maintaining high precision [19]. Moreover, innovative group-based monitoring techniques have been applied for open-circuit fault detection in converter systems, enabling fast diagnosis with a reduced number of sensors and minimal computational overhead [20]. These approaches reflect the broader trend toward efficient and accurate fault detection across various PV-related applications. Additionally, in [21] to combine various base classifiers using irradiance, temperature, current, and power as inputs. Its advantage lies in improved accuracy (94.25%) over individual classifiers. However, its limitation is high computational complexity due to ensemble learning. In [22], a hybrid classification system using chaotic-enhanced butterfly optimization and MLP was applied to electroluminescence (EL) images. It achieved 98.2% accuracy and robustness across evaluation metrics. Its strength is strong image-based detection using swarm intelligence, but the system depends on high-quality EL imaging, which can be costly and impractical for real-time industrial use. Another study [23] deployed AI-assisted monitoring across 150 PV stations in Taiwan. It used short-term power prediction combined with fault diagnosis and inverter efficiency tracking. Its advantage is real-world deployment with over 4,700 inverters and scalable architecture. However, its integration requires access to frequent irradiance simulation and high infrastructure setup. A focused method in [24] proposed ANN-based fault detection for standalone PV systems using power loss parameters, achieving 95% accuracy. The advantage is simple, interpretable input and suitability for real-time environments. Yet, it depends on simulation-derived fault data, which may limit generalization.

These diverse methodologies underscore the increasing sophistication of DL techniques in tackling the complex

challenges of PV fault diagnosis, adeptly addressing intricate fault patterns and adapting to real-world environmental variations, thus laying a solid groundwork for future advancements in this field.

In this paper, we propose a customized CNN-ResNet architecture that utilizes a learning rate scheduler (LRS). This architecture combines the strengths of CNN and ResNet. CNN architectures, along with some other models, face the vanishing gradient problem as the number of layers increases. As mentioned in [25], ResNet improves this challenge with shortcut connections. These connections in proposed architecture add the output of previous convolutional layers to the input, allowing the model to learn both previous and current features. The proposed model effectively classifies the considered faults, and when combined with a LRS, it results in reduced loss. Besides the model's power, it also provides a lightweight solution that can be used in real-time industrial environments. Moreover, it is important to note that the high accuracy achieved by the proposed method is not easily replicable using conventional classification techniques. As highlighted in [26] and [27], the dataset involves both simulated and real-world PV fault scenarios with overlapping patterns and varying environmental conditions. Classical machine learning methods such as k-NN, SVM, or shallow neural networks struggle with distinguishing between subtle fault types like degradation and shadowing under irradiance and temperature fluctuations. In contrast, our customized CNN-ResNet architecture, together with its adaptive learning strategy, addresses these challenges and provides a more robust and scalable solution.

This study is the first to couple a 1-D CNN–ResNet backbone with a two-phase adaptive learning-rate scheduler (linear warm-up + exponential decay) for photovoltaic fault diagnosis. The scheme achieves 99.8 % accuracy on a hybrid real/simulated dataset that includes five fault categories and severe class imbalance, and lowers validation loss by 63 % relative to the same network trained with a fixed LR. Thanks to lightweight 1-D convolutions and three residual blocks, the final model is suitable for real-time deployment on low-cost edge devices, outperforming MLP-, SVM- and vanilla CNN-based baselines reported in [26] (95.3 %) and [27] (97.4 %).

II. Methodology

A. Dataset and Faults

In this paper, the dataset from [26] was utilized, which was generated using a hybrid approach that combines real data from a PV plant with simulated data. It contains 16 days of data from a grid-tied photovoltaic plant, with two strings, each consisting of 8 PV modules, connected to a 5 kW grid-tied power inverter. Faults were introduced under controlled field conditions using hardware modifications. Short-circuit faults were created by connecting adjacent module terminals using sockets. Degradation was simulated by inserting resistors between modules, representing gradual performance decline. Open-circuit faults were introduced by disconnecting one string entirely from the inverter using circuit breakers. Shadowing was caused by physically blocking sunlight with opaque materials on selected modules. These fault types were also reproduced in simulation using a PSIM/Simulink model calibrated to the real system, enabling broader environmental coverage for training purposes. The details of the faults and their corresponding labels are summarized in Table I. Additionally, the dataset includes six features, with measurements for the voltage and current of each string, as well as irradiance and temperature. The temperature and irradiance ranges for the dataset were fixed, which could impact the model's ability to generalize to environmental conditions outside these predefined limits, potentially affecting its performance under more extreme or varying temperature and irradiance conditions.

TABLE I Fault categories and labels

Label	Fault	Description
0	Normal Operation	No faults
1	Short-Circuit	Short Circuit between 2 modules of a String
2	Degradation	There is a resistance between 2 modules of a String
3	Open Circuit	One String disconnected from the power inverter
4	Shadowing	Shadow in one or more modules

The parameters selected for analysis and used as inputs to the model are: irradiance, module temperature, string 1 voltage, string 1 current, string 2 voltage, and string 2 current. These parameters were chosen based on their strong influence on the power output of the system and their relevance in detecting fault conditions. The final input feature set consists of these six variables, which are available both in real measurements and in the simulated data. This ensures consistency and comparability across the full dataset.

B. Preprocessing

Data preprocessing is a crucial step in deep learning to enhance model performance. It involves cleaning the data by addressing noise, missing values, and irrelevant features. The data is then normalized and scaled to ensure all features are within a similar range, which helps the model learn more effectively. This preprocessing ensures consistency across training, validation, and testing datasets, leading to improved accuracy and faster model training.

To provide a clearer understanding of the dataset preprocessing, the dataset was analyzed and confirmed to not contain any missing values. Additionally, the number of samples for each class, as shown in Table II, is imbalanced, with the majority of the samples belonging to the normal operation class. Nevertheless, the proposed

TABLE II Sample count for each class in the dataset

Class	Count
Normal Operation	1162931
Short-Circuit	5999
Degradation	10371
Open Circuit	6024
Shadowing	188473

model is capable of effectively classifying between the different classes. To further improve the model's performance, the dataset was standardized after splitting it into training, validation, and test sets. Specifically, each feature in the dataset was scaled by subtracting the mean of the training data and dividing by the standard deviation of the training data. The values shown
in Table II represent the number of 1-second labeled samples recorded at a fixed sampling rate of 1 Hz. These counts do not indicate the number of physical fault-switching events, but rather the total number of seconds during which the system operated under each condition. The data were collected during 16 full days of system operation under natural environmental conditions and controlled fault scenarios. Partial shading occurred naturally at the test site during the early morning and late afternoon hours due to surrounding structures.

C. Model Architecture

To address the multi-class classification problem of photovoltaic (PV) system faults, which are often characterized by complex and subtle variations in time-series data (voltage, current, irradiance, and temperature), a CNN-based architecture is proposed that incorporates key principles from Computer Vision models. Specifically, the concept of Residual Blocks, inspired by the ResNet model [25], is leveraged to effectively capture intricate dependencies within the time-series data while maintaining computational efficiency.

The proposed model, depicted in Fig. 1, consists of a 1D Convolutional Neural Network (CNN) with a series of convolutional layers, followed by custom residual blocks and fully connected layers for classification. As shown in Fig. 1, the input time-series data is first processed by two consecutive convolutional layers. Each convolutional layer is followed by batch normalization and a ReLU activation function. These layers extract low-level features from the time-series data. For photovoltaic fault detection, a CNN-based architecture is chosen to capture both temporal dependencies and local features. Faults, such as short circuits, often occur within a very short time frame, and CNNs are well-suited for identifying these rapid changes in

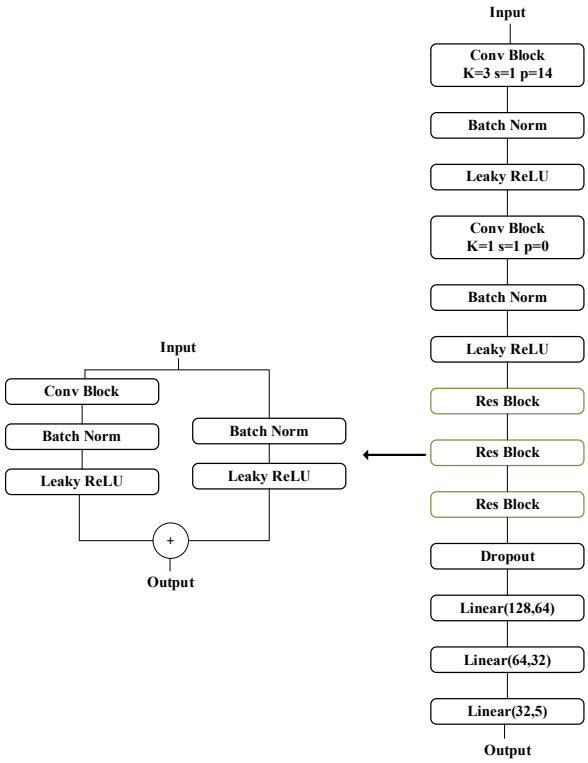

Fig. 1. Proposed model architecture that includes convolutional layers, ResNet blocks, and classifier layers, with the ResNet block providing downsampling and a skip connection.

the data. While models like RNNs or LSTMs are effective at capturing long-term dependencies in sequential data, CNNs offer advantages for this specific task due to their ability to efficiently extract local features from time-series data. Additionally, CNNs are expected to achieve higher accuracy than simpler architectures, as they are better suited to handle the complex patterns associated with photovoltaic system faults.

Following the convolutional layers, a sequence of three custom residual blocks is employed. As illustrated in Fig. 1, each residual block consists of two convolutional layers with a kernel size of 9 and a stride of 2. These layers extract higher-level features. Batch normalization and Leaky ReLU activation are applied after each convolutional operation. A downsampling path is incorporated within each block: the original input to the block is downsampled via average pooling, ensuring that the dimensions align for the residual connection. This allows the model to learn deeper representations and capture more complex relationships within the time-series data.

The output from the final residual block is flattened into a 1D vector. Subsequently, three fully connected layers are employed. These layers gradually reduce the dimensionality of the feature representation and ultimately produce the final

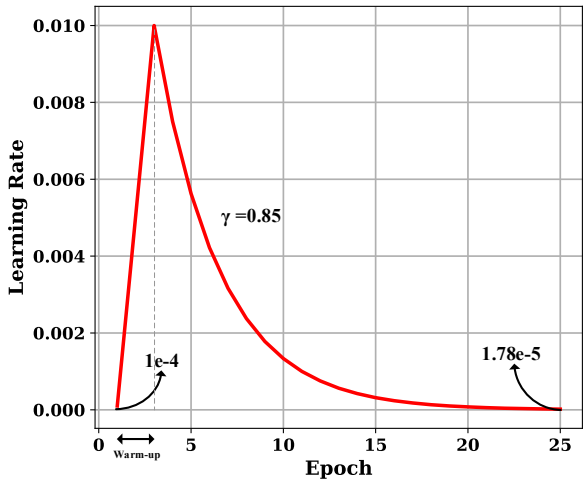

Fig. 2. Custom learning rate Scheduler: The learning rate begins at 0.0001 and increases linearly to 0.01 by the 3rd epoch. Following this, an exponential decay scheduler is applied.

output, consisting of five neurons corresponding to the five distinct PV fault classes.

Furthermore, to mitigate the risk of overfitting and improve generalization, dropout layers are incorporated after each fully connected layer. A dropout mechanism is selected to randomly deactivate neurons during training, ensuring that the model does not rely too heavily on specific features. Additionally, L2 regularization is applied to all convolutional and fully connected layers to penalize excessively large weight values, further promoting a more robust and generalizable model.

This architecture effectively combines the feature extraction capabilities of convolutional layers with the improved information flow and gradient propagation facilitated by residual connections. The downsampling within the residual blocks allows the model to capture both local and global temporal dependencies within the time-series data, while the expanded feature space enables the model to learn more complex and discriminative representations.

D. Model Training

During training, the Custom ResNet adapts its weights and biases with the objective of minimizing the error between the predicted fault labels and their ground-truth counterparts. The model was trained for 25 epochs, starting with an initial learning rate of 0.0001, which was incremented to 0.01 during the warm-up phase. The Adam optimizer was selected due to its known efficiency in updating weights and its adaptive learning rate properties. Adam combines the advantages of AdaGrad and RMSProp, performing particularly well on sparse gradients and non-stationary objectives. The optimizer function is defined as follows:

$$Adam(\theta) = \theta - \alpha \frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon} \qquad (1)$$

TABLE III Hyperparameter

Hyperparameter	Value	Role / Description	Selection Method
Initial LR (η_0)	1e-4	Starting learning rate for warm-up phase	Grid search (1e-5 to 1e-3) to prevent early instability
Max LR (η_{max})	0.01	Peak LR after warm-up; starting point for decay	Optimized via LR range test (loss decrease vs. stability tradeoff)
Warm-up Epochs	3	Linear increase of LR to η_{max}	Empirical: Minimum epochs to stabilize gradients (tested 1-5 epochs)
Decay Factor (γ)	0.85	Multiplier in exponential LR decay	Calculated to reach 1.78e-5 at epoch 25
Final LR	1.78e-5	Target learning rate at final epoch	Derived from γ decay
Batch Size	256	Number of samples per iteration	Max stable size for GPU memory
Epochs	25	Total training cycles	With early stopping (no improvement after 5 epochs)
Weight Decay	1e-6	L2 regularization strength	Grid search (1e-8 to 1e-4)

where α is the learning rate, $\hat{m}$ and $\hat{v}$ estimates of the first and second moments of the gradients , respectively, and ϵ is a small constant to prevent division by zero.

Cross-Entropy Loss, shown in equation (2), was used as the loss function. It computes the divergence between the predicted probability distribution and the ground-truth distribution, making it a critical metric for optimizing performance in multi-class classification tasks:

$$Cross\ Entropy\ Loss(y, \hat{y}) = -\sum_{i=1}^{c} y_i \log \hat{y}_i \qquad (2)$$

where y is the true label, $\hat{y}$ is the predicted probability distribution and c is the number of classes.

To optimize hyperparameters efficiently, Small Grid Search was used to fine-tune the learning rate and weight decay, ensuring effective hyperparameter selection. This approach improves model generalization, training stability, and convergence speed while maintaining computational efficiency. Fine-tuning these parameters plays a crucial role in preventing overfitting and achieving better overall performance.

Furthermore, Gradient Clipping was applied during training to prevent gradient explosion, a phenomenon where excessively large gradients cause unstable updates and hinder model convergence. A maximum gradient norm of 5 was used, ensuring that the gradients remain within a controlled range, leading to stable training dynamics.

Additionally, other hyperparameters such as batch size and the number of epochs were determined through empirical analysis, considering their impact on model effectiveness. The final values for all selected hyperparameters are summarized in Table III, reflecting the balance between performance and efficiency achieved during optimization.

The implementation of the proposed deep learning model was carried out using the Python programming language and

the PyTorch deep learning framework. PyTorch was selected due to its dynamic computation graph support, flexibility, and ease of integration with GPU-based acceleration. This environment facilitated the efficient implementation of the CNN-ResNet architecture, adaptive learning rate scheduling, and all training, validation, and evaluation procedures.

E. Learning Rate Scheduler

A custom LRS, combining linear warm-up and exponential decay, was employed to optimize training. The warm-up phase begins with a minimum learning rate and linearly increases it to the maximum value over a specified number of warm-up epochs. After the warm-up period, the learning rate exponentially decays at each epoch. This approach allows the model to explore regions of interest during the early stages, where stabilization is crucial, and in the later stages, the model can begin adjusting the weights more effectively.

During the warm-up phase, the learning rate increases from a minimal initial value to a predefined maximum. This guarantees the stability of the training process, ensuring that the model's parameters are not subject to large updates that could destabilize learning. At this stage, the learning rate at the t-th epoch is defined as:

$$LR_{warmup}(t) = initial_{lr} + (max_{lr} - initial_{lr}) \times \left(\frac{t}{warmup_{epochs}}\right) \quad (3)$$

where $initial_{lr}$ is the initial learning rate, max_{lr} is the maximum learning rate and $warmup_{epochs}$ is the number of epochs over which the learning rate is linearly increased.

The learning rate after the warm-up phase follows an exponential decay function, which facilitates model convergence. This decay acts as a regularization mechanism, ensuring that larger updates occur in the beginning, while smaller updates happen as training progresses, thus preventing overshooting of the optimal solution. The learning rate at the t-th epoch during this phase is derived as:

$$LR_{decay}(t) = max_{lr} \times \gamma^{t - warmup_{epochs}} \quad (4)$$

where γ (gamma) is the decay factor and t is the current epoch.

The scheduler promotes stable early training by mitigating disruptive parameter updates, enabling controlled fine-tuning, and offering flexibility through adjustable parameters ($initial_{lr}$, max_{lr}, $initial_{lr}$, γ). These tunable hyperparameters allow adaptation to different datasets and models. This approach helps achieve better performance during training by leveraging the benefits of a high initial learning rate (post-warm-up), followed by exponential decay.

The key hyperparameters used in the learning rate scheduler and training configuration are summarized in Table III. The initial learning rate (η_0 = 1e-4) was selected via grid search to ensure stable gradient updates during the early training steps. The maximum learning rate (η_{max} = 0.01) was determined based on an LR range test, balancing the rate of loss decrease with model stability. The warm-up phase was set to 3 epochs, which was found to be sufficient for smoothing the transition into full training based on empirical analysis across a range of 1 to 5 epochs. After this phase, an exponential decay was applied with a decay factor γ = 0.85, chosen to gradually reduce the learning rate to a final value of approximately 1.78e-5 by epoch 25. This learning rate strategy enables the model to benefit from both rapid early learning and fine-tuned convergence in later stages. Other hyperparameters such as batch size (256), total epochs (25), and weight decay (1e-6) were also tuned via grid search and empirical constraints such as GPU memory limitations. Early stopping was used to prevent overfitting if validation loss failed to improve over five consecutive epochs.

In Fig. 2, the changes in the learning rate during the training epochs are plotted. The plot shows linear growth during the warm-up phase, followed by exponential decay.

III. Simulation Results

This section evaluates the proposed CNN-ResNet architecture and highlights the positive impact of learning rate scheduling. The validation strategy involves evaluating the model on the validation dataset after each epoch and calculating metrics such as loss, accuracy, precision, recall, and F1 score. The validation dataset is considered unseen data, as it is separate from the training process. Additionally, to ensure the model generalizes well to unseen data, performance on the test dataset, which was not part of the training process, is also assessed using the same metrics. Table IV compares the model's accuracy and loss in two scenarios: with and without learning rate scheduling. The results show that the model achieves lower loss for the training, validation, and testing datasets when learning rate scheduling is used. Specifically, the loss decreases by approximately 69.51% for training data, 63.88% for validation data, and 64.86% for test data, compared to the scenario where learning rate scheduling is not used. Moreover, the model attains high accuracy—exceeding 99.8%—across all datasets, with learning rate scheduling contributing to accuracy increases of about 020% for the training dataset, 0.18% for the validation dataset, and 0.18% for the test dataset. Additionally, with learning rate scheduling, the Matthews Correlation Coefficient (MCC) reaches values of 0.9777 for training, 0.9731 for validation, and 0.9960 for testing, reflecting the model's improved ability to classify faults accurately and consistently across different datasets. As shown in Fig. 2, the learning rate starts at 1e-4, increases to 0.01 in the third epoch, and then decreases to approximately 1.78e-5.

This adjustment leads to greater fluctuations in the learning curve shown in Fig. 3 compared to Fig. 4, indicating that learning rate scheduling facilitates faster convergence

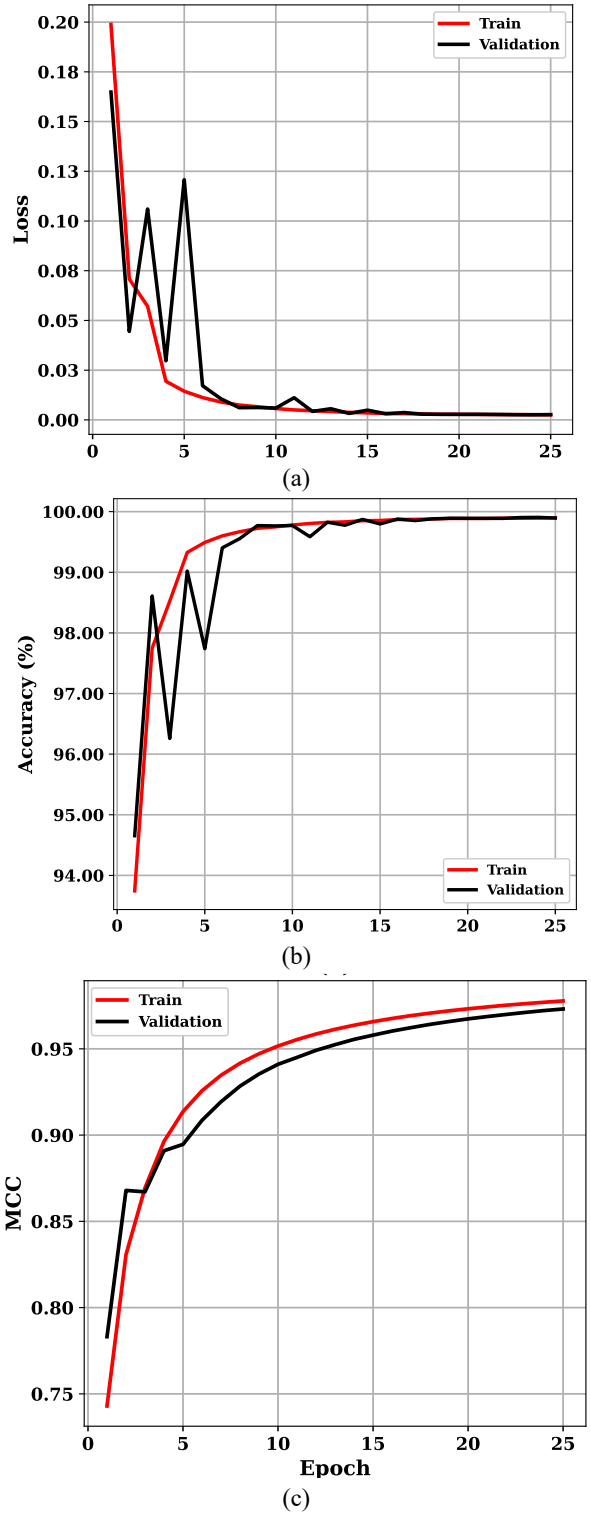

Fig. 3. Learning curve: (a) Loss per epoch, (b) Accuracy per epoch, (c)MCC per epoch with LRS.

during the increasing learning rate phase. Additionally, the learning curve becomes smoother after the learning rate decreases, showing reduced fluctuations compared to the scenario without scheduling.

Beyond the impact of learning rate scheduling, the proposed CNN-ResNet architecture demonstrates strong precision in

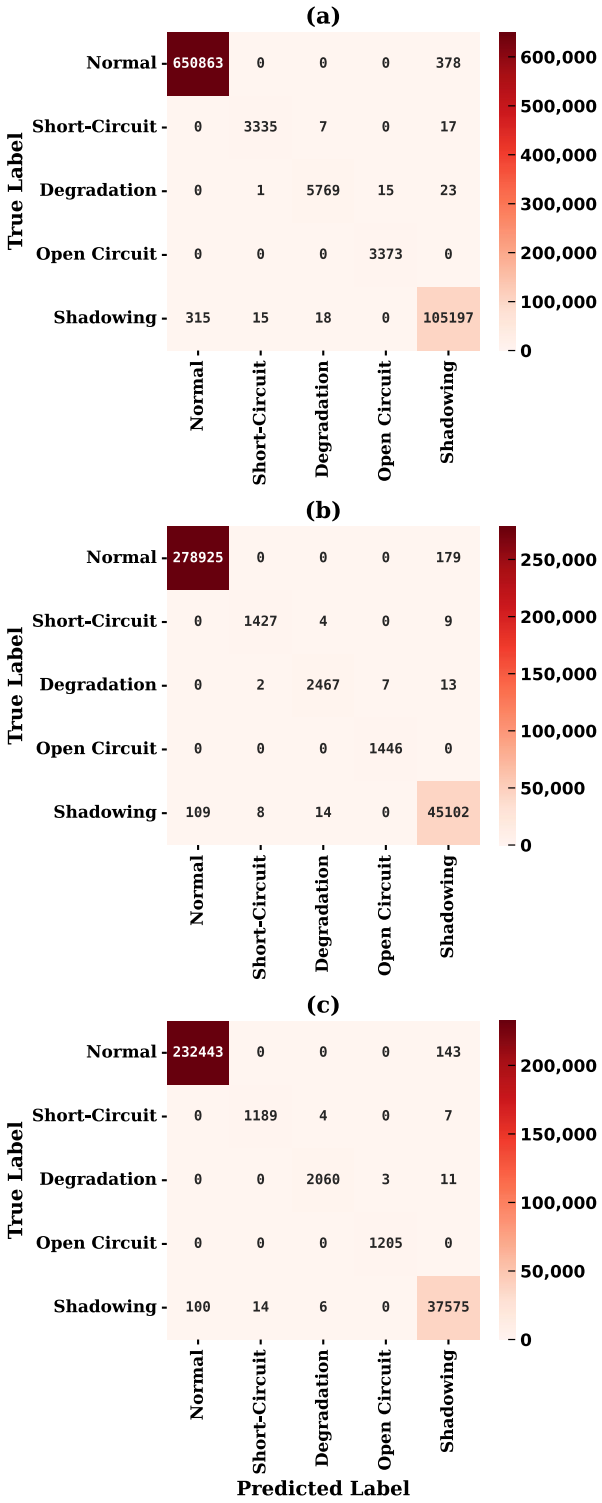

Fig. 4. Classification confusion matrix: (a) Training, (b) Validation, (c) Test.

predicting the target classes. Four key metrics—precision, ecall, F1 score, and AUROC—were used to evaluate model performance. Precision, which measures the proportion of true positive predictions among all positive predictions, and recall, which measures the proportion of true positives among all actual positive instances, both show impressive results. According to Tables V, VI, and VII, recall exceeds 0.99 for all classes across the training, validation, and test datasets, indicating that most actual positive instances are correctly identified, minimizing false negatives.

Furthermore, precision is greater than 0.98 for most classes in all datasets, suggesting that most positive predictions are true positives, thus minimizing false positives. The F1 score exceeds 0.95 for most classes, demonstrating a good balance between precision and recall. The AUROC value of approximately 0.99 for each class indicates excellent class discrimination, with the model effectively distinguishing between true positive instances and other classes (false positives). The confusion matrices in Fig. 4 illustrate the model's classification performance across training, validation, and testing phases. During training, the model effectively classifies a large number of instances across all fault categories. In the validation phase, the classification remains highly accurate, maintaining strong detection across different fault types. The testing phase further confirms the model's reliability in identifying photovoltaic system faults, demonstrating consistent and precise classification across all conditions. These results highlight the model's effectiveness in accurately detecting and distinguishing various operating states.

TABLE IV Accuracy and loss with/without LRS for training, validation and test data

Metric	LRS	Train	Validation	Test
Accuracy (%)	with	99.89	99.89	99.89
	without	99.69	99.71	99.71
Loss	with	0.0025	0.0026	0.0026
	without	0.0082	0.0072	0.0074
MCC	with	0.9777	0.9731	0.9960
	without	0.9768	0.9792	0.9891

TABLE V Metrics results for training datasets

Metric	LRS	Class0	Class 1	Class 2	Class 3	Class 4
F1Score	with	0.9967	0.9528	0.9552	0.9930	0.9815
	without	0.9965	0.9790	0.9779	0.9941	0.9785
Recall	with	0.9994	0.9928	0.9932	1.	0.9967
	without	0.9984	0.9913	0.9907	1.	0.9881
Precision	with	0.9995	0.9952	0.9956	0.9955	0.9960
	without	0.9981	0.9884	0.9905	0.9955	0.9900
AUROC	with	0.9999	0.9999	0.9999	0.9999	0.9999
	without	0.9999	0.9999	0.9999	0.9999	0.9999

TABLE VI Metrics results for validation datasets

Metric	LRS	Class0	Class 1	Class 2	Class 3	Class 4
F1Score	with	0.9959	0.9683	0.9533	0.9914	0.9768
	without	0.9969	0.9910	0.9844	0.9972	0.9802
Recall	with	0.9993	0.9909	0.9911	1.	0.9971
	without	0.9990	0.9902	0.9931	1.	0.9858
Precision	with	0.9996	0.9930	0.9927	0.9951	0.9955
	without	0.9978	0.9951	0.9860	0.9958	0.9938
AUROC	with	0.9999	0.9999	0.9999	0.9999	0.9999
	without	0.9999	0.9999	0.9999	0.9999	0.9999

TABLE VII Metrics results for test datasets

Metric	LRS	Class0	Class 1	Class 2	Class 3	Class 4
F1Score	with	0.9994	0.9895	0.9942	0.9987	0.9962
	without	0.9984	0.9863	0.9913	0.9983	0.9895
Recall	with	0.9993	0.9908	0.9932	1.	0.9968
	without	0.9989	0.9908	0.9942	0.9991	0.9859
Precision	with	0.9995	0.9883	0.9951	0.9975	0.9957
	without	0.9978	0.9818	0.9884	0.9975	0.9932
AUROC	with	0.9999	0.9999	0.9999	0.9999	0.9999
	without	0.9999	0.9999	0.9999	0.9999	0.9999

To further validate the proposed CNN-ResNet architecture and the contribution of the learning rate scheduler (LRS), a series of model comparisons and ablation studies were conducted. Table VIII summarizes the results of five model variants evaluated on the test dataset: the proposed CNN-ResNet with and without LRS, the same model without residual blocks, a standard CNN architecture, and a standard ResNet without scheduler. As shown, the full CNN-ResNet model with LRS achieves the highest test accuracy (99.89%) and lowest loss (0.0026), confirming the effectiveness of combining residual learning with an adaptive learning rate schedule. Removing the scheduler slightly decreases performance (99.71%, 0.0074), while removing residual blocks causes a significant drop in accuracy (92.44%) and a large increase in loss. The standard CNN and ResNet models also performed well (99.77% and 99.79%, respectively), but did not outperform the proposed approach. These results highlight the synergistic effect of residual connections and learning rate scheduling in improving both accuracy and convergence.

This result is particularly meaningful given the complexity of the dataset and the subtle nature of some fault classes. The dataset includes both real and simulated samples under a variety of realistic environmental conditions, such as irradiance and temperature variations, making accurate classification more challenging. Previous studies on this dataset report lower performance using other architectures: in [26], a Multi-Layer Perceptron (MLP)-based classifier achieved a test accuracy of 95.3%, and in [27], a 1D CNN architecture reached 97.4%. These comparative results highlight that the high performance of the proposed model is not trivially obtainable using standard classification techniques, and instead, results from the tailored integration of residual learning and dynamic learning rate scheduling.

In summary, the results highlight the effectiveness of the proposed CNN-ResNet architecture combined with learning rate scheduling. The model consistently delivers high accuracy, low loss, and robust classification metrics across the training, validation, and test datasets. Learning rate scheduling plays a crucial role in reducing loss by over 74% and slightly increasing accuracy, underlining its importance

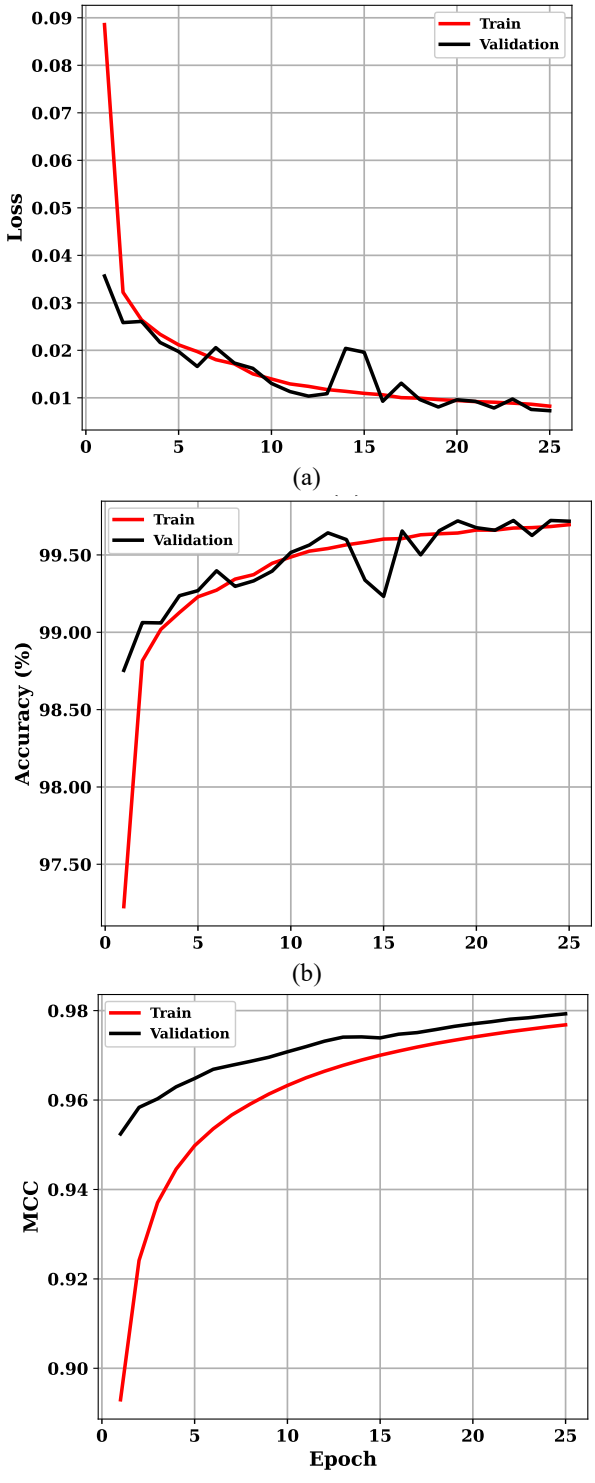

Fig. 5. Learning curve: (a) Loss per epoch, (b) Accuracy per epoch without LRS.further confirm that the model is well-trained and performs accurate predictions.

in improving model performance. Additionally, the model demonstrates excellent precision, recall, F1 score, and AUROC values, confirming its strong ability to distinguish between classes and make accurate predictions. To further validate the design choices, several model variants were implemented and evaluated on the test dataset, including a standard CNN, a standard ResNet, and ablated versions of the proposed architecture without residual connections or without learning rate scheduling. As summarized in Table VIII, the full CNN-ResNet with LRS achieved the best performance, while removing residual blocks or the scheduler led to significant drops in accuracy and increases in loss. These results emphasize the individual and combined impact of residual learning and adaptive scheduling on model generalization and convergence, further supporting the robustness of the proposed approach.

TABLE VIII Baseline comparisons and ablation studies on the test dataset, demonstrating the impact of residual connections and learning rate scheduling on the performance of different model variants.

Model	Accuracy (%)	Loss
CNN-ResNet + LRS	99.89	0.0026
CNN-ResNet (no LRS)	99.71	0.0074
CNN-ResNet (no residual)	92.44	0.1906
CNN	99.77	0.0057
ResNet	99.79	0.0048

IV. Conclusions

To address the challenges of fault detection in photovoltaic (PV) systems, this study aimed to develop a robust ResNet-based deep learning framework. The model was trained using the cross-entropy loss function and the Adam optimizer on a preprocessed and normalized version of the dataset to ensure input data consistency. Key improvements to the learning strategy included learning rate scheduling, with a strong emphasis on early warm-up iterations to prevent fast convergence, followed by fine-tuning of weight updates using an exponential decay function in the later training phases. This approach led to a significant improvement in model generalization, reducing the validation error by 63% compared to baseline training methods.

Our proposed model achieved an impressive accuracy of 99.8% across the training, validation, and test datasets, demonstrating both strong performance and significant generalization, with no overfitting. It also proved to be robust, with class-specific F1-scores of 98.95%, 95.28%, 95.52%, 99.30%, and 98.15% on the test dataset, showing its ability to discriminate between multiple fault types in diverse scenarios.

The results confirm the efficacy of the ResNet-based technique for PV fault detection and suggest its potential for adaptation to complex, real-world scenarios. Future work could integrate visual data (e.g., electroluminescence or thermal images of solar panels), where ResNet's prowess in image recognition could be leveraged. This would enable simultaneous analysis of electrical and physical defects, marking a major improvement for diagnostics.

REFERENCES

[1] S. S. Sakthivel, V. Arunachalam, and K. Jagatheesan, "Detection, classification, and location of open-circuit and short-circuit faults in solar photovoltaic array: an approach using single sensor," *IEEE Journal of Photovoltaics,* 2023, doi: 10.1109/JPHOTOV.2023.3304113.

[2] N.-C. Yang and M. Faizan, "Long Short-term Memory-based Feedforward Neural Network Algorithm for Photovoltaic Fault Detection under Irradiance Conditions," IEEE Transactions on Instrumentation and Measurement, 2024, doi: 10.1109/TIM.2024.3413159.

[3] A. Mellit, G. M. Tina, and S. A. Kalogirou, "Fault detection and diagnosis methods for photovoltaic systems: A review," Renewable and Sustainable Energy Reviews, vol. 91, pp. 1-17, 2018, doi: 10.1016/j.rser.2018.03.062.

[4] M. Mansouri, M. Trabelsi, H. Nounou, and M. Nounou, "Deep learning-based fault diagnosis of photovoltaic systems: A comprehensive review and enhancement prospects," IEEE Access, vol. 9, pp. 126286-126306, 2021, doi: 10.1109/ACCESS.2021.3110947.

[5] M. Dhimish and A. M. Tyrrell, "Photovoltaic bypass diode fault detection using artificial neural networks," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-10, 2023, doi: 10.1109/TIM.2023.3244230.

[6] Z. He, H. C. Chen, S. Shan, Y. Hu, K. Zhang, and H. Wei, "Shading type and severity diagnosis in photovoltaic systems via IV curve imaging and two-stream deep neural network," Energy Conversion and Management, vol. 324, p. 119311, 2025, doi: 10.1016/j.enconman.2024.119311.

[7] U. Kumar, S. Mishra, and K. Dash, "An IoT and semi-supervised learning-based sensorless technique for panel level solar photovoltaic array fault diagnosis," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-12, 2023, doi: 10.1109/TIM.2023.3287247.

[8] B. Ren, Q. Wang, N. Zhou, and Y. Chi, "Fault Diagnosis of Photovoltaic Strings Based on SiamMN Networks and Panoramic IV Features," IEEE Transactions on Industrial Electronics, 2024, doi: 10.1109/TIE.2024.3443964.

[9] Y.-Y. Hong, L.-F. Chen, and W. Zhang, "Classification of Photovoltaic Faults Using PSO-Optimized Compact Convolutional Transformer," IEEE Access, 2023, doi: 10.1109/ACCESS.2023.3341889.

[10] M. Alrifaey et al., "Hybrid deep learning model for fault detection and classification of grid-connected photovoltaic system," IEEE Access, vol. 10, pp. 13852-13869, 2022, doi: 10.1109/ACCESS.2022.3140287.

[11] A. Eskandari, J. Milimonfared, and M. Aghaei, "Fault detection and classification for photovoltaic systems based on hierarchical classification and machine learning technique," IEEE Transactions on Industrial Electronics, vol. 68, no. 12, pp. 12750-12759, 2020.

[12] W. Yin, S. Lingxin, L. Maohuan, S. Qianlai, and L. Xiaosong, "PV-YOLO: lightweight YOLO for photovoltaic panel fault detection," IEEE Access, vol. 11, pp. 10966-10976, 2023, doi: 10.1109/ACCESS.2023.3240894.

[13] J. Qu, Q. Sun, Z. Qian, L. Wei, and H. Zareipour, "Fault diagnosis for PV arrays considering dust impact based on transformed graphical features of characteristic curves and convolutional neural network with CBAM modules," Applied Energy, vol. 355, p. 122252, 2024, doi: 10.1016/j.apenergy.2023.122252.

[14] M. Bougoffa, S. Benmoussa, M. Djeziri, and T. Contaret, "Stacked AutoEncoder based diagnosis applied on a Solar Photovoltaic System," IFAC-PapersOnLine, vol. 58, no. 4, pp. 384-389, 2024, doi: 10.1016/j.ifacol.2024.07.248.

[15] A. Sohail, N. U. Islam, A. U. Haq, S. U. Islam, I. Shafi, and J. Park, "Fault detection and computation of power in PV cells under faulty conditions using deep-learning," Energy Reports, vol. 9, pp. 4325-4336, 2023, doi: 10.1016/j.egyr.2023.03.094.

[16] A. Teta et al., "Fault detection and diagnosis of grid-connected photovoltaic systems using energy valley optimizer based lightweight CNN and wavelet transform," Scientific Reports, vol. 14, no. 1, p. 18907, 2024, doi: 10.1038/s41598-024-69890-7.

[17] A. F. Amiri, S. Kichou, H. Oudira, A. Chouder, and S. Silvestre, "Fault detection and diagnosis of a photovoltaic system based on deep learning using the combination of a convolutional neural network (cnn) and bidirectional gated recurrent unit (Bi-GRU)," Sustainability, vol. 16, no. 3, p. 1012, 2024, doi: 10.3390/su16031012.

[18] A. Seghiour, H. A. Abbas, A. Chouder, and A. Rabhi, "Deep learning method based on autoencoder neural network applied to faults detection and diagnosis of photovoltaic system," Simulation Modelling Practice and Theory, vol. 123, p. 102704, 2023, doi: 10.1016/j.simpat.2022.102704.

[19] E. Moshksar, "An Explicit and Accurate I-V Characteristic for Photovoltaic Modules Based on Piecewise Quadratic Function," International Journal of Industrial Electronics Control and Optimization, vol. 7, no. 4, pp. 291-299, 2024, doi: 10.22111/ieco.2024.48517.1554.

[20] M. Rahmani, S. M. Barakati, S. Yousofi Darmian, V. Barahouei, and M. Bagheri Hashkavayi, "Optimized Algorithm for Open-Circuit Fault Detection in Switches and Capacitor Voltage Balancing Control in Modular Multilevel Converters," International Journal of Industrial Electronics Control and Optimization, vol. 7, no. 1, pp. 15-27, 2024, doi: 10.22111/ieco.2024.47310.1509.

[21] B. Liu, K. Sun, X. Wang, J. Zhao, and X. Hou, "Fault diagnosis of photovoltaic strings by using machine learning-based stacking classifier," IET Renewable Power Generation, vol. 18, no. 3, pp. 384-397, 2024, doi: 10.1049/rpg2.12755.

[22] W. M. Shaban, "Detection and classification of photovoltaic module defects based on artificial intelligence," Neural Computing and Applications, vol. 36, no. 27, pp. 16769-16796, 2024, doi: 10.1007/s00521-024-10000-z.

[23] M. Chang, K.-H. Chen, Y.-S. Chen, C.-C. Hsu, and C.-C. Chu, "Developments of AI-assisted fault detection and failure mode diagnosis for operation and maintenance of photovoltaic power stations in Taiwan," IEEE Transactions on Industry Applications, 2024, doi: 10.1109/TIA.2024.3379319.

[24] N. Sabri, A. Tlemçani, A. Chouder, and W. Merrouche, "An Improved Fault Diagnosis in Stand-Alone Photovoltaic System Using Artificial Neural Network," Iranian Journal of Science and Technology, Transactions of Electrical Engineering, vol. 48, no. 1, pp. 325-336, 2024, doi: 10.1007/s40998-023-00671-0.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[26] A. E. Lazzaretti et al., "A monitoring system for online fault detection and classification in photovoltaic plants,"

Sensors, vol. 20, no. 17, p. 4688, 2020, doi: 10.3390/s20174688.

[27] 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, 2022, doi: 10.3390/s22218515.

Sepehr Shakibi received the B.S. degree in electrical engineering from the University of Science and Culture, Tehran, Iran, in 2021, and the M.S. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2023. He is currently a Researcher with the Photovoltaic Systems Laboratory, the Department of Electrical and Electronics Engineering, University of Tehran, Tehran, Iran. His current research interests include photovoltaic systems and artificial-intelligence applications in this domain.

Amir Mohammad Farahani was born in Tehran, Iran. He received the B.S. degree in Electrical Engineering from the University of Science and Culture, Tehran, Iran, in 2019, and the M.S. degree in Digital Electronics Engineering from the Iran University of Science and Technology (IUST), Tehran, Iran. From 2019 to 2021, he was with GADA Group, Tehran, Iran. From 2021 to 2023, he was with Borna Electronics, Shahr-e Rey, Iran, as an Embedded Systems Developer. His current research interests include AI-based medical systems, natural language processing (NLP) and large language models (LLMs), and IoT-based edge devices.

Mohsen Hamzeh received the B.Sc. and M.Sc. degrees in Electrical Engineering from the University of Tehran, Tehran, Iran, in 2006 and 2008, respectively, and the Ph.D. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2012. From 2013 to 2018, he served as an Assistant Professor at Shahid Beheshti University, Tehran, Iran. In 2018, he joined the School of Electrical and Computer Engineering at the University of Tehran, where he is currently an Associate Professor. His research interests include photovoltaic systems, microgrid control, and the applications of power electronics in power distribution networks.