IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 August 2022, accepted 15 September 2022, date of publication 20 September 2022,
date of current version 28 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208115

RESEARCH ARTICLE

TL–LED^{arc}Net: Transfer Learning Method for Low-Energy Series DC Arc-Fault Detection in Photovoltaic Systems

YOONDONG SUNG[1], (Member, IEEE), GIHWAN YOON[1], JI-HOON BAE[2], (Member, IEEE), AND SUYONG CHAE[3], (Member, IEEE)

[1]Department of Energy ICT Convergence Research, Korea Institute of Energy Research (KIER), Daejeon 34129, Republic of Korea
[2]Department of AI and Big Data Engineering, Daegu Catholic University (DCU), Gyeongsan 38430, Republic of Korea
[3]Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, Republic of Korea

Corresponding author: Suyong Chae (sychae@postech.ac.kr)

This work was supported in part by the Technology Development Program to Solve Climate Changes through the National Research Foundation of Korea (NRF) by the Ministry of Science, ICT, under Grant 2021M1A2A2043890; and in part by the Research and Development Program of the Korea Institute of Energy Research under Grant C2-2421-01.

ABSTRACT The arc-fault phenomenon in photovoltaic (PV) systems has emerged as a major problem in recent years. Existing studies on arc-fault detection in conventional PV systems primarily focus on detecting typical stable arc-faults. Low-energy arc-faults are more challenging to detect than stable arc-faults because of their low current distortions, short durations, and nonlinear properties. These low-energy arc-faults, which are precursors to stable arc-faults, could even inflict serious damage on the system components. Here, a transfer learning-based low-energy arc-fault detection network (TL–LED^{arc}Net) using a two-stage training method is proposed to proactively detect series DC arc-faults by considering low-energy arc-faults. A one-layer long short-term memory network combined with a lightweight one-dimensional convolutional neural network was developed to detect low-energy arc-faults by only using the sensed current information. The results of offline and online experiments conducted with a commercial grid-connected PV inverter indicate that the proposed method can perform real-time operations on a single-board computer and detect low-energy arc-faults with an accuracy of 95.8%, which is higher than previous methods considered in this study.

INDEX TERMS Long short-term memory, low-energy arc-fault, photovoltaic systems, proactive detection, transfer learning.

I. INTRODUCTION

Renewable energy is a promising solution to the climate change problems caused by the overuse of fossil fuels. Solar power generation accounts for a peak of 600 GW, which is more than 3% of the global power-generation output [1], [2].

Despite the improvement and steady development of related technologies in the photovoltaic (PV) fields, arc-faults are a severe problem as they reduce the lifespan of PV systems. Most arc-faults in PV systems have been observed in electrical conversion devices, such as inverters and combiners, and in junction boxes consisting of different types of connectors [3]. Arc-faults are typically attributed to wire damage caused by external factors (e.g., long-term weathering and aging or wiring problems), which produce high-temperature plasma through an unexpected ionization process and continually damage the internal system components.

In terms of the arc-fault characteristics according to the circuit type, as shown in Fig. 1, the parallel arc-fault current is generally higher than the series arc-fault current. Thus, the input current can be readily cut off using a physical circuit breaker. In addition, the detection of arc-faults is relatively easier at zero-crossing points for a series AC circuit [4]. In contrast, it is difficult for a protection relay to detect arc-faults in the case of a series DC circuit because there are no zero-crossing points in the input current. Furthermore,

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Tucci.

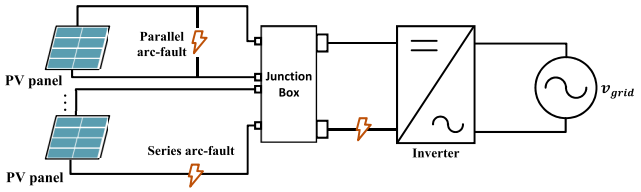

FIGURE 1. Photovoltaic (PV) system indicating possible locations of series and parallel arc-faults.

arc-faults are difficult to detect in most commercial inverters owing to the large amount of electronic noise and ripple currents they generate [5]. Therefore, the UL1996B standard was established in the USA to provide guidelines for arc-fault detection in PV systems [6].

Various arc-fault detection methods have been proposed, including physical modeling, time–frequency domain analysis, and artificial intelligence (AI) methods. For example, Uriarte *et al*. analyzed the arc-fault generated in a microgrid while controlling the speed and gap in an arc-fault generator in the time domain and verified the arc-fault properties through simulations by modeling the results [7]. In addition, impedance modeling and empirical methods have been employed to solve DC low-power and low-voltage arc-fault problems [8]. Most notably, the effects of the contact material properties, current, voltage, arc-fault speed, and arc-fault gap on the arc-to-glow evolution process were demonstrated [8].

Fault detection methods that make use of sensor signal-based preprocessing analysis have also been studied. For example, a statistically varying threshold-based fault detection method using a microgrid consisting of multiple sources and loads was proposed [9]. Additionally, [10], [12], [13], arc-fault detection approaches using both time- and frequency-domain analysis were introduced based on short-time Fourier transform (STFT) and discrete wavelet transform. However, these techniques are limited because they depend on the frequency band and specific characteristics of the device, such as the switching noise from the inverter, control signals, and background noise signals [14]. In addition, several methods to improve the optimization performance of rule-based fault detection methods and solve the uncertainty in the real environment were presented [15], [16], [17].

In an attempt to address these shortcomings, AI-based arc-fault detection methods have recently garnered considerable interest [18]. For example, machine learning (ML) algorithms based on an ensemble tree were investigated [19], and a stacking ensemble model was proposed. However, the arc-fault data were created and evaluated using simulation-based circuit design, which presented various drawbacks. In [20], [21], and [22], various arc-fault detection methods were proposed using the support vector machine (SVM) technique with the input current in the time-frequency domains. However, compared to deep learning (DL) approaches based on deep neural networks (DNNs), existing ML-based methods are inefficient for solving nonlinear problems because their preprocessing approaches are highly complex and require expert domain knowledge. This motivated the development of various DL-based methods for arc-fault detection [23], [24], [25], [26] as well as fault diagnosis [27]. In section II, we introduce existing DL methods for arc-fault detection and analyze their weaknesses particularly with respect to the detection of low-energy arc-faults. Furthermore, we discuss the contributions of the proposed detection method by considering the characteristics of low-energy arc-faults.

II. RELATED WORK AND PROPOSED CONCEPT

Arc-fault detection methods based on convolutional neural networks (CNNs), have recently garnered considerable attention as DL-based methods. One approach involves analyzing the arc characteristics from reconstructed two-dimensional (2D) images in the time–frequency domains [23], [24]. Other than this, Lu *et al*. [25] introduced an approach that expanded the formation of a training dataset and improved the arc-fault detection accuracy by generating dummy arc-fault data using a domain adaptive deep convolutional generative adversarial network.

However, previous studies focused mainly on typical stable arc-faults and did not consider low-energy arc-faults in detail. Low-energy arc-faults precede stable arc-faults and occur when the fault energy is not sufficient. In contrast to stable arc-faults, low-energy arc-faults are nonlinear, have low distortions, and short durations [26]. The same researchers [26] discovered that low-energy arc-faults are the precursors to stable arc-faults and are capable of damaging system components. In this regard, low-energy arc-fault detection by analyzing the DC is a challenging problem as commercial inverters generally produce high ripple currents and a large amount of switching noise in a real environment.

Various studies [8], [11], [28], [29], [30] were devoted to the experimental analysis of the input current and arc voltage characteristics of low-energy arc-faults and the effects of these characteristics on the systems were studied. A data-driven method was recently introduced [26] to detect low-energy arc-faults; however, this method was restricted to AC circuits, and no distinction was made between low-energy arc-faults and stable arc-faults. Furthermore, the trained model was not tested in a real environment owing to the complexity of the preprocessing procedures for the SVM classifier.

Therefore, a detailed analysis of the nonlinear phenomena associated with low-energy arc-faults and the development of an efficient method to detect these faults accurately in real time has become necessary. This study extends existing arc-fault detection capability, including low-energy series DC arc-faults, to improve the reliability of PV systems, and led to the proposal of a new proactive arc-fault detection method named the transfer learning (TL)-based low-energy arc-fault detection network (TL–LED^{arc}Net).

TL allows new problems to be solved more efficiently by using knowledge acquired in the past. Generally, when a deep neural network (DNN) is successfully trained, it usually

contains valuable information in the form of internal weights that have already been learned. In most cases, a DNN-based TL technique for solving classification problems reuses most of the remaining architecture of the already trained DNN model with multiple convolutional layers. Hence, pretrained DNN models that have acquired a wealth of knowledge are widely used in the field of TL. Following the aforementioned approach to TL, we propose a two-stage training method combining a one-dimensional convolutional neural network (1D CNN) and long short-term memory (LSTM) based on TL.

The main contributions of this study are as follows:

1) This study extends the range of arc-fault detection to incorporate low-energy arc-faults and improve conventional approaches, which are generally sensitive to stable arc-faults. The detection of low-energy arc-faults, which can cause the reliability of PV systems to deteriorate, is crucial to prevent severe damage to these systems.
2) The input current is analyzed in the time–frequency domain through visualization using gradient-weighted class activation mapping (Grad-CAM), which is employed in the field of explainable AI [31]. The results of the visualization analysis are extracted from the spectrogram image in the frequency region by activating a heatmap of the arc-fault class below the 20-kHz frequency band, and are used as the basis for designing a lightweight arc-fault detection model.
3) A proactive arc-fault detection method named TL–LED^{arc}Net is proposed. This method analyzes the arc-fault characteristics from an energy perspective and distinguishes between low-energy and stable arc-faults. In addition, an improvement in the accuracy is successfully demonstrated by conducting real-time experiments using a lightweight model.

The remainder of this paper is organized as follows: Section III describes the arc-fault detection platform and presents the analysis from the energy perspective. Section IV explains the architecture and optimization procedure of TL–LED^{arc}Net. Section V presents the evaluation of the performance of TL–LED^{arc}Net in an actual online/offline test environment. Section VI contains our concluding remarks.

III. DATA ACQUISITION AND ANALYSIS

This section describes the data acquisition platform used to develop the proposed proactive arc-fault detection algorithm. The low-energy series DC arc-faults are analyzed statistically based on the time-domain data, and a labeling method is introduced based on the energy perspective proposed in this study.

A. ARC-FAULT DATA ACQUISITION AND EXPERIMENTAL PLATFORM

As shown in Fig. 2, the series DC arc-fault detection platform comprised four parts: series DC arc-fault generator,

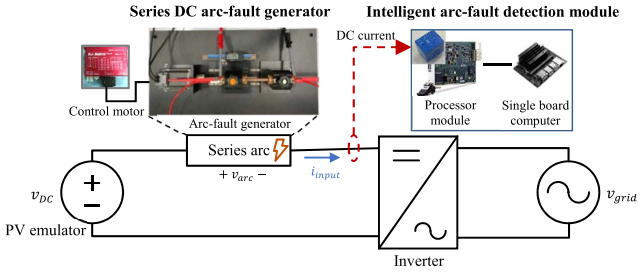

FIGURE 2. Series DC arc-fault detection platform.

PV emulator, grid-connected commercial inverter, and intelligent arc-fault detection module. In the figure, v_{arc} denotes the arc voltage, i_{input} denotes the input current of the inverter, and v_{DC} denotes the series DC bus voltage, which was 600 V. The rated power of the grid-connected inverter was 3.8 kW.

Additionally, a series DC arc-fault generator was implemented based on the UL1699B standard [6]. Arc-faults with a small gap length or low speed are more likely to exhibit low-energy series DC arc-fault characteristics, as demonstrated previously [8]. Therefore, experiments were conducted repeatedly to generate statistical arc-fault data consisting of arc-fault gap lengths ranging from 0.4–1.1 mm and control motor speeds ranging from 0.4–1.1 mm/s. Lastly, the intelligent arc-fault detection module used to mount the proposed TL–LED^{arc}Net was implemented using a single-board computer. Section V-B presents a detailed verification of the real-time arc-fault detection performance.

B. ARC-FAULT FEATURE ANALYSIS

Fig. 3 shows the arc voltage, v_{arc}, and the input current of the inverter, i_{input}, measured at a sampling frequency of $f_s = 100$ kHz under the experimental conditions presented in Section III-A.

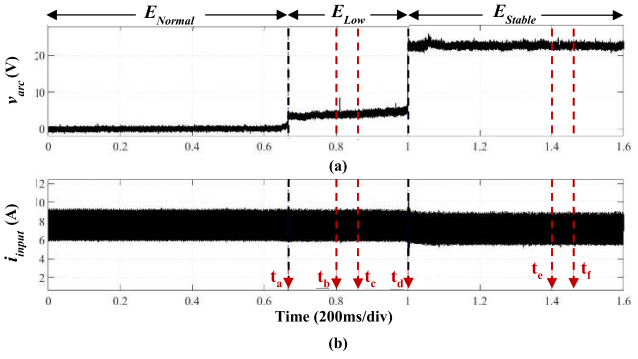

FIGURE 3. Measurements recorded using the platform in Fig. 2. (a) Series DC arc voltage. (b) Input current of inverter.

Fig. 3 indicates that the measured data could be classified based on the energy level as being in a normal state, E_{Normal} ($0 \sim t_a$), a low-energy arc-fault state, E_{Low} ($t_a \sim t_d$), or a stable arc-fault state, E_{Stable} (after t_d), where E_{Stable} includes the transient phase in which the input current increases rapidly.

The E_{Low} state was produced owing to a lack of arc-fault energy in the ionization process, and it exhibited nonlinear and low distortion characteristics [8], [23]. Conversely, when the arc-fault energy accumulated and preheating occurred completely, it developed into the E_{Stable} state through the transient phase. The pattern of the arc voltage and input current gradually converged and stabilized in the E_{Stable} state. In this regard, the current fluctuation in this stabilized E_{Stable} state existed within the range allowed by the protection relay [4]. Furthermore, the experimental results show that the input current variation of the E_{Low} state is similar to that of the E_{Stable} state.

To analyze the characteristics of the low-energy arc-fault, the E_{Low} state was calculated with a time duration of 50 ms, as follows:

$$E_{Low} = \frac{1}{t_c - t_b} \int_{t_b}^{t_c} v_{arc}(t) \cdot i_{input}(t) dt, \tag{1}$$

where E_{Low} represents the arc-fault energy between times t_b and t_c, v_{arc} denotes the arc voltage, and i_{input} denotes the input current.

To simplify the energy calculation, the arc voltage and input current data were approximated using their root mean square (RMS) values, denoted v_{RMS} and i_{RMS}. Similarly, E_{Stable} was calculated via (1) using the data from t_e to t_f.

Table 1 presents the statistical analysis of the measured arc-fault energies, in which 500 samples with a time duration of 50 ms were analyzed for E_{Low} and E_{Stable}. Fig. 4 illustrates the energy densities of series DC arc-faults for two classes in a grid-connected PV system. The results in Table 1 and Fig. 4 indicate that the E_{Low} values are primarily distributed between 22 J and 56 J, and the average energy is approximately 42 J.

TABLE 1. Comparative statistical analysis of E_{Low} and E_{Stable}.

State	Mean	Standard deviation	5% energy	95% energy
E_{Low}	42.1 J	9.9 J	22.1 J	55.9 J
E_{Stable}	149.8 J	26.1 J	97.8 J	192 J

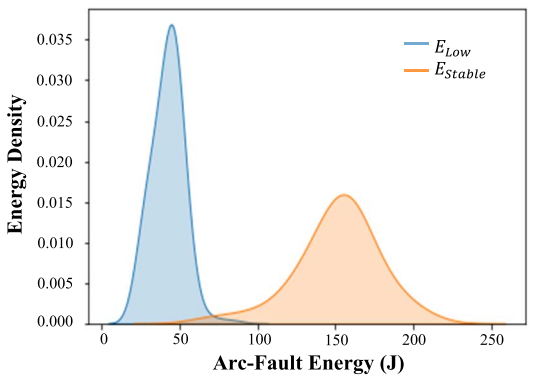

FIGURE 4. Comparison of arc-fault energy densities.

Therefore, the upper boundary for E_{Low} ($E_a \approx 56$ J) and the lower boundary for E_{Stable} ($E_b \approx 97$ J) were defined as the 95% confidence interval corresponding to each population. More importantly, E_{Low} continuously affected the system adversely and could have developed into the E_{Stable} state. Therefore, based on statistical analysis from the perspective of the arc-fault energy, developing a fast and accurate arc-fault detection algorithm is essential to solving this problem.

IV. PROPOSED PROACTIVE ARC-FAULT DETECTION METHOD

A. ARC-FAULT SIGNAL ANALYSIS USING GRAD-CAM TECHNIQUE

The characteristics of the series DC arc-fault data in the time–frequency domain were analyzed by applying the Grad-CAM technique, which is used in the explainable AI field. The analysis of the arc-fault characteristics consisted of a stage for preprocessing the time–frequency input data, a training stage that employed a 2D CNN model, and a Grad-CAM application stage, as shown in Fig. 5. First, to simultaneously observe the characteristics of the input current in the time–frequency domains, STFT signal preprocessing was applied to the input current as follows:

$$\mathrm{STFT}\{x(t)\}(\tau, w) = \int_{-\infty}^{\infty} x(t) w(t-\tau) e^{-jwt} dt, \tag{2}$$

where $x(t)$ is the input current, $w(\tau)$ is the window function, and τ is the time index. A sampling rate of 100 kHz and a time duration of 50 ms were used.

Subsequently, the spectrogram image generated using (2) was input into the 2D CNN model presented in Fig. 5, and the classifier was used for arc-fault detection training in three categories: E_{Normal}, E_{Low}, and E_{Stable}. The Grad-CAM technique visually analyzed the arc-fault data with a class activation heatmap. Grad-CAM provided important regional information on the arc-fault class in the spectrogram image by extracting the feature map, F^k, of the last convolutional

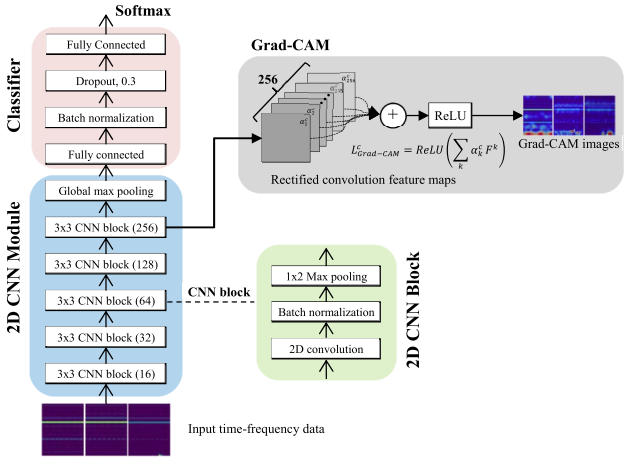

FIGURE 5. Arc-fault analysis architecture built by applying the Grad-CAM technique.

layer. It also calculated the gradient average of the class for all the extracted feature map channels as shown in (3).

$$\alpha_k^c = \frac{1}{z}\sum_i \sum_j \frac{\partial y^k}{\partial F_{i,j}^k}, \tag{3}$$

where y^c is the probability value for class c and Z is a normalization factor. Accordingly, the gradient was first calculated by performing backpropagation on y^c, and the weighting α_k^c with the feature map channel was obtained by calculating the average gradient for each channel.

A linear combination was then performed to multiply the importance of each channel (α_k^c) by the corresponding channel of the feature map (F^k) and generate a spatial heatmap for the arc-fault class as follows:

$$L_{Grad-CAM}^c = \mathrm{ReLU}(\sum_k \alpha_k^c F^k), \tag{4}$$

where ReLU denotes rectified linear unit [32].

Fig. 6 shows the spectrogram images of E_{Normal}, E_{Low}, and E_{Stable} obtained by implementing the Grad-CAM technique. In the figure, the important regional information that was used for classification is highlighted in red. It was found that the devised 2D CNN model avoided the 32 kHz switching noise of the inverter in the high-frequency domain and mainly utilized frequency information of less than 20 kHz for arc-fault detection.

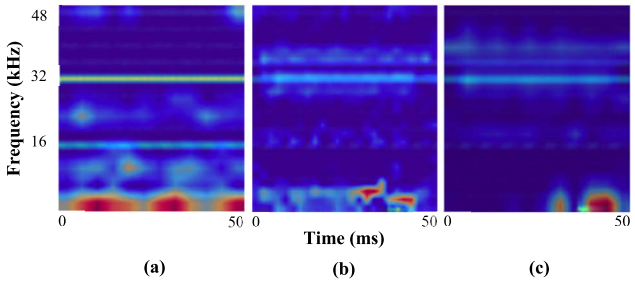

FIGURE 6. Heatmap visualization for the arc-fault classes using the Grad-CAM technique: (a) E_{Normal} state, (b) E_{Low} state, (c) E_{Stable} state.

According to the results of the Grad-CAM visualization analysis, the complexity of the data preprocessing and the training model architecture can be effectively reduced by selecting the appropriate frequency range of the arc-faults.

B. DESIGN OF TL–LED^{arc} NET FOR PROACTIVE ARC-FAULT DETECTION

This section describes the structure and optimization procedure of TL–LED^{arc}Net. First, based on the analysis results presented in Section IV-A, a 20-kHz lowpass filter was used to pass the low-frequency components of the input current. Considering the mean value of the input current, data normalization for preprocessing was then performed by applying mean subtraction normalization (MSN) to the lowpass filtered data as follows:

$$\tilde{x}_i(n) = x_i(n) - \frac{1}{N}\sum_{j=1}^{N} x_j, \tag{5}$$

where $x_i(n)$ represents the discrete time-sequence sample feature for the input current $x(t)$, N denotes the total number of points per sample feature, and $\tilde{x}_i$(n) denotes the MSN output feature of a sample i. To maintain the existing signal characteristics (e.g., the standard deviation) while ensuring that the average for sample i is zero, the arc-fault data were normalized by considering only the mean value of the input data, as indicated in (5).

The network architecture of TL–LED^{arc}Net, shown in Fig. 7, is now explained in detail. 1) Stage 1 performs pretraining, in which the normal state and the arc-fault state are binarized in a lightweight 1D CNN model consisting of two convolutional layers. 2) A pretrained 1D CNN network, which is a feature extractor, is loaded into only one LSTM layer. 3) Stage 2 detects a series DC arc-fault based on the training using the three-class classifier. TL is performed based on prior knowledge of the local features of the arc-fault extracted from Stage 1 using normal and arc-fault data as the training dataset. This process retrains the sequence association between the arc-fault features to differentiate between E_{Low} and E_{Stable}.

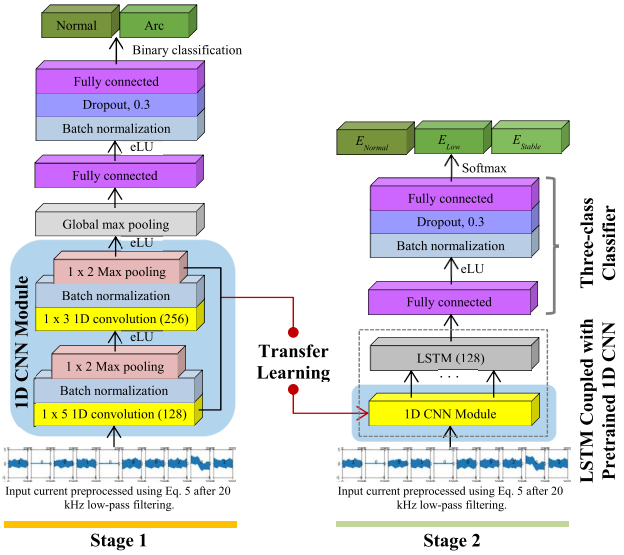

FIGURE 7. Complete architecture of TL–LED^{arc}Net for two-stage training.

The network architecture of TL–LED^{arc}Net shown in Stage 1 was composed of two 1D convolutional layers and a two-class classifier. Batch normalization [33] was used for all the convolutional layers to prevent the vanishing gradient problem. In addition, a max-pooling layer was added after batch normalization to reduce the spatial dimensionality of the convolutional layer output, which preserved relevant features and removed irrelevant details. The kernel size for the convolution operation in the first convolutional layer was set to 5, and the stride was set to 2. In the second convolutional layer, the kernel size of the 1D convolution window was set to 3, and the stride was set to 2.

The convolutional operation in the convolutional layer is given by

$$z(n) = \tilde{x}(n) * w(n) = \sum_{m=-l}^{l} \tilde{x}(m)w(n-m), \quad (6)$$

where $\tilde{x}_i(n)$ denotes a normalized 1D input current using (5), $w(n)$ denotes a kernel function with a window size of l, and $z(n)$ represents a 1D feature map resulting from a convolutional layer calculated using $\tilde{x}_i$(n) and $w(n)$.

The exponential linear unit (eLU) was used as an activation function in this study [34]. The advantage of this activation function is that it can increase the speed of training in the designed networks and produce higher recognition accountability than the original ReLU [32].

Finally, a classifier was introduced to include two fully connected layers and a dropout layer to perform binary classification, which ensured that overfitting did not occur. In Stage 2, E_{Low} was detected by an LSTM model and a classifier that performed three-class classification, in which a TL technique with a pretrained 1D CNN model was considered. In this study, the LSTM network was applied with feedback neural connections to effectively train the sequence between the pretrained arc-fault features in Stage 1. The LSTM model is based on the existing recurrent neural network (RNN) model, but it has additional cell state memory in the hidden state, making it more efficient for extracting long-term features for time-series data than the existing RNN. The LSTM network designed in Stage 2 consisted of one layer composed of 128 hidden neurons and 256 features extracted from the 1D CNN module.

Here, the input gateway of a cell constituting the LSTM was composed of a forget gate, input gate, and output gate [35]. These respective components are defined as:

$$\begin{aligned} i_t &= \sigma(w_i[h_{t-1}, z_t] + b_i), \\ f_t &= \sigma(w_f[h_{t-1}, z_t] + b_f), \\ o_t &= \sigma(w_o[h_{t-1}, z_t] + b_o), \end{aligned} \quad (7)$$

where i_t denotes the input gate, f_t denotes the forget gate, o_t denotes the output gateway, z_t denotes the output of the 1D CNN module in Stage 2 of Fig. 7, σ represents the sigmoid function, w_i, w_f, and w_o denote the weights for input z_t at each gate, respectively, h_{t-1} denotes the output value of the previous LSTM cell, and b_i, b_f, and b_o denote the respective bias values for the input z_t at each gate. In the LSTM structure, the input gateway determines whether to store the current information, the forget gateway determines whether to utilize past information, and the output gateway determines which value to present as output. For example, in the forget gateway, the sigmoid function utilizes past information when the output value is one. Additionally, a cell state (C_t) corresponds to memory, which refers to the storage space occupied by the information determined by the input and forget gates. In the output gate, the cell gate performs the $o_t * \tanh(C_t)$ function and determines its value to determine the value of the model output in (7).

The classifier in Stage 2 used two fully connected layers, while batch normalization was applied to prevent the problem of vanishing gradients. Additionally, a dropout was applied to prevent the arc-fault detection model from overfitting the data. Lastly, the final output layer used a probability-based softmax function to perform the three-class classification, distinguishing the E_{Normal}, E_{Low}, and E_{Stable} states.

To optimize the proposed two-stage training model, categorical cross entropy used for multi-classification was applied as a loss function, defined as follows:

$$\text{Categorical Cross Entropy} = -\frac{1}{N}\sum_i y_i \cdot \log(\tilde{y}_i), \quad (8)$$

where N is the total number of samples, $\tilde{y}_i$ denotes the predicted class probability, and y_i denotes the actual class. During model training, TL–LED^arcNet weights were iteratively adjusted to minimize the cross-entropy loss between $\tilde{y}_i$ and y_i. In this regard, the Adaptive Moment Estimation (Adam) optimizer was adopted in TL–LED^arcNet [36]. The Adam optimizer, which served to minimize the loss, is a stochastic gradient descent method based on the adaptive estimation of first- and second-order moments. The advantages of this optimizer are that it is straightforward, easy to use, and requires less memory than other optimizers. Most importantly, it is a robust optimizer and well suited for non-convex optimization problems such as our problem in the field of ML and DL. The overall learning procedure for the proposed method is summarized in Algorithm 1.

Algorithm 1 Two-Stage Training Procedure for Proactive Arc-Fault Detection Employed by TL–LED^arcNet

- **Stage 1**: Train the 1D CNN model to extract compressed regional features of the arc-fault signal from time-series current data.

1) Configure the 1D CNN module and the two-class classifier, as shown in Fig. 7.

2) Extract arc-fault features from the 1D convolutional layers using the convolutional operation of (6).

- **Stage 2**: Retrain the LSTM network coupled with the pretrained 1D CNN module for three-class classification.

1) Load the feature extractor of the pretrained 1D CNN module into the one-layer LSTM, as shown in Fig. 7.

2) Optimize the model weights using the Adam optimizer through the retraining process using TL-based prior knowledge.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the verification of the performance of TL-LED^arcNet by conducting online and offline experiments. The offline experiments with a 3-phase PV inverter operating at 3.8 kW (v_{DC} = 500 V, i_{input} = 7.5 A, f_{sw} = 32 kHz) were executed to verify the superior performance of the proposed TL-LED^arcNet compared with existing ML- and DL-based methods. In addition, online experiments were also

performed to verify the real-time arc-fault detection capability of the proposed method using a single-board computer (NVIDIA Jetson Nano). The NVIDIA Jetson Nano is configured with a 1.43 GHz Quad-core ARM Cortex A57 MPCore, 4 GB memory, and a 10 W power unit.

A. OFFLINE EXPERIMENTAL RESULTS

The offline experiments utilized input current data from the experimental platform, as shown in Fig. 2. The training and test datasets were collected using the intelligent arc-fault detection module. The input current was first measured using a current sensor (LEMHX 15-p) with a 50-kHz bandwidth, and analog-to-digital conversion with 20-kHz lowpass filtering of the input current was performed using a processor module (NI sbRIO-9606). The dataset consisted of 10,279 E_{Normal}, 10,018 E_{Low}, and 10,062 E_{Stable} data values. The test dataset used for the model evaluation comprised 3,036 data values sampled from the total collected data, and the remaining data values were used to train the model. Furthermore, the validation dataset used for hyperparameter optimization in the training process comprised 10% of the training data.

The Adam learning algorithm was applied to each stage, and the initial learning rate was set to 0.0001 to perform 60 training epochs. In this case, to ensure that the algorithm accurately and rapidly converged to the optimal point, the adaptive learning rate technique was applied. In other words, the adaptive learning rate was decreased step-wise by 1/10 from the initial learning rate. Fig. 8(a) shows the loss of training and verification data as a function of the number of epochs, and Fig. 8(b) depicts the accuracy of the training and verification data. The results shown in Fig. 8 indicate that overfitting or underfitting for a given verification dataset did not occur. This is because the training was performed by optimizing the hyperparameters of TL–LED^{arc}Net.

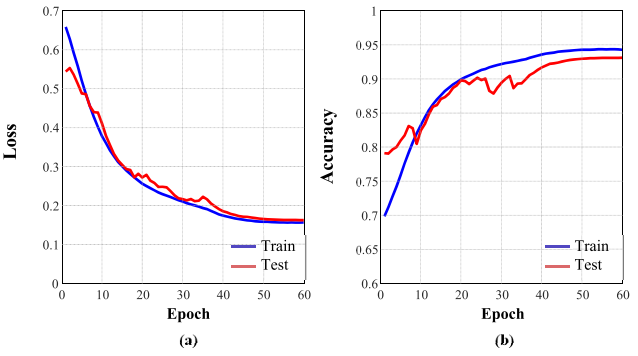

FIGURE 8. Variation in the (a) loss and (b) accuracy for the training and verification data with the number of epochs.

The performance accuracy of TL–LED^{arc}Net was then determined using the statistical multi-class classification evaluation method [37]: accuracy, sensitivity (recall), specificity, and fallout (false alarm). Table 2 presents the confusion matrix results for the proposed method, of which 3,036 test data values were randomly extracted for the evaluation of the model. Among the data values in the test set, 1,015 E_{Normal}, 1,007 E_{Low}, and 1,014 E_{Stable} data values were used for the evaluation. The experimental results are presented in Table 2, which indicates the average of a total of five experiments. As shown in Table 2, the results of the diagnostic performance evaluation of the E_{Low} class of TL–LED^{arc}Net presented an overall accuracy of 95.8% and a sensitivity of 93.3%. If all the arc-fault states are integrated into the positive class, and the normal states are redefined as the negative class, then the binary classification performance of the arc-fault detection model produced an overall accuracy of 99.6%, and a sensitivity of 99.8%. Note that the sensitivity indicator represents the result of the trained model, i.e., its ability to predict the input sample as an arc-fault in the actual arc-fault class.

TABLE 2. Confusion matrix results.

		Predicted class			
	Class	E_{Normal}	E_{Low}	E_{Stable}	Total
Actual class	E_{Normal}	1,011	4	0	1,015
	E_{Low}	12	942	53	1,007
	E_{Stable}	0	58	956	1,014
	Total	1,023	1,004	1,009	3,036

The arc-fault detection accuracy of TL–LED^{arc}Net was compared with that of five representative ML-based and three DL-based methods in the offline experiment. First, Table 3 compares the arc-fault detection performance of TL–LED^{arc}Net with that of existing ML-based methods: SVM, random forest (RF), light gradient boosting machine (Light GBM), extra gradient boost (XGBoost), and a stacked ensemble model composed of RF, XGBoost, and Light GBM submodels. In this case, all the existing ML methods employed the following five features: average, median, variation, RMS, and the difference between the maximum and minimum values for the time-domain input current (introduced in [19]). The optimized ML training parameters obtained using grid search with cross-validation were also applied to each corresponding ML method.

TABLE 3. Comparison of arc-fault detection accuracy of TL–LED^{arc}Net with that of existing ML and DL methods.

ML Methods	SVM	RF	Light GBM	XGBoost	Stacked Ensemble
Accuracy (%)	71.8	75.0	81.0	83.6	88.5
DL Methods	**2D CNN**	**1D CNN**	**Directed-Coupled 1D CNN and LSTM**		**TL–LED^{arc}Net (Ours)**
Accuracy (%)	92.9	92.1	92.8		95.8

In addition, the arc-fault detection performance of TL–LED^{arc}Net was compared with that of DL-based methods such as 2D CNN, 1D CNN, and direct-coupled 1D

TABLE 4. Performance comparison between TL–LED^{arc}Net and the 2D CNN model for each class of arc-faults.

Methods	Class	Accuracy	Sensitivity	Specificity	Fallout	Parameters
2D CNN	E_{Normal}	99.0	98.7	99.2	0.8	440,067
	E_{Low}	92.9	86.8	96.0	4.0	
	E_{Stable}	93.9	93.2	94.2	5.8	
TL–LED^{arc}Net (Ours)	E_{Normal}	99.5	99.5	99.5	0.5	332,803
	E_{Low}	95.8	93.5	96.9	3.1	
	E_{Stable}	96.3	94.2	97.4	2.6	

CNN as well as LSTM models. In the case of the 2D CNN model in Table 3, the spectrogram image was used with the STFT method for the input data and a design involving three convolutional layers, similar to the structure presented in Section IV. Subsequently, Stage 1 in Fig. 7 without TL was applied to the time-domain input current in the case of the 1D CNN model. Lastly, the direct-coupled 1D CNN and LSTM models used a simple combination of the 1D CNN and LSTM models described in Stage 2 without TL except for Stage 1, as shown in Fig. 7.

In terms of the arc-fault detection accuracy, the overall performance of the DL-based methods was superior to that of the existing ML-based methods, as indicated in Table 3. Among the DL-based methods, the proposed TL–LED^{arc}Net delivered the best performance with a detection accuracy of 95.8%, attributed to the stage-wise training approach using TL. This result confirmed that TL–LED^{arc}Net is a more efficient training method than the direct-coupled model of 1D CNN and LSTM as well as the 2D CNN model.

Table 4 compares the results of TL–LED^{arc}Net and the 2D CNN model using the STFT-based spectrogram, which was analyzed by applying the statistical classification evaluation. The experimental results indicate that the overall accuracy of TL–LED^{arc}Net was 2.9% higher than that of the 2D CNN model. Notably, in the case of the sensitivity for the E_{Low} class, the detection accuracy was 6.7% higher, and the false alarm rate was 0.9% lower, indicating that the false detection performance was lower than that of the 2D CNN model.

Considering the number of parameters as an indicator of the complexity of a model, TL–LED^{arc}Net was 24.4% more lightweight compared to the 2D CNN model. Given that TL–LED^{arc}Net utilized only the time-domain input current, the computational complexity of $O(TN\log_2 N)$ corresponding to the STFT transformation was further reduced, unlike the 2D CNN model. Here, T denotes the window size for the STFT conversion, and N denotes the number of samples.

Finally, a model operation experiment was conducted on the NVIDIA Jetson Nano to verify the superiority of the proposed algorithm in terms of complexity. The inference times in Table 5 demonstrate the remarkable performance of the proposed model, which required 23.11 ms for inference; that is, our model is 15.7 ms faster than the 38.81 ms elapsed time for 2D CNN inference. Even more remarkable, preprocessing every input current data value for each method

TABLE 5. Comparison of processing time between 2D CNN and TL-LED^{arc}Net.

Method	w/ preprocessing delay (ms)	w/o preprocessing delay (ms)
2D CNN	45.04	38.81
TL-LED^{arc}Net	24.22	23.11

shortens the total elapsed time, including the inference time for the proposed method, to 20.82 ms, which is 46.2% faster than the 2D CNN using the STFT preprocessing.

Consequently, TL–LED^{arc}Net outperformed the conventional 2D CNN method because the proposed approach used a lightweight 1D CNN as a preprocessor as part of the TL technique and applied an additional one-layer LSTM to improve the training accuracy.

B. ONLINE ARC-FAULT DETECTION EXPERIMENTAL RESULTS

This subsection presents the results of the online arc-fault detection experiments, which were conducted to verify the online performance of the trained TL–LED^{arc}Net. The experiments considered various initial conditions such as several normal states, the arc-gap length, and motor speed.

In this case, the ''TensorFlow Lite'' [38] open-source DL framework, which generates an optimized inference model for a given trained model, was applied to TL–LED^{arc}Net to perform real-time inference on the NVIDIA Jetson Nano. The intelligent arc-fault detection module generated the test samples using preprocessing and inferred time-domain current data collected within a 10-ms window interval in real time. The prediction result using the proposed inference model was output as a 3.3-V level signal from the general-purpose input/output (GPIO) of the NVIDA Jetson Nano.

First, to perform an actual environmental experiment using the intelligent arc-fault detection module, the following three normal states in PV systems were considered. 1) Turn-on state: the grid-connected inverter turns on owing to the increase in solar radiation; 2) turn-off state: the grid-connected inverter turns off owing to the reduction in solar radiation; and 3) step-up and 4) step-down states: rapid fluctuations in the input current because of changes in the solar radiation (e.g., owing to shading), detected using commercial inverter-controlled maximum power point tracking [39].

Fig. 9 presents the results of real-time verification with TL–LED^{arc}Net for the above-mentioned four representative normal state conditions. As shown in Fig. 9(a) and (b), neither of the E_{Low} and E_{Stable} arc-fault detection signals were activated, even though the input current either increased gradually or decreased drastically according to the grid-connected inverter operation. Similarly, in the experiments shown in Fig. 9(c) and (d), TL–LED^{arc}Net successfully classified the observed input current of the inverter as the E_{Normal} class despite rapid fluctuations.

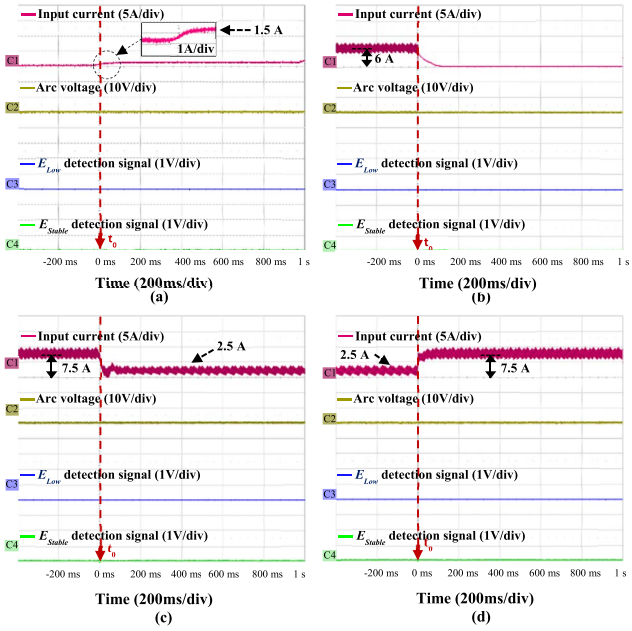

FIGURE 9. Results of online detection experiments for E_{Normal}: (a) turn-on, (b) turn-off, (c) step-down, (d) step-up states.

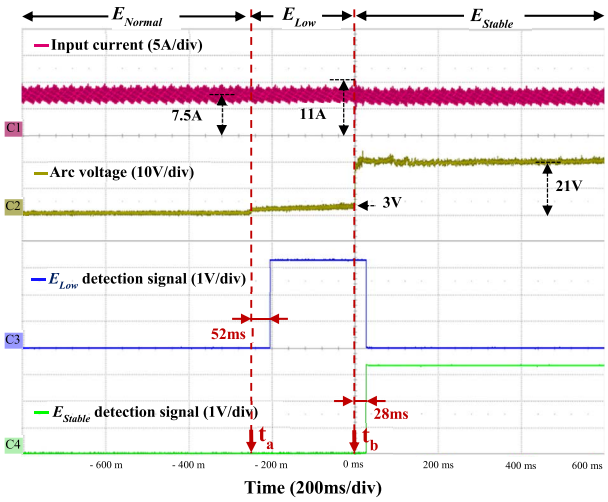

FIGURE 10. First set of online experimental results for arc-fault detection. The arc-fault gap length and control motor speed were 0.5 mm and 0.8 mm/s, respectively.

Fig. 10 and 11 present the real-time arc-fault detection performance of TL–LED^{arc}Net corresponding to the two representative experimental conditions for actual arc-fault currents. The arc-fault detection experiment was first performed with an arc-fault gap length of 0.5 mm and a control motor speed of 0.8 mm/s, as shown in Fig. 10. As a result, the E_{Low} state with the arc voltage of 3 V was observed from time t_a to t_b, where the average energy value using (1) was approximately 23 J. The arc voltage of 21 V and input current of 11 A, corresponding to 162 J, were then observed at time t_b in the E_{Stable} state. The online experimental result

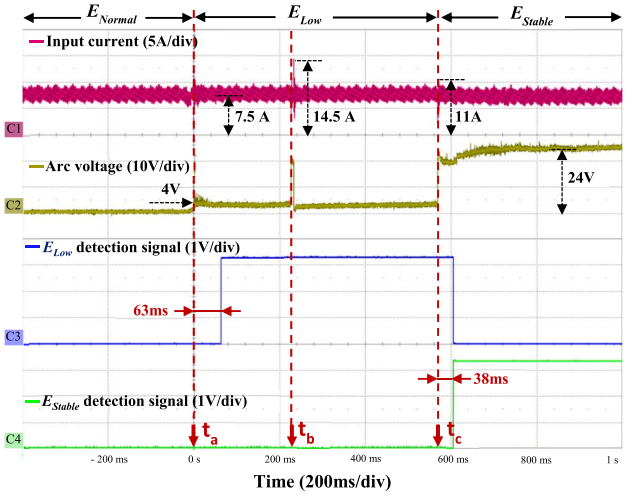

FIGURE 11. Second set of online experimental results for arc-fault detection. The arc-fault gap length and control motor speed were 0.8 mm and 0.8 mm/s, respectively.

indicates that the proposed lightweight inference model can successfully perform fast arc-fault detection within tens of ms by distinguishing the E_{Low} and E_{Stable} states.

The results of the second online arc-fault detection experiment are presented in Fig. 11, in which the arc-fault gap length was 0.8 mm, and the control motor speed was 0.8 mm/s. The E_{Low} state was observed at time t_a, and returned to the E_{Stable} state at time t_c via a continuous ionization process. In this experiment, the average value of the arc voltage was greater than that in the first online experiment (Fig. 10) because the arc-fault gap length was longer than that in the first case. In particular, the arc voltage increased instantaneously to 22 V at t_b and then decreased rapidly. Instantaneously, an input current of 14.5 A was observed at time t_b; this current was higher than the current of the transient phase in the E_{Stable} state (11 A). Additionally, the arc-fault energy at time t_b was 62.3 J, which fell within the common range of the category corresponding to the E_{Low} and E_{Stable} states.

Development of the arc-fault detection algorithm such that it considers only input current amplitude information would limit it to the reliable detection of the E_{Low} state at time t_a. In this work, we overcame this drawback by implementing a training architecture that can differentiate between E_{Low} and E_{Stable} by training with the energy amplitude and with the variation in the energy over time. Ultimately, TL–LED^{arc}Net successfully distinguished E_{Low}, even when a high transient input current occurred momentarily, as shown in Fig. 11.

Fundamentally, TL–LED^{arc}Net learned the regional level characteristics of the input current and trends (i.e., temporal changes in the energy) to ensure that E_{Low} can be reliably detected even in real environments. Furthermore, TL–LED^{arc}Net can detect the E_{Low} state in real time within 63 ms and the E_{Stable} state within 38 ms on a single-board computer.

VI. CONCLUSION

This study proposed TL–LED^{arc}Net, a proactive DC arc-fault detection method for PV systems using a commercial PV inverter. In this approach, the meaningful information in the frequency domain was visually extracted from the arc-fault data by applying the Grad-CAM technique. TL–LED^{arc}Net was then employed to classify series DC arc-faults according to their energy levels. In Stage 1, a lightweight 1D CNN model classifies the normal and arc-fault states by automatically extracting important non-handcrafted regional features. In Stage 2, the LSTM network coupled with the 1D CNN model implements a step-by-step training process based on TL to distinguish among three classes of states. This process involves analyzing the change in energy over time between the compression features extracted from the pretrained 1D CNN model of Stage 1. In an offline environment, TL–LED^{arc}Net exhibited a mean overall arc-fault detection accuracy of 95.8%, considerably higher than that of the existing methods considered in previous studies. Lastly, an online experiment verified that TL–LED^{arc}Net was able to classify the arc-fault signal in real-time within 63 ms on a single-board computer. These results conclusively demonstrated the capability of TL–LED^{arc}Net to detect a low-energy arc-fault state with high accuracy as well as a stable arc-fault state based on the energy characteristics of each type of arc-fault. This new approach is expected to improve the stability and reliability of PV systems from an energy perspective and is expected to find application in prognostics and health management technologies of future renewable energy systems.

ACKNOWLEDGMENT

Yoondong Sung would like to express his gratitude to his advisor, Prof. Yung Yi, who gave helpful advice and opinions to improve the quality of this paper.

REFERENCES

[1] O. Gandhi, D. S. Kumar, C. D. Rodríguez-Gallegos, and D. Srinivasan, "Review of power system impacts at high PV penetration part I: Factors limiting PV penetration," *Sol. Energy*, vol. 210, pp. 181–201, Nov. 2020, doi: 10.1016/j.solener.2020.06.097.

[2] I. Lillo-Bravo, P. González-Martínez, M. Larrañeta, and J. Guasumba-Codena, "Impact of energy losses due to failures on photovoltaic plant energy balance," *Energies*, vol. 11, no. 2, pp. 363, Feb. 2018, doi: 10.3390/en11020363.

[3] S. Bankability, "Technical risks in PV projects-report on technical risks in PV project development and PV plant operation," Sol. Bankability Project, European Union, Brussels, Belgium, 2016, pp. 1–139.

[4] G.-S. Seo, K. A. Kim, K.-C. Lee, K.-J. Lee, and B.-H. Cho, "A new DC arc fault detection method using DC system component modeling and analysis in low frequency range," in *Proc. IEEE Appl. Power Electron. Conf. Expo. (APEC)*, Charlotte, NC, USA, Mar. 2015, pp. 2438–2444.

[5] S. Lu, B. T. Phung, and D. Zhang, "A comprehensive review on DC arc faults and their diagnosis methods in photovoltaic systems," *Renew. Sustain. Energy Rev.*, vol. 89, pp. 88–98, Jun. 2018, doi: 10.1016/j.rser.2018.03.010.

[6] U. Labs, *Photovoltaic (PV) DC Arc-Fault Circuit Protection*, Standard UL 1699B, Aug. 2018.

[7] F. M. Uriarte, A. L. Gattozzi, J. D. Herbst, H. B. Estes, T. J. Hotz, A. Kwasinski, and R. E. Hebner, "A DC arc model for series faults in low voltage microgrids," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 2063–2070, Dec. 2012, doi: 10.1109/TSG.2012.2201757.

[8] B. Miedzinski, G. Wisniewski, S. N. Kharin, H. Nouri, and N. Grechanyuk, "Arc-to-glow transition approach for practical use in DC low-power, low-voltage electric grids," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 8, no. 6, pp. 932–938, Feb. 2018, doi: 10.1109/TCPMT.2018.2791480.

[9] S. Chae, J. Park, and S. Oh, "Series DC arc fault detection algorithm for DC microgrids using relative magnitude comparison," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 4, no. 4, pp. 1270–1278, Jul. 2016, doi: 10.1109/JESTPE.2016.2592186.

[10] S. Chen, X. Li, and J. Xiong, "Series arc fault identification for photovoltaic system based on time-domain and time-frequency-domain analysis," *IEEE J. Photovolt.*, vol. 7, no. 4, pp. 1105–1114, Jul. 2017, doi: 10.1109/JPHOTOV.2017.2694421.

[11] X. Yao, L. Herrera, Y. Huang, and J. Wang, "The detection of DC arc fault: Experimental study and fault recognition," in *Proc. 27th Annu. IEEE Appl. Power Electron. Conf. Expo. (APEC)*, Orlando, FL, USA, Feb. 2012, pp. 1720–1727.

[12] Z. Wang, S. McConnell, R. S. Balog, and J. Johnson, "Arc fault signal detection–Fourier transformation vs. wavelet decomposition techniques using synthesized data," in *Proc. IEEE 40th Photovoltaic Spec. Conf. (PVSC)*, Denver, CO, USA, Jun. 2014, pp. 3239–3244.

[13] S. Kumar, T. Kolekar, S. Patil, A. Bongale, K. Kotecha, A. Zaguia, and C. Prakash, "A low-cost multi-sensor data acquisition system for fault detection in fused deposition modelling," *Sensors*, vol. 22, no. 2, p. 517, Jan. 2022, doi: 10.3390/s22020517.

[14] R. D. Telford, S. Galloway, B. Stephen, and I. Elders, "Diagnosis of series DC arc faults—A machine learning approach," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1598–1609, Nov. 2017, doi: 10.1109/TII.2016.2633335.

[15] Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, and W. Paszke, "Iterative learning control for repetitive tasks with randomly varying trial lengths using successive projection," *Int. J. Adapt. Control Signal Process.*, vol. 36, no. 5, pp. 1196–1215, Feb. 2022, doi: 10.1002/acs.3396.

[16] Y. Jiang, W. Gao, J. Na, D. Zhang, T. T. Hämäläinen, V. Stojanovic, and F. L. Lewis, "Value iteration and adaptive optimal output regulation with assured convergence rate," *Control Eng. Pract.*, vol. 121, Apr. 2022, Art. no. 105042, doi: 10.1016/j.conengprac.2021.105042.

[17] X. Zhang, H. Wang, V. Stojanovic, P. Cheng, S. He, X. Luan, and F. Liu, "Asynchronous fault detection for interval type-2 fuzzy nonhomogeneous higher level Markov jump systems with uncertain transition probabilities," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 7, pp. 2487–2499, Jul. 2022, doi: 10.1109/TFUZZ.2021.3086224.

[18] Y. Wang, F. Zhang, X. Zhang, and S. Zhang, "Series AC arc fault detection method based on hybrid time and frequency analysis and fully connected neural network," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6210–6219, Dec. 2019, doi: 10.1109/TII.2018.2885945.

[19] V. Le, X. Yao, C. Miller, and B.-H. Tsao, "Series DC arc fault detection based on ensemble machine learning," *IEEE Trans. Power Electron.*, vol. 35, no. 8, pp. 7826–7839, Jan. 2020, doi: 10.1109/TPEL.2020.2969561.

[20] J. Jiang, Z. Wen, M. Zhao, Y. Bie, C. Li, M. Tan, and C. Zhang, "Series arc detection and complex load recognition based on principal component analysis and support vector machine," *IEEE Access*, vol. 7, pp. 47221–47229, 2019, doi: 10.1109/ACCESS.2019.2905358.

[21] N. Qu, J. Zuo, J. Chen, and Z. Li, "Series arc fault detection of indoor power distribution system based on LVQ-NN and PSO-SVM," *IEEE Access*, vol. 7, pp. 184020–184028, 2019, doi: 10.1109/ACCESS.2019.2960512.

[22] K. Xia, S. He, Y. Tan, Q. Jiang, J. Xu, and W. Yu, "Wavelet packet and support vector machine analysis of series DC ARC fault detection in photovoltaic system," *IEEJ Trans. Elect. Electron. Eng.*, vol. 14, no. 2, pp. 192–200, Feb. 2019, doi: 10.1002/tee.22797.

[23] K. Yang, R. Chu, R. Zhang, J. Xiao, and R. Tu, "A novel methodology for series arc fault detection by temporal domain visualization and convolutional neural network," *Sensors*, vol. 20, no. 1, p. 162, Dec. 2019, doi: 10.3390/s20010162.

[24] T. Li, Z. Jiao, L. Wang, and Y. Mu, "A method of DC arc detection in all-electric aircraft," *Energies*, vol. 13, no. 16, p. 4190, Aug. 2020, doi: 10.3390/en13164190.

[25] S. Lu, T. Sirojan, B. T. Phung, D. Zhang, and E. Ambikairajah, "DA-DCGAN: An effective methodology for DC series arc fault diagnosis in photovoltaic systems," *IEEE Access*, vol. 7, pp. 45831–45840, 2019, doi: 10.1109/ACCESS.2019.2909267.

[26] Y.-L. Shen and R.-J. Wai, "Wavelet-analysis-based singular-value-decomposition algorithm for weak arc fault detection via current amplitude normalization," *IEEE Access*, vol. 9, pp. 71535–71552, 2021, doi: 10.1109/ACCESS.2021.3077871.
[27] P. V. Kamat, R. Sugandhi, and S. Kumar, "Deep learning-based anomaly-onset aware remaining useful life estimation of bearing," *PeerJ Comput. Sci.*, vol. 7, p. e795, Nov. 2021, doi: https://doi.org/10.7717/peerj-cs.795.
[28] V. V. Terzija and Z. M. Radojevic, "Numerical algorithm for adaptive autoreclosure and protection of medium-voltage overhead lines," *IEEE Trans. Power Del.*, vol. 19, no. 2, pp. 554–559, Mar. 2004, doi: 10.1109/TPWRD.2003.823184.
[29] Z. Zhang, Y. Nie, and W.-J. Lee, "Approach of voltage characteristics modeling for medium-low-voltage arc fault in short gaps," *IEEE Trans. Ind. Appl.*, vol. 55, no. 3, pp. 2281–2289, Dec. 2019, doi: 10.1109/TIA.2018.2886751.
[30] J. C. Kim, B. Lehman, and R. Ball, "DC arc fault model superimposing multiple random arc noise states on an average model," in *Proc. 20th Workshop Control Model. Power Electron. (COMPEL)*, Toronto, ON, Canada, Jun. 2019, pp. 1–8.
[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Oct. 2020, doi: 10.1007/s11263-019-01228-7.
[32] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 807–814.
[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.
[34] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*.
[35] Y. Ma, D. Oslebo, A. Maqsood, and K. Corzine, "DC fault detection and pulsed load monitoring using wavelet transform-fed LSTM autoencoders," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 6, pp. 7078–7087, Dec. 2021, doi: 10.1109/JESTPE.2020.3019382.
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2017, *arXiv:1412.6980*.
[37] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," 2020, *arXiv:2008.05756*.
[38] (2022). *TensorFlow Lite Guide*. [Online]. Available: https://www.tensorflow.org/lite/guide
[39] K. Lappalainen and S. Valkealahti, "Experimental study of the maximum power point characteristics of partially shaded photovoltaic strings," *Appl. Energy*, vol. 301, Nov. 2021, Art. no. 117436, doi: 10.1016/j.apenergy.2021.117436.

YOONDONG SUNG (Member, IEEE) received the M.S. degree in information and communication engineering from the Gwangju Institute of Science and Technology, Gwangju, Republic of Korea, in 2012. He is currently pursuing the Ph.D. degree with the Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea. Since 2012, has been a Senior Researcher with the Korea Institute of Energy Research. His current research interests include data driven-based fault diagnosis, artificial intelligence applications for renewable energy, and energy management systems.

GIHWAN YOON received the B.S. and M.S. degrees in electrical engineering from Chonbuk National University, Jeonju, South Korea, in 2013 and 2015, respectively. Since 2015, he has been a Researcher with the Korea Institute of Energy Research, Daejeon, Republic of Korea. His research interests include renewable energy, distributed generation, power system reliability, and energy management systems.

JI-HOON BAE (Member, IEEE) received the B.S. degree in electronic engineering from Kyungpook National University, Daegu, Republic of Korea, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from the Pohang University of Science and Technology, Pohang, Gyeongbuk, Republic of Korea, in 2002 and 2016, respectively. From 2002 to 2019, he was a Principal Researcher with the Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea. He joined the Faculty of the Department of AI and Big Data Engineering, Daegu Catholic University, Gyeongsan, Gyeongbuk, in 2019, where he is currently an Assistant Professor. His research interests include deep learning, transfer learning, radar imaging, radar signal processing, and optimized techniques. He is a member of the Korea Institute of Information Technology and the Korea Institute of Electromagnetic Engineering and Science.

SUYONG CHAE (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, in 1998 and 2000, respectively, and the Ph.D. degree from Seoul National University, Seoul, Republic of Korea, in 2009. From 2000 to 2010, he was a Research Engineer with Samsung SDI. From 2010 to 2022, he was a Principle Researcher at the Korea Institute of Energy Research. In 2022, he accepted a position as a Faculty Member at the Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, South Korea, where he is employed as an Associate Professor. His research interests include power electronics and modeling and control of distributed power systems.