Robust Open-Switch Fault Diagnosis of Three-Level NPC Inverters Based on Data Augmentation With White Noise Injection

Jiwon Jung, *Member, IEEE*, Dyan Puspita Apsari, *Graduate Student Member, IEEE*, and Dong-Choon Lee, *Fellow, IEEE*

***Abstract*—This article proposes a novel real-time fault diagnosis approach for three-level neutral-point-clamped inverters based on a one-dimensional (1-D) convolutional neural network (CNN). The proposed method incorporates data augmentation into simulation data, enhancing the generalization capabilities of deep learning models. This allows fault diagnostic models to have high robustness even in untrained system conditions. In such scenarios, the application of 1-D CNN models with data augmentation surpasses the performance of the same models without the incorporation of white noise, resulting in accuracy improvements of up to 1.71%. Furthermore, deep learning models trained on simulation data with data augmentation give a better performance when compared to those trained using experiment data. The proposed method has been verified through simulation with offline testing and experimentation with real-time deep learning algorithms.**

***Index Terms*—Convolutional neural network (CNN), data augmentation, fault diagnosis, three-level NPC inverter, white noise injection.**

I. Introduction

According to a recent research report, the failure rate of switching power devices within power conversion systems constitutes approximately 31% of the total failures [1]. This figure surpasses the failure rates observed in other components. Moreover, it is critical to develop an efficient and robust fault diagnosis to enhance the reliability of power electronic converter systems.

There are two types of faults in power devices: short-circuit and open-circuit. In the short-circuit fault tremendous current will flow instantaneously and cause permanent damage to the circuit within an extremely short time. To avoid damage to the system, hardware protection circuits are used to block the gating signal immediately [2]. However, open-circuit faults of the inverter do not cause immediate damage to the circuits, but they can gradually degrade performance over time, potentially leading to system malfunctions. Therefore, it is imperative to promptly identify open-circuit faults. There have been numerous fault diagnosis methods introduced and applied to inverter applications [3], [4], [5]. However, in recent years, multilevel inverters have gained attention, especially in high-power applications due to their ability to provide better output waveform quality compared to the two-level inverters [6]. In practice, the three-level neutral-point-clamped inverter (3L-NPC) is a widely used multilevel inverter in a variety of industrial applications [7], [8], [9]. From a fault analysis standpoint, it is important to note that there is an elevated potential for switch faults attributed to the larger number of switching devices.

For the 3L-NPC inverter topology, numerous methodologies have been developed to diagnose switch faults. Model-based fault diagnosis methods have been introduced in [2], [10], [11], and [12], which exhibit limited robustness under diverse system conditions due to their reliance on mathematical modeling and threshold settings, thereby introducing complexities into the implementation. Another approach applied involves the use of highly accurate sensors [13]. Nevertheless, the sensor-based method comes at the price of increased cost and decreased system reliability due to the additional hardware components. Due to these considerations, artificial intelligence (AI) based methodologies have emerged to diagnose system faults. Existing AI-based fault diagnosis techniques involve the utilization of machine learning or artificial neural networks, as detailed in [14], [15], [16], and [17]. While these techniques are computationally less intensive than deep learning approaches, they do require the use of additional preprocessing techniques to extract features from the data, which increases the computational time of the diagnosis.

Recently, there has been a study of fault diagnosis employing one of deep learning-based architecture, which is convolutional neural networks (CNNs). Moreover, a notable advantage of this architecture is its ability to eliminate the necessity for preprocessing methods [18], [19], [20]. In [19], open-circuit faults in hybrid active neutral-point clamped (HANPC) inverters have been diagnosed using two-dimensional (2-D) CNN. However,

Received 2 August 2024; revised 30 September 2024; accepted 17 October 2024. Date of publication 31 October 2024; date of current version 18 December 2024. This work was supported by the National Research Foundation of Korea funded by the Korean Government [Ministry of Science and ICT (MSIT)] under Grant 2022R1A4A1031885. Recommended for publication by Associate Editor Z. Chen. *(Corresponding author: Dong-Choon Lee.)*

Jiwon Jung is with the Hanwha Systems Company, Ltd., Daegu 42709, South Korea (e-mail: zxcv6210@ynu.ac.kr).

Dyan Puspita Apsari and Dong-Choon Lee are with the Department of Electrical Engineering, Yeungnam University, Gyeongbuk 38541, South Korea (e-mail: dyanpuspita@yu.ac.kr; dclee@yu.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2024.3488094.

Digital Object Identifier 10.1109/TPEL.2024.3488094

2-D CNN is used to classify faults from image data features rather than time series features. Since the size of 2-D data is significantly larger compared to that of 1-D data, which makes 2-D CNN hard to be implemented in real-time with low computational power. Yuan et al. [20] proposed a fault diagnosis method for 3L-NPC inverters based on 1-D CNN, where an AdaMod as an optimization method for training in 1-D CNN was suggested [21].

In fault diagnosis applications using AI, class imbalance presents a significant challenge. This imbalance can bias the model towards the majority class, reducing its sensitivity to detecting faults [22], [23]. To address this issue, specialized techniques are employed, including both data-level methods such as oversampling and undersampling to rebalance the class distribution and algorithm-level methods that modify learning algorithms to give more weight to minority class instances. These strategies aim to improve the sensitivity of abnormal detection models to minority class instances, thus enhancing their overall performance in class imbalance. The use of generative adversarial networks (GANs) addresses sample imbalances and improves classification accuracy by generating new, synthetic samples that resemble the minority class, as proposed in [23].

Another significant challenge in fault diagnosis is the insufficient availability of labeled fault samples. However, obtaining labeled abnormal samples can be challenging and expensive, limiting the effectiveness of supervised learning approaches. In practical applications, acquiring a comprehensive set of fault samples is difficult. To address this, using finite element method (FEM) simulations to generate synthetic samples has been proposed [23], [24], [25]. Augmenting the dataset enhances accuracy and versatility by enabling deep learning models to learn more comprehensively. However, physically increasing the dataset size, especially through experiments under diverse conditions, is constrained due to the expense and time involved. To overcome these limitations, data augmentation methods are employed, increasing the dataset size by applying various algorithms. The various types, characteristics, and algorithms of data augmentation methods applied to time series data have been described in [26] and [27].

Moreover, in practical scenarios, noise influences the effectiveness of fault diagnosis. Noisy data can degrade the performance of AI models, leading to false positives or false negatives. To address this, simulation data with noise helps replicate the inherent variability and unpredictability found in actual datasets [22]. One common technique for introducing noise is to add random values to the simulated data points. This can be done by generating random values from a specified distribution and incorporating them into the simulation data. The magnitude and distribution of the noise can be adjusted to reflect the characteristics observed in experiment data.

In this article, a novel real-time fault diagnosis for a 3L-NPC inverter incorporating a 1-D CNN and data augmentation is proposed, which is extended from the previous work [28]. The proposed technique involves a training strategy that incorporates the injection of white noise into the output current. The modified CNN algorithm adopts a dilated CNN, global average pooling layer, and ResNet. The objective of these modifications is to generate an accurate model that can be applied over a broad frequency range without increasing the number of parameters, emphasizing the importance of the computational efficiency process in achieving real-time operational demands in diagnostic methods. Furthermore, the inclusion of white noise injection into the simulation data during model training enhances its generalization capabilities. This improvement enables accurate fault diagnosis even in untrained system conditions. Moreover, it is noteworthy that the model is solely trained using simulation data, simplifying the process of data extraction. Simulation and experimental results validate the feasibility and robustness of the proposed method, showcasing its effectiveness in diverse conditions, including different load characteristics and fault occurrence timings.

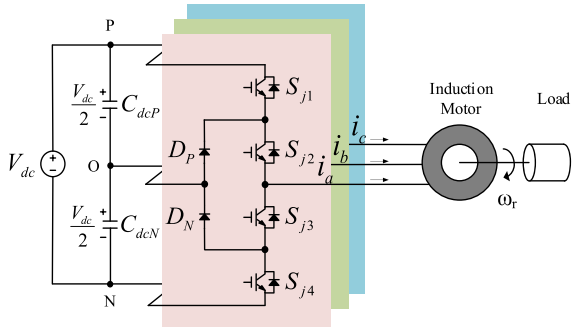

Fig. 1. Three-level NPC inverter topology.

The rest of this article is organized as follows. In the introduction, an overview of the background, and basic concepts of artificial intelligence and inverters, with their respective applications have been described. In Section II, the inverter topology considered in this study, including an open-circuit fault, is analyzed. In Section III, the framework of the 1-D CNN, involving data acquisition, preprocessing, and the proposed modified algorithm, is outlined. Section IV delves into the implementation of the proposed data augmentation method with white noise injection. The subsequent Sections V and VI present the simulation and experimental results along with real-time implementation using CAN communication. Finally, Section VII concludes this article.

II. 3L-NPC Inverter Topology and Fault Analysis

A. Topology and Operating Principle

The configuration of a three-phase 3L-NPC inverter is shown in Fig. 1, which consists of six clamping diodes, twelve switches, and two dc-link capacitors. Note that j symbolizes phases (a, b, and c). The pole voltage has three switching states, denoted as "P," "O," and "N". In the "P" state, switches S_{j1} and S_{j2} are turned ON. In the "O" state, the current flows through S_{j2} and S_{j3}, resulting in zero current flow. In the "N" state, switches S_{j3} and S_{j4} are turned ON. The current paths in the 3L-NPC inverter are shown in Fig. 2 and the switching states are listed in Table I. The operating principle of the 3L-NPC inverter has been described well in [7], [8], [9]. Fig. 3 shows the balancing control scheme of the dc-link capacitor voltages, where a simple P controller is used [9].

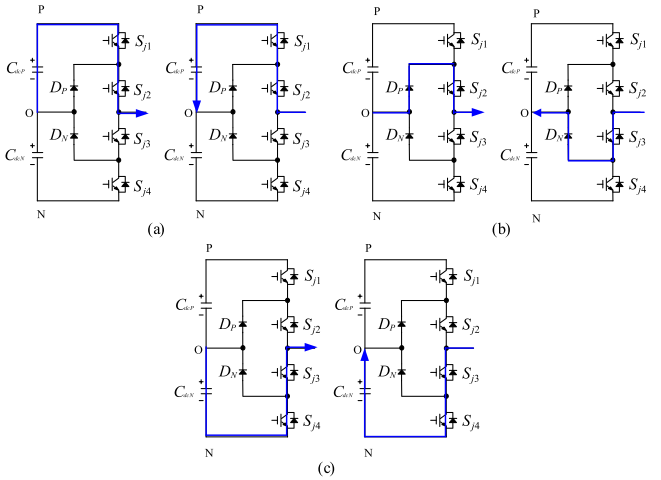

Fig. 2. Conduction paths according to switching states for 3L-NPC inverter. (a) P. (b) O. (c) N.

TABLE I
SWITCHING STATES OF 3L-NPC INVERTER

State	Switching state				Pole voltage
	S_{j1}	S_{j2}	S_{j3}	S_{j4}	
P	1	1	0	0	$+V_{dc}$
O	0	1	1	0	0
N	0	0	1	1	$-V_{dc}$

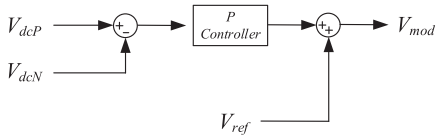

Fig. 3. DC-link balancing control.

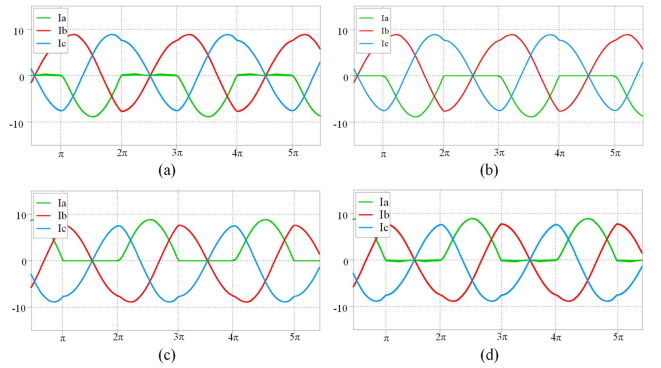

Fig. 4. Output current waveform in open-circuit faults of switches with gain $K_P = 50$. (a) S_{a1}. (b) S_{a2}. (c) S_{a3}. (d) S_{a4}.

B. Open-Circuit Fault Analysis

As the 3L-NPC inverter has four switches per phase, there can exist twelve single-switch open-circuit faults in the three-phase system. Figs. 4 and 5 show the three-phase current under the open-circuit fault of each switch and the conduction paths in the cases of open-circuit fault for the 3L-NPC inverter, respectively.

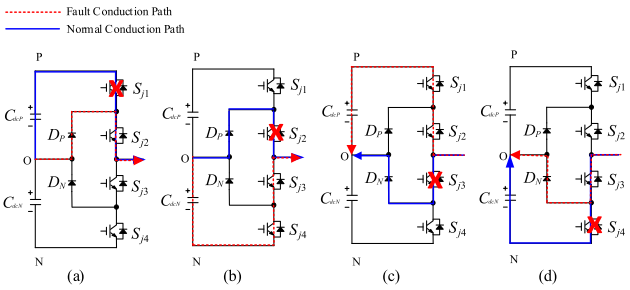

Fig. 5. Conduction paths in faulty state of (a) S_{j1} (+P). (b) S_{j2} (+O). (c) S_{j3} (-O). (d) S_{j4} (-N).

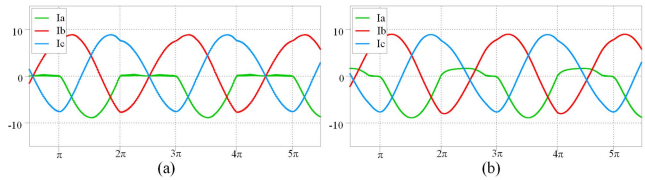

Fig. 6. Output current waveform in open-circuit fault of S_{a1} with different gains of DC-link capacitor balancing controller. (a) $K_P = 50$. (b) $K_P = 2$.

During either the positive or negative half cycles, the occurrence of a fault in S_{j1} or S_{j4} leads to a reduction in the output current below the normal. This is due to the inability to form the "P" or "N" state. In contrast, when a fault in S_{j2} or S_{j3} occurs, the output current does not flow in one of the half cycles (positive or negative). This occurs because not only is the "P" or "N" state not formed, but also the midpoint of the dc link (D_P or D_N) cannot be formed [14]. When an open-circuit fault occurs at S_{j1} or S_{j4}, the magnitude of the associated phase current decreases depending on the dc-link capacitor voltage balancing control gains.

Fig. 6 shows the output current waveform in the occurrence of an open-circuit fault of S_{a1} with different dc-link voltage balancing control gains. Fig. 6(a) and (b) illustrates the output current waveforms with proportional (K_P) gains of 50 and 2 for the balancing controller, respectively. When the gain is low, the drop in the output current is less significant than the case with a high gain where the output current shows distinct values in the half-cycle wave.

III. ONE-DIMENSIONAL CNN

A. Data Acquisition and Preprocessing

The three-phase 3L-NPC inverter has one normal state and twelve single-switch open-circuit fault states. Therefore, a deep learning model for fault diagnosis has 13 classifiers as the output. For data acquisition, current data for the deep learning classifier is obtained as a CSV file through simulation using PLECS software. The three output currents are extracted in a 1-D time series data. The recording time is double the cycle of the current waveform, with a sampling frequency of 10 kHz. As shown in Fig. 4, when open-circuit faults occur, the phase current exhibits a periodic characteristic. These fault features reveal a recurring pattern over time, making them suitable as

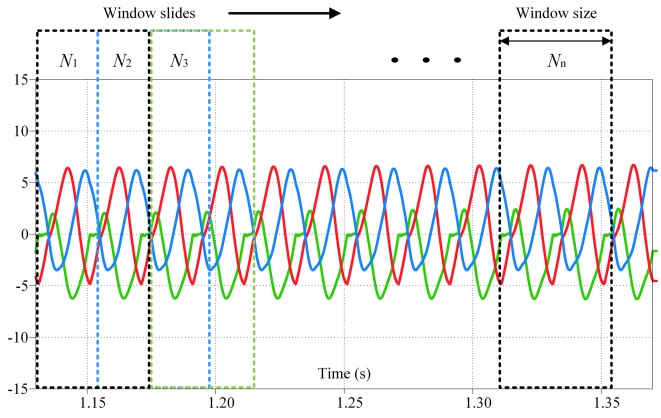

Fig. 7. Window slicing method.

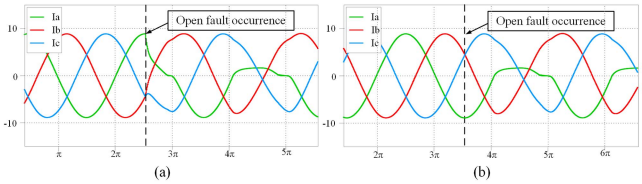

Fig. 8. Output current waveform in a transient state of S_{a1} fault. (a) Fault occurring at $\theta = 5\pi/2$. (b) Fault occurring at $\theta = 7\pi/2$.

input data for the CNN model. To train these features, the data is divided using a window slicing method as preprocessing time series data. Using this method, phase current data with a two-cycle length is sliced with a one-cycle size window. As a result, the data of the output current is in the form of a 3x(window size) matrix when considering the three-phase channels. Fig. 7 depicts the window-slicing method. When window slicing or data segmentation is applied, the necessity to synchronize the beginning point of the periodic data is eliminated.

An open-circuit fault produces different shapes of transient waveforms depending on the instant of fault occurrence within a cycle. To develop a fault diagnosis model with high accuracy, faults at various instants are investigated in the cycle regardless of the fault occurrence. Specifically, fault scenarios are introduced at intervals of one-tenth of the fundamental cycle to ensure the model is trained on a variety of transient states. In Fig. 8(a) and (b), the output current waveforms during the transient state of the S_{a1} fault are shown for two specific fault cases: one at the positive peak of the cycle ($\theta = 5\pi/2$) and another at the negative peak of the cycle ($\theta = 7\pi/2$). These differences in waveforms occur since the timing of the fault affects the shape of the transient response. Such variations can result in reduced diagnostic accuracy or longer detection times, which is why training on faults occurring at multiple points in a cycle was critical for improving the performance of the model.

For data acquisition in the simulation, the proposed system uses 12 (fault cases) x 41 (windows slicing) x 21 (transient cases) = 10 332 fault data and 362 normal state data. Since the normal state does not contain a transient state, a large amount of normal state is not required. Deep learning methods can provide high accuracy even without the need for feature extraction. However,

TABLE II
ACCURACY WITH DIFFERENT FREQUENCY CONDITIONS

50 Hz trained model	
Frequency of data	Accuracy [%]
30 Hz	80.95
40 Hz	86.64
70 Hz	100
100 Hz	100

normalization can be used as a data preprocessing that enhances model generalization without computationally intensive operations, unlike fast fourier transform (FFT) or discrete wavelet transform (DWT). The normalization process scales data in the range between [-1, 1] to generate a robust deep-learning model that is independent of variations in the current level and preserves polarity information. In the absence of this process, deep learning models necessitate training that includes varying magnitudes of current data.

B. Modified 1-D CNN Algorithm

The implementation of a deep-learning model, specifically designed for diagnosing open-circuit faults over a wide range of frequencies is vital to ensuring the reliable operation of an inverter. Based on Table II, the accuracy of a model trained with 50 Hz data across various frequencies by validating it with data at 30 Hz, 40 Hz, 70 Hz, and 100 Hz. This approach aims to understand the generalization across a frequency spectrum, revealing accuracy differences due to the feature generalization limitations at the training frequency. The results show how well the model diagnoses open-circuit faults under these conditions. For instance, using 200 data points as input to represent a 50 Hz frequency at a 10 kHz sampling rate highlights the limitations when faults occur at lower frequencies, as the increased data points per cycle introduce periodic properties not effectively learned by the model. Conversely, using 1000 data points to represent a 10 Hz frequency at the same sampling rate allows the model to learn from multiple cycles, capturing periodic properties even when the frequency increases. This indicates that training solely at high frequencies may hinder accurate fault detection across the entire frequency spectrum.

When the periodic properties of the data are excluded from the input of the CNNand the model is solely trained under high-frequency conditions, there is a degradation in accuracy, as shown in Table II. Therefore, by utilizing a low-frequency training dataset for the model, the fault diagnosis model becomes applicable through a broader frequency spectrum. However, employing low frequencies presents a challenge, as it results in an increased number of model parameters, contrasting with training with high frequencies.

To address the limitations of conventional CNN size, which depends on the input size, a modified CNN architecture is proposed. A comparison between the LeNet-based conventional CNN and the proposed CNN architectures can be seen, as shown in Fig. 9. The modified CNN architecture differs from conventional CNN ones in its approach to feature extraction in

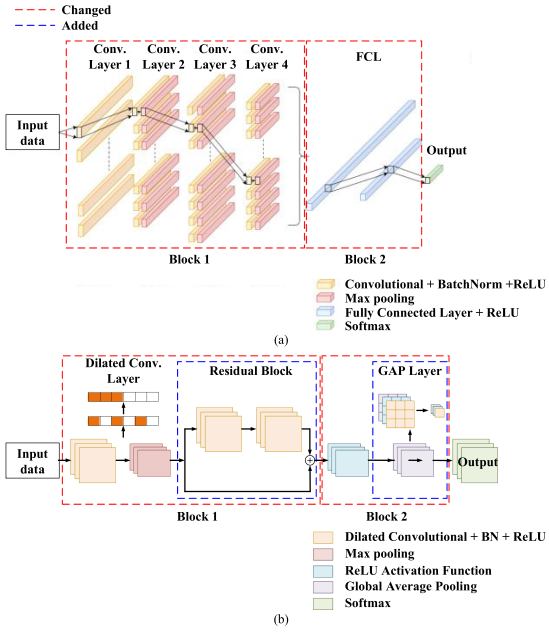

Fig. 9. Model architecture with (a) conventional CNN. (b) Proposed CNN.

block 1. Conventional architectures typically rely on multiple convolution and pooling layers to process data. In contrast, the modified architecture presented in this work integrates dilated convolution within a single convolution layer [29], [30]. This strategic modification enhances performance by expanding the receptive field of the convolutional layer. It also includes a data-skipping operation to reduce the number of parameters. In addition, instead of the usual sequential arrangement of convolution and pooling layers, residual blocks or ResNet are utilized [31]. These blocks feature two 3×3 convolutional layers followed by an activation layer, effectively addressing the vanishing gradient problem. Moreover, in the classification of faulty switches, the traditional fully-connected layer (FCL) is replaced by global average pooling (GAP) [32]. This modification enables the model to determine the number of parameters independently of the input length.

Setting the model and parameters for the modified 1-D CNN involves a process and optimization using a trial-and-error method. The architecture is designed to balance computational efficiency with model performance, including decisions on the number and size of convolutional layers, the dilation rate in dilated convolution, and the number of residual blocks. The proposed model, as compared to its conventional CNN counterpart on the same input conditions, is characterized by a reduction in parameters and, thus, leads to smaller models. A detailed comparison of the parameter numbers and sizes for both the conventional and proposed CNN architectures is presented in Table III. Changes in input parameters over different frequencies, particularly when frequencies are reduced, increase the number of parameters and an increase in model size as demonstrated by conventional CNN models. However, the number of parameters in the proposed model is unaffected by the variability of the input data. As a result, regardless of changes in the input

TABLE III
PARAMETERS COMPARISON OF CNN ARCHITECTURE

Conventional CNN model		
Input data frequency	Number of parameters	Model size
50 Hz	140 933	0.98 MB
40 Hz	170 933	1.20 MB
30 Hz	220 933	1.59 MB
10 Hz	640 933	4.74 MB
Proposed CNN model		
Input data frequency	Number of parameters	Model size
50 Hz	7213	0.16 MB
40 Hz	7213	0.19 MB
30 Hz	7213	0.25 MB
10 Hz	7213	0.70 MB

data, the size of the deep learning model remains relatively constant, reducing the size of the deep learning model by 85%.

Based on Table III and Fig. 9, the reduction in the number of parameters achieved by the proposed CNN model represents a significant advancement over the conventional CNN. This reduction is primarily due to two key modifications in the network architecture. First, the conventional CNN employs a standard CNN layer and a fully connected layer after flattening, leading to a substantial number of parameters. In contrast, the proposed method replaces the standard CNN layer with a dilated CNN layer, which expands the receptive field without increasing the number of parameters. In addition, the proposed model utilizes a GAP layer instead of flattening followed by a dense layer, drastically reducing the number of parameters. These changes lead to a substantial reduction in the number of parameters, from hundreds of thousands in the conventional method to just hundreds in the proposed method, without compromising feature extraction and classification performance.

IV. PROPOSED DATA AUGMENTATION METHOD

Data augmentation comprises a wide range of methods that can be divided into different categories. These classifications range from traditional approaches that require user intervention to manipulate data values to more sophisticated techniques that use generative models for augmentation. Traditional methods typically involve simple operations such as flipping, cropping, rotation, shifting, or noise injection. Advanced techniques, on the other hand, use generative models to generate synthetic data points that adhere to the underlying statistical characteristics of the original dataset. In addition, GANs have shown significant efficacy in simulating realistic data [33]. Since time series data has frequency and time domain features unlike image data, it is necessary to understand the system to properly apply the data augmentation method.

Conventional data augmentation methods, although somewhat effective, exhibit limitations as they may not fully encapsulate the diverse range of data variations and patterns. In contrast, advanced data augmentation approaches offer a more extensive scope for augmentation by generating synthetic data samples;

however, these methods often demand substantial computing power. In light of these considerations, the white noise injection method emerges as a practical and effective alternative. Notably, compared to generative models, the additive white Gaussian noise (AWGN) or Gaussian white noise injection method is simpler to implement. Moreover, it can effectively generate datasets, and improve model generalization without computational expense [34].

A. Concept of White Noise and THD+N

The output currents of pulse width modulation (PWM) inverters indeed encompass various types of noise, particularly in ac systems with fixed frequencies. These noises stem from a multitude of sources, including switching actions, sensors, thermal effects, and load characteristics. To provide a robust fault diagnosis model against noises, Gaussian white noise is added to the current data in simulation. Therefore, integrating white noise can produce practical noisy data while exerting minimal impact on the original characteristics of the data.

The noise existing in the inverter output current poses challenges for interpretation, given its varying characteristics based on the load and system conditions. Rather than analyzing and incorporating various types of noise, a straightforward method is adopted by creating a noise band within the current using white noise spanning the entire frequency spectrum. This injection of Gaussian white noise effectively broadens the bandwidth (variance) of the data, which indicates how much random noise information is contained in the data. The current data bandwidth is determined with reference to total harmonic distortion (THD). The THD is defined as

$$\text{THD} = \frac{\sqrt{I_2^2 + I_3^2 + I_4^2 \cdots}}{I_1} \tag{1}$$

where I_1 and I_n denote the root-mean-square (rms) values of the fundamental and nth-order harmonic components of phase current.

The presence of injected Gaussian white noise can potentially influence the THD of the output current. However, calculating the THD is challenging since it introduces stochastic elements, making traditional harmonic analysis methods inapplicable. The inherent variability and nonperiodic nature of Gaussian white noise require specialized techniques for accurately characterizing its harmonic content. Hence, the total harmonic distortion plus noise (THD+N) approach is employed to facilitate the calculation of THD in noisy environments [35]. This concept is extended by incorporating the rms amplitude with additional noise (N), aiming to attain a specific desired distortion level. The THD+N is defined as

$$\text{THD} + \text{N} = \frac{\sqrt{I_2^2 + I_3^2 + I_4^2 \cdots + I_{\text{noise}}^2}}{I_1} \tag{2}$$

where I_{noise} denotes the rms value of the noise components added to the measured phase current.

This method allows for a more robust assessment of harmonic distortion by considering both the inherent noise and harmonic components, providing a comprehensive evaluation of the signal quality in the presence of injected white noise [36]. The adjustment of THD+N is achieved by manipulating the standard deviation of Gaussian white noise, thereby enabling precise control over the levels of distortion and noise within the signal. The variance of the random variable can be defined as follows:

$$\sigma^2 = E((X - \mu)^2) \tag{3}$$

where σ is a standard deviation, X is the particular random variable, E is the expected value of X, and μ is the average of X.

In this context, variance (σ^2) serves as a metric to gauge the spread or dispersion of a random variable X, capturing its deviations from the mean value (μ). Higher variance indicates greater dispersion of data points, signifying increased variability within the dataset. Conversely, lower variance suggests data points are closer to the mean, indicating reduced variability. For Gaussian white noise, each element in the vectors is statistically independent, ensuring that all variables have a mean of zero and equal variance. Hence, the rms value of Gaussian white noise is equivalent to the standard deviation, as follows [37]:

$$\text{rms} = \sqrt{E(X^2)} = \sqrt{\sigma^2} = \sigma. \tag{4}$$

The square root of the variance, known as the standard deviation (σ), serves as a measure of the amplitude within the noise distribution, aligning seamlessly with the rms amplitude of the noise. This intrinsic relationship highlights the pivotal role of standard deviation in assessing the dispersion in Gaussian white noise, aiding in understanding the distribution and variability of the dataset. As a result, the standard deviation serves as a direct reflection of the injected white noise (I_{noise}). Determining the σ value of white noise is necessary to alter the THD of the original waveform to the desired THD+N level. This calculation can be obtained from (1) and (2), which is given by

$$\sigma = I_{\text{noise}} = \sqrt{(\text{THD} + \text{N})^2 - (\text{THD})^2} \cdot I_1. \tag{5}$$

In the proposed method, the selection of I_{noise} is important. Excessive noise injection may lead to the loss of original data features, thus diminishing the performance of the deep learning model.

B. Proposed White Noise Injection

The data augmentation method is applied under consideration of the specific characteristics and operation of the system to ensure that the augmented data accurately reflects practical conditions. In the following, the output current features of the open-circuit fault in the 3L-NPC inverter are described, which has been discussed previously in Section II-B. The current under open-circuit fault in S_{j2} or S_{j3} does not flow in one of the half cycles of positive or negative directions. As a result, limiters are used to block white noise. In the case of the S_{j1} or S_{j4} fault cases, only the magnitude of the output current decreases, thus eliminating the limiter.

Unpredictable noises in experiment data are challenging to simulate due to the variability of hardware and operation environment. White noise, with its constant spectral density, is used in simulations to approximate measurement noise. This simplifies calculations and facilitates the analysis of system

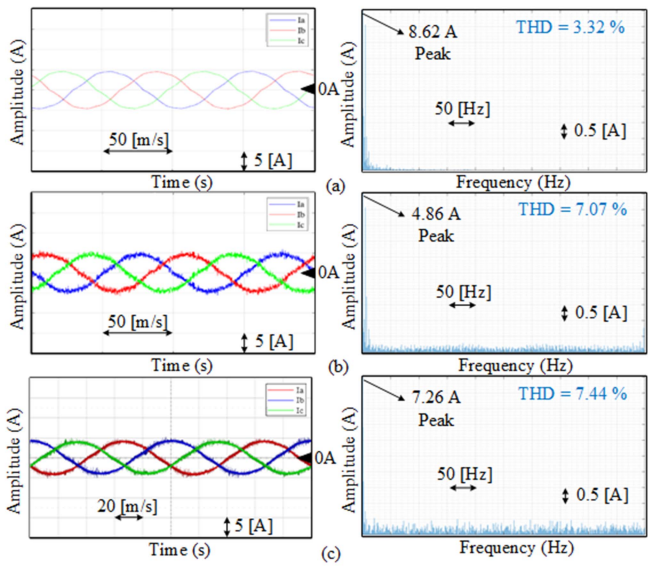

Fig. 10. Waveform of phase current and FFT in 1 N·m, 300 r/min. (a) Simulation data without injection. (b) Simulation data with white noise injection (THD+N 3 %). (c) Experiment data.

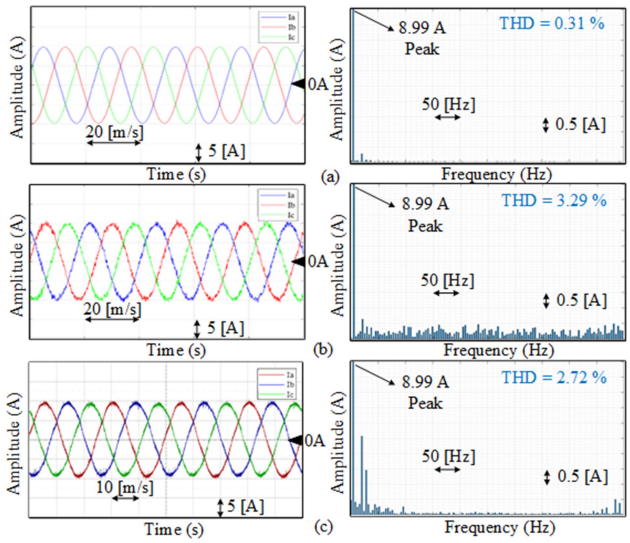

Fig. 11. Waveform of phase current and FFT in 10 N·m, 1200 r/min. (a) Simulation data without injection. (b) Simulation data with white noise injection (THD+N 3 %). (c) Experiment data.

behavior under noisy conditions. In this study, white noise is utilized by quantizing the THD levels. This method allows the deep learning model to better understand the features of ripples and harmonics at various frequencies, considering that noise can arise at any frequency. Incorporating random noise complicates the deep-learning model to memorize the training samples, as it alters the THD bands that characterize the real waveform. Various THD+N values are selected to enhance the robustness of the training data generated from simulations under different conditions.

Figs. 10 and 11 illustrate the effect of injecting white noise into simulation data. Simulation data with low THD levels inherently lack noise components, making them less suitable for real-time training. However, by injecting white noise into these simulations, a random frequency waveform is introduced that raises THD levels. This approach can reflect the wide range of random frequencies encountered in real experimental setups for measurement noise.

TABLE IV
PARAMETERS FOR SIMULATION AND EXPERIMENT

Parameters	Value
RL Load	
Dc-link voltage	200 V
Dc-link capacitors	4700 μF
Modulation index	0.9
Switching frequency	5 kHz
Sampling frequency	10 kHz
I_{norm}	6.31 A
RL load	10 Ω/6 mH
Motor Load	
Rated speed	1500 r/min
Rated torque	19.1 N.m
Stator resistance	R_s= 0.833 Ω
Rotor resistance	R_r=0.596 Ω
Leakage inductance	L_{lr}=L_{ls}=0.003 H
Magnetizing inductance	L_m=0.09 H
Moment of inertia	J=0.0071
Pole	4

V. SIMULATION AND EXPERIMENT RESULTS

In this section, the data utilized to train and validate the proposed 1-D CNN method is obtained through an offline test with both simulation and experiment data. The dc source voltage is set at 200 V, with a charging voltage of 100 V per capacitor. The modulation method employed is the level-shift PWM, operating at a switching frequency of 5 kHz, and the sampling frequency is set at 10 kHz. Throughout the training and validation processes, diverse load conditions are considered, wherein the magnitude of current and power factor (PF) vary according to resistive (R) and inductive (L) loads. The system parameters of the 3L-NPC inverter listed in Table IV are used in the simulation, which will be used for experiments as well. Details regarding the dataset used for training and validation will be explained in the following section.

A. Training Using Modified 1-D CNN

The deep learning model is trained using datasets obtained from simulations and experiments including different current magnitudes and power factors, as listed in Table V. The three schemes are employed for training the model. First, the dataset derived from the simulation is divided into two groups: 80% (8544) for training and 20% (2150) for validation. Second, the dataset obtained from the experiment contains 3075 samples, which is less than the simulation dataset. It is also divided into 2460 training and 615 validation sets. Finally, when the data augmentation method is used, the dataset size is doubled, resulting in 21 388 samples. It includes 17 110 training and 4278 validation sets.

TABLE V
SYSTEM CONDITIONS USED FOR TRAINING DEEP LEARNING MODEL

DATA TYPE	Load conditions
	RL Load
Simulation	I_{norm}, PF = 0.841
Simulation	0.5 I_{norm}, PF = 0.914
Simulation	0.33 I_{norm}, PF = 0.942
Simulation	I_{norm}, PF = 0.841 with THD+N 3 % white noise
Simulation	I_{norm}, PF = 0.841 with THD+N 5 % white noise
Simulation	I_{norm}, PF = 0.841 with THD+N 7 % white noise
Simulation	I_{norm}, PF = 0.841 with THD+N 10 % white noise
Experiment	I_{norm}, PF = 0.841
Experiment	0.5 I_{norm}, PF = 0.914
Experiment	0.33 I_{norm}, PF = 0.942
	Motor Load
Simulation	0.3 ω_{rated}, 0.7 T_{rated}
Simulation	0.2 ω_{rated}, no load and 0.53, 0.7 T_{rated}
Simulation	0.2, no load and 0.53 ω_{rated}, 0.7 T_{rated} with THD+N 3% white noise
Experiment	0.2 ω_{rated}, no load and 0.53 ω_{rated}, 0.27 T_{rated}

To find the optimal value of the proposed data augmentation, four datasets with different THD+N values are used for deep learning model training (3%, 5%, 7%, 10%). Each model has 30 epochs for training and AdaMod is used as an optimization method [21]. The total 10 deep learning models are divided into three types: 1) simulation data, 2) simulation data with data augmentation, and 3) experiment data.

B. Validation Using Modified 1-D CNN

To verify the proposed data augmentation method and the deep learning model for fault diagnosis, an offline test using experiment data is performed. The verification results of deep learning models at various power factors and current amplitudes are listed in Table VI, where the datasets used for validation differ from those used for training. The validation results reveal an average accuracy difference of 0.42% and 0.26% between the models trained using simulation under a load condition of 0.33 I_{norm}, PF $=$ 0.942, and those trained with experimental data under a load condition of 0.5 I_{norm}, PF $=$ 0.914, respectively. It is noteworthy that the models trained with experiment data outperform those trained with simulation data. Furthermore, the model trained with the additional white noise of THD+N at 3% attains the highest accuracy, reaching 99.72% throughout the entire deep learning model. This demonstrates that the inclusion of additional white noise with a total harmonic distortion plus noise THD+N at 3% results in an accuracy improvement of up to 1.71% in every condition compared to the model trained solely with simulation or experiment data. In addition, the models were also validated under different motor load conditions. As shown in Table VII, the simulation data with white noise injection (THD+N 3%) exhibited the highest accuracy compared to models trained solely on simulation or experiment data, achieving an average accuracy of 99.13%.

The dataset includes variations in the K_P gain, with a value of 50 used for training data and 2 for validation data. This adjustment is intended to demonstrate the effectiveness of the proposed method across different gain settings and its ability to replicate experimental conditions accurately. A comparison of deep learning model performance under various dc-link balancing control gains is presented in Table VIII. This table illustrates how K_P gains of the balancing controller impact model performance. Specifically, models trained with pure simulation data achieve an average accuracy of 83.00%, whereas models trained with simulation data augmented with white noise, particularly THD+N 3%, reach an average accuracy of 92.60%. Furthermore, the deep learning model utilizing the proposed data augmentation method shows up to a 12.97% improvement in accuracy compared to models that do not employ this method.

C. Optimal Value of White Noise Injection

In this section, offline verification is performed by changing the THD+N value in various original THD conditions to find the optimal value of the white noise to be injected. To simulate the impact of a broad range of frequencies on the system, the effects of varying switching frequencies in RL load conditions are analyzed. These variations affect the THD values observed in the output current, ranging from 1% to 9%. Consequently, the optimal THD+N value is determined based on eight distinct THD conditions.

The effects of various output current THDs or switching frequencies will introduce a switching-ripple component in the actual ac currents of an NPC inverter, especially when connected to the grid. In grid-connected applications, these switching-ripple components often lead to increased THD, which can be represented by white noise that worsens the THD by increasing the bandwidth of the waveform. The proposed method remains effective in these scenarios, demonstrating performance comparable to RL load applications. However, the effectiveness of white noise diminishes at very low switching frequencies, as the noise characteristics may deviate from the assumptions of white noise. Therefore, selecting the appropriate THD+N value for specific switching frequencies is essential to ensure accurate representation and optimal system performance.

The verification results under various THD+N conditions and their effects on different output current THD or switching frequencies are listed in Table IX. The trend line of accuracy shows the highest value at THD+N value, which is double the THD of the original output current. Fig. 12 shows the THD+N value with respect to the global maximum value as a graph. When THD+N exceeds 17%, the accuracy of the deep learning model begins to decrease. This implies that the maximum limit of the white noise injection method is approximately 17%.

VI. REAL-TIME IMPLEMENTATION

The effectiveness of the proposed fault diagnosis method is evaluated through practical implementation involving real-time communication with the Python software program. The experimental setup, as shown in Fig. 13, includes DSP TMS320F28335 for controlling the 3L-NPC inverter and multi-CAN analyzer for real-time communication. Fig. 14 shows the communication structure for hardware implementation. The real-time hardware implementation process integrates the DSP control board, executing the fault diagnosis algorithm, with a PC running a

TABLE VI
VALIDATION RESULTS OF VARIOUS TRAINED DEEP LEARNING MODELS IN RL LOAD CONDITIONS

Training data		Validation data accuracy [%]				
Data type	System condition	190 mA, PF = 0.914	631 mA, PF = 0.975	1.33 A, PF = 0.975	4.42 A, PF = 0.914	Average
Trained data (Simulation)	I_{norm}, PF = 0.841	98.93	98.99	98.93	98.77	98.91
	0.5 I_{norm}, PF = 0.914	98.64	98.20	98.64	99.81	98.82
	0.33 I_{norm}, PF = 0.942	99.03	99.62	98.83	99.73	99.30
Trained data (Experiment)	I_{norm}, PF = 0.841	99.42	99.51	98.97	99.53	99.36
	0.5 I_{norm}, PF = 0.914	99.61	99.51	99.11	99.62	99.46
	0.33 I_{norm}, PF = 0.942	99.03	100.00	98.97	98.29	99.07
Trained data (Simulation with white noise)	**THD +N 3 %**	**99.73**	**99.62**	**99.51**	**100.00**	**99.72**
	THD +N 5 %	99.24	99.31	99.05	98.98	99.15
	THD +N 7 %	99.43	99.60	99.11	98.74	99.22
	THD +N 10 %	99.14	99.41	98.98	99.41	99.24

TABLE VII
VALIDATION RESULTS OF VARIOUS TRAINED DEEP LEARNING MODELS IN MOTOR LOAD CONDITIONS

Training data		Validation data accuracy [%]				
Data type	System condition	0.2 ω_{rated}, no load	0.8 ω_{rated}, no load	0.6 ω_{rated}, 0.27 T_{rated}	0.6 ω_{rated}, 0.38 T_{rated}	Average
Trained data (Simulation)	0.3 ω_{rated}, 0.7 T_{rated}	100	95.01	89.44	89.01	93.37
	0.2 ω_{rated}, no load and 0.53 ω_{rated}, 0.7 T_{rated}	100	100	99.02	94.02	98.26
Trained data (Experiment)	0.2 ω_{rated}, no load and 0.53 ω_{rated}, 0.27 T_{rated}	100	99.58	97.41	96.21	98.30
Trained data (Simulation with white noise)	**0.2 ω_{rated}, no load and 0.53 ω_{rated}, 0.7 T_{rated} with THD+N 3% white noise**	**100**	**100**	**99.42**	**97.11**	**99.13**

TABLE VIII
VALIDATION RESULTS WITH DIFFERENT DC-LINK BALANCING CONTROLLER GAINS

Training data with $K_P = 50$		Validation data accuracy with $K_P = 2$ [%]			
Data type	System condition	I_{norm}, PF = 0.841	0.5 I_{norm}, PF = 0.914	0.33 I_{norm}, PF = 0.942	Average
Trained data (Simulation)	I_{norm}, PF = 0.841	92.21	84.11	72.69	83.00
Trained data (Simulation with white noise)	**THD +N 3 %**	**98.78**	**93.37**	**85.66**	**92.60**
	THD +N 5 %	91.51	85.74	79.94	85.73
	THD +N 7 %	91.51	85.74	79.92	85.72
	THD +N 10 %	88.54	84.11	75.44	82.70

TABLE IX
VALIDATION RESULTS OF VARIOUS OUTPUT CURRENT THD

Validation data		Training data for THD+N of output current with white noise injection [%]														
THD of output current [%]	Switching frequency [kHz]	1	1.5	2	3	4	5	6	7	8	9	10	12	15	17	20
1	10	85.9	-	**93.3**	82.5	-	82.5	-	77.1	-	-	76.6	81.9	76.1	82.9	73.4
1.5	5	-	92.2	86.3	**98.8**	-	91.5	-	91.5	-	-	88.5	88.2	80.5	83.4	76.8
2	5	-	-	77.3	95.1	97.6	**99.6**	-	87.4	-	-	91	79.4	94.6	84.3	79.2
3	3	-	-	-	72.6	-	72.9	92.2	**93.7**	-	-	87.9	91.5	78.9	88.9	77.1
4	3	-	-	-	-	76.3	90.2	-	**98.9**	87.8	86.6	82.5	74.1	89.4	76.7	73.7
5	3	-	-	-	-	-	75.4	-	85.3	-	-	**95.5**	80.2	88.3	81.2	78.2
6	2	-	-	-	-	-	-	80.5	82.4	-	-	81.1	**93.1**	89.3	83.9	80.1
9	1	-	-	-	-	-	-	-	-	-	90.7	95.7	**100**	100	82.6	79.9

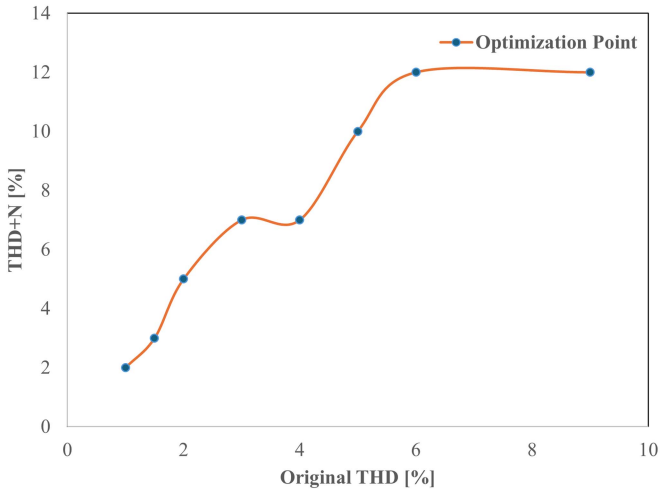

Fig. 12. Optimal value trend of THD+N in white noise injection method.

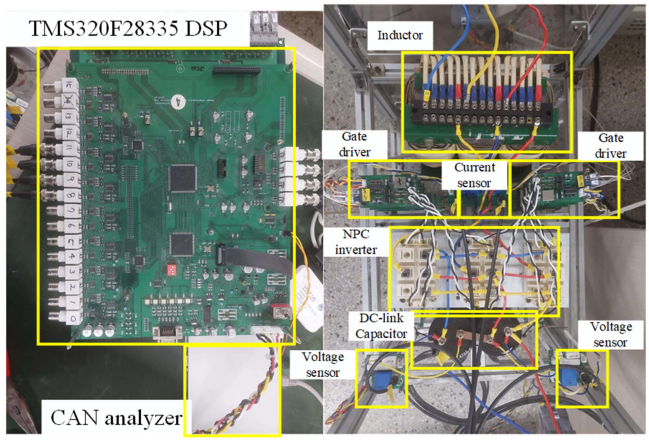

Fig. 13. Experimental setup of the proposed three-level NPC inverter.

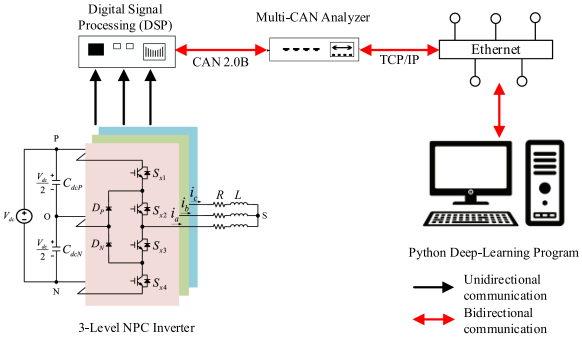

Fig. 14. Communication setup for a real-time implementation using Python program.

Python program. This integration involves transforming the CAN communication protocol into TCP/IP socket communication, facilitated by the multi-CAN analyzer. This configuration enables robust data exchange, where the algorithm efficiently handles incoming data, conducts fault diagnosis computations, and transmits the results to a PC for monitoring and analysis. Initially, the deep learning model is carefully optimized to minimize computational complexity, while still maintaining high

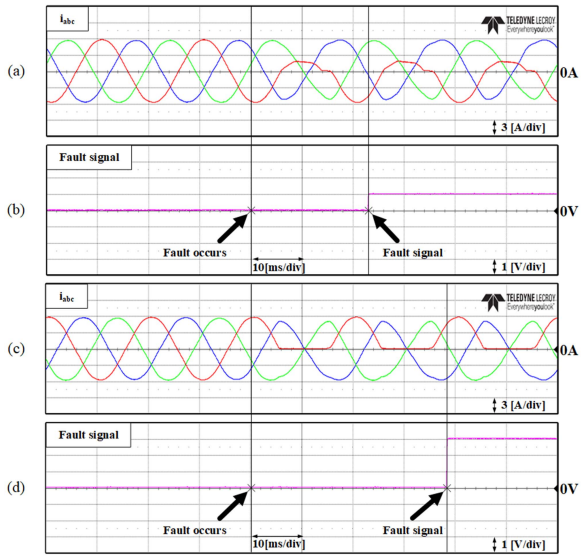

Fig. 15. Real-time fault diagnosis with deep learning program. (a) Three-phase current waveform in S_{a1} fault. (b) Fault flag of S_{a1}. (c) Three-phase current waveform in S_{a3} fault. (d) Fault flag of S_{a3}.

accuracy. This optimization involves employing a 1-D CNN model with strategic modifications like dilated convolution, residual blocks, and the GAP layer. These adjustments enable efficient feature extraction and reduce the computational burden, making the algorithm feasible for implementation on hardware platforms with limited resources. Through this integration and algorithm design, the system operates in real-time to ensure effective fault diagnosis.

Fig. 15 shows the test results of real-time fault diagnosis with a deep learning method. Fig. 15(a) and (b) shows the current waveforms and a fault flag of switch S_{a1}, respectively. Fig. 15(c) and (d) corresponds to the fault case of switch S_{a2}. The diagnostic time interval of real-time implementation executed on a CPU is from 18 to 50 ms, which varies depending on the operating communication conditions and the implementation of deep learning algorithms. When executing the code using the computational resources of a GPU within the Google Colab environment, the execution time ranges from 7.02 to 12 ms. When the same code is executed on a CPU, the execution time ranges from 14 to 27 ms.

It is important to analyze the performance of deep learning models under various conditions. Fig. 16 shows the response of the system to a load change of 2.15 A with a PF $= 0.998$. Similarly, Fig. 17 illustrates the response to a change in voltage, specifically with $V_{dc} = 150$ V. Notably, both conditions show that despite variations in load and voltage the reliability of the system remains robust. This observation demonstrates the ability of the system to diagnose faults in a consistent duration spanning from 18 to 50 ms.

Inverter fault detection algorithms must account for load variations to prevent false alarms. It is imperative to consider these conditions in the performance evaluation of fault detection algorithms, given their higher frequency compared to actual faults. The comprehensive examination of load variations and speed change transients demonstrates the effectiveness of the

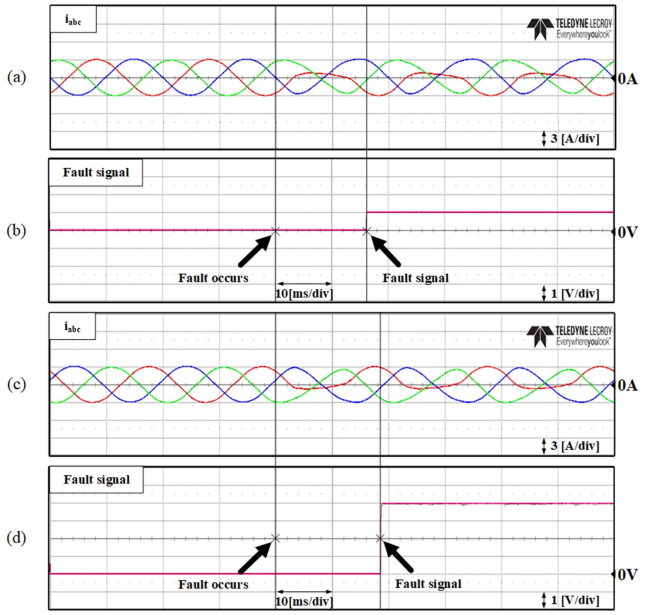

Fig. 16. Fault diagnosis in load change condition at 2.15 A, PF = 0.998. (a) Three-phase current waveform in S_{a1} fault. (b) Fault flag of S_{a1}. (c) Three-phase current waveform in S_{a4} fault. (d) Fault flag of S_{a4}.

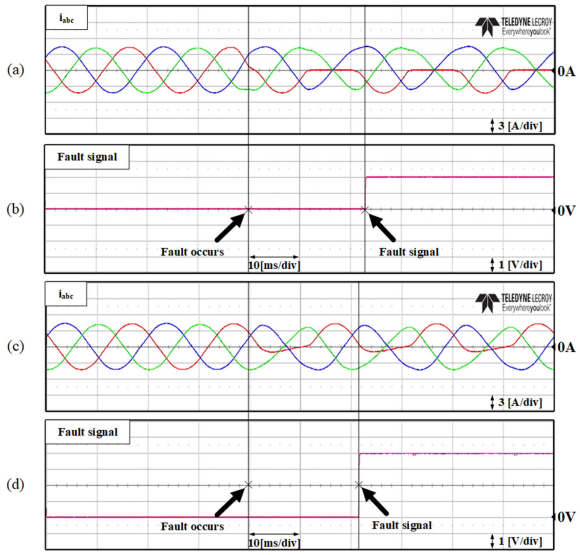

Fig. 17. Fault diagnosis in voltage change condition at V_{dc} = 150 V. (a) Three-phase current waveform in S_{a2} fault. (b) Fault flag of S_{a2}. (c) Three-phase current waveform in S_{a4} fault. (d) Fault flag of S_{a4}.

fault detection algorithm in distinguishing between real faults and normal operational changes. The resilience of the algorithm to dynamic conditions is illustrated by its performance during load variations from 798 W to 158 W and speed (frequency) changes from 600 r/min to 1200 r/min (20 Hz to 40 Hz). Figs. 18 and 19 reveal that the fault flag remains deactivated even when the transient occurs in the system. The absence of false alarms during load and speed (frequency) changes indicates the robustness of the algorithm and its ability to accurately diagnose faults without triggering incorrect alarms.

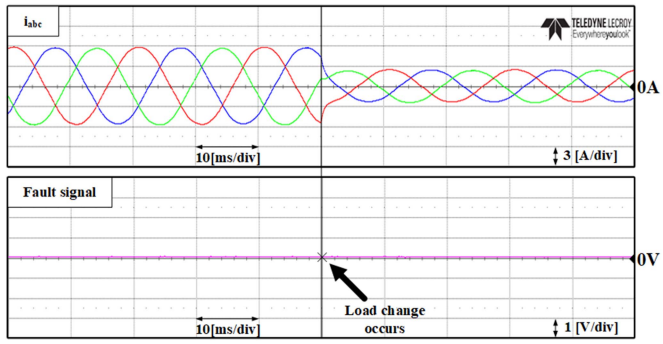

Fig. 18. Fault diagnosis in load transient condition from 798 W to 158 W. (a) Three-phase current waveforms. (b) Fault flag.

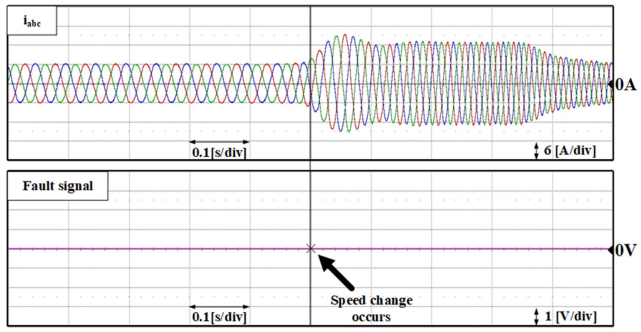

Fig. 19. Fault diagnosis in a step change of speed from 600 r/min to 1200 r/min. (a) Three-phase current waveforms. (b) Fault flag.

TABLE X
DIAGNOSING TIME OF CONVENTIONAL CNN AND MODIFIED CNN

Method	Conventional CNN	Modified CNN
Time [ms]	20-110	18-50

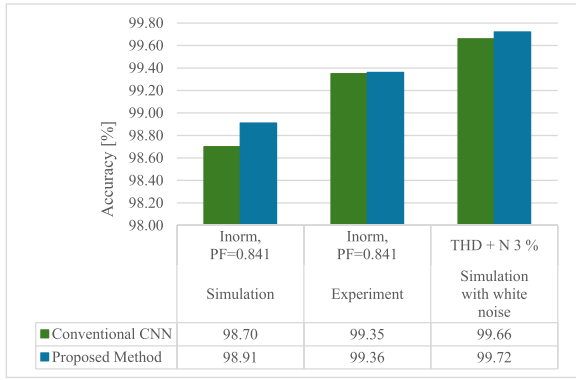

Fig. 20. Performance of conventional CNN and modified CNN.

To ensure a fair comparison between the conventional and proposed models, both are trained and validated using identical datasets. Specifically, both models are trained and validated using simulation data, experiment data, and simulation data with white noise injection. The effectiveness of the modified CNN compared to conventional CNN is shown in Fig. 20. The modified CNN is better with slight improvement and achieves a diagnosing performance of 98.1%, 99.36%, and 99.72%. The

TABLE XI
COMPARISON WITH THE EXISTING METHODS

Reference Paper	Additional Hardware	Accuracy	Diagnostic Time	Robustness
[10]	Three voltage sensors	99.54	Not mentioned	Not analyze
[11]	Two voltage sensors	Not mentioned	≤ 10 ms	Robust to different load
[12]	Two voltage sensors	Not mentioned	minimum < 5 ms maximum < 20 ms	Robust to different modulation methods, carrier frequencies, and load
[14]	No additional sensors	98.07	≥ 20 ms	Robust to different load
Proposed method	**No additional sensors**	**99.33**	**18 – 50 ms**	**Robust to different loads, speeds, switching frequencies, and balancing gain**

diagnosing times of both the conventional CNN and the modified CNN are presented in Table X. As indicated, the modified CNN demonstrates significantly reduced diagnosing times, with the maximum time recorded at 50 ms, highlighting its efficiency in fault detection and analysis.

However, by utilizing a CAN analyzer for real-time implementation, advancements in the system can further decrease the diagnosing time. The diagnosing speed varies depending on the fundamental frequency and the performance of the PC, with different fault detection times caused by real-time program delay. Using a GPU for executing the program can result in faster fault detection compared to using a CPU. Testing the execution speed of the deep learning model shows that data can be classified in 8.3 ms with a GPU, whereas using a CPU results in a speed of 17 ms. These values enable the calculation of the bidirectional data transfer time from the DSP board to the PC and back to the DSP to indicate the fault flag. By subtracting the diagnosing time and the deep learning processing time, a delay range of 1 to 33 ms in bidirectional data transfer is determined.

Finally, a comparison of the proposed method with existing fault diagnosis techniques for 3L-NPC inverters is provided in Table XI, where hardware requirements, diagnostic accuracy, detection time, and robustness are evaluated. The proposed method can diagnose open-switch faults without additional sensors and achieve a high diagnosis accuracy of 99.33%.

A microcontroller can serve as the ultimate implementation platform, but this research utilizes a PC with Python to simplify the development process. For microcontroller deployment, the pretrained model from Python can be optimized and converted into ONNX format. This ONNX model can then be further optimized with TensorRT, allowing it to be embedded into microcontroller platforms such as the Jetson Nano [38].

VII. CONCLUSION

This article has proposed a robust fault diagnosis method for identifying faults and locating faulty switches in the open-circuit fault of a 3L-NPC inverter. This method utilizes a 1-D CNN to extract fault features from three-phase current data. To mitigate challenges related to data extraction from experimental conditions and the need for a substantial dataset, a data augmentation method has been proposed. This method improves model generalization by increasing the robustness and accuracy of CNN models for various system conditions using only simulation data from a single system condition. However, the approach of using data augmentation with white noise injection and modified CNN architectures applies to various inverter configurations based on specific application needs and fault detection goals. In addition, training the model for the specific inverter topology ensures accurate fault detection, considering distinct fault characteristics. Validation through an offline test with experiment data has confirmed its efficacy. In particular, under conditions featuring untrained load conditions, the evaluation results have demonstrated a significant accuracy improvement of up to 1.71% compared to the same models without the injection of white noise. In addition, real-time deep learning implementation has enabled fault diagnosis within a timeframe of 18 to 50 ms. Ongoing advancements in software optimization and hardware capabilities suggest the possibility of even faster and more cost-effective fault diagnosis approaches in the future.

REFERENCES

[1] S. Yang, A. Bryant, P. Mawby, D. Xiang, L. Ran, and P. Tavner, "An industry-based survey of reliability in power electronic converters," *IEEE Trans. Ind. Appl.*, vol. 47, no. 3, pp. 1441–1451, May/Jun. 2011, doi: 10.1109/TIA.2011.2124436.

[2] U.-M. Choi, H.-G. Jeong, K.-B. Lee, and F. Blaabjerg, "Method for detecting an open-switch fault in a grid-connected NPC inverter system," *IEEE Trans. Power Electron.*, vol. 27, no. 6, pp. 2726–2739, Jun. 2012, doi: 10.1109/TPEL.2011.2178435.

[3] Y. Xia, Y. Xu, and N. Zhou, "A transferrable data-driven method for IGBT open-circuit fault diagnosis in three-phase inverters," *IEEE Trans. Power Electron.*, vol. 71, no. 7, pp. 8017–8027, Jul. 2024, doi: 10.1109/TIE.2023.3301508.

[4] Z. Jian-Jian, C. Yong, C. Zhang-Yong, and Z. Anjian, "Open-switch fault diagnosis method in voltage-source inverters based on phase currents," *IEEE Access*, vol. 7, pp. 63619–63625, 2019, doi: 10.1109/ACCESS.2019.2913164.

[5] S. Zhang, R. Wang, Y. Si, and L. Wang, "An improved convolutional neural network for three-phase inverter fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3510915, doi: 10.1109/TIM.2021.3129198.

[6] A. Nabae, I. Takahashi, and H. Akagi, "A new neutral-point-clamped PWM inverter," *IEEE Trans. Ind. Appl.*, vol. IA-17, no. 5, pp. 518–523, Sep. 1981, doi: 10.1109/TIA.1981.4503992.

[7] F. Blaabjerg, *Control of Power Electronic Converters and Systems: Volume 1*. London, U.K.: Academic, 2018.

[8] A. I. Maswood and H. D. Tafti, *Advanced Multilevel Converters and Applications in Grid Integration*. Hoboken, NJ, USA: Logo, 2018.

[9] B. Wu and M. Narimani, *High-Power Converters and AC Drives*, 2nd ed. Hoboken, NJ, USA: Wiley, 2017.

[10] Y. Yu and S. Pei, "Open-circuit fault diagnosis of neutral point clamped three-level inverter based on sparse representation," *IEEE Access*, vol. 6, pp. 73499–73508, 2018, doi: 10.1109/ACCESS.2018.2883219.

[11] X. Wu, T. F. Chen, S. Cheng, T. Yu, C. Xiang, and K. Li, "A noninvasive and robust diagnostic method for open-circuit faults of three-level inverters," *IEEE Access*, vol. 7, pp. 2006–2016, 2019, doi: 10.1109/ACCESS.2018.2886706.
[12] Y. Hu, S. Cheng, X. Wu, C. Xiang, and Z. Li, "A diagnostic method for open-circuit faults of loads and semiconductors in 3L-NPC inverters," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 11, no. 3, pp. 2577–2590, Jun. 2023, doi: 10.1109/JESTPE.2022.3205897.
[13] Y. Li, H. Xiao, N. Jin, and G. Yan, "Model predictive control of NPC three-level grid-tied converter based on reconstructed current," *Arch. Elect. Eng.*, vol. 71, no. 2, pp. 363–377, 2022, doi: 10.24425/aee.2022.140716.
[14] L. Kou, C. Liu, G. W. Cai, J. N. Zhou, Q. De Yuan, and S. M. Pang, "Fault diagnosis for open-circuit faults in NPC inverter based on knowledge-driven and data-driven approaches," *IET Power Electron.*, vol. 13, no. 6, pp. 1236–1245, 2020, doi: 10.1049/iet-pel.2019.0835.
[15] D.-E. Kim and D.-C. Lee, "Fault diagnosis of three-phase PWM inverters using wavelet and SVM," *J. Power Electron.*, vol. 9, no. 3, pp. 377–385, May 2009, doi: 10.6113/JPE.2009.9.3.377.
[16] S. Khomfoi and L. M. Tolbert, "Fault diagnostic system for a multilevel inverter using a neural network," *IEEE Trans. Power Electron.*, vol. 22, no. 3, pp. 1062–1069, May 2007, doi: 10.1109/TPEL.2007.897128.
[17] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 4633–4658, Apr. 2021, doi: 10.1109/TPEL.2020.3024914.
[18] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Al-Emadi, and M. Gabbouj, "Real-time fault detection and identification for MMC using 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8760–8771, Nov. 2019, doi: 10.1109/TIE.2018.2833045.
[19] S.-H. Kim, D.-Y. Yoo, S.-W. An, Y.-S. Park, J.-W. Lee, and K.-B. Lee, "Fault detection method using a convolution neural network for hybrid active neutral-point clamped inverters," *IEEE Access*, vol. 8, pp. 140632–140642, 2020, doi: 10.1109/ACCESS.2020.3011730.
[20] W. Yuan, Z. Li, Y. He, R. Cheng, L. Lu, and Y. Ruan, "Open-circuit fault diagnosis of NPC inverter based on improved 1-D CNN network," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3510711, doi: 10.1109/TIM.2022.3166166.
[21] J. Ding, X. Ren, R. Luo, and X. Sun, "An adaptive and momental bound method for stochastic learning," 2019, *arXiv:1910.12249*.
[22] J. Xiang and Y. Zhong, "A novel personalized diagnosis methodology using numerical simulation and an intelligent method to detect faults in a shaft," *Appl. Sci.*, vol. 6, no. 12, p. 414, 2016, doi: 10.3390/app6120414.
[23] Y. Gao, X. Liu, and J. Xiang, "FEM simulation-based generative adversarial networks to detect bearing faults," *IEEE Trans. Ind. Inform.*, vol. 16, no. 7, pp. 4961–4971, Jul. 2020, doi: 10.1109/TII.2020.2968370.
[24] Y. Lou, A. Kumar, and J. Xiang, "Machinery fault diagnosis based on domain adaptation to bridge the gap between simulation and measured signals," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3514709, doi: 10.1109/TIM.2022.3180416.
[25] Y. Gao, X. Liu, and J. Xiang, "Fault detection in gears using fault samples enlarged by a combination of numerical simulation and a generative," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 3798–3805, Oct. 2022, doi: 10.1109/TMECH.2021.3132459.
[26] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," in *Proc. ECML/PKDD Workshop Adv. Analytics Learn. Temporal Data*, Riva Del Garda, Italy, 2016, pp. 11–18.
[27] Q. Wen et al., "Time series data augmentation for deep learning: A survey," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 2021, pp. 4653–4660, doi: 10.24963/ijcai.2021/631.
[28] J. Jung, "Open-circuit fault diagnosis of three-level NPC inverters based on 1-D convolutional neural network with data augmentation," M.S. thesis, Yeungnam Univ., Gyeongsan, South Korea, 2023.
[29] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. 4th Int. Conf. Learn. Representations*, San Juan, Puerto Rico, 2016, pp. 1–13.
[30] A. van den Oord et al., "WaveNet: A generative model for raw audio," in *Proc. 9th ISCA Speech Synth. Workshop*, 2016, p. 125.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[32] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. 2nd Int. Conf. Learn. Representation*, Banff, AB, Canada, 2014, pp. 1–10.
[33] I. Goodfellow et al., "Generative adversarial networks," *Assoc. Comput. Mach.*, vol. 63, no. 11, pp. 139–144, 2020, doi: 10.1145/3422622.
[34] F. J. Moreno-Barea, F. Strazzera, J. M. Jerez, D. Urda, and L. Franco, "Forward noise adjustment scheme for data augmentation," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2018, pp. 728–734, doi: 10.1109/SSCI.2018.8628917.
[35] *IEEE Standard for Terminology and Test Methods of Digital-to-Analog Converter Devices*, IEEE Standard 1658, pp. 1–126, Feb. 2012, doi: 10.1109/IEEESTD.2012.6152113.
[36] W. G. Jung, *Op Amp Applications Handbook*. Oxford, U.K.: Newnes, 2005.
[37] J. A. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[38] C. Yao et al., "Online open-circuit fault diagnosis for ANPC inverters using edge-based lightweight two-dimensional CNN," *IEEE Trans. Power Electron.*, vol. 39, no. 4, pp. 3979–3984, Apr. 2024, doi: 10.1109/TPEL.2024.3351911.

Jiwon Jung (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Yeungnam University, Gyeongsan, South Korea, in 2021 and 2023, respectively.

He is currently with Hanwha Systems Company, Ltd., Daegu, South Korea. His research interests include micro PV inverters and AI-based inverter fault diagnosis.

Dyan Puspita Apsari (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from the University of Indonesia, Depok, Indonesia, in 2020. She is currently working toward the Ph.D. degree in electrical engineering with Yeungnam University, Gyeongsan, South Korea.

Her current research interests include intelligent control of power electronic systems and AI-based fault diagnosis.

Dong-Choon Lee (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 1985, 1987, and 1993, respectively.

He was a Research Engineer with Daewoo Heavy Industry, South Korea, from 1987 to 1988. He has been a faculty member with the Department of Electrical Engineering, Yeungnam University, Gyeongsan, South Korea, since 1994. He was a Visiting Scholar in the Power Quality Laboratory, Texas A&M University, College Station, TX, USA, in 1998; the Electrical Drive Center, University of Nottingham, Nottingham, U.K., in 2001; the Wisconsin Electric Machines and Power Electronics Consortium, University of Wisconsin, Madison, WI, USA, in 2004; the FREEDM Systems Center, North Carolina State University, Raleigh, NC, USA, from 2011 to 2012. His current research interests include power converter design and control, renewable energy and its grid connection, ac machine drives, and dc power system.

Dr. Lee was the Editor-in-Chief of the *Journal of Power Electronics* of the Korean Institute of Power Electronics (KIPE), from 2015 to 2017. In 2019, he was the President of KIPE. He was also the General Chair of ICPE 2023-ECCE Asia held in Jeju, South Korea. He is currently an Associate Editor for IEEE TRANSACTIONS ON POWER ELECTRONICS and the IEEE Power Electronics Society Chapter Chair in Taegu Section.