International Journal of Research and Technology in Electrical Industry

journal homepage: ijrtei.sbu.ac.ir

Shahid Beheshti University

Transfer Learning-Based Open-Circuit Fault Detection Using Time-Frequency Analysis on Small Datasets for Voltage Source Inverters

Abbas Babaei Birag[1],Ahmad Salemnia[1,*],Nazanin Pourmoradi[1]

[1] *Department of Electrical Engineering, Abbaspour School of Engineering, Shahid Beheshti University, Tehran, Iran*

ARTICLE INFO

Article history:
Received: 09 February 2025
Revised: 06 December 2025
Accepted: 20 December 2025

Keywords:
Transfer learning
Continuous wavelet transform
Lightweight convolution neural network fault Detection
Voltage source inverters
Open-circuit fault

ABSTRACT

Open-circuit (OC) fault identification in Voltage-Source Inverters (VSIs) is a critical challenge for the reliability of power systems and motor drives. While Deep Learning (DL) technologies offer automatic feature extraction, they typically suffer from high computational costs and the requirement for massive labeled datasets, which are scarce in real-world industrial scenarios. This paper provides an innovative lightweight diagnostic approach utilizing Transfer Learning (TL) to address current issues of data scarcity and training inefficiency. The pre-trained SqueezeNet model, an efficient Convolutional Neural Network (CNN) with a lower parameter count, is fine-tuned to properly categorize various fault states. In the proposed methodology, the three-phase output current signals are first converted into time-frequency scalograms using the Continuous Wavelet Transform (CWT) to capture rich transient fault features. Subsequently, these visual representations are processed by the network. The proposed method achieves 99.90% accuracy on a small dataset (2000 samples) with considerably reduced training time relative to training deep models from scratch. These findings demonstrate the effectiveness and robustness of the suggested methodology for real-time fault diagnosis in inverters.

1. Introduction

Voltage-Source Inverters (VSIs) are widely employed in modern industrial applications, such as motor drive systems, electric vehicle propulsion, and power quality enhancement, due to their excellent energy efficiency and exceptional control flexibility [1]. Nevertheless, the reliability of these converters is a major concern, as power switching components, especially Insulated Gate Bipolar Transistors (IGBTs), are prone to malfunctions. Faults in power switches are mainly classified into two categories: short-circuit (SC) and open-circuit (OC). While short-circuit faults are typically mitigated by immediate hardware protection measures to prevent sudden failure, open-circuit faults are more insidious. Initially, they often remain undetected; however, they cause current distortion and torque fluctuations, ultimately leading to catastrophic secondary failures and significant economic losses [2]. Consequently, developing robust and accurate fault diagnosis techniques is vital for ensuring system safety and continuous operation. Generally, fault detection techniques are classified into model-based, signal-based, and data-driven methodologies [3].

Model-based techniques depend on exact mathematical representations of the system, employing estimators like the interval sliding mode observer [4] or Kalman filter [5] to predict system states. Although these methods are accurate, they often require a lot of computing power and are sensitive to parameter variation. Signal-based techniques utilize tools such as Fast Fourier Transform [6] or Wavelet Transform [7] to extract fault features directly from measured signals. Although these methods are simpler than model-based approaches, they

* Corresponding author
E-mail address: a_salemnia@sbu.ac.ir
https://orcid.org/0000-0001-9892-1331
https://doi.org/10.48308/ijrtei.2025.238719.1074

frequently suffer from sensitivity to noise and variations in operating conditions [8].

To address the challenges caused by traditional methods, data-driven approaches employing Machine Learning (ML) received significant attention. Early implementations focused on classic algorithms; for instance, Yu [9] enhanced Support Vector Machines (SVM) using particle swarm optimization for small sample classification, while Changli et al. [10] employed the K-Nearest Neighbors (KNN) algorithm alongside dimensionality reduction techniques. To further improve classification efficiency, hybrid frameworks were developed, such as combining Principal Component Analysis (PCA) with Multiclass Relevance Vector Machines (RVM) [11], or utilizing hybrid ensemble learning based on Extreme Learning Machines (ELM) [12]. Moreover, ensemble techniques have demonstrated robustness in handling noisy data; Shan et al. [13] and Kou et al. [14] applied Random Forests (RF) to analyze transient fault characteristics. Similarly, the Ensemble Bagged Tree technique has been utilized to effectively diagnose multiple open-switch faults by analyzing statistical features [15]. nevertheless, as emphasized in [16] and [17], most traditional machine learning algorithms rely heavily on manual feature extraction and specific knowledge, limiting their generalizability in complex, dynamic environments. As a result, Deep Learning (DL) has become the preferred approach for autonomous feature learning.

Initial studies mainly adopted one-dimensional Convolutional Neural Networks (1D-CNNs), regarded due to their computational efficiency and suitability for real-time applications. Notable examples involve the real-time 1D-CNNs developed by Kiranyaz et al. [18] and the optimized lightweight models created by Yuan et al. [19]. Subsequent advancements adopted complex-valued CNNs to collect phase-amplitude information [20] and sample-amplification approaches to address limited training data [21]. However, these one-dimensional approaches frequently fail to represent complex time-frequency textures and transitory patterns crucial for To accurate diagnosis in variable operating conditions. address these restrictions, the research emphasis moved to two-dimensional (2D) frameworks by converting one-dimensional (1D) information into images. Sun et al. [22] employed Continuous Wavelet Transform (CWT) along with the AlexNet architecture to extract precise transient features for OCF detection. Other studies utilized Short-Time Fourier Transform (STFT) together with multiscale kernel CNNs to enhance the quality of time-frequency representation [23]. To address STFT's fixed resolution limits, subsequent developments combined the Synchrosqueezing Wavelet Transform (SWT) and an Extreme Learning Machine (ELM) classifier [24]. Furthermore, normalized current vector trajectory graphs have been presented as an approach to capture detailed fault-related patterns when input into a Wavelet CNN [25]. Despite their effectiveness, these 2D techniques typically require complex designs (such as AlexNet) with a large number of parameters, making them unsuitable for embedded devices with limited resources.

Additionally, the lack of labeled fault data motivated the use of advanced generative and attention-based methods. Synthetic fault samples have been synthesized using WAC-GANs [26] and hybrid frameworks integrating Conditional GANs with Vision Transformers [27]. Furthermore, adaptive sparse attention networks have been established to improve model noise robustness [28]. While these approaches improve accuracy, they introduce substantial structural complexity, training instability, and high computational overhead.

Consequently, there is an essential need for a diagnostic framework that is simultaneously lightweight and data-efficient while maintaining high accuracy. This study suggests a Transfer Learning (TL) technique employing the pre-trained SqueezeNet architecture to address this issue. The suggested method achieves robust detection of both single and concurrent open-circuit faults on small datasets by integrating CWT scalograms with the lightweight architecture of SqueezeNet. This method efficiently eliminates dependence on costly synthetic data generation or significant processing resources, ensuring acceptability for real-time applications.

The main contributions of this work are summarized as follows:

1. **Enhanced Noise Robustness:** The combination of CWT scalograms and deep feature extraction provides accurate fault diagnosis even under severe electromagnetic interference (up to 10 dB SNR) and fluctuating load conditions.
2. **Effective Small-Data Learning:** By employing a transfer learning strategy, the proposed method effectively overcomes the industrial challenge of data scarcity, achieving high precision on limited datasets without complex data augmentation.
3. **Real-Time Efficiency:** The SqueezeNet architecture has a significantly reduced memory usage and computational demand compared to typical CNNs, making it a superior solution for real-time deployment on resource-constrained embedded systems.

The rest of the paper is structured as follows: The theoretical basis of transfer learning, SqueezeNet design, and CWT are presented in Section 2. The suggested fault detection technique and implementation procedures are described in depth in Section 3. Section 4 shows the results of the simulation and the implementation of the suggested model, along with a comparative analysis under various conditions. Finally, Section 5 concludes the paper and outlines future research directions.

2. THEORETICAL BACKGROUNDS

Fig. 1 illustrates the configuration of a three-phase two-level Voltage Source Inverter employed in this research. The system operates using a DC-link voltage source (Vdc) and consists of three phase legs, each having two complementary IGBTs with antiparallel diodes (e.g., Sa1 and Sa2). A Pulse-Width Modulation (PWM) technique handles the switching states to provide sinusoidal three-phase AC currents (ia, ib, ic) for the RL load. An LC filter is incorporated at the output stage to reduce harmonic distortion and maintain power quality. Open-circuit failures in the IGBTs directly cause asymmetry and distortion in the output currents, making them one of the

main monitoring signals for the proposed diagnostic architecture.

2.1. Open-Circuit Fault

In contrast to SC faults that activate immediate hardware protection, OC faults in IGBTs allow the VSI to continue operation under damaged conditions. This faulty operation results in considerable asymmetry and distortion in the three-phase output currents, potentially

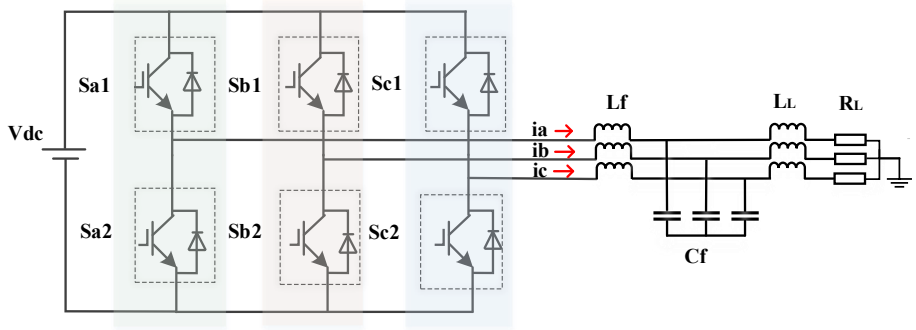


Fig. 1. Circuit topology of the three-phase two-level voltage source inverter with an RL load.

causing serious secondary damage if not identified. Thus, this study categorizes the inverter's operational situations into 10 specific categories as outlined in Table I. The categories comprise the healthy condition (Class 1), six cases of single-switch failures (e.g., Sa1 fault), and three cases of simultaneous double-switch failures (e.g., Sa1 and Sb2 faults).

Table I. Classification of open-circuit fault modes and associated labels.

Fault type	Faulty switches	Labels
Normal	No fault	class 1
single-switch faults	S_{a1} , S_{a2} , S_{b1}, S_{b2}, S_{c1}, S_{c2}	class 2, class 4, class 6 class 8, class 9, class 10
double-switch faults	S_{a1b2} , S_{a2c1}, S_{b1c2} ,	class 3, class 5, class7

2.2. Continuous Wavelet Transform

Traditional signal processing techniques, such as the Fourier Transform, are effective for stationary signals but fail to capture temporal information regarding transient frequency changes. Although the STFT utilizes a windowing technique to address this, it suffers from a fixed resolution trade-off dictated by the uncertainty principle [29]. This work employs the Continuous Wavelet Transform to address these restrictions. In contrast to Fourier-based approaches, the CWT decomposes non-stationary signals into wavelets that are localized in both time and frequency domains. Fundamentally, the transform calculates the inner product of the input signal with the wavelet function at various scales and time shifts. This process generates a two-dimensional time-scale representation known as a scalogram, which effectively visualizes the signal's energy distribution [30]. Due to its multi-resolution analytical capacity, it provides high-frequency resolution at low frequencies and high-time resolution at high frequencies. CWT is optimal for detecting sudden fault transients in inverter currents. Mathematically, the CWT of a continuous signal $x(t)$ is defined as the convolution of the signal with a mother wavelet function $\psi(t)$:

$$C(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi \left(\frac{t-b}{a} \right) dt, \quad (1)$$

Scale (a) is inversely related to frequency, whereas translation (b) quantifies the temporal variation [31]. By changing the values of a and b, certain signal characteristics in the time-frequency domain can easily be found and investigated. This work utilizes the Haar wavelet as its foundation due to its discontinuous nature, which is suitable for capturing switching failures caused by abrupt transitions. To analyze the signal in its entirety, the translation parameter b is adjusted in steps that correspond to the sample frequency, and the scale parameter an is adjusted throughout a logarithmic range.

2.3. Convolutional Neural Network

Since the paradigm of fault diagnostics has shifted from manual feature engineering to automatic representation learning, CNNs have emerged as the dominant deep learning architecture [32], [33]. In contrast to traditional networks, which consider inputs as flat vectors, CNNs' hierarchical structure includes convolutional, pooling, and fully connected (FC) layers that capture the spatial and temporal patterns of data. The convolutional layer serves as the fundamental feature extractor. It employs a set of learnable kernels (filters) that slide over the input signal to generate feature maps, effectively capturing local patterns such as edges or transient impulses. The mathematical process for the j-th feature map in layer l can be described as [33]:

$$y_j^l = f\left(b_j^l + \sum_{i \in M_j} x_i^{l-1} * w_{ij}^l\right) \quad (2)$$

Where y_j^l represents output of layer l, and the output of layer $l-1$, which is the input for layer l, is represented as x_i^{l-1}. w_{ij}^l is the weight matrix of the convolution kernel and M_j is the set of input feature vectors. Additionally, b_j^l represents the network bias, and $f(\cdot)$ denotes the activation function. Also * represents the convolutional operator. To control the spatial dimensions of the output

feature maps and manage computational complexity, hyperparameters such as input width (W_{in}), kernel size (K), padding (P), and stride (S) are adjusted. The output size is calculated by [34]:

$$size_{out} = \left[\frac{W_{in} + 2P - K}{S}\right] + 1 \quad (3)$$

Subsequent to the convolutional layers, pooling layers are employed to lower the dimensionality of the feature maps. These layers successfully prevent the network from overfitting by drastically reducing the number of parameters and computational cost via subsampling the input [32]. Max-pooling is a widespread method that extracts the most prominent features by selecting the maximum value from local non-overlapping regions. The calculation process is expressed as:

$$y_j^l = f(\beta_j^l dowm(x_j^{l-1}) + b_i^l) \quad (4)$$

Where down(.) represents the subsampling function, and β represent the pooling kernel. This operation provides translation invariance, making the model more robust to small variations in the signal. Finally, FC layers get the flattened high-level features that were obtained by the applied convolutional and pooling layers. These layers combine the global information and transfer the acquired features to the final output classes. The output of the K-th neuron in the FC layer is computed as:

$$y^k = f(\omega^k x^{k-1} + b^k) \quad (5)$$

Furthermore, Batch Normalization (BN) is often implemented after convolutional layers to ensure training stability and convergence speed. Additionally, to add sparsity and eliminate the vanishing gradient problem, the Rectified Linear Unit (ReLU) is used as the activation function. Dropout regularization, which randomly removes units from the neural network during training, is used to prevent the model from overfitting, particularly when training on small datasets [34].

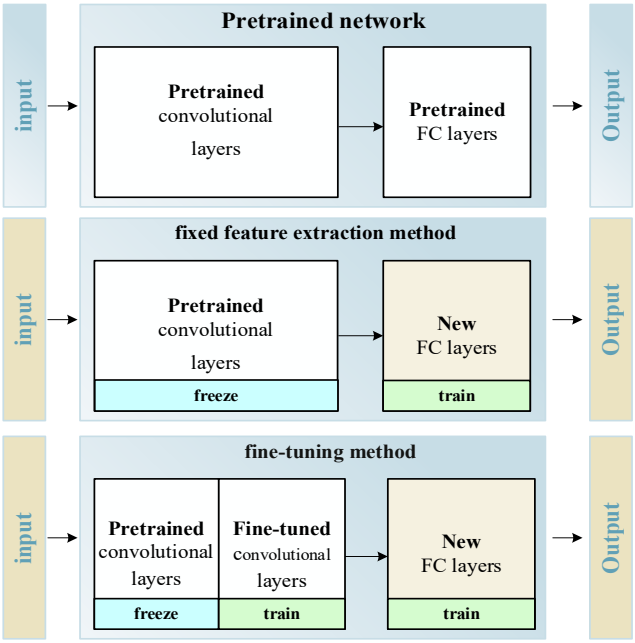


Fig. 2. Schematic comparison between fixed feature extraction and fine-tuning strategies in transfer learning for pre-trained networks.

2.4. Transfer Learning

The effectiveness of deep learning models depends on the availability of massive labeled datasets. However, in industrial fault detection, large-scale fault data collecting and labeling can sometimes become impracticable or prohibitively costly [35]. TL uses information from a source domain (D_s) with large amounts of data (such as ImageNet) to improve learning performance in a target domain (D_T) with limited data [36]. In the context of CNN architectures, TL leverages the hierarchical characteristics of feature extraction. The initial layers provide fundamental patterns, including edges and textures, that remain consistent throughout multiple tasks. In contrast, the deeper layers encode certain semantic features relevant to the target application. Starting with these pre-trained weights helps the model learn faster and prevents overfitting, especially when training data is limited. As depicted in Fig. 2, TL methods are primarily classified into two principal approaches:

1. **Fixed Feature Extraction**: The pre-trained feature extraction layers have been frozen to preserve the acquired general representations, while only the final classification layers undergo training. This approach regards the network as a fixed feature extractor.
2. **Fine-tuning**: The fine-tuning approach is used in this study. With this approach, the new classification layers are updated along with the parameters of the previous layers. In this way, the learned features of the model are adjusted to fit the distinctive characteristics of the inverter fault signals. When compared to training a model from scratch, this procedure leads to increased accuracy as well as faster training [37]. As explained in the next section, the lightweight SqueezeNet architecture is used to accomplish this method.

2.5. SqueezeNet structure

In this study, SqueezeNet, a lightweight CNN architecture, is used to provide effective and quick fault diagnosis with low computational overhead. While requiring around 50 times fewer parameters (model size < 0.5 MB), SqueezeNet achieves similar classification performance to more conventional deep models like AlexNet [38]. For real-world applications where processing speed and memory efficiency are crucial, this significant model size reduction makes it desirable [39]. As shown in Fig. 3, the "Fire module" is the basic building unit of this architecture. The purpose of this module is to decrease the number of parameters by substituting 1×1 filters for the conventional 3×3 filter. In particular, there are two stages for each Fire module:

1. **Squeeze Layer**: The 1×1 kernel convolution is used to compress the feature maps in a more efficient manner, which leads to the reduction of the total number of training parameters.
2. **Expand Layer**: This layer utilizes both 1×1 and 3×3 convolutional kernels. The simultaneous use of various filter sizes enables the network to effectively capture spatial patterns across multiple scales. This design guarantees precise classification with minimal computational cost by allowing the model to provide a comprehensive feature representation [40].

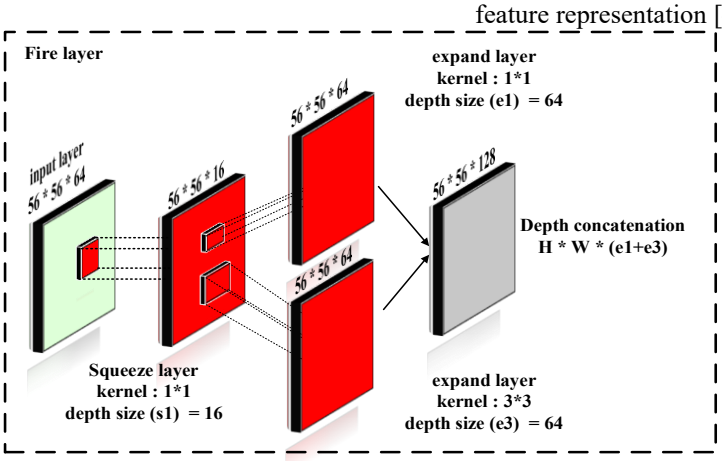


Fig. 3. Internal structure of the SqueezeNet Fire module, illustrating the squeeze and expand layers.

Finally, the feature maps from the expand layer are concatenated to merge the extracted information. These Fire modules are then stacked sequentially to create the entire SqueezeNet architecture, which is a deep and effective neural network.

3. PROPOSED FAULT DETECTION METHOD

The proposed structure for detecting OC faults in VSIs is shown in Fig. 4. The process starts with the collection of three-phase current data, which are then normalized and transformed using CWT into 2D scalograms. The balanced dataset is divided into subsets for testing, validation, and training during the offline training phase. A pre-trained SqueezeNet model is used via TL to overcome data limitation, and the network is fine-tuned to adapt to certain fault identification tasks. Finally, in the online application stage, real-time current signals are converted into CWT scalograms and processed by the trained model. Upon detecting a fault, the system generates a diagnostic report and, if necessary, activates fault resilience mechanisms to ensure system reliability.

3.1. Data Generation

The data acquisition procedure is designed to record the dynamic behavior of the VSI under a variety of operating conditions in order to guarantee that the diagnostic model demonstrates strong generalization capabilities. Three-phase output currents (i_{abc}) are sampled while key system parameters, mainly the modulation index (ma) and the load factor, are systematically varied, as compared with depending on a single operating point. For every defined fault class k, these variations are organized into a structured feature matrix, where each element corresponds to a specific sampling window in the time domain. The mathematical structure of this dataset generation is expressed as:

$$Class_k = \begin{bmatrix} I_{abc}(ma_1, LF_1) & I_{abc}(ma_1, LF_2) & \cdots & I_{abc}(ma_1, LF_n) \\ I_{abc}(ma_2, LF_1) & I_{abc}(ma_2, LF_2) & \cdots & I_{abc}(ma_2, LF_n) \\ \vdots & \vdots & \ddots & \vdots \\ I_{abc}(ma_m, LF_1) & I_{abc}(ma_m, LF_2) & \cdots & I_{abc}(ma_m, LF_n) \end{bmatrix} \quad (6)$$

where m and n represent the discrete variation steps for the load factor and modulation index, respectively, ensuring that the dataset covers transient fault characteristics across the whole operating envelope. Following data acquisition, a systematic preprocessing pipeline is executed. Initially, all signals are normalized to the [-1, 1] range using amplitude scaling. This phase is essential for eliminating numerical bias towards larger magnitudes, hence ensuring training stability and convergence. The CWT employing the Haar mother wavelet is subsequently utilized to transform the normalized 1D data into 2D time-frequency scalograms. The scalograms are subsequently adjusted to accurately conform to the input dimension specifications of the SqueezeNet architecture.

3.2. Offline Model Training

Due to the limited availability of labeled data in industrial environments, training a deep model from

scratch, where weights are typically initialized randomly, is challenging and often leads to slow convergence or overfitting. To address this, this study utilizes a TL strategy with the pre-trained SqueezeNet architecture. As depicted in Fig. 5, Rather than initiating the training process with random parameters, the model is initialized with optimized weights acquired from the extensive ImageNet dataset. As a result, the final classification layers are updated to identify the specific fault categories of the VSI, while the first layers are frozen to preserve these general characteristics. Even on small datasets without data augmentation, the network achieves good accuracy and robustness by using these pre-optimized weights, which drastically reduce the number of iterations needed for convergence.

The execution of the training follows a defined and systematic procedure. The dataset is first divided into training, validation, and testing subsets. The Adam optimizer efficiently modifies model weights based on computed errors, thereby enhancing network performance. The cross-entropy loss function quantifies the disparity between actual fault labels and projected probabilities, facilitating the minimization of this distance during the learning process. The Softmax activation function is utilized in the final layer to transform the raw outputs into probability scores for each of the 10 fault categories. To make sure the network learns reliable characteristics rather than just memorization of the training data, the validation set is regularly checked during the training phase to assess the model's capacity for generalization and identify any possible cases of overfitting. The reliability of the fully trained model for real-world diagnosis is then confirmed through an extensive evaluation utilizing the unseen testing dataset.

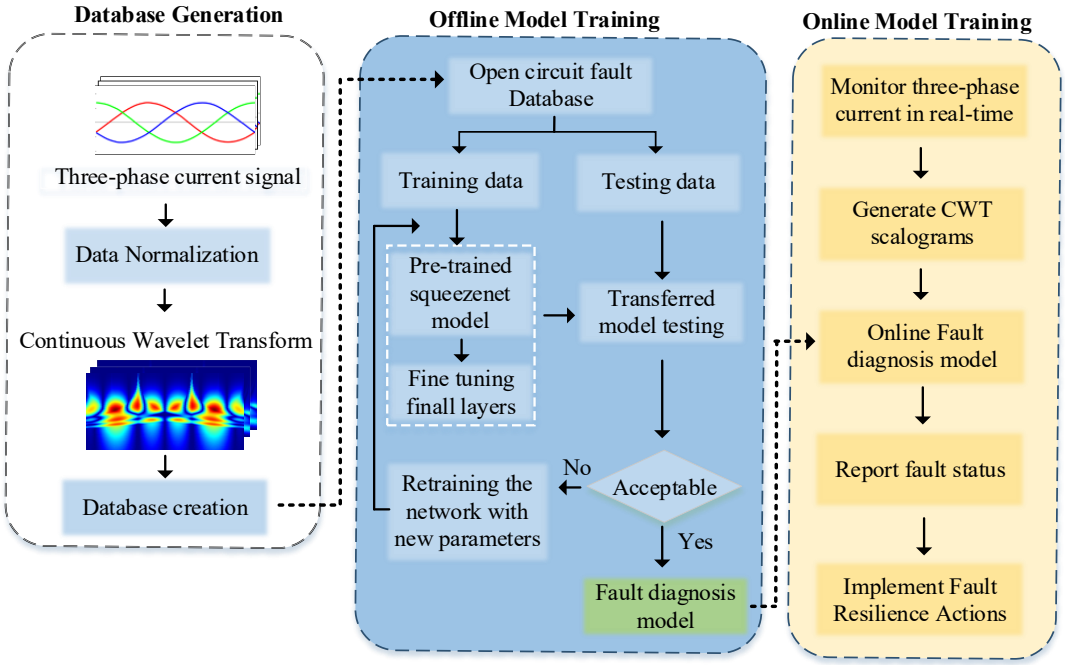


Fig. 4. Overall flowchart of the proposed fault diagnosis framework.

3.3. Online Fault Diagnosis

Throughout the online application phase, the inverter's three-phase output currents are continuously monitored in real time. The raw 1D signals are rapidly transformed into CWT scalograms for explaining their transitory time-frequency features. The produced scalograms thereafter function as inputs to the enhanced SqueezeNet model, which was trained in the offline phase. The model analyzes these images to categorize the system's condition as either healthy or indicative of various fault types. Upon identifying a fault, the system produces a diagnostic report, facilitating the swift implementation of fault tolerance measures to guarantee system reliability and In this article, the performance of the model was safety. evaluated in detail as various evaluation metrics, such as accuracy, precision, recall, and F1 score, were used. These metrics are widely applied to evaluate classification models. While they are intrinsically made for binary classification, their definitions and computations are adjusted when they are applied in multiclass cases to handle the presence of many classes. The accuracy metric is constant between binary and multi-class classification simply because it only measures the proportion of correct predictions to the total number of samples. However, other metrics such as precision, recall, and F1 score differ for multi-class classification. Each of these metrics is calculated separately for each class, and then the overall values are combined using methods such as macro-averaging or micro-averaging which add up the scores of the individual classes in various ways to result in the final model performance evaluation across all the classes [40]. The following equations express all metrics as percentages:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

where TP represents true positives, FN denotes false negatives, TN indicates true negatives, and FP represents false positives for each class. Accuracy is calculated as the ratio of correctly classified instances (true positives and

true negatives) to the total number of samples. Precision measures the accuracy of positive predictions for each class k, while Recall indicates the model's ability to detect all positive instances of that class. These are calculated as follows [40]:

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \quad (8)$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \quad (9)$$

To provide a single performance indicator across all classes, the Macro-Precision and Macro-Recall are computed by averaging the metric values over all k classes:

$$MAPrecision = \frac{\sum_{k=1}^{K} Precision_k}{K} \quad (10)$$

$$MARecall = \frac{\sum_{k=1}^{K} Recall_k}{K} \quad (11)$$

Finally, the Macro F1-Score, which represents the harmonic mean of precision and recall, is used to balance the two metrics.

$$MF1Score = 2 * \left(\frac{MAPrecision * MMARecall}{MAPrecision + MARecall}\right) \quad (12)$$

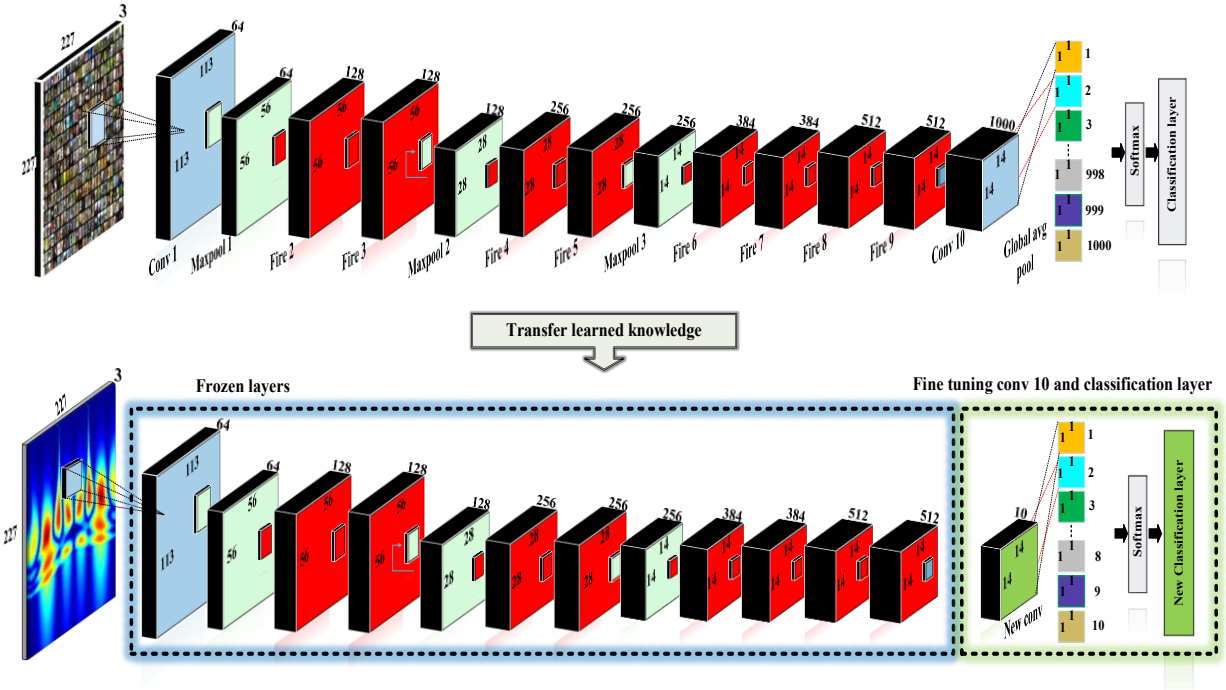


Fig. 5. Modified SqueezeNet architecture for fault classification, highlighting the frozen pre-trained layers and the fine-tuned classification head.

4. SIMULATION AND VERIFICATION

To evaluate the effectiveness of the proposed DTCNN architecture, many simulations were carried out utilizing the MATLAB/Simulink environment for VSI modeling, coupled with the Deep Network Designer toolbox for neural network implementation. The performance benchmarking, specifically comparing the transfer learning approach against a model trained from scratch, was conducted on a workstation equipped with an 11th Gen Intel Core i5-11300H processor, 16 GB of RAM, and an Intel Iris Xe GPU.

The three-phase VSI simulation model used in this investigation is shown in Fig. 6. To simulate a realistic industrial driving system, the circuit parameters shown in Table II were used. Unlike invasive diagnostic methods, this study leverages the existing current sensors embedded in the inverter's control loop to acquire the three-phase output currents, thereby eliminating the need for additional hardware and reducing implementation costs.

4.1. Small database creation

To create the experimental dataset, the VSI three-phase currents were recorded at a frequency of 5 kHz, yielding 100 data points per fundamental cycle (0.02 s). The operational parameters were systematically changed as described in the methodology, with the load factor adjusted in 40 steps (ranging from 0.7 to 1.3 p.u.) and the modulation index set to five distinct values (0.6 to 1.0). This combinatorial approach yielded a total of 2000 samples (200 samples for each of the 10 fault classes). Following acquisition, the raw signals were normalized and transformed into time-frequency scalograms using the

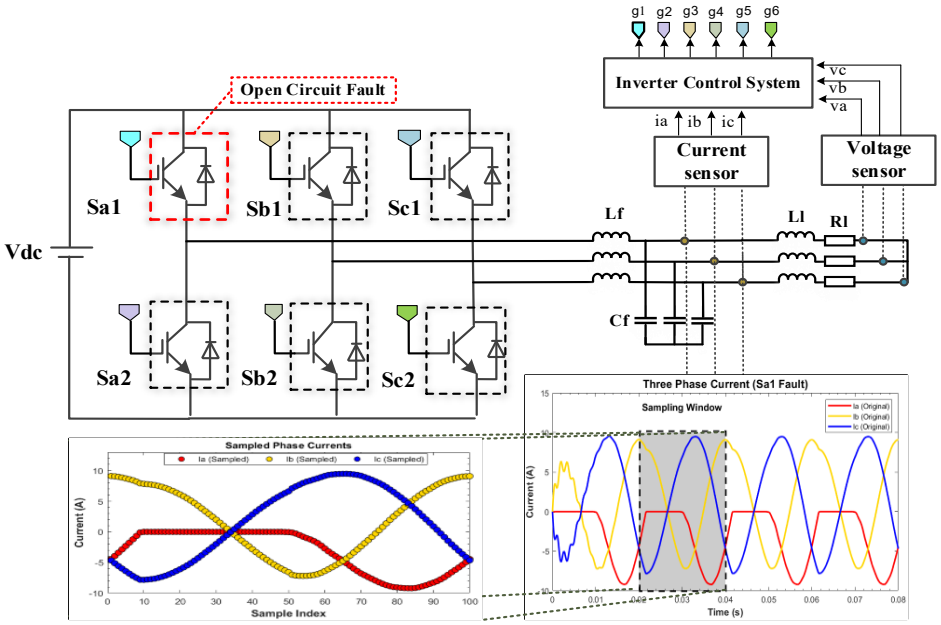


Fig. 6. Simulink model implementation of the three-phase VSI system used for fault data generation.

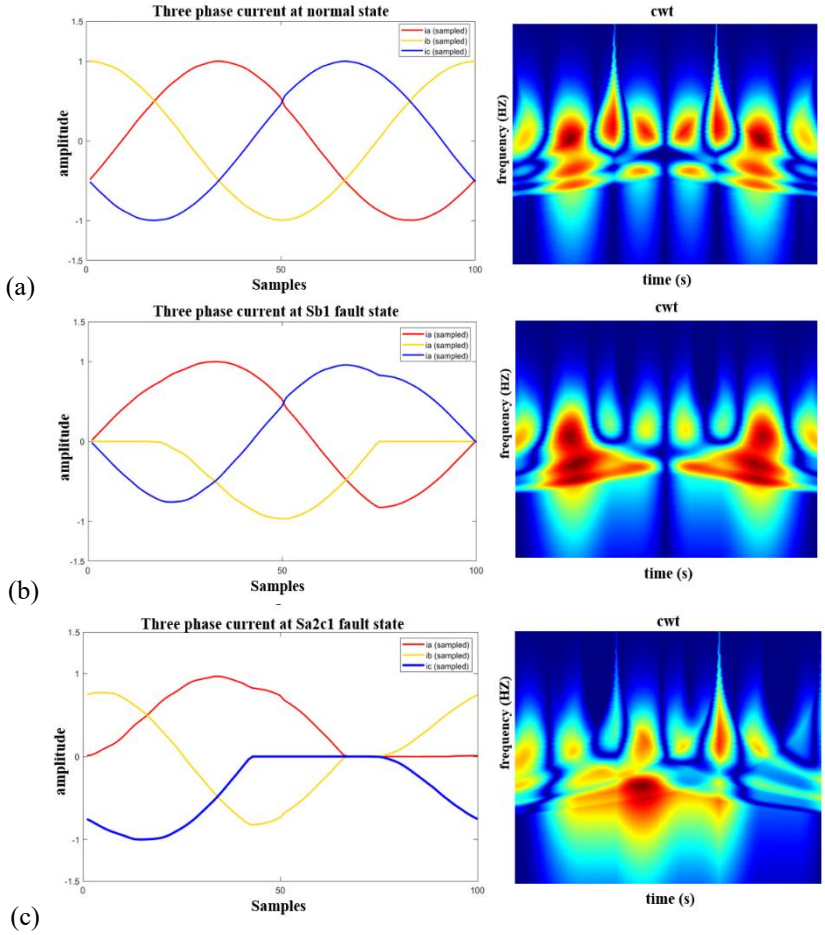


Fig. 7. Visual representation of normalized three-phase current signals and their corresponding CWT scalograms under: a) Normal operating condition, b) Single-switch open-circuit fault, and c) Double-switch open-circuit fault.

Haar wavelet, then resized to 227 ×227 pixels to match the SqueezeNet input requirements. Representative samples of these scalograms under normal and faulty conditions are depicted in Fig. 7.

Table II. System parameters used in the VSI simulation model.

Parameter	Values
DC-link voltage (V_{DC})	100V
Load Inductance (L_l)	5mH
Load Resistance (R_l)	4.7Ω
Inductive Filter (L_f)	4.05mH
Capacitive Filter (C_f)	25μF
Fundamental Frequency (f_f)	50HZ
Switching Frequency (f_{sw})	20KHz
Sampling frequency (f_s)	5KHz

To evaluate the model's robustness against training data availability, the total dataset was stratified into three distinct configurations: dataset A, dataset B, and dataset C, utilizing 80%, 70%, and 60% of the data for training, respectively. The detailed distribution of training, validation, and testing samples for each configuration is provided in Table III. The SqueezeNet model's fine-tuning was finally handled by particular hyperparameters intended to guarantee steady convergence on this small dataset. Using the Adam optimizer and cross-entropy loss function over 10 epochs, a conservative learning rate of 0.0001 and a batch size of 32 were chosen, as shown in Table IV.

Table III. Distribution of training, validation, and testing samples across datasets.

Dataset	Training data	Validation data	Testing data
Dataset A (80%)	1280	320	400
Dataset B (70%)	1120	280	600
Dataset C (60%)	960	240	800

4.2. Detection performance evaluation

To guarantee the reliability of the proposed DTCNN framework, a 5-fold cross-validation strategy was implemented on dataset A. By testing the model on various data segments rather than just one training run, this approach verifies the model's consistency. This rigorous approach ensures that the high accuracy achieved is a genuine reflection of the model's performance, eliminating the risk of results being influenced by a specific data split. By averaging the results over these folds, the potential bias of random data partitioning is removed, confirming the true stability of the diagnostic system.

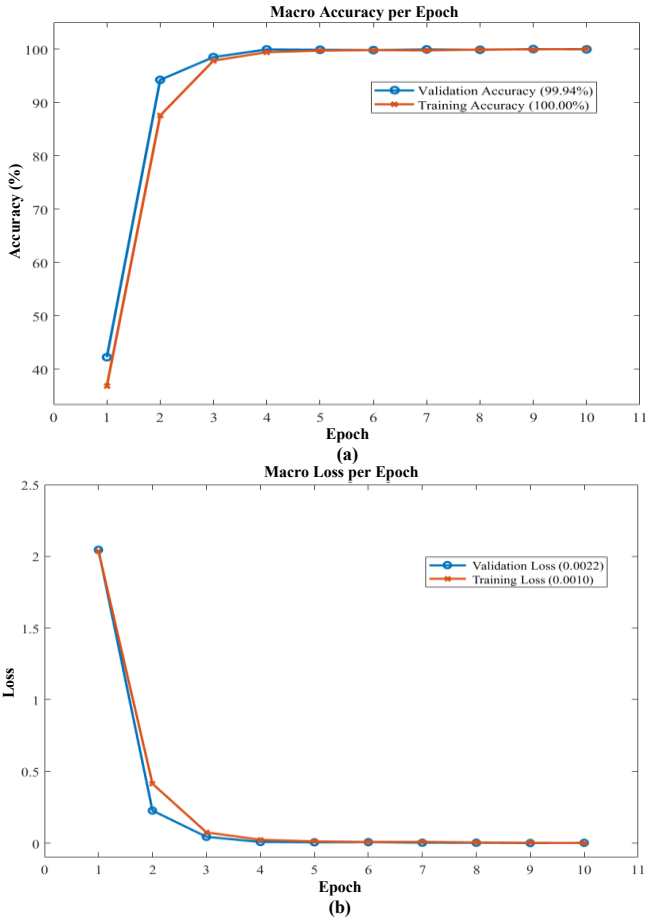
Table IV. Hyperparameter settings for training the SqueezeNet model.

Hyperparameters	Values
Learning rate	0.0001
Epoch	10
Batch size	32
Optimizer	Adam
Loss function	Cross entropy

Fig. 8 displays the accuracy and loss curves for dataset A across 10 training epochs. A notable feature is the model's starting accuracy of approximately 40%, which is significantly higher than a random initialization. The initial advantage arises from the SqueezeNet model being pre-trained on the extensive ImageNet dataset, enabling it to utilize previously acquired feature extraction capabilities while dealing with new inverter fault data.

Fig. 8. Learning performance of the proposed model on dataset A: (a) Training and validation accuracy, and (b) Training and validation loss over epochs.

Consequently, the network converges rapidly, surpassing 90% accuracy by the third epoch and stabilizing at 99.94% validation accuracy by the final epoch.



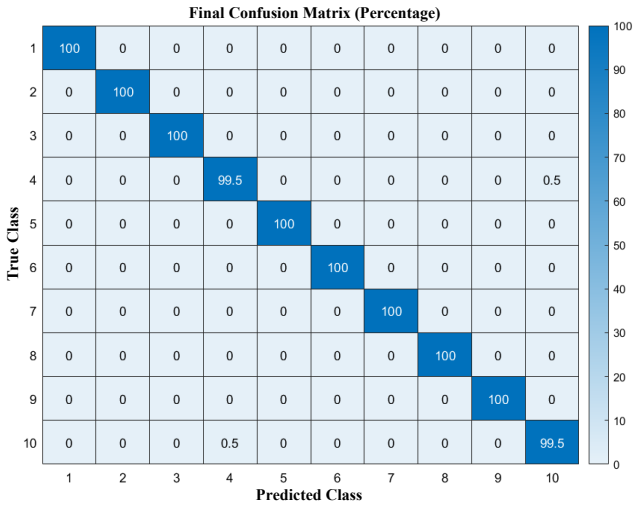
Furthermore, the loss values steadily decrease to nearly zero (0.0022), and the close alignment between the training and validation curves confirms that the model has

successfully adapted to the fault patterns without overfitting.

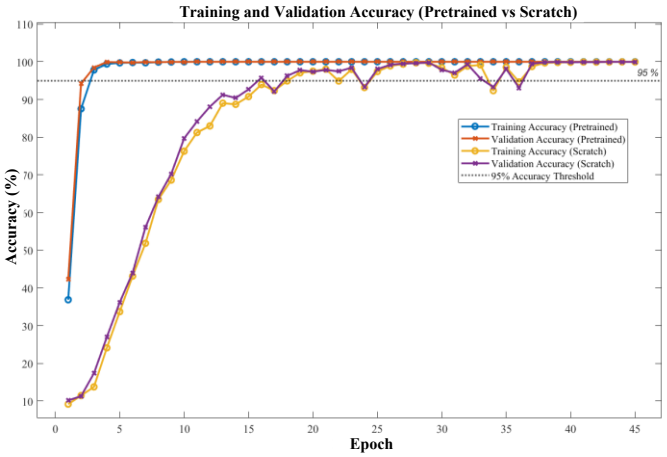
The confusion matrix averaged over the five cross-validation folds is shown in Fig. 9 to illustrate the model's classification performance on test data. For the majority of categories, the diagonal figures show 100% accurate classification rates, indicating the high reliability of the proposed method. There is only a very slight 0.5% misclassification rate between Classes 4 and 10. This minor error is technically attributed to the high physical similarity between the current waveforms of these specific fault scenarios, which makes them slightly difficult to distinguish compared to other distinct fault types.

Fig. 9. Aggregate confusion matrix of the proposed method on dataset A, averaged over 5-fold cross-validation.

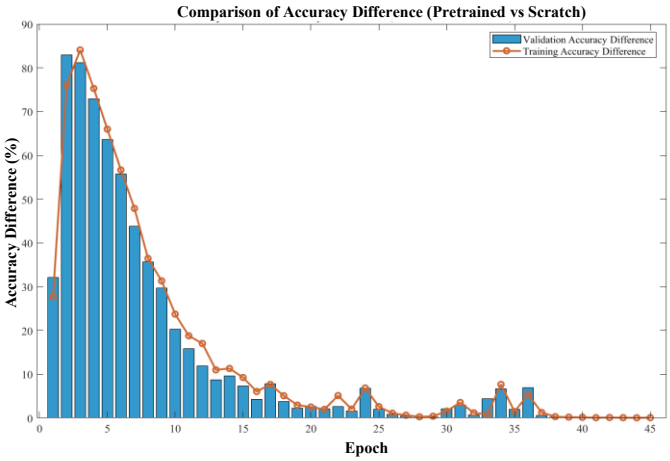
Table V provides an accurate quantitative assessment of precision, recall, and F1-score for each fault type. The



results verify the model's remarkable reliability, with perfect scores (100%) in 8 of the 10 categories. Only Class 4 and Class 10 had minor declines, but the metrics remained extremely robust, exceeding 99.50%. The model achieved a macro-average accuracy of 99.90%, empirically validating its efficacy for high-precision diagnosis in manufacturing settings. In conclusion, a thorough examination using 5-fold cross-validation, convergence analysis, and confusion matrix inspection confirms that the fine-tuned SqueezeNet is suitable for this diagnostic task. The model learns fast and maintains high precision across a wide range of fault severity with minimal error. After establishing the absolute performance of the proposed framework, the following section compares its relative efficiency to a typical training technique to emphasize the special benefits of the transfer learning strategy.



(a)



(b)

Table V. Detailed classification performance metrics for dataset A.

Class	Precision %	Recall %	F1-Score %	Accuracy %
Class 1	100	100	100	100
Class 2	100	100	100	100
Class 3	100	100	100	100
Class 4	99.50	99.50	99.50	99.50
Class 5	100	100	100	100
Class 6	100	100	100	100
Class 7	100	100	100	100
Class 8	100	100	100	100
Class 9	100	100	100	100
Class 10	99.50	99.50	99.50	99.50
Macro	99.90	99.90	99.90	99.90

4.3. Comparative Analysis

The proposed framework was thoroughly evaluated with a two-phase comparative study. The finely tuned model was compared to a standard SqueezeNet that was trained from scratch to determine the effectiveness of the transfer learning technique. The second step was to compare the proposed lightweight framework with several existing

deep learning architectures to confirm its superiority in terms of accuracy, efficiency, and fault coverage.

Fig. 10 compares the learning process between the two training strategies and provides a visual quantification of the performance difference between them. Using the generic features transferred from ImageNet, the pre-trained model exhibits a superior convergence trajectory, as shown in Fig. 10a. It efficiently bypasses the first feature-learning stage and reaches an impressive accuracy of over 95% in the first 4 epochs. On the other hand, the model that was trained from scratch has a slow start at about 10% accuracy, necessitating a long iterative process to learn significant patterns.

Also, the accuracy gap between the two models can be seen over time in Fig. 10b, which provides a more detailed look at the efficiency differential. There is a significant gap at the beginning of training, indicating that the pre-trained weights have an immediate advantage. However, this difference disappears as the scratch model, starting with random initializations, gradually learns feature patterns and updates over time. It is notable that the gap is very small by epoch 30. This confirms that while the pre-trained model stabilized almost immediately, the scratch model lagged behind significantly, requiring an additional 30 epochs of training just to reach a comparable performance level.

Table VI details the specific training requirements and resources needed to maintain the stability of both models. The results indicate that the entire training phase for the pre-trained model was completed in only 10 epochs (245 seconds), whereas the scratch model required 45 epochs

Fig. 10. Performance comparison between transfer learning and training from scratch: (a) Accuracy progression over epochs, (b) Accuracy differential highlighting the convergence speed.

Table VI. Performance comparison: Pre-trained SqueezeNet vs. model trained from scratch.

Metric	Pretrained Model	From Scratch Model
No. of Epochs	10	45
Total Iterations	400	1800
Training Accuracy	100	99.98
Validation Accuracy	99.94 %	99.95 %
Training Loss	0.001	0.0041
Validation Loss	0.0022	0.0035
Final Accuracy	99.90 %	99.95 %
Total Training Time	245 s	1100 s

(1100 seconds) to ensure consistent performance. Importantly, the suggested TL-based SqueezeNet achieved nearly the same precision of 99.90% while requiring 78% less computational time, whereas the scratch model achieved a slightly higher accuracy of 99.95%. This result demonstrates that the proposed framework achieves comparable high-precision performance at a lower cost by utilizing TL, making it the far superior approach for industrial deployment. The suggested method was also compared with established ML and DL approaches in the literature to illustrate its advantages. Table VII presents a detailed comparison of

Table VII. Comparative analysis of the proposed framework against state-of-the-art methods.

Ref.	Methodology	Feature Extraction / Learning	Strategy for Limited Data	Model Complexity (Params)	Accuracy	Noise Robustness	Remarks/ Limitation
[15]	Ensemble Bagged Tree	Manual	None	Low (Model) / High Effort	99.70%	Moderate	Relies on handcrafted features; limited adaptability under varying conditions.
[22]	CWT + AlexNet	Automatic (Deep CNN)	None	Very High	98.86%	High	Computationally heavy; unsuitable for embedded/real-time deployment.
[27]	ViT + CGAN	Automatic (Transformer)	Data Generation (GAN)	Very High	98.46%	High	Extremely complex architecture; GAN instability; expensive training and inference.
[26]	1D-CNN + WAC-GAN	Automatic (1D-CNN)	Data Generation (GAN)	High	93.71%	Moderate	GAN convergence issues; 1D features miss detailed time–frequency patterns.
[21]	Improved 1D-CNN	Automatic (1D-CNN)	Sample Amplification	Low	99.20%	Moderate	Limited capability in modeling complex 2D transient characteristics; limited generalization
Proposed	CWT + SqueezeNet + TL	Automatic (Deep CNN via TL)	Transfer Learning	Very Low	99.90%	High	Optimal balance of accuracy, speed, and stability; highly suitable for real-time industrial use.

various methodologies. Although traditional ML techniques like the Ensemble Bagged Tree [15] attain exceptional accuracy, they depend significantly on manually produced features, limiting their adaptability under varying operational conditions. In contrast, deep CNN-based methods such as CWT–AlexNet [22] automate feature extraction but suffer from extensive model complexity, making them unsuitable for embedded or real-time applications. To address data scarcity, recent advanced studies such as ViT–CGAN [27] and WAC-GAN–1DCNN [26] have employed GAN-based sample generation; however, these approaches introduce unstable training dynamics and substantial architectural overhead. Although improved 1D-CNN models [21] are computationally lighter, their inability to capture rich 2D transient time–frequency structures restricts their generalization. In sharp contrast, the proposed SqueezeNet-based transfer learning framework offers an optimal trade-off between accuracy and efficiency. With only 1.24M parameters, it delivers the highest accuracy (99.90%) and strong robustness to noise, demonstrating its clear superiority for real-time industrial fault diagnosis under limited data.

4.4. Robustness of Models to Dataset Variations and Noise

This section analyzes the models' robustness in a variety of circumstances, with particular focus on distinct data distributions and the addition of Gaussian white noise (AWGN) at various levels to simulate real-world scenarios. This study is an essential instrument for evaluating the models' flexibility and adaptability to natural variations and noise that accumulate data in real industrial environments.

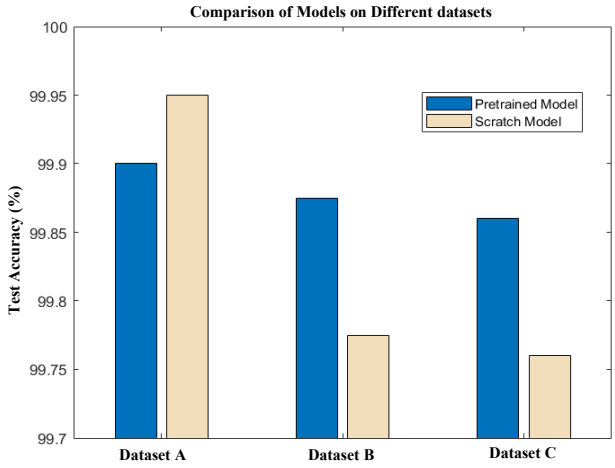


Fig. 11. Accuracy comparison between the proposed transfer learning model and the model trained from scratch across datasets A, B, and C.

To evaluate the model's robustness to limited data availability, training was performed on datasets of varying sizes, as shown in Fig. 11. The scratch model outperforms the pre-trained model (99.95% vs. 99.90%) on the largest subset, dataset A (1,280 training samples). However, as the amount of training data decreases, this trend reverses. On dataset B (1,120 samples) and dataset C (only 960 samples), the pre-trained model demonstrates greater stability and outperforms the scratch model, achieving accuracies of 99.88% and 99.86%, respectively. The pre-trained model attains these higher accuracies in about 10 epochs, but the scratch model necessitates considerably more training duration to achieve identical results. The findings validate that the transferred knowledge effectively mitigates the constraints of limited data availability and improves overall model resilience.

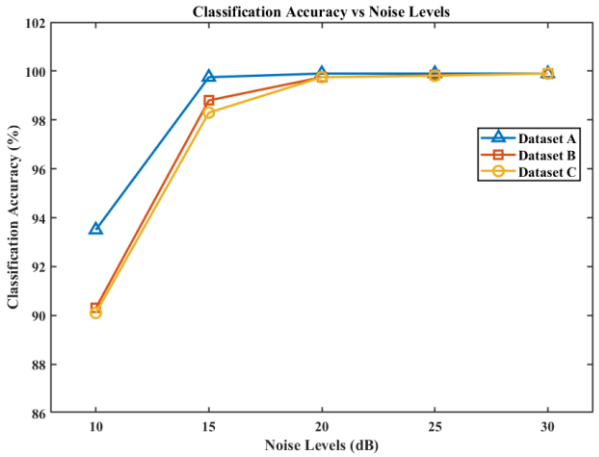


Fig. 12. Evaluating the accuracy of the proposed model against noise using different datasets.

To assess the model's robustness against real electromagnetic interference, test signals were exposed to AWGN at Signal-to-Noise Ratios (SNR) from 30 dB to 10 dB. The proposed framework displays remarkable resilience in this spectrum, as demonstrated in Fig. 12. Classification accuracy is consistently excellent, reaching 99.8% across all dataset configurations at SNR levels of 20 dB or higher, capturing the noise-free baseline. Crucially, even at 15 db, the model remains highly stable, with an accuracy of more than 98%. Only at 10 dB of noise does a notable performance reduction exist. The model trained on dataset A retains a robust accuracy of 93.5%, but models trained on smaller subsets (dataset B and C) drop to about 90%. These results confirm that the combination of CWT scalograms and SqueezeNet feature extraction effectively filters out noise, ensuring reliable diagnosis even in severe operating environments.

5. Conclusion

This study establishes a robust and computationally efficient framework for diagnosing open-circuit faults in Voltage-Source Inverters (VSIs) by synergizing the time-frequency localization of the Continuous Wavelet Transform (CWT) with the advanced feature extraction capabilities of a pre-trained SqueezeNet. A key advantage of this methodology lies in its ability to extract rich, discriminative features from a limited dataset of just 2,000 samples, effectively eliminating the reliance on complex data augmentation or synthetic data generation techniques often required in deep learning. This capability enables the system to identify complex single and simultaneous switch fault patterns with exceptional precision, even when labeled industrial data is scarce. The quantitative simulation results decisively validate the superiority of the transfer learning strategy. The fine-tuned model achieved a remarkable classification accuracy of 99.90% in a training duration of only 245 seconds (10 epochs).

This represents a ~78% reduction in computational time compared to training the model from scratch, which required 1100 seconds to reach a comparable level. Beyond computational efficiency, the model proved to be highly reliable; it maintained an accuracy exceeding 98% even under significant electromagnetic interference (15 dB) and reduced training data availability (60%). This stability makes it an ideal candidate for real-time embedded implementation.

Given that this study primarily focused on diagnosing internal faults under controlled operating conditions, future research is recommended to extend the framework to grid-connected scenarios. Specifically, further investigation is required to distinguish between intrinsic component failures and external anomalies, such as grid disturbances or Low Voltage Ride Through (LVRT) events. Additionally, to enhance the system's adaptability to unforeseen failure modes, it is suggested to develop mechanisms for identifying "unknown faults," thereby ensuring reliable diagnosis even when facing unclassified or novel fault patterns.

6. References

[1] B. Liu, T. Shi, Y. Yan, G. Zhang, and C. Xia, "Open-Circuit Fault Diagnosis Method for Inverters Based on Critical Characteristics of Half-Wave Current," *IEEE Trans. Power Electron.*, 2025.

[2] Y. Luo, L. Zhang, C. Chen, K. Li, and K. Li, "Real-time diagnosis of open circuit faults in three-phase voltage source inverters," *IEEE Trans. Power Electron.*, vol. 39, no. 6, pp. 7572–7585, 2024.

[3] Y. Ajra, G. Hoblos, H. Al Sheikh, and N. Moubayed, "A Literature Review of Fault Detection and Diagnostic Methods in Three-Phase Voltage-Source Inverters," *Machines*, vol. 12, no. 9, p. 631, 2024.

[4] S. Xu *et al.*, "Multiple open-switch fault diagnosis for three-phase four-leg inverter under unbalanced loads via interval sliding mode observer," *IEEE Trans. power Electron.*, vol. 39, no. 6, pp. 7607–7619, 2024.

[5] F. Naseri, E. Schaltz, K. Lu, and E. Farjah, "Real-time open-switch fault diagnosis in automotive permanent magnet synchronous motor drives based on Kalman filter," *IET Power Electron.*, vol. 13, no. 12, pp. 2450–2460, 2020.

[6] J. A. Reyes-Malanche, F. J. Villalobos-Pina, E. Cabal-Yepez, R. Alvarez-Salas, and C. Rodriguez-Donate, "Open-circuit fault diagnosis in power inverters through currents analysis in time domain," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[7] F. J. Villalobos-Pina, J. A. Reyes-Malanche, E. Cabal-Yepez, and E. Ramirez-Velasco, "Open Circuit Fault Diagnosis in Induction Motor Driver Inverter," in *Induction Motors-Latest Research and Applications*, IntechOpen, 2024.

[8] M. Dale, V. H. Kamble, R. B. Dhumale, and A. Nanthaamornphong, "Open Switch Fault Diagnosis in Three-Phase Voltage Source Inverters Using Single Neuron Implementation," *Processes*, vol. 13, no. 4, p. 1070, 2025.

[9] S. Ding, M. Hao, Z. Cui, Y. Wang, J. Hang, and X. Li, "Application of multi-SVM classifier and hybrid GSAPSO algorithm for fault diagnosis of electrical machine drive system," *ISA Trans.*, vol. 133, pp. 529–538, 2023.

[10] S. Changli, Y. He, Q. Huo, Z. Liu, and H. Xue, "Inverter open circuit fault diagnosis strategy based on KNN algorithm," in *2024 IEEE 10th International Power Electronics and Motion Control Conference (IPEMC2024-ECCE Asia)*, IEEE, 2024, pp. 1453–1456.

[11] T. Wang, H. Xu, J. Han, E. Elbouchikhi, and M. E. H. Benbouzid, "Cascaded H-bridge multilevel inverter system fault diagnosis using a PCA and multiclass relevance vector machine approach," *IEEE Trans. Power Electron.*, vol. 30, no. 12, pp. 7006–7018, 2015.

[12] Y. Xia, Y. Xu, and B. Gou, "A data-driven method for IGBT open-circuit fault diagnosis based on hybrid ensemble learning and sliding-window classification," *IEEE Trans. Ind. Informatics*, vol. 16, no. 8, pp. 5223–5233, 2019.

[13] R. Shan, J. Yang, and S. Huang, "Open-circuit fault diagnosis of three-phase PWM rectifier circuits based on transient characteristics and random forest classification," *J. Power Electron.*, vol. 24, no. 1, pp. 130–139, 2024.

[14] L. Kou, C. Liu, G. Cai, J. Zhou, and Q. Yuan, "Data-driven design of fault diagnosis for three-phase PWM rectifier using random forests technique with transient synthetic features," *IET Power Electron.*, vol. 13, no. 16, pp. 3571–3579, 2020.

[15] C. N. Ibem, M. E. Farrag, A. A. Aboushady, and S. M. Dabour, "Multiple open switch fault diagnosis of three phase voltage source inverter using ensemble bagged tree machine learning technique," *IEEE access*, vol. 11, pp. 85865–85877, 2023.

[16] M. A. Gultekin and A. Bazzi, "Review of fault detection and diagnosis techniques for AC motor drives," *Energies*, vol. 16, no. 15, p. 5602, 2023.

[17] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mech. Syst. Signal Process.*, vol. 138, p. 106587, 2020.

[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, 2018.

[19] 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, pp. 1–11, 2022.

[20] B. Cui, S. Zhang, J. Su, and H. Cui, "Fault diagnosis for inverter-fed motor drives using one dimensional complex-valued convolutional neural network," *IEEE Access*, 2024.

[21] H. Shen, X. Tang, Y. Luo, F. Xie, and Z. Shi, "Online open-circuit fault diagnosis for neutral point clamped inverter based on an improved convolutional neural network and sample amplification method under varying operating conditions," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, 2024.

[22] Q. Sun, X. Yu, H. Li, F. Peng, and G. Sun, "Fault detection for power electronic converters based on continuous wavelet transform and convolution neural network," *J. Intell. Fuzzy Syst.*, vol. 42, no. 4, pp. 3537–3549, 2022.

[23] A. Sivapriya, N. Kalaiarasi, R. Verma, B. Chokkalingam, and J. L. Munda, "Fault diagnosis of cascaded multilevel inverter using multiscale kernel convolutional neural network," *IEEE Access*, 2023.

[24] Y. Zhang, Z. Cheng, Z. Wu, E. Dong, R. Zhao, and G. Lian, "Research on Electronic Circuit Fault Diagnosis Method Based on SWT and DCNN-ELM," *IEEE Access*, 2023.

[25] J. Hang, X. Shu, S. Ding, and Y. Huang, "Robust open-

circuit fault diagnosis for PMSM drives using wavelet convolutional neural network with small samples of normalized current vector trajectory graph," *IEEE Trans. Ind. Electron.*, vol. 70, no. 8, pp. 7653–7663, 2023.

[26] Q. Sun, F. Peng, X. Yu, and H. Li, "Data augmentation strategy for power inverter fault diagnosis based on wasserstein distance and auxiliary classification generative adversarial network," *Reliab. Eng. Syst. Saf.*, vol. 237, p. 109360, 2023.

[27] M. S. Mahmoud, A. Salem, and K. G. Robbersmyr, "A few–shot open-circuit fault diagnosis of F-type inverters using CGAN-based vision transformer," *IEEE J. Emerg. Sel. Top. Power Electron.*, 2024.

[28] Y. Jiang, L. Chen, M. Wang, L. Wei, and M. Li, "Adaptive sparse attention wavelet network for the robust open-circuit fault diagnosis in PMSM drives," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024.

[29] M. S. Priyadarshini, M. Bajaj, L. Prokop, and M. Berhanu, "Perception of power quality disturbances using Fourier, Short-Time Fourier, continuous and discrete wavelet transforms," *Sci. Rep.*, vol. 14, no. 1, p. 3443, 2024.

[30] S. Djaballah, K. Meftah, K. Khelil, and M. Sayadi, "Deep transfer learning for bearing fault diagnosis using CWT time–frequency images and convolutional neural networks," *J. Fail. Anal. Prev.*, vol. 23, no. 3, pp. 1046–1058, 2023.

[31] D. Łuczak, "Machine fault diagnosis through vibration analysis: Continuous wavelet transform with complex morlet wavelet and time–frequency RGB image recognition via convolutional neural network," *Electronics*, vol. 13, no. 2, p. 452, 2024.

[32] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[33] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, 2017.

[34] 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, pp. 1–15, 2021.

[35] S. Azad and M. T. Ameli, "An Imbalanced Deep Learning Framework for Pre-Fault Flexible Multi-Zone Dynamic Security Assessment Via Transfer Learning Based Graph Convolutional Network," *Results Eng.*, p. 104172, 2025.

[36] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.

[37] Z. Chen, J. Cen, and J. Xiong, "Rolling bearing fault diagnosis using time-frequency analysis and deep transfer convolutional neural network," *Ieee Access*, vol. 8, pp. 150248–150261, 2020.

[38] F. N. Iandola, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," *arXiv Prepr. arXiv1602.07360*, 2016.

[39] M. Dai and Z. Huang, "Research on fault diagnosis of drilling pump fluid end based on time-frequency analysis and convolutional neural network," *Processes*, vol. 12, no. 9, p. 1929, 2024.

[40] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv Prepr. arXiv2008.05756*, 2020.