*electronics*

MDPI

*Article*

# DGA-Based Fault Diagnosis Using Self-Organizing Neural Networks with Incremental Learning

**Siqi Liu *, Zhiyuan Xie and Zhengwei Hu**

School of Electrical and Electronic Engineering, North China Electric Power University, Baoding 071003, China; zhiyuanxie@ncepu.edu.cn (Z.X.); hzwwizard@ncepu.edu.cn (Z.H.)
* Correspondence: liusiqi@ncepu.edu.cn

**Abstract:** Power transformers are vital components of electrical power systems, ensuring reliable and efficient energy transfer between high-voltage transmission and low-voltage distribution networks. However, they are prone to various faults, such as insulation breakdowns, winding deformations, partial discharges, and short circuits, which can disrupt electrical service, incur significant economic losses, and pose safety risks. Traditional fault diagnosis methods, including visual inspection, dissolved gas analysis (DGA), and thermal imaging, face challenges such as subjectivity, intermittent data collection, and reliance on expert interpretation. To address these limitations, this paper proposes a novel distributed approach for multi-fault diagnosis of power transformers based on a self-organizing neural network combined with data augmentation and incremental learning techniques. The proposed framework addresses critical challenges, including data quality issues, computational complexity, and the need for real-time adaptability. Data cleaning and preprocessing techniques improve the reliability of input data, while data augmentation generates synthetic samples to mitigate data imbalance and enhance the recognition of rare fault patterns. A two-stage classification model integrates unsupervised and supervised learning, with k-means clustering applied in the first stage for initial fault categorization, followed by a self-organizing neural network in the second stage for refined fault diagnosis. The self-organizing neural network dynamically suppresses inactive nodes and optimizes its training parameter set, reducing computational complexity without sacrificing accuracy. Additionally, incremental learning enables the model to continuously adapt to new fault scenarios without modifying its architecture, ensuring real-time performance and adaptability across diverse operational conditions. Experimental validation demonstrates the effectiveness of the proposed method in achieving accurate, efficient, and adaptive fault diagnosis for power transformers, outperforming traditional and conventional machine learning approaches. This work provides a robust framework for integrating advanced machine learning techniques into power system monitoring, paving the way for automated, real-time, and reliable transformer fault diagnosis systems.

**Keywords:** multi-fault diagnosis; power transformers; artificial intelligence; dissolved gas analysis (DGA)

check for updates

Academic Editor: Davide Astolfi

Received: 30 December 2024
Revised: 15 January 2025
Accepted: 16 January 2025
Published: 22 January 2025

**Citation:** Liu, S.; Xie, Z.; Hu, Z. DGA-Based Fault Diagnosis Using Self-Organizing Neural Networks with Incremental Learning. *Electronics* **2025**, *14*, 424. https://doi.org/10.3390/electronics14030424

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Power transformers play a crucial role in electrical power systems, acting as the interface between high-voltage transmission and low-voltage distribution networks [1,2]. They are essential for the reliable and efficient transfer of electrical energy over long distances. As global energy demand continues to increase, ensuring the smooth operation

of power transformers becomes even more critical. However, like all electrical machinery, power transformers are susceptible to faults, which can range from insulation breakdowns and winding deformations to more complex failures such as partial discharge (PD) and short circuits [3–5]. These faults can have serious consequences, including the disruption of electrical service, economic losses due to extended downtimes, and even safety hazards.

The failure of a power transformer can lead to significant financial losses. According to industry reports, transformer failures in electrical networks can cost utilities millions of dollars annually due to both the repair costs and the cascading impact on grid stability [6]. In addition, failure events often lead to power outages, which can disrupt industrial processes, commercial operations, and household services [7]. To prevent such catastrophic failures, early detection and diagnosis of faults in transformers are essential. A timely diagnosis can provide a window for preventive maintenance, reduce the need for costly repairs, and ultimately extend the transformer's service life.

Traditionally, fault diagnosis in transformers has been achieved through methods like visual inspection [8], oil sampling [9], dissolved gas analysis (DGA) [10], and thermal imaging [11]. However, these methods often have limitations. For example, visual inspection is subjective and may not detect latent faults, while oil sampling provides only intermittent information. Thermal imaging is also a non-invasive technique but does not provide in-depth analysis of fault mechanisms, particularly in complex electrical or mechanical failures [12]. These methods also tend to rely heavily on the expertise of operators, which may lead to human errors or oversights.

Given these limitations, there is a growing demand for automated, reliable, and real-time fault diagnosis systems. These systems are expected to detect incipient faults at an early stage, well before they escalate into major failures. With advancements in sensor technology and data analytics, there is now an opportunity to develop predictive fault diagnosis systems that can analyze a variety of operational parameters continuously. Over the years, several advanced methods have been proposed to detect transformer faults using data-driven approaches. These methods leverage a variety of data sources, including vibration [13], temperature [13], acoustic signals [14], and electrical measurements [15] such as current and voltage waveforms. Among these techniques, machine learning algorithms have gained popularity. They can analyze large volumes of data to detect patterns indicative of faults in a more automated, reliable, convenient, and accurate way. In recent years, deep learning has made significant progress in the field of visual inspection, especially in detecting insulation defects in transformers. For example, one study [16] designed a robotic inspection fish that uses deep learning networks to visually inspect the insulation paper inside transformers, identifying discharge carbon marks and cracks. Furthermore, the transformer architecture has demonstrated high fidelity in structural defect detection [17]. By incorporating learnable down-sampling and up-sampling modules, transformers achieve an excellent balance between global and local semantic information. These studies indicate that deep learning techniques, especially visual analysis methods, have been successfully applied to the automated health evaluation of transformers. Including these methods in the introduction aids in comprehensively showcasing the potential of deep learning in transformer condition monitoring.

However, several challenges remain:

- Data quality and imbalance issues: The quality of the data used in DGA can be compromised due to factors such as noise in gas concentration readings, instrument limitations, and sampling inconsistencies. These issues can negatively impact the performance of AI models in fault detection. Moreover, in many cases, fault data for specific failure modes are sparse, leading to a data imbalance that causes AI models to underperform, particularly in fault diagnosis for rare transformer issues.

- High computational complexity: Although using neural networks can improve diagnostic accuracy, the neural network models are complex and have high computational complexity.
- Real-time performance and adaptability: DGA data typically require significant time to be collected and processed, which can affect the real-time capabilities of fault diagnosis systems. Furthermore, transformer operating conditions can vary significantly across regions, affecting model performance and generalizability.

To address the above issues, this paper proposes a self-organizing neural network for multi-fault diagnosis of power transformers, combining data augmentation and incremental learning methods. The following key points characterize the originality of this contribution:

- A two-stage classification model that combines unsupervised and supervised methods is proposed. In the first stage, the k-means algorithm is used for unsupervised classification. Based on the classification results, a self-organizing neural network is trained in the second stage for fault diagnosis.
- A self-organizing neural network approach is proposed, which adaptively suppresses the activated nodes of the neural network based on the training process, dynamically constructing the neural network training parameter set, thereby significantly reducing the model's computational complexity while ensuring accuracy.
- The introduction of incremental learning allows the model to continuously learn new knowledge without altering the neural network architecture, improving the real-time performance and adaptability of the model.

The rest of this paper is organized as follows. Section 2 provides a review of related work in the field of transformer fault diagnosis, including traditional and machine learning-based methods. Section 3 describes the methodology for the proposed fault diagnosis system, detailing the data collection process, feature extraction techniques, and incremental learning. Section 4 presents the simulation results and outperforms the benefits of the proposed framework. Finally, we conclude the paper and discuss the future work in Section 5.

## 2. Related Work

The use of DGA for diagnosing faults in power transformers has been a standard practice for decades. By analyzing the concentrations of various gases dissolved in transformer oil, DGA provides insights into the internal health of transformers and can help in identifying fault types such as overheating, partial discharge, and short circuits. DGA is considered a reliable technique for early fault detection, but traditional methods of analysis often rely on heuristic rules and expert knowledge, making them subjective and limited in terms of scalability and automation. As such, there has been increasing interest in integrating artificial intelligence (AI) techniques with DGA to improve fault detection accuracy, automate the diagnostic process, and reduce the dependence on expert judgment.

In recent years, machine learning algorithms, such as support vector machines (SVM), decision trees (DT), and k-means, have been widely applied to DGA data for transformer fault diagnosis. These methods can classify fault types based on the concentration of dissolved gases, offering a significant advantage over traditional approaches by reducing human involvement in the analysis process. For example, Zhang et al. [18] applied an SVM-based model for classifying transformer faults using DGA data and achieved a high accuracy rate of over 90%. Their results demonstrated that machine learning models could effectively handle complex patterns in gas concentrations, which are often challenging for conventional methods. Similarly, in a study by Xiao et al. [19], a decision tree algorithm was used to classify fault conditions in transformers. The study showed that decision trees could accurately categorize faults such as arcing and overheating with high sensitivity.

Nanfak et al. [20] proposed a hybrid method that uses k-means clustering for data preprocessing and an SVM for fault classification, enhancing the accuracy and efficiency of transformer fault diagnosis.

Another important advancement in the field has been the use of artificial neural networks (ANNs), which are well-suited to handle nonlinear relationships between gas concentrations and fault conditions. Liu et al. [21] proposed an ANN model to diagnose transformer faults based on DGA data. The model was able to learn complex patterns in the data, outperforming traditional rule-based methods, and was particularly effective in detecting incipient faults before they manifested as major failures. Furthermore, Zhang et al. [22] demonstrated the use of a deep learning approach, leveraging a multi-layered neural network to process DGA data. Wang et al. [23] leveraged a brain-inspired neural-transformer architecture to achieve high-precision and robust fault diagnosis while maintaining low computational cost. The study incorporated a two-dimensional representation method, the frequency-slice wavelet transform (FSWT), to enhance the fault identifiability of vibration signals. Their results indicated that deep neural networks could significantly enhance fault detection accuracy compared to simpler machine learning models.

The integration of deep learning techniques such as convolutional neural networks (CNN) and recurrent neural networks (RNN) for transformer fault diagnosis has also gained traction. These advanced AI techniques offer the advantage of automatic feature extraction from raw data, which reduces the need for manual intervention and predefined features. For instance, Li et al. [24] proposed a hybrid model combining CNN and RNN to analyze time-series DGA data. Their model showed a remarkable ability to capture both temporal and spatial patterns in the gas concentrations, leading to improved diagnostic performance. The study found that deep learning-based models could provide higher diagnostic accuracy and robustness, but the neural network models are complex and have high computational complexity.

Despite the promising results of AI algorithms in DGA-based fault detection, some challenges remain. One key challenge is the need for large and high-quality datasets to train AI models effectively. In particular, the quality and consistency of DGA data can vary depending on operational conditions, making it difficult to develop generalized models. Several studies have explored methods to address this issue, such as using data augmentation techniques or transferring knowledge from other datasets [25]. Additionally, the operating conditions and fault types of transformers may vary in different regions and operating environments, leading to insufficient adaptability of existing models.

In conclusion, integrating AI with DGA for transformer fault diagnosis holds significant potential to improve fault detection accuracy, automate the diagnostic process, and reduce dependence on expert judgment. However, challenges related to data quality, imbalance issue, high computational complexity, and the need for real-time performance and adaptability remain important areas of current research.

## 3. Methodology

### *3.1. Problem Formulations*

The data-driven transformer fault diagnosis can be described as a supervised classification problem based on labeled data. Let the input be an M-dimensional sample, where each sample contains M key performance indicators (KPIs) reflecting the transformer's performance, i.e., $X = [KPI_1, KPI_2, \ldots KPI_M] \in \mathbb{R}^M$. The output is a set of anomaly causes $Y = \{F_1, \ldots, F_i, \ldots F_N\}$, where $F_i$ represents the i-th anomaly cause. The fault diagnosis process is to determine the correspondence between the input vector representing the transformer's performance state and the output set, i.e., $f(X) \rightarrow Y$. Let $FX_i$ represent the distribution of the input vector $X = [KPI_1, KPI_2, \ldots KPI_M] \in \mathbb{R}^M$ when the i-th anomaly

cause occurs, and let $FC_i$ represent the distribution of the output $F_i$ when the i-th anomaly cause occurs. Then, the anomaly diagnosis problem can be transformed into the expression $FX_i \rightarrow FC_i, \forall i \in N$, where N is the total number of anomaly causes.

### *3.2. Data Balancing*

In fault diagnosis, there is often an imbalance in the data between samples from different fault causes. This class imbalance can significantly affect the classification performance of the model. Therefore, this section introduces several common data balancing methods and compares the impact of different data balancing techniques on the model's accuracy in the subsequent model training process.

(1) Random sampling

Random sampling methods are primarily divided into random over-sampling and random under-sampling. Random over-sampling balances the sample sizes of the minority and majority classes by duplicating data from the minority class and adding it to the original dataset. Since random over-sampling simply increases the number of minority class samples by copying, it can easily lead to overfitting, where the model learns overly specific information and performs poorly in terms of generalization. On the other hand, random under-sampling removes some data from the majority class to balance the number of samples in both classes. However, a drawback of this method is that it may result in the loss of important data.

(2) Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE [26] is an improved version of the random over-sampling algorithm that selects neighboring samples using the k-nearest neighbor (KNN) algorithm and synthesizes new minority class samples based on these neighbors. A new minority class sample can be generated through the following process. First, for a minority class sample $X_i$ in the sample set MMM, the k-nearest neighbors in the feature space are found. Then, a sample $X_j$ is randomly selected from these neighbors, and the difference $diff = X_i - X_j$ is calculated. Finally, a new sample $X_n$ is generated using the formula $X_n = X_i + \text{rand}(0,1) \cdot diff$, where rand(0,1) is a random number between 0 and 1.

(3) Adaptive Synthetic Sampling (ADASYN)

ADASYN [26] uses a systematic method to adaptively create different amounts of synthetic data according to their distributions. The main idea of the ADASYN algorithm is to use the density distribution as a criterion for automatically determining the number of synthesized samples that need to be generated for each minority example. The basic process of ADASYN is summarized as follows:

Firstly, calculate the total number of minority class samples that need to be synthesized:

$$G = (M_{\max} - M_{\min}) \times \alpha \tag{1}$$

where $M_{min}$ is the set of minority class samples and $M_{max}$ is the set of majority class samples. $\alpha \in [0,1]$ represents the expected balance level after the addition of the synthetic sample. When $\alpha = 1$, it means that the dataset is completely balanced after adding the synthetic samples.

Secondly, for each sample $X_i \in M_{\min}$, find their k-nearest neighbors in the n-dimensional space and calculate the ratio:

$$r_i = \Delta_i / K, i = 1, 2, \cdots, m \tag{2}$$

where $\Delta_i$ is the number of examples in the k-nearest neighbors of $X_i$ that belong to $M_{\max}$.

Next, regulate $r_i$ and express a

$$r'_i = r_i / \sum_{i=1}^{M_{\min}} r_i \tag{3}$$

Calculate the number of samples that need to be synthesized for each $X_i \in M_{\min}$:

$$g_i = r'_i \times G \tag{4}$$

Finally, for each $X_i \in M_{\min}$, generate $g_i$ synthetic data samples according to

$$X_{new} = X_i + (X_j - X_i) \times \lambda \tag{5}$$

where $X_j$ is one of the k-nearest neighbors for $X_i$ and $\lambda \in [0,1]$ is a random number.

### *3.3. The Proposed Method*

As shown in Figure 1, this paper proposes a two-stage classification model that combines unsupervised and supervised methods. In the first stage, the k-means algorithm is used for unsupervised classification; based on the classification results, a self-organizing neural network is trained in the second stage for fault diagnosis.

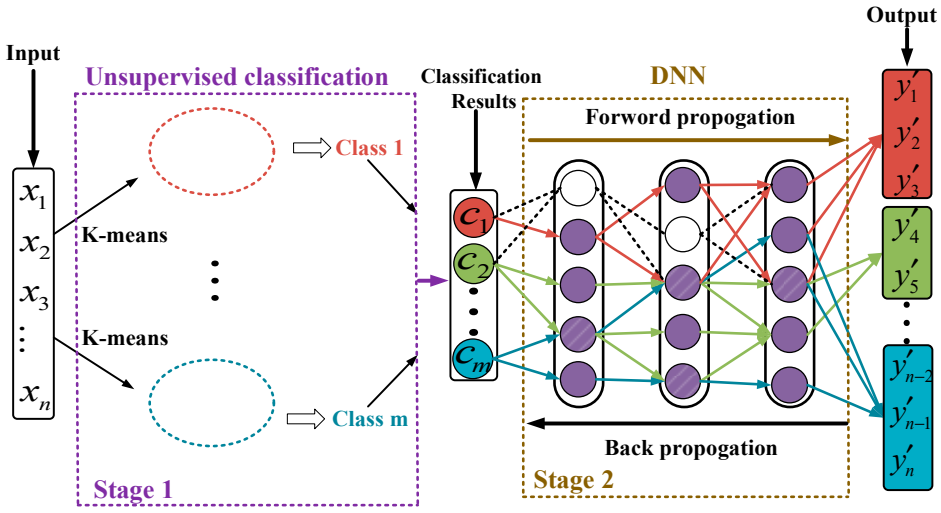



**Figure 1.** The framework of the proposed model.

(1) Stage 1

K-means is a popular unsupervised clustering algorithm that partitions a dataset into a predefined number of clusters *K*. The goal is to group similar data points together based on their features. In reference to [20], the study generates 120 clusters using the k-means algorithm. Since each cluster may contain one or more fault types, during the subset analysis phase, human experts employ a traditional sub-model based on gas ratios to distinguish between the different fault types within the same cluster. The various sub-models are then combined to form the final diagnostic model. In reference [20], $R_1$–$R_{15}$ represent the ratios utilized by the human expert to differentiate between the fault types within the same subset.

(2) Stage 2

A fully connected neural network consists of an input layer, multiple hidden layers, and an output layer, with each layer containing multiple neurons. Let the input sample set be $\{x_i, y_i\}_{i=1}^{N}$, where $N$ is the number of samples. The input vector $X$ propagates

forward through each hidden layer to the output layer. The output of each hidden layer is $h_i(\mathbf{x}) = g(w_i^T \mathbf{x} + b_i)$, where $w_i \in R^{d_i \times d_{i-1}}$, $b \in R^{d_i}$, $d_i$ are the dimensions of the output vector, and $g$ is the nonlinear activation function. The neural network output $f_W(\mathbf{x})$ can be expressed as

$$f_W(\mathbf{x}) = H_L(H_{L-1}(\cdots(H_1(\mathbf{x})))) \tag{6}$$

where $L$ is the total number of hidden layers, and $W = (W_1, W_2, \ldots, W_l, \ldots, W_L)$ represents the total parameter vector of the neural network.

Fault diagnosis is essentially a multi-class classification problem. Let the total number of classification categories be $C$. The loss function is computed using cross-entropy, which can be expressed as

$$L_{CE}(W) = -\sum_{k=1, k\neq c}^{C} P(c_c/x) \cdot \log(P(c_k/x)) \tag{7}$$

where $C$ is the total number of classes, $P(c_c/\mathbf{x})$ is the probability that the input x is predicted to $y_c$ belongs to the true class $c$, and $f_W(x) = P(c_k/x)$ is the probability that the input is predicted to belong to any class $k$. If $k! = c$, it indicates the probability of misclassification.

To minimize the loss function $L_{CE}(W)$, the backpropagation algorithm is commonly used to reduce the error. The stochastic gradient descent (SGD) method trains the model by randomly sampling small batches of data, which is efficient and converges quickly. Therefore, it is often used to optimize the network parameters $W$. The parameter update can be expressed as

$$W^n = W^{n-1} - \alpha \nabla_W L_{CE}(W; \mathbf{x}_i^n, y_i^n) \tag{8}$$

where $\mathbf{x}_k^n, y_k^n \in \text{batch}_n = \{(\mathbf{x}_1^n, y_1^n), (\mathbf{x}_2^n, y_2^n), \ldots, (\mathbf{x}_k^n, y_k^n)\}$ represents the loss for the $n$-th batch $\text{batch}_{n(1\leq n\leq N_{batches})}$ of training samples, $\mathrm{x}_i^n$ and $\mathrm{y}_i^n$ are the input vector and the sample class, respectively, $W^t = \{W_l^n\}_{l=1}^L$ is the network parameter vector after training on the $n$-th batch of samples, and $\alpha$ is the adaptive learning rate.

Reference [27] proposed two cascaded DNNs. The first-level DNN is used to partition the original dataset into multiple subsets, where each subset contains data with similar characteristics. In the second level, a DNN subnet is dynamically and adaptively created for each subset, with the selection of nodes and connections optimized to minimize training errors. Inspired by reference [27], in this paper, a self-organizing neural network method is introduced, which adaptively suppresses activated nodes during the training process. This approach dynamically builds the neural network's training parameter set, significantly lowering the model's computational complexity without compromising accuracy. Different appropriate partial networks are chosen for different categories to minimize the overall loss function $L_{CE}(W)$.

From Equation (2), it can be seen that the complexity of the backpropagation algorithm depends on the number of hidden layers and the number of nodes in each layer, i.e., it depends on the weight vector $W = (W_1, W_2, \ldots, W_l, \ldots W_L)$ of the network. Since there is currently no effective method to precisely estimate the required number of layers and neurons in a neural network in advance, the appropriate network configuration must be selected through multiple trial-and-error processes. To reduce the computational complexity of SGD, this paper proposes a novel self-organizing stochastic gradient algorithm. This algorithm can autonomously suppress the activation of inactive neurons as the training process evolves and automatically select the network parameters for backpropagation that need to be updated. This significantly reduces the parameter tuning required for the backpropagation network while ensuring training accuracy. Figure 2 illustrates the process of selecting activated neurons and autonomously constructing the neural network

training parameter set in the self-organizing stochastic gradient descent method. The backpropagation network training parameter set is updated in epochs, with each epoch serving as the smallest update cycle. Figure 2a,b show the schematic diagrams of the selected backpropagation training parameter set and activated neurons at epochs $n-1$ and $n$, respectively. As training progresses, after each epoch, the neural network autonomously decides which inactive neurons to suppress at the beginning of the next epoch. Based on the current set of activated neurons, the backpropagation training parameter set is constructed. The suppressed inactive neurons, along with all the weight parameters connected to them, will not be updated in subsequent training processes. In this paper, whether a neuron is an inactive node is determined by checking if the change in its weights at $epoch_{n-1}$ and $epoch_n$ is smaller than a predefined threshold $\tau$. Specifically, for the $i$-th hidden node in the $l$-th layer $u_l^i$, if the weight distance $d_{l_2} = \left\| w_{u_l^i}^n - w_{u_l^i}^{n-1} \right\|_2$ is smaller than the threshold, then the hidden node is considered inactive.

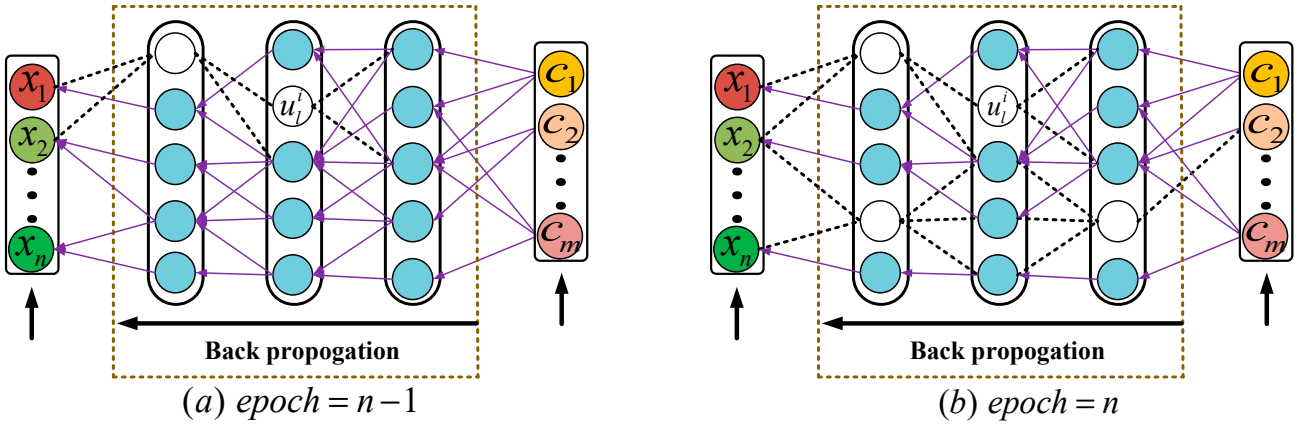



**Figure 2.** Illustration of autonomous node selection in backpropagation.

### *3.4. Incremental Learning*

Incremental learning allows the model to continuously learn from new data while retaining its ability to generalize from previously seen examples. This is achieved by updating the model incrementally after each new batch of data is processed. Incremental learning with deep learning neural networks (DNNs) is a challenging task due to the complexity of neural networks and the problem of catastrophic forgetting (where the model forgets previously learned knowledge when new data is introduced). Fine-tuning is the most commonly used technique in incremental learning with deep neural networks. In this approach, a pre-trained model is used as a starting point, and the model is fine-tuned with new data. Fine-tuning allows the network to retain useful knowledge learned from previous tasks while adapting to new data. When new data arrive, some layers of the pre-trained model are frozen, and only the last few layers are updated to adapt to the new task or data distribution. However, if the new data differ greatly from the old data, fine-tuning may not be sufficient. To solve this problem, in this paper, we learn new knowledge by activating the previously frozen nodes from the previous step. Specially, for any active hidden units $i$ at the $l$-th layer $u_l^i$, if the $\ell_2$-distance ($d_{l_2} = \left\| w_{u_l^i}^t - w_{u_l^i}^{t-1} \right\|_2$) between the weight of $u_l^i$ at time $t$ and $t-1$ is larger than the threshold $\tau$, we will randomly activate a frozen node at layer $l$. After all the actively hidden nodes have been checked, we need to retrain the weights of the new network. At the same time, due to the reasonable initial parameters from last training, the retraining convergence time will be greatly reduced. Unlike reference [27], which adds new neural nodes to the existing network to learn new knowledge, this paper adopts a method of activating existing frozen neural nodes for

learning. In comparison, the proposed method features a simpler network structure and significantly lower computational complexity.

### *3.5. Training and Comparation*

In order to highlight the performance of our proposed algorithm, several commonly used regression algorithms are used for performance comparison.

(1) Traditional DNN

To highlight the effect of a self-organizing neural network, we adopt the traditional DNN model to evaluate the performance of the dataset.

(2) Support Vector Regression (SVM)

SVM attempts to minimize the error bound and maximize the margin to get an optimal separating hyper-plane. For the decision function $f(x) = \langle \omega, \phi(x) \rangle + b$, we can get the SVM model by solving the following problem.

$$\begin{aligned} &\text{minimize} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i{}') \\ &\text{subject to} \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \leq \varepsilon + \xi_i{}' \\ \xi_i, \xi_i{}' \geq 0 \end{cases} \end{aligned} \tag{9}$$

where $C$ is the regularization parameter, $\phi(x)$ is the kernel function that maps the input date to a high-dimensional feature space, $\varepsilon$ is the error tolerant threshold, $\xi_i$ and $\xi_i{}'$ is the $i$-th slack variable.

(3) Decision Tree (DT)

DT observes input properties and learns the decision rules using a kind of tree structure to predict the continual output. Decision tree is constructed by recursive partitioning that starts from the root node (top node) and splits it into left and right child nodes. These child nodes will be further split themselves until the leaves are pure, i.e., leaf nodes. Each non-leaf node represents a test on a feature attribute. Each branch represents the output of this feature attribute in a certain value range, and each leaf node stores the output variable. The RMSE will be applied as a loss function to get the best division.

(4) Random Forest (RF)

RF is an ensemble learning algorithm that combines multiple decision trees for regression. Compared with DT, RF has better generalization performance and is less sensitive to outliers. The operation of RFR is to construct several decision trees during training and then output the average value as the prediction of all trees.

### *3.6. Computational Complexity Analysis*

The computational complexity of a deep neural network (DNN) is generally represented by the number of floating-point operations (FLOPs) required for the forward propagation process. For each layer of a fully connected neural network, the FLOPs can be expressed as $F_i = (2I_i - 1) \cdot O_i$, where $I_i$ and $O_i$ represent the input and output dimensions of the $i$-th layer, respectively. For a fully connected neural network with $L$ hidden layers, the forward propagation computational complexity is expressed as

$$FLOPs_{forward} = N_{epo}\sum_{i=1}^{L}(2I_i - 1) \cdot O_i \tag{10}$$

where $N_{epo}$ is the number of epochs at convergence.

For a neural network with a symmetric structure in both forward and backward propagation, the computational complexity of backward propagation is the same as that of forward propagation. If some neurons are inactive during backward propagation (i.e., their gradients are set to 0), the computational complexity of backward propagation is directly proportional to the total number of active neurons.

## 4. Experiment

### *4.1. Dataset*

The proposed model was evaluated on the dataset provided by reference [20], which contains 849 labeled DGA data. It utilized a dataset comprising gas concentration measurements from dissolved gases in transformer oil, which are key indicators of faults in power transformers. The dataset includes six main fault types, including low energy discharge ($D_1$), high energy discharge ($D_2$), low thermal fault (PD), partial discharge ($T_3$), medium thermal fault ($T_2$), and high thermal fault ($T_1$). The dataset specifically focuses on the gas ratios, including hydrogen ($H_2$), methane ($CH_4$), ethane ($C_2H_6$), ethylene ($C_2H_4$), and acetylene ($C_2H_2$), which are crucial for diagnosing transformer faults. In order to better capture the data features and improve diagnostic performance, like in reference [20], this paper has generated seven new features based on the original dataset. Table 1 presents the seven input feature vectors used in this paper. The importance and function of each vector are described below:

**Table 1.** Seven new features used in this paper.

| New Features | Descriptions |
|---|---|
| $X = [X(i)]\ i \in [1,5]$ | $X(i) = \frac{C_i}{\sum_{j=1}^{5} C_j}$ where $C_i$ and $C_j$ represent the concentrations of $H_2$, $CH_4$, $C_2H_6$, $C_2H_4$, and $C_2H_2$, respectively. |
| $X = [R_1, R_2, R_3]$ | $R_1 = \frac{CH_4}{H_2}, R_2 = \frac{C_2H_2}{C_2H_4}, R_3 = \frac{C_2H_4}{C_2H_6}$ |
| $X = [R_1, R_2, R_3]$ | $R_1 = \frac{C_2H_2}{C_2H_4}, R_2 = \frac{C_2H_4+C_2H_6}{C_2H_2+H_2}, R_3 = \frac{C\ H_4+C_2H_2}{C_2H_4}$ |
| $X = [R_1, R_2, R_3, R_4, R_5]$ | $R_1 = \frac{H_2}{H_{2\ \mathrm{lim}}}, R_2 = \frac{C\ H_4}{C\ H_{4\ \mathrm{lim}}}, R_3 = \frac{C_2H_6}{C_2H_{6\ \mathrm{lim}}}$, $R_4 = \frac{C_2H_4}{C_2H_{4\ \mathrm{lim}}}, R_5 = \frac{C_2H_2}{C_2H_{2\ \mathrm{lim}}}$ |
| $X = [R_1, R_2, R_3, R_4, R_5]$ | $R_1 = \frac{H_2}{C_{MAX}}, R_2 = \frac{C\ H_4}{C_{MAX}}, R_3 = \frac{C_2H_6}{C_{MAX}}$, $R_4 = \frac{C_2H_4}{C_{MAX}}, R_5 = \frac{C_2H_2}{C_{MAX}}$ |
| $X = [CH_4, C_2H_4, C_2H_2]$ | $X(i) = \frac{C_i}{\sum_{j=1}^{3} C_j}$, where $C_i$ and $C_j$ represent the concentrations of $CH_4$, $C_2H_4$, and $C_2H_2$, respectively. |
| $X = [R_1, R_2, R_3]$ | $R_1 = \frac{C_2H_2}{C_2H_4}, R_2 = \frac{C_2H_6+C_2H_4}{C_2H_2+H_2}, R_3 = \frac{C\ H_4+C_2H_2}{C_2H_4}$ |

Figure 3 shows the proportion of each fault type in the overall dataset. From the figure, it can be observed that there is a severe data imbalance between the PD (partial discharge) class and the other classes, which could significantly affect the training of the diagnostic model. As in [20], the data collected were randomly divided into two datasets: a test set and a training set with a 70:30 training-to-test ratio. Moreover, the IEC TC10 database containing 117 labeled DGA data [28] will be used for incremental learning.

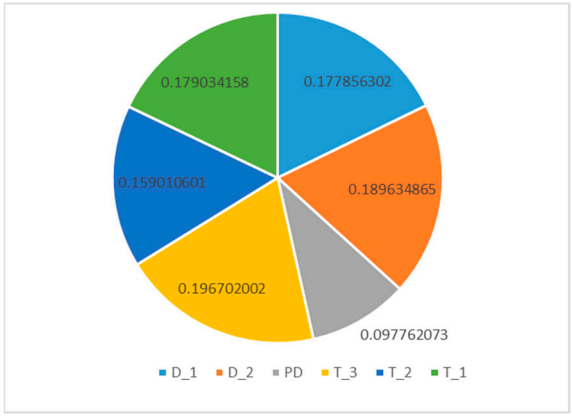



**Figure 3.** The proportion of each fault type in the overall dataset.

### *4.2. Evaluation Metrics*

The diagnostic performance of the system is evaluated based on the accuracy of case diagnosis. In this paper, the diagnosis accuracy rate (DAR) is used to evaluate the performance of the diagnostic system. It is calculated by the ratio of correctly classified samples ($N_{da}$) to the total number of samples ($N_{all}$), i.e., $\frac{N_{da}}{N_{all}}$. A higher accuracy rate indicates better system performance. Additionally, considering only diagnostic accuracy is insufficient to assess the algorithm's performance on imbalanced data, so this paper introduces a confusion matrix to evaluate the algorithm's performance on imbalanced data by comparing the mapping probabilities between the predicted results and the true values. Each column of the confusion matrix represents the predicted class, while each row represents the true class, as shown in Figure 4. $C_i$, $D_i$, and $P(C_i/D_i)$ represent the true class label, the predicted class label, and the corresponding probability, respectively.

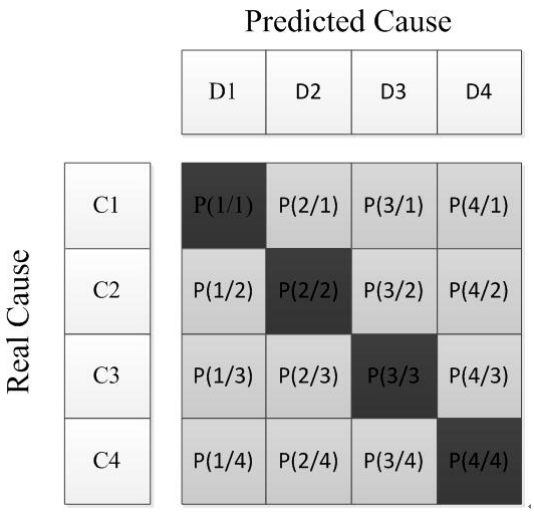



**Figure 4.** Confusion matrix.

### *4.3. Experiment Setting*

To compare the performance of our proposed algorithm with traditional classification methods, several algorithms implemented by sklearn packet, traditional DNN, SVM, DT, and RF, are used. The output of the model is the RMSE estimated by the predicted value and actual output. The parameters of other commonly used classification algorithms were optimized with the grid search method. The optimal parameter settings of the models are shown in Tables 2 and 3.

**Table 2.** Parameter settings for DNN.

| Parameters | Values |
|---|---|
| Number of nodes per layer | (100,100,6) |
| Activation function | ReLU |
| $\alpha$ | $0.01 * 0.5^{epoch/10}$ |
| $epoch_{max}$ | 100 |
| $N_{batches}$ | 100 |
| $\tau$ | 0.001 |
| $m$ | 6 |

**Table 3.** The optimal parameter settings of other classification algorithms.

| Model Type | Parameters | Iterations |
|---|---|---|
| SVM | C = 50, σ = 0.01 | 100 |
| DT | max_depth = 8 | 100 |
| RF | min_samples_split = 5, n_estimators = 100 | 100 |

### *4.4. Results and Comparation*

(1) The results of data balancing

To further understand the differences between the ADASYN and SMOTE methods in terms of data balancing, Figure 5 presents 3D scatter plots of a portion of the dataset after processing with these two methods. The 3D scatter plot of the original training set is shown in Figure 5a, while the 3D scatter plots of the data after balancing with SMOTE and ADASYN are shown in Figure 5b,c, respectively. In Figure 5b, the SMOTE algorithm generates the same number of new samples for each minority class sample, but it does not consider the neighboring samples from other classes, which results in some overlapping of the generated new samples (highlighted in the black dashed box). In Figure 5c, the dataset generated by the ADASYN algorithm not only balances the data distribution but also pays more attention to the difficult-to-learn samples in the minority class. Compared to Figure 5b, ADASYN in Figure 5c reduces the amount of additional noise and overlapping samples, thus performing better than SMOTE.

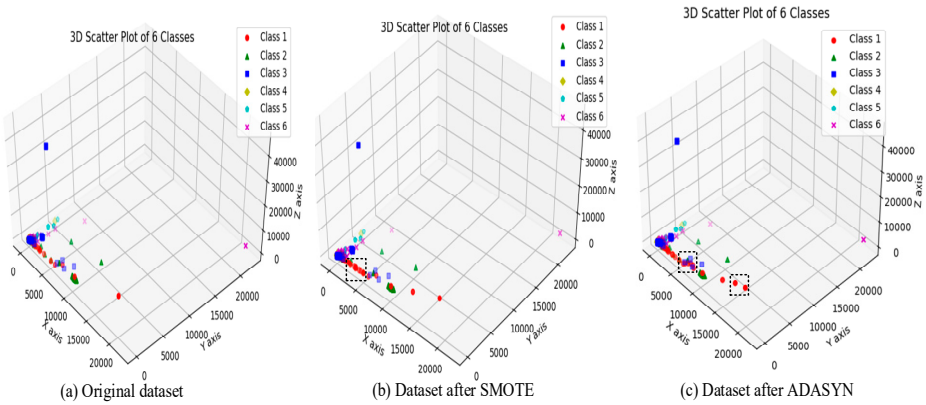



**Figure 5.** 3D scatter plots.

(2) The results of the proposed method

In stage 1, six clusters are generated using the k-means algorithm. The role of classification is to group samples with similar features together, so that in stage 2, different

sub-networks can be selected and trained for different classes, thereby improving training accuracy and convergence speed. Based on the classification results, a self-organizing neural network is trained in the second stage for fault diagnosis. Traditional DNNs train samples with different features together, and the network structure and connections between hidden nodes are fixed. However, the significant differences between sample features can not only lead to large training errors but also make the training convergence process difficult due to the large number of neurons and network connections in DNNs. To address these issues, this paper adopts a differentiated dynamic sub-network selection training method to improve fitting accuracy and reduce the overall regression error of the model. This method primarily targets each classified sub-dataset, dynamically selects neurons, and constructs sub-DNN networks to minimize the overall loss function. Figure 6 shows the convergence process of the model on the training and testing sets. It can be seen that when the epoch is greater than 40, the curve changes smoothly and tends to converge, with an accuracy rate maintained at around 92%.

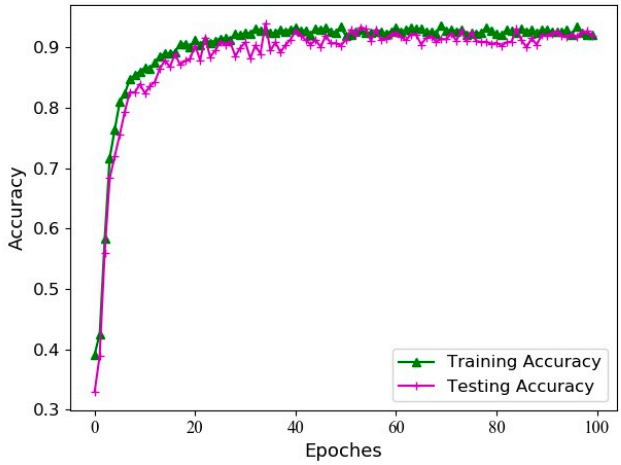



**Figure 6.** The convergence process of the model.

To visualize the suppression effect of the proposed algorithm on inactive nodes during training, Figure 7 shows the curve of the number of activated nodes in each layer of the neural network as the epoch increases. As can be seen in the figure, when epoch > 20, the number of activated neurons in each layer gradually decreases. When 20 < epoch < 30, nearly half of the neuron activations in layer 1 and layer 2 are suppressed, significantly reducing the number of network parameters that need to be updated. When epoch > 40, all neuron nodes are suppressed, resulting in no parameters needing to be updated, and the backpropagation complexity becomes zero, causing the training curve to converge.

Table 4 presents the confusion matrix of the proposed algorithm on the original dataset. The diagonal elements of the confusion matrix represent the number of correctly classified instances for each category, while the off-diagonal elements indicate the number of instances misclassified by the classifier. As shown in the table, the proposed algorithm achieves a high diagnostic accuracy of approximately 92.12%, which is close to the final convergence accuracy depicted in Figure 6. To verify that data augmentation techniques can improve model performance, Table 5 presents the confusion matrix of the proposed algorithm on the test set after augmenting the training data using the ADASYN algorithm. As shown in Table 5, after using the data augmentation technique, the number of misdiagnosed PD cases decreased from 6 to 2. Although the proportion of other categories misclassified as the PD category increased, the overall diagnostic accuracy improved. Specifically, the ADASYN data augmentation technique increased the accuracy from 92.12% to 93.7%. In addition, the

test accuracies achieved using the SMOTE and random sampling data balancing methods are 92.91% and 84.25%, respectively.

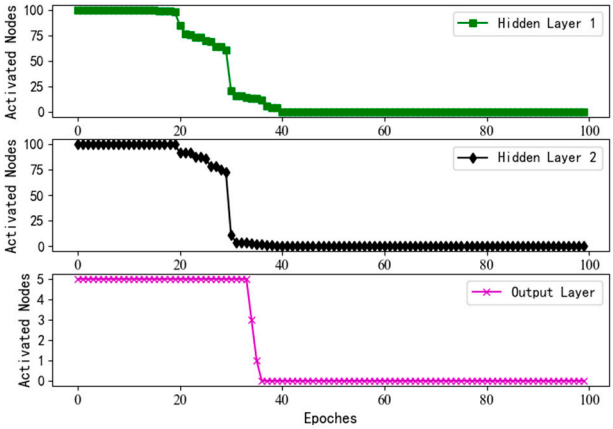



**Figure 7.** Activated nodes vs. epochs.

**Table 4.** The confusion matrix on original dataset.

| True \ Predicted | $D_1$ | $D_2$ | PD | $T_3$ | $T_2$ | $T_1$ |
|---|---|---|---|---|---|---|
| $D_1$ | 43 | 2 | | | | |
| $D_2$ | 2 | 45 | 1 | | | |
| PD | | 2 | 19 | 4 | | |
| $T_3$ | | | 2 | 45 | | 3 |
| $T_2$ | | 1 | | 1 | 37 | 1 |
| $T_1$ | | | | | 1 | 45 |

**Table 5.** The confusion matrix on dataset after ADASYN.

| True \ Predicted | $D_1$ | $D_2$ | PD | $T_3$ | $T_2$ | $T_1$ |
|---|---|---|---|---|---|---|
| $D_1$ | 42 | 1 | 2 | | | |
| $D_2$ | 2 | 43 | 3 | | | |
| PD | | 2 | 23 | | | |
| $T_3$ | | 1 | 3 | 45 | | 1 |
| $T_2$ | | | | | 40 | |
| $T_1$ | | | | | 1 | 45 |

Incremental learning allows the model to continuously learn from new data while retaining its ability to generalize from previously seen examples. However, if the new data differs greatly from the old data, fine-tuning may not be sufficient. To solve this problem, in this paper, we learn new knowledge by activating the previously frozen nodes from the previous step. Specially, if the weight of $u_l^i$ at time $t$ and $t-1$ is larger than the threshold $\tau$, the node $i$ at layer $l$ ($u_l^i$) will be activated. The IEC TC10 database containing 117 labeled DGA data [24] is used for incremental learning. Table 5 presents the sensitivity analysis of threshold $\tau$ to test accuracy. As shown in Table 6, as the threshold $\tau$ decreases, the number of activated nodes in each layer increases to facilitate learning new knowledge. When the threshold $\tau$ is reduced to 0.0001, the DAR reaches 100%, with all samples correctly diagnosed. However, when the threshold is further reduced to 0.00001, the DAR remains at 100%, but the number of activated nodes increases, leading to higher computational complexity. Therefore, the threshold is set to 0.0001 in this paper.

**Table 6.** The sensitivity analysis of threshold $\tau$ to test accuracy.

| $\tau$ | **0.01** | **0.001** | **0.0001** | **0.00001** |
|---|---|---|---|---|
| DAR | 94.01% | 98.29% | 100% | 100% |
| Number of activated nodes in the *k*-th layer | layer 1: 42<br>layer 2: 37 | layer 1: 65<br>layer 2: 52 | layer 1: 82<br>layer 2: 69 | layer 1: 94<br>layer 2: 82 |

(3) Performance comparations

To highlight the superiority of the proposed algorithm, we compared its performance with existing methods on the two datasets mentioned in the paper, as shown in Figure 8. The figure demonstrates that the proposed algorithm achieves the highest diagnostic accuracy, indicating its effectiveness and potential for further application and deployment in real-world scenarios. Although traditional DNNs can achieve relatively high diagnostic accuracy, the algorithm proposed in this paper features significantly lower computational complexity.

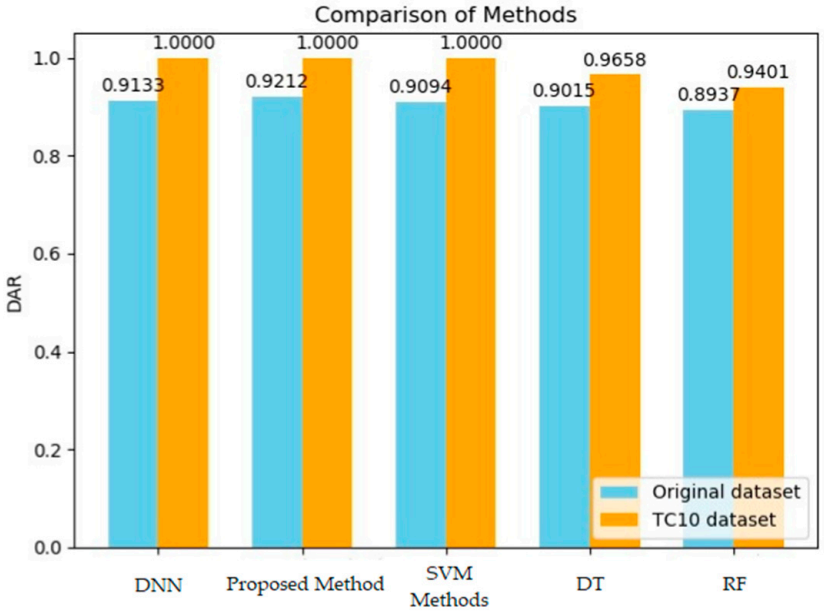



**Figure 8.** Performance comparation.

## 5. Conclusions

In conclusion, this paper proposes a novel fault diagnosis method for power transformers that combines data augmentation, clustering-based classification, and self-organizing neural networks with incremental learning. The ADASYN-based data augmentation effectively addresses data imbalance, improving diagnostic accuracy to 93.7%, while the two-stage classification framework enhances training precision and convergence speed. The dynamic optimization of the self-organizing neural network significantly reduces computational complexity, and the incremental learning mechanism ensures adaptability to new data while maintaining generalization. Comparative analysis demonstrates the superiority of the proposed method over traditional approaches, highlighting its potential for real-world deployment in power transformer fault diagnosis and grid reliability enhancement. The proposed method can be implemented as part of an online monitoring and diagnostic system, leveraging data collected from sensors installed on transformers, providing actionable insights for predictive maintenance and operational decision-making.

**Author Contributions:** Conceptualization, S.L. and Z.H.; methodology, S.L.; software, S.L.; validation, S.L.; formal analysis, S.L.; writing—original draft preparation, Z.X.; writing—review and editing, Z.X. and Z.H.; supervision, Z.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the General Program of National Natural Science Foundation of China (Grant No. 52177083 and 62001166) and by Major Science and Technology Projects in Hebei Province (Grant No. 23281701Z).

**Data Availability Statement:** The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Hossain, M.S.A.; Das, S.K.; Ghosh, A. Transformer fault diagnosis using vibration analysis and machine learning. *IEEE Trans. Power Deliv.* **2020**, *35*, 681–688.
2. Lee, J.M.; Choi, Y.H.; Lee, H.S. Application of machine learning algorithms for transformer fault diagnosis: A review. *IEEE Access* **2020**, *8*, 90873–90887.
3. Rahman, R.S.H.I.A.; Khaleed, H.M.G.D.A.; Krishnan, M.N.S.R.K. A study of transformer fault diagnosis using online monitoring and diagnostic systems. *IEEE Trans. Ind. Electron.* **2014**, *61*, 2617–2625.
4. Kumar, F.C.S.R.N.; Saravanan, P.M.G.R.; Sharma, P.B.S.R.H.K. Partial discharge detection and fault diagnosis in power transformers using wavelet transform. *IEEE Trans. Dielectr. Electr. Insul.* **2012**, *19*, 212–218.
5. Goh, R.S.; Tan, R.K.A.; Khilnani, A.G. Fault diagnosis of power transformers using advanced signal processing techniques. *J. Electr. Eng. Technol.* **2017**, *12*, 465–474.
6. Williams, B.J.; El-Sayed, R.A. Impact of transformer failures on utility performance and economy. *IEEE Trans. Power Deliv.* **2008**, *23*, 1283–1289.
7. Mark, P.B.G.; Patel, R.K.; Agarwal, P.B.S. Unsupervised learning for transformer fault detection: K-means clustering-based approach. *IEEE Trans. Power Syst.* **2018**, *33*, 2310–2321.
8. Parsa, S.M.; Kazemi, A.M.S.H.; Zadeh, H.B. A comprehensive review on the diagnostic techniques for fault detection and condition monitoring of transformers. *Int. J. Electr. Power Energy Syst.* **2012**, *43*, 1–10.
9. McDonnell, M.J.S.; Goulbourne, M.P.E. Evaluation of insulating oil and gas samples for transformer fault diagnosis. *IEEE Trans. Power Deliv.* **1998**, *13*, 1155–1161.
10. Hechifa, A.; Labiod, C.; Lakehal, A.; Nanfak, A.; Mansour, D.-E.A. A Novel Graphical Method for Interpretating Dissolved Gases and Fault Diagnosis in Power Transformer Based on Dynamique Axes in Circular Form. *IEEE Trans. Power Deliv.* **2024**, *39*, 3186–3198. [CrossRef]
11. Xie, S.J.; Zhao, Y.H.; Jin, Q.X. Application of infrared thermography for transformer fault diagnosis. *IEEE Trans. Power Deliv.* **2007**, *22*, 450–457.
12. Lee, C.M.; Lee, J.H.; Kim, J.G. Transformer fault detection and diagnosis using online monitoring. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1214–1221.
13. Al-Dulaimi, S.A.; Raza, S.R.J.; Khokhar, A.Z. Condition monitoring and fault diagnosis of power transformers using vibration and temperature data. *J. Electr. Eng. Technol.* **2020**, *15*, 1703–1712.
14. Ezzat, H.M.; El-Gohary, H.M.; Hamed, A.R. Fault detection in power transformers using acoustic and vibration signal analysis. *IEEE Trans. Power Deliv.* **2020**, *35*, 584–591.
15. Zhang, Z.; Liu, H.; Li, Y. Transformer fault diagnosis using electrical data and machine learning techniques. *IEEE Trans. Smart Grid* **2020**, *11*, 2048–2057.
16. Qiao, L.; Zhang, X.; He, S. Visual Defect Detection and Analysis of Digital Robot Based on Virtual Artificial Intelligence Algorithm. *Procedia Comput. Sci.* **2024**, *243*, 601–609.
17. Wang, J.; Xu, G.; Yan, F.; Wang, J.; Wang, Z. Defect transformer: An efficient hybrid transformer architecture for surface defect detection. *arXiv* **2022**, arXiv:2207.08319.
18. Zhang, J.; Liu, T.; Wang, X. A machine learning approach to fault diagnosis of power transformers using dissolved gas analysis. *IEEE Trans. Power Deliv.* **2019**, *35*, 1784–1792.
19. Xiao, L.; Yang, C.; Zhao, J. Fault diagnosis of power transformers based on decision tree algorithm using dissolved gas analysis data. *Electr. Eng. J.* **2020**, *51*, 2321–2330.
20. Nanfak, A.; Hechifa, A.; Eke, S.; Lakehal, A.; Kom, C.H.; Ghoneim, S.S. A combined technique for power transformer fault diagnosis based on k-means clustering and support vector machine. *IET Nanodielectr.* **2024**, *7*, 175–187. [CrossRef]

21. Liu, Z.; Wang, J.; Zhang, H. Artificial neural network-based transformer fault diagnosis using dissolved gas analysis. *Int. J. Electr. Power Energy Syst.* **2018**, *102*, 73–80.
22. Zhang, Y.; Li, X.; Cheng, X. Deep learning approaches for fault diagnosis of power transformers using dissolved gas analysis. *IEEE Access* **2020**, *8*, 12345–12356.
23. Wang, C.; Tian, B.; Yang, J.; Jie, H.; Chang, Y.; Zhao, Z. Neural-transformer: A brain-inspired lightweight mechanical fault diagnosis method under noise. *Reliab. Eng. Syst. Saf.* **2024**, *251*, 110409. [CrossRef]
24. Li, M.; Chen, X.; Yu, Y. Hybrid deep learning model for transformer fault diagnosis based on time-series dissolved gas analysis. *IEEE Trans. Ind. Inform.* **2021**, *17*, 3971–3980. [CrossRef]
25. Chen, L.; Zhang, Q.; Wu, S. Data augmentation techniques for improving machine learning performance in transformer fault diagnosis based on dissolved gas analysis. *IEEE Trans. Power Syst.* **2017**, *32*, 3072–3080.
26. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
27. Liu, X.; Chuai, G.; Wang, X.; Xu, Z.; Gao, W. QoE Assessment Model Based on Continuous Deep Learning for Video in Wireless Networks. *IEEE Trans. Mob. Comput.* **2023**, *22*, 3619–3633. [CrossRef]
28. Duval, M.; de Pabla, A. Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC10 databases. *IEEE Electr. Insul. Mag.* **2001**, *17*, 31–41. [CrossRef]


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.