



## Article

# Severity Estimation of Inter-Turn Short-Circuit Fault in PMSM for Agricultural Machinery Using Bayesian Optimization and Enhanced Convolutional Neural Network Architecture

Mingsheng Wang <sup>1,\*</sup>, Wuxuan Lai <sup>2</sup>, Peng Sun <sup>2</sup>, Hong Li <sup>1</sup> and Qiang Song <sup>2,\*</sup>

<sup>1</sup> College of Mechanical Electrification Engineering, Tarim University, Alar 843300, China; 120140003@taru.edu.cn

<sup>2</sup> National Engineering Laboratory for Electric Vehicles, Beijing Institute of Technology (BIT), Beijing 100081, China; 3120195251@bit.edu.cn (W.L.); sydney\_peng@126.com (P.S.)

\* Correspondence: wangmingsheng\_08@163.com (M.W.); songqiang@bit.edu.cn (Q.S.)

**Abstract:** The permanent magnet synchronous motor (PMSM) is a key power component in agricultural machinery. The harsh and variable working environments encountered during the operation of agricultural machinery pose significant challenges to the safe operation of PMSMs. Early diagnosis of inter-turn short-circuit (ITSC) faults is crucial for improving the safety of the motor. In this study, a fault diagnosis method based on an improved convolutional neural network (CNN) architecture is proposed, featuring two main contributions. First, a dilated convolutional neural network is combined with residual structures, multi-scale structures, and channel attention mechanisms to enhance the training efficiency of the model and the quality of feature extraction. Second, Bayesian optimization algorithms are applied for the automatic tuning of architecture hyperparameters in deep learning models, achieving automatic optimization of the hyperparameters for the fault diagnosis model of ITSCs. To validate the effectiveness of the proposed algorithm, 17 simulated tests of ITSC fault severities were conducted under both constant conditions and dynamic conditions. The results show that the proposed model achieves the best performance regarding the validation accuracy (98.2%), standard deviation, F1 scores, and feature learning capability compared to four other models with different architectures, demonstrating the effectiveness and superiority of the algorithm.

**Keywords:** agricultural mechanization; fault diagnosis; permanent magnet synchronous motors (PMSMs); inter-turn short-circuit (ITSC) fault; Bayesian optimization



**Citation:** Wang, M.; Lai, W.; Sun, P.; Li, H.; Song, Q. Severity Estimation of Inter-Turn Short-Circuit Fault in PMSM for Agricultural Machinery Using Bayesian Optimization and Enhanced Convolutional Neural Network Architecture. *Agriculture* **2024**, *14*, 2214. <https://doi.org/10.3390/agriculture14122214>

Academic Editors: Chung-Liang Chang and Mustafa Ucgul

Received: 15 October 2024

Revised: 2 December 2024

Accepted: 2 December 2024

Published: 3 December 2024



**Copyright:** © 2024 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

Since the 21st century, the world has been experiencing rapid changes due to issues such as global, ecological, and climate shifts, along with population growth [1]. Environmental protection and food security have garnered increasing attention, creating an urgent need for safe, intelligent, and sustainable solutions [2]. Agricultural mechanization plays a vital role in advancing agricultural modernization and sustainable development, making intelligent fault diagnosis research of paramount importance [3]. Agricultural machinery is extensively utilized in all aspects of modern agricultural production, including tillage, fertilization, sowing, and harvesting. Given their vital role in the production process, the efficient operation of these machines directly impacts both the efficiency and yield of agricultural output [4]. Motors are key power components in agricultural machinery, being responsible for providing stable torque and achieving efficient energy conversion, thereby enhancing mechanical efficiency and reducing energy loss, which makes them more environmentally friendly [5]. PMSMs are widely used in agricultural mechanization due to their excellent torque control performance, high power density, and high efficiency, coupled with China's natural advantage in rare earth resources [5]. They are utilized in equipment such as electric tractors, seeders, harvesters, spraying equipment, tillage, and

seeding machinery, significantly enhancing the intelligence and automation levels of agricultural production. However, the operating conditions and environment of agricultural machinery pose challenges to the safe operation of PMSMs [6]. The faults of PMSMs can generally be categorized into mechanical faults, permanent magnet faults, and electrical faults [7]. Mechanical faults primarily refer to the failures caused by the damage of mechanical components, such as bearings, rotors, and shafts, with common fault types including eccentricity and bearing failures. Permanent magnet faults refer to irreversible partial or total demagnetization of the permanent magnets fixed on the rotor, which is unique to PMSMs and can be caused by various factors. Electrical faults usually occur due to damage to the stator windings, and the main fault types include open-circuit winding faults, ITSC faults, phase-to-phase short-circuit faults, and winding ground faults. Due to the limited installation space and the high power density requirements of PMSMs in agricultural machinery, the winding design becomes highly compact, which poses significant challenges for the heat dissipation of the motor windings. Furthermore, the operating environment of agricultural machinery is particularly harsh and variable, including exposure to dust, high temperatures, high humidity, complex modal vibrations, as well as frequent instantaneous overloads and fluctuating loads [8]. These factors make ITSC faults one of the most common failures in PMSMs [9]. The occurrence of these faults generates significant fault currents within the short windings, which not only affects the distribution of the air gap's magnetic field and exacerbates motor vibrations but also causes excessive heat generation in the affected windings. If these issues are not detected and addressed in time, it can lead to a rapid increase in the stator winding temperature, damaging the insulation of nearby windings and further worsening the fault condition [10]. This may even result in a loss of control over the motor and agricultural machinery, leading to catastrophic accidents and significant economic losses. Therefore, it is crucial to diagnose and address ITSC faults in their early stage.

Traditional fault diagnosis methods that rely heavily on regular maintenance and experience not only fail to provide early warnings but also are inefficient and costly [11]. Thanks to the advancements in computer technology and sensor technology, intelligent fault diagnosis methods have received widespread attention and application in recent years [12,13]. Jiang et al. implemented fault diagnosis for the rolling bearings of a combine harvester using an improved variational modal decomposition (VMD) and machine learning method, with experimental results demonstrating the superiority of this approach [14]. Parvin proposed a transformer neural network (TNN) model for diagnosing the severity of ITSC faults [15]. By employing a multihead attention mechanism, this algorithm enables the model to concentrate on specific aspects of the input signals, achieving an experimental accuracy exceeding 96%. Li et al. used the correlation coefficient of permutation entropy as an evaluation index, combining random forest algorithms with support vector machines to identify the engine state of a tractor [16]. Their experiments show that this algorithm has good recognition accuracy under small sample conditions. Fan et al. implemented a sparse classification framework for the composite fault diagnosis of tractor bearings, utilizing adaptive feature dictionary learning to automatically extract fault features, which improved the accuracy of fault state identification under heavy noise conditions [17]. Lee et al. proposed an ITSC fault diagnosis model by combining an attention mechanism with a recurrent neural network (RNN) to realize the fault severity estimation [18]. Xu et al. used Time Generative Adversarial Networks (Time GANs) for data augmentation to overcome the issue of limited fault samples and combined it with transformers to perform fault diagnosis of tractor transmission systems [19].

Despite significant achievements in research on fault diagnosis using machine learning algorithms, there has been limited study on early fault diagnosis, let alone the early diagnosis of ITSC faults [20]. In ITSC faults, early fault diagnosis is crucial, as overcurrent and overheating can lead to more severe issues. The existing ITSC fault models inadequately consider the impact of the coil structure within the winding on the fault model, failing to accurately reflect the relationship between winding parameters and fault severity [21].

Moreover, the three-phase current signals utilized are generally lengthy one-dimensional signals that are highly susceptible to electromagnetic interference and can change with varying operating conditions [22]. Consequently, accurately diagnosing ITSC faults requires the extraction of more profound and higher-dimensional features from the collected current signals, particularly when dealing with signals under dynamic operating conditions. This necessitates that the deep learning models employed have sufficient network depth and complexity [23]. However, tests indicate that when the depth of the model increases to a certain extent, its performance tends to saturate and then rapidly decline, which is different from overfitting [24]. Therefore, as the network depth increases, some performance degradation issues will arise. Additionally, the automatic tuning of hyperparameters for the network model is another pressing problem that needs to be addressed. The hyperparameters for the network architecture and training in the aforementioned studies largely rely on manual tuning based on experience, which can consume a significant amount of time and computational resources, even for those with considerable experience.

To address the aforementioned issues, a novel Bayesian optimization-based improvement algorithm was proposed for the enhancement of the ITSC fault diagnosis model. The primary improvements of this paper are outlined as follows:

- (1) By conducting a mechanism analysis of PMSMs, this study investigates the relationship between the parameters of different winding components and the severity of ITSC faults. It proposes a fault model for ITSCs that considers the winding coil structure, as well as indicators that can be used to guide the setting of the severity of ITSC faults.
- (2) A well-crafted deep learning network is proposed, which incorporates residual structures, multi-scale structures, and channel attention mechanisms. This network utilizes dilated convolutions for signal feature extraction, employs residual structures to enhance learning efficiency, and leverages multi-scale structures to enrich the scale of extracted features. Finally, the channel attention mechanism is used to adjust the weight of effective features in fault recognition, thereby improving the accuracy of fault severity identification.
- (3) The Bayesian optimization algorithm is employed to address the tuning of hyperparameters for the fault diagnosis, enabling the automatic optimization of the model's hyperparameters. Building upon the automatic optimization of model training hyperparameters using Bayesian optimization, the network's feature extraction layers are divided into a three-layer architecture, integrating three improved CNN structures to achieve automatic optimization of the model architecture hyperparameters.
- (4) The effectiveness of the proposed fault diagnosis method was evaluated through simulated ITSC fault tests conducted under both constant and dynamic operating conditions. By comparing it with five other fault diagnosis models of different structures, the advantages of the proposed method were validated.

The remainder of this paper is structured as follows: Section 2 presents the ITSC model that considers the winding coil structure and derives an index that can be used to set ITSC fault parameters. Section 3 introduces the proposed algorithm model along with the structure and components of each part. Section 4 describes the experimental equipment used and the settings required for simulating fault tests, as well as detailing the generated dataset. In Section 5, the fault diagnosis model proposed in this paper is compared with five other models of different structures, with experimental results demonstrating the effectiveness and superiority of the proposed algorithm. Finally, Section 6 summarizes the work presented in this paper and discusses future improvements.

## 2. ITSC Fault in PMSMs

The estimation of ITSC faults is critically important for two main reasons. On one hand, these faults are very difficult to detect in their early stages [25]. On the other hand, an ITSC fault can lead to overcurrent and overheating, which can cause more severe issues [26]. In previous research, no index is particularly suitable for the estimation of an early-stage

ITSC fault. In this paper, an equivalent circuit model is proposed, and an index is derived to guide the setting of the ITSC fault severity in experiments.

Currently, the winding structure of a PMSM mostly uses distributed winding arrangements. The coils are wound into appropriate shapes and distributed across two stator slots with a specific pitch. When an ITSC fault occurs in a few turns of the coil within a particular slot, the wires within the corresponding slot will also be shorted, as shown in Figure 1a. Figure 1a is a cross-sectional view of a PMSM with 8 poles and 36 slots. Every turn of the wire within the slot is labeled as Pc-t. For example, A1-3 denotes the 3rd turn wire of the first coil within winding phase A. The red section of the stator winding in the figure indicates the location where the ITSC fault happens, and the corresponding enlarged view shows the labels of the wires involved in the short circuit. Assuming an ITSC fault occurs in the first coil of winding phase A, the schematic diagram of the equivalent circuit model is shown in Figure 1b. From the figure, it can be seen that after the fault occurs, the faulty phase winding will be divided into two parts. One part is the shorted section, and the other is the remaining healthy section. Additionally, the winding of the shorted section will form a new closed loop at the point of the shorted wires. When the current of phase A winding flows through the newly formed closed loop, it divides into the current  $i_f$  passing through the fault resistance  $R_f$  and the current  $(i_a - i_f)$  passing through the shorted winding. Let  $N_c$  be the number of coils in each phase winding,  $N_t$  be the number of turns per coil, and  $N_s$  be the number of turns shorted in the case of an ITSC fault. The degree of winding shorted can be expressed as:

$$\mu = \frac{N_s}{N_c N_t} \tag{1}$$

where  $\mu$  indicates the proportion of shorted turns in the fault phase winding relative to the total number of turns in that phase winding. Based on the above analysis, the description of the equivalent circuit model is as follows:

$$\mathbf{V}_{abcn} = \mathbf{R}_{abcf} \mathbf{I}_{abcf} + \frac{d}{dt} (\mathbf{L}_{abcf} \mathbf{I}_{abcf}) + \mathbf{e}_{abcf} \tag{2}$$

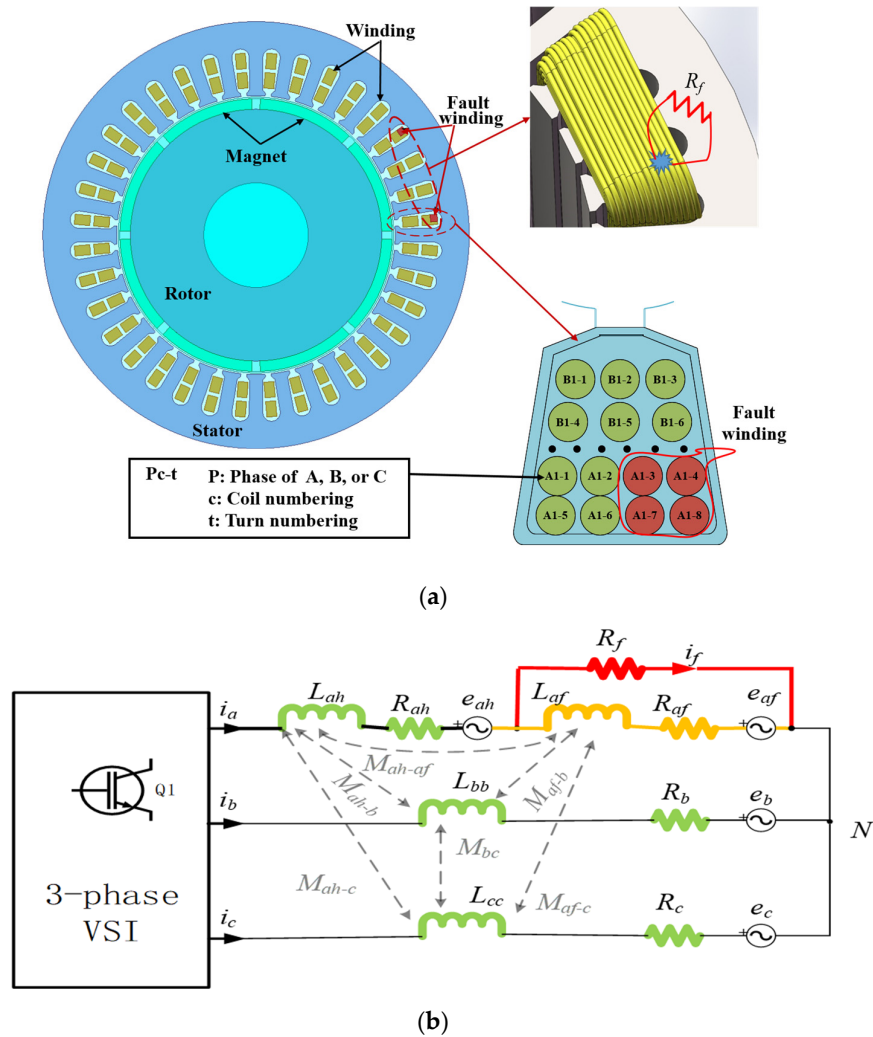
where

$$\begin{aligned} \mathbf{V}_{abcn} &= [v_{an} \quad v_{bn} \quad v_{cn} \quad 0]^T \\ \mathbf{R}_{abcf} &= \begin{bmatrix} R_{ah} + R_{af} & & & R_{af} \\ & R_b & & \\ & & R_c & \\ R_{af} & & & R_{af} + R_f \end{bmatrix} \\ \mathbf{I}_{abcf} &= [i_a \quad i_b \quad i_c \quad -i_f]^T \\ \mathbf{L}_{abcf} &= \begin{bmatrix} L_{ah} + L_{af} + 2M_{ahf} & M_{ahb} + M_{afb} & M_{ahc} + M_{afc} & L_{af} + M_{ahf} \\ M_{ahb} + M_{afb} & L_{bb} & M_{bc} & M_{afb} \\ M_{ahc} + M_{afc} & M_{bc} & L_{cc} & M_{afc} \\ L_{af} + M_{ahf} & M_{afb} & M_{afc} & L_{af} \end{bmatrix} \\ \mathbf{e}_{abcf} &= [e_{fah} + e_{faf} \quad e_{fb} \quad e_{fc} \quad e_{faf}]^T = \frac{d}{dt} [\Psi_{fah} + \Psi_{faf} \quad \Psi_{fb} \quad \Psi_{fc} \quad \Psi_{faf}]^T \end{aligned}$$

In the formula,  $R_{ah}$ ,  $R_{af}$ , and  $R_f$  represent the resistance of the remaining healthy portion, the resistance of the shorted portion, and the fault resistance at the shorted point in fault phase winding A, respectively.  $i_a$ ,  $i_b$ , and  $i_c$  represent the current flowing through phase winding A, phase winding B, and phase winding C, respectively.  $v_{an}$ ,  $v_{bn}$ , and  $v_{cn}$  represent the voltages of the three-phase windings with respect to the neutral point.  $i_f$  represents the current flowing through the fault resistance.  $L_{aa}$  and  $L_{bb}$  denote the self-inductance of phase windings B and C, respectively.  $L_{ah}$  and  $L_{af}$  denote the self-inductance of the remaining healthy portion and the shorted portion of fault phase winding A, respectively.  $M_{bc}$  indicates the mutual inductance between phase windings B and C.  $M_{ahf}$  indicates the mutual inductance between the two portions of fault phase winding A.  $M_{ahb}$  and  $M_{afb}$  represent the mutual inductance between the two portions of fault phase winding A and



phase winding B, respectively.  $M_{ahc}$  and  $M_{afc}$  denote the mutual inductance between the two portions of fault phase winding A and phase winding C, respectively.  $e_{fah}$  and  $e_{faf}$  represent the induced electromotive forces generated by the permanent magnet in the shorted part and the remaining healthy part of phase winding A, respectively.  $e_{fb}$  and  $e_{fc}$  represent the induced electromotive forces generated by the permanent magnet in phase winding B and phase winding C, respectively.  $\Psi_{fah}$  and  $\Psi_{faf}$  represent the flux linkage of the healthy portion and the shorted portion of fault phase winding A, respectively, associated with the permanent magnet.  $\Psi_{fb}$  and  $\Psi_{fc}$  represent the flux linkage of phase winding B and phase winding C induced by the permanent magnet, respectively.



**Figure 1.** (a) Schematic representation of a motor cross-section with an ITSC fault. (b) Equivalent circuit diagram of the motor with an ITSC fault.

Determining the parameters in the fault model is a crucial step for modeling and studying different fault states of the motor. The resistances of the healthy portion and the shorted portion are proportional to the contribution of each part to the total number of turns in the faulted phase winding. The expressions are as follows:

$$\begin{aligned} R_{af} &= \mu R_a \\ R_{ah} &= (1 - \mu) R_a \end{aligned} \tag{3}$$

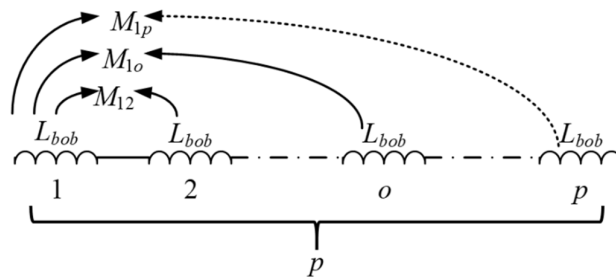
where  $R_a$  stands for the resistance of phase winding A when there is no ITSC fault.

The flux linkage of the permanent magnet in the winding is proportional to the number of turns of the winding. The flux linkages of the healthy portion and the shorted portion of the fault phase winding are represented as follows:

$$\begin{aligned} \Psi_{faf} &= \mu\Psi_f \\ \Psi_{fah} &= (1 - \mu)\Psi_f \end{aligned} \tag{4}$$

where  $\Psi_f$  stands for the flux linkage of the permanent magnet in phase winding A when there is no ITSC fault.

In the fault model of ITSCs in PMSMs, determining the parameters for the stator winding’s self-inductance and mutual inductance is the most complex part. This complexity arises from the changes in the magnetic field caused by the presence of the ITSC fault. The stator winding of a motor is typically composed of multiple coils, as shown in Figure 2. For each coil within a given phase winding, it is necessary to separately discuss the coil’s self-inductance, the mutual inductance between this coil and other coils within the same phase winding, and the mutual inductance between this coil and different coils in the rest of the phase windings.



**Figure 2.** Schematic diagram of the coil composition within the phase winding and the mutual inductance relationships between the coils.

When studying the relationship between the mutual inductance of a coil within a given phase winding and another phase winding, and if the fault occurs only within a single coil, the mutual inductances between the two portions of the fault coil with another phase winding are described by the following equations:

$$\begin{aligned} M &= \sum_{i=1}^{N_c} M_{ip} = M_{afp} + M_{ahp} \\ M_{afp} &= \eta M_{1p} \\ M_{ahp} &= (1 - \eta)M_{1p} + \sum_{i=2}^{N_c} M_{ip} \end{aligned} \tag{5}$$

where  $M$  stands for the mutual inductance between the given phase winding and another phase winding.  $M_{ip}$  stands for the mutual inductance between the  $i$ -th coil within the given phase winding and another phase winding.  $M_{afp}$  represents the mutual inductance between the shorted wires within the fault coil and another phase winding, while  $M_{ahp}$  represents the mutual inductance between the remaining unshortened wires of the fault phase winding and another phase winding.

When the fault occurs in more than one coil, assuming that the ITSC fault occurs in the first  $n$  coils, where  $n \geq 2$ , and the first  $n - 1$  coils are also shorted, the mutual inductances between the two portions of coils within the fault phase winding with another phase winding are described by Equation (6). The meanings of the parameters are consistent with those described earlier.

$$\begin{aligned}
 M &= \sum_{i=1}^{N_c} M_{ip} = M_{afp} + M_{ahp} \\
 M_{afp} &= \sum_{i=1}^{n-1} M_{ip} + (\mu N_c - n + 1)M_{np} \\
 M_{ahp} &= (n - \mu N_c)M_{np} + \sum_{i=n+1}^{N_c} M_{ip}
 \end{aligned} \tag{6}$$

When studying the self-inductance and mutual inductance relationships between coils in a phase winding, since each phase winding is composed of multiple coils connected in series, and assuming a symmetrical distribution of stator winding coils, the self-inductance of each coil is essentially consistent. However, the mutual inductance between coils is related to their relative positions. Let  $L_{bob}$  denote the self-inductance of a single coil in the phase winding, and  $M_{ij}$  denote the mutual inductance between two coils in the same winding, which depends on their relative positions, as described by Expression (7).

$$\begin{aligned}
 M_{ij} = M_{ji} \ \& \ M_{ij} = M_{kl}, \ \text{if } |i - j| = |k - l| \ \text{or } |i - j| = |k - l - n| \ \text{or } |i - j| = |k - l + n| \\
 i, j, k, l \in \mathbb{Z}, \ 1 \leq i, j, k, l \leq N_c \ \& \ i \neq j, k \neq l
 \end{aligned} \tag{7}$$

Here,  $i, j, k$ , and  $l$  represent the positions of each coil in the A-phase winding.

Based on the above analysis, the inductance of each coil can be described as:

$$L = N_c(L_{bob} + \sum_{i=1}^{N_c-1} M_{ij}) \tag{8}$$

where  $\sum_{i=1}^{N_c-1} M_{ij}$  represents the mutual inductance between the chosen coil and the remaining coils in the same winding.  $L$  denotes the self-inductance of the phase winding. Assuming an ITSC fault occurs on the first coil of phase winding A under the condition of no distinction and neglecting the leakage inductance between the wires within the coil, the inductance between the coils in the fault phase winding satisfies the following relationship:

$$\begin{aligned}
 L_{bobf} &= \eta^2 L_{bob} = (\mu N_c)^2 L_{bob} \\
 L_{bobh} &= (1 - \eta)^2 L_{bob} = (1 - N_c \mu)^2 L_{bob} \\
 M_{bobfh} &= \eta(1 - \eta)L_{bob} = N_c \mu(1 - N_c \mu)L_{bob} \\
 L_{bobf} + 2M_{bobfh} + L_{bobh} &= L
 \end{aligned} \tag{9}$$

where  $L_{bobf}$  represents the self-inductance of the shorted wires within the fault coil.  $L_{bobh}$  represents the self-inductance of the unshortened wires within the fault coil.  $M_{bobf}$  stands for the mutual inductance between the shorted wires and unshortened wires within the fault coil. The mutual inductances between the two portions of the fault coil and the other remaining coils within the fault winding satisfies the relationship:

$$\begin{aligned}
 M_{bobf} &= \eta \sum_{i=1}^{N_c-1} M_{ij} = N_c \mu \sum_{i=1}^{N_c-1} M_{ij} \\
 M_{bobh} &= (1 - \eta) \sum_{i=1}^{N_c-1} M_{ij} = (1 - N_c \mu) \sum_{i=1}^{N_c-1} M_{ij}
 \end{aligned} \tag{10}$$

where  $M_{bobf}$  represents the mutual inductance between the shorted wires within the fault coil and the other remaining coils within the fault winding.  $M_{bobh}$  represents the mutual inductance between the unshortened wires within the fault coil and the other remaining coils within the fault winding.

Based on the above analysis, the inductance of each portion in the fault winding can be described as:

$$\begin{aligned}
 L_{af} &= (\mu N_c)^2 L_{bob} \\
 L_{ah} &= (1 - N_c \mu)^2 L_{bob} + (n_p - 1)(L_{bob} + \sum_{i=1}^{N_c-2} M_{ij}) + 2(1 - N_c \mu) \sum_{i=1}^{N_c-1} M_{ij} \\
 M_{afh} &= N_c \mu (1 - N_c \mu) L_{bob} + N_c \mu \sum_{i=1}^{N_c-1} M_{ij}
 \end{aligned} \tag{11}$$

Substituting Equations (3)~(11) into Equation (2), the resistance, inductance, and back electromotive force in the voltage balance voltage equation under ITSC fault conditions can be described as:

$$\begin{aligned}
 \mathbf{R}_{abc f} &= \begin{bmatrix} R_a & & & \mu R_a \\ & R_b & & \\ & & R_c & \\ \mu R_a & & & \mu R_a + R_f \end{bmatrix} \\
 \mathbf{L}_{abc f} &= \begin{bmatrix} L_{aa} & M_{ab} & M_{ac} & L_{af} + M_{ahf} \\ M_{ab} & L_{bb} & M_{bc} & \mu M_{ab} \\ M_{ac} & M_{bc} & L_{cc} & \mu M_{ac} \\ L_{af} + M_{ahf} & \mu M_{ab} & \mu M_{ac} & L_{af} \end{bmatrix} \\
 \mathbf{e}_{abc f} &= [ e_{fa} \quad e_{fb} \quad e_{fc} \quad e_{ff} ]^T = \frac{d}{dt} [ \Psi_{fa} \quad \Psi_{fb} \quad \Psi_{fc} \quad \mu \Psi_{fa} ]^T
 \end{aligned} \tag{12}$$

The  $e_{fa}$  represents the induced electromotive forces generated by the permanent magnet in phase winding A.  $\Psi_{fa}$  represents the flux linkage of phase winding A induced by the permanent magnet.

Since the analyzed stator winding is Y-connected, it follows, from Kirchhoff's Current Law, that:

$$i_a + i_b + i_c = 0 \tag{13}$$

From Equations (2), (12), and (13), the expression for the fault current can be derived as:

$$i_f = \frac{\mu v_{an} + (\mu(L_{af} + M_{ahf}) - L_{af}) \frac{di_f}{dt}}{\mu R_a + R_f - \mu^2 R_a} \tag{14}$$

Let  $d_1 = \mu R_a + R_f - \mu^2 R_a$ ,  $d_2 = \mu(L_{af} + M_{ahf}) - L_{af}$ , and  $v_{an} = v_a - v_n$ ; then, the above equation can be rewritten as:

$$\frac{di_f}{dt} = \frac{1}{d_2} (d_1 i_f) - \frac{1}{d_2} \mu (v_a - v_n) \tag{15}$$

Since the focus of the study is on the early stage of ITSC faults, the amplitude of voltage  $v_n$  is much smaller than that of  $v_a$ , so  $v_a \approx v_a - v_n$ . Assuming  $v_a = V_a \sin(\omega t)$ , the analytical solution of Equation (15) can be described as:

$$i_f = e^{\frac{d_1 t}{d_2}} \left( i_f(0) - \frac{\mu V_a \omega}{d_2 (\omega^2 + \frac{d_1^2}{d_2^2})} \right) + \frac{\mu V_a (\omega \cos(\omega t) + \frac{d_1 \sin(\omega t)}{d_2})}{d_2 (\omega^2 + \frac{d_1^2}{d_2^2})} \tag{16}$$

At the early stage of an ITSC fault, the fault usually occurs in a single coil, there are fewer shorted turns of wires, and the fault resistance at the shorted point is relatively large. Therefore,  $d_1 > 0$ ,  $d_2 < 0$ , and  $|d_1| \gg |d_2|$ . As a result,  $d_1/d_2$  tends towards  $-\infty$ , and  $d_2/d_1$  tends towards 0. Thus, from Equation (16), the approximate expression for the current amplitude can be obtained:

$$I_f \approx \frac{\mu V_a}{\mu R_a + R_f - \mu^2 R_a} \tag{17}$$

According to reference [27], it is known that the amplitude of the three-phase voltage in the stator winding of the PMSM is positively correlated with the motor speed. Therefore, Equation (17) can be rewritten as:

$$I_f \approx K \frac{\mu \omega_r}{\mu R_a + R_f - \mu^2 R_a} \quad (18)$$

where  $\omega_r$  represents the mechanical speed of the PMSM.  $K$  represents a known coefficient. By analyzing the above equation, it can be seen that the resistance of fault phase winding  $R_a$  can be regarded as a known quantity in the equation, and the remaining parameters  $\mu$ ,  $R_f$ , and  $\omega_r$  can directly affect the amplitude of the fault current  $i_f$ . However, among these parameters,  $\mu$  and  $R_f$  are related to the severity of the ITSC fault, while  $\omega_r$  is not. If  $\omega_r$  is excluded from Equation (18), an expression related only to the shorted degree  $\mu$  and the fault resistance  $R_f$  will be derived:

$$\frac{I_f}{\omega_r} \approx K \frac{\mu}{\mu R_a + R_f - \mu^2 R_a} := FI \quad (19)$$

where  $FI$  stands for the severity index of the ITSC fault. When the tested motor is in a healthy state, this index is 0. When the winding of a certain phase of the motor is completely shorted and the fault resistance is 0, this index becomes infinite.

In the early stages of an ITSC fault, this index is essentially unaffected by speed and increases as the fault resistance  $R_f$  decreases or the degree of shorted turns  $\mu$  increases, and vice versa. Each fault severity can be considered as a combination of different  $R_f$  and  $\mu$  values. Of course, in actual motor operation, it is difficult to directly detect the fault resistance  $R_f$  and the degree of shorted turns  $\mu$ , so this severity index is not suitable for estimating the severity of an ITSC fault. However, it can be used as an index for fault severity in experiments to guide the setting of ITSC fault severity.

### 3. Proposed Algorithm

#### 3.1. Convolutional Neural Networks

A CNN is an evolved form of artificial neural networks, currently widely used in image processing and fault diagnosis. A typical convolutional neural network structure is generally a multi-layer feedforward neural network composed of an input layer, convolutional layer, and output layer [2]. The convolutional layer typically needs to be used in conjunction with various functional layers, such as pooling layers, activation layers, normalization layers, and dropout layers, to enhance the performance of the convolutional module [28]. A deep convolutional network structure is formed by stacking multiple convolutional modules, with its complete structure shown in Figure 3. Compared to traditional artificial neural networks, the convolutional layer of a CNN has the characteristics of weight sharing and local connections, which significantly reduces the number of model parameters and lowers the difficulty of training the model. A CNN model with multiple hidden layers can automatically extract multiple features from the input signal, where the lower hidden layers learn the generalized features of the input signal, and the higher hidden layers can obtain more abstract, high-dimensional features through the abstraction and extraction of lower-level features, enabling more precise classification tasks.

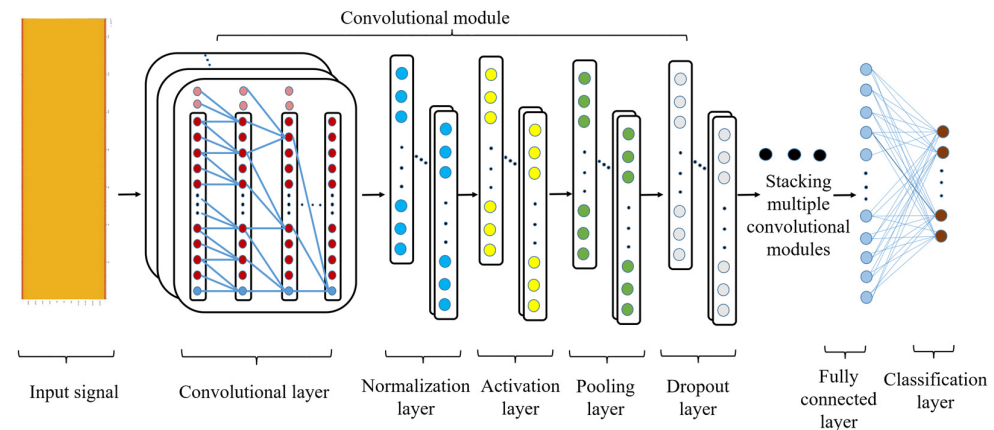
In this research, the dilated CNN is adopted due to its ability to enlarge the receptive field while minimizing the loss of resolution, which is an important quality for forming a deep network architecture [29]. The process of dilated convolution can be expressed as:

$$F(x) = (S *_d f)(x) = \sum_{i=0}^{k-1} f(i) \cdot S_{x-d \cdot i} \quad (20)$$

where  $S$  stands for the 1-dimensional series data of the input signal and  $S \in R^n$ ,  $f$  represents the convolutional kernel and  $f: \{0, 1, \dots, k-1\} \rightarrow R$ ,  $d$  denotes the size of dilation,  $k$  stands



for the convolutional kernel size, and  $x-d \cdot i$  represents the elements of segment  $x$  undergoing the  $i$ -th iteration of the operation. Conventional convolution is a special case where the dilation factor  $d = 1$ . As the network depth increases, the dilation factor also grows, and the receptive field of the final output layer expands accordingly.



**Figure 3.** Schematic diagram of a conventional CNN model structure.

### 3.2. Improvement Measures for Network Architecture

In the early stages of CNN development, scholars generally believed that increasing the width and depth of the network could enhance the model's fitting ability. Therefore, when designing models, they aimed to improve the performance by continuously expanding the width and depth of the network. However, as the complexity of models increased, several common issues in deep learning began to emerge, such as overfitting, gradient explosion or vanishing problems, and network degradation [30]. To address these issues, improvement measures such as residual network structures, multi-scale network structures, and attention mechanisms have gained increasing attention in the design of deep learning models.

#### 3.2.1. Residual Neural Network

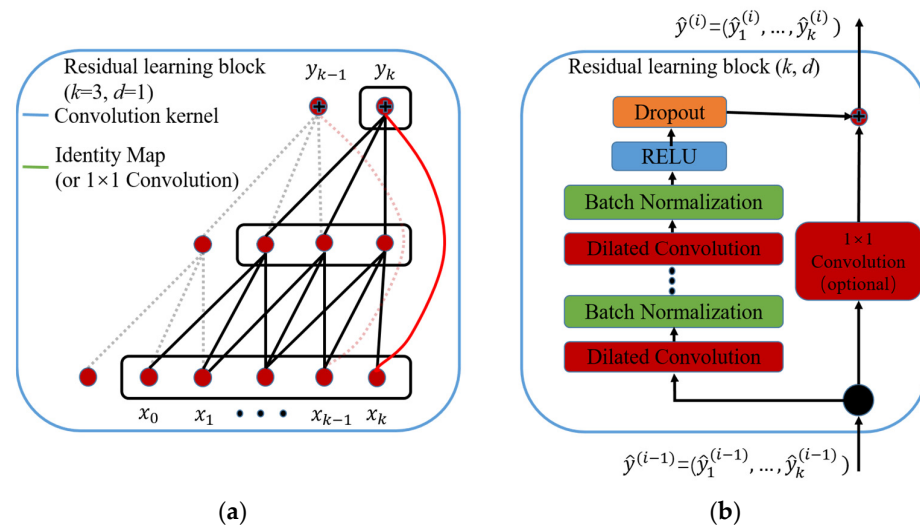
Residual neural networks (Resnets) are mainly used to solve issues such as gradient vanishing or explosion and network degradation during the training process of deep learning models [28]. Theoretically, due to the convolution operation, as the number of network layers increases, the feature information extracted by the model is progressively compressed layer by layer. If the network depth becomes too large, the extracted features will be overly compressed, which will affect the final recognition accuracy, leading to the network degradation problem [24]. In theory, if the newly added layers simply repeat the features from the previous layer without learning new features (called identity mapping), the model's performance will neither improve nor decline. Inspired by this, He et al. introduced identity mapping between branches of different depths in the network, ensuring that the subsequent layers contain more enriched feature information than the previous ones, thereby addressing the degradation issue caused by an increasing network depth [31].

For a residual network structure, let the input be  $x$ , and after passing through the residual branch, the learned feature representation is  $F(x)$ . By directly connecting  $x$  and  $F(x)$  through identity mapping and integrating them,  $y$  represents the total output of the residual network structure. This process can be expressed as:

$$y = x + F(x) \quad (21)$$

A schematic diagram of the residual network structure module is shown in Figure 4. This module consists of ReLU activation layers, batch normalization layers, dilated convolution layers, and dropout layers. In a standard residual network structure, the input  $x$  is directly added to the features extracted by the residual branch,  $F(x)$ , as shown in Figure 4a. However, if there is a dimensional inconsistency between the two, a  $1 \times 1$  convolution is

required in the residual branch to match their dimensions, followed by feature addition, as shown in Figure 4b. If the feature information after the residual branch is not zero, the model’s performance can be enhanced by adding more layers. Conversely, if the feature information after the residual branch is zero, the model’s performance will neither improve nor degrade. Therefore, increasing the network depth through residual structures can avoid the network degradation problem.



**Figure 4.** (a) An example of a residual neural network. The black line represents the convolutional kernel in the Resnet, and the green line represents the identity mapping in the Resnet. (b) Residual neural network. When the input and output of the Resnet have different dimensions, a  $1 \times 1$  convolution is added.

### 3.2.2. Multi-Scale Kernel Network

The core of the multi-scale kernel network is the Inception module. Before its introduction, most CNN models could only enhance their performance by increasing the number of convolution kernels or the depth of convolutional layers. However, this not only leads to a significant increase in the computational burden but may also cause overfitting or even network degradation [32]. To address this issue, the Inception module was proposed, with its structure shown in Figure 5. As illustrated, when data flow into the Inception module, convolution operations are performed simultaneously in multiple convolution kernels of various sizes in parallel branches, extracting features at different scales from the input data [33]. These features are then adjusted to a consistent dimension for concatenation and integration. Thanks to this characteristic, the network structure enhances the model’s perceptive ability while also using  $1 \times 1$  convolution kernels to reduce the number model’s parameters and overall computational burden.

There are a total of four parallel branches in the Inception module. The orange boxes represent the dilated CNN modules, with the convolution kernel sizes from right to left being  $1 \times 5, d = 5$ ;  $1 \times 3, d = 3$ ; and  $1 \times 3, d = 1$ . The other boxes represent conventional CNN modules, with convolution kernel sizes of  $1 \times 5, 1 \times 3$ , and  $1 \times 1$ , respectively. Since the features extracted by the dilated CNN are discontinuous, the introduction of conventional CNNs in each branch can supplement the types of extracted features. After the input data pass through convolution kernels of different sizes and types, features at various scales can be extracted. Let  $X$  represent the input information to this module, and then the expressions of the feature vectors extracted at different scales by the three branches are as follows:

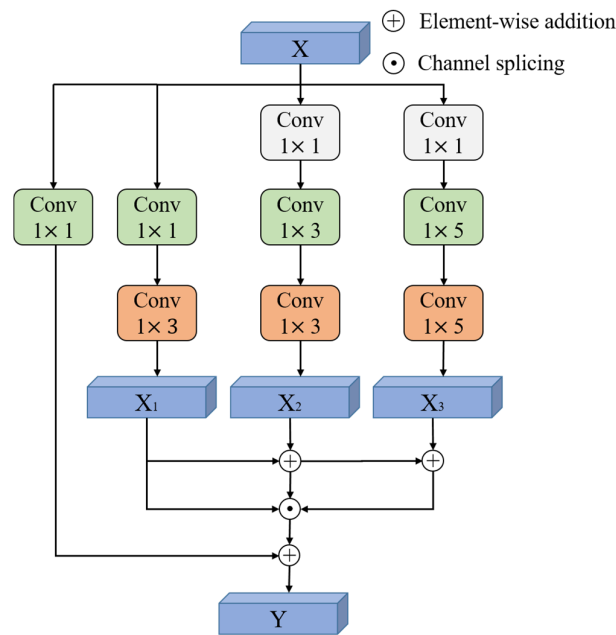
$$\begin{aligned}
 X_1 &= X * C_{1 \times 1} * D_{f=1} \\
 X_2 &= X * C_{1 \times 1} * C_{1 \times 3} * D_{f=3} \\
 X_3 &= X * C_{1 \times 1} * C_{1 \times 5} * D_{f=5}
 \end{aligned}
 \tag{22}$$

Here,  $X_1$ ,  $X_2$ , and  $X_3$  represent the feature vectors extracted by different branches, the symbol  $*$  denotes the convolution operation,  $C_{1 \times 1}$ ,  $C_{1 \times 3}$ , and  $C_{1 \times 5}$  represent conventional convolutions, and  $D_{f=1}$ ,  $D_{f=3}$ , and  $D_{f=5}$  represent dilated convolutions.

In this parallel branch structure, features at different scales are fused through stacking, so the feature vector output by this module can be expressed as:

$$Y = \{X + X_1, X + X_1 + X_2, X + X_1 + X_2 + X_3\} \quad (23)$$

Here, the symbol  $+$  represents the element-wise addition of corresponding elements in the feature vectors, and the symbol  $\cdot$  represents the concatenation operation of feature vectors from different branches.



**Figure 5.** Schematic diagram of a multi-scale network structure.

From the structure of the Inception module, it can be seen that the residual network can serve as a branch of the multi-scale parallel structure connected to this module. Therefore, after the input information passes through this module, the output feature vector not only contains features extracted by dilated convolutions with different kernel sizes and dilation factors but also includes features extracted by conventional convolutions with different kernel sizes. This gives the model a richer ability to represent multi-scale features.

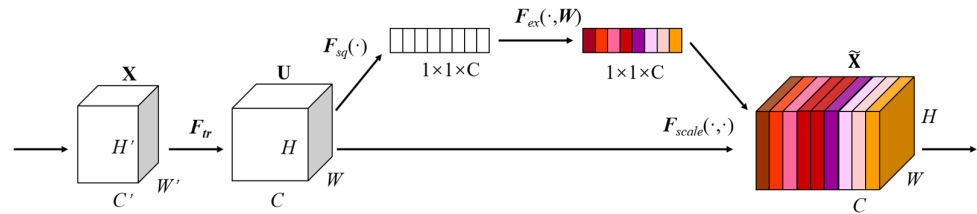
### 3.2.3. The Attention Mechanism

Recent research has shown that in addition to network structure parameters and training parameters, another factor affecting the performance of deep learning models, the attention mechanism, has also been receiving increasing attention [34]. The essence of the attention mechanism is to guide the model toward the task objective by adjusting the weights, thereby filtering out information irrelevant to the task and focusing the model's "attention" on the feature information that is more useful for achieving the task objective.

Commonly used attention mechanisms in CNN models mainly include channel attention, spatial attention, and mixed attention. In this paper, the channel attention mechanism is adopted to enhance the performance of the ITSC fault diagnosis model. By adjusting the weights of the input features from different channels, the model's attention is redistributed, thereby adjusting the contribution of different feature channels to the model and improving its recognition accuracy.

Figure 6 shows a schematic diagram of a typical network structure for the channel attention mechanism. This network structure consists of three main steps: squeezing,

excitation, and scaling. The squeezing and excitation steps are the core parts of the module, which is why this module is called SENet (Squeeze and Excitation Net) [35]. During model training, SENet continuously adjusts the weight distribution of different feature channels through its internal squeeze and excitation modules. By increasing the weights of important feature channels, SENet highlights their contribution and suppresses the contribution of irrelevant and redundant feature channels.



**Figure 6.** Schematic diagram of the channel attention mechanism.

In the channel attention mechanism, the squeezing operation is performed first. This involves compressing the feature information of the channels to adjust the weight relationships between channels. In this process, the input feature information with dimensions  $W \times H \times C$  is converted into  $1 \times 1 \times C$  features through global average pooling, transforming the entire spatial feature of all channels into  $C$  global features. This process can be expressed as:

$$z_c = F_{sq}(u_c) = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H u_c(i, j) \quad (24)$$

Here,  $Z_c$  represents the feature output by the squeezing operation,  $F_{sq}$  represents the squeezing operation, and  $u_c$  represents the input features. After obtaining global features through the squeezing operation, the next step is excitation to capture the relationships between different channels. In the excitation operation, the compressed global feature information is delivered to a fully connected layer, with the dimension of the fully connected layer being  $C \div r \times C$ , where  $r$  is the scaling factor of a channel, being mainly used to reduce the number of channels. As a result, the parameters and computational complexity of the entire module are correspondingly reduced. The output information of this fully connected layer is passed through a ReLU activation layer and then into a second fully connected layer, where the number of feature channels reduced in the previous fully connected layer is restored in this layer. Throughout this process, the two fully connected layers mainly function to organize the feature information obtained using different feature channels, and the Sigmoid activation layer maps the input feature information of each channel into the range of (0, 1), obtaining the weights corresponding to each channel, thus completing the weight adjustment of the channel attention. This process can be expressed as:

$$s = F_{ex}(z_c, W_i) = \delta(W_2\sigma(W_1z_c)) \quad (25)$$

where  $F_{ex}$  represents the excitation operation,  $\sigma$  represents the ReLU activation function,  $W_1$  and  $W_2$  represent the first and second fully connected layers, respectively,  $\delta$  represents the Sigmoid activation function, and  $s$  represents the weights of each channel.

Once the excitation operation is completed and the weights corresponding to each channel are obtained, the scaling operation is performed on the features of each channel. The features within the channel are multiplied element wise by the corresponding channel weights obtained in the previous excitation operation, thereby completing the recalibration of the original features in the channel dimension and achieving the readjustment of attention across all channel features. The entire scaling process can be expressed as:

$$\tilde{x}_c = F_{scale}(u_c, s_c) = s_c \cdot u_c \quad (26)$$

where  $\tilde{x}_c$  represents the features after channel attention adjustment, and  $F_{scale}$  represents the scaling operation.

### 3.3. Bayesian Optimization Algorithm

The CNN-based ITSC fault diagnosis model has a flexible structure and numerous hyperparameters. The impact of varying these hyperparameters on the model's convergence speed and validation accuracy is difficult to predict, which increases the complexity of model design and parameter tuning. To achieve a high-performance diagnosis model, it is necessary to optimize the combination of hyperparameters. Common optimization algorithms include grid search, random search, and Bayesian optimization. Grid search and random search are exhaustive methods that not only require significant computational resources but also can waste resources on ineffective hyperparameter combinations, especially when computational resources are limited. In contrast, Bayesian optimization is an efficient global optimization algorithm, named due to the use of the well-known Bayes' theorem in its optimization framework [36]. It can adjust the optimization strategy based on prior knowledge of existing data, making it more efficient. Therefore, this study uses Bayesian optimization to fine-tune the hyperparameters of the inter-turn short-circuit fault diagnosis model. The expression of Bayes' theorem is shown in Equation (27). The way Bayesian optimization is applied to the hyperparameter optimization of the ITSC fault diagnosis model is achieved by approximating the posterior distribution of the objective function based on Bayes' theorem. By constructing a mapping relationship between the observed values in the set and the maximum value of the objective function, it searches for the hyperparameter combination that maximizes the objective function.

$$p(f | D_{1:t}) = \frac{p(D_{1:t} | f)p(f)}{p(D_{1:t})} \quad (27)$$

Here,  $f$  stands for the unknown objective function, and in the hyperparameter optimization of the ITSC fault diagnosis model, it refers to the performance metric of the model.  $D_{1:t}$  represents the set of observed points, where the number of observed points in the set is  $t$ .  $p(f | D)$  and  $p(f)$  denote the posterior and prior probability distributions of  $f$ , respectively.  $p(D | f)$  represents the likelihood distribution of the observed points in the set, and  $p(D)$  represents the marginal likelihood distribution of  $f$ .

The core of the Bayesian optimization algorithm mainly consists of two parts: the acquisition function and the surrogate model [37]. The surrogate model aims to model the distribution of the unknown objective function using a prior model, thereby finding the optimal parameter combination within the given search space. The acquisition function selects the next observation point to evaluate based on the results obtained from the surrogate model. The framework of the entire Bayesian optimization algorithm is shown in Algorithm 1.

---

#### Algorithm 1. Bayesian optimization algorithm framework process.

---

**Input:** Surrogate model  $f$ , acquisition function  $\alpha$ , and two values,  $t$  and  $k$ .

**Output:** The optimal combination in the hyperparameter vector  $X$ .

- 1: Initialize  $D_t$ ;
  - 2: **for**  $i = 1, 2, \dots, k$  **do** // Iterative search;
  - 3:   Update the initial surrogate model  $Sur_g$ ;
  - 4:   Maximize the acquisition function to select a new evaluation point:  
 $x_{t+1} = \arg \max_{x \in X} \alpha(x | Sur_g, D_{1:t});$
  - 5:   Evaluate the sample point  $y_{t+1} = f(x_{t+1}) + \varepsilon_t$ ;
  - 6:   Update the dataset  $D_{t+1} = D_t \cup \{x_{t+1}, y_{t+1}\}$ , and update the surrogate model;
  - 7: **end for**.
-



First, it is necessary to choose the type of surrogate model and acquisition function, determine the maximum number of iterations  $k$  for the entire optimization process, and generate  $t$  initial sample points as the initialization samples for the Bayesian optimization algorithm. Then, the iterative search process of Bayesian optimization begins. The surrogate model is updated based on the existing sample set  $D_t$ , and the potential point  $x_{t+1}$  of the next optimal value is calculated based on the optimized acquisition function. The objective function is evaluated at the sample point to obtain new observations, and the surrogate model is updated based on the new observations. This iterative process is repeated until the maximum number of iterations  $k$  is reached, at which point the optimal hyperparameter combination is output, completing the entire Bayesian optimization process.

### 3.4. Bayesian Optimization-Based Improvement Algorithm for CNN Models

The performance of the deep learning models used for fault diagnosis is greatly influenced by the model architecture hyperparameters and training hyperparameters. The process of building and training the model involves tuning different combinations of hyperparameters. Selecting the right combination of hyperparameters for model building and training can enhance its performance and improve the accuracy of fault diagnosis identification. However, there are numerous types and quantities of hyperparameters in the model. The architecture hyperparameters of the model include the depth of the CNN mode, the number of convolutional kernels in each layer, the size of the convolutional kernels, and the dropout probability for each dropout layer, among others. Additionally, the type of different layers and the connection methods between different layers will also have a significant impact on the fault diagnosis model. The training hyperparameters of the model include key factors such as the initial learning rate, the decay strategy of learning rate and decay factor, the type of optimizer used, the  $L2$  regularization coefficient, and the size of the mini-batch, among others. In deep learning models, due to the large search space, numerous parameters, and the difficulty in representing the connections between layers, architecture hyperparameters are typically set based on experience and are rarely optimized using optimization algorithms, which will increase the uncertainty of the model. In contrast, training hyperparameters are typically determined based on experience to define the optimization range and then optimized using hyperparameter optimization methods.

In this study, to improve the efficiency of hyperparameter tuning and enhance the model's performance, Bayesian optimization is utilized to optimize the architecture hyperparameters and training hyperparameters of the model. To facilitate the optimization of architecture hyperparameters, the model's feature extraction layers are divided into a three-stage architecture based on the characteristics of the features extracted by convolutional networks. The number of convolutional blocks within each structure is an adjustable parameter, and each convolutional block employs the same number of convolutional kernels, which is also an optimizable parameter. The number of layers within each convolutional block is fixed. The optimization range for each architecture hyperparameter is determined by the experience, the characteristics of the model, and the available computer hardware resources. The maximum number of layers for the model is set as  $d$  and the maximum allowed width for each layer as  $w$ . The maximum number of layers for each stage of the three-stage structure is denoted as  $d_1$ ,  $d_2$ , and  $d_3$ , with the maximum allowed width for each structure also being  $w$ . The number of combinations for training hyperparameter optimization is  $n$ . The comparison of computational complexity between the two architectures is shown in Figure 7. It can be observed that the number of hyperparameters has been reduced from  $(d + 1)$  dimensions to six dimensions, comparing the parameter amount of a conventional model with the three-stage structure with that of a conventional model structure, which makes the automated hyperparameter optimization possible for the model. In addition, the three-stage structure ensures that the model possesses sufficient flexibility and the ability to represent fault features.

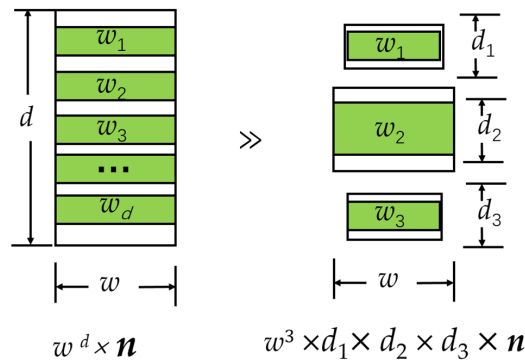


Figure 7. Comparison of the complexity of hyperparameters to be optimized under two architectures.

To achieve time efficiency while ensuring model performance, the previously mentioned Bayesian optimization algorithm with global optimization capabilities is used for automated hyperparameter tuning. The structural diagram of tuning the model’s hyperparameters using the Bayesian optimization algorithm is shown in Figure 8, which is the core part of the Bayesian optimization-based improvement algorithm for CNN models. From the figure, it can be seen that the whole process is divided into two parts: one is the Bayesian optimization part (indicated by the blue box) and the other is the training part of the ITSC fault diagnosis model (indicated by the red box). The entire Bayesian optimization process involves the iterative exploration of different hyperparameter combinations until the set stopping condition is met. At that point, the optimization process concludes, selecting the optimal combination of hyperparameters, which corresponds to the best model for ITSC fault diagnosis under the current conditions.

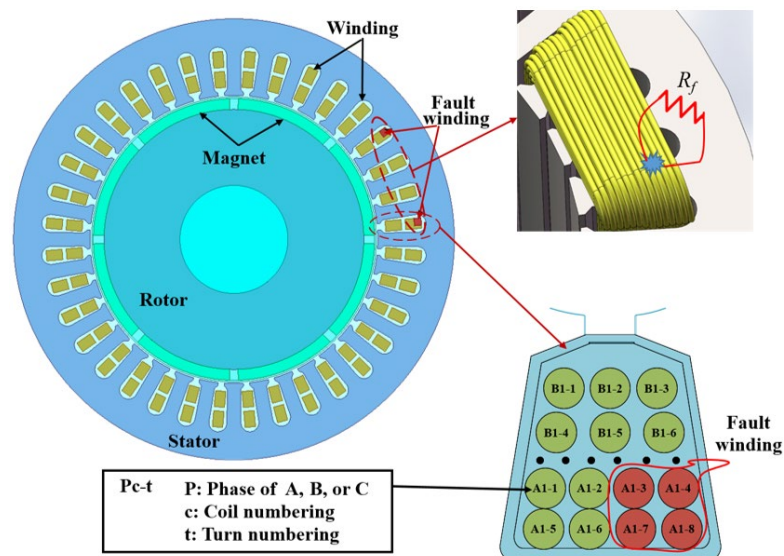


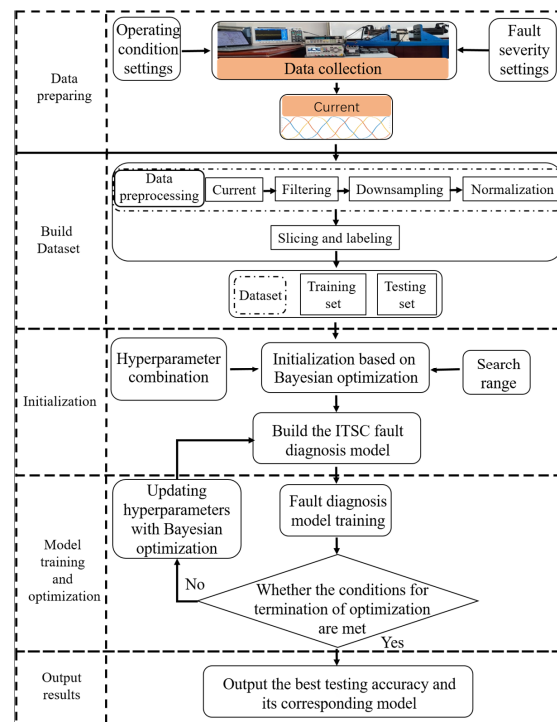
Figure 8. Schematic diagram of hyperparameter tuning for fault diagnosis models based on Bayesian optimization algorithms.

The Bayesian optimization section primarily focuses on two aspects: the initialization of hyperparameter combinations and the updating of model parameters based on the training results feedback from the ITSC fault diagnosis model. The hyperparameter combinations that need to be optimized include architecture hyperparameters and training hyperparameters. Architecture hyperparameters include the depth of convolution blocks in each stage ( $d_1$ ,  $d_2$ , and  $d_3$ ), the width of convolution blocks in each stage ( $w_1$ ,  $w_2$ , and  $w_3$ ), and the dropout layer probability ( $P$ ). Training hyperparameters include the initial learning rate ( $L_{init}$ ), regularization coefficient ( $L_{2R}$ ), and gradient optimization coefficient ( $G_1$ ).

The tasks required for training the ITSC fault diagnosis model include constructing a CNN model according to the architecture hyperparameters provided by Bayesian optimization and training the model based on the given training hyperparameters and dataset. Once the model training is completed, the training results are fed back into the Bayesian optimization algorithm, which then uses the newly obtained hyperparameter combinations to reconstruct and retrain the model. This process of Bayesian optimization and model training continues to iterate until the set termination conditions are met.

The flowchart of the Bayesian optimization-based improvement algorithm for CNN models is shown in Figure 9. The entire process can be divided into five steps:

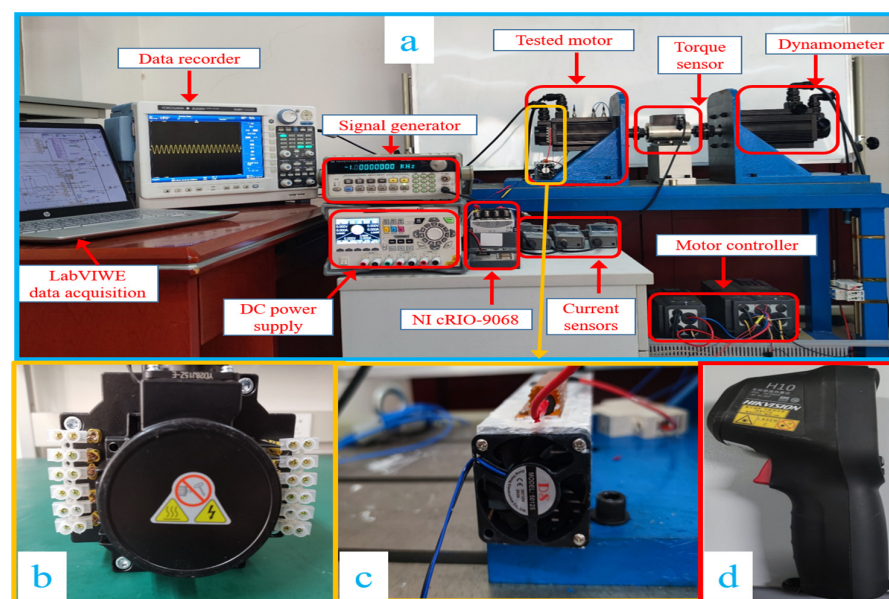
- (1) Data preparation: Set different combinations of fault resistance and shorted ratios to simulate varying severities of faults. Conduct different operating conditions for each fault severity and collect the three-phase currents during the experiment.
- (2) Building of dataset: Perform preprocessing operations such as filtering, downsampling, normalization, segmentation, and label classification. Then, divide all data segments into two non-overlapping datasets, namely the training set and the testing set.
- (3) Initialization: Determine the structure of the CNN blocks, the combinations of hyperparameters to be optimized, and the search ranges for each parameter. Use Bayesian optimization to select the initialization parameters for the model and construct the network structure of the fault diagnosis model according to the specified parameter set.
- (4) Model training and optimization: Train the fault diagnosis model based on the given training hyperparameters, and then test the model's performance using the testing dataset. Record the results obtained along with the corresponding hyperparameter combinations and feed the testing results back into the Bayesian optimization algorithm to update the model's hyperparameters. Repeat the process until the stopping criteria are triggered.
- (5) Output result: When the maximum optimization iterations are met, select the best testing accuracy of the ITSC fault diagnosis model and its relevant hyperparameter combination as the results and output them.



**Figure 9.** The flowchart of the Bayesian optimization-based improvement algorithm for CNN models.

#### 4. Experimental Setup and Data Description

In order to verify the validity of the proposed Bayesian optimization-based improvement algorithm for the ITSC fault diagnosis model. Experiments are carried out on a PMSM. The setup consists of a simulated fault motor and its controller, an auxiliary test motor and its controller, data acquisition equipment, etc., as shown in Figure 10. The fault motor is an 8-pole, 36-slot PMSM, with the windings configured in a star connection, featuring 108 turns of wire per phase. The specific parameters of the faulty motor are shown in Table 1. The fault motor simulates different severities of ITSC faults by combining different fault resistances and shorted ratios. To prevent damage to the fault resistor, a cooling device is required for heat dissipation during the experiment. Temperature monitoring of the entire setup is conducted during the experiment to prevent overheating and damage. The fault resistor and its cooling device are shown in Figure 10c, the fault motor and its shorted winding point terminals are shown in Figure 10b, and the temperature measurement device is shown in Figure 10d.



**Figure 10.** Composition diagram of equipment for ITSC simulation test bench. (a) Test bench and its testing equipment. (b) The faulty motor. (c) The fault resistance and its heat dissipation device. (d) The temperature measurement device.

**Table 1.** Specifications of the tested PMSM.

Parameters	Values	Parameters	Values
Rated power	2.3 kW	Line–line resistance	1.1 $\Omega$
Rated torque	15 Nm	Line–line inductance	4.45 mH
Rated current	9.5 A	Number of turns per phase	108
Rated speed	1500 rpm	Number of coils per phase	12
Pole pairs	4	Voltage constant	114 V/1000 r/min

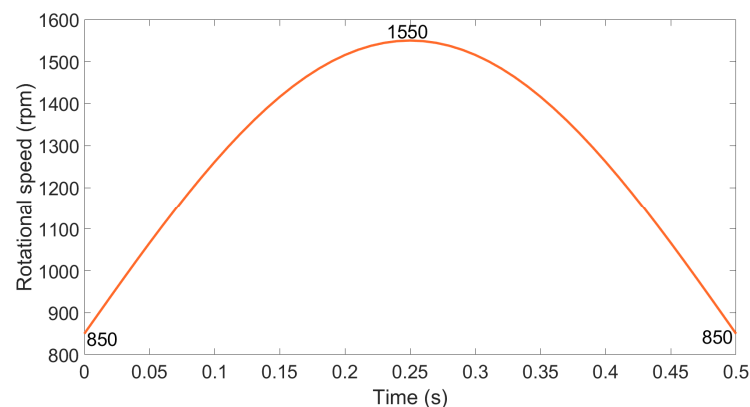
A fault motor simulation test was conducted using a test bench to replicate 17 different fault states of a PMSM exhibiting ITSC faults. This includes one healthy state and sixteen distinct fault conditions. The severity of the ITSC faults is determined by combinations of shorted degrees and fault resistances. The shorted degrees are defined as 5 turns, 9 turns, 11 turns, and 15 turns, totaling four categories. The fault resistances are set at 5  $\Omega$ , 1  $\Omega$ , 0.5  $\Omega$ , and 0.1  $\Omega$ , also totaling four categories, resulting in 16 fault levels. Considering the healthy state of the motor as having a fault level of 0, the experimental data encompass a total of 17 fault severities. To simulate the motor's operating conditions during

agricultural machinery acceleration, deceleration, and constant speed driving, 8 constant speed scenarios and 2 variable speed scenarios were established during the bench test, as detailed in Table 2.

**Table 2.** Operating conditions of the tested PMSM.

Case	Constant			Dynamic	
	150	450	900	2	
Speed (rpm)	150	450	900	1350	850~1550~850
Torque (Nm)	3.0/7.5	3.0/7.5	3.0/7.5	3.0/7.5	3.0/7.5

It can be seen from Table 2 that there are 10 different operating conditions in the test process, each generated by combinations of five speeds and two torques. The two load torques are both constant, while among the five speeds, four are constant and one represents an acceleration and deceleration condition. The dynamic speed variation ranges from 850 rpm to 1550 rpm and then back to 850 rpm, as shown in Figure 11. For each distinct fault condition of the motor, ITSC fault tests are conducted under the aforementioned 10 conditions. The Yokogawa DL850EA oscilloscope is used to record the three-phase current, with a sampling frequency of 1 MHz. The data sampling duration for the fault motor under each operating condition is 10 s. The entire data collection process employs a field-oriented control (FOC) strategy using the VFD037C23A inverter, operating at the switching frequency of 15 kHz, with the auxiliary test motor using speed closed-loop control and the tested motor using current closed-loop control.



**Figure 11.** Schematic diagram of speed variation under variable operating conditions.

During the experiment, due to the absence of hardware filtering, a relatively high sampling frequency of 1 MHz was chosen to avoid signal aliasing caused by interference and other factors during data acquisition. If the raw data were directly used for dataset construction, it would impose a significant challenge on computer hardware resources and severely impact the training speed. The goal of this study is to use deep learning models to extract low-frequency features from the acquired experimental data that are useful for classifying the severity of ITSC faults. Therefore, during data preprocessing, the acquired data are first filtered and then down-sampled to retain low-frequency features while reducing the memory usage of the dataset. A zero-phase low-pass filter is applied to the data, and the down-sampled sampling frequency is set to 15 kHz, matching the switching frequency of the controller. To facilitate the comparison of data under different fault severities and operating conditions and to accelerate the convergence of the deep learning model, the acquired data are normalized to the range of  $[-1, 1]$ . To aid in training the deep learning model, the down-sampled three-phase current data are divided into equal-length data slices, each containing sufficient feature information. The length of each data slice is set to 3000 sampling points, which ensures that, at the lowest operating speed,



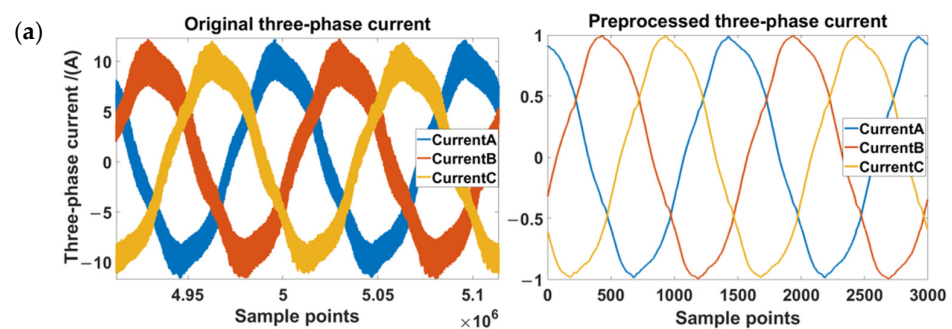
the three-phase current signal collected over one cycle of the motor's rotation is captured in each slice.

The labels of the data slices correspond to their fault severity, as shown in Table 3. In the labels, "HL" denotes the data collected under healthy motor conditions, while "A\*R\*" stands for the data collected under different combinations of fault resistors and shorted ratios. "A2", "A4", "A5", and "A6" represent shorted turns of 5, 9, 11, and 15, respectively. "R5", "R1", "R0.5", and "R0.1" indicate fault resistances of 5  $\Omega$ , 1  $\Omega$ , 0.5  $\Omega$ , and 0.1  $\Omega$ , respectively. The fault severities in Table 3 are arranged in ascending order based on the severity calculated using Equation (18). The sampled data were organized into datasets according to different fault severities, ensuring that the amount of data for each condition under a specific fault severity was equal and the quantities of data corresponding to each fault severity were also equal. For each fault level, the number of data samples is set at 1200, with 360 samples randomly selected for testing, leaving 840 samples for training, resulting in a ratio of 3:7. Ultimately, all training samples form the training set, while all testing samples comprise the validation set.

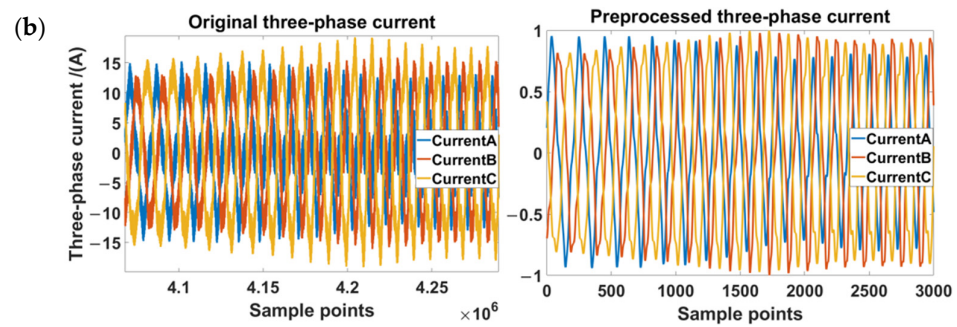
**Table 3.** Dataset description.

Label	Fault Setting		Training	Sample Size		Total
	Fault Resistance ( $\Omega$ )	Shorted Turn Ratio (%)		Testing		
HL	Inf	0	840	360	1200	
A2R5	5	4.6	840	360	1200	
A4R5	5	8.3	840	360	1200	
A5R5	5	10.2	840	360	1200	
A6R5	5	13.8	840	360	1200	
A2R1	1	4.6	840	360	1200	
A4R1	1	8.3	840	360	1200	
A2R0.5	0.5	4.6	840	360	1200	
A5R1	1	10.2	840	360	1200	
A6R1	1	13.8	840	360	1200	
A4R0.5	0.5	8.3	840	360	1200	
A5R0.5	0.5	10.2	840	360	1200	
A6R0.5	0.5	13.8	840	360	1200	
A2R0.1	0.1	4.6	840	360	1200	
A4R0.1	0.1	8.3	840	360	1200	
A5R0.1	0.1	10.2	840	360	1200	
A6R0.1	0.1	13.8	840	360	1200	

The comparison of the data before and after preprocessing is shown in Figure 12. In each figure, the left side displays the original three-phase current signal, while the right side shows the three-phase current after data preprocessing. Figure 12a illustrates the three-phase current under healthy conditions at a speed of 150 rpm and a torque of 3.0 Nm. Figure 12b depicts the three-phase current of a faulty motor with the fault label "A5R0.1", collected under dynamic speed conditions at a torque of 3.0 Nm.



**Figure 12.** Cont.



**Figure 12.** Comparison of three-phase current signals before and after data preprocessing. The left side of the figure shows the original signal, while the right side shows the preprocessed signal. (a) The three-phase current signals are collected under constant operating conditions. (b) The three-phase current signals are collected under dynamic operating conditions.

### 5. Results and Comparisons

After completing the data preprocessing and dataset construction, the proposed Bayesian optimization-based ITSC fault diagnosis model is used to analyze the three-phase current signals. The whole procedure is carried out offline. The hyperparameter combinations to be optimized and their search space are shown in Table 4.

**Table 4.** Hyperparameters to be optimized.

Hyperparameters	Search Intervals	Data Types	Transform	Best Result
$L_{init}$	$[1 \times 10^{-5} \ 1]$	real	log	$1.6227 \times 10^{-4}$
$G_1$	$[0.5 \ 1]$	real	log	0.8747
$L_{2R}$	$[1 \times 10^{-10} \ 1 \times 10^{-2}]$	real	log	$7.4777 \times 10^{-8}$
$d_1$	$[2 \ 8]$	integer	none	5
$d_2$	$[4 \ 16]$	integer	none	9
$d_3$	$[2 \ 8]$	integer	none	6
$w_1$	$[2 \ 60]$	integer	none	18
$w_2$	$[40 \ 160]$	integer	none	66
$w_3$	$[2 \ 60]$	integer	none	38
$P$	$[1 \times 10^{-5} \ 1]$	real	log	$5.1585 \times 10^{-4}$

Among them,  $L_{init}$  represents the initial learning rate of the entire model,  $G_1$  represents the gradient optimization coefficient of the Adam optimizer,  $L_{2R}$  represents the  $L_2$  regularization coefficient,  $P$  represents the probability of dropout, and the data type for the above hyperparameters is all real numbers. The depths of the three convolutional layers are denoted by  $d_1$ ,  $d_2$ , and  $d_3$ , and the numbers of convolutional kernels for each layer are represented by  $w_1$ ,  $w_2$ , and  $w_3$ . Both the number of convolutional kernels and the depth of the convolutional layers are integer types. The size of the convolutional kernels is set to a fixed value of  $1 \times 3$ , the dilation rate is set to 2, the learning rate decay step size is set to 20, and the decay factor is set to 0.1. “Transform” indicates whether the hyperparameters are searched on a logarithmic scale during the search process in the set space. Based on experience, the maximum number of iterations for Bayesian optimization is set to 60, with 40 training epochs per iteration. The values of the hyperparameters for the optimal combination obtained are shown in Table 4, and the corresponding schematic diagram of the optimal model architecture is presented in Figure 13.

To verify the advantages of the proposed improved CNN architecture and to compare the performance improvements brought by different combinations of enhancements, several models are constructed: a conventional CNN model without any architecture enhancement (CNN); a conventional CNN model with the residual network structure (Res); a CNN model that shares both residual and multi-scale networks (MK-Res); and a CNN model that shares both residual and attention mechanisms (SE-Res). The architecture hyperparameters of the feature layers for these four models are set consistently with the proposed improved CNN model. The training hyperparameters for the four models were obtained through

hyperparameter tuning using Bayesian optimization. The error loss and validation accuracy of the five models throughout the training process were recorded as they varied with the number of training epochs, and the results are compared in Figure 14.

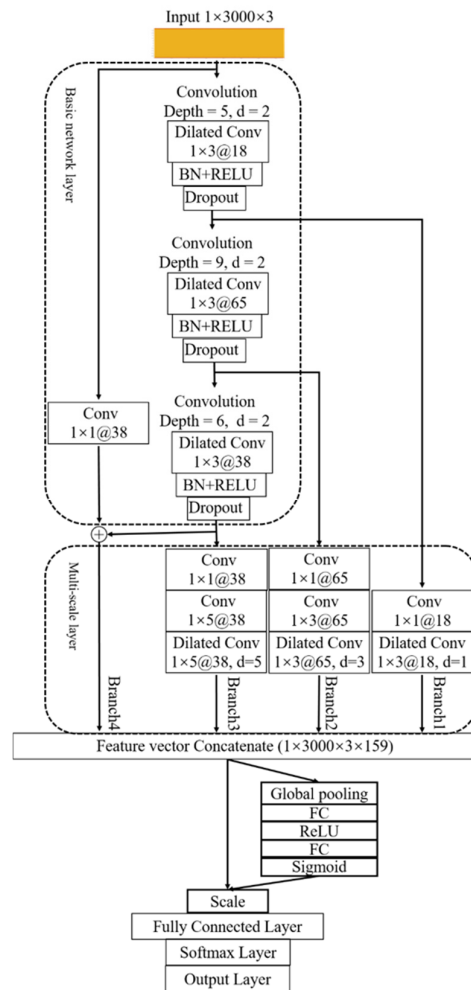
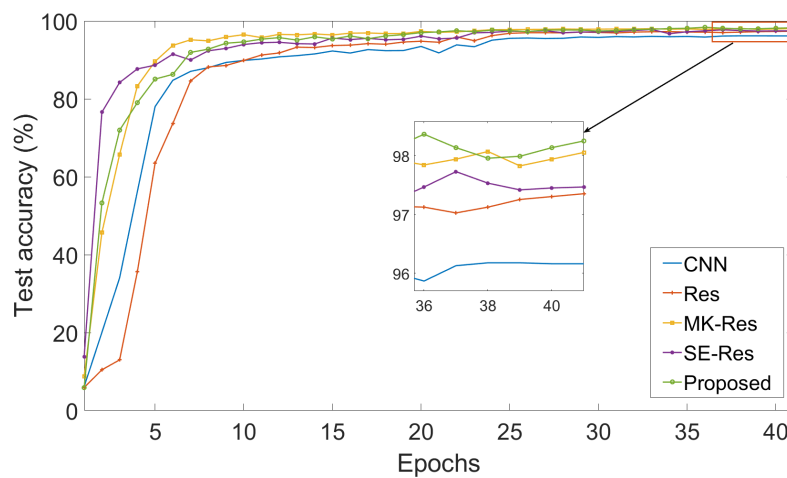
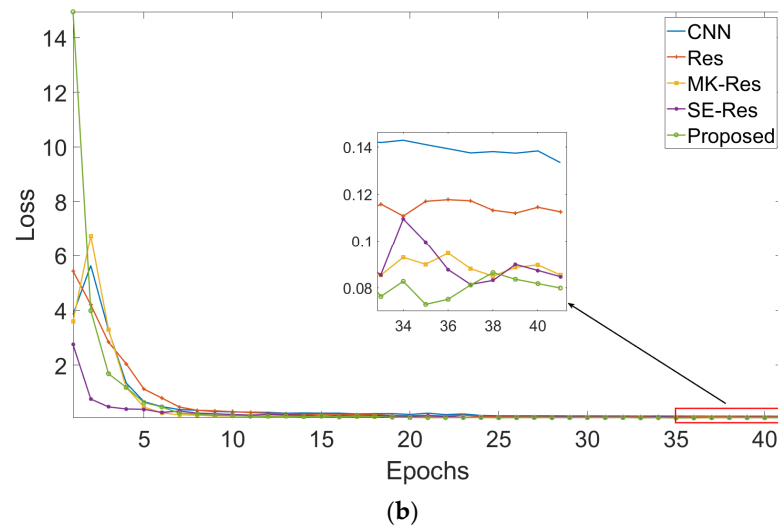


Figure 13. Schematic diagram of the optimal model structure obtained using a Bayesian optimization algorithm.



(a)

Figure 14. Cont.



**Figure 14.** Comparison diagram of ITSC fault diagnosis results of the five models. (a) Overall testing accuracy trends of the five algorithms. (b) The trend of loss function changes in each algorithm.

Figure 14a and Table 5 compare the test accuracy trends of the five models as training epochs progress. It can be seen that, compared to the CNN model, all four improved models exhibit varying extends of enhancement in the final test accuracy. The final test accuracy of the CNN model is 96.16%. The final test accuracy of the Res model is 97.35%, which represents an improvement of 1.19% over the CNN model. The MK-Res model achieves a final test accuracy of 98.06%, improving by 1.90% compared to the CNN model. The SE-Res model has a final test accuracy of 97.47%, an increase of 1.31% over the CNN model. The proposed model reaches a final validation accuracy of 98.25%, marking an improvement of 2.09% compared to the CNN model.

**Table 5.** Result comparison of different methods.

Method	Test Accuracy	Loss
CNN	96.16%	0.1333
Res	97.35%	0.1125
MK-Res	98.06%	0.0854
SE-Res	97.47%	0.0847
Proposed	98.25%	0.0799

It is equally important to note that the feature extraction layers of all five models are consistent, with the differences between the models lying in the use of various improved architectures within the feature extraction layers. From the final results, it is evident that the residual network structure, multi-scale network structure, and channel attention mechanism all contribute to varying degrees of performance improvement, with the combination of all three achieving the most significant enhancement. Based on the principles of these improved architectures, the channel attention mechanism is able to discard irrelevant parameters during training, thus not only improving the model's performance but also accelerating the overall convergence speed. The residual network structure helps the model train more effectively and improves recognition accuracy. The multi-scale network architecture enriches the scale of the extracted fault features, enhancing the diversity of the fault feature space, which, in turn, boosts the model's final recognition accuracy. From the final results, it can be seen that for complex tasks such as ITSC fault severity recognition, the multi-scale architecture has the greatest impact on the model's performance, followed by the channel attention mechanism, with the combined use of all three yielding the best results. Figure 14b and Table 5 show the comparison of the loss trends of the five models as training epochs progress. From the figure, it is evident that the final error losses of the four





compared to the CNN model, the *F1* scores for different fault severity labels reveal mixed performances among the five models. The four improved models exhibit clear advantages in labels associated with lighter fault degrees, showing significant increases in *F1* scores. Among the 17 different fault classifications, the proposed models achieved the highest scores in 12 of the fault categories, demonstrating the best performance.

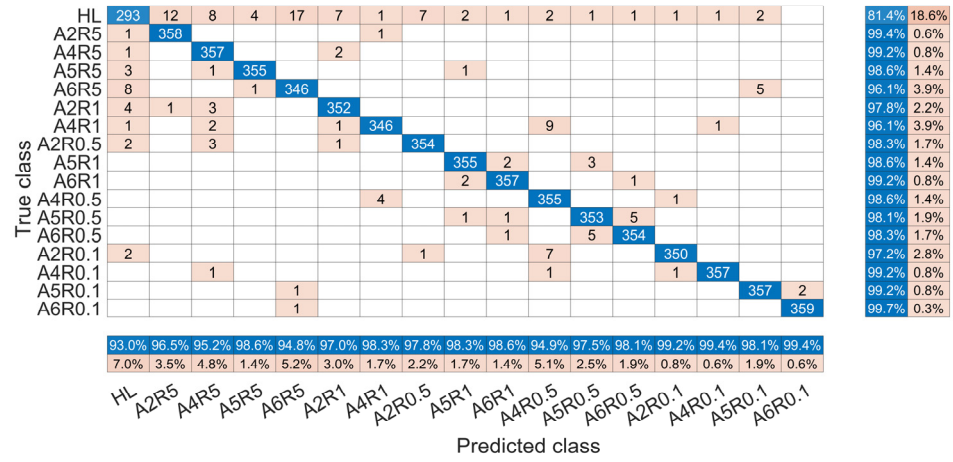


Figure 16. The confusion matrix of the Res model.

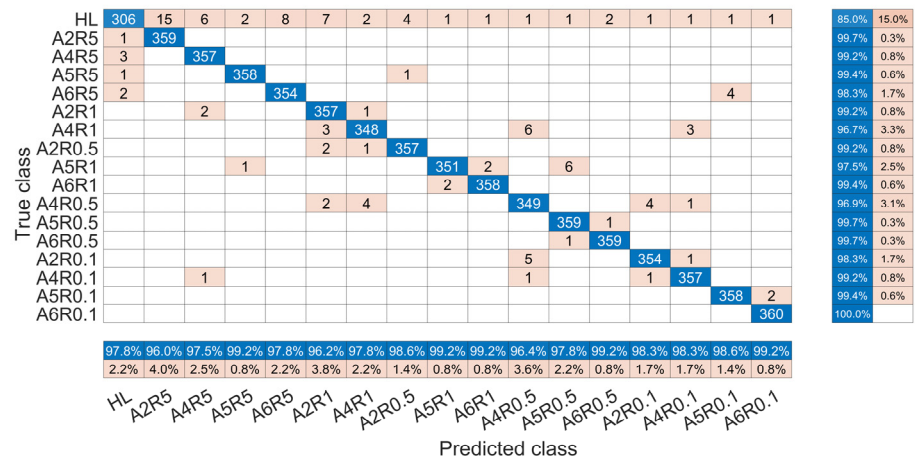


Figure 17. The confusion matrix of the MK-Res model.

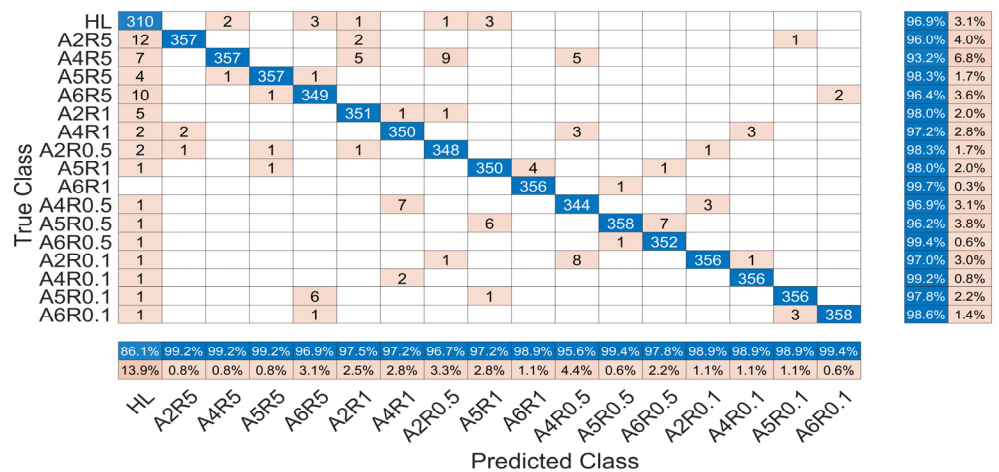


Figure 18. The confusion matrix of the SE-Res model.

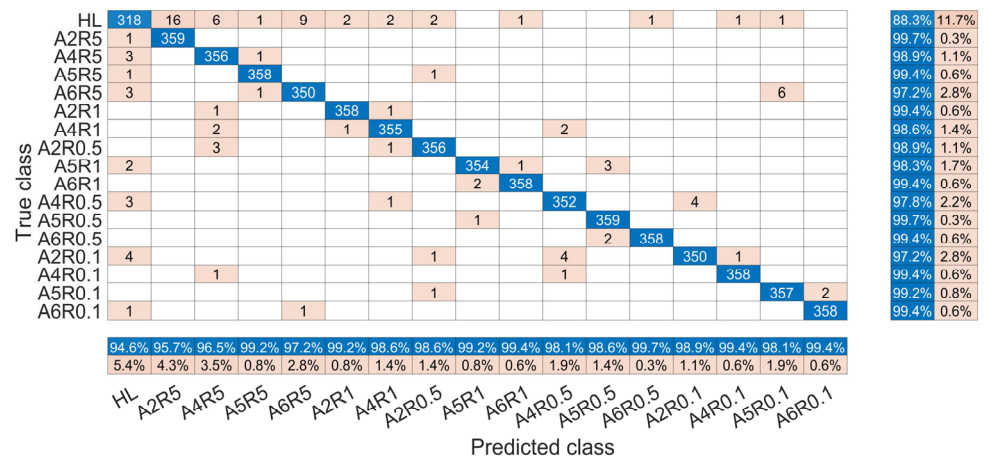


Figure 19. The confusion matrix of the proposed model.

Table 6. F1 score comparison of five methods under different fault labels.

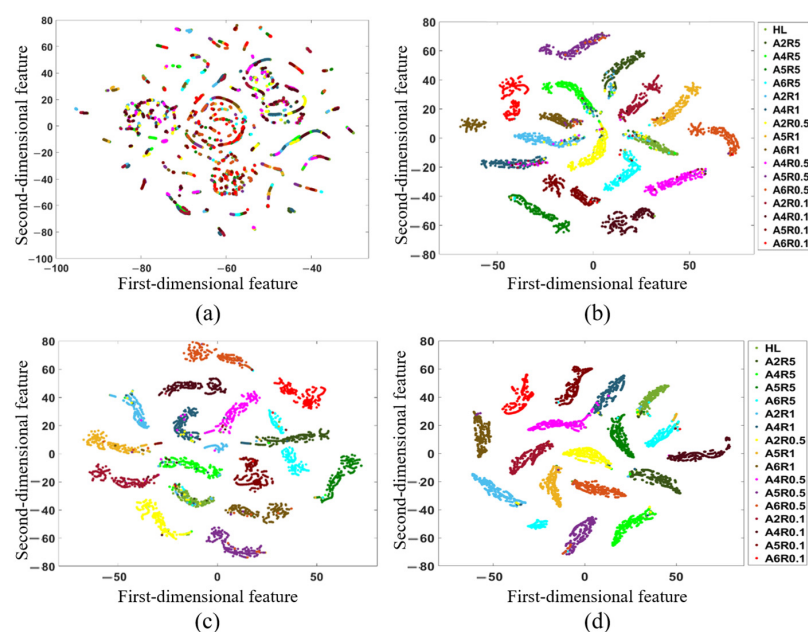
Label	CNN (%)	Res (%)	MK-Res (%)	SE-Res (%)	Proposed (%)
Acc	96.16	97.35	98.06	97.47	98.25
HL	83.05	86.81	90.94	91.18	91.38
A2R5	96.16	97.95	97.82	97.54	97.69
A4R5	95.38	97.14	98.35	96.10	97.67
A5R5	98.07	98.61	99.31	98.76	99.31
A6R5	95.74	95.45	98.06	96.68	97.22
A2R1	95.00	97.37	97.67	97.77	99.31
A4R1	95.69	97.19	97.21	97.22	98.61
A2R0.5	97.66	98.06	98.89	97.48	98.75
A5R1	96.41	98.47	98.32	97.63	98.74
A6R1	97.37	98.89	99.31	99.30	99.44
A4R0.5	94.68	96.73	96.68	96.22	97.91
A5R0.5	96.45	97.78	98.76	97.81	99.17
A6R0.5	97.90	98.20	99.45	98.60	99.58
A2R0.1	97.37	98.18	98.33	97.94	98.04
A4R0.1	98.07	99.30	98.76	99.03	99.44
A5R0.1	99.03	98.62	99.03	98.34	98.62
A6R0.1	100	99.58	98.06	99.03	99.44

To reduce the impact of randomness, the diagnostic results from five repeated experiments were averaged, and the standard deviation of the results for each experiment was calculated. Additionally, the time taken by the model to recognize the test set in each experiment was recorded, and the average recognition time per data slice was computed, as shown in Table 7. From Table 7, it can be seen that the proposed model not only achieves the highest average accuracy of 98.20% but also has the smallest standard deviation of 0.105%, indicating both good accuracy and stability. The complexity of the deep learning model is represented by the total number of adjustable parameters, including weights and biases, as shown in Table 7. It is evident from the table that each improvement measure added to the model increases its complexity. The proposed model, incorporating the most improvements, has the highest complexity. The average recognition time of the model reflects its data processing speed. From the table, it can be seen that the proposed model has the longest average recognition time, reaching 1.14 ms, but this is still much smaller than the 0.2 s sampling time per data slice, meaning the model can meet the required time for data processing.

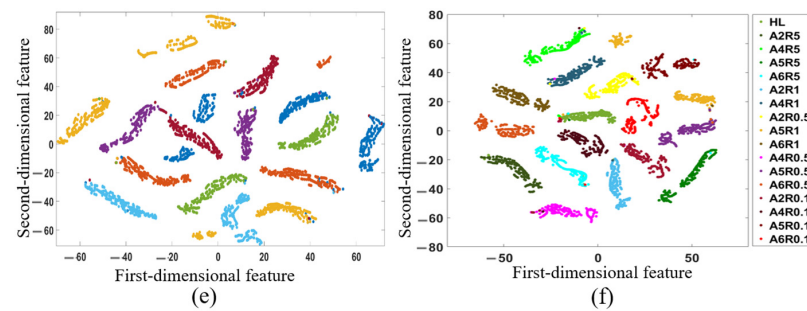
**Table 7.** Stability comparison of different methods and the average computation time for each slice of test data.

Method	Average Accuracy (%)	Average Computation Time (ms)	Model Complexity
CNN	96.19 ± 0.133	0.946 ± 0.0195	5,552,359
Res	97.19 ± 0.137	0.652 ± 0.0039	5,952,918
MK-Res	97.91 ± 0.112	0.852 ± 0.0693	24,510,130
SE-Res	97.59 ± 0.147	0.689 ± 0.0431	5,954,419
Proposed	98.20 ± 0.105	1.140 ± 0.0052	24,689,824

Through a comprehensive analysis of the five ITSC fault diagnosis models, it is evident that the proposed model exhibits the best performance in terms of the final test accuracy and stability. In the *F1* scores across 17 different fault severity labels, the proposed model demonstrates overall superior performance, making it the best-performing model among the five. To validate the feature learning capability of the proposed model, the t-distribution stochastic neighbor-embedding algorithm (T-SNE) was used to visualize the features from the final output layer of the ITSC fault diagnosis model, and the results were compared with the other four models. The two-dimensional visualization results are shown in Figure 20. From the figure, it can be observed that the feature map contains 17 colors, each corresponding to a specific fault severity label, with each point representing a data sample. Figure 20a shows the feature distribution of the input layers of each model. It is apparent that the feature distribution of the input data is chaotic, with significant overlap among samples of different colors, making it difficult to discern the fault severity of the corresponding samples in the dataset based solely on the input data. Figure 20b–f display the feature distribution maps of the classification layers for the CNN model, Res model, MK-Res model, SE-Res model, and the proposed model, respectively. From these figures, it can be seen that after feature extraction by the model, the samples within the same ITSC fault severity labels exhibit good intra-class clustering characteristics. The proposed ITSC fault diagnosis model has the fewest misclassified sample points compared to the other four models. Additionally, the boundaries between different ITSC fault labels are clear and more distant, resulting in better separation characteristics among different categories. Thus, the proposed model demonstrates superior feature learning and discrimination capabilities.



**Figure 20.** Cont.



**Figure 20.** Comparison of visualized features extracted by different algorithms. (a) Feature map of the input data. (b) Feature map of the CNN model. (c) Feature map of the Res model. (d) Feature map of the MK-Res model. (e) Feature map of the SE-Res model. (f) Feature map of the proposed model.

## 6. Conclusions

In this research, a novel Bayesian optimization-based improvement algorithm was proposed for the enhancement of an ITSC fault diagnosis model. The results indicate that the proposed method is applicable under both dynamic and steady-state operating conditions. Firstly, a fault model was proposed for the analysis of ITSC faults, and a severity index was derived for the guiding of ITSC fault severity settings. Secondly, a residual network, multi-scale network, and attention mechanism were applied to prevent network functionality degradation, increase the richness of the extracted features, and enhance the proportion of useful features in the model, ultimately enhancing the network's performance. Then, to facilitate the optimization of architecture hyperparameters, the model's feature extraction layers were divided into a three-stage architecture based on the characteristics of the features extracted by convolutional networks. Furthermore, the ITSC fault motor test was carried out with the fault severity set to 17 different levels. The proposed algorithm was conducted to analyze the three-phase current signals that were collected in the motor test. Conventional CNN, Res, MK-Res, and SE-Res models were also employed in the same dataset for comparison. The results illustrated that the proposed algorithm not only achieved the best final test accuracy but also provided the best feature extraction capability.

This study aims to improve the accuracy of ITSC fault diagnosis using deep learning methods. The approach relies on supervised training with a sufficient and balanced sample size. However, in practical applications, challenges such as an insufficient sample size, imbalanced sample distribution, or lack of labeled samples often arise. Under these conditions, the application of the proposed method would be significantly limited. Future research will focus on addressing these challenges and improving the accuracy of ITSC fault diagnosis under such adverse conditions.

**Author Contributions:** Conceptualization, M.W., Q.S. and W.L.; methodology, M.W. and W.L.; software, M.W. and W.L.; validation, M.W.; formal analysis, M.W. and W.L.; investigation, M.W., Q.S. and W.L.; resources, Q.S.; data curation, M.W. and W.L.; writing—original draft preparation, M.W.; writing—review and editing, M.W., H.L., P.S. and W.L.; visualization, M.W.; supervision, Q.S.; project administration, M.W. and Q.S.; funding acquisition, Q.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this study are available from the corresponding author upon reasonable request ([wangmingsheng\\_08@163.com](mailto:wangmingsheng_08@163.com)).

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

## References

1. Xie, F.; Li, G.; Liu, H.; Sun, E.; Wang, Y. Advancing Early Fault Diagnosis for Multi-Domain Agricultural Machinery Rolling Bearings through Data Enhancement. *Agriculture* **2024**, *14*, 112. [[CrossRef](#)]



2. Xie, F.; Sun, E.; Wang, L.; Wang, G.; Xiao, Q. Rolling Bearing Fault Diagnosis in Agricultural Machinery Based on Multi-Source Locally Adaptive Graph Convolution. *Agriculture* **2024**, *14*, 1333. [[CrossRef](#)]
3. Xie, F.; Wang, Y.; Wang, G.; Sun, E.; Fan, Q.; Song, M. Fault Diagnosis of Rolling Bearings in Agricultural Machines Using SVD-EDS-GST and ResViT. *Agriculture* **2024**, *14*, 1286. [[CrossRef](#)]
4. Wang, H. Comparative investigation and evaluation of electric-drive seed-metering systems across diverse speed ranges for enhanced high-precision seeding applications. *Comput. Electron. Agric.* **2024**, *222*, 108976. [[CrossRef](#)]
5. Scolaro, E.; Estevez, M.P.; Renzi, M.; Mattetti, M. Electrification of Agricultural Machinery: A Review. *IEEE Access* **2021**, *9*, 164520–164541. [[CrossRef](#)]
6. Ni, H.; Lu, L.; Sun, M.; Bai, X.; Yin, Y. Research on Fault Diagnosis of PST Electro-Hydraulic Control System of Heavy Tractor Based on Support Vector Machine. *Processes* **2022**, *10*, 791. [[CrossRef](#)]
7. Riera-Guasp, M.; Antonino-Daviu, J.A.; Capolino, G.A. Advances in electrical machine, power electronic, and drive condition monitoring and fault detection: State of the art. *IEEE Trans. Ind. Electron.* **2015**, *62*, 1746–1759. [[CrossRef](#)]
8. Li, Y.; Liu, Y.; Ji, K.; Zhu, R. A Fault Diagnosis Method for a Differential Inverse Gearbox of a Crawler Combine Harvester Based on Order Analysis. *Agriculture* **2022**, *12*, 1300. [[CrossRef](#)]
9. Husari, F.; Seshadrinath, J. Stator Turn Fault Diagnosis and Severity Assessment in Converter-Fed Induction Motor Using Flat Diagnosis Structure Based on Deep Learning Approach. *IEEE J. Emerg. Sel. Top. Power Electron.* **2023**, *11*, 5649–5657. [[CrossRef](#)]
10. Mahmoudi, A.; Jlassi, I.; Cardoso, A.J.M.; Yahia, K.; Sahraoui, M. Inter-Turn Short-Circuit Faults Diagnosis in Synchronous Reluctance Machines, Using the Luenberger State Observer and Current's Second-Order Harmonic. *IEEE Trans. Ind. Electron.* **2022**, *69*, 8420–8429. [[CrossRef](#)]
11. Choe, H.O.; Lee, M.-H. Artificial Intelligence-Based Fault Diagnosis and Prediction for Smart Farm Information and Communication Technology Equipment. *Agriculture* **2023**, *13*, 2124. [[CrossRef](#)]
12. Yang, G.; Cheng, Y.; Xi, C.; Liu, L.; Gan, X. Combine Harvester Bearing Fault-Diagnosis Method Based on SDAE-RCmvMSE. *Entropy* **2022**, *24*, 1139. [[CrossRef](#)] [[PubMed](#)]
13. Liu, Z.; Sun, W.; Chang, S.; Zhang, K. Corn Harvester Bearing Fault Diagnosis Based on ABC-VMD and Optimized EfficientNet. *Entropy* **2023**, *25*, 1273. [[CrossRef](#)] [[PubMed](#)]
14. Jiang, W.; Shan, Y.; Xue, X.; Ma, J.; Chen, Z.; Zhang, N. Fault Diagnosis for Rolling Bearing of Combine Harvester Based on Composite-Scale-Variable Dispersion Entropy and Self-Optimization Variational Mode Decomposition Algorithm. *Entropy* **2023**, *25*, 1111. [[CrossRef](#)] [[PubMed](#)]
15. Parvin, F.; Faiz, J.; Qi, Y.; Kalhor, A.; Akin, B. A Comprehensive Interturn Fault Severity Diagnosis Method for Permanent Magnet Synchronous Motors Based on Transformer Neural Networks. *IEEE Trans. Ind. Inform.* **2023**, *19*, 10923–10933. [[CrossRef](#)]
16. Li, J.; Li, X.; Li, Y.; Zhang, Y.; Yang, X.; Xu, P. A New Method of Tractor Engine State Identification Based on Vibration Characteristics. *Processes* **2023**, *11*, 303. [[CrossRef](#)]
17. Fan, W.; Yang, C.; Chen, C.; He, C.; Yuan, Y.; Li, Y. Adaptive Feature-Oriented Dictionary Learning and Sparse Classification Framework for Bearing Compound Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 3518010. [[CrossRef](#)]
18. Lee, H.; Jeong, H.; Koo, G.; Ban, J.; Kim, S.W. Attention Recurrent Neural Network-Based Severity Estimation Method for Interturn Short-Circuit Fault in Permanent Magnet Synchronous Machines. *IEEE Trans. Ind. Electron.* **2021**, *68*, 3445–3453. [[CrossRef](#)]
19. Xu, L.; Zhang, G.; Zhao, S.; Wu, Y.; Xi, Z. Fault Diagnosis of Tractor Transmission System Based on Time GAN and Transformer. *IEEE Access* **2024**, *12*, 107153–107169. [[CrossRef](#)]
20. Husari, F.; Seshadrinath, J. Early Stator Fault Detection and Condition Identification in Induction Motor Using Novel Deep Network. *IEEE Trans. Artif. Intell.* **2022**, *3*, 809–818. [[CrossRef](#)]
21. Zhao, J.; Guan, X.; Li, C.; Mou, Q.; Chen, Z. Comprehensive Evaluation of Inter-Turn Short Circuit Faults in PMSM Used for Electric Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 611–621. [[CrossRef](#)]
22. Zhang, Y.; Liu, G.; Zhao, W.; Zhou, H.; Chen, Q.; Wei, M. Online Diagnosis of Slight Interturn Short-Circuit Fault for a Low-Speed Permanent Magnet Synchronous Motor. *IEEE Trans. Transp. Electrif.* **2021**, *7*, 104–113. [[CrossRef](#)]
23. Ehya, H.; Skreien, T.N.; Nysveen, A. Intelligent Data-Driven Diagnosis of Incipient Interturn Short Circuit Fault in Field Winding of Salient Pole Synchronous Generators. *IEEE Trans. Ind. Inform.* **2022**, *18*, 3286–3294. [[CrossRef](#)]
24. Zhao, M.; Zhong, S.; Fu, X.; Tang, B.; Pecht, M. Deep Residual Shrinkage Networks for Fault Diagnosis. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4681–4690. [[CrossRef](#)]
25. Jafari, A.; Faiz, J.; Jarrahi, M.A. A Simple and Efficient Current-Based Method for Interturn Fault Detection in BLDC Motors. *IEEE Trans. Ind. Inform.* **2021**, *17*, 2707–2715. [[CrossRef](#)]
26. Hu, R.; Wang, J.; Mills, A.R.; Chong, E.; Sun, Z. Current-Residual-Based Stator Interturn Fault Detection in Permanent Magnet Machines. *IEEE Trans. Ind. Electron.* **2021**, *68*, 59–69. [[CrossRef](#)]
27. Hang, J.; Zhang, J.; Cheng, M.; Huang, J. Online Interturn Fault Diagnosis of Permanent Magnet Synchronous Machine Using Zero-Sequence Components. *IEEE Trans. Power Electron.* **2015**, *30*, 6731–6741. [[CrossRef](#)]
28. Yao, Y.; Wang, J.; Xie, M. Adaptive residual CNN-based fault detection and diagnosis system of small modular reactors. *Appl. Soft Comput.* **2022**, *114*, 108064. [[CrossRef](#)]
29. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271.

30. Shao, S.; Yan, R.; Lu, Y.; Wang, P.; Gao, R.X. DCNN-Based Multi-Signal Induction Motor Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 2658–2669. [[CrossRef](#)]
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Xu, Y.; Yan, X.; Sun, B.; Liu, Z. Hierarchical Multiscale Dense Networks for Intelligent Fault Diagnosis of Electromechanical Systems. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3505312. [[CrossRef](#)]
33. Li, Y.; Du, L.; Wei, D. Multiscale CNN Based on Component Analysis for SAR ATR. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5211212. [[CrossRef](#)]
34. Weng, C.; Lu, B.; Gu, Q. A multi-scale kernel-based network with improved attention mechanism for rotating machinery fault diagnosis under noisy environments. *Meas. Sci. Technol.* **2022**, *33*, 055108. [[CrossRef](#)]
35. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Volume 42, pp. 2011–2023.
36. Gelbart, M.A.; Snoek, J.; Adams, R.P. Bayesian Optimization with Unknown Constraints. *arXiv* **2014**, arXiv:1403.5607.
37. Han, S.; Eom, H.; Kim, J.; Park, C. Optimal DNN architecture search using Bayesian Optimization Hyperband for arrhythmia detection. In Proceedings of the 2020 IEEE Wireless Power Transfer Conference (WPTC), Seoul, Republic of Korea, 15–19 November 2020; pp. 357–360.

**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.