An Adaptive Sliding Window-Based Online Data-Driven Diagnosis Method for Inverter Open-Circuit Fault in WRSM Drive Systems

Chenyun Wu, Rabia Sehab, Javier Ojeda, *Member, IEEE*, Ahmad Akrad, and Cristina Morel

***Abstract*—This article presents a novel data-driven real-time open-circuit fault diagnosis strategy for three-phase inverters, integrating the adaptive sliding window (ASWIN) method with machine learning algorithms, enabling 22 state classifications. In addition to stationary single faults, the proposed approach further targets two fault modes that have received less emphasis in the existing literature: nonstationary single faults and mixed stationary faults. Dynamic adaptation of the sliding window size based on signal characteristics improves diagnostic accuracy across the machine's entire speed range while reducing latency. A range-based evaluation strategy is proposed to assess the dynamic diagnostic performance when multiple operating states coexist within the same sliding window. First, machine learning models trained on multiscale experimental current data demonstrated robust fault diagnosis under various operating conditions. Second, a dynamic scaling factor for window size calculation was developed, incorporating signal metrics, such as dominant frequency proportion, signal variation rate, and signal-to-noise ratio, enabling real-time adjustment of window sizes based on current information and short-term prediction. Third, offline optimization determines speed-specific parameter combinations, facilitating dynamic self-tuning of ASWIN parameters according to variable machine speed. Experimental results validated the proposed strategy, achieving 97.70% accuracy with a diagnosis time of approximately 60 ms, corresponding to a range of 0.4–2.4 current fundamental periods across a speed range of 200–1200 r/min.**

***Index Terms*—Adaptive sliding window (ASWIN), fault diagnosis, machine learning (ML), open-circuit (OC) faults, three-phase inverter.**

I. Introduction

THREE-PHASE pulsewidth modulation (PWM) inverters are widely adopted across industrial sectors for their high performance, compact design, and simple control structure, for instance, the high-performance drive system for wound rotor synchronous machines (WRSMs). However, under harsh conditions—including high temperatures, high-frequency operation, and factors like component aging and physical damage—motor drive systems are prone to faults [1]. Industrial surveys indicate that semiconductors are the most failure-prone components in variable-speed drives, especially in high-power and heavy-load applications [2].

Open-circuit (OC) and short-circuit (SC) faults in power semiconductors present significant challenges for inverters [3]. SC faults cause rapid current surges, risking inverter damage and cascading failures that destabilize or shut down systems [4]. These faults are typically managed by hardware protection, such as fuses and circuit breakers [5]. Conversely, while not immediately catastrophic, OC faults distort current waveforms, induce phase imbalances, and impair dynamic performance [6]. In motor drives, asymmetric currents from OC faults cause electromagnetic torque ripples [7]. Unlike SC faults, OC faults lack hardware protection and, if undiagnosed, can lead to secondary failures and system damage [8]. Therefore, a high-accuracy, low-latency algorithm for real-time OC fault diagnosis is crucial for early detection and intervention, ensuring the reliability and safety of motor drive systems.

Inverter fault diagnosis typically involves three key steps: information acquisition, feature extraction, and fault state identification [9]. Based on their implementation, fault diagnosis methods are classified as model-based, signal-based, or knowledge-based (data-driven). Model-based methods identify faults by analyzing the residuals between measured signals and ideal model outputs, leveraging techniques like observer-based [10], parity space [11], and parameter estimation methods [12]. These methods offer fast detection and real-time performance but are sensitive to system parameter variations and modeling inaccuracies. Recent efforts have focused on enhancing robustness against such challenges. For example, a difference observer-based method with adaptive thresholds [13] mitigates misdiagnoses caused by varying operating conditions, demonstrating robustness against parameter changes. However, it increases hardware complexity by requiring additional voltage sensors. To address this, a sliding mode observer-based approach [14] improves detection accuracy for nonlinear systems with disturbances without extra sensors, showing resilience to load torque ripples. Similarly, a flux linkage observer based on a

Received 2 September 2025; revised 13 October 2025 and 18 November 2025; accepted 10 December 2025. Date of publication 15 December 2025; date of current version 25 February 2026. The work of Chenyun Wu was supported by China Scholarship Council (CSC) for his Ph.D. project, under Grant 202108070008. Recommended for publication by Associate Editor J. Ye. *(Corresponding author: Chenyun Wu.)*

Chenyun Wu is with ESTACA'Lab–Laval, ESTACA, 53000 Laval, France, and also with Université Paris-Saclay, 91190 Gif-sur-Yvette, France (e-mail: chenyun.wu@estaca.fr).

Rabia Sehab, Ahmad Akrad, and Cristina Morel are with ESTACA'Lab–Laval, ESTACA, 53000 Laval, France (e-mail: rabia.sehab@estaca.fr; ahmad.akrad@estaca.fr; cristina.morel@estaca.fr).

Javier Ojeda is with SATIE, CNRS, ENS Paris-Saclay, Université Paris-Saclay, 91190 Gif-sur-Yvette, France (e-mail: javier.ojeda@ens-paris-saclay.fr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2025.3644145.

Digital Object Identifier 10.1109/TPEL.2025.3644145

hybrid voltage–current model [15] uses a second-order moving average filter to enhance disturbance robustness while reducing hardware complexity. Despite these advancements, observer-based methods face limitations at zero or extremely low speeds, where minimal three-phase current amplitudes prevent accurate estimation of back electromotive force (EMF) and flux linkage, rendering the system unobservable.

Signal-based inverter fault diagnosis methods extract features from measured signals using signal processing algorithms to analyze distortion characteristics and achieve fault diagnosis. These methods do not require precise system parameters but are sensitive to noise and rely on empirical thresholds for fault localization and classification. For instance, Wang et al. [16] introduced a hierarchical method based on voltage vector estimation and current path reconstruction, mitigating misdiagnoses caused by sampling errors and dead-time effects. Similarly, Wang et al. [17] applied complete ensemble empirical mode decomposition (CEEMD) to voltage signals for short-term energy feature extraction of intrinsic mode function (IMF), enhancing robustness against mode aliasing and noise. While both methods provide relatively fast diagnostics, they require additional voltage measurement modules. In contrast, Zhou et al. [18] presented a seasonal-trend decomposition-based method by simplifying current waveforms and performing approximate integration to extract characteristic values. Wu et al. [19] proposed an entropy-based approach to classify OC faults using fuzzy and slope entropy algorithms to calculate local entropy values of current signals within a sliding window. While effective, their robustness diminishes with small current amplitudes, complicating fault classification. Beyond current and voltage, Abari et al. [20] proposed a noninvasive method using dc bus electromagnetic characteristics for fault identification. Although effective, this approach requires external antennas or filters, increasing complexity and cost.

Recently, real-time data-driven fault diagnosis has gained attention due to improvements in computational power and the widespread adoption of end-to-end approaches. The core methodology involves extracting the mapping relationships between measured data and operational states or fault types through large volumes of real-world data, offering robustness against parameter variations and transient dynamics [21], [22]. For instance, Kiranyaz et al. [23] proposed an adaptive 1-D convolutional neural network (CNN) for OC fault localization, achieving high diagnostic accuracy, while Deng et al. [24] combined a sliding time window feature extraction algorithm with a 2-D CNN, eliminating the need for empirical thresholds. However, these methods lack validation regarding robustness under dynamic operating conditions. Further studies have explored hybrid approaches. For instance, Xia et al. [22] introduced an OC fault diagnosis algorithm by combining a sliding window with a random vector functional link (RVFL) network for induction motor drives, integrating fast Fourier transform (FFT) and ReliefF for feature extraction with multiobjective optimization for parameter tuning. Despite resilience to disturbances, its performance at low speeds is unverified. Cai et al. [6] applied principal component analysis (PCA) to reduce dimensionality, improving fault diagnosis accuracy but increasing implementation costs due to voltage sensors. Wu et al. [25] introduced a machine learning (ML)-based diagnosis method, employing k-nearest neighbors (KNN) for fault state preprocessing and support vector machine (SVM) and extreme gradient boosting (XGBoost) for OC fault classification. While high diagnostic accuracy, the method lacks validation of its effectiveness under dynamic operating conditions. In addition, Xia and Xu [26] combined manifold feature learning with an extreme learning machine (ELM) to propose a transfer learning-based fault diagnosis approach. The method showed insensitivity to significant parameter variations in systems with identical structures, but challenges remain regarding generalizability under variable-speed conditions. Gou et al. [27] demonstrated excellent diagnosis capability by combining FFT with an RVFL classifier, accurately identifying 21 insulated-gate bipolar transistor (IGBT) OC faults and nine current-sensor faults with a delay of approximately 22 ms. While highly effective, the method relies on large hidden layers and careful parameter tuning, which may increase its complexity and training overhead. A concise overview of representative fault diagnosis methods and their advantages and limitations is summarized in Table I.

Recent studies have proposed innovative OC fault diagnostic methods, achieving promising results, particularly with data-driven approaches. These methods excel in classifying diverse fault types and exhibit strong robustness against parameter variations and disturbances. However, most lack a discussion on nonstationary faults, which are more representative of real-world scenarios involving continuous speed variations in motor drives. Their applicability across the entire speed range, particularly at zero or low speeds, remains underexplored. Low-speed conditions, characterized by minimal current amplitudes, pose significant challenges for feature extraction and fault classification. For instance, Gou et al. [26] achieved 100% diagnostic accuracy under steady-state conditions near 100 rad/s with commendable disturbance robustness but did not validate performance at lower speeds. Similarly, Zhou et al. [18] demonstrated rapid fault detection with 100% accuracy but limited validation to rotor speeds exceeding 600 r/min, leaving low-speed performance unaddressed. Furthermore, studies employing sliding window methods typically use fixed-length windows under variable-speed conditions [22], [24] (e.g., 1–4 cycles). Such approaches may fail to capture a complete signal cycle at low frequencies or become overly conservative at high frequencies, adversely affecting feature extraction. In addition, Diao et al. [28] proposed a horizon-adaptive period correction method that utilizes phase-domain period extraction and virtual time-domain period correction to map a variable-frequency signal into a single fundamental period. Liu et al. [29] proposed an adaptive secondary sampling mechanism for multiphase driving to ensure that the input of a deep sparse filtering network always covers exactly one fundamental period. Although these two variable sampling strategies have been satisfactorily validated under nonstationary conditions, the single-cycle synchronization constraint limits flexibility at extremely low speeds.

This article proposes a diagnostic strategy for three-phase inverter OC faults in WRSM machine drive systems operating under variable speed and torque load conditions. The approach integrates an adaptive sliding window (ASWIN) algorithm with ML models to achieve high accuracy and low latency

TABLE I
SUMMARY OF EXISTING FAULT DIAGNOSIS METHODS FOR POWER ELECTRONIC CONVERTERS

Method	Ref.	Algorithm	Required signals	Fault types	Advantages	Limitations
Model-based	[10]	Observer residual analysis	Input/output voltages and currents	Interleaved Buck converter OC	Fast response Suitable for deployment	Extra sensors needed Requires accurate model
	[11]	Parity-space fault detection	Three-phase voltages and currents	Current/voltage sensor faults	FPGA deployment Real-time detection implementation	Few types of diagnostic faults Sensitive to noise
	[12]	Adaptive parameter estimation	Voltage and current	Passive component faults(open, short)	High accuracy Combines prognosis and diagnosis	Few types of diagnostic faults Sensitive to variations
	[13]	Differential current observer	Three-phase currents	PMSM inverter OC	Fast response Robust to load/speed changes	Requires accurate model
	[14]	Adaptive Sliding-Mode Observer	Three-phase currents DC bus voltage	IGBT OC Sensor faults	Fast response Multiple fault diagnosis	Complex observer model Sensitive to parameter variations
	[15]	Hybrid flux observer	Three-phase currents DC bus voltage	Single/multiple IGBT OC	Very low latency Robust under motoring/regeneration	Requires accurate model Sensitive to parameter variations
Signal-based	[16]	Current path tracking	Three-phase currents Voltage vectors	T-type inverter OC	Fast and robust performance No extra sensors needed	High algorithm complexity Weak generalization
	[17]	CEEMD IMF energy analysis	Three-phase voltages	T-type inverter OC	High experimental accuracy Suitable for multilevel topologies	High computational cost Real-time challenge
	[18]	Seasonal-trend decomposition	Three-phase currents	IGBT OC	High accuracy; model-free Low computational cost	Threshold tuning needed Sensitive in varying conditions
	[19]	Fuzzy Entropy Slope Entropy	Three-phase currents	inverter IGBT OC	High accuracy No extra sensors needed	Sensitive to noise Empirical thresholds required
	[20]	EMI signature analysis	DC bus EMI	NPC inverter diode and IGBT OC	Accurate and robust to variations Low-cost non-intrusive diagnosis	Sensitive to EMI noise Environment-dependent
Data-driven	[22]	FFT + ReliefF + ELM + RVFL	Three-phase currents	Single/double IGBT OC	Fast response Robust to load/parameter variations	Requires large training dataset High training costs
	[23]	1D CNN	MMC voltages and currents	MMC submodule OC	Fast response Robust to variations	Requires extensive dataset Generalization limited
	[24]	2D CNN	Module voltages and currents	MMC submodule OC	High accuracy and fast response Suitable for multi-switch MMC	Requires large training dataset
	[25]	KNN anomaly detection SVM, RF, DT, LR, XGBoost	Three-phase currents	inverter IGBT OC	High accuracy and robust No extra sensors needed	Requires large training dataset Few types of diagnostic faults
	[26]	Manifold learning + ELM Transfer learning	Three-phase currents	Single/double IGBT OC	High cross-system accuracy Transferable with extra training	Needs target-domain data Complex parameter tuning
	[27]	FFT + ReliefF + RVFL	Three-phase currents DC bus, speed optional	Inverter IGBT OC Sensor faults	No extra hardware Robust to load/parameter variations	Requires large training dataset Complex parameter tuning

in diagnosing stationary and nonstationary faults. The main contributions of this work are summarized as follows.

1) *ASWIN framework:* This algorithmic framework is based on local signal metrics, such as dominant frequency, signal variation, and signal-to-noise ratio, and performs metric fusion using predefined weight parameters. The window size is dynamically adjusted using real-time current data and short-term predictions, enabling the window to adapt online to nonstationary and disturbed current signals.
2) *Range-based evaluation:* An evaluation strategy for mixed-label scenarios during dynamic transitions, extending the method in [30]. This strategy assesses classification accuracy and diagnostic latency, adapted to the real-time sliding window approach proposed in this study.
3) A speed-based parameter self-tuning method is proposed to enhance the ASWIN algorithm's robustness at different speeds. The optimal parameter combinations of the ASWIN algorithm can be automatically switched according to the changes in speed, and these optimal parameter combinations are achieved through offline segmented optimization.
4) *Comprehensive experimental validation:* Detailed validation framework results cover extreme-low, low, medium, and high speeds, nonstationary operating conditions, and torque disturbances, covering 22 healthy and faulty operating modes.

The rest of this article is organized as follows. Section II introduces the structure of the WRSM drive system utilized and the types and modes of inverter OC faults under investigation. Section III outlines the overall methodology for OC fault diagnosis based on a sliding window strategy and ML models. The proposed range-based evaluation strategy and ASWIN approach are detailed in Section IV. Section V presents and analyzes the experimental validation results. Finally, Section VI concludes this article.

II. SYSTEM MODELING AND FAULT DESCRIPTION

This section introduces the topology and analytical model of the three-phase inverter used in the WRSM drive system, which serves as the research object of this study. The fault types investigated in this article are also described in detail.

A. Modeling of the Three-Phase Inverter

The topology of the WRSM drive system is shown in Fig. 1 [19]. The three-phase inverter consists of six IGBTs

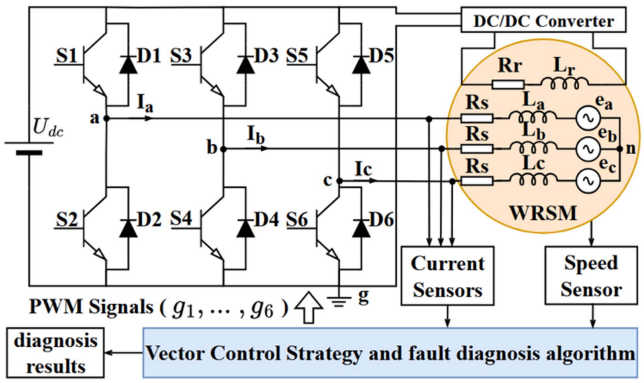

Fig. 1. Circuit topology diagram of the utilized WRSM drive system.

$(S_1, S_2, \ldots, S_6)$ and six diodes $(D_1, D_2, \ldots, D_6)$, with a dc link voltage, denoted as U_{dc}, to supply the inverter and chopper. The three-phase output currents, $I_a, I_b,$ and I_c, are connected to the three-phase stator windings of the WRSM. The WRSM is represented by an equivalent circuit model, where R_r and L_r denote the rotor winding resistance and inductance, R_s represents the stator resistance, $L_a, L_b,$ and L_c are the stator inductances, and $e_a, e_b,$ and e_c denote the back EMF.

The three-phase voltage equations of the WRSM drive system depicted in Fig. 1 can be expressed using the following mathematical model:

$$\begin{cases} u_{an} = R_s \cdot i_a + L_a \cdot \dfrac{di_a}{dt} + e_a \\ u_{bn} = R_s \cdot i_b + L_b \cdot \dfrac{di_b}{dt} + e_b \\ u_{cn} = R_s \cdot i_c + L_c \cdot \dfrac{di_c}{dt} + e_c \end{cases} \tag{1}$$

where $u_{an}, u_{bn},$ and u_{cn} are the phase voltages.

This formula describes the coupling between phase voltage, current, and induced EMF, indicating that the measured current signal will reflect an inverter's OC fault's physical mechanism and phenomenon. Furthermore, this data-driven approach based on three-phase current can be applied to WRSM drive systems and other three-phase inverter-powered machines.

Fig. 2 illustrates a complete cycle experimental test of the WRSM drive system under healthy conditions based on the speed reference shown in Fig. 2(a). Specifically, Fig. 2(a) presents an example of a speed reference and the corresponding motor speed response. Fig. 2(b) shows the three-phase current signals measured via current sensors on the inverter. Fig. 2(c) provides an enlarged view of the red-boxed region in Fig. 2(b), highlighting two key challenges for fault diagnosis in the full-speed operating range: first, the three-phase currents exhibit very low amplitudes during the initial startup phase; second, the current signals during speed transitions are non-stationary, with time-varying frequencies.

B. Description of OC Fault of Three-Phase Inverter

To better align the test conditions in this study with practical scenarios and improve the feasibility of fault diagnosis algorithms under diverse operating conditions, this work

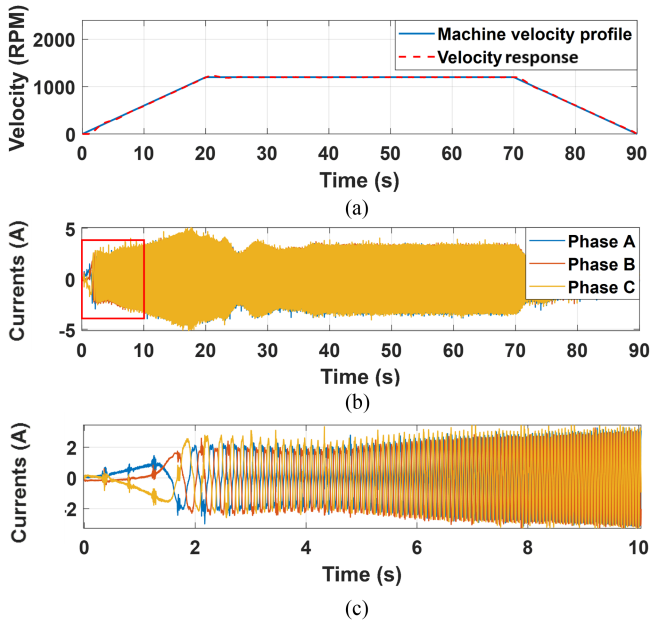

Fig. 2. Velocity and current responses of a full-cycle experimental test in healthy mode. (a) Velocity reference and response at a maximum velocity of 1200 r/min. (b) Current signal of the three-phase inverter. (c) Enlarged view of the current signal during the machine startup phase.

investigates three fault injection modes: nonstationary single faults, mixed stationary dual faults, and stationary single faults, as illustrated in Fig. 3. Fig. 3 illustrates various representative fault scenarios; the subfigures correspond to different experimental tests, rather than a single continuous test that includes multiple faulty scenarios. Fig. 3(a) demonstrates that during each complete experimental test cycle, only one mode among the three fault modes is introduced and persists until the end of the test. Fig. 3(b)–(d) illustrates the different three-phase current responses under three fault injection modes, respectively, where stationary and nonstationary $S1$ OC faults and the mixed case of stationary $S1S2$ OC fault and stationary $S1$ OC fault are taken as examples. These OC fault modes are implemented through hardware, deactivating the PWM signals in the three-phase inverter, as summarized in Table II. The table provides a detailed classification of operating conditions into healthy and fault modes, with the fault modes further subdivided into single-phase and dual-phase inverter faults. Each fault type is systematically labeled to facilitate supervised learning in ML models and to enable performance evaluation of fault diagnosis algorithms. The training and validation datasets were derived from experimental measurements. The original data were resampled to achieve an approximately balanced distribution, as shown in the table, to preliminarily mitigate the potential impact of class imbalance on model performance.

III. Methodology of OC Fault Diagnosis Based on ML

This section introduces the analytical formulation of the sliding window method, its practical implementation, and a real-time OC fault diagnosis methodology for three-phase inverters by integrating the sliding window frame with multiclassification ML models.

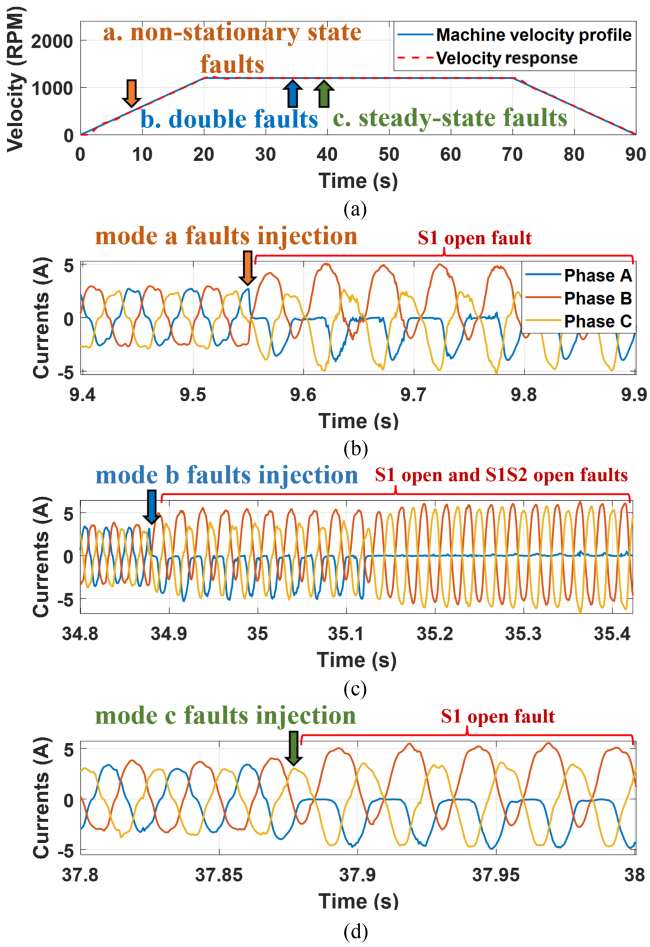

Fig. 3. Three fault scenarios for experimental testing. (a) Machine velocity reference of a complete experimental test. (b) Nonstationary single fault mode. (c) Mixed stationary fault mode. (d) Stationary single fault mode.

A. Sliding Window Method and Implementation

For a signal containing N sampling points, the kth sliding window, denoted as W_k, is illustrated in Fig. 4. The length of W_k is l_k, and the overlap with the adjacent window W_{k+1} is s. A positive s indicates overlapping windows, while a negative s implies a gap of $|s|$ points between adjacent windows, which can be used to resample the signal. In this study, $s = 0$ is chosen for simplicity. With zero overlap, sliding the window by l_{k+1} leads to the next window W_{k+1}. The sliding length l_{k+1} can either be constant or variable. A key innovation of this study is the real-time computation of an ASWIN length l_{k+1} tailored to dynamic current signals. The mathematical description of an arbitrary sliding window W_k is provided as follows:

$$W_k^p(x; l_k, s) = \left\{ x^p[i_k : j_k] \middle| x^p \in x^p[1:N] \right\}, p \in \{a, b, c\} \tag{2a}$$

$$\text{where} \quad i_k = 1 + \sum_{k=1}^{n} l_{k-1} - (k-1) \cdot s, \quad l_0 = 0 \tag{2b}$$

TABLE II
INVERTER OC FAULT TOPOLOGIES AND TRAINING DATA LABELS

Phase	Fault Type	Faulty IGBT(s)	Label	Share (%)
–	Healthy	–	0	6.16
Single	Upper arm OC	S_1	1	5.53
		S_3	3	5.46
		S_5	5	5.73
	Lower arm OC	S_2	2	5.47
		S_4	4	5.18
		S_6	6	5.72
	Upper and lower arms OC	S_1S_2	7	4.00
		S_3S_4	8	4.16
		S_5S_6	9	4.09
Dual	Two upper arms OC	S_1S_3	10	3.97
		S_1S_5	11	4.20
		S_3S_5	12	4.00
	Two lower arms OC	S_2S_4	13	4.00
		S_2S_6	14	3.93
		S_4S_6	15	4.12
	Upper and lower arms OC	S_1S_4	16	4.03
		S_1S_6	17	4.13
		S_3S_2	18	3.96
		S_3S_6	19	4.07
		S_5S_2	20	3.92
		S_5S_4	21	4.18

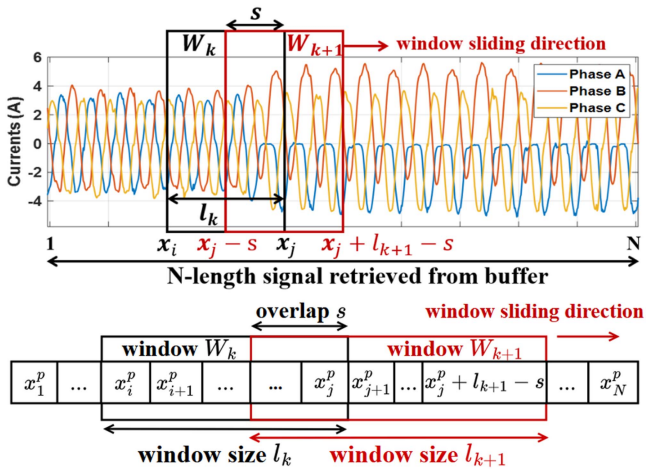

Fig. 4. Analytical representation of the sliding window method.

$$j_k = i_k + l_k - 1 \tag{2c}$$

$$\sum_{k=1}^{n} l_k - (k-1) \cdot s \le N, n = 1, 2, \ldots, K. \tag{2d}$$

Real-time system software interfaces typically incorporate buffer modules to manage data retrieval modes and rates. A fixed-length array of size N is used to store the data retrieved from the buffer to implement the diagnosis algorithm. The sliding window mechanism dynamically discards outdated data

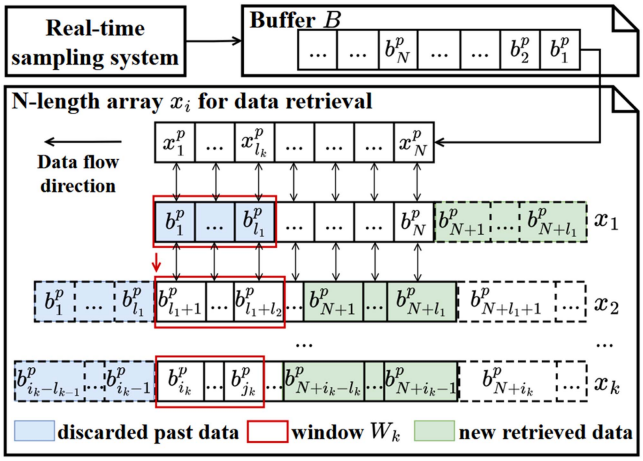

Fig. 5. Implementation of sliding window method in real-time systems.

and retrieves new data, as illustrated in Fig. 5. The blue section represents outdated data discarded after each operation, including fault diagnosis within the current window W_k and the computation of the next optimal window length l_{k+1}. The green section represents newly retrieved data of the same length, maintaining a constant array size of N.

B. Methodology for Fault Diagnosis Using ML

The dynamic sliding window method is integrated with a multiclass ML model for real-time fault diagnosis. The methodology, as illustrated in Fig. 6, includes two phases: offline model training and online fault diagnosis. It also illustrates the correspondence between the main functional modules and the subsequent algorithms, as well as the data flow within specific modules.

1) Offline Training: The offline training process consists of four stages: data acquisition, feature extraction and labeling, data segmentation, and training and validation of the ML model. The first two stages are detailed below:

a) Step 1—Data acquisition and segmentation: Raw three-phase current signals are collected during experiments and splitted into each group using the sliding window method, as shown in Fig. 7. Each sliding window captures local signal characteristics at different operating points, reflecting speed and load torque variations. To enhance model generalization, experimental operating conditions are varied by altering the machine's maximum velocity and torque across multiple tests. Multiple fixed window lengths l^d are utilized during training to generate multilength segments from the raw current signals, thereby enriching the training samples to cover a wider range of situations. This enables the classifier to learn more patterns through multiscale features without relying on any specific data segmentation strategy. The aim is to make the trained model more generalizable with other sliding window strategies, increasing its robustness to feature extraction with variable-length windows.

An adaptive resampling segmentation strategy is employed to mitigate the redundancy caused by similar signal values at nearly identical operating points (e.g., during steady-state operation after reaching maximum speed). This corresponds to the case where s is negative in Fig. 4. The absolute value of s is dynamically adjusted based on speed variation: smaller intervals are used for significant speed changes to capture diverse features. In contrast, larger intervals are applied when changes are minimal. The computation of s is given by

$$\begin{aligned} |s| &= f(\Delta\Omega) \\ &= s_{\max} - \frac{s_{\max} - s_{\min}}{\Omega_{\max}} \cdot \Delta\Omega, \quad 0 \le \Delta\Omega \le \Omega_{\max} \end{aligned} \tag{3}$$

where $f(\Delta\Omega)$ maps speed variation to value $|s|$ within predefined bounds. $s_{\max}$ and $s_{\min}$ are parameters related to resampling requirements, which were empirically selected to reduce redundancy and balance the distribution. In this work, the maximum fixed window size $l_{\max}$ is set to 800 sampling points, $s_{\max}$ is set to $30 \cdot l_{\max}$, and $s_{\min}$ is set to $5 \cdot l_{\max}$

$$\begin{aligned} \Delta\Omega &= \left| \bar{\Omega}(W_k) - \bar{\Omega}(W_{k+1}) \right| \\ &= \left| \frac{1}{l_k} \sum_{i=i_k}^{j_k} \Omega(x_i) - \frac{1}{l_{k+1}} \sum_{i=i_{k+1}}^{j_{k+1}} \Omega(x_i) \right| \end{aligned} \tag{4}$$

where Ω is the speed measured or estimated for each window.

b) Step 2—Feature extraction and labeling: For each resampled signal segment, features are extracted, including time-domain, statistical, and frequency-domain features. The signal features used in this work were preliminarily selected based on empirical and computational time considerations, achieving a desirable tradeoff between diagnostic accuracy, computational cost, and generalization ability, while avoiding overfitting. Based on the proposed framework, it can also be extended to include more signal features. The former two categories and their computations are summarized in Table XII. Frequency-domain features are extracted using the discrete Fourier transform, expressed as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N}, k = 1, \ldots, N-1 \tag{5}$$

where k is the kth frequency component, N is the total number of sample points of the signal, and k represents the index of the frequency component.

The overall process is depicted in Fig. 8, producing a feature vector set $\mathbf{F}$ used for ML model training and validation.

2) Online Diagnosis: During online diagnosis, data from sensors are first buffered in a real-time systems module, then retrieved and stored to arrays of size N for each phase current, as illustrated in Fig. 5. The dynamic sliding window method processes these signals, extracts features, and feeds them into the trained ML model for fault diagnosis. After saving the diagnosis result for the current window W_k, the process advances to the next window W_{k+1}, repeating the procedure. Its primary implementation method is shown in Algorithm 1. The function *diagnosis(·)* diagnoses each signal segment, and the specific signal features extracted by it are shown in Table XII. The function *window_length(·)* is called to calculate the next window

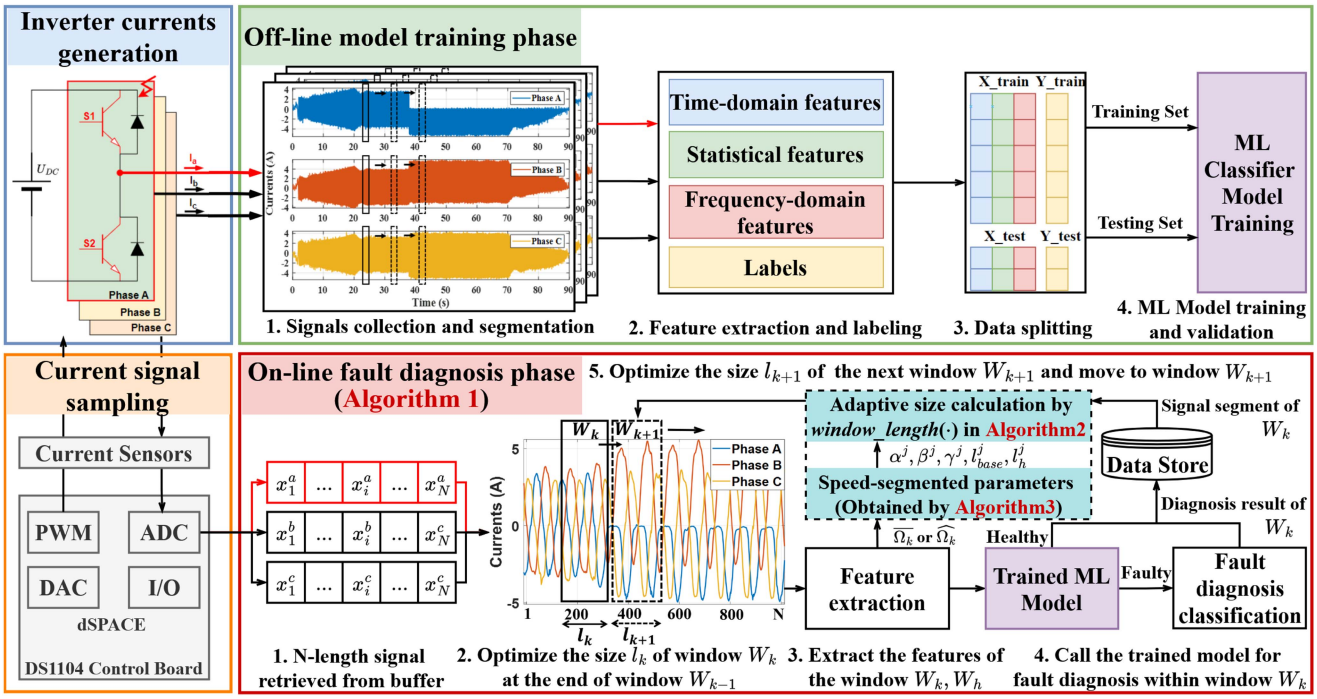

Fig. 6. Methodology for real-time fault diagnosis strategy based on ML model.

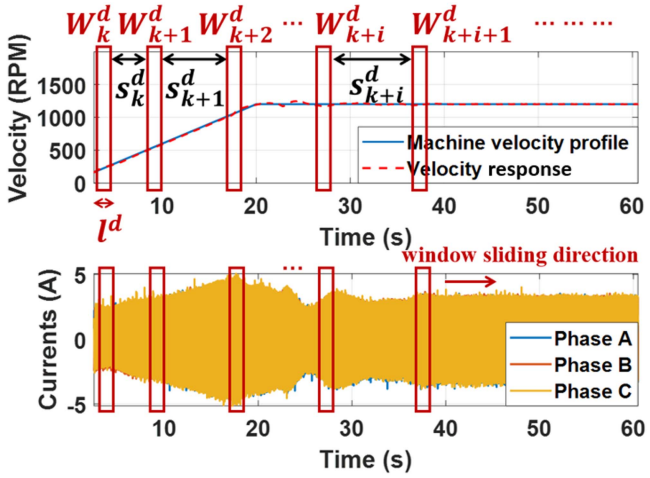

Fig. 7. Adaptive signal segmentation based on speed variation.

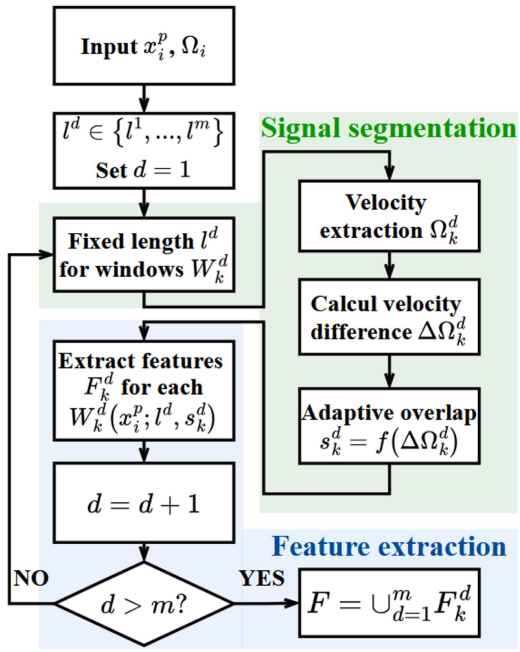

Fig. 8. Flowchart of adaptive segmentation and multiscale feature extraction of raw current signal.

length, which can be a constant function or a variable-length calculation function defined in Algorithm 2.

Unlike the fixed-length sliding window method ($l_k = l_{k+1}$), a variable-length method is proposed in the next section to optimize l_{k+1} based on historical data and short-term predictions at the end of W_k. To improve robustness and minimize misdiagnosis by adapting to current signal variations.

IV. Proposed OC Fault Diagnosis Strategies

A. Range-Based Multiclassification Evaluation

Point-based evaluation methods can be used for a single sampling point or a single set of data with a specific label, which is effective for static scenes. However, they are limited in the dynamic process of sliding windows. Each window W_k may contain multiple sampling points with different labels. The diversity of operating states within a window poses a challenge to capturing the collective characteristics of the entire window. The range-based evaluation strategy estimates the main label as a range-based label by calculating the frequency of all labels

Algorithm 1: Real-Time ML-based Fault Diagnosis.

Input: Retrieved N-length data from buffer $b_i \in \mathcal{B}$
Output: List of tuples (l_k, result_k) for each window
Initialize: Retrieved data length : N
Initial window length : l_{init}
Window counter : $k \longleftarrow 1$
Start and end index for retrieved data x^k:
$index\_start, index\_end \longleftarrow 0$

while *true* **do**
  if $k = 1$ **then**
    $l_1 \longleftarrow l_{init}$
    $x_i^1 \longleftarrow [b_1, \ldots, b_N] \in \mathcal{B}$
  else
    $l_k \longleftarrow$ **window_length**(x_i^{k-1})
    $x_i^k \longleftarrow [b_{index\_start}, \ldots, b_{index\_end}] \in \mathcal{B}$
  end
  $W_k \longleftarrow x^k[1 : l_k]$
  $result_k \longleftarrow$ **diagnosis**$(W_k, diagnosis\_model)$
  $List\_results$.**Append**$((l_k, result_k))$
  $index\_start \longleftarrow 1 + \sum_k l_k$
  $index\_end \longleftarrow N + \sum_k l_k$
  $k \longleftarrow k + 1$
end
Function `diagnosis`$(W_k, diagnosis\_model)$**:**
  $features_k \leftarrow$ **feature_extaction**(W_k)
  $result_k \leftarrow$ **diagnosis_model**$(features_k)$
  return $result_k$
End Function

within the window and applying a multithreshold criterion. This method enhances robustness to noise and ensures that the window-level prediction is consistent with the overall characteristics of the signal within the window.

Formally, for a window W_k, the frequency of each label c is defined as follows:

$$p_k(c) = \frac{\mathbf{count}\left(\mathbf{label}(x_i) = c, i \in [i_k, j_k], c \in \mathcal{C}\right)}{|W_k|} \tag{6}$$

where $\mathcal{C}$ is the set of all possible labels and $|W_k|$ is the number of points in the window.

The following multithreshold criteria determine the estimated window's overall label $\hat{c}_k$. The formula's α and β are preset thresholds to determine the proportion of labels the sampling points represent in the entire window

$$\hat{c}_k = \begin{cases} \arg\max\limits_{c \neq 0} \{p_k(c)\}, & \text{if } p_k(0) > 0.5, \exists c \neq 0,\ p_k(c) > \alpha \\ 0, & \text{if } p_k(0) > 0.5, \forall c \neq 0,\ p_k(c) < \alpha \\ \arg\max\limits_{c \neq 0} \{p_k(c)\}, & \text{if } \exists c_1, c_2 \neq 0,\ p_k(c_1), p_k(c_2) > 0.1 \\ & \quad \text{and } \exists c,\ p_k(c) > \beta \\ \arg\max\limits_{c} \{p_k(c)\}, & \text{otherwise.} \end{cases} \tag{7}$$

The criteria filter out labels with frequencies below a predefined threshold, focusing on the dominant labels within the window to determine the overall range-based label. Given the critical importance of transitions between healthy and faulty states, the multithreshold criteria are refined to emphasize the handling of label 0, representing the healthy state.

After determining the range-based window label $\hat{c}_k$ for each sliding window W_k through the multithreshold criteria, the model's performance can be evaluated by comparing $\hat{c}_k$ with the predicted label $\tilde{c}_k$ produced by the ML algorithm for the same window. Since $\mathcal{C}$, the set of labels, contains multiple classes, this is a multiclass classification problem. Performance evaluation involves calculating standard metrics such as true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) for each class $c \in \mathcal{C}$.

For a given class c, these metrics are defined as follows:

$$\text{TP}(c) = \text{count}\left(\hat{c}_k = c \wedge \tilde{c}_k = c\right) \tag{8a}$$

$$\text{FP}(c) = \text{count}\left(\hat{c}_k \neq c \wedge \tilde{c}_k = c\right) \tag{8b}$$

$$\text{FN}(c) = \text{count}\left(\hat{c}_k = c \wedge \tilde{c}_k \neq c\right) \tag{8c}$$

$$\text{TN}(c) = \text{count}\left(\hat{c}_k \neq c \wedge \tilde{c}_k \neq c\right). \tag{8d}$$

Multiclass metrics, such as weighted accuracy, precision, recall, and F1-score, can be computed using these counts. The weighted accuracy is defined as follows:

$$\text{Accuracy} = \frac{\sum_{c \in \mathcal{C}} w_c \cdot \text{TP}(c)}{\sum_{c \in \mathcal{C}} w_c \cdot (\text{TP}(c) + \text{FN}(c))}. \tag{9}$$

All metrics are weighted to account for the class imbalance among the 22 health and fault categories. The weight w_c of each class c is proportional to its occurrence frequency in the evaluation dataset and is calculated as follows:

$$w_c = \frac{n_c}{\sum_{j \in \mathcal{C}} n_j} \tag{10}$$

where n_c is the number of test samples labeled as class c.

Similarly, weighted precision, recall, and F1-score are given by

$$\text{Precision} = \sum_{c \in \mathcal{C}} w_c \cdot \frac{\text{TP}(c)}{\text{TP}(c) + \text{FP}(c)} \tag{11a}$$

$$\text{Recall} = \sum_{c \in \mathcal{C}} w_c \cdot \frac{\text{TP}(c)}{\text{TP}(c) + \text{FN}(c)} \tag{11b}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{11c}$$

Accuracy measures how many predictions are correct overall. Precision shows how often predicted faults are accurate, while Recall shows how many real faults the model can detect. F1-score combines both Precision and Recall into a single balanced metric. Weighted metrics mitigate class imbalance by proportionally accounting for each class in the evaluation.

To reduce redundancy while retaining the evaluation of effectiveness, we focus only on accuracy and F1-score in the following performance evaluation, which is critical for detecting rare but important fault types in practice.

In addition to estimating the overall range-based label based on the frequency of the sampling point labels within a window, two key metrics, namely, delayed-detection and front-detection,

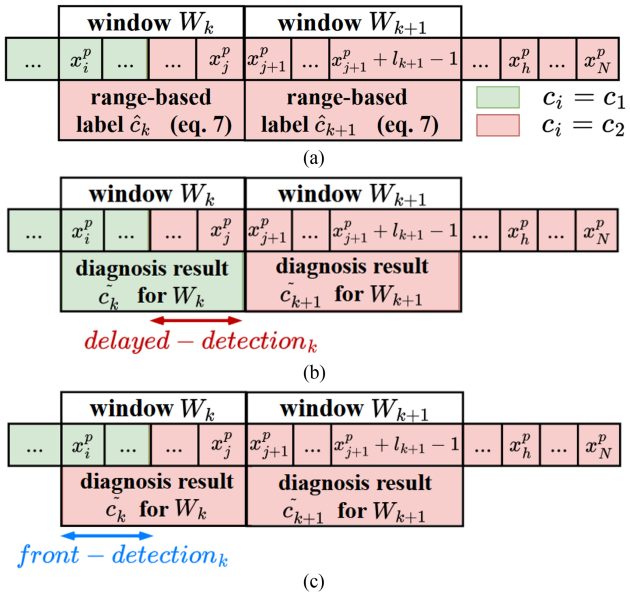

Fig. 9. Range-based window overall label estimation. (a) Estimating range-based labels for windows W_k and W_{k+1}. (b) Delayed-detection for fault diagnosis. (c) Front-detection for fault diagnosis.

are also introduced to measure the time deviation between the occurrence of a fault detected by the algorithm and the actual occurrence of the fault, to comprehensively evaluate the proposed diagnosis strategy with the adaptive sliding window approach, particularly concerning the suitability of window length. An inappropriate window length can lead to detection time biases. For instance, larger windows extract more stable features for diagnosis but may introduce delays.

Fig. 9(a) depicts the estimation of the range-based label. Each sampling point x_i in a sequence of length N has a specific state label c_i. Using a binary classification example (red and green denoting c_1 and c_2), window W_k may cover points with both c_1 and c_2. Based on the defined formulas and criteria, the range-based label $\hat{c}_k$ for W_k is determined as c_2. Fig. 9(b) illustrates the delayed-detection. While the estimated label for W_k is $\hat{c}_k = c_2$, the diagnosis output $\tilde{c}_k$ is c_1. Only in W_{k+1} does the algorithm output $\tilde{c}_{k+1} = c_2$, aligning with reality. Therefore, the time interval between the appearance of the actual fault signal in W_k and the correct fault diagnosis given by W_{k+1} is defined as the delayed-detection. Conversely, Fig. 9(c) illustrates front-detection, which follows a similar principle but represents an earlier-than-actual detection scenario, maintaining a dual relationship with delayed-detection. Front-detection and delayed-detection quantify the sensitivity of the diagnostic model when a sliding window contains a mixed fault transition: front-detection indicates that partial fault samples are sufficient for fault diagnosis. In contrast, delayed-detection reflects the additional delay required when a less sensitive classifier does not diagnose correctly until the next window containing the faulty signal completely.

B. ASWIN Algorithm

The fixed-length sliding window method struggles to balance diagnostic precision and response latency. At low frequencies, a more extended window size is needed to obtain a complete cycle, while at high frequencies, an overly conservative length may result in a delay. However, if the length is adaptively adjusted only according to the signal's frequency, then key factors, such as noise and some transient changes, will be ignored, thereby reducing its robustness.

We proposed an adaptive window scaling algorithm by integrating multiple signal characteristics into a unified scaling factor calculation to address the above limitations in real-time fault diagnosis. By dynamically optimizing length based on historical data and short-term predictions, it can adapt to signal changes while ensuring robustness under different conditions.

1) Generalized Framework: The proposed method generalizes the scaling factor computation to incorporate n signal metrics, each contributing through a normalized mapping. The scaling factor λ_s is defined as follows:

$$\lambda_s = \sum_{i=1}^{n} \omega_i \cdot \mathrm{Norm}(\lambda_i) \tag{12}$$

where λ_i represents the ith metric, ω_i is its corresponding weight, and i is the metric index. $\mathrm{Norm}(\cdot)$ is a normalization function mapping λ_i into the scaling range $[0, 2]$.

In this study, to address three types of practical operating challenges: frequency variation, signal transients, and noise disturbance, three physically interpretable indicators are introduced, dominant frequency ratio (λ_{dfr}), variance (λ_{var}), and signal-to-noise ratio (λ_{snr}), each reflecting a distinct aspect of the signal dynamics. Their definitions are as follows:

$$\lambda_{\mathrm{dfr}} = \frac{|X[f_{\mathrm{df}}]|^2}{\sum_{f=0}^{N-1} |X[f]|^2} \tag{13a}$$

$$\lambda_{\mathrm{var}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2, \quad \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{13b}$$

$$\lambda_{\mathrm{snr}} = \frac{\sum_{i=1}^{N} x_i^2}{\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{13c}$$

where $X[f_{\mathrm{df}}]$ represents the magnitude of the dominant frequency and N is the total number of sampling points.

Specifically, λ_{dfr} quantifies how concentrated the spectral energy is around the dominant frequency; a higher value indicates frequency stability and allows for a shorter window, while a lower value suggests frequency dispersion and thus requires a longer window for sufficient spectral resolution. λ_{var} captures time-domain volatility; larger values correspond to rapid signal fluctuations or transients, motivating shorter windows to improve responsiveness, whereas smaller values imply steady operation suitable for longer windows. Finally, λ_{snr} measures the relative strength of the useful signal against noise; when λ_{snr} is low, a longer window helps suppress noise and stabilize features, whereas a high λ_{snr} enables faster adaptation with a shorter window.

For a given window W, the scaling factor is calculated as follows:

$$\lambda_s^W = \alpha \cdot \mathrm{Norm}(\lambda_{\mathrm{dfr}}^W) + \beta \cdot \mathrm{Norm}(\lambda_{\mathrm{var}}^W) + \gamma \cdot \mathrm{Norm}(\lambda_{\mathrm{snr}}^W) \tag{14}$$

Algorithm 2: Sliding Window Length Optimization.

Input: N-length signal retrieved from buffer $\mathcal{B}$: $x = [x_1, \ldots, x_i, \ldots, x_j, x_{j+1}, \ldots, x_h, \ldots, x_N], N > 1$
Weights for the coefficients : α, β, γ
Window length base for calculation: l_{base}
Length of the horizon window W_h : l_h

Output: The length of the sliding window W_{k+1} : $l_{k+1}, k = 1, 2, 3, \ldots$

Initialize: Length of the initial window W_1 : $l_1 \leftarrow l_{init}$
Weight for current and horizon windows: μ

Function `window_length` $(x, \alpha, \beta, \gamma, l_{base}, l_h)$ **:**
  $W_k \longleftarrow x[i:j]$
  $W_h \longleftarrow x[j+1 : j+l_h]$
  $(\lambda_{dfr}^k, \lambda_{var}^k, \lambda_{snr}^k, f_{df}^k) \longleftarrow$ **coefficients**(W_k)
  $(\lambda_{dfr}^h, \lambda_{var}^h, \lambda_{snr}^h, f_{df}^h) \longleftarrow$ **coefficients**(W_h)
  $\lambda_s^{W_k} \longleftarrow \alpha \cdot \lambda_{dfr}^k + \beta \cdot \lambda_{var}^k + \gamma \cdot \lambda_{snr}^k$
  $\lambda_s^{W_h} \longleftarrow \alpha \cdot \lambda_{dfr}^h + \beta \cdot \lambda_{var}^h + \gamma \cdot \lambda_{snr}^h$
  $\lambda_s^{W_{k+1}} \longleftarrow \mu \cdot \lambda_s^{W_k} + (1-\mu) \cdot \lambda_s^{W_h}$
  $l_{k+1} \longleftarrow \lambda_s^{W_{k+1}} \cdot l_{base}$
  return l_{k+1}
End Function
Function `coefficients` (x) **:**
  $x \longleftarrow x \odot$ **Hamming(n)**
  $X \longleftarrow$ **FFT**(x)
  $f_{df} \longleftarrow \arg\max_f |X(f)|$
  $\lambda_{dfr} \longleftarrow |X[f_{df}]|^2 / \sum_f |X[f]|^2$
  $\lambda_{var} \longleftarrow$ **var**(x)
  $\lambda_{snr} \longleftarrow$ **rms**$(x)^2$/**var**$(x -$ **mean**$(x))$
  normalize $(\lambda_{dfr}, \lambda_{var}, \lambda_{snr})$
  return $(\lambda_{dfr}, \lambda_{var}, \lambda_{snr}, f_{df})$
End Function

where W can either be current instant window W_k or horizon window W_h. α, β, and γ are weights reflecting the importance of each metric.

To incorporate both historical data and short-term predictions into the scaling factor for the $k+1$th window W_{k+1}, the scaling factors for W_k and W_h are weighted and combined as follows:

$$\lambda_s^{W_{k+1}} = \mu \cdot \lambda_s^{W_k} + (1-\mu) \cdot \lambda_s^{W_h} \tag{15}$$

where μ represents the relative weight of the current window W_k compared to the horizon window W_h, trades off current information and short-term prediction.

Subsequently, the scaling factor $\lambda^{W_{k+1}}$ is used to compute the length l_{k+1} of the window W_{k+1}, by multiplying it with the baseline window length l_{base}

$$l_{k+1} = \lambda_s^{W_{k+1}} \cdot l_{\text{base}}. \tag{16}$$

The pseudocode for this part is shown in Algorithm 2. It takes the current signal data retrieved from buffer $\mathcal{B}$ and stored in an array of length N as input, sets the parameters, and the final output is the length l_{k+1} of window W_{k+1}. The parameter combination $(\alpha, \beta, \gamma, l_{\text{base}}, l_h)$ required by *diagnosis(·)* can be the global optimal solution or the speed-based segmented optimal solution obtained by Algorithm 3.

2) Optimal Parameters Sliding Window (OPSWIN) Algorithm: A mathematical optimization problem is formulated to determine the globally optimal parameter set that maximizes diagnostic accuracy. This approach aims to optimize five key parameters: the feature weights α, β, and γ, the sliding window base length l_{base}, and the horizon window length l_h.

The optimization is conducted offline using a dataset of experimentally collected current signals. The fault diagnosis algorithm processes all signals for each parameter combination, and the diagnostic results are compared to the ground truth to compute accuracy. The parameter set that maximizes the accuracy is identified as the optimal solution.

The optimization problem is formulated as follows:

$$\max_{\alpha,\beta,\gamma,l_{\text{base}},l_h} \quad \text{accuracy}\left(\sum_{i=1}^{f}\sum_{k=1}^{m} x_k^i; \alpha, \beta, \gamma, l_{\text{base}}, l_h\right) \tag{17a}$$

$$\text{s.t.} \quad \alpha + \beta + \gamma = 1 \tag{17b}$$

$$\alpha \geq 0,\ \beta \geq 0,\ \gamma \geq 0 \tag{17c}$$

$$l_{\text{base\_min}} \leq l_{\text{base}} \leq l_{\text{base\_max}} \tag{17d}$$

$$l_{h\_\text{min}} \leq l_h \leq l_{h\_\text{max}} \tag{17e}$$

where the variable x_k^i represents the experimentally collected current signals, i represents the label of different operating modes, and k represents the sequence number of the collection experiment. There are f different operating modes, and for each operating mode, m experimental results are collected. This work features 22 operating modes, and their labels are listed in Table II. The constraint $\alpha + \beta + \gamma = 1$ ensures the normalization of the feature weights. α, β, and γ must be nonnegative to ensure valid weight assignments. The values of l_{base} and l_h are restricted within the predefined minimum and maximum limits, denoted as $l_{\text{base\_min}}, l_{\text{base\_max}}, l_{h\_\text{min}}$, and l_{h_max}, as operational constraints.

3) ASWIN Algorithm: We analyzed current signals from healthy motor states to improve the algorithm's robustness and applicability under real-world motor speed variations across the entire speed range. Using sliding windows of varying sizes, statistical analyses of local signal metrics were performed for each window. As illustrated in Fig. 10, the statistical distributions of the three metrics reveal distinct patterns across different speed ranges: in the low-speed range (0–400 r/min), the metrics remain relatively stable with minimal fluctuations; in the high-speed range (800–1200 r/min), the metrics show moderate increases in variability, but reflecting their consistent dynamics; and in the medium-speed range (400–800 r/min), the metrics exhibit the most significant variations, indicating heightened sensitivity to operating conditions. These findings suggest that hardware dynamics influence the relative magnitudes of these metrics across speed ranges. As a result, the contribution of each metric to the scaling factor should be adjusted to reflect the varying dynamics of each speed stage. This implies that a globally optimized set of five parameters may be less effective than locally optimized parameter combinations tailored to distinct speed ranges, enhancing the algorithm's performance.

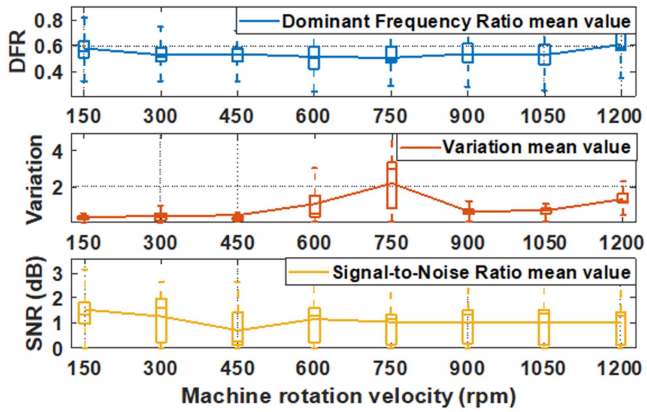

Fig. 10. Statistics of the changes in the values of the three signal metrics at various machine velocities.

The global optimization problem was redefined as a segmented optimization framework to overcome these challenges. The full speed range of the motor can be divided into several ranges $[\Omega_{j-1}, \Omega_j]$ using a predefined speed threshold Ω_j, where the subscript j represents the speed threshold index. An optimal parameter set was identified for each range. This approach can be mathematically expressed as follows:

$$\max_{\{\alpha^j,\beta^j,\gamma^j,l_{\text{base}}^j,l_h^j\}_{j=1}^t} \sum_{j=1}^{t} \text{accuracy}\left(\sum_{i=1}^{f}\sum_{k=1}^{m} x_j^{i,k}; \alpha,\beta,\gamma,l_{\text{base}},l_h\right) \tag{18a}$$

$$\text{s.t.} \quad x_j^{i,k} = \{x \mid \Omega_{j-1} < \Omega(x) \le \Omega_j\}, \quad j = 1,\ldots,t \tag{18b}$$

$$\alpha+\beta+\gamma = 1 \tag{18c}$$

$$\alpha \ge 0,\ \beta \ge 0,\ \gamma \ge 0 \tag{18d}$$

$$l_{\text{base\_min}} \le l_{\text{base}} \le l_{\text{base\_max}} \tag{18e}$$

$$l_{h\_\text{min}} \le l_h \le l_{h\_\text{max}} \tag{18f}$$

where $x_j^{i,k}$ represents the signal segment within the jth speed range in the kth experimental data of the ith operating mode.

4) Implementation of the Two Optimization Methods:

a) Global optimization for OPSWIN algorithm: The MATLAB built-in genetic algorithm (GA) is used for global optimization, searching the parameter space for the best combination that maximizes diagnostic accuracy. This method is particularly well-suited for nonlinear optimization involving multiple parameter combinations, as it efficiently searches multiple solution spaces to avoid local optima. However, GA entails high computational costs, especially for problems involving various parameters. Based on testing with experimental data, the sliding window base length l_{base} and the horizon window length l_h were found to perform ideally within the range of 200–800.

b) Segmented Optimization for ASWIN Algorithm: Given the segmentation of speed ranges, applying GA to each segment would result in prohibitively long computation times. Instead, an exhaustive search was used to determine the optimal parameters for each speed stage. MATLAB's multicore parallel processing capability was leveraged to enhance computational efficiency by replacing traditional FOR loops with the PARFOR function. The traversal range and step size can be set based on the requirements for accuracy. In this study, $l_{\text{base}}, l_h \in [200:100:800]$, $\alpha,\beta,\gamma \in [0:0.05:1]$. For speed segmentation, the range of 0–1200 r/min was divided into three stages using thresholds at 400 and 800 r/min. The pseudocode for this process is shown in Algorithm 3.

Algorithm 3: Segmented Optimal Parameters Search.

Input: Various faulty signal files: $\mathcal{F} = \{f_1,\ldots,f_n\}$
Fault types: $label_i = \textbf{label}(x_i) \in \{0,\ldots,t\}$
Output: Optimal parameters for each speed range j : $\alpha^{j*},\beta^{j*},\gamma^{j*},l_{base}^{j*},l_h^{j*}$
Initialize: Threshold of speed range $\Omega_{th}^j, j = 0,1,\ldots$

foreach $f_i \in \mathcal{F} = \{f_1,f_2,\ldots,f_n\}$ **do**
  Load current signal x_i from file f_i
end
foreach $(\alpha,\beta,\gamma,l_{base},l_h) \in [0,1]\times[0,1]\times[0,1]\times[200,800]\times[200,800]$ **do**
  if $a+b+c=1$ **then**
    foreach $x_i \in \{x_1,\ldots,x_n\}$ **do**
      $\{(l_1,f_{df}^1),\ldots,(l_k,f_{df}^k)\} \longleftarrow$ **window_length**$(x_i,\alpha,\beta,\gamma,l_{\text{base}},l_h)$
      $\{W_1,\ldots,W_k\} \longleftarrow \{x_i[1:l_1],\ldots,$
      $x_i[1+\sum_k l_{k-1} : l_k+\sum_k l_{k-1}]\}$
      foreach $W_k \in \{W_1,\ldots,W_k\}$ **do**
        $\Omega_k \longleftarrow f_{df}^k \cdot 30$ **or** $\overline{\Omega_k}$
        $result_k \longleftarrow$ **diagnosis**(W_k)
        $label_k \longleftarrow$ **label**(W_k)
        if $\Omega_k \in [0,\Omega_{th}^1]$ **then**
          $counter^1 \longleftarrow counter^1+1$
          if $result_k = label_k$ **then**
            $true\_pos^1 \longleftarrow true\_pos^1+1$
          end
        . . . . . .
        else if $\Omega_k \in (\Omega_{th}^{j-1},\Omega_{th}^j]$ **then**
          $counter^j \longleftarrow counter^j+1$
          if $result_k = label_k$ **then**
            $true\_pos^j \longleftarrow true\_pos^j+1$
          end
        end
      end
    end
    $accuracy^j = true\_pos^j/counter^j$
  end
  $List\_results$.**Append**$(\alpha,\beta,\gamma,l_{\text{base}},l_h,accuracy^j)$
end
$(\alpha^{j^*},\beta^{j^*},\gamma^{j^*},l_{base}^{j^*},l_h^{j^*}) \leftarrow \underset{\alpha^j,\beta^j,\gamma^j,l_{base}^j,l_h^j}{\textbf{arg max}} (accuracy^j)$
return $\alpha^{j^*},\beta^{j^*},\gamma^{j^*},l_{base}^{j^*},l_h^{j^*}$

The above two optimization solutions were performed using MATLAB R2020b on the PC equipped with an Intel Core i7-8550U and 16 GB RAM. Table III presents the two methods' final results. Repeated optimization runs produced slightly different parameters due to solver settings. However, the resulting diagnostic accuracy, F1-score, and latency remained nearly

TABLE III
RESULTS OF TWO OPTIMIZATION METHODS

Algorithm	Ω (r/min)	α^*	β^*	γ^*	l^*_{base}	l^*_h
OPSWIN	[0 , 1200]	0.2042	0.1591	0.6367	353	244
ASWIN	[0 , 400)	0.15	0.20	0.65	700	400
	[400 , 800)	0	0.10	0.90	500	300
	[800 , 1200]	0.45	0.40	0.15	500	200

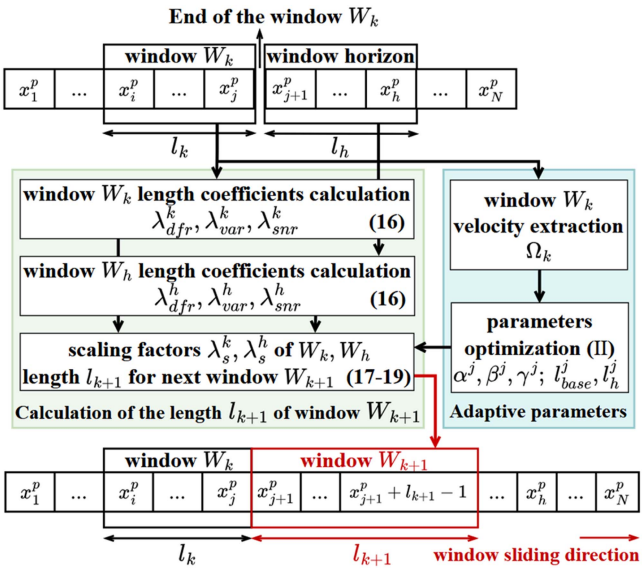

Fig. 11. General methodology of ASWIN strategy.

identical; hence, the proposed parameter optimization-based framework is robust and reproducible.

After determining the optimal parameter configuration for each speed segment through offline optimization, the real-time speed can be used to self-tune the parameter combination to better adapt it to the dynamics of the current signal. The overall method is shown in Fig. 11, where the adaptive tuning of the optimal parameters is based on the speed information.

Real-time speed can be obtained via two approaches. The first method is to directly read the speed data Ω_i of the encoder and calculate the average speed $\overline{\Omega_k}$ in each window W_k, which is accurate and fast. The second method is to estimate the mechanical speed $\hat{\Omega}_k$ by extracting the dominant frequency f^k_{df} of the current signal within the sliding window W_k, which corresponds to the electrical frequency of the motor, and using the number of pole pairs. Although this method is more complex, it offers greater scalability, particularly for sensorless control strategies. Experimental results indicate that both methods yield comparable performance in practice.

V. EXPERIMENTAL VALIDATION OF THE PROPOSED DIAGNOSTIC STRATEGY

This section introduces the validation and analysis of the proposed diagnostic strategies using experimental data collected from a WRSM drive system real-time test bench.

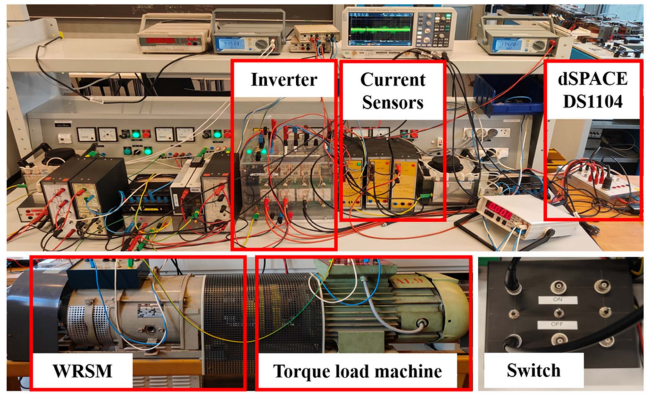

Fig. 12. Test bench for WRSM drive system.

A. Experimental Test Platform and Data Acquisition

The test bench utilized to validate the proposed method is shown in Fig. 12. The main components include an ARCEL three-phase inverter, a dSPACE DS1104 Controller Board, a WRSM (ECODIME TA-100L4, rated speed at 1500 r/min), a dc motor providing torque load, and the switch applied to simulate OC faults in the inverter IGBTs.

The test bench achieves real-time interaction with a host PC through the dSPACE DS1104 Controller Board, which connects to MATLAB/Simulink using the real-time interface module. The ControlDesk was also employed to interact with Simulink and to facilitate real-time data visualization and acquisition. During operation, the PWM control signals for the inverter are represented as $[g_1, g_2, \ldots, g_6]$, where $g_i = 0$ or 1. To simulate one or multiple IGBT OC faults, switches in the corresponding circuits, as shown in Fig. 12, are set to the open state for the duration of the fault. The current sampling frequency f_s of the three-phase inverter is 2 kHz.

To illustrate the data acquisition and labeling procedure, Fig. 13 provides a representative experiment conducted at a maximum reference speed of 1200 r/min with injected $S1S2$ OC fault under torque disturbances. The maximum reference speed can be adjusted; it is also set to 400 r/min to obtain more current signals under low speed conditions. Fig. 13(a) shows the overall experimental timeline: the blue and black curves represent the reference and measured speeds, respectively; the colored bar at the top indicates the time instants of injected faults; and the red curve denotes the estimated load torque, where step changes correspond to applied disturbances. Fig. 13(b) details the manual labeling process, where each sampled point is annotated by its operating state, including speed level, stationarity, and torque-disturbance status. Fig. 13(c) displays the three-phase current signals for the entire experiment, with the annotated operating states overlaid. These labeled signals are directly used for model training and validation. Finally, Fig. 13(d) magnifies the current waveforms around $t \approx 67.15$ s, where the transition from an $S1$ OC fault to an $S1S2$ OC fault, highlighting the abrupt change in the current response at the fault-injection instant.

Following the rule of thumb in [31], the class imbalance rate measures the used dataset: the original dataset exhibits

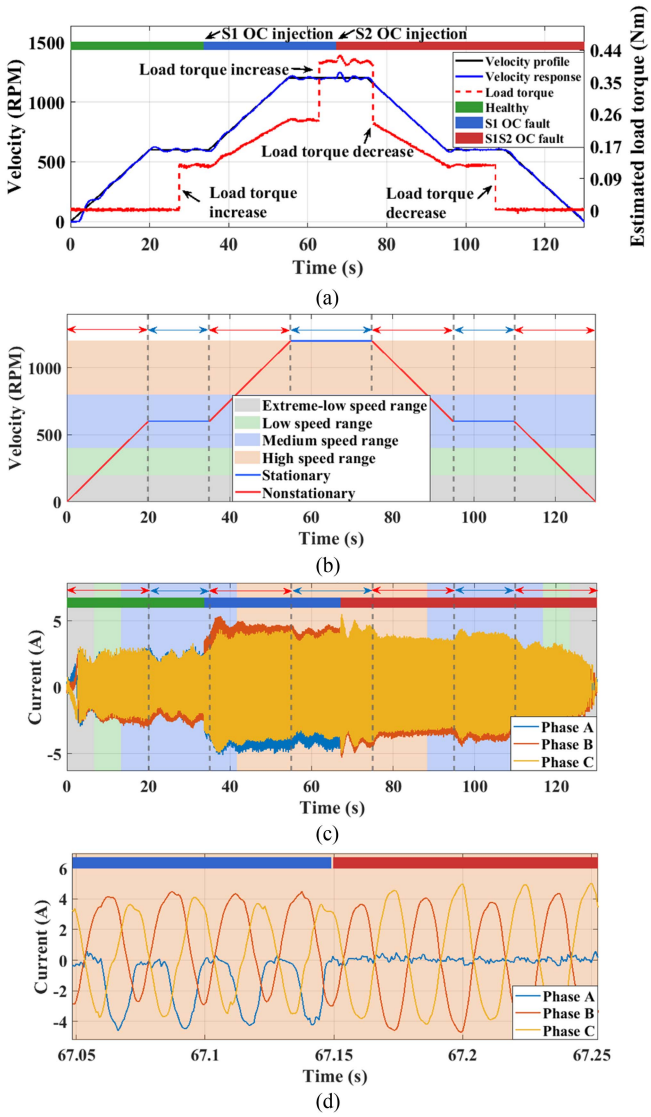

Fig. 13. Representative experiment with S1S2 OC fault injection and data labeling. (a) Experimental timeline of speed, torque, and fault injection. (b) Manual labeling of operating states. (c) Three-phase current signals with annotated regimes. (d) Zoom in around $t \approx 67.15$ showing the fault transition from $S1$ OC fault to $S1S2$ OC fault.

moderate imbalance, while the resampled dataset is close to balanced. The distribution is shown in Table II. However, a weighted evaluation is always used to ensure that the error of the 22-category classification evaluation is minimized.

B. ML Model Training and k-Fold Validation

The Methodology described in Section IV was followed to train and validate the ML models, using the three-phase current signals in healthy and faulty states collected from the above experimental test bench. At this stage, fixed multiscale sliding windows were applied to traverse and resample all current signals. Feature extraction was performed on each segment to create training data and enhance the model's generalization ability across various operating conditions. Table IV lists the main parameter settings of the six diagnostic models implemented in MATLAB to ensure the reproducibility of the training phase.

TABLE IV
COMMON PARAMETER SETTINGS FOR DIAGNOSTIC MODEL TRAINING USING MATLAB

Model(Matlab Function)	Parameter Settings
SVM (fitcecoc)	KernelFunction: Linear; BoxConstraint: 1; Standardize: true; Coding: One-vs-One
DT (fitctree)	MaxNumSplits: 50; MinLeafSize: 1; SplitCriterion: Gini; Prune: on
RF (TreeBagger)	NumTrees: 100; MinLeafSize: 1; Bootstrap: true; NumLearningCycles: 50
AdaB (fitcensemble)	Learner: DT ; MaxNumSplits: 50; Method: AdaBoostM2; LearningRate: 0.1; NumLearningCycles: 100
LR (fitcecoc)	Learner: logistic; Regularization: lasso; Solver: LBFGS; Lambda: 0.01
KNN (fitcknn)	NumNeighbors: 5; Distance: euclidean; DistanceWeight: equal; Standardize: true

TABLE V
PERFORMANCE EVALUATION OF SIX CLASSIFICATION MODELS USING ORIGINAL AND RESAMPLED DATASETS

	Dataset Type	Accuracy (%)	F1-score (%)	Training Time (s)	Inference Time (s)	Model Size (KB)
SVM	Original	96.58	95.93	831.2	0.0357	61837
	Resampled	96.30	95.58	175.8	0.0329	20908
DT	Original	96.18	94.87	2.4	0.0009	61645
	Resampled	95.78	94.50	1.1	0.0009	20706
RF	Original	97.80	97.75	37.6	0.0828	140073
	Resampled	97.48	97.36	13.6	0.0526	48986
AdaB	Original	97.94	97.62	249.5	0.1121	159964
	Resampled	97.37	97.22	84.0	0.0988	56933
LR	Original	95.89	94.95	46.5	0.0376	484
	Resampled	95.22	94.10	26.2	0.0369	483
KNN	Original	96.18	94.92	0.9	0.0207	52819
	Resampled	95.31	93.62	0.4	0.0108	17711

To evaluate and compare the diagnostic performance of different ML models and to verify whether the proposed method improves their performance, six ML models were tested: SVM, decision tree (DT), random forest (RF), AdaBoost (AdaB), logistic regression (LR), and KNN. The performance metrics are summarized in Table V. The calculation time refers to the time it takes to call the model once to calculate a validation dataset. Here, it is the average value of traversing all the validation data results.

Overall, resampling reduces redundancy in the training set and leads to a very minor performance loss: the average accuracy and F1-score decrease by less than 1%. In contrast, the computational benefits are substantial: the mean training time is reduced by approximately 60%, and the average model size shrinks by more than 50% across all classifiers. Inference time remains nearly unchanged, with differences typically below a few milliseconds. For deployment, models like LR, KNN, and DT offer favorable tradeoffs between accuracy, efficiency, and resource requirements. In contrast, ensemble models like RF and

TABLE VI
FAULT DIAGNOSIS PERFORMANCE EVALUATION UNDER FIXED-LENGTH SLIDING WINDOW STRATEGIES OF DIFFERENT LENGTHS

Model	Window Size	Accuracy (%)	F1-score (%)	Stationary Acc.(%)	Nonstationary Acc.(%)	Extreme-low-speed Acc.(%)	Low-speed Acc.(%)	Torque-disturbance Acc.(%)	Front-detection (s)	Delayed-detection (s)	Inference Time(s)
SVM	200	94.90	96.01	98.46	92.67	81.29	98.79	100.00	0.0116	0.0208	0.0340
	400	95.62	96.79	98.49	93.83	84.71	98.89	100.00	0.0231	0.0437	0.0339
	600	95.98	97.15	98.50	94.41	86.41	98.93	100.00	0.0227	0.0487	0.0341
	800	96.18	97.31	98.48	94.73	87.36	98.94	100.00	0.0257	0.0686	0.0342
DT	200	92.02	94.67	96.54	89.18	74.24	95.76	97.53	0.0223	0.0192	0.0017
	400	93.17	95.79	97.07	90.72	77.76	96.61	100.00	0.0262	0.0206	0.0016
	600	93.67	96.25	97.25	91.42	79.43	96.95	97.96	0.0258	0.0241	0.0016
	800	93.95	96.48	97.32	91.84	80.42	97.12	98.33	0.0377	0.0312	0.0016
RF	200	96.13	97.21	98.69	94.52	86.76	98.78	100.00	0.0137	0.0184	0.0551
	400	96.74	97.84	98.75	95.47	89.26	98.97	100.00	0.0327	0.0290	0.0560
	600	97.03	98.11	98.80	95.92	90.42	99.03	100.00	0.0287	0.0340	0.0560
	800	97.19	98.21	98.79	96.19	91.11	99.07	100.00	0.0400	0.0484	0.0563
AdaB	200	95.84	97.14	98.52	94.15	85.52	98.81	100.00	0.0133	0.0188	0.1115
	400	96.54	97.81	98.61	95.24	88.48	98.94	100.00	0.0252	0.0329	0.1064
	600	96.83	98.05	98.66	95.68	89.65	99.01	100.00	0.0251	0.0360	0.0991
	800	96.96	98.10	98.65	95.90	90.24	99.03	100.00	0.0253	0.0559	0.0954
LR	200	93.91	95.18	98.22	91.21	77.25	98.33	100.00	0.0136	0.0183	0.0386
	400	94.79	96.19	98.26	92.61	81.30	98.50	100.00	0.0214	0.0452	0.0393
	600	95.19	96.64	98.29	93.25	83.16	98.58	100.00	0.0172	0.0499	0.0393
	800	95.41	96.87	98.27	93.63	84.26	98.59	100.00	0.0189	0.0706	0.0392
KNN	200	91.68	94.31	96.93	88.39	72.89	95.05	100.00	0.0191	0.0141	0.0118
	400	92.69	95.45	97.24	89.83	76.17	95.78	100.00	0.0401	0.0206	0.0118
	600	93.19	95.99	97.34	90.59	77.86	96.17	100.00	0.0411	0.0227	0.0118
	800	93.52	96.33	97.39	91.11	79.00	96.39	100.00	0.0484	0.0264	0.0119

AdaB provide superior accuracy but at higher computational and memory costs. Since static verification uses preprepared data, Table V reports the inference time excluding feature extraction.

These results indicate that the resampling strategy preserves almost all diagnostic accuracy while significantly reducing training burden and storage requirements, thereby improving the practical feasibility of the proposed framework for real-time deployment. Therefore, models trained based on resampled data were selected in the subsequent diagnostic strategies combined with the ASWIN method.

C. Dynamic Validation of Proposed Diagnostic Strategies

The k-fold validation described above only represents a static evaluation of segmented validation datasets. Next, the real-time current signals over the entire test cycle, as shown in Fig. 3, were analyzed to dynamically validate the proposed diagnostic strategy. Both fixed and ASWIN methods were applied to traverse the current signals, with each window's local signals diagnosed individually. The overall diagnostic performance was evaluated by comparing the results with the ground truth across all windows.

The current signals were collected under multiple operating conditions for the validation phase: maximum machine speeds $\Omega_{\max} \in [200, 800, 1200]$ r/min and load torques varied between 0.33 and 2.16 N · m. Fault scenarios were injected based on the three modes shown in Fig. 3 and detailed in Table II, covering a range of fault types. Performance metrics included the above-mentioned metrics, the average computation time per diagnosis, and delay detection and early detection times, as introduced in Section IV.

1) Performance of Fixed-Length Sliding Window Method: The diagnostic performance using fixed-length sliding window lengths was first evaluated. Window lengths of 200, 400, 600, and 800 sample points were tested across all six ML models to explore the tradeoff between diagnostic accuracy and time delay. Results are given in Table VI. Longer sliding windows generally improved accuracy but introduced a more significant detection delay, particularly for delay detection times, which are critical for practical applications. Stationary regimes are generally well diagnosed ($>97\%$), while ultra-low-speed operation remains the most challenging, with accuracy reduced to 72%–87% depending on model and window size. Extreme low-speed conditions impose significant difficulty due to weak current amplitude and noise. Nevertheless, even under such conditions, ensemble methods such as RF and AdaB still sustain above 85%–90% accuracy, demonstrating fault detection's feasibility despite adverse signal quality. Noted that the inference time here includes the feature extraction module compared to the static verification results in Table V, the difference between

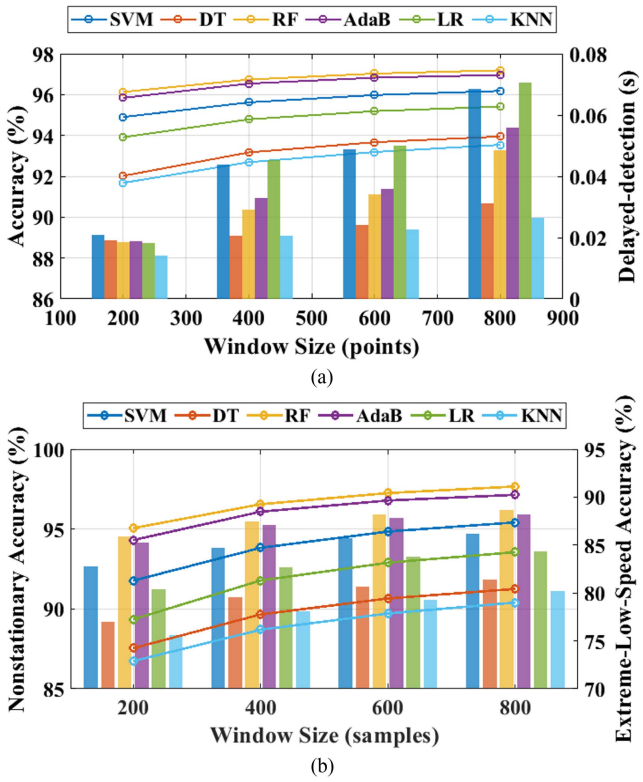

Fig. 14. Comparison of accuracy and delayed detection of six ML models at various fixed window lengths. (a) Line chart represents global accuracy, and the bar chart represents delayed-detection time. (b) Line graph shows the accuracy in a nonstationary state, while the bar graph shows the accuracy at extremely low speeds.

the two is the approximate time required for feature extraction. Notably, the accuracy gain between lengths of 600 and 800 was minimal, while detection delays continued to increase. This observation may support the hypothesis proposed in Section IV that a dynamic adjustment of sliding windows could achieve high accuracy with reduced delays.

Fig. 14 visually illustrates the tradeoff between accuracy and latency. The line graph in Fig. 14(a) shows global accuracy on the left *y*-axis, while the bar graph shows delayed-detection time on the right *y*-axis. The line graph in Fig. 14(b) shows accuracy under nonstationary conditions on the left *y*-axis, while the bar graph shows accuracy at extreme-low speeds on the right *y*-axis.

2) Feasibility Verification of the Proposed Strategies: The proposed diagnostic strategies were validated with seven ML models, including the OPSWIN and ASWIN algorithms. The following examples show the experimental results. The validation is performed under three fault injection modes: nonstationary single S_1 OC fault, stationary single S_1 OC fault, and the mixed case of S_1 OC stationary fault and S_1S_2 OC stationary fault. Once a fault occurs, it will persist until the end of the entire test cycle.

a) Nonstationary single S_1 OC Fault: Fig. 15(a) illustrates the three-phase current signals over the entire test cycle, with colored bars indicating diagnostic results. The upper one is the diagnosis result based on the OPSWIN strategy, and the lower one is based on the ASWIN strategy. Green represents healthy states, red denotes S_1 OC faults, yellow represents S_2 OC faults, and purple indicates S_1S_3 OC faults. Fig. 15(b)–(d) is the enlarged views of the three red-framed parts in Fig. 15(a), observing the signal conditions at the moment of fault occurrence and the two extreme-low-speed motor start and stop stages. The red dotted line in Fig. 15(b) represents the actual moment of fault occurrence, the red solid line represents the moment diagnosed by the algorithm, and the red arrow represents the delayed detection. It shows that at the stage of fault occurrence, both algorithms can achieve accurate fault diagnosis within a delay of 0.1 s, and the delay of the ASWIN strategy is less. Fig. 15(c) and (d) shows that at the two highly low-speed stages, the misdiagnosis probability of the ASWIN strategy is lower than that of the OPSWIN strategy.

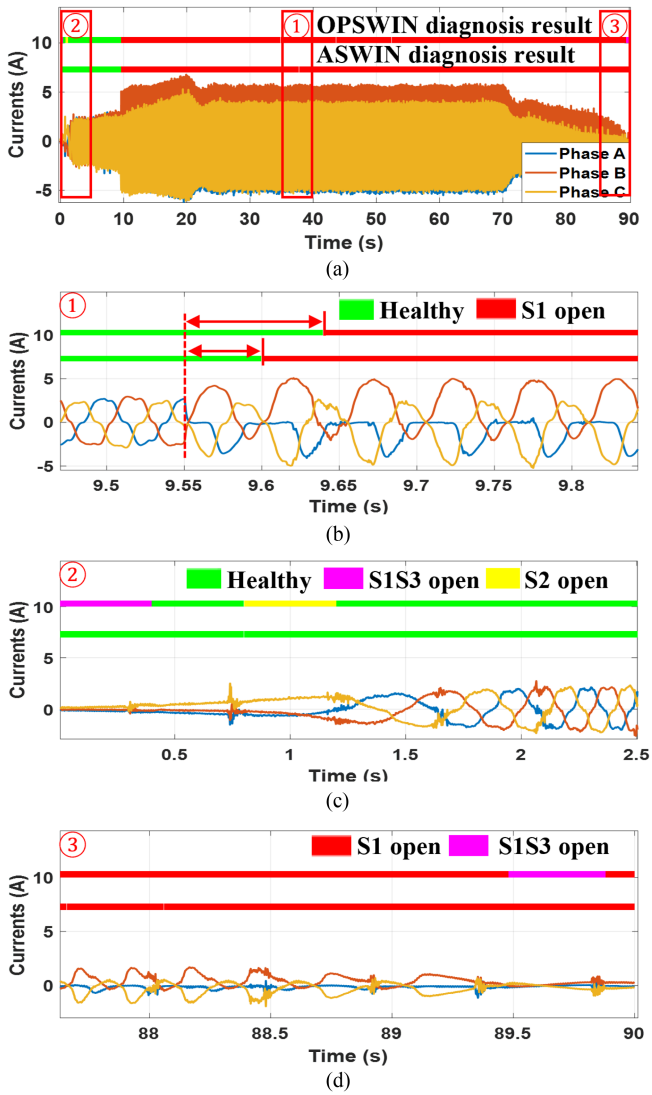

Fig. 15. Experimental data diagnosis results of nonstationary S_1 OC fault. (a) Current signal of a complete fault state test cycle. (b) Enlarged views of the current signal at the occurrence of the fault. (c) Enlarged views of the current signal during the machine startup phase. (d) Enlarged view of the current signal as the machine approaches a stop.

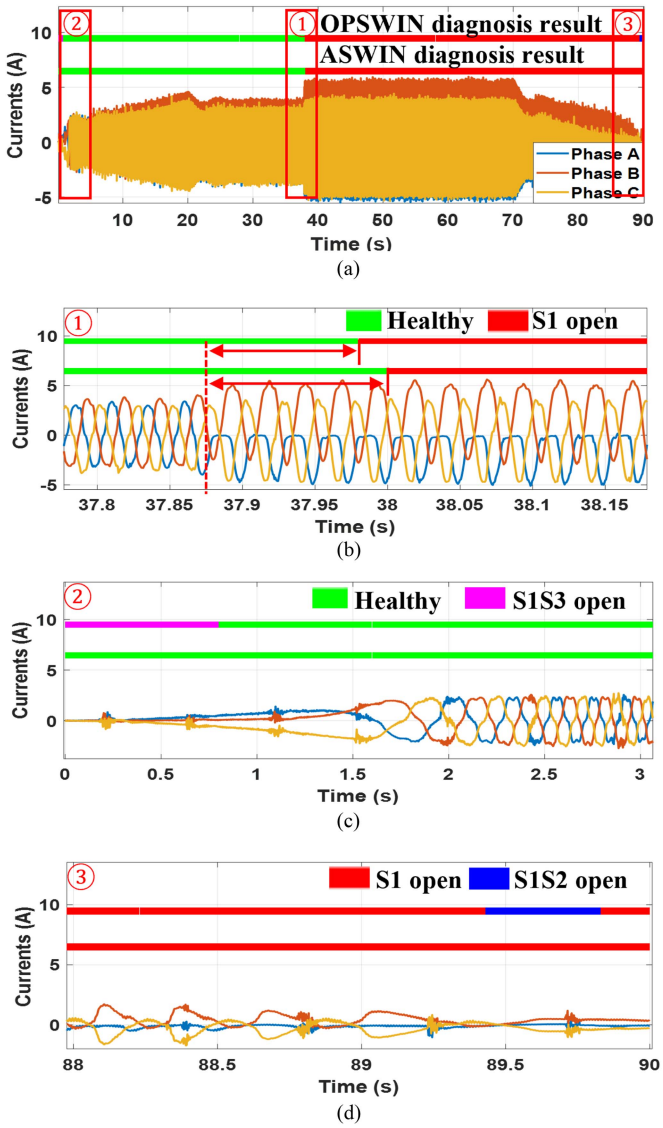

Fig. 16. Experimental data diagnosis results of stationary S_1 OC fault. (a) Current signal of a complete fault state test cycle. (b) Enlarged views of the current signal at the occurrence of the fault. (c) Enlarged views of the current signal during the machine startup phase. (d) Enlarged view of the current signal as the machine approaches a stop.

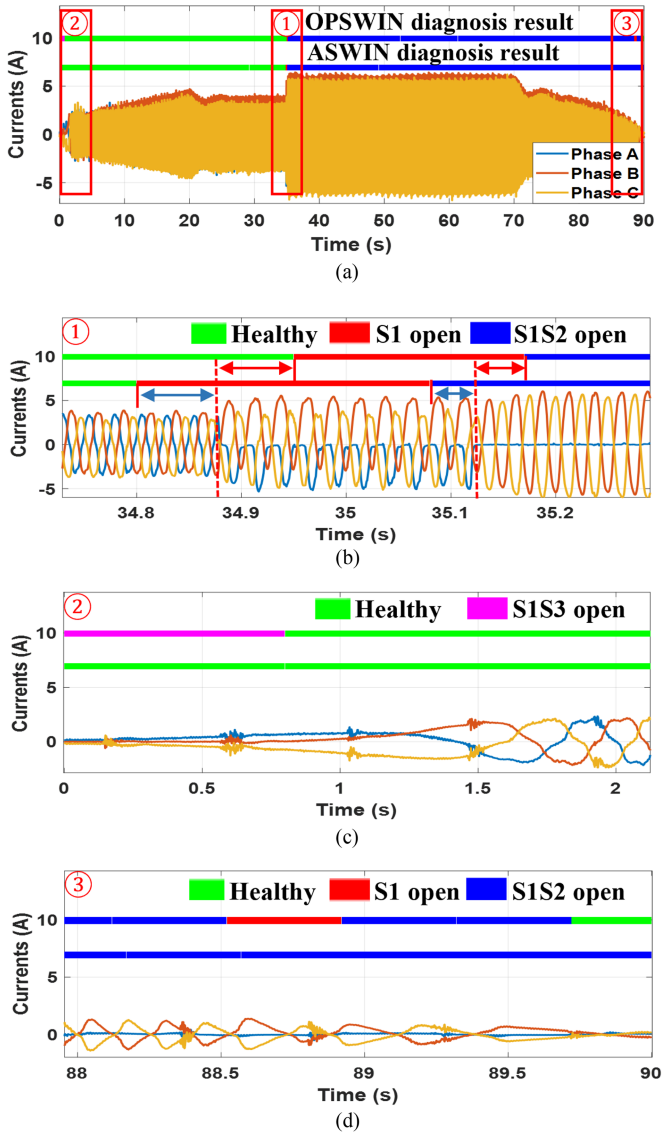

Fig. 17. Experimental data diagnosis results of a mixed case of a stationary S_1 OC fault and a stationary S_1S_2 OC fault. (a) Current signal of a complete fault state test cycle. (b) Enlarged views of the current signal at the occurrences of two faults. (c) Enlarged views of the current signal during the machine startup. (d) Enlarged view of the current signal as the machine nears the stop.

b) Stationary single S_1 OC Fault: Fig. 16(a) shows the three-phase current response signal of the entire test cycle. In this figure, green represents the healthy mode, red represents the $S1$ OC fault, blue represents the $S1S2$ OC fault, and purple represents the $S1S3$ OC fault. Fig. 16(b)–(d) are the enlargements of the three red box parts in Fig. 16(a). Fig. 16(b) shows that in the stage where the fault occurs, both algorithms can achieve accurate fault diagnosis within a delay of 0.1 s, but the delay of the ASWIN strategy in this test is longer. Fig. 16(c) and (d) shows that in the two highly low-speed stages, the misdiagnosis probability of the ASWIN strategy is still lower than that of the OPSWIN strategy.

c) Mixed case of stationary S_1 OC fault and stationary S_1S_2 OC Fault: Fig. 17(a) shows the three-phase current response signal of the entire test cycle of the mixed case of two stationary faults. In this figure, green represents the healthy mode, red represents the $S1$ OC fault, blue represents the $S1S2$ OC fault, and purple represents the $S1S3$ OC fault. The blue arrow represents the front detection. Fig. 17(b) shows the stages of the occurrence of the two faults, which shows that in the stages of the two faults, both algorithms can achieve accurate fault diagnosis within a delay of 0.1 s. The ASWIN strategy has two front detections, while OPSWIN still has delayed detection in both fault detections. Fig. 17(c) and (d) shows that in the two highly low-speed stages, the misdiagnosis probability of the ASWIN strategy is still lower than that of the OPSWIN strategy.

The above is the feasibility verification of the proposed diagnostic strategies under the three fault injection modes. Here, only three corresponding injection mode experiments are taken

TABLE VII
EVALUATION AND COMPARISON OF FAULT DIAGNOSIS PERFORMANCE UNDER OPSWIN AND ASWIN SLIDING WINDOW STRATEGIES

Model	Methods	Accuracy (%)	F1-score (%)	Stationary Acc.(%)	Nonstationary Acc.(%)	Extreme-low-speed Acc.(%)	Low-speed Acc.(%)	Torque-disturbance Acc.(%)	Front-detection (s)	Delayed-detection (s)	Diagnosis Time(s)
SVM	OPSWIN	97.62	97.89	98.43	97.09	93.89	99.03	100.00	0.0347	0.0488	0.0583
	ASWIN	97.70	97.98	98.49	97.19	94.38	99.03	100.00	0.0366	0.0497	0.0632
DT	OPSWIN	95.96	97.18	97.81	94.77	87.03	98.13	100.00	0.0383	0.0291	0.0217
	ASWIN	96.47	97.47	98.08	95.40	87.82	98.42	100.00	0.0365	0.0288	0.0278
RF	OPSWIN	98.38	98.64	98.77	98.12	96.23	99.36	100.00	0.0442	0.0469	0.0696
	ASWIN	98.54	98.71	98.89	98.31	96.28	99.36	100.00	0.0452	0.0409	0.0764
AdaB	OPSWIN	98.12	98.43	98.63	97.79	94.95	99.21	100.00	0.0118	0.0471	0.1609
	ASWIN	98.26	98.54	98.81	97.90	95.14	99.28	100.00	0.0132	0.0456	0.1668
LR	OPSWIN	97.11	97.68	98.47	96.22	91.57	98.87	100.00	0.0241	0.0429	0.0783
	ASWIN	97.16	97.67	98.29	96.41	91.84	98.62	100.00	0.0151	0.0422	0.0879
KNN	OPSWIN	96.04	97.11	98.09	94.71	87.07	98.31	100.00	0.0430	0.0245	0.0449
	ASWIN	96.23	97.44	97.85	95.16	86.96	98.16	100.00	0.0428	0.0213	0.0502

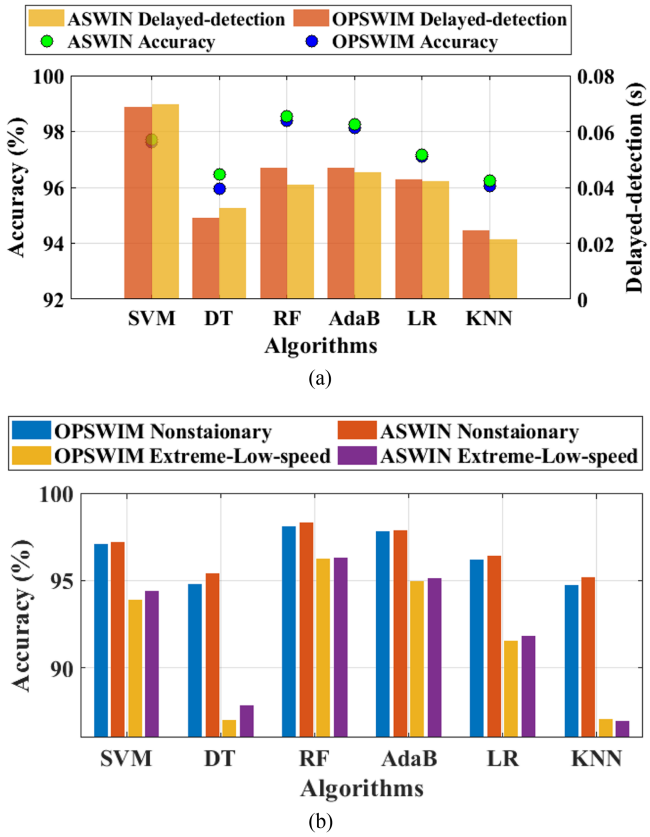

Fig. 18. Comparison of accuracy and delayed detection of six ML models under OPSWIN and ASWIN algorithms. (a) Dot plot shows the global accuracy, and the bar plot shows the delayed detection time. (b) Accuracy of various ML models under extreme conditions with two ASWIN algorithms.

as examples. For time deviation, the difference between the two algorithms is not significant. Still, in the highly low-speed stage, the misdiagnosis of the ASWIN algorithm is better than that of the OPSWIN algorithm.

3) Performance of Proposed Diagnosis Strategies: Next is a quantitative performance evaluation of the two diagnostic strategies, OPSWIN and ASWIN. Two dynamic sliding window strategies are still applied to the six ML algorithms. The specific evaluation results are given in Table VII, which tests the collected current signals of all the above operating and fault conditions. In addition, essential metrics from Table VII are visualized, as shown in Fig. 18. The dot plot in Fig. 18(a) represents the global accuracy on the left y-axis, and the bar chart represents the delayed-detection time on the right y-axis. The dot plot in Fig. 18(b) shows the accuracy of different algorithms under two extreme conditions, using different ASWIN strategies.

Table VII compares the proposed adaptive windowing strategies (OPSWIN and ASWIN) against fixed-length sliding windows. Both adaptive methods consistently improve real-time

TABLE VIII
FEATURE COEFFICIENT OF VARIATION UNDER SELECTED WINDOWING STRATEGIES

Operating Condition	Sliding Window Method	CV of Peak FFT Amplitude	CV of Kurtosis	CV of Impulse Factor
Extreme Low Speed	Fixed 200	0.4813	0.3887	0.3651
	Fixed 800	0.4311	0.3349	0.3082
	OPSWIN	0.3645	0.3656	0.3154
	ASWIN	0.3102	0.3631	0.3049
Low Speed	Fixed 200	0.3298	0.3915	0.2819
	Fixed 800	0.3147	0.3606	0.2677
	OPSWIN	0.3098	0.3219	0.2673
	ASWIN	0.3074	0.3191	0.2595
Nonstationary Range	Fixed 200	0.4467	0.4261	0.3372
	Fixed 800	0.4218	0.4070	0.3249
	OPSWIN	0.3942	0.3797	0.3159
	ASWIN	0.3911	0.3764	0.3161
Torque Disturbance	Fixed 200	0.4484	0.3949	0.3844
	Fixed 800	0.3331	0.3353	0.3637
	OPSWIN	0.2829	0.2654	0.2944
	ASWIN	0.2674	0.2633	0.1988

TABLE IX
COMPARISON OF VARIOUS OC FAULT DIAGNOSIS METHODS

Method Type	Specific method	Fault type	Accuracy (%)	Diagnosis time (ms)	Extra hardware	Empirical thresholds	Requirement of training
Model-based	Sliding Mode Observer [14]	6	/	≈ 1	NO	YES	NO
	Flux Observer [15]	7	/	< 15	NO	YES	NO
Signal-based	S-T Decomposition [18]	10	100	≈ 60	NO	YES	NO
	CEEMD [17]	12	/	≈ 30	YES	YES	NO
	Fuzzy Entropy [19]	6	100	≈ 50	NO	YES	NO
Data-driven	FFT+RVFL [27]	22	98.83	≈ 22	NO	NO	YES
	ELM [26]	22	96.76	≈ 50	NO	NO	YES
	Hybrid Ensemble Learning [22]	22	96.70	≈ 30	NO	NO	YES
	1D - CNN [23]	8	99.70	≈ 100	YES	NO	YES
	KNN+SVM [25]	6	98.61	≈ 60	NO	NO	YES
	Proposed method	22	97.70	≈ 60	NO	NO	YES

TABLE X
MAIN PARAMETER SETTINGS FOR REPRODUCED DIAGNOSTIC MODEL

Model	Parameter Settings	Training Time (s)	Model Size (KB)
1D - CNN	Raw signal sequence; 4 conv blocks (64/64/128/256, kernels 7–3, dilated); BN, ReLU, GAP, dropout 0.5, FC; Adam (60 epochs, batch 256).	821.6	996
RVFL	FFT + ReliefF (top-10/phase, 30-D); z-score normalization; 2500 hidden nodes, sigmoid activation, direct link on (C=10).	257.3	1299
ELM	PCA + GFK; internal normalization; 1800 hidden nodes, sigmoid activation (C=1).	216.6	1036
Hybrid method (RVFL + ELM)	Shared FFT+ReliefF features (30-D); GA/NSGA-II tunes hidden nodes and activations; diagnosis fusion by voting.	257.3	2335

ReLU: rectified linear unit.

diagnostic accuracy, particularly under nonstationary transitions and ultra-low-speed regimes, while incurring only modest computational overhead. For instance, ASWIN achieves up to 96% accuracy at ultra-low speeds, several percentage points higher than the fixed-window baseline, and overall accuracy exceeding 97%. The results also show that OPSWIN and ASWIN yield nearly identical overall accuracy ($>97\%$), while ASWIN offers slight advantages at very low speeds due to its speed-based self-tuning mechanism. Furthermore, despite the high computational requirements of the AdaB model, diagnostic latency is generally maintained within tens of milliseconds, with an average computation time of around 50 ms for each diagnosis. The diagnostic time deviation, including front-detection and delayed-detection, is around 35 ms, fully meeting real-time requirements. These results demonstrate that models trained on fixed-length multiscale samples can be directly combined with adaptive windowing, achieving desirable performance under extreme operating conditions, which confirms the generalizability and robustness of the proposed diagnosis strategy.

In addition, regarding the online computing time cost of the two proposed algorithms, OPSWIN requires about 20–30 ms and ASWIN about 30–40 ms on average, as shown by comparing Tables VI and VII. This time cost is acceptable, while significantly improving robustness under extreme conditions.

4) Analysis of Feature Extraction Robustness: To further explain the performance differences among the above diagnostic strategies and to evaluate the robustness of feature extraction under different sliding window strategies, three physically meaningful signal features were specifically analyzed: the peak FFT amplitude, which represents the main harmonic content and corresponds to the mechanical rotation frequency; the kurtosis factor, a fourth-order statistical measure sensitive to signal asymmetry and impulsivity; and the impulse factor, which quantifies the intensity of transient spikes relative to the average signal level. These features are extracted across four representative operating conditions: extreme-low speed, low speed, nonstationary range, and torque disturbance, using four windowing strategies: two fixed-length windows (200 and 800 samples) and two proposed adaptive methods (OPSWIN and ASWIN). For each case, the coefficient of variation (CV) was computed as a dimensionless indicator of relative feature dispersion

$$\mathrm{CV} = \frac{\sigma}{\mu + \varepsilon} \tag{19}$$

where μ and σ denote a given feature's mean and standard deviation over multiple signal segments, and ε is a small constant added to avoid division by zero.

A quantitative comparison is provided in Table VIII. The CV values reveal several key trends. First, increasing the fixed window length generally reduces CV, indicating improved feature extraction stability due to more complete signal coverage. However, fixed windowing still suffers from poor adaptability under nonstationary or disturbed conditions. In contrast, both adaptive windowing methods, OPSWIN and ASWIN, consistently achieve lower or comparable CV values under challenging regimes, such as extreme-low speed and torque disturbance. These results confirm that dynamic segmentation can better accommodate signal characteristics and window alignment changes.

These observations intuitively explain the performance differences between the fixed-size sliding window and the two

TABLE XI
EVALUATION AND COMPARISON OF THE DIAGNOSTIC PERFORMANCE OF THE REPRODUCED OC FAULT DIAGNOSIS METHODS

Model	Accuracy (%)	F1-score (%)	Stationary Acc.(%)	Nonstationary Acc.(%)	Extreme-low -speed Acc.(%)	Low-speed Acc.(%)	Medium-speed Acc.(%)	High-speed Acc.(%)	Disturbed Acc.(%)	Front -detection (s)	Delayed -detection (s)	Diagnosis Time(s)
1D - CNN	96.63	97.22	98.72	95.31	88.96	97.73	98.12	98.68	100.00	0.0474	0.0499	0.0881
RVFL	91.49	93.80	93.79	89.74	81.13	88.90	94.46	95.12	96.16	0.0589	0.0495	0.0396
ELM	90.42	92.25	92.96	88.67	80.81	89.51	92.50	93.55	94.63	0.0432	0.0511	0.0361
Hybrid method (RVFL + ELM)	90.81	92.82	93.29	88.09	81.17	88.81	93.30	93.63	94.79	0.0544	0.0358	0.0869
Proposed method	97.70	97.98	98.49	97.19	94.38	99.03	98.63	98.16	100.00	0.0366	0.0497	0.0632

TABLE XII
CALCULATION OF TIME-DOMAIN AND STATISTICAL FEATURES

Feature Type	Feature	Formula
Time Domain Features	Maximum	$\max(x)$
	Minimum	$\min(x)$
	Peak to Peak Value	$\max(x) - \min(x)$
	Root Mean Square (RMS)	$\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
	Peak to RMS	$\frac{\max(x)}{\mathrm{RMS}(x)}$
Statistical Features	Average ($\overline{x}$)	$\frac{1}{N}\sum_{i=1}^{N} x_i$
	Standard Deviation (σ)	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$
	Skewness	$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \overline{x}}{\sigma}\right)^3$
	Kurtosis	$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \overline{x}}{\sigma}\right)^4$
	Shape Factor	$\frac{\mathrm{RMS}(x)}{\overline{x}}$
	Impulse Factor	$\frac{\max(x)}{\overline{x}}$
	Margin Factor	$\frac{\max(x) - \min(x)}{\overline{x}}$

proposed methods, supporting the motivation of this study: the adaptive window enhances the performance of feature extraction, leading to more robust dynamic fault diagnosis under various operating conditions.

5) Comparison With Other OC Fault Diagnosis Methods: The introduction reviewed several recent research advancements in fault diagnosis, demonstrating promising validation results. To provide a comprehensive evaluation, we selected representative methods from these studies for comparative analysis against the proposed OC fault diagnosis strategy, as Table IX summarizes. The proposed framework enables a unified evaluation of diagnostic algorithms regarding accuracy, F1-score, and latency. Still, the results are not intended as dimensionless cross-comparisons with prior studies due to differences in datasets and task formulations.

Model-based approaches exhibit significant advantages in terms of diagnosis speed. However, they often rely on manually defined, complex threshold settings, making them less effective for multifault classification scenarios. Signal-based methods generally achieve high diagnostic accuracy but require longer computation times and face similar challenges due to the need for multithreshold configurations. Data-driven approaches, in contrast, typically do not require empirical threshold settings or additional sensors for acquiring supplementary data, making them particularly well-suited for multifault classification tasks. Compared with these methods, the method proposed in this article has the characteristics of simple implementation and a balanced tradeoff between diagnosis accuracy and computational efficiency.

The proposed ASWIN framework enhances diagnostic robustness across diverse operating regimes while maintaining real-time feasibility. Since the framework is independent of classifiers, the ensemble methods (e.g., RF and AdaB) are preferred when maximum accuracy under extreme conditions is required. In contrast, lightweight models (e.g., LR and DT) are more suitable for embedded applications with strict latency or memory constraints. The SVM model may be a tradeoff between multiple performance indicators; combined with the ASWIN algorithm, the diagnostic accuracy reaches 97.70% and the diagnosis time is about 60 ms. To account for variable-speed operation, the diagnosis time t_{diag} can be normalized as $\tau = t_{\text{diag}}/T_e = t_{\text{diag}} \cdot f_e$, the number of electrical cycles required to reach a decision. T_e and f_e represent the fundamental period and electrical frequency. In our experimental setup, this corresponds to approximately 0.4–2.4 current fundamental periods across the tested speed range of 200–1200 r/min, reflecting the adaptability and responsiveness of the proposed method. At extremely low speeds, the current remains theoretically periodic but exhibits strong distortion and nonperiodic components due to long fundamental periods. Thus, normalized representation is emphasized for speeds above 200 r/min, where the current waveform is stably periodic.

For more accurate analysis and comparison, some representative data-driven OC fault diagnosis methods were reproduced, trained, and validated using the same dataset and performance evaluation framework. The reproduced CNN is based on built-in MATLAB functions, while the reproduced RVFL and ELM models are implemented by utilizing and modifying the open-source codes proposed in [32] and [33]. Table X presents the main implementation parameters, as well as the training time and size of the trained models from an engineering perspective. According to the literature description, the sliding window size for the dynamic validation phase of all reproduced models is 800 sampling points.

As summarized in Table XI, the proposed method achieves the highest overall accuracy (97.70%) and F1 score (97.98%), particularly outperforming the reproduced methods under extreme operating conditions. These improvements are evident under nonstationary, low-speed, and very low-speed conditions. In contrast, traditional CNN, RVFL, and ELM models exhibit increased sensitivity to spectral distortion and transient imbalance, which reduces their performance under extreme conditions. In addition, some reproduced methods adopt a multiple-vote strategy, which increases the diagnosis time of a single diagnosis. These improvements demonstrate that the proposed adaptive variable sliding window method can ensure the robustness of fault diagnosis under extreme operating conditions without significantly increasing the delay.

VI. Conclusion

This article proposed a novel data-driven real-time OC fault diagnosis strategy for three-phase inverters, integrating an ASWIN approach with ML models to enable full-speed-range fault detection in WRSM drive systems. A range-based evaluation strategy is also proposed to assess real-time diagnostic performance under dynamic scenarios where a single window may encompass multiple operating states. Range-based accuracy metrics optimize global parameter combinations and segment-specific optimal parameters for different speed ranges. In addition, a performance evaluation across multiple operating regimes, including extreme-low to high speed, nonstationary and stationary operation, and torque-disturbance intervals, was proposed.

First, six ML models were trained on multiscale segmented current signals to diagnose faults under various operating conditions. The best-performing RF model achieved a static validation accuracy of 97.48%. Second, a dynamic scaling factor for window size calculation was developed, incorporating signal metrics, such as dominant frequency ratio, signal variation rate, and signal-to-noise ratio. This approach enables the window size to adapt effectively to the dynamic characteristics of current signals based on current information and short-term prediction, thereby better extracting local features. Third, two ASWIN algorithms were implemented: the OPSWIN algorithm, employing global parameters optimization through GA, and the ASWIN algorithm, which performs online speed-based parameter self-tuning through offline segmented parameter optimization. Experimental validation using collected current signals demonstrated the effectiveness of the proposed strategies across various fault scenarios, including stationary single faults, nonstationary single faults, and mixed stationary faults. The ASWIN algorithm combined with the SVM model achieved a diagnostic accuracy of 97.70% with a diagnosis time of approximately 60 ms, corresponding to a range of 0.4–2.4 current fundamental periods across a speed range of 200–1200 r/min. Compared to fixed-length sliding window methods, the ASWIN approach significantly reduced diagnostic delay while maintaining high accuracy.

References

[1] C. Wu, R. Sehab, A. Akrad, and C. Morel, "Fault diagnosis methods and fault tolerant control strategies for the electric vehicle powertrains," *Energies*, vol. 15, no. 13, 2022, Art. no. 4840.

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

[3] L. M. A. Caseiro and A. M. S. Mendes, "Real-time IGBT open-circuit fault diagnosis in three-level neutral-point-clamped voltage-source rectifiers based on instant voltage error," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1669–1678, Mar. 2015.

[4] J. Lamb and B. Mirafzal, "Open-circuit IGBT fault detection and location isolation for cascaded multilevel converters," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4846–4856, Jun. 2017.

[5] U.-M. Choi, F. Blaabjerg, and K.-B. Lee, "Study and handling methods of power IGBT module failures in power electronic converter systems," *IEEE Trans. Power Electron.*, vol. 30, no. 5, pp. 2517–2533, May 2015.

[6] B. Cai, Y. Zhao, H. Liu, and M. Xie, "A data-driven fault diagnosis methodology in three-phase inverters for PMSM drive systems," *IEEE Trans. Power Electron.,*vol. 32, no. 7, pp. 5590–5600, Jul. 2017.

[7] I. Jlassi, J. O. Estima, S. K. El Khil, N. M. Bellaaj, and A. J. M. Cardoso, "A robust observer-based method for IGBTs and current sensors fault diagnosis in voltage-source inverters of PMSM drives," *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 2894–2905, May/Jun. 2017.

[8] B. Lu and S. K. Sharma, "A literature review of IGBT fault diagnostic and protection methods for power inverters," *IEEE Trans. Ind. Appl.*, vol. 45, no. 5, pp. 1770–1777, Sep./Oct. 2009.

[9] J. O. Estima and A. J. Marques Cardoso, "A new algorithm for real-time multiple open-circuit fault diagnosis in voltage-fed PWM motor drives by the reference current errors," *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3496–3505, Aug. 2013.

[10] N. Wassinger, E. Penovi, R. G. Retegui, and S. Maestri, "Open-circuit fault identification method for interleaved converters based on time-domain analysis of the state observer residual," *IEEE Trans. Power Electron.*, vol. 34, no. 4, pp. 3740–3749, Apr. 2019.

[11] H. Berriri, M. W. Naouar, and I. Slama-Belkhodja, "Easy and fast sensor fault detection and isolation algorithm for electrical drives," *IEEE Trans. Power Electron.*, vol. 27, no. 2, pp. 490–499, Feb. 2012.

[12] J. Poon, P. Jain, C. Spanos, S. K. Panda, and S. R. Sanders, "Fault prognosis for power electronics systems using adaptive parameter identification," *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 2862–2870, May/Jun. 2017.

[13] X. Zhou, J. Sun, P. Cui, Y. Lu, M. Lu, and Y. Yu, "A fast and robust open-switch fault diagnosis method for variable-speed PMSM system," *IEEE Trans. Power Electron.*, vol. 36, no. 3, pp. 2598–2610, Mar. 2021.

[14] S. Xu et al., "A novel adaptive SMO-based simultaneous diagnosis method for IGBT open-circuit faults and current sensor incipient faults of inverters in PMSM drives for electric vehicles," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 3526915.

[15] N. Diao, Y. Zhang, X. Sun, C. Song, W. Wang, and H. Zhang, "A real-time open-circuit fault diagnosis method based on hybrid model flux observer for voltage source inverter FED sensorless vector controlled drives," *IEEE Trans. Power Electron.*, vol. 38, no. 2, pp. 2539–2551, Feb. 2023.

[16] B. Wang, Z. Li, Z. Bai, P. T. Krein, and H. Ma, "A voltage vector residual estimation method based on current path tracking for T-type inverter open-circuit fault diagnosis," *IEEE Trans. Power Electron.*, vol. 36, no. 12, pp. 13460–13477, Dec. 2021.

[17] R. Wang, L. Song, J. Cheng, Y. Huang, and J. Ke, "Open-circuit diagnosis method for T-type three-level inverter based on extracted IMF energy feature," *IEEE Trans. Instrum. Meas.*, vol. 73, 2024, Art. no. 3530413.

[18] Y. Zhou, J. Zhao, Y. Song, J. Sun, H. Fu, and M. Chu, "A seasonal–trend-decomposition-based voltage-source-inverter open-circuit fault diagnosis method," *IEEE Trans. Power Electron.*, vol. 37, no. 12, pp. 15517–15527, Dec. 2022.

[19] C. Wu, R. Sehab, C. Morel, and A. Akrad, "Entropy-based diagnosis method for open-circuit fault of an inverter supplying wound rotor synchronous machine of electric vehicle powertrain," in *Proc. 4th Int. Conf. Adv. Elect. Energy Syst.*, 2023, pp. 252–257.

[20] I. Abari, A. Lahouar, M. Hamouda, J. B. H. Slama, and K. Al-Haddad, "Fault detection methods for three-level NPC inverter based on DC-bus electromagnetic signatures," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5224–5236, Jul. 2018.

[21] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.

[22] Y. Xia, Y. Xu, and B. Gou, "A data-driven method for IGBT open-circuit fault diagnosis based on hybrid ensemble learning and sliding-window classification," *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, pp. 5223–5233, Aug. 2020.

[23] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Al-Emadi, and M. Gabbouj, "Real-time fault detection and identification for MMC using 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8760–8771, Nov. 2019.

[24] F. Deng, M. Jin, C. Liu, M. Liserre, and W. Chen, "Switch open-circuit fault localization strategy for MMCs using sliding-time window based features extraction algorithm," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 10193–10206, Oct. 2021.

[25] C. Wu, R. Sehab, A. Akrad, and C. Morel, "Online data-driven diagnosis method for open-circuit fault of an inverter supplying wound rotor synchronous machine of electric vehicle powertrain," in *Proc. 2023 Int. Conf. Control, Automat. Diagnosis*, 2023, pp. 1–7.

[26] Y. Xia and Y. Xu, "A transferrable data-driven method for IGBT open-circuit fault diagnosis in three-phase inverters," *IEEE Trans. Power Electron.*, vol. 36, no. 12, pp. 13478–13488, Dec. 2021.

[27] B. Gou, Y. Xu, Y. Xia, Q. Deng, and X. Ge, "An online data-driven method for simultaneous diagnosis of IGBT and current sensor fault of three-phase PWM inverter in induction motor drives," *IEEE Trans. Power Electron.*, vol. 35, no. 12, pp. 13281–13294, Dec. 2020.

[28] N. Diao, X. Zhang, Y. Zhang, X. Guo, and X. Sun, "An open-circuit fault diagnosis method based on horizon-adaptive period correction for voltage source inverter in motor drive," *IEEE Trans. Ind. Electron.*, vol. 72, no. 3, pp. 3161–3171, Mar. 2025.

[29] Z. Liu, L. Fang, D. Jiang, and R. Qu, "A machine-learning-based fault diagnosis method with adaptive secondary sampling for multiphase drive systems," *IEEE Trans. Power Electron.*, vol. 37, no. 8, pp. 8767–8772, Aug. 2022.

[30] N. Tatbul, T. J. Lee, S. B. Zdonik, M. Alam, and J. E. Gottschlich, "Precision and recall for time series," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 1924–1934.

[31] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[32] L. Zhang and P. Suganthan, "A comprehensive evaluation of random vector functional link networks," *Inf. Sci.*, vol. 367-368, pp. 1094–1105, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025515006799

[33] V. H. A. Ribeiro, G. Reynoso-Meza, and H. V. Siqueira, "Multi-objective ensembles of echo state networks and extreme learning machines for streamflow series forecasting," *Eng. Appl. Artif. Intell.*, vol. 95, 2020, Art. no. 103910. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197620302451