IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 15, 2021, accepted March 20, 2021, date of publication April 1, 2021, date of current version April 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3070483

Modified Fuzzy-Q-Learning (MFQL)-Based Mechanical Fault Diagnosis for Direct-Drive Wind Turbines Using Electrical Signals

HASMAT MALIK[1], (Senior Member, IEEE), AND ABDULAZIZ ALMUTAIRI[2], (Member, IEEE)
[1]Berkeley Education Alliance for Research in Singapore (BEARS), Singapore 138602
[2]Department of Electrical Engineering, College of Engineering, Majmaah University, Al Majma'ah 11952, Saudi Arabia

Corresponding authors: Abdulaziz Almutairi (ad.almutiri@mu.edu.sa) and Hasmat Malik (hasmat.malik@gmail.com)

This work was supported by the Deanship of Scientific Research, Majmaah University under Grant R-2021-68.

ABSTRACT In this paper, a self-learning multi-class intelligent model for wind turbine fault diagnosis is proposed by using MFQL (Modified-Fuzzy-Q-Learning) technique. The MFQL is adaptive in nature and extension of fuzzy-Q-learning method where look-up table of Q-learning is conquered by fuzzy based approximation strategy to reduce the curse of dimensionality of the Q-learning. The proposed MFQL classifier diagnoses the mechanical and imbalance faults without using mechanical sensors. Proposed methodology is addressed with relying on PMSG (Permanent Magnet Synchronous Generator) stator current signals, which is already being used by protection system of wind turbines. According to the aforementioned description, non-stationary current signals of PMSG have been pre-processed to extract the input features by empirical mode decomposition followed with J48 algorithm based most relevant input feature selection. For the one-step ahead performance demonstration of the proposed MFQL approach, results have been compared with neural network, support vector machines, fuzzy logic, and conventional Fuzzy-Q-Learning techniques. Demonstrated results outperform the capability of proposed MFQL approach. Moreover, MFQL is developed first time to implement in the area of WTGS fault diagnosis in the literature.

INDEX TERMS J48 algorithm, machine learning, fault diagnosis, FAST, dynamic modeling, wind turbine, TurbSim, real-time analysis, imbalance fault, non-intrusive.

I. INTRODUCTION

Wind industry is growing up day-by-day to meet consumer demand and established power in India was 343789 MW upto April 2018 leads to fifth in rank in the world [1]. Under the national wind resource assessment program, MNRE (ministry of new and renewable energy) through CWET (centre of wind energy technology), state and private nodal agencies, has installed grid interactive wind power of 45.2% (36368 MW) of total renewable energy. So, power system dependability on wind energy is increasing day-by-day, which may lead difficulty in dynamic healthy operation with uninterrupted power supply to end user. Hence, condition monitoring, fault detection & diagnosis (CM-FDD) of WT is become more important, which is very difficult to perform under perturbs operating conditions. As per available study in [2]–[4], the down time period of WT is from 52 to 237 hours in a year. The main causes of WT downtime are the failure of its components and related sub-system i.e., failure to blades/hub/pitch (13.7%), failure to control system (12.9%), failure to drive train (1.1%), failure to electric system (17.5%), failure to gearbox (9.8%), failure to generator (5.5%), failure to hydraulic and brakes (14.5%), failure to sensors (14.1%), failure to structure (1.5%), failure to yaw system (6.7%), and failure to other (2.7%) [29]. Some imbalance faults also play a major role in WT downtime. The most common imbalance faults (IF) are imbalance mass density in blades (IFB: IF in blade) (symbol: AdjBlM), shaft imbalance (IFS: IF in shaft), tail furl imbalance (IFT: IF in TailFurl) (symbol: TailFurl), rotor furl imbalance (IFR: IF in rotor) (symbol: RotFurl), nacelle-yaw imbalance (IFY: IF in yaw) (symbol: NacYaw) and aerodynamic asymmetry (IFP: IF in pitch) (symbol: BlPitch) in wind-turbine-generating-system (WTGS) which leads 16.22% of total failure rate.

NacYaw imbalance fault can be generated by control error in yaw system of WT which modify the required position of nacelle system. RotFurl and TailFurl imbalance faults can be developed by amend furl angle in RotFurl and/or TailFurl respectively of WT. Similarly, AdjBlM imbalance fault in

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwei Gao.

blade can be generated by change in mass density of the one blade as compare with other blades. The variation of the mass density in blade is due to manufacturing errors, icing condition, deterioration, and fatigue during operation of WT which develop additional loading on tower leading to collapse the system. BlPitch imbalance fault can be generated by aerodynamic asymmetry caused by malfunctioning of control mechanism and high wind shear which amend BlPitch angle of one blade from the desired value. Due to this, an unbalance torque is generated which leads to aerodynamic asymmetry on rotating shaft.

Presented maximum research for fault analysis of WTGS imbalance faults are based on sensors [6], [7]. While sensors are not easily accessible due to high height of tower and also impact approximate 14% failure [28]. Therefore, machine current signals analysis (MCSA) based condition monitoring become enthusiastic which decrease the maintenance cost, improve the system life and protect the system from catastrophic failures.

For the condition monitoring of WTGS, several artificial intelligence schemes (i.e., based on fuzzy-logic, MLP, LVQ, PNN, SVM etc.) have recommended using current signature based approach for fault diagnosis as mentioned in [2], [13], [15], [26]–[28]. However, these approaches have some drawbacks as mentioned in [2], [13], [15]. Majority of the mentioned problems of [2], [13], [15] (i.e., low diagnosis accuracy in fuzzy-logic, required huge amount of training data in neural networks, difficult in parameter selection in SVM & PSVM, required large storage memory in PNN, low processing speed in LVQ, addition of inherent noise in PLL, etc.) can be overcome with the proposed MFQL based classifier. The MFQL classifier is based on reinforcement learning based classifier The main properties of MFQL based classifier are: 1) classifier assign the penalty for each wrong classification and allow for correction in next decision stage, 2) it is adaptable to correct own behaviour as per gained experimental knowledge, 3) it does not require previous system information like model information or parameters, 4) the procedure for consequent rewards-punishment adjustment for the fuzzy-q-values is in an incremental manner, which increase classification accuracy, 5) it includes a heuristic feedback mechanism (i.e., reward/punishment mechanism), 6) it learn in a sufficient number of training samples to classify mechanical faults correctly.

This paper is well ordered in the six sections: the main problematic are listed in Section I. In Section II, the dynamic of WTGS model using amalgam platform of FAST, TurbSim and Simulink is presented. In Section-III, feature extraction and feature selection methodology are presented by using EMD and J48 algorithm respectively. The detailed procedure for implementation of proposed MFQL classifier is also presented in Section-III. In Section-IV, mathematical validation of MFQL based results are describes. In Section V, results-and-discussion are mentioned and finally the conclusion of the study is explained in Section-VI.

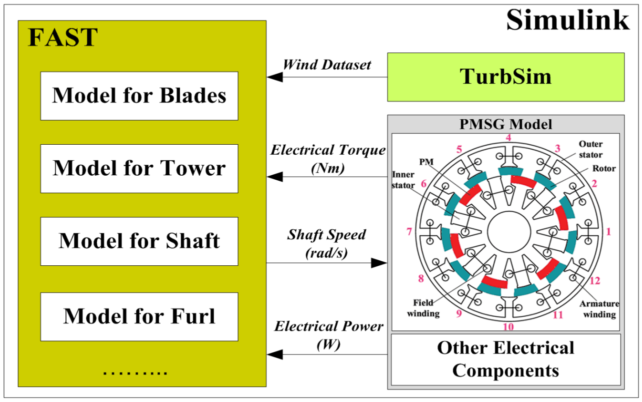

FIGURE 1. Complete dynamical WTGS model arrangement in amalgamate environment of simulink of MATLAB, FAST and TurbSim.

II. WTGS DYNAMIC FORMATION

A. BRIEF DETAILS OF WTG MODEL IMITATION

In this study, a coalesced platform of three distinct software (i.e., FAST [5], TurbSim [22] and Simulink) is developed to form the whole dynamics of real-time WTGS as demonstrated in Figure 1. The FAST (fatigue, aerodynamic, structure, turbulence) is utilized for developing the dynamics of a real-time WT. TurbSim is utilized to generate time series aeroelastic imitation for wind data (i.e., non-linear and non-stationary in nature) and Simulink platform is utilized to design the PMSG and other electrical equipments.

FAST is open access, most advanced code, which can be utilized for designing of onshore and/or offshore, rigid and/or teethering hub, upwind and/or downwind rotor, pitch and/or stall regulation, lattice and/or tubular tower, 2 and/or 3 blade horizontal axis non-linear WT and its performance was certified by Germanischer Lloyd [21]. FAST code is the amalgamation of 3-distinict codes (i.e., FAST2, FAST3 and AeroDyn aerodynamics subroutines). FAST code also includes model for blades, tower, shaft and furl. The performed subroutine by FAST is amalgamated Simulink with the help of speed, power and torque signals.

TurbSim is an emulator which emulates a stochastic inflow turbulence wind velocity vectors of 15.7 m/s (i.e., a mean value) [22]. It is the advanced model than IEC based Turbulence Models. The main advantage of the TurbSim is that the dynamic of time-series 3-D wind velocity vector in stochastic, full-field, turbulent can be generated numerically at points in a vertical rectangular grid, which is used as an input into the AeroDyn-based codes such as FAST, YawDyn, or MSC.ADAMS®.

In this study, NREL, USA based a standard WTGS model of 10 kW rating [5] is used which has 34 m tower height, 3-blades of rotor diameter of 5.8 m, nacelle mass of 260 kg, hub mass of 113 kg with 48 poles of PMSG. PMSG and other electrical components are imitated in MATLAB simulink platform. PMSG is used to generate electrical power from the WT mechanical power, which is based on the series of real-time wind speed. In the recorded data, PMSG stator current, output electric power, shaft rotation speed and torque

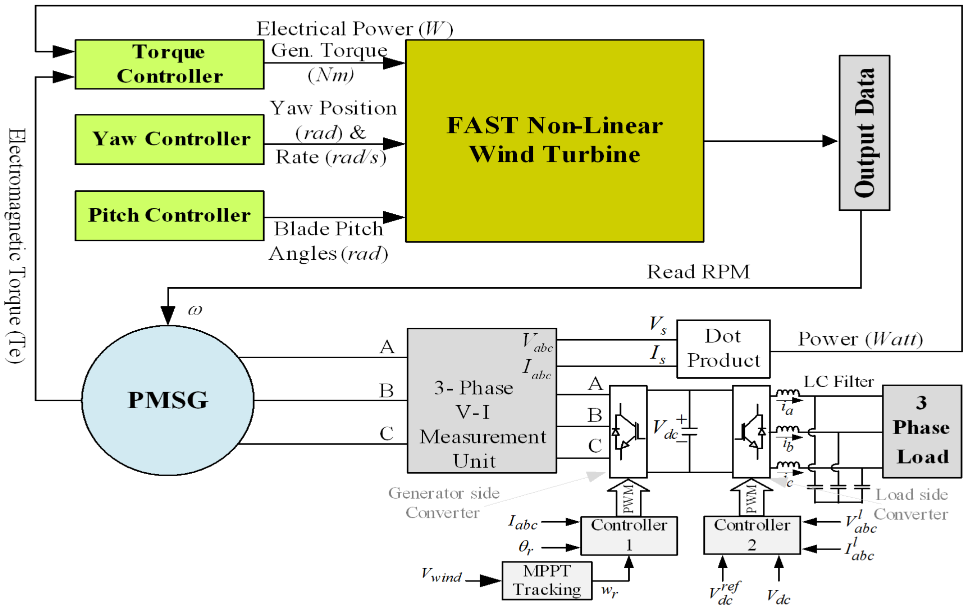

FIGURE 2. Model of WTGS in FAST and simulink combined simulation platform.

TABLE 1. Fault simulation files and its associated parameters [5].

Fault name	Parameters	FAST File
IFT	TailFurl	Test17.fst
IFP	BlPitch	Test17.fst
IFB	AdjBlMs	SWRT_Blade.dat
IFR	RotFurl	Test17.fst
IFY	NacYaw	Test17.fst

are collected. The maximum logged current amplitude is 35 A at the maximum wind speed scenario and the output electrical power of PMSG is varied in the range of 7-14 kW, where maximum power limitation of PMSG is not modeled. Recorded signals (i.e., non-linear and non-stationary in nature) of PMSG stator side is utilized for further study in WTGS fault analysis.

B. IMBALANCE FAULT FORMULATION FOR FURTHER STUDY

Developed 10 kW WTGS (as depicted in Figure 2) are tested for five faulty and one healthy operating scenario. AdjBlM imbalance fault is generated by varying mass density (MD) (+2%, +5% and −3%) of one blade which creates diverse distribution of mass w.r.t. rotor. TailFurl and/or RotFurl imbalance fault is generated by varying tail and/or rotor furl angle (+10°, −5° and +5°) apart of essential position, which creates irregular WT direction. NacYaw imbalance fault is generated by varying yaw angle from required position ($+20^0$, $+10^0$ and -10^0), which creates irregular position of rotor toward the wind. Finally, BlPitch imbalance fault is created by varying pitch angle of one blade from the required position ($+10^0$, $+5^0$ and -8^0), which generates irregular torque on rotor. Essential library and its associated variables of FAST code are tabulated in Table 1 which are utilized for creation of imbalance faults.

Emulated WTGS model under six distinct conditions are executed for 40s with 2kHz sampling frequency and electric power, stator current, turbine shaft toque and wind speed are recorded for further study. The input feature extraction and selection using PMSG stator current is demonstrated in subsequent sections.

III. METHODOLOGY

The proposed approach for the implementation of the whole methodology for non-intrusive fault detection and diagnosis is presented in Figure 3, which includes the following operation: 1) Dynamic model development of the WTGS using FAST, TurbSim and Simulink, 2) Different type of the imbalance fault creation using dynamic model of FAST, 3) capture the different type of electrical and mechanical signals under different operating conditions with and without fault scenario, 4) data pre-processing for filling the missing value and spikes removal (if any), 5) feature extraction using EMD method, 6) most relevant feature selection using machine learning method of J48 algorithm, 6) MFQL model development, 7) perform the training and testing of the developed intelligent model, and 8) after cross validation of the performance, save the model for future use. The detailed information for each subsection of the proposed approach is represented in this paper.

A. FEATURE EXTRACTION USING EMD [23]

In this study, features are extracted by using EMD (Empirical Mode Decomposition) technique, which is a data dependent adaptive-signal processing method which decomposes non-stationary and/or stationary signal into intrinsic mode functions (IMFs). The step-by-step process for creating IMFs from a signal $y(t)$ is as:

A1. Load the data set signal $y(t)$ first, then find the extrema values i.e., minima & maxima and Use cubic spline interpolation for connecting them.

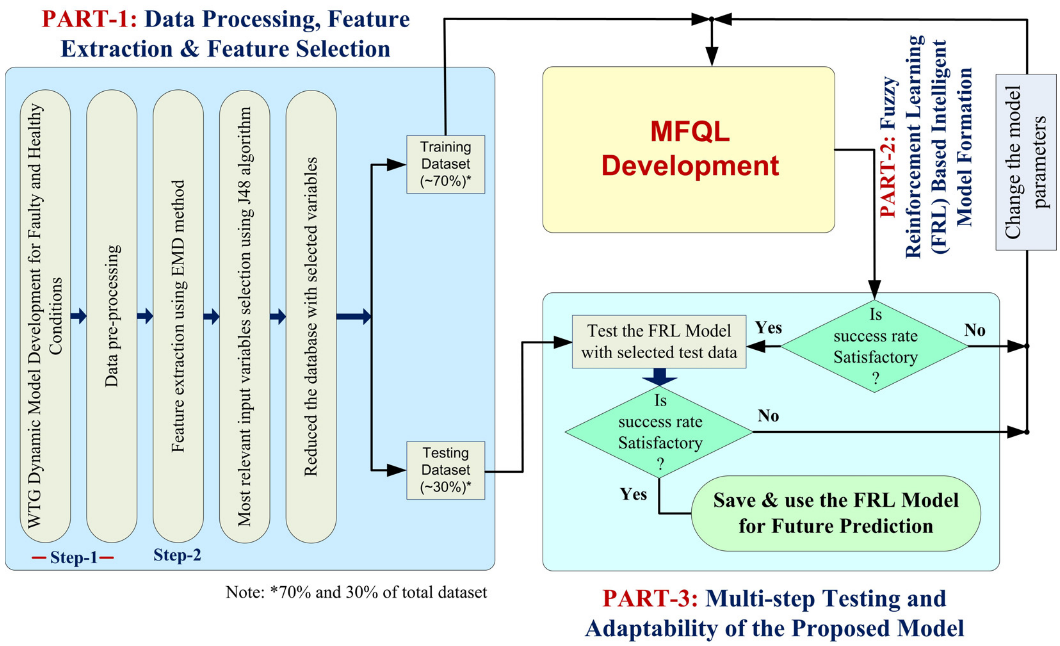

FIGURE 3. Proposed approach for non-intrusive fault detection and diagnosis of WTGS.

A2. Estimate an upper $[e_m(t)]$ and lower $[e_t(t)]$ envelope and then mean value $m(t)$:

$$m(t) = [em(t) + e_t(t)]/2 \tag{1}$$

A3. Define the value of $y(t) - m(t)$:

$$H_1(t) = [y(t) - m(t)] \tag{2}$$

A4. Check $H_1(t)$ fulfils both situations of IMFs. If yes, $H_1(t)$ is become IMF#1, else $H_1(t)$ is treated as original signal and redo 1-4 steps. Follow this procedure k-time, $H_1(k)$ become an IMF:

$$H_1(k) = H_{1(k-1)} - a(t) \tag{3}$$

A5. Delineate $\Omega_1(t) = H_{1k}(t)$, with $\Omega_1(t)$ being IMF#1 of original signal. (where, $\Omega_1(t)$ = smallest temporal scale)

A6. Compute residue value:

$$\psi_1(t) = y(t) - \Omega_1(t) \tag{4}$$

assume $\psi_1(t)$ = indigenous signal and redo above process to evaluate IMF#2.

A7. Redo this method i-times to generate i IMFs of $y(t)$ and dismiss the procedure if $\psi_1(t)$ = monotonic function. At last, after implementation the process (from point A1-A7), the $y(t)$ is retrieved by Eq. (5):

$$y(t) = \sum_{l=1}^{L} \Omega_l(t) + \psi_L(t) \tag{5}$$

where $\psi_L(t)$ = residue, $\Omega_l(t) = l^{th}$ IMF and, L = number of IMFs

From the energy level of each IMF, normal and faulty conditions can be distinguishes, as depicted in Figure 4. The E_e (energy entropy) is determined by Equation (6) and are listed in Table 2, which shows the difference in magnitude of entropy for each case.

$$E_e = -\sum_{i=1}^{L} \mathrm{P}_i \log \mathrm{P}_i \tag{6}$$

here, $\mathrm{P}_i = E_i/E$ energy magnitude in (%) for i^{th} IMF, where

$$E = \sum_{i=1}^{L} E_i = \text{energy of } y(t) \tag{7}$$

Based on Table 2 and Figure 4, it is analyzed that energy distribution of IMFs varies w.r.t. fault type. Here, the y-axis of each IMF represents the energy magnitude of the IMF, whereas total number of data samples are represented on x-axis of the figure 4.

B. MOST INFLUENCING INPUT (MII) SELECTION USING J48

The selection of the MII is a big research area, which affect the model performance. In this study, J48 algorithm is used to select the MII, which is extensively utilized to assemble a typical decision tree (DT) and utilizing theory of information entropy for attribute selection and identification [18]. In this study, generated IMFs vector of EMD (as demonstrated in A1-A7 of section 3.1 and in a matrix of Eq.8) are pruned with redundant attribute to form a group of utmost suitable attributes.

$$H = [imf_1, imf_2, imf_3, \ldots\ldots, imf_{17}]_{96000\times 17} \tag{8}$$

Utilized the input matrix $H : x_i \in R^n, i = 1, 2, \ldots, l$, target: $y \in R^l$, then J48 split space with same target samples and

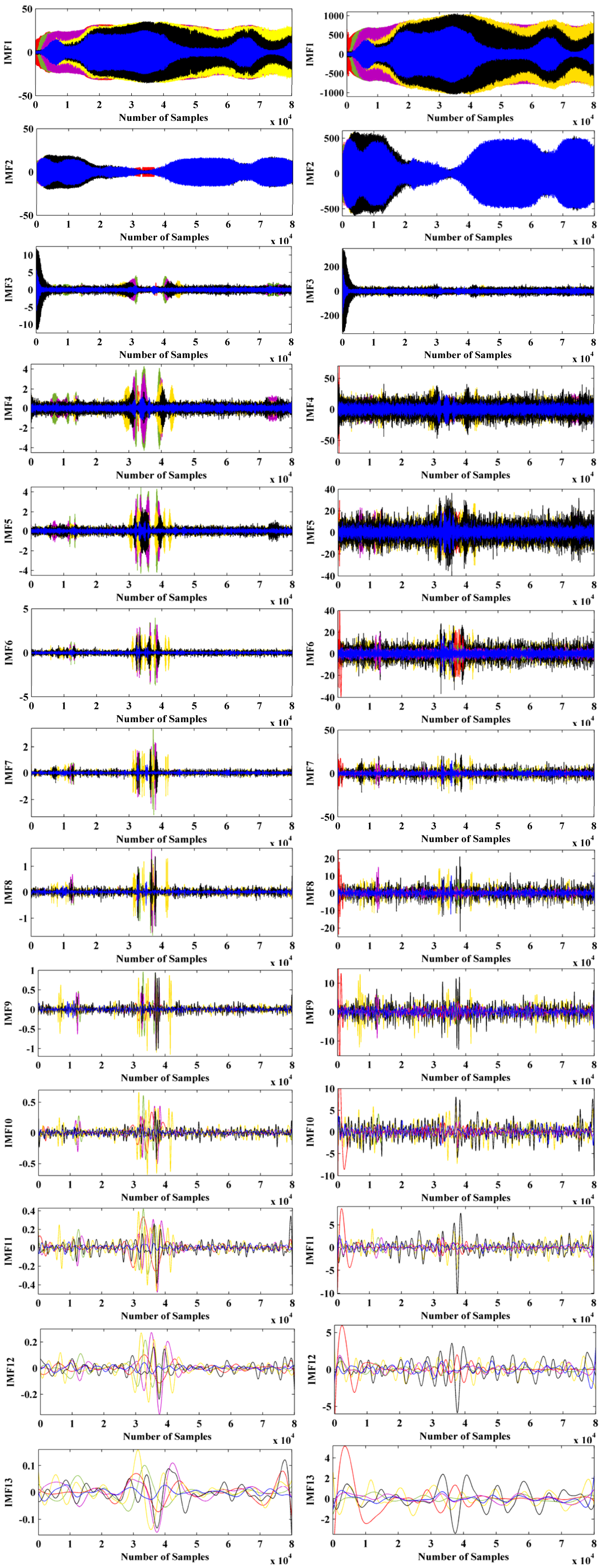

FIGURE 4. IMFs representation for 5-Faults and healthy condition of: (a) Current signal and (b) Voltage signal.

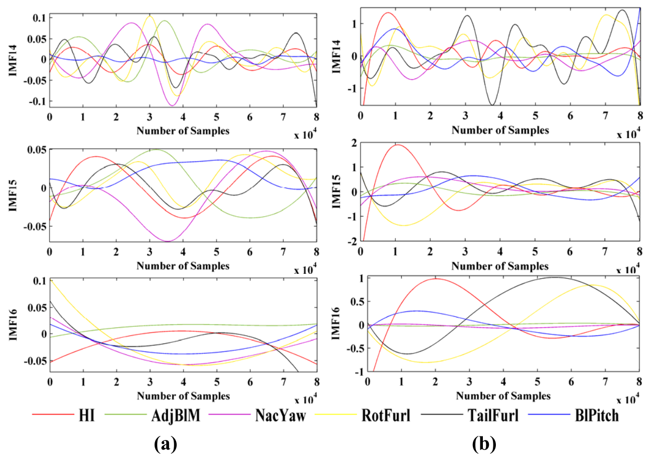

FIGURE 4. *(Continued)* IMFs representation for 5-Faults and healthy condition of: (a) Current signal and (b) Voltage signal.

TABLE 2. Determined energy entropies for current based IMFs.

Condition	Healthy	IFB	IFP	IFY	IFR	IFT
Magnitude	0.673	0.697	0.697	0.697	0.724	0.753

are grouped together. Assune data at node m be designated by β for each specimen divide $\lambda = (j, t_m)$ with feature j and threshold t_m, and data can be divided into $\beta_{left}(\lambda)$ and $\beta_{right}(\lambda)$ subgroups as:

$$\beta_{left}(\lambda) = (x, y)|\, x_j <= t_m \text{ and } \beta_{right}(\lambda) = \beta \backslash \beta_{left}(\lambda) \quad (9)$$

Impurity at m is evaluated by its function $H()$ according to performed task (such as classification/ regression).

$$G(\beta, \lambda) = \frac{\eta_{left}}{N_m} H(\beta_{left}(\lambda)) + \frac{\eta_{right}}{N_m} H(\beta_{right}(\lambda)) \quad (10)$$

Optimized the parameters to reduce impurity as:

$$\lambda^* = \arg\min_{\theta} G(\beta, \lambda) \quad (11)$$

Iterate again-and-again for $\beta_{left}(\lambda^*)$ and $\beta_{right}(\lambda^*)$ till $N_m <$ $\min_{samples}$ or $N_m = 1$.

If problem is formulated for classification, then target is $0, 1, \ldots, K-1$, for m node, and notifying a region R_m with N_m instances is

$$p_{mk} = 1/N_m \sum_{x_i \in R_m} I(y_i = k) \quad (12)$$

Generally, evaluation of impurity (i.e., Gini):

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}) \quad (13)$$

Cross-entropy:

$$H(X_m) = -\sum_k p_{mk} \log(p_{mk}) \quad (14)$$

Misclassification:

$$H(X_m) = 1 - \max(p_{mk}) \quad (15)$$

TABLE 3. J48 model performance analysis.

Model	Total variables	Selected variables	Classified Instances	Processing time (second)	Leaves Number	Tree Size
Model1: Current based IMFs	16	8	99.98%	2.45	31	61
Model2: Voltage based IMFs	17	15	99.02%	11.45	150	299

TABLE 4. Class-wise classification matrix.

C0	C1	C2	C3	C4	C5	<-- classified as
6000	0	0	0	0	0	C0 = HI
0	17996	0	3	0	1	C1 = IFB
0	5	17994	1	0	0	C2 = IFP
0	2	1	17996	0	1	C3 = IFY
0	0	4	3	17993	0	C4 = IFR
0	0	0	0	0	18000	C5 = IFT

Here, two models have been created based on J48 algorithm (Table 3). Comparative analysis of Table 3 shows that J48 selects 8 IMFs for Model#1 and 15 IMFs for Model#2 as MII to the classifier. As per the performance analysis, Model#1 is comparable over Model#2. Thus, Model#1 is selected as suitable model for WTGS diagnosis and performance analysis, and has been depicted in Table 4 for each case of Model#1.

C. MODIFIED FUZZY Q LEARNING (MFQL) FRAMEWORK

In this section, implementation of proposed MFQL classifier has been explained in detail. Firstly, detail of Q-learning and then Fuzzy-Q-learning details have been presented for proper understanding of MFQL implementation.

1) Q-LEARNING (QL)

The QL is a model free incremental reinforcement learning technique with proven convergence [19], which shows several applications in control domain with numerous benchmark nonlinear problem solutions [19], [20]. QL is a model free algorithm for optimal decision making under uncertainty.

QL includes an agent-making a sequence of attempts at classifying the condition, starting from a preliminary position:

$$s^k \in S(k = 0, 1, 2 \ldots) \tag{16}$$

where, k = stage variable/time instant; and S = state space.

This series of actions either achieve success or it may be a failure. This agent is appraised for it success and panelized for its failure. Q based function is utilized for the quality judgment of state action sequence:

$$Q(s^k, y(s^k)) \tag{17}$$

where $y(s^k) \in Y(s^k)$ = action taken in state $s^k \in S$.

Here the system builds a transition to next state: $s^k \xrightarrow{r} s^{k+1}$ with the agent getting a reinforcement signal or reward r. The r performs as a sign of "bad" or "good" deed medley by the agent. These all are performed to attain an optimal decision. In this study, agent tries to classify the faults appropriately by analyzing rewards/punishment acknowledged at the end of an endeavor. After adequate repetitions, the classifier is competent to classify proper type of fault. However, Q-learning is useful for small and/or discrete state space by using look-up table for storing the q values. If a problem requires state-space for continuous action or state space become very large, then look-up table of Q-learning is become infeasible. For such type of problem, Fuzzy or ANN based Q-learning become feasible.

2) FUZZY Q LEARNING (FQL)

Approach based on look-up table of Q-learning is also known as "Curse of Dimensionality" which can be conquered by utilizing approximation technique to approximate the Q function. The approximation can be performed by using ANN or fuzzy approach for replacing the look-up table to enhance the Q-learning [20].

In this paper, fuzzy method is utilized for the approximation of Q-function. Fuzzy-Q-learning tally input vector at instant k:

$$s^k = \left\{ s_1^k, s_2^k, \ldots, s_n^k \right\} \tag{18}$$

where, n = state or number of state variables

Generated n based rule firing strengths:

$$R_i : \alpha_i \left(s^k \right) \tag{19}$$

With the help of each rule, m actions $Y = \{y_1, y_2, \ldots, y_m\}$ can be chosen where q is the quality of each action in the particular rule. Fuzzy-inference-system (FIS) for each rule $R_i, i \in N$ is described as:

$$\left(\begin{array}{ll} R_i: \text{ If } s_1^k \text{ is } T_1^i \text{ and} & \\ \ldots\ldots \text{ and } s_n^k \text{ is } T_n^i & \text{then } y = y_1 \text{ with } q(i,1) \\ & \text{or y = y}_2\text{with q(i,2)} \\ & \ldots\ldots \\ & \text{or y = y}_\text{m}\text{with q(i,m)} \end{array} \right) \tag{20}$$

where, T_x^i = linguistic value of s_x^k of R_i rule. The membership function is represented by $\alpha_{T_x^i}$. In this study, Takagi-Sugeno FIS has been utilized. The q-values are utilized to locate best possible action among the possible actions (m) by choosing highest q value action of each rule R_i. The most favorable action $y(s^k)$ at situation s^k is computed:

$$y(s^k) = \sum_{i=1}^{N} \chi y_i \Bigg/ \sum_{i=1}^{N} \chi;\ y_i \in Y;\ where,\ \alpha_i(s^k) = \chi \tag{21}$$

where, y_i = best action of rule R_i.

For better rewards, the performer has to go for other optional actions in reinforcement learning (RL) based FQL. The pseudo-stochastic procedure based exploration can be applied, and exploration-exploitation (EEP) chooses random action with minimum probability (ε). Based on EEP, selected action is represented as:

$y_i^\dagger = \varepsilon$-greedy y_i

if y_i^* = maximizing action, then

$q(i, y_i^*) = \max_{b \leq m} q(i, b)$, for a continuous action $y(s^k)$, the q-value is defined as:

$$Q(s^k, y(s^k)) = \sum_{i=1}^{N} \chi q(i, y_i^\dagger) \Big/ \sum_{i=1}^{N} \chi \tag{22}$$

Computed state value is:

$$V(s^k) = \sum_{i=1}^{N} \chi q(i, y_i^*) \Big/ \sum_{i=1}^{N} \chi \tag{23}$$

$y(s^k)$ is implemented to change the stage and/or state for creating a RL signal, which is utilized for evaluation of *temporal difference* (TD) error given by:

$$\Delta Q = r + \gamma V(s^{k+1}) - Q(s^k, y(s^k)) \tag{24}$$

and updated q-values are represented as:

$$q(i, y_i^\dagger) \leftarrow q(i, y_i^\dagger) + \eta \Delta Q \left(\chi \Big/ \sum_{i=1}^{N} \chi\right) \tag{25}$$

where, γ = discount factor in range of $0 \le \gamma < 1$ and η = learning rate

Here, FQL is implemented for classifying the WTGS operating stage, which provide only reinforcement signal which decide the quality of identification. Therefore, optimization of these signals maximizes cumulative rewards received at each stage. This system is made adaptive in nature by creating MFQL.

3) MODIFIED FUZZY Q LEARNING CLASSIFIER (MFQL)

Firstly, prepared input vector s^k (Eq.12) from generated IMFs of EMD with the help of J48 algorithm, which is utilized n FIS.

$$s^k = [IMF_1^k, IMF_2^k, \ldots, IMF_8^k] \tag{26}$$

Now, TSK type rule base is represented as:

$$\begin{aligned} R_i : &\text{ If } s_1^k \text{ is } IMF_1^i \text{ and} \\ &\ldots\ldots \text{ and } s_n^k \text{ is } IMF_8^i \text{ then } y = y_1 \text{ with } q(i,1) \\ &\qquad \text{or } y = y_2 \text{ with } q(i,2) \\ &\qquad \ldots\ldots \\ &\qquad \text{or } y = y_{11} \text{ with } q(i,11) \end{aligned} \tag{27}$$

where, $y_1, y_2, \ldots y_n$ = fault type/number, and s_1^k = crisp value of input variable.

s^k is quantized into 3 fuzzy subsets by using Gaussian membership function as given by Eq.14:

$$\alpha_{IMF}(s_1) = e^{-(s_1-c_1)^2/2\sigma_1^2} \tag{28}$$

where, c_1 = central and σ_1 = standard deviation.

The IMFs for each input variables are represented as:

$$\alpha_{l_p}(s_j) = e^{\frac{-(s_j - s_j^{l_p})}{\sigma_j^2}}; \quad l_p = 1, 2, 3;\ j = 1, \ldots, 8; \tag{29}$$

where l_p = fuzzy labels and s_j = crisp value of MII I_j

The centers of each MF is stipulated:

$$c_j^{l_p} = a_j + b_j(l_p - 1) \tag{30}$$

where, $a_j = s_j^{\min}$, and $b_j = s_j^{\min} + \frac{(s_j^{\max} - s_j^{\min})}{2} \times (l_p - 1)$

The width of MF of each input variable is stipulated:

$$\sigma_j = \frac{(s_j^{\max} - s_j^{\min})}{5} \tag{31}$$

where, $s_j^{\min}$ = minimum value of input variable of I_j, $s_j^{\max}$ = minimum value of input variable of I_j.

Consequently, each fuzzify input variable has three labels ($s_j^{\min}$, $c_j^{l_p}$ and $s_j^{\max}$) to form the total number of fuzzy rules for n number of faults are:

$$R_i = (l_p)^j \times n = (\mathit{fuzzy\ partition})^{(no.of\ \ input)} \times (no.of\ fault) \tag{32}$$

So, as per Eq. (x), fuzzy q-values vector has been initialized in order of $3^8 \times 6$ for 6-type fault analysis by using 8 IMFs. The most optimal action and q-values are obtained for each rule:

$$q^*(i, y) = \max_{y \in Y} q(i, y) \tag{33}$$

$$y_i^* = \arg\max_{y \in Y} q(i, y) \quad \forall i \in N \tag{34}$$

Here, all firing strength value for every action for all rules are aggregated and a vector is produced as:

$$\alpha(y^*) = \alpha(y_i^*) + \alpha(i, y_i^*) \quad \forall i \in N\ \forall y^* \in Y \tag{35}$$

$\forall y^* \in Y_{=}1 \times 6$ vector for 6 type faults.

All the action specific firing strength values are normalized as Eq. 36:

$$\alpha(y^*) = \alpha(y^*) \Big/ \sum_{y^* \in Y} \alpha(y^*) \text{ (Normalized value)} \tag{36}$$

To search optimal action for the discrete selection in the task of identification, which is the continuous action computed by Eq. 24. Here it is released that the task of identification in WTGS has a different concept, therefore an action of highest cumulative firing strength (α), shown as global identifier of Eq. 37, whereas in FQL, a TSK type mechanism is applied to find out the action.

$$y_{Iden}(s^k) = \arg\max_{y \in Y} \alpha(y) \tag{37}$$

The identifier action is based on a max firing strength action selection mechanism for selecting a discrete action, whereas fuzzy Q learning framework employs a simplified TSK type mechanism to generate a continuous action (24).

Here, EEP is applied for getting better action $y_i^\dagger$ as:

$$\begin{aligned} y_i^\dagger &= y_i^r \text{ with probability } \varepsilon \\ &= y_i^* \text{ otherwise} \end{aligned} \tag{38}$$

where, y_i^r = random action and y_i^* = optimal action Computation of global q-value is represented as:

$$Q(s^k, y(s^k)) = \sum_{i=1}^{N} \chi q(i, y_i^\dagger) \Big/ \sum_{i=1}^{N} \chi \tag{39}$$

Evaluation of global target value is performed as:

$$V(s^k) = \sum_{i=1}^{N} \chi q(i, y_i^*) \Big/ \sum_{i=1}^{N} \chi \tag{40}$$

MFQL identifier function for comparison of actual fault $y_{true}(s^k)$ with the identified fault $y_{Iden}(s^k)$.

$$r = \begin{cases} +10 \text{ if } y_{Iden} = y_{true} \\ -10 \text{ if } y_{Iden} \neq y_{true} \end{cases} \tag{41}$$

Now, TD error is computed as:

$$\Delta Q = r + \gamma\, V(s^{k+1}) - Q(s^k, y(s^k)) \tag{42}$$

where, $\gamma \in [0, 1]$ = discount factor.
Update eligibility traces for every rule, are applied having delay parameter with $\xi \in [0, 1]$.

$$\begin{aligned} e(i, y) &\leftarrow e(i, y) \times \gamma \times \xi + \frac{\alpha_i}{\sum \alpha_i} \text{ for } y = y_{Iden} \\ &\leftarrow e(i, y) \times \gamma \times \xi \textit{ for } y \neq y_{Iden} \end{aligned} \tag{43}$$

Further step is upgradation of incremental change in q-value, for upgradation of $q(i, y)$ value:

$$\Delta Q(i) = e(i, y) \times \Delta Q \times \left(\alpha_i \Big/ \sum \alpha_i\right) \tag{44}$$

Updated version of identifier for fuzzy $q(i, a)$ values is:

$$q(i, y_{prev}) \leftarrow q(i, y_{prev}) + \eta[\Delta Q] \tag{45}$$

where, $y_{prev} = y(s^{k-1})$ = elected action by classifier at stage (k-1) and $\eta \in [0, 1]$ = learning rate.

Exploration rate can be decreased as per requirement (i.e., increased the search at the start of procedure and go on decreasing after gained) as:

$$\in \rightarrow \frac{\in}{2} \text{ every 50 samples till } \varepsilon = 0.01 \tag{46}$$

Presented procedure is followed till q-value start converging. The triumph rate of MFQL classifier is evaluated here as:

$$\%\text{ success Rate} = \frac{\text{success}}{\text{success} + \text{failure}} \times 100\% \tag{47}$$

Step-by-step demonstration of the proposed MFQL classifier for a given 5 mechanical faults problem and a healthy condition has been presented in subsequence section.

IV. DEMONSTRATION OF MECHANICAL FAULT DIAGNOSIS USING MFQL

For the experimental demonstration of the performance of the proposed approach, the training and testing data files are prepared first after the pre-processing the data. In this study, ~70% and ~30% datasets are used as a training and testing purpose for the intelligent model respectively. In this section, step-by-step demonstration for validation of proposed MFQL based mechanical fault classifier is presented for WTGS.

Step 1: a rule firing strength vector α_i $\forall i \in N$ as per Eq. 27 is created by mapping input IMFs as given in Table 5.

TABLE 5. Rule firing strength.

Sample	β_i values ($3^8 = 6561$) vector
Healthy	0.0043 0.0000 0.0346 x10^{-4}…0.0000 0.0345 0.2947 x10^{-6}
Fault 1	0.0000 0.0001 0.1921 x10^{-5}…0.0000 0.0023 0.0198 x10^{-3}
Fault 2	0.0035 0.1402 0.0000x10^{-6}…0.0000 0.0000 0.2180x10^{-3}
Fault 3	0.0012 0.0000 0.3306 x10^{-5}…0.0021 0.0000 0.1797 x10^{-5}
Fault 4	0.0000 0.0000 0.3953x10^{-4}…0.0000 0.0000 0.3293 x10^{-6}
Fault 5	0.0183 0.0000 0.3489 x10^{-6}…0.1654 0.0000 0.0000 x10^{-3}

TABLE 6. q-value action y_i^*.

Sample	a_i^* values ($3^8 = 6561$) vector
Healthy	2 4 1 5….2 2 6 1
Fault 1	5 4 3 5… 2 6 1 5
Fault 2	1 4 6 5….3 4 1 2
Fault 3	6 4 2 5….2 5 3 1
Fault 4	5 4 3 5….2 6 1 5
Fault 5	1 5 2 3….4 6 6 1

TABLE 7. Aggregated action of $\alpha(y^*) \forall y \in Y$.

Sample	$\beta(a^*) \forall a \in A$ values (1 x 6) vector (Normalized)					
Healthy	**0.6445**	0.1730	0.0000	0.0000	0.0000	0.1824
Fault 1	0.1193	**0.7444**	0.0001	0.0000	0.0001	0.1361
Fault 2	0.0931	0.0000	**0.7445**	0.0000	0.0002	0.1623
Fault 3	0.0362	0.0000	0.0000	**0.7192**	0.0003	0.2443
Fault 4	0.1035	0.0000	0.0000	0.0001	**0.7445**	0.1519
Fault 5	0.1184	0.0002	0.0000	0.1444	0.0000	**0.7370**

TABLE 8. Aggregated output $V(y^k)$.

Sample	$V(y^k)$ value
Healthy	147.5599
Fault 1	137.9781
Fault 2	142.8714
Fault 3	152.8822
Fault 4	138.6053
Fault 5	145.7546

Step 2: now determine highest q-value action y_i^* according to Eq. 34 as represented in Table 6 for each operating condition.

Step 3: evaluate aggregate action specific firing strength values $\alpha(y^*) \forall y \in Y$ by using Eq. 36 as represented in Table 7.

Step 4: a global target value $V(s^k)$ is produced by mixing output of every rule with collaboration of fuzzy rule fire $\alpha_i > 0$ by using Eq. 40 as represented in Table 8.

Step 5: classify fault condition as per q-value (as maximum cumulative firing strength) by using Eq. 37 as represented in Table 9.

Step 6: As per MFQL, produced value of punishment/reward of each fault condition by using Eq. 41.

Step 7: by using Eq. 42, evaluate the TD error for arranging q-values.

Step 8: by using Eq. 43, update $q(i, y)$ values and $e(i, y)$.

Step 9: by using Eq. 46, shrink exploration rate according to the requirement.

Step 10: reiterate the step1 to 9 till $\Delta q \rightarrow 0$ (converge the q-values to smaller one) to enhance the diagnosis accuracy of WTGS.

TABLE 9. Classifier's output.

Sample	Output $a_{Iden}(y^k)$
Healthy	$a_{Iden}(y^k)$=1 with $\beta(a_1^*)$=**0.6445** value
Fault 1	$a_{Iden}(y^k)$=2 with $\beta(a_2^*)$=**0.7444** value
Fault 2	$a_{Iden}(y^k)$=3 with $\beta(a_3^*)$=**0.7445** value
Fault 3	$a_{Iden}(y^k)$=4 with $\beta(a_4^*)$=**0.7192** value
Fault 4	$a_{Iden}(y^k)$=5 with $\beta(a_5^*)$=**0.7445** value
Fault 5	$a_{Iden}(y^k)$=6 with $\beta(a_6^*)$=**0.7370** value

TABLE 10. WTGS fault diagnosis accuracy analysis.

Model No.	% Accuracy (minimum)	% Accuracy (maximum)	% Accuracy (average)
SVM Model - 08 IMFs	-	-	96.80
ANN Model - 08 IMFs	-	-	94.95
Fuzzy Model - 08 IMFs	-	-	86.4
FQL Model - 08 IMFs	91.02	98.67	94.845
MFQL Model1 - 10 IMFs	80.4	97.5	88.95
MFQL Model2 - 08 IMFs	**98.962**	**99.998**	**99.48**
MFQL Model3 - Voltage Based 15 IMFs	87.25	95.50	91.375

V. RESULTS AND DISCUSSION

The performance analysis of proposed MFQL approach for WTGS mechanical fault diagnosis using selected current signature based IMFs has been presented in Table 10 and also compared its performance with SVM, ANN, Fuzzy, and conventional FQL models. It is analyzed that diagnosis accuracy of MFQL model2 (based on electrical current signature) has utmost than others. Moreover, q-values ($3^8 \times 6$) are lower than other MFQL models i.e., model1 and model3 has $3^{10} \times 6$ and $3^{15} \times 6$ q-values respectively which affect the computational burden of the classifier.

For the further validation of proposed MFQL approach, four different AI techniques (i.e., SVM, ANN, Fuzzy and FQL methods) have been implemented using 8-selected input IMFs and its diagnosis accuracy of WTGS has been listed in Table 10.

A multi-class of 8-21-6 architecture based BP ANN model has been implemented, and a tree based 5-binary SVM models have been designed to classify 5 WTGS mechanical faults and healthy condition with its optimal parameters of $C = 1$ and $\lambda = 1$. A detailed explanation for the implementation of the 5-binary classifier has been given in Appendix-A. Implementation steps of FQL based WTGS fault diagnosis model has been presented in part 2 of subsection 3.3.2. Proposed MFQL model is competent to attain a diagnosis accuracy of 99.48% which shows the superiority than other AI techniques such as ANN, SVM, Fuzzy and FQL achieve 94.95%, 96.80%, 86.4% and 94.85% respectively. Moreover, over-fitting and sluggish processing-speed problem arised in ANN does not present in proposed MFQL based classifier. FQL is not adaptive in nature so it needs to optimize the generated reinforcement signal similar to SVM parameters optimization problem, while MFQL approach do this itself.

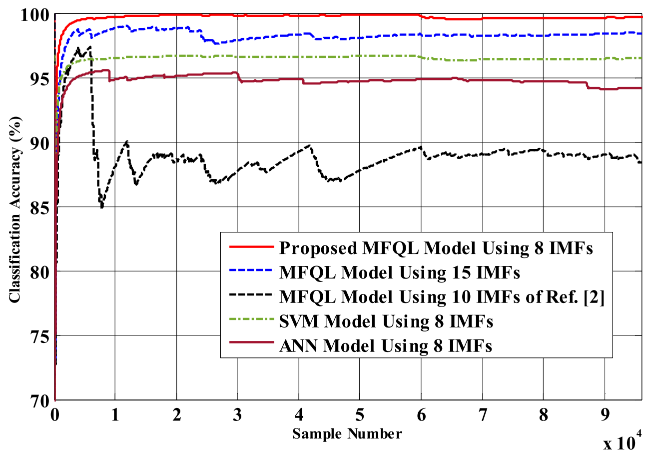

FIGURE 5. Performance learning curves of proposed approach, ANN and SVM.

The performance learning curve for proposed MFQL approach along with ANN, and SVM is represented in Figure 5 which shows the diagnosis accuracy with respect to the number of specimen/samples. The diagnosis accuracy is attained by proposed approach is 98.5%, 99% and 99.998% in just 1534, 2300 and 14300 specimen respectively, which can also be utilized to select minimum number of specimen for higher accuracy.

VI. CONCLUSION

In this study, a novel MFQL classifier for WT IF diagnosis has been presented and demonstrated. Proposed classifier shows a considerably outperformed accuracy than other computational techniques as given in Table 10, which shows its effectiveness in terms of diagnosis accuracy enhancement, reduction in computational burden, self-learning & optimization, adaptive in nature and no overfitting at each level. This is the first attempt in implementation of FQL and proposed MFQL approach for fault diagnosis of WTGS. This novel approach can be applied in future for nonintrusive fault classification of WTGS control without using any mechanical sensors.

APPENDIX-A

SVM BASED MODELLING

According to the selected variables, a tree based binary SVM models have been designed to classify the six operating conditions (5-faulty and one healthy) as shown in Figure 6. Using whole training datasets (i.e., input feature matrix) of six distinct conditions, SVM model#1 is trained to separate the normal operating condition from the five faulty conditions (i.e., RotFurl, TailFurl, BlPitch, AdjBlM and NacYaw). To represent the healthy operation, the output of SVM model#1 is set with -1. Now, SVM model#2 is trained by utilizing faulty datasets (i.e., RotFurl, TailFurl, BlPitch, AdjBlM and NacYaw) to categorize the category#1 faults (i.e., BlPitch, AdjBlM and NacYaw) from the category #2 faults (i.e., RotFurl, and TailFurl). To represent the

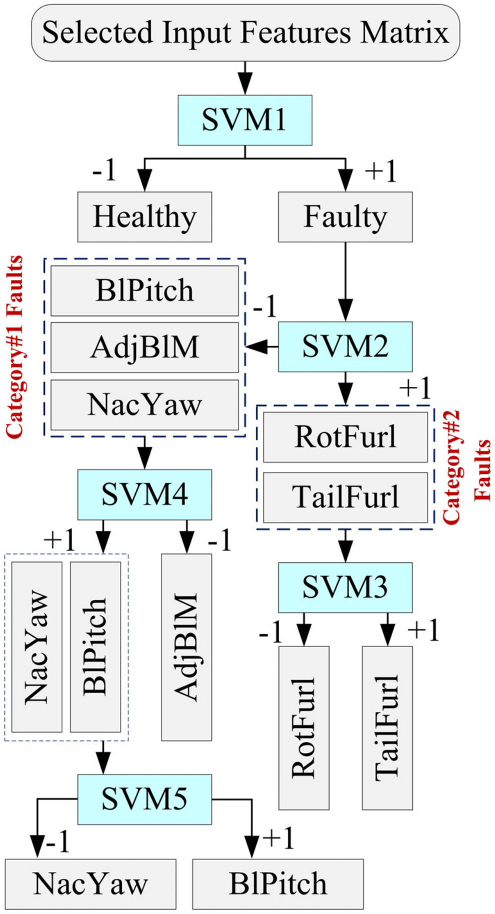

FIGURE 6. SVM based fault diagnosis model implementation.

TABLE 11. Utilized coding to represent the model output.

Fault Category	SVM Model#1	SVM Model#2	SVM Model#3	SVM Model#4	SVM Model#5
Healthy	-1	×	×	×	×
RotFurl	+1	+1	-1	×	×
TailFurl	+1	+1	+1	×	×
NacYaw	+1	-1	×	+1	-1
BlPitch	+1	-1	×	+1	+1
AdjBlM	+1	-1	×	-1	×

×=not in used

"category#1 faults", the output of SVM model#2 is set with −1. Then SVM model#3 is trained by utilizing "(category#2 faults)" datasets to separate "RotFurl" fault and "TailFurl" fault. To represent the "RotFurl" fault condition, the output of SVM model#3 is set with −1. Thereafter, SVM model#4 is trained by utilizing "(category#1 faults)" datasets to segregate "(BlPitch and NacYaw)" fault from "(AdjBlM)" faults. To segregate "AdjBlM" fault condition, the output of SVM model#4 is set with −1. Finally, SVM model#5 is trained by using "(BlPitch and NacYaw)" fault datasets to segregate "NacYaw" fault condition from "BlPitch" condition. To segregate "NacYaw" fault from "BlPitch", output of SVM model#5 is set with −1. Therefore, the complete codification for all fault condition is tabulated in Table 11.

REFERENCES

[1] The Government of India MNRE. (Feb. 2015). *Wind Power Programme Report 2015*. Accessed: Feb. 2015. [Online]. Available: https://mnre.gov.in/knowledge-center/reports

[2] H. Malik and S. Mishra, "Artificial neural network and empirical mode decomposition based imbalance fault diagnosis of wind turbine using TurbSim, FAST and simulink," *IET Renew. Power Gener.*, vol. 11, no. 6, pp. 889–902, May 2017.

[3] J. Ribrant, "Reliability performance and maintenance—A survey of failures in wind power systems," M.S. thesis, KTH School of Electrical Engineering, Stockholm, Sweden, 2006.

[4] S. A. Saleh and C. R. Moloney, "Development and testing of wavelet packet transform-based detector for ice accretion on wind turbines," in *Proc. Digit. Signal Process. Signal Process. Educ. Meeting (DSP/SPE)*, Jan. 2011, pp. 72–77.

[5] J. M. Jonkman and M. L. Buhl, Jr., *FAST User's Guide*. Golden, CO, USA: National Renewable Energy Lab, 2005.

[6] M. Zhao, D. Jiang, and S. Li, "Research on fault mechanism of icing of wind turbine blades," in *Proc. World Non-Grid-Connected Wind Power Energy Conf.*, Sep. 2009, pp. 1–4.

[7] W. Yang, P. J. Tavner, C. J. Crabtree, and M. Wilkinson, "Cost-effective condition monitoring for wind turbines," *IEEE Trans. Ind. Electron.*, vol. 57, no. 1, pp. 263–271, Jan. 2010.

[8] W. Qiao and D. Lu, "A survey on wind turbine condition monitoring and fault diagnosis—Part I: Components and subsystems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6536–6545, Oct. 2015.

[9] W. Qiao and D. Lu, "A survey on wind turbine condition monitoring and fault diagnosis—Part II: Signals and signal processing methods," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6546–6557, Oct. 2015.

[10] X. Gong and W. Qiao, "Imbalance fault detection of direct-drive wind turbines using generator current signals," *IEEE Trans. Energy Convers.*, vol. 27, no. 2, pp. 468–476, Jun. 2012.

[11] X. Jin, W. Qiao, Y. Peng, F. Cheng, and L. Qu, "Quantitative evaluation of wind turbine faults under variable operational conditions," *IEEE Trans. Ind. Appl.*, vol. 52, no. 3, pp. 2061–2069, May 2016.

[12] T. Gerber, N. Martin, and C. Mailhes, "Time-frequency tracking of spectral structures estimated by a data-driven method," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6616–6626, Oct. 2015.

[13] H. Malik and S. Mishra, "Application of LVQ network in fault diagnosis of wind turbine using Turbsim, fast and simulink," in *Proc. MFIIS*, Vol. 2, 2015, pp. 474–480.

[14] W. Yang, P. J. Tavner, and W. Tian, "Wind turbine condition monitoring based on an improved spline-kernelled chirplet transform," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6565–6574, Oct. 2015.

[15] H. Malik and S. Mishra, "Application of GEP to investigate the imbalance faults in direct-drive wind turbine using generator current signals," *IET Renew. Power Gener.*, vol. 11, no. 6, pp. 279–291, Oct. 2017.

[16] J. Hang, J. Zhang, and M. Cheng, "Application of multi-class fuzzy support vector machine classifier for fault diagnosis of wind turbine," *Fuzzy Sets Syst.*, vol. 297, pp. 128–140, Aug. 2016.

[17] X. Gong and W. Qiao, "Current-based mechanical fault detection for direct-drive wind turbines via synchronous sampling and impulse detection," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1693–1702, Mar. 2015.

[18] W. Qiao, X. Yang, and X. Gong, "Wind speed and rotor position sensorless control for direct-drive PMG wind turbines," *IEEE Trans. Ind. Appl.*, vol. 48, no. 1, pp. 3–11, Jan. 2012.

[19] M. Wiering and M. van Otterlo, "Reinforcement learning: State-of-the-art," in *Adaptation, Learning and Optimization*, vol. 12. Berlin, Germany: Springer, 2012.

[20] L. Busoniu, *Reinforcement Learning and Dynamic Programming using Function Approximator*. Boca Raton, FL, USA: CRC Press, 2010.

[21] (Aug. 18, 2014). *NWTC Information Portal (Certification of FAST and ADAMS With AeroDyn)*. Accessed: May 21, 2016. [Online]. Available: https://nwtc.nrel.gov/SimulatorCertification.

[22] B. J. Jonkman, *TurbSim User's Guide (V1.06.00)*. Golden, CO, USA: National Renewable Energy Laboratory, 2012.

[23] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London Ser. A, Math., Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[24] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[25] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *J. Artif. Intell. Res.*, vol. 4, pp. 77–90, Mar. 1996.

[26] B. M. Ebrahimi, M. J. Roshtkhari, J. Faiz, and S. V. Khatami, "Advanced eccentricity fault recognition in permanent magnet synchronous motors using stator current signature analysis," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 2041–2052, Apr. 2014.

[27] S. Simani, S. Farsoni, and P. Castaldi, "Fault diagnosis of a wind turbine benchmark via identified fuzzy models," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3775–3782, Jun. 2015.

[28] Y. Fu, Z. Gao, Y. Liu, A. Zhang, and X. Yin, "Actuator and sensor fault classification for wind turbine systems based on fast Fourier transform and uncorrelated multi-linear principal component analysis techniques," *Processes*, vol. 8, no. 9, p. 1066, Sep. 2020.

[29] Z. Gao and X. Liu, "An overview on fault diagnosis, prognosis and resilient control for wind turbine systems," *Processes*, vol. 9, no. 2, p. 300, Feb. 2021.

HASMAT MALIK (Senior Member, IEEE) received the M.Tech. degree from the National Institute of Technology (NIT) at Hamirpur, India, and the Ph.D. degree from the Indian Institute of Technology (IIT), Delhi, both in electrical engineering.

He served as an Assistant Professor for more than five years with the Division of Instrumentation and Control Engineering, Netaji Subhas Institute of Technology (NSIT) at Delhi, Government of Delhi, India. He is currently a Chartered Engineer (C.Eng.) and a Professional Engineer (P.Eng.). He is also a Research Fellow at BEARS, Singapore. He has supervised 23 PG students. He is involving in several large research and development projects. He has authored/coauthored more than 100 research articles and eight books and thirteen chapters in nine other books, published by IEEE, Springer, and Elsevier. He has published widely in international journals and conferences where his research findings are related to intelligent data analytic, artificial intelligence, and machine learning applications in power systems, power apparatus, smart building and automation, smart grid, forecasting, prediction, and renewable energy sources. His principle area of research interests include artificial intelligence, machine learning and big-data analytics for renewable energy, smart building and automation, condition monitoring, and online fault detection and diagnosis (FDD). He is a Life Member of the Indian Society for Technical Education (ISTE), the Institution of Electronics and Telecommunication Engineering (IETE), and the International Society for Research and Development (ISRD) London, and a member of the Computer Science Teachers Association (CSTA), USA, the Association for Computing Machinery (ACM) EIG, and the Mir Labs, Asia. He received the POSOCO Power System Award (PPSA-2017) for his Ph.D. work for research and innovation in the area of power system, and the best research papers awards at IEEE INDICON-2015, and Full Registration Fee Award at IEEE SSD-2012, Germany. He has organized five international conferences, and proceedings have been published by Springer Nature.

ABDULAZIZ ALMUTAIRI (Member, IEEE) received the B.Sc. degree in electrical engineering from Qassim University, Buraydah, Saudi Arabia, in 2009, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2014 and 2018, respectively. He is currently an Assistant Professor with the Department of Electrical Engineering, Majmaah University, Saudi Arabia. His research involves both experimental studies and modeling of many system problems. His recent research interests include asset management in smart grids, power system reliability and resilience, and development of innovating techniques for integrating renewable energy resources and electric vehicle to power systems.