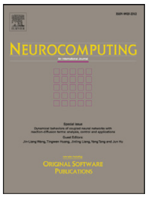
Neurocomputing 359 (2019) 139–152

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



An improved fuzzy ARTMAP and Q-learning agent model for pattern classification

Farhad Pourpanah[a], Ran Wang[a,b,*], Chee Peng Lim[c], Xizhao Wang[d], Manjeevan Seera[e], Choo Jun Tan[f]

[a] *College of Mathematics and Statistics, Shenzhen University, China*
[b] *Shenzhen Key Lab. of Advanced Machine Learning and Applications, Shenzhen University, China*
[c] *Institute for Intelligent Systems Research and Innovation, Deakin University, Australia*
[d] *College of Computer Science and Software Engineering, Guangdong Key Lab. of Intelligent Information Processing, Shenzhen University, China*
[e] *Faculty of Engineering, Computing and Science, Swinburne University of Technology, Sarawak Campus, Malaysia*
[f] *School of Science and Technology, Wawasan Open University, Malaysia*

ARTICLE INFO

Article history:
Received 28 March 2019
Revised 22 May 2019
Accepted 1 June 2019
Available online 4 June 2019

Communicated by Zhaohong Deng

Keywords:
Fuzzy ARTMAP
Multi-agent system
Q-learning
Pattern classification
Motor fault detection
Human motion detection

ABSTRACT

The Fuzzy ARTMAP (FAM) network is an online supervised neural network that operates by computing the similarity level between the new sample and those prototype nodes stored in its network against a threshold. In our previous study, we have developed a multi-agent system consisting of an ensemble of FAM networks and Q-learning, known as QMACS, for data classification. In this paper, an Improved QMACS (IQMACS) model with trust measurement using a combination of Q-learning and Bayesian formalism is proposed. A number of benchmark and real-world problems, i.e., motor fault detection and human motion detection, are conducted to evaluate the effectiveness of IQMACS. Statistical features are extracted from real-world case studies and utilized for classification with IQMACS, QMACS, and their constituents. The experimental results indicate that IQMACS produces better classification performance by combining the outcomes of its constituents as compared with those of QMACS and other related methods.

1. Introduction

Artificial Neural networks (ANNs), including deep learning, are useful techniques that have been successfully applied to many fields such as biomedical imaging [1], diagnosis of electrical power plant [2], prediction [3,4], decision making [5] and human activity recognition [6]. Among them, pattern classification is one of the primary areas that many ANNs have been used, e.g., classification of images [7] and electroencephalogram (EEG) signals [8], as well as fault detection [9].

An ANN can be structured differently to deal with various problems, ranging from small to big and noisy data. However, it is well-known that each individual ANN behaves differently in tackling data with various characteristics, which cover variations in the number of input samples and the respective features in either noisy or noise-free conditions. To effectively handle multi-modality problems, ensemble methods [10] have been introduced. Given a set of data samples, an ensemble model employs a number of individual algorithms to produce multiple predictions, and then combines the predictions to make an overall final decision pertaining to each data sample. It has been shown that ensemble methods are able to achieve better performances than those of individual algorithms.

In our previous work [11], we have proposed an ensemble model consisting of supervised Fuzzy ARTMAP (FAM) [12] and Q-learning [13] methodologies, i.e., Q-learning-based Multi-Agent Classifier System (QMACS), to tackle classification problems. QMACS is able to produce comparable results as those from state-of-the-art methods in tackling both noisy and noise-free problems. Importantly, QMACS includes online learning agents with the capability of learning incrementally, which is able to overcome the stability-plasticity dilemma [12]. In other words, the model is *stable* to remember previously learned samples, and *plastic* to learn new samples.

Q-learning with aggregation (QA-learning) [14], as another Q-learning-based algorithm, is used for cooperative policy construc-

[*] Corresponding author at: College of Mathematics and Statistics, Shenzhen University, China.

E-mail addresses: farhad@szu.edu.cn (F. Pourpanah), wangran@szu.edu.cn, ranwang3-c@my.cityu.edu.hk (R. Wang), chee.lim@deakin.edu.au (C.P. Lim), xizhaowang@ieee.org (X. Wang), msseera@swinburne.edu.my (M. Seera), cjtan@wou.edu.my (C.J. Tan).

https://doi.org/10.1016/j.neucom.2019.06.002
0925-2312/© 2019 Elsevier B.V. All rights reserved.

tion for independent learners. It divides the space problem into a number of sub-regions, and then assigns a number of worker agents to achieve the goals in each sub-region. When the worker agents finish their duties in their corresponding sub-regions, they are re-distributed to other active sub-regions to help other agents to achieve their goals. Different from QA-learning, in our proposed QAMCS, all sibling agents are assigned to perform the same task, which is predicting the target class of the current input sample. The team managers are responsible to select the best sibling agent while the parent agent is responsible to select the best team manager, respectively. Then, the parent agent produces the final prediction pertaining to the current input sample in accordance with the output from the best team manager.

However, QMACS selects both the winning agent and team based on a trust value measured for each agent. In this case, it is possible for QMACS to select the weak agent teams to make decision. In particular, pruned QFAM-based and QFAM-GA [15]-based agent teams, which only retain the prototype nodes with a Q-value larger than a threshold during the pruning procedure, can have trust values higher than those in QFAM [16]-based agent team.

In this paper, we enhance the performance of QMACS by proposing an Improved QMACS (IQMACS) model by adding an additional trust measurement scheme at the team level. We also demonstrate the applicability of IQMACS and its constituents to two problems: (i) human motion detection, and (ii) induction motor fault detection. Both cases have important industrial implications, particularly for the healthcare and manufacturing sectors.

Recognizing and recording human activities has become an important task in many applications, e.g., falls in elderly patients [17] and monitoring the location of Parkinson patients [18]. To this end, many sensing modalities and corresponding algorithms have been proposed to recognize human motions, such as computer vision systems [19], sensor networks [20], wearable sensors [21], and video cameras [22]. Although the conventional methods are able to provide good performance in recognizing human activities, they suffer from several drawbacks. As an example, wearable sensors interfere with users' motion habits, which may compromise recognition accuracy. Video cameras are able to represent visible features of human actions. However, their performance relies on illumination conditions, and image processing procedures usually involve high computational complexity for real-time applications. In addition, some of the methods are not effective for long-term recognition because they need ground truths [23]. On the other hand, smartphones are becoming common and powerful devices with accurate sensing devices such as cameras, GPS sensors and accelerometers. These sensing devices enable smartphones to be used as a flexible and cost-effective tool for many applications, such as localization, tracking, and recognizing human activities [24].

On the other hand, fault detection and diagnosis of induction motors has become an important issue due to their extensive use in diverse industrial applications [25]. Performing online fault detection and diagnosis while the motor is running can dramatically reduce losses [26]. It is crucial that electric motors always operate at their high performance and reliability, and also it is vital to know different types of faults that may happen to induction motors. These faults include: broken rotor bars, misalignment, mechanical and voltage unbalance [27]. Many techniques have been proposed to diagnose these faults, such as mechanical sensors, temperature, X-ray, vibration and current signals [28]. However, most of these techniques are expensive and are not able to diagnose faults online. In addition, they are not able to act as stand-alone equipment, which may cause loading issues on motors. The Motor Current Signature Analysis (MCSA) [29] technique has been extensively used. It has the advantages of low-cost and non-invasiveness by utilizing standard instruments.

In both real-world case studies, the data samples are corrupted with noise at different levels (i.e., 10%, 30% and 50%), in order to evaluate the robustness of IQMACS. The results are compared and quantified statistically using the bootstrap method [30]. To sum up, the main contributions of this research include:

- enhancement of QMACS by incorporating a trust measurement scheme at its team level in order to mitigate the influence of weak agent teams in making the final decision;
- demonstration of the usefulness of IQMACS in real-world case studies, i.e., human motion detection and motor fault detection, which are important for the healthcare and manufacturing sectors.

The rest of this paper is organized as follows. In Section 2, a review on human motion detection and motor fault detection is presented. The structure of Q-learning and FAM-based learning agents are discussed in Section 3. In Section 4, the dynamic of IQMACS is explained in detail. In Section 5, the experimental setup and the associated results are presented. Conclusions and suggestion for further work are given in Section 6.

2. Related work

2.1. Human motion detection

A method for combination of integrated motion images and eigenspace technique to detect human fall was introduced in [31]. In this method, the Multi-Layer perceptron (MLP) was employed to classify and determine the fall situation. The experimental results showed that MLP was an effective method for detection of fall. In [32], a method based on key-frame was proposed to recognize human actions. The Pyramidal Motion Feature (PMF) of each frame was firstly extracted. Then, the AdaBoost learning algorithm was used to select the key frame for each action. Finally, a SVM-based classifier was employed to recognize actions. The experimental results showed that the proposed model outperformed those methods reported in [33,34]. FAM-BSO [35], which is a hybrid model of FAM and brain storm optimization (BSO), was proposed to recognize human motion, i.e., walking, running or jumping. A human recognition system based on pyroelectric infrared sensors and GA-based neural network was proposed in [36].

A human action recognition model was proposed in [37] to solve the problem associated with RGB-D (Red, Green, Blue-Depth) in human action recognition. This model was based on the outputs of Kinect, and contained three parts: feature extraction, feature representation and classification. The results demonstrated that the proposed model significantly improved the robustness of depth features and RGB features based on different actions. In [38], a head detector technique using motion histogram features was presented. A new motion feature, known as the Relative Motion Distance (RMD), was proposed to combine two motion histogram features, i.e., Histogram of Oriented Optical Flow (HOOF) and Motion Boundary Histogram (MBH). In order to distinguish human heads, a two stage SVM-based classifier was used. The results demonstrated the effectiveness of the proposed model in comparison with other methods reported in [39,40]. In [41], a new method using single channel sEMG (surface Electromyography) was proposed to identify the movement of hands. The wavelet transformation and ANN were employed to extract features and classify sEMG signals, respectively. The experimental results indicated the efficiency of ANN in detecting and classifying hand movement.

2.2. Motor fault detection

In [42], a hybrid model of fuzzy Min-Max (FMM) and CART, i.e., FAM-CART, was proposed to detect and recognize faults of a three-phase induction motor. The FMM-CART model tackled both rule extraction and classification problems. The results demonstrated that FMM-CART performed better or similar in comparison with FMM and CART. Later in [43], a hybrid model of FMM and BSO was developed to classify faults of a three-phase induction motor. In [44], a supervised ANN was used to detect motor faults and their locations. Firstly, the significant features were extracted from raw signals, and used to train the ANN. This model was able to produce high classification rates of faults and their locations. In [45], motor current signature analysis technique was used to monitor the motors of an unmanned aerial vehicle (UAV).

An ANN-based approach was proposed to detect faults related to the stator of induction motors [46]. A total of 16 parameters were extracted to train the ANN. The results showed that the ANN with 10 hidden neurons produced the best performance. A technique to detect Inter-Turn Short Circuit (ITSC) faults related to the induction motor stator was presented in [47], where an ANN-based non-invasive heuristic technique was used. This method was able to detect ITSC faults with high accuracy. In [48], an MLP-based network was proposed to detect ITSC faults and unbalanced supply voltage. Fault detection was performed by monitoring the negative sequence voltage and the three-phase shift between the current and phase voltage. The simulation results showed that this model could distinguish ITSC faults and unbalanced voltage accurately.

A fuzzy-based fault detection or diagnosis system was introduced in [49]. The current of stator and timestamp were used as the inputs to the proposed fuzzy system, and the direct torque control (DTC) technique was employed to control the system. The outcome showed the efficiency of this system in detecting faults for the adopted driving cycles. In [50], a two-stage technique based on ANN and wavelet transform was used to detect and classify the power quality (PQ) disturbances in the supply to induction motors. The PQ disturbance features were extracted by applying the discrete wavelet transform technique to sampled current signals. The extracted features were used as inputs to a feed-forward NN for classification of PQ disturbances. The results demonstrate that the proposed model was a powerful network in classifying PQ disturbances.

In [51], three intelligent models, e.g., FAM, MLP and SVM, were used to classify the short-circuit faults in stator winding of induction motors. SVM and MLP performed better in classifying the stator faults. In [52], Naive Bayes, K-Nearest Neighbor (KNN), SVM, MLP and C4.5, were employed to identify the faults of induction motors. The experimental results demonstrated that KNN, MLP and SVM performed better than other techniques.

3. Q-learning and the learning agents

In this section, we introduce some preliminary knowledge for the proposed work. We firstly present the Q-learning algorithm. Then, the structure of the FAM-based learning agents are described in detail.

3.1. Q-learning

Reinforcement learning (RL) [13] is a machine learning method that receives a series of patterns (extracted from data samples) from the environment, selects the best action, and executes it. Then, it utilizes the reinforcement signal to reward or penalize the selected action. RL can be viewed as a class of Markov decision processes (MDPs) that contains four components, i.e., state (S), action (A), transition (T) and reward/reinforcement function (R). Specifically, $S = \{s_0, s_1, \ldots, s_{n-1}\}$ covers all possible states, $A = \{a_0, a_1, \ldots, a_{m-1}\}$ represents all actions for each state, T represents a function that maps a state to an action, i.e., T: $S \times A \rightarrow S$, and R is a reward function (which can be a scalar).

Q-learning [13] is one of the most popular RL algorithms that has been widely used to provide solutions for MDPs. Q-learning uses temporal difference to map state-action pairs into values, i.e., Q-values. It approximates the optimal action value function by learning the action value function (i.e., known as Q-function), as follows:

$$Q(s,a) = r(s,a) + \gamma V^{\pi}(s,a), \tag{1}$$

where $r(s,\ a)$ is the reinforced signal, $V^{\pi}(s,\ a)$ is the value function of action a in state s under policy π, and $\gamma \in [0,\ 1]$ is the discount factor. Q-learning selects an action that has the highest $Q(s,\ a)$ score, i.e.,

$$a^{*} = \underset{a \in A}{\operatorname{argmax}}(Q(s,a)). \tag{2}$$

Once an action is selected, it is executed, and a reward signal is received. Then, it moves to the next state, i.e., s', and $Q(s,\ a)$ is updated as follows:

$$Q(s,a) \leftarrow Q(s,a) + \xi[r + \gamma \max_{a' \in A} Q(s',a) - Q(s,a)], \tag{3}$$

where $\xi \in [0,\ 1]$ is the learning rate.

3.2. Learning agents

In the following sections, the dynamic of individual agents are described.

3.2.1. QFAM

QFAM [16] is a hybrid model of FAM [12] and Q-learning [13] for tackling classification problems. Its operation consists of two phases, i.e., training and test, as follows.

Training phase: QFAM uses the FAM training process to acquire knowledge from the training samples, and assigns a Q-value to each prototype node stored in its network. FAM [12] is a supervised neural network that combines the capability of Adaptive Resonance Theory (ART) with fuzzy set theory in its dynamics. As shown in Fig. 1, FAM contains two fuzzy ART [53] modules, i.e., ART_a and ART_b, for receiving the input and output vectors, respectively, and a map-field, f^{ab}, to map input samples into their corresponding output. Each Fuzzy ART model (ART_a/ART_b) contains three layers: *(i)* pre-processing layer f_0^a (f_0^b), *(ii)* input layer f_1^a (f_1^b), and *(iii)* recognition layer f_2^a (f_2^b). Complement coding [12] is used in the pre-processing layer to avoid the problem of category proliferation. It transfers an M-dimensional input, $a \in [0,\ 1]^M$, into a $2M$-dimensional output, as follows:

$$A = (a_1, \ldots, a_M, 1 - a_1, \ldots, 1 - a_M). \tag{4}$$

ART_a and ART_b receive the complement-coded input vector, A, and its corresponding target, $B = (b,\ 1\text{-}b)$, respectively. The similarity score among current input A and the jth prototype node in f_2^a layer is computed based on following choice function:

$$T_j = \frac{|A \wedge W_j^a|}{\alpha + |W_j^a|}, \tag{5}$$

where $\alpha > 0$ is the choice parameter, and $W_j^a \equiv (w_{j,1}^a, \ldots, w_{j,2M}^a)$ is the weight vector of the jth node in f_2^a. Initially, $W_{j,1}^a(0) = \cdots = W_{j,2M}^a(0) = 1$ and $Q(j) = 0$ is the Q-value of the jth prototype node, and $\wedge$ indicates the fuzzy *and* operator [54]:

$$(u \wedge v)_i \equiv \min(u_i, v_i), \tag{6}$$

where u_i and v_i are the ith dimension of A and W_j^a, respectively. The node with the highest choice value in f_2^a is chosen as the winning node, as follows:

$$T_J = \max\{T_j : j = 1, 2, \ldots, N\}, \tag{7}$$

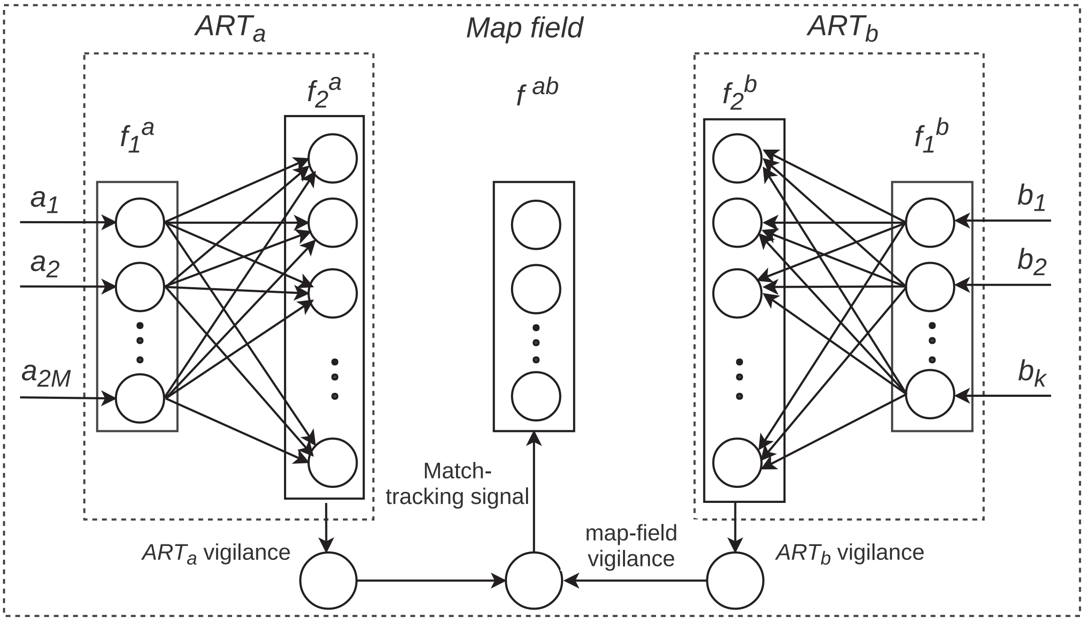


Fig. 1. The structure of FAM.

where J is the index of the winning node, and N is the total number of prototype nodes in f_2^a. Resonance is said to occur for the selected prototype node if the following vigilance test is satisfied:

$$vig(J) = \frac{|A \wedge W_J^a|}{|A|} > \rho_a, \tag{8}$$

where ρ_a is the vigilance parameter of ART_a. However, if the condition in Eq. (8) failed, the winning prototype node is de-activated, and a new search cycle is triggered to select a new winning node. If none of the existing prototype nodes is able to satisfy the condition in Eq. (8), a new prototype node in f_2^a is created to absorb the current input sample. Simultaneously, the same procedure takes place in ART_b. Once the vigilance tests in both ART_a and ART_b are satisfied and the wining nodes are identified, the map-field vigilance test is applied, as follows:

$$\frac{|y^b \wedge W_J^{ab}|}{|y^b|} > \rho_{ab}, \tag{9}$$

where $\rho_{ab} \in [0, 1]$ is the map-field vigilance parameter, W_J^{ab} is the weight vector from f_2^a to f^{ab}, and y^b is the output vector of f_2^b. If the winning node in ART_b is denoted by K, then:

$$y^b = \begin{cases} 1, & k = K \\ 0, & otherwise \end{cases}. \tag{10}$$

If the condition in Eq. (9) is not fulfilled, a match-tracking happens. This means that the current wining node makes an incorrect prediction. In this case, the vigilance value of ART_a is increased to cause the ART_a vigilance test to fail, as follows:

$$\rho_a = \frac{|A \wedge W_J^a|}{|A|} + \delta, \tag{11}$$

where $\delta > 0$ is a small positive value.

And the Q-value of the winning node is updated according to:

$$Q(j)_t = Q(j)_{t-1} + \xi[r(j)_t + \gamma vig(j)_t], \tag{12}$$

where $vig(j)$ denotes the vigilance value of the jth winning node in f_2^a layer (i.e., Eq. (8)), $\xi \in [0, 1]$ and $\gamma \in [0, 1]$ are the learning rate and discount factor, respectively, and $r(j)$ is the reinforcement signal [16]:

$$r(j) = \begin{cases} 1, & if\ learning\ happens \\ 0, & if\ match-tracking\ happens \end{cases}. \tag{13}$$

This search cycle continues until the condition in Eq. (9) is satisfied. Then, the winning node in ART_a is updated to:

$$W_J^{a(new)} = \beta_a(A \wedge W_J^{a(old)}) + (1 - \beta_a)W_J^{a(old)}, \tag{14}$$

where β_a is the learning rate of ART_a.

Note that the Q-value of the winning node is updated in two scenarios, i.e., match-tracking and learning. A summary of QFAM training phase is shown in Algorithm 1.

Test Phase: to predict the target class of a test sample, QFAM first computes the choice values (Eq. (5)) among current input and all prototype nodes in f_2^a, and selects those prototype nodes that are able to satisfy the vigilance test (Eq. (8)). Then, it computes the strength of those selected f_2^a nodes as follows [16]:

$$strength(j) = \lambda T(j) + (1 - \lambda)Q(j), \tag{15}$$

where $\lambda \in [0, 1]$ is a weighting factor. Finally, it selects the node with the highest strength as the winner to predict the target label of the current input. Algorithm 2 summarizes the test phase of QFAM.

3.2.2. The pruned QFAM model

As stated in the previous section, QFAM assigns a Q-value to each prototype node in f_2^a. This Q-value indicates the “goodness” of the corresponding prototype node. In this regard, the network complexity of QFAM can be reduced by pruning less informative prototype nodes. As such, prototype nodes in f_2^a with Q-values less than a user-defined threshold are removed. This network is known as the Pruned-QFAM model.

3.2.3. The QFAM-GA model

QFAM-GA is a hybrid model of QFAM and GA for data classification and rule extraction [15]. Increasing classification accuracy and reducing the network complexity are two important objectives. GA is used to select an optimal subset of features [55]. The GA fitness function is formulated as follows:

$$maximize f(s) = W_{NCP}.NCP(s) - W_s.|S|, \tag{16}$$

Algorithm 1 The learning phase of QFAM.

```
Input: Parameters of QFAM and training samples.
Output: Parameters of trained QFAM.
 1: for each training sample do
 2:     Perform complement-coding (Eq. (4));
 3:     Determine choice value (Eq. (5)) for all prototype nodes in $f_2^a$, and select the node with the highest choice value as winning node;
 4:     Perform the vigilance test (Eq. (8));
 5:     if the vigilance test fails then
 6:         Deactivate the winning node, and go to Step 3 to select new winning node.
 7:     end if
 8:     (Simultaneously, repeat Steps 2–7 in $ART_b$, in order to select winning node ($K$th node in $ART_b$)).
 9:     Perform map-filed vigilance test (Eq. (9));
10:     if map-field vigilance test fails then
11:         Update Q-value of the winning node in $f_2^a$ (Eqs. (12) and (13));
12:         Perform match-tracking (Eq. (11)).
13:         if there is no node in $f_2^a$ satisfying Eq. (11) then
14:             Add a new node;
15:             Go to Step 3.
16:         end if
17:     else
18:         Update Q-value of the winning node in $f_2^a$ (Eqs. (12) and (13));
19:         Update the winning node in $f_2^a$ (Eq. (14)).
20:     end if
21: end for
```

Algorithm 2 QFAM testing phase.

```
Input: Parameters of the trained QFAM and test samples.
Output: Performance indicator.
1: for each test sample do
2:     Perform complement-coding (Eq. (4));
3:     Determine choice value (Eq. (5)) for all prototype nodes in $f_2^a$;
4:     Perform the vigilance test (Eq. (8)), and select those nodes which satisfy the vigilance test;
5:     Compute the strength of the selected nodes (Eq. (15));
6:     Select the node with the highest strength as the wining node;
7:     Update the performance indicator.
8: end for
```

where $NCP(s)$ denotes the number of samples which have been classified correctly, $|S|$ is the number of selected features and $0 < W_s \ll W_{NCP}$. More details can be referred to [15].

4. The proposed IQMACS

Fig. 2 depicts the structure of IQMACS. It consists of three layers. The agent layer, which is placed in the bottom layer, consists of a number of individual agents. Each agent makes decision based on its trained model. The team managers, which are located in the middle layer, are responsible to select the best agent of their corresponding teams. The parent layer is responsible to make final decision by combining the outcome of its team agents. In this study, IQMACS consists of three teams, and each team includes three agents. Specifically, team 1 includes three QFAM [16] models, team 2 includes three pruned QFAM [15] models, while team 3 includes three QFAM-GA [15] models.

To build an IQMACS model, the data set is split into three sub-sets, i.e., learning, validation, and test samples. All agents are trained using the learning samples, and evaluated using the test. The validation samples are used to compute the confusion matrix of each agent, as follows [56]:

$$CM^k = \begin{bmatrix} n_{11}^k & n_{12}^k & \cdots & n_{1(M+1)}^k \\ n_{21}^k & n_{22}^k & \cdots & n_{2(M+1)}^k \\ \vdots & \vdots & \ddots & \vdots \\ n_{M1}^k & n_{M2}^k & \cdots & n_{M(M+1)}^k \end{bmatrix}, \tag{17}$$

where n_{ij}^k, $i = 1, \ldots, M$, and $j = 1, \ldots, M+1$ indicates the total number of samples that belong to class i while have been predicted as class j by agent k. Note that $M + 1$ represents those samples that could not be classified by the corresponding agent.

The confusion matrix can be used to calculate the uncertainty of proposition, i.e. $e_k(x) = j$ (x is predicted as class j by agent k), as follows [56]:

$$p(x \in C_i|e_k(x) = j)) = \frac{n_{ij}^k}{n_j^k} = \frac{n_{ij}^k}{\sum_{i=1}^M n_{ij}^k}, i = 1, \ldots, M. \tag{18}$$

The confusion matrix supports all the target classes, and for a classification algorithm the belief function can be formulated as [57]:

$$bel(x \in C_i|e_k(x) = j_k, EN) = P(x \in C_j|e_k(x) = j_k), i = 1, \ldots, M. \tag{19}$$

where EN indicates the environment in which all agents are active. The belief functions are propagated and used in accordance with the Bayesian formalism. Therefore, Eq. (18) can be formulated as [56]:

$$\begin{aligned} bel(i) &= bel(x \in C_i|e_1(x) = j_1, \ldots, e_k(x) = j_k, EN) \\ &= P(x \in C_i|e_1(x) = j_1, \ldots, j_k, EN) \\ &= \frac{P(e_1(x) = j_1, \ldots, e_k(x) = j_k, EN|x \in C_i, EN)P(x \in C_i|EN)}{P(e_1(x) = j_1, \ldots, e_k(x) = j_k|EN)}. \end{aligned} \tag{20}$$

To simplify Eq. (20), it is considered that all K agents operate independently in EN. As such,

$$\begin{aligned} &\frac{P(e_1(x) = j_1, \ldots, e_k(x) = j_k, EN|x \in C_i, EN)P(x \in C_i|EN)}{P(e_1(x) = j_1, \ldots, e_k(x) = j_k|EN)} \\ &= \frac{\prod_{k=1}^K P(e_k(x) = j_k, EN|x \in C_i, EN)}{\prod_{k=1}^K P(e_k(x) = j_k|EN)}. \end{aligned} \tag{21}$$

By applying Bayes rule:

$$\frac{P(e_k(x) = j_k, EN|x \in C_i, EN)}{P(e_k(x) = j_k|EN)} = \frac{P(x \in C_i e_k(x) = j_k, EN)}{P(x \in C_i|EN)}, \tag{22}$$

Eq. (22) is updated to:

$$\begin{aligned} &\frac{\prod_{k=1}^K P(e_k(x) = j_k, EN|x \in C_i, EN)}{\prod_{k=1}^K P(e_k(x) = j_k|EN)} \\ &= \frac{\prod_{k=1}^K P(x \in C_i|e_k(x) = j_k, EN)}{\prod P(x \in C_i|EN)}. \end{aligned} \tag{23}$$

By replacing Eqs. (23) and (22) into Eq. (21):

$$bel(i) = \frac{\prod_{k=1}^K P(x \in C_i|e_k(x) = j_k, EN)}{\prod P(x \in C_i|EN)} P(x \in C_i|EN). \tag{24}$$

To approximate the combined belief function of K classifiers, the independence assumption is used [58], i.e.,

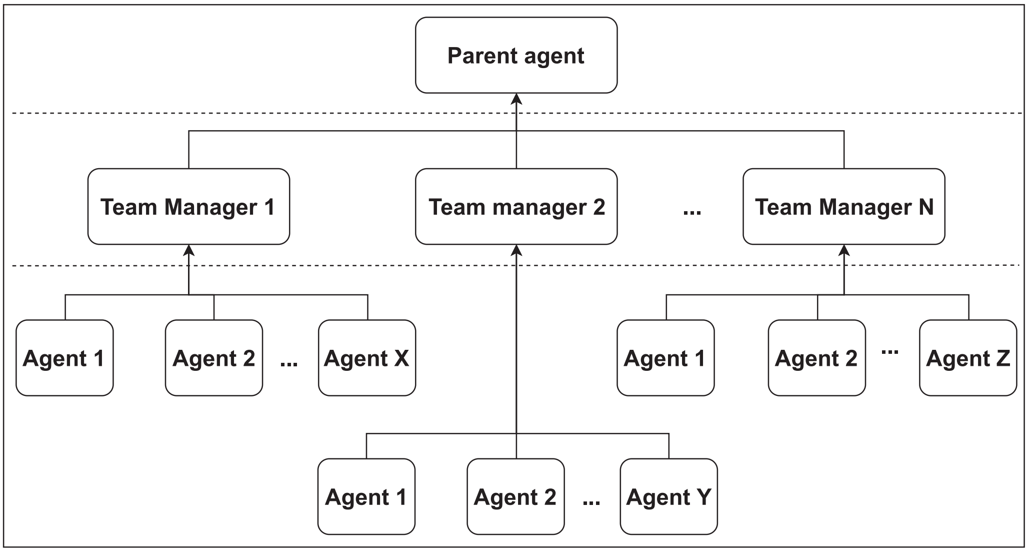


Fig. 2. The IQMACS model contains three layers of agents: parent, team, and sibling.

$$bel(i) = \frac{\prod_{k=1}^{K} P(x \in C_i | e_k(x) = j_k, EN)}{\sum_{i=1}^{M} \prod P(x \in C_i | e_k(x) = j_k)}. \tag{25}$$

where $P(x \in C_i|e_k(x) = j_k)$ can be calculated using Eq. (18). The belief function resulted from integrating K agents can be used to compute the initial trust value of each agent, as follows:

$$initial_{trust}(i) = bel(i) = \frac{\prod_{k=1}^{K} P(x \in C_i | e_k(x) = j_k)}{\sum_{i=1}^{M} \prod_{k=1}^{K} P(x \in C_i | e_k(x) = j_k)}. \tag{26}$$

After calculating the initial trust, the trust of each agent is updated based on prediction of the agent for each test sample as follows:

$$trust_{agent}^{k}(i) = trust(i-1) + \xi(r_{i-1}^{k} + \gamma(Q - value_{C_m}^{k})), \tag{27}$$

where $\gamma \in [0, 1]$ and $\xi \in [0, 1]$ are the discount factor and learning rate, respectively, $trust_{agent}^{k}(i)$ indicates trust value of the ith test sample measured by agent k, and $Q - value_{C_m}^{k}$ represents the mean Q-value of all prototype nodes that belong to class m of agent k [11]:

$$Q_{Value_{C_m}^{k}} = \frac{1}{Z} \sum_{z=0}^{Z} Q_{C_{mz}}, \tag{28}$$

where Z and $Q_{C_{mz}}$ (Eq. (12)) indicate the number of prototypes in f_2^a and the Q-value of the zth prototype that belong to class m, respectively, and r_i^k is the reinforcement signal generated by agent k for the ith input as follows:

$$r(i)^k = \begin{cases} 1, & correct\ \ prediction\ \ by\ \ agent\ \ at\ \ i-1 \\ -1, & incorrect\ \ prediction\ \ by\ \ agent\ \ at\ \ i-1 \end{cases}, \tag{29}$$

where the initial setting of $r_i^k(t = 0)$ is 0.

In our previous work, the QMACS model uses the *sealed bid* $-$ *first price auction* method to select both the winning agent and team using Eq. (27). This can affect the performance of QMACS negatively. As an example, consider two different agents, i.e., pruned QFAM and QFAM. Given an input sample, both agents correctly predict the target class. However, the pruned QFAM agent normally has a higher trust value, because pruned QFAM retains only prototype nodes with Q-values higher than a threshold (Eq. (28)), which provides a higher trust value as compared with that of QFAM. To exploit this situation, IQMACS adds another trust measurement scheme (similar to Eq. (27)) in the team level to compute the trust of each agent team, as follows:

$$trust_{team}^{m}(i) = trust_{team}^{m}(i-1) + \xi(r_{i-1}^{m} + \gamma(Q - value_{C_m}^{m})), \tag{30}$$

where $trust_{team}^{m}(i)$ is the trust value of the mth team for the ith sample, $Q - Value_{C_m}^{m}$ is the Q-value of the predicted class by agent m, r_i^m is the reinforced signal of the mth team, which is updated as follows:

$$r_i^m = \begin{cases} 1, & correct\ \ prediction\ \ by\ \ team\ \ at\ \ i-1 \\ -1, & incorrect\ \ prediction\ \ by\ \ team\ \ at\ \ (i-1) \end{cases}. \tag{31}$$

The initial trust value of each team is as follows:

$$in - trust - team(m) = \frac{1}{L} \sum_{l=1}^{L} initial_{trust}^{l}, \tag{32}$$

where L is the total number of agents in team m, and $initial_{trust}^{l}$ (Eq. (26)) is the initial trust of the l-th agent. The *sealed bid-first price auction* method is used to select the best team at parent level. A summary of IQMCAS is shown in Algorithm 3.

5. Experimental studies

To evaluate the applicability of IQMACS, six benchmark problems from UCI machine learning repository [59] (Table 1), and two real-world case studies, i.e., human motion detection and motor fault detection, were selected for experiment. Firstly, the UCI data sets were used to compare the performance of IQMACS with those of QMACS and other state-of-the-art voting and ensemble methods reported in [60–62]. Then, the real-world case studies were used to compare the performance of IQMACS with those of QMACS and their agent teams. The experimental parameters after several trials were set according to those in Table 2.

In this study, three performance indicators, including accuracy (ACC), sensitivity (SENS), and specificity (SPEC), were used for performance evaluation. SENS/SPEC is the ratio of correctly classified positive/negative samples over the total number of positive/negative samples [63]. Note that SENS and SPEC can be applied to two-class classification problems. The bootstrap method [30] was used to quantify results statistically.

5.1. UCI data sets

In this section, six benchmark problems (four biomedical data samples, i.e., PID, Bupa, WDBC and WBC, and two artificially generated data sets, i.e., waveform and LED) from the UCI machine

Algorithm 3 The Improved QMACS.

```
Input: The parameters of agents and data samples.
Output: Performance indicator.
Split data samples into three subsets, i.e., learning, validation
  and testing samples.
for each agent do
    Train agent using learning samples;
    Update confusion matrix (Eq. (17)) using validation samples;
    Determine initial trust of each agent (Eq. (26)).
end for
for each team do
    Determine initial trust of each team (Eq. (32)).
end for
for each test sample do
    Predict target class by each agent;
    Update trust value of the agent (Eq. (27));
    Use the sealed bid-first price auction to select the best agent
      at team level;
    Update trust of each team (Eq. (30));
    Use the sealed bid-first price auction to select the best team
      at parent level;
    Update the performance indicator.
end for
```

Table 1
Details of UCI data sets.

Data set	# of samples	# of features	# of classes
Waveform Database Generator (Version 2)	5000	40	3
LED display domain	1000	24	10
Pima Indian Diabetes (PID)	768	8	2
BUPA	345	6	2
WDBC	569	32	2
WBC	699	10	2

Table 2
The experimental parameters.

$\beta_a = \beta_b$	1	γ	0.3	Mutation probability	0.1
$\alpha_a = \alpha_b$	0.001	ξ	0.3	Crossover probability	0.9
$\rho_b = \rho_{ab}$	1	W_{NCP}	0.1	Population size	100
ρ_a	0.9	W_s	0.01	Stopping condition	1000
λ	0.95	Pruning threshold	0.3	Number of prototypes replaced in each population	20

Table 4
The ACC rates with standard deviations (SD) for the waveform data set.

Method	Waveform
Locally weighted naive Bayes (LWNB) $K = 50$ [61]	81.88 ± 1.8
Locally weighted naive Bayes (LWNB) $K = 30$ [61]	81.79 ± 1.9
Locally weighted naive Bayes (LWNB) $K = 100$[61]	82.34 ± 1.8
Naive Bayes (NB) [61]	80.01 ± 1.4
K-nearest neighbours with distance weighting (KNNDW) $K = 5$ [61]	79.33 ± 1.8
K-nearest neighbours with distance weighting (KNNDW) $K = 10$ [61]	81.12 ± 2.0
K-nearest neighbours with distance weighting (KNNDW) $K = 20$ [61]	83.12 ± 1.9
QMACS [11]	92.98 ± 2.0
IQMACS	**93.85** ± **1.9**

learning repository were used. In the first experiment, the biomedical data sets were employed. Following the same procedure in [60], the 10-fold cross validation was repeated 10 times (with 90% for training (80% and 10% for learning and validation, respectively), and the remaining 10% for test.

Table 3 shows the ACC, SENS and SPEC rates of IQMACS, QMACS and other methods reported in [60], i.e., Bagging, Accuracy-Weighting, HM-BagMoov, AdaBoot and Majority voting. In term of accuracy, IQMACS outperformed all methods for PID, Bupa and WBDC, while for WBC, both QMACS and IQMACS produced inferior results. In term of SENS, IQMACS outperformed QMACS and other methods in [60]. In term of SPEC, IQMACS recorded the highest rate only for WBDC.

In the second experiment, the effectiveness of IQMACS was evaluated using two noisy data sets, i.e., Waveform and LED. The same experimental procedure in [61,62] was followed. For the Waveform problem, the 10-fold cross-validation was repeated 10 times. As shown in Table 4, IQMACS yielded the highest accuracy rate as compared with those of QMACS and other methods in [61].

For the LED problem, two-third of the data samples were used for training (with 56.7% for learning and 10% for validation), and the remaining for test. Table 5 shows the error rates of IQMACS, QMACS and other methods in [62] with varying noise levels, i.e., 0% (noise-free), 10% and 25%. While, IQMACS produced inferior results for the noise-free case, it outperformed QMACS and other methods in [62] for both noisy cases.

5.2. Real-world case studies

In this section, two case-studies were conducted, i.e., human motion detection and motor fault detection. Nine time domain

Table 3
The ACC, SENS and SPEC rates for biomedical data sets.

Method	PID			BUPA		
	Acc (%)	Sens (%)	Spec (%)	Acc (%)	Sens (%)	Spec (%)
HM-BagMoov [60]	78.21	78.65	**92.60**	70.16	68.71	89.52
Bagging [60]	77.99	75.96	85.00	69.02	66.72	**92.90**
AdaBoot [60]	76.43	52.99	89.00	68.41	53.79	79.00
Majority voting [60]	76.30	50.00	90.40	71.88	45.52	91.00
Accuracy-Weighting [60]	77.00	65.54	85.55	67.01	59.99	75.98
QMACS [11]	91.13	**96.31**	80.33	81.12	77.29	83.39
IQMACS	**92.03**	**96.31**	83.34	**81.64**	**79.40**	83.39
	WBDC			WBC		
HM-BagMoov [60]	95.56	95.00	99.16	97.11	95.78	95.85
Bagging [60]	94.19	94.60	95.04	**97.57**	94.13	95.08
AdaBoot [60]	96.13	94.34	97.20	95.85	92.95	**97.37**
Majority voting [60]	95.61	89.15	99.44	96.71	97.38	95.44
Accuracy-Weighting [60]	96.01	94.23	98.5	96.98	93.28	95.01
QMACS [11]	97.53	94.25	**99.25**	93.50	**100**	81.51
IQMACS	**97.74**	**95.96**	**99.25**	95.70	**100**	85.78

Table 5
The error rates for the LED data set with different level of noise.

Method	level of noise		
	0%	10%	25%
Plurality (Pl) [62]	**0**	0.3123	0.6937
Anti-plurality (Anti-Pl) [62]	**0**	0.5435	0.7988
Borda Count (BC) [62]	**0**	0.3063	0.7087
Plurality with Elimination (PL-Elm) [62]	**0**	0.3123	0.6907
Pairwise Comparisons (PC) [62]	**0**	0.3123	0.6877
QMACS [11]	0.0454	0.2146	0.4128
IQMACS	0.0462	**0.2126**	**0.3945**

Table 6
Details of human motion data set.

Data set	Extracted features	Data samples
Belt pocket	27	152
Front pocket	27	102
Shirt pocket	27	136

statistical features, e.g., mean, Crest factor, root mean square, Skewness, standard deviation, Impulse factor, Latitude factor, Kurtosis and Crest factor and Shape factor, were extracted from both data sets. These features were selected based on the study in [64]. The data samples were normalized between 0 and 1. In order to evaluate the effectiveness of IQMACS, QMACS and their team agents, varying levels of noise were injected to the data samples.

5.2.1. Human motion detection

An accelerometer embedded in a smartphone was used to collect the raw data samples [65]. Three data sets were established with respect to the locations of smartphone: in the front, belt, and shirt pockets of a human. After recording three activities, namely climbing, walking, and running, statistical features were extracted from the raw waveform signals, and normalized between 0 and 1. The task was to classify human's motion, whether climbing, walking or running. Table 6 shows the details of the recorded data.

Three experiments were conducted. In the first experiment, the performance of individual agents was evaluated. For each agent, the 10-fold cross validation was repeated 10 times. A total of 90% and 10% of data samples were used for learning and test, respectively. The mean ACC rates with 95% confidence intervals are shown in Table 7. As can be seen, team 3 outperformed teams 1 and 2. While, team 2 was not able to perform as good as team 1, pruning reduced the network complexity (as shown in Table 8).

In the second experiment, 90% of data samples were used for training (80% for learning and 10% for validation) and 10% for test. The mean ACC rates with 95% confidence intervals are shown in Table 9. Team 3 outperformed two other teams for all data sets. Team 2 was not able to achieve similar accuracy rates to those of teams 1 and 3, owing to pruning. In general, both IQMACS and QMACS could produce better or similar results as compared with those of the best team by combining the outcomes of three teams. In addition, IQMACS outperformed QMACS in term of the mean ACC rate.

In the third experiment, the performance of IQMACS, QMACS and their team agents were compared in terms of ACC, SENS and SPEC. All data samples were combined into a two-class problem, i.e., data from climbing and walking were considered as one class. Therefore, a total of 390 samples with 27 statistical features belonging to two classes, i.e., running and walking, were used. In addition, noise at different levels, i.e., 10%, 30% and 50%, were injected to the class labels of the learning samples. The 10-fold cross validation was repeated 10 times, with 90% of data samples for training (80% and 10% for learning and validation, respectively) and 10% for test. Fig. 3 shows the ACC rates. The ACC rates of all models reduced when the level of noise increased. Teams 1 and 3 achieved similar ACC rates for noise-free data. Team 1 produced similar results to that of team 2 when the level of noise increased. Although both IQMACS and QMACS models produced comparable results by combining the decisions of their team agents, IQMACS outperformed QMACS.

The numbers of prototype nodes before and after pruning are shown in Fig. 4. As can be seen, the number of prototype nodes before pruning (team 1) increased steadily in line with increasing noise level, while teams 2 and 3 approximately used the same numbers of prototype nodes after pruning. Team 2 produced inferior results but a less complex network.

Figs. 5 and 6 show the SENS and SPEC rates of IQMACS, QMACS, and their team agents, respectively. Similar to ACC, both SENS and SPEC rates degraded with increasing noise level. Team 3 outperformed the other two teams in terms of SENS and SPEC rates.

Table 7
The ACC rates (%) of individual agents with 95% confidence intervals for human motion detection.

Location of camera	Method	Lower bound of the 95% confidence interval	Mean	Upper bound of the 95% confidence interval
Belt pocket	QFAM	94.35	95.39	96.29
	Pruned QFAM	85.53	87.80	89.56
	QFAM-GA	**97.47**	**98.02**	**98.55**
Front pocket	QFAM	93.04	94.21	95.31
	Pruned QFAM	82.16	85.62	87.47
	QFAM-GA	**97.09**	**98.04**	**98.90**
Shirt pocket	QFAM	89.42	91.82	92.66
	Pruned QFAM	80.79	84.52	86.65
	QFAM-GA	**96.66**	**97.52**	**98.09**

Table 8
The number of prototypes before and after pruning with 95% confidence intervals.

Location of camera	Method	Lower bound of the 95% confidence interval	Mean	Upper bound of the 95% confidence interval
Belt pocket	Before pruning	61.81	63.93	65.51
	After pruning	19.03	20.29	21.37
Front pocket	Before pruning	42.44	45.41	46.99
	After pruning	14.25	16.72	20.14
Shirt pocket	Before pruning	49.71	51.11	52.24
	After pruning	14.87	17.26	19.03

Table 9
The ACC rates (%) of IQMACS, QMACS and their team agents with 95% confidence intervals for human motion detection.

Location of camera	Method	Lower bound of the 95% confidence interval	Mean	Upper bound of the 95% confidence interval
Belt pocket	Team 1 (QFAM)	89.90	92.63	93.94
	Team 2 (Pruned QFAM)	84.19	86.64	88.69
	Team 3 (QFAM-GA)	93.40	95.16	**96.95**
	QMACS	93.68	94.22	95.75
	IQMACS	**95.67**	**96.18**	96.74
Front pocket	Team 1 (QFAM)	91.76	94.44	96.27
	Team 2 (Pruned QFAM)	69.51	76.24	82.02
	Team 3 (QFAM-GA)	**96.51**	**97.31**	**97.80**
	QMACS	94.18	95.16	96.58
	IQMACS	96.16	97.13	97.56
Shirt Pocket	Team 1 (QFAM)	89.21	94.17	95.62
	Team 2 (Pruned QFAM)	81.19	85.73	88.45
	Team 3 (QFAM-GA)	**95.77**	**96.61**	**97.17**
	QMACS	94.41	95.47	96.01
	IQMACS	95.67	96.45	97.02

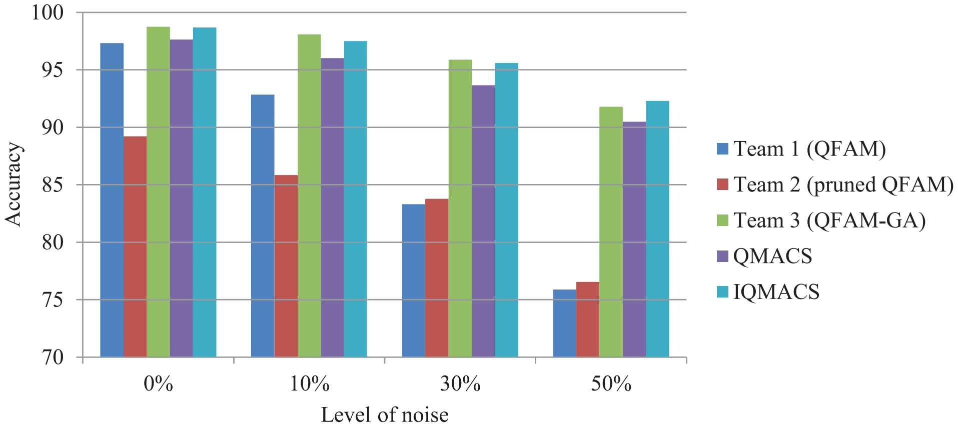


Fig. 3. The ACC rates of IQMACS, QMACS and their team agents for human motion detection in the presence of noise.

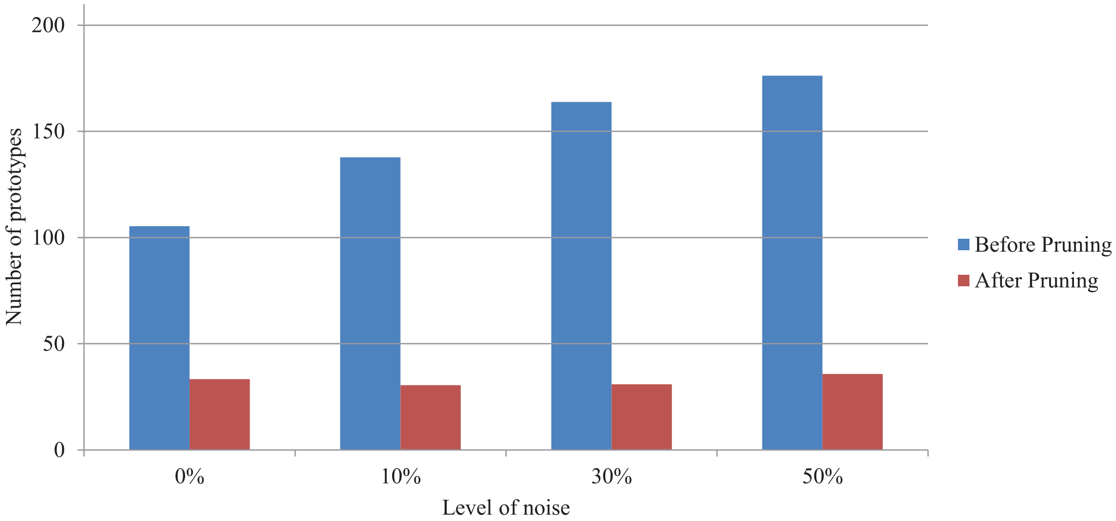


Fig. 4. The number of prototypes before and after pruning in the presence different levels of noise.

5.2.2. *Motor fault detection*

In this case study, fault detection of three-phase electric motors was studied. Current signals of electric motors were recorded under different conditions, i.e., normal, broken rotor bars, eccentricity, unbalanced voltages, and stator winding faults [42]. A total of 50 data samples were recorded. In order to better represent data samples, nine statistical features were extracted from each phase of data samples. Therefore, each data sample was represented by 27 features (3 phases × 9 features).

Firstly, the performance of individual agents was examined with the use of 10-fold cross validation (90% and 10% of data samples for learning and test, respectively) repeated 10 times. Table 10 shows the overall results. QFAM and QFAM-GA achieved similar results statistically, with an overlap between their 95%

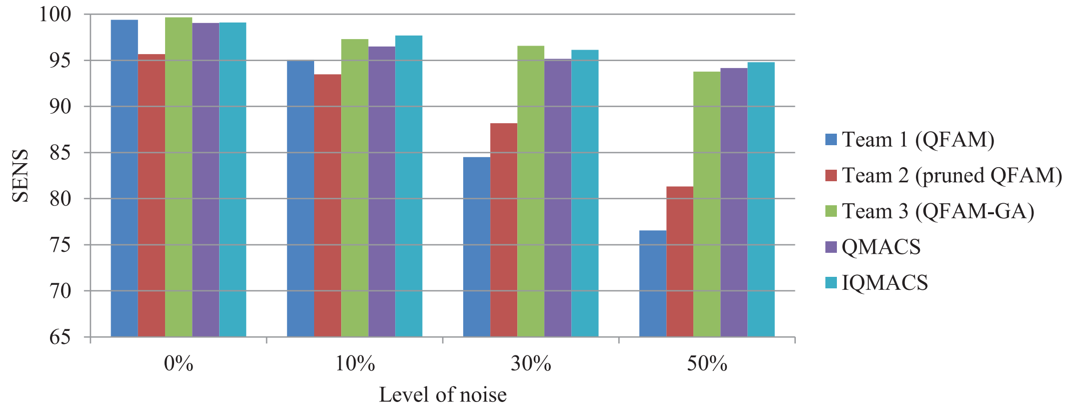


Fig. 5. The SENS rates of IQMACS, QMACS and their team agents for human motion detection.

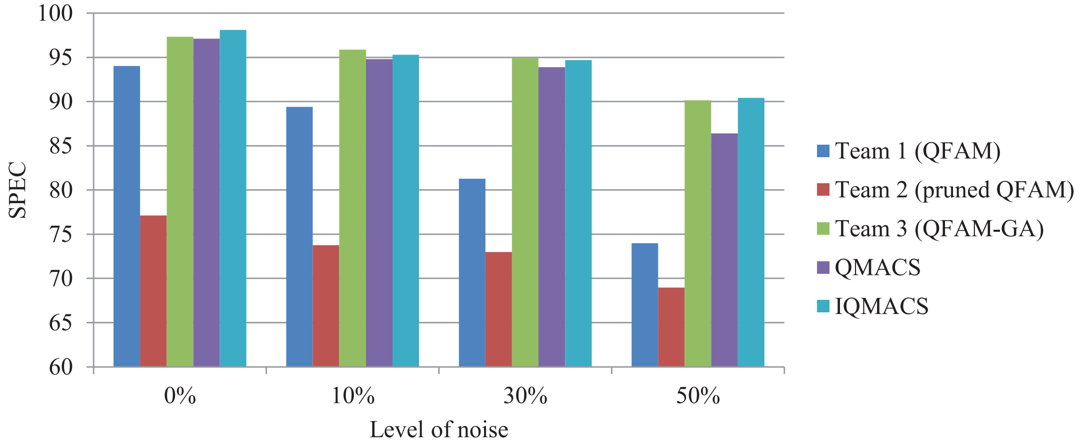


Fig. 6. The SPEC rates of IQMACS, QMACS and their team agents for human motion detection.

Table 10
The ACC rates (%) of individual agents with 95% confidence intervals for motor fault detection.

Method	Lower bound of the 95% confidence interval	Mean	Upper bound of the 95% confidence interval
QFAM	**98.60**	**99.20**	**99.80**
Pruned QFAM	93.20	94.80	96.20
QFAM-GA	97.60	99.01	**99.80**

Table 11
The number of prototypes before and after pruning with 95% confidence intervals for motor fault detection.

Method	Lower bound of the 95% confidence interval	Mean	Upper bound of the 95% confidence interval
Before pruning	16.36	17.30	18.22
After pruning	7.10	7.89	8.72

confidence intervals. Although, Pruned QFAM was not able to perform as good as the other two agents, it managed to reduce the network complexity more than 50% (Table 11).

Next, the effectiveness of IQMACS and QMACS was evaluated. The 10-fold cross validation method was used, with 80%, 10% and 10% of data samples used for learning, validation, and test, respectively. The accuracy rates of IQMACS, QMACS and their constituents are shown in Table 12. Team 1 outperformed other teams. QMACS performed similarly to team 3, while IQMACS produced similar results as to that of team 1 (which is the best team).

The motor fault data set was further evaluated in a two-class problem, e.g., faulty and fault-free. The 5-fold cross-validation method was used, due to the small size, and was repeated 10 times. A total of 60%, 20% and 20% of data samples were used for learning, validation, and test, respectively. To further evaluate the effectiveness of IQMACS, the class label of training samples was injected with noise levels of 10%, 30%, and 50%.

The ACC rates of IQMACS, QMACS, and their teams are shown in Fig. 7. The ACC scores of IQMACS and QMACS, and their teams decreased with increasing the level of noise. Teams 1 and 3 produced similar results in dealing with the noise-free problems, while team 2 outperformed the other two teams when the level of noise increased. Fig. 8 shows the number of prototype nodes before and after pruning. QFAM was more complex as compared with QFAM-GA and Pruned QFAM. Again, both IQMACS and QMACS could achieve comparable results by combining the decisions of three teams.

The SENS and SPEC rates of IQMACS, QMACS and their teams are shown in Figs. 9 and 10, respectively. As expected, the SENS and SPEC rates of all models degraded when the level of noise increased. For the noise-free data set, all teams perfectly recognized the normal motor condition (Fig. 9), while team 2 (Pruned QFAM) achieved the highest SPEC rates as compared with those of teams 1 and 3 for the noisy problem (Fig. 10).

5.3. Discussion

The empirical results demonstrate that individual ANNs produce different classification rates for various data sets with different characteristics. As shown in Figs. 3 and 7, although Pruned QFAM produces inferior results as compared with those from both QFAM and QFAM-GA, it achieves the highest ACC scores for the multi-class and noisy motor fault diagnosis problems, except for

Table 12
The ACC rates (%) of IQMACS, QMACS and their team agents with 95% confidence intervals for motor fault detection.

Method	Lower bound of the 95% confidence interval	Mean	Upper bound of the 95% confidence interval
Team 1 (QFAM)	**97.80**	**98.60**	**99.40**
Team 2 (Pruned QFAM)	82.60	88.01	91.40
Team 3 (QFAM-GA)	92.80	94.63	96.80
QMACS	93.80	94.68	95.56
IQMACS	97.58	98.44	99.12

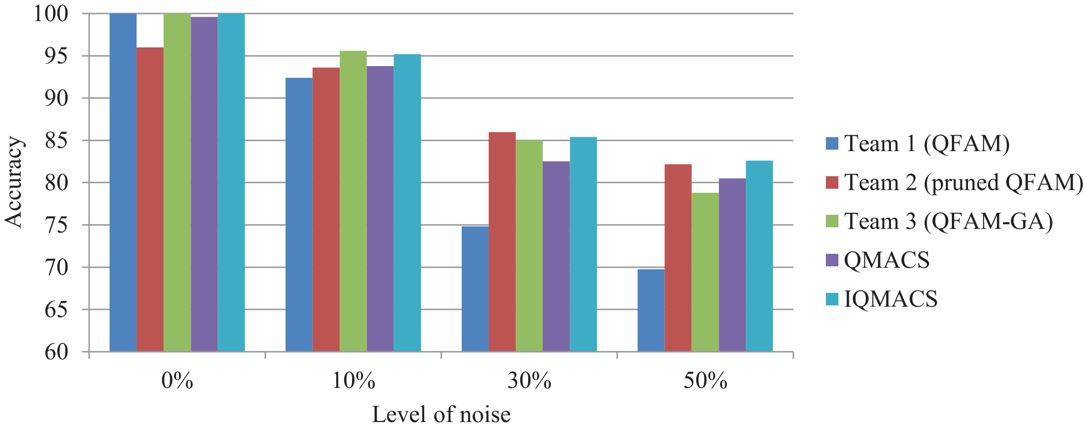


Fig. 7. The ACC rates(%) of IQMACS, QMACS and their team agents for motor fault detection.

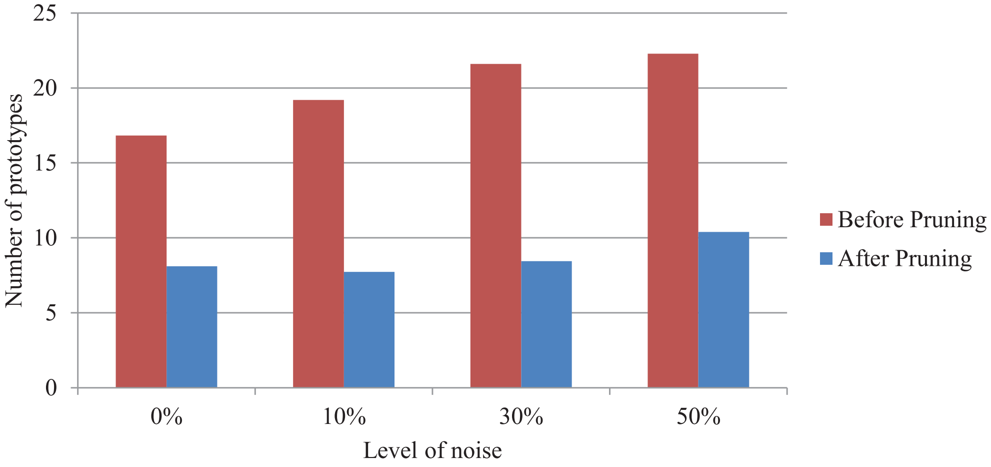


Fig. 8. The number of prototypes before and after pruning in presence of different levels of noise.

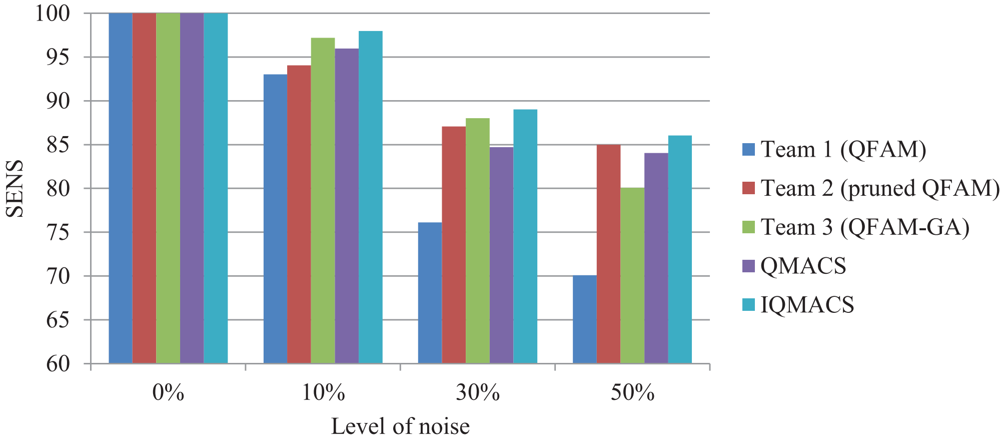


Fig. 9. The SENS rates of IQMACS, QMACS and their team agents for the motor fault detection data set.

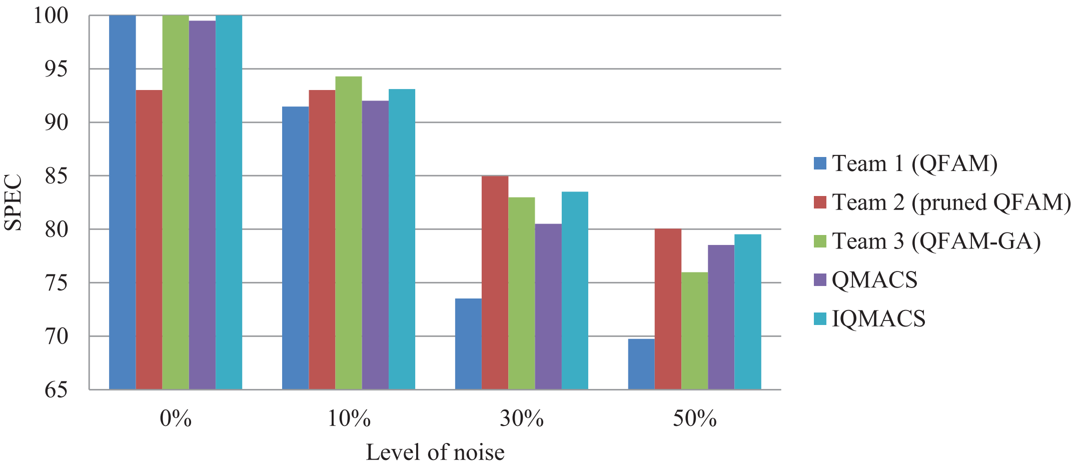


Fig. 10. The SPEC rates of IQMACS, QMACS and their team agents for the motor fault detection data set.

the case with 50% noise. In other words, Pruned QFAM has the advantage of tackling noisy problems, with a parsimonious network structure after pruning. Both IQMACS and QMACS are able to produce promising results by combining the outcomes of individual models.

6. Summary

In this paper, we have introduced IQMAC by enhancing our previous QMACS model with an additional trust measurement scheme at the team level. The performance of IQMACS, QMACS, and the constituent agent teams have been evaluated using benchmark problems as well as two real-world case studies. For each real-world case study, statistical features were extracted to represent the data samples for classification in noise-free and noisy conditions. Several performance indicators including ACC, SENS and SPEC have been computed. The outcome indicates, although individual teams perform differently in tackling different problems, IQMACS is able to produce promising results by combining the predictions of their agent teams. In addition, IQMACS is able to enhance the performance of QMACS through the trust measurement at the team level.

For further work, QFAM-based models can be replaced by other online ANNs, e.g. FMM or probabilistic neural networks that have similar structures as that of QFAM. Moreover, QFAM-based networks can be adapted to deal with function approximation problems, especially noisy functions, by exploiting the embedded Q-learning method in the structures.

Declaration of competing interest

The corresponding author certifies that all authors have read the manuscript before submission, and there is no conflict of interest associated with this publication, and there has been no financial support for this work that could have influenced its outcome.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (Grant nos. 61772344, 61811530324 and 61732011), the Natural Science Foundation of SZU (Grant nos. 827-000140, 827-000230, and 2017060), and the Interdisciplinary Innovation Team of SZU.

References

[1] M. Vardhana, N. Arunkumar, S. Lasrado, E. Abdulhay, G. Ramirez-Gonzalez, Convolutional neural network for bio-medical image segmentation with hardware acceleration, Cognit. Syst. Res. 50 (2018) 10–14.

[2] M. Talaat, M. Gobran, M. Wasfi, A hybrid model of an artificial neural network with thermodynamic model for system diagnosis of electrical power plant gas turbine, Eng. Appl. Artif. Intell. 68 (2018) 222–235.

[3] H. Zhang, S. Zhang, Y. Yin, X. Chen, Prediction of the hot metal silicon content in blast furnace based on extreme learning machine, Int. J. Mach. Learn. Cybern. 9 (10) (2018) 1697–1706.

[4] S. Rezvani, X. Wang, F. Pourpanah, Intuitionistic fuzzy twin support vector machines, IEEE Trans. Fuzzy Syst. (2019) 1-1, doi:10.1109/TFUZZ.2019.2893863.

[5] X. Feng, Q. Tan, C. Wei, Hesitant fuzzy linguistic multi-criteria decision making based on possibility theory, Int. J. Mach. Learn. Cybern. 9 (9) (2018) 1505–1517.

[6] Action recognition and localization with spatial and temporal contexts, Neurocomputing 333 (2019) 351–363.

[7] T. Rao, X. Li, H. Zhang, M. Xu, Multi-level region-based convolutional neural network for image emotion classification, Neurocomputing 333 (2019) 429–439.

[8] J. Yang, Y. Bai, F. Lin, M. Liu, Z. Hou, X. Liu, A novel electrocardiogram arrhythmia classification method based on stacked sparse auto-encoders and softmax regression, Int. J. Mach. Learn. Cybern. 9 (10) (2018) 1733–1740.

[9] D.P. Mishra, P. Ray, Fault detection, location and classification of a transmission line, Neural Comput. Appl. 30 (5) (2018) 1377–1424.

[10] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (1990) 993–1001.

[11] F. Pourpanah, C.J. Tan, C.P. Lim, J. Mohamad-Saleh, A Q-learning-based multi--agent system for data classification, Appl. Soft Comput. 52 (2017) 519–531.

[12] G.A. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Comput. Vis. Gr. Image Process. 37 (1) (1987) 54–115.

[13] R.S. Sutton, A.G. Barto, et al., Introduction to Reinforcement Learning, 135, MIT Press Cambridge, 1998.

[14] B.H. Abed-alguni, S.K. Chalup, F.A. Henskens, D.J. Paul, A multi-agent cooperative reinforcement learning model using a hierarchy of consultants, tutors and workers, Vietnam J. Comput. Sci. 2 (4) (2015) 213–226.

[15] F. Pourpanah, C.P. Lim, J.M. Saleh, A hybrid model of fuzzy ARTMAP and genetic algorithm for data classification and rule extraction, Expert Systems with Applications 49 (2016) 74–85.

[16] F. Pourpanah, C.P. Lim, Q. Hao, A reinforced fuzzy ARTMAP model for data classification, Int. J. Mach. Learn. Cybern. (2018) 1–13, doi:10.1007/s13042-018-0843-4.

[17] S. Abdelhedi, R. Bourguiba, J. Mouine, M. Baklouti, Development of a two-threshold-based fall detection algorithm for elderly health monitoring, in: Proceedings of the IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), 2016, pp. 1–5.

[18] P. Lorenzi, R. Rao, G. Romano, A. Kita, M. Serpa, F. Filesi, R. Parisi, A. Suppa, M. Bologna, A. Berardelli, F. Irrera, Smart sensing systems for the detection of human motion disorders, Proc. Eng. 120 (2015) 324–327. Eurosensors 2015

[19] N. MÂnera, C. Alvarez, S. Sastoque, M. Iregui, Human features extraction by using anatomical and low level image descriptors from whole body images, in: Proceedings of the XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA), 2016, pp. 1–6.

[20] A. Narayan, H. Dasgupta, H. Pal, Instrumentation of a portable EMG sensor in a single PCB for human motion pattern detection, in: Proceedings of the Second International Symposium on Physics and Technology of Sensors (ISPTS), 2015, pp. 1–5.

[21] O. Politi, I. Mporas, V. Megalooikonomou, Human motion detection in daily activity tasks using wearable sensors, in: Proceedings of the Twenty-Second European Signal Processing Conference (EUSIPCO), 2014, pp. 2315–2319.
[22] R.F.S.C. Oliveira, C.J.A. Bastos-Filho, Human detection in digital videos using motion features extractors, in: Proceedings of the IEEE Latin American Conference on Computational Intelligence (LA-CCI), 2016, pp. 1–2.
[23] A.M. Khan, Y.K. Lee, S.Y. Lee, T.S. Kim, Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis, in: Proceedings of the Fifth International Conference on Future Information Technology, 2010, pp. 1–6.
[24] C.A. Ronao, S. Cho, Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models, in: Proceedings of the Tenth International Conference on Natural Computation (ICNC), 2014, pp. 681–686.
[25] D.J.T. Siyambalapitiya, P.G. McLaren, Reliability improvement and economic benefits of on-line monitoring systems for large induction machines, in: Proceedings of the Conference Record of the 1988 IEEE Industry Applications Society Annual Meeting, 1, 1988, pp. 231–238.
[26] K. goston, Fault detection of the electrical motors based on vibration analysis, Proc. Technol. 19 (2015) 547–553. 8th International Conference Interdisciplinarity in Engineering, INTER-ENG 2014, 9–10 October 2014, Tirgu Mures, Romania
[27] A.G. Garcia-Ramirez, L.A. Morales-Hernandez, R.A. Osornio-Rios, J.P. Benitez-Rangel, A. Garcia-Perez, R. de Jesus Romero-Troncoso, Fault detection in induction motors and the impact on the kinematic chain through thermographic analysis, Electr. Power Syst. Res. 114 (2014) 1–9.
[28] A. Siddique, G.S. Yadava, B. Singh, A review of stator fault monitoring techniques of induction motors, IEEE Trans. Energy Convers. 20 (1) (2005) 106–114.
[29] L. Frosini, E. Bassi, Stator current and motor efficiency as indicators for different types of bearing faults in induction motors, IEEE Trans. Ind. Electron. 57 (1) (2010) 244–251.
[30] B. Efron, Bootstrap methods: another look at the Jackknife, Ann. Stat. 7 (1) (1979) 1–26.
[31] H. Foroughi, A. Naseri, A. Saberi, H.S. Yazdi, An eigenspace-based approach for human fall detection using integrated time motion image and neural network, in: Proceedings of the Ninth International Conference on Signal Processing, 2008, pp. 1499–1503.
[32] L. Liu, L. Shao, P. Rockett, Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition, Pattern Recognit. 46 (7) (2013) 1810–1818.
[33] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Proceedings of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72.
[34] H. Jhuang, T. Serre, L. Wolf, T. Poggio, A biologically inspired system for action recognition, in: Proceedings of the IEEE Eleventh International Conference on Computer Vision, ICCV 2007, IEEE, 2007, pp. 1–8.
[35] F. Pourpanah, Y. Shi, C.P. Lim, Q. Hao, C.J. Tan, Feature selection based on brain storm optimization for data classification, Appl. Soft Comput. 80 (2019) 761–775.
[36] F. Pourpanah, B. Zhang, R. Ma, Q. Hao, Non-intrusive human motion recognition using distributed sparse sensors and the genetic algorithm based neural network, in: Proceedings of the IEEE SENSORS, IEEE, 2018, pp. 1–4.
[37] J. Luo, W. Wang, H. Qi, Spatio-temporal feature extraction and representation for RGB-D human action recognition, Pattern Recognit. Lett. 50 (2014) 139–148. Depth Image Analysis
[38] F.-C. Hsu, J. Gubbi, M. Palaniswami, Head detection using motion features and multi level pyramid architecture, Comput. Vis. Image Underst. 137 (2015) 38–49.
[39] C. Wojek, S. Walk, B. Schiele, Multi-cue onboard pedestrian detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 794–801.
[40] P. Dollár, Z. Tu, P. Perona, S. Belongie, Integral Channel Features, BMVC Press, 2009.
[41] S. Mane, R. Kambli, F. Kazi, N. Singh, Hand motion recognition from single channel surface EMG using wavelet & artificial neural network, Proc. Comput. Sci. 49 (2015) 58–65.
[42] M. Seera, C.P. Lim, D. Ishak, H. Singh, Fault detection and diagnosis of induction motors using motor current signature analysis and a hybrid FMMCART model, IEEE Trans. Neural Netw. Learn. Syst. 23 (1) (2012) 97–108.
[43] F. Pourpanah, C.P. Lim, X. Wang, C.J. Tan, M. Seera, Y. Shi, A hybrid model of fuzzy min-max and brain storm optimization for feature selection and data classification, Neurocomputing 333 (2019) 440–451.
[44] S. Altaf, A. Al-Anbuky, H. GholamHosseini, Fault diagnosis in a distributed motor network using artificial neural network, in: Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2014, pp. 190–197.
[45] F. Pourpanah, B. Zhang, R. Ma, Q. Hao, Anomaly detection and condition monitoring of UAV motors and propellers, in: Proceedings of the IEEE SENSORS, 2018, pp. 1–4.
[46] R.C. Bhavsar, R.A. Patel, B.R. Bhalja, Condition monitoring of induction motor using artificial neural network, in: Proceedings of the Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD), 2014, pp. 1–6.
[47] S. Morsalin, K. Mahmud, H. Mohiuddin, M.R. Halim, P. Saha, Induction motor inter-turn fault detection using heuristic noninvasive approach by artificial neural network with levenberg marquardt algorithm, in: Proceedings of the International Conference on Informatics, Electronics Vision (ICIEV), 2014, pp. 1–6.
[48] S.S. Refaat, H. Abu-Rub, M.S. Saad, E.M. Aboul-Zahab, A. Iqbal, Detection, diagnoses and discrimination of stator turn to turn fault and unbalanced supply voltage fault for three phase induction motors, in: Proceedings of the IEEE International Conference on Power and Energy (PECon), 2012, pp. 910–915.
[49] S.O. Ibrahim, khaled Nagdy Faris, E.A. Elzahab, Implementation of fuzzy modeling system for faults detection and diagnosis in three phase induction motor drive system, J. Electr. Syst. Inf. Technol. 2 (1) (2015) 27–46.
[50] S. S., K.U. Rao, S. Jade, Detection and classification of pq disturbances in the supply to induction motor using wavelet transform and feedforward neural network, in: Proceedings of the IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2015, pp. 1–5.
[51] W.F. Godoy, I.N. da Silva, A. Goedtel, R.H.C. Palcios, Evaluation of stator winding faults severity in inverter-fed induction motors, Appl. Soft Comput. 32 (2015) 420–431.
[52] R.H.C. Palcios, I.N. da Silva, A. Goedtel, W.F. Godoy, A comprehensive evaluation of intelligent classifiers for fault identification in three-phase induction motors, Electr. Power Syst. Res. 127 (2015) 249–258.
[53] G.A. Carpenter, S. Grossberg, D.B. Rosen, Fuzzy art: fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Netw. 4 (6) (1991) 759–771.
[54] L.A. Zadeh, Fuzzy sets, in: Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh, World Scientific, 1996, pp. 394–432.
[55] H. Ishibuchi, T. Murata, I. Tõrkʖen, Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems, Fuzzy Sets Syst. 89 (2) (1997) 135–150.
[56] A. Quteishat, C.P. Lim, J.M. Saleh, J. Tweedale, L.C. Jain, A neural network-based multi-agent classifier system with a Bayesian formalism for trust measurement, Soft Comput. 15 (2) (2011) 221–231.
[57] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Elsevier, 2014.
[58] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Syst. Man Cybern. 22 (3) (1992) 418–435.
[59] M. Lichman, UCI machine learning repository, 2013, http://archive.ics.uci.edu/ml.
[60] S. Bashir, U. Qamar, F.H. Khan, Intellihealth: a medical decision support application using a novel weighted multi-layer classifier ensemble framework, J. Biomed. Inform. 59 (2016) 185–200.
[61] E. Frank, M. Hall, B. Pfahringer, Locally weighted naive Bayes, in: Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2002, pp. 249–256.
[62] K.T. Leung, D.S. Parker, Empirical comparisons of various voting methods in bagging, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 595–600.
[63] J.L. Melsa, D.L. Chon, in: Decision and Estimation Theory, McGraw-Hill, New York, 1978.
[64] Z. Xu, J. Xuan, T. Shi, B. Wu, Y. Hu, A novel fault diagnosis method of bearing based on improved fuzzy ARTMAP and modified distance discriminant technique, Expert Syst. Appl. 36 (9) (2009) 11801–11807.
[65] C.J. Tan, C.P. Lim, Y. Cheah, A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models, Neurocomputing 125 (2014) 217–228.

Farhad Pourpanah received his Ph.D. degree in computational intelligence from the University Science Malaysia in 2015. He is currently an associate researcher with the College of Mathematics and Statistics, Shenzhen University (SZU), China. Before joining to SZU, he was a postdoctoral research fellow with the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), China. His research interests include computational intelligence, especially pattern recognition, feature section and evolutionary algorithms.

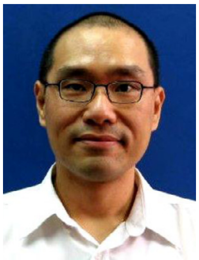
Ran Wang received the B.Eng. degree in computer science from the College of Information Science and Technology, Beijing Forestry University, Beijing, China, in 2009, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2014. From 2014 to 2016, she was a Post-Doctoral Researcher with the Department of Computer Science, City University of Hong Kong. She is currently an Assistant Professor with the College of Mathematics and Statistics, Shenzhen University, Shenzhen, China. Her current research interests include pattern recognition, machine learning, fuzzy sets and fuzzy logic, and their related applications.

Chee Peng Lim received his Ph.D. degree in intelligent systems from the University of Sheffield, UK, in 1997. His research interests include computational intelligence-based systems for data analytics, condition monitoring, optimization, and decision support. He has published more than 350 technical papers in journals, conference proceedings, and books, received 7 best paper awards, edited 3 books and 12 special issues in journals, and served in the editorial board of 5 international journals. He is currently a professor at Institute for Intelligent Systems Research and Innovation, Deakin University.

Xizhao Wang served in Hebei University as a professor and the dean of school of Mathematics and Computer Sciences before 2014. After 2014 Prof. Wang worked as a professor in Big Data Institute of Shenzhen University. Prof. Wang's major research interests include uncertainty modeling and machine learning for big data. Prof. Wang has edited 10+ special issues and published 3 monographs, 2 textbooks, and 200+ peer-reviewed research papers. As a Principle Investigator (PI) or co-PI,

Prof. Wang has completed 30+ research projects. Prof. Wang has supervised more than 100 M.phil. and Ph.D. students. Prof. Wang is an IEEE Fellow, the previous BoG member of IEEE SMC society, the chair of IEEE SMC Technical Committee on Computational Intelligence, the Chief Editor of Machine Learning and Cybernetics Journal, and associate editors for a couple of journals in the related areas. He was the recipient of the IEEE SMCS Outstanding Contribution Award in 2004 and the recipient of IEEE SMCS Best Associate Editor Award in 2006. He is the general Co-Chair of the 2002–2017 International Conferences on Machine Learning and Cybernetics, cosponsored by IEEE SMCS. Prof. Wang was a distinguished lecturer of the IEEE SMCS.

Manjeevan Seera received the B.Eng. (Hons) degree in Electronics and Electrical Engineering from the University of Sunderland, U.K., in 2007 and Ph.D. degree in Computational Intelligence from University Science Malaysia in 2012. He is currently an Adjunct Research Fellow with Swinburne University of Technology (Sarawak Campus), Malaysia. His research interests include soft computing, pattern recognition, fault detection and diagnosis, and human-robot interaction.



Choo Jun Tan received his Ph.D. degree from School of Computer Sciences, University Science of Malaysia in 2014. He has more than 10-year experience in software design and development, as well as research and development in computational intelligence. He has designed and developed a number intelligent software systems that have received national and international awards. His research interests include computational intelligence, especially evolutionary algorithms and their application to multi-objective optimization problems.