A type-2 neuro-fuzzy system with a novel learning method for Parkinson’s disease diagnosis

Armin Salimi-Badr · Mohammad Hashemi · Hamidreza Saffari

Accepted: 17 October 2022 / Published online: 24 November 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
In this paper, an interpretable classifier for Parkinson’s Disease (PD) diagnosis based on analyzing the gait cycle is presented. The proposed method utilizes clinical features extracted from the vertical Ground Reaction Force (vGRF) measured by wearable sensors. Therefore, experts can verify the decision made by the proposed method. Type-2 fuzzy logic is applied to increase the robustness against noisy sensor data. First, the initial fuzzy rules are extracted using a K-Nearest-Neighbor-based clustering approach. Next, a novel quasi-Levenberg-Marquardt (qLM) learning approach is proposed and applied to fine-tune the initial rules based on minimizing the cross-entropy loss function using a trust-region optimization method. Finally, complementary online learning is proposed to improve rules by encountering new labeled samples. The performance of the method is evaluated to classify patients and healthy subjects in different conditions, including the presence of noise or observing new samples. Moreover, the performance of the model is compared to some previous supervised and unsupervised machine learning approaches. The final Accuracy, Precision, Recall, and F1 Score of the proposed method are 97.61%, 97.58%, 99.02%, and 98.30%.

Keywords Parkinson’s disease · Interval type-2 fuzzy neural networks · Explainable artificial intelligence (XAI) · quasi-Levenberg-Marquardt (qLM) learning method · Continual learning · Gait cycle

1 Introduction
Emerging symptoms of some chronic diseases, like Parkinson’s Disease, are diagnosed many years after infection. This delay reduces the chance of encountering the disease with some more simple treatment methods. Parkinson’s disease (PD) is the second most common neurological disease that puts the lives of a large portion of the elderly population in danger [1–4]. Research studies show that the origin of this neurodegenerative illness is the destruction of the dopaminergic neurons of the Substantia Nigra pars Compacta in the Basal Ganglia [1, 5–7]. According to the Parkinson’s disease foundation, the majority of patients suffering from PD have movement problems such as Bradykinesia (slowness), Akinesia (impaired power of voluntary movements), stiffness, and resting tremor [5]. Nearly 5 million people are affected by PD in the world [1].

It would be possible to control and decrease the syndromes through medical treatments or by applying appropriate lifestyle changes during the early stages of the disease. However, more complex and aggressive approaches, including surgical therapy like the Deep Brain Stimulus (DBP), are required for its more severe stages. Unfortunately, symptoms of PD appear a long time after the beginning of this disease [1]. Generally, the diagnosis task is challenging based on the reported misdiagnosis rate of 25% of cases [8]. Machine learning, ambient intelligence, and context-aware agents can help us enhance the quality of healthcare services and illness diagnoses.

Two important symptoms, studied in the previous methods to classify the patients and healthy persons are speech disorders (vocal impairment) [2, 8, 9] and gait cycle changing [1, 10, 11]. It is expected that PD causes changes in the gait cycle due to its effects on the patient’s movements. Moreover, considering the effects of PD on eye movements, some previous methods investigate these effects to detect PD patients [12].

Recently, machine learning approaches, including Deep Learning methods, have been successfully applied for medical diagnosis [13–15]. Although these methods have a
good performance in detecting patients, they work as black-boxes and lack interpretability and transparency [16]. In applications like the medical diagnosis, experts need more information from the model than a simple binary prediction to support their diagnosis [17]. Therefore, explanations supporting the output of a model are crucial for medical applications [16–18].

Most of the previous approaches that applied machine learning paradigms for classifying patients suffering from PD have not used either an interpretable artificial intelligence method [1, 10] or expert-understandable clinical features [11, 19, 20]. However, in a sensitive clinical application, using an interpretable method with clinical features is more acceptable for two reasons: (1) an expert can verify the decision of the model, and (2) an expert can modify and fine-tune the parameters of the model.

In this paper, we propose an explainable artificial intelligence (XAI) approach using the clinical features extracted from the gait cycle. The paradigm is proposed based on training a self-organizing interval-type-2 fuzzy neural network that is able to learn the interpretable fuzzy rules. A hybrid batch-online learning method is proposed to extract the parameters values of this structure. First, high-level features explaining the behavior of the gait cycle obtained from wearable sensors are extracted. Next, using available training instances, the parameters of an interval type-2 fuzzy neural network are initialized using a K-Nearest-Neighbor-based clustering approach. Next, a novel quasi-Levenberg-Marquardt (qLM) method is proposed to fine-tune these initialized parameters. Finally, considering the progress of studies and collecting more labeled instances in the future, a complementary online learning method is proposed to add new fuzzy rules encountering new training samples.

Based on the training of this interpretable structure, some fuzzy rules are extracted that can be used as a guide for experts to diagnose patients. On the other hand, experts can modify and fine-tune the parameters of extracted fuzzy rules based on their knowledge.

Moreover, considering that the sensors’ measurements are noisy, using the interval type-2 fuzzy logic increases the robustness of the extracted fuzzy rules against the noisy data.

In brief, the main contributions of our proposed method are listed as follows:

1. Presenting an interpretable model based on clinical features. Therefore, the decisions made by the model can be verified by the experts. Moreover, extracted rules can be used by experts or be adjusted based on the experts’ knowledge;

2. Increasing the robustness of the method against uncertainty and the sensor’s noisy measurements using Interval Type-2 Fuzzy logic;

3. Proposing an initial batch learning to extract fuzzy rules based on available training samples along with online learning for improving the model’s rule base using new labeled instances;

4. Proposing a novel learning approach inspired by the Levenberg Marquardt method for classification applications based on minimizing the Cross-Entropy loss function using a trust-region optimization method.

The rest of this paper is organized as follows: First, in Section 2 the related studies are reviewed. Afterward, in Section 3 the proposed paradigm, including the utilized features, preprocessing, the proposed Interval Type-2 Fuzzy Neural Network architecture, and the learning methods are explained. Next, the results on some labeled data are reported and compared with some other approaches that used similar clinical features in Section 4. At the end of Section 4, the extracted fuzzy sets are reported. Finally, conclusions are presented in Section 5.

2 Related work

The deterioration of executive functions and movement disorders in patients with PD have been shown extensively [21–23]. Yogev et al. [24] studied the impacts of different types of dual-tasking and cognitive function on the gait of patients with PD and control subjects. They also showed contrasting measures of gait rhythmicity for patients with PD in comparison to other features. Additionally, in [25] it is investigated that Parkinson’s disease has a great impact on the left-right symmetry of gait. Yogev et al. [25] conducted a similar walking condition for both patients with PD and healthy control subjects and they demonstrated that asymmetry of gait increased mainly during the dual-task condition in patients with PD but not in the healthy control subjects. Considering the Hausdorff et al. [26] studies on gait variability and Basal Ganglia disorders, it can be concluded that the ability to maintain a steady gait with low stride-to-stride variability of gait cycle timing, would be decreased in patients with PD. Parkinson’s disease symptoms also include speech disorders as well as cognitive impairments [27, 28]. In addition, 90% of the patients with PD themselves report speech impairments as one of the most significant symptoms [29].

To classify the patients with PD and healthy control subjects based on their gait cycles, both wearable [30–36] and non-wearable [37–47] sensors have been used in various experiments [4]. For instance, Jean et al. [30], conducted the classification using Spatial-Temporal Image of Plantar pressure (STIP) among normal step and patient steps with PD. In [36] the wearable sensors on-shoe along with some algorithms are presented to characterize the Parkinson’s disease motor symptoms.
 Accordingly, the classification of healthy control subjects and patients with PD using ground reaction force sensors placed in shoes has been extensively studied [1, 10, 11, 19, 48–50]. In [50], the vGRFs measurements of both left and right foot were used to extract statistical features including minimum, maximum, average, and the standard deviation of each time series. Their extracted features then were fed to machine learning binary classifiers, including Support Vector Machine (SVM). In [48] the gait asymmetry (GA) was calculated based on the difference of the ground reaction force (GRF) features of the left and right feet. This was done by decomposition of the GRF into components of different frequency sub-bands via the wavelet transform and Multi-Layer Perceptron (MLP) models. In [19], Lee et al. utilized the gait characteristics of idiopathic PD patients who shuffle their feet while they are walking to classify patients with PD and healthy control subjects. They trained a neural network with weighted fuzzy membership functions (NEWFM) using extracted 40 statistical and wavelet-based features.

Different supervised and unsupervised methods such as Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gaussian-Mixture Model (GMM), and K-Means are extensively utilized to classify patients with PD and healthy control subjects. In [51], automatic noninvasive identification of PD is used with the combination of wavelet analysis and SVM which led to an accuracy of 90.32%. Although most prior research focused on time-domain and frequency-domain features, only the clinical features extracted from vertical Ground Reaction Forces (vGRFs) were considered in [1]. Accordingly, nineteen statistical features are extracted and used as the input of machine learning-based classifiers.

A time-delay neural network classifier learned by a Q-back propagation learning approach was proposed in [20] using temporal information of vGRF time-series to classify the PD patients and healthy subjects. Data from three Parkinson’s disease research projects are used for evaluation of this approach [24, 52, 53]. The accuracy on the three sub-datasets reached 91.49%, 92.19%, and 90.91%, respectively.

Although most of the previous works tried to extract feature vectors based on some human knowledge, recently published method tried to extract high-level features based temporal and spatial analysis of the gait-cycle using deep neural networks [11, 13–15]. In [11], we have demonstrated that the spatial correlation among different sensors data during time placed in each left and right foot are useful for diagnosing the patients. To consider the temporal dependencies, a structure using Long-Short Term Memory (LSTM) cell layers has been proposed to build a Recurrent Neural Network (RNN). In [13], a deep neural network consists of one dimensional convolutional layers is used to classify patients. The final accuracy of the method reaches 98.7%. In [14] and [15], a combination of Long-short Term Memory (LSTM) and Convolutional Neural Network (CNN) are used to detect patients suffering from PD with accuracy equal to 98.61% and 99.22%, respectively. Although these deep methods perform efficiently, they have many parameters that must be determined based on a supervised learning method. Therefore, to learn these deep structures, considering the small size of available datasets, authors of these methods have segmented the sequence related each subject into very small pieces (10 to 20 steps) to increase the number of training samples. Moreover, these deep structures lack the interpretability and their reasoning for classification is not clear and verifiable for an expert.

Most previous studies have not used clinical features or have not applied an interpretable method for classifying PD patients. Using an interpretable machine learning approach that the base of its decision making is transparent and expert understandable is more agreeable in a clinical application [17]. Fuzzy Neural Networks are interpretable structures with neural representation [54–62]. In this study, the clinical features which are understandable for experts are extracted from the recorded vGRF signal and an interpretable structure, based on Interval Type-2 Fuzzy Neural Network [62–64] is utilized to classify subjects using these clinical features. Consequently, contrary to the previous studies that used clinical features with non-interpretable machine learning methods like [1], or used sequence of vGRF like [19], the proposed method is able to extract expert understandable rules. These rules can be verified and modified based on the knowledge of experts. On the other hand, experts can utilize the extracted rules to detect patients suffering from PD.

3 Materials and methods

In this section, we explain the proposed method, including the preprocessing, the architecture of the Interval type-2 Fuzzy Neural Network, the algorithm to learn its parameters, and finally a complementary online learning for improving the method when encountering new labeled examples. Figure 1 shows the whole proposed paradigm of this article. In the following subsections, different parts of this paradigm will be explained.

3.1 Features and preprocessing

Although healthy people’s walking is characterized by a repetition of the gait cycle pattern, significant variations could be seen between different gait cycles during patients’ walking (see Fig. 2) [10]. Consequently, the gait cycle pattern is examined in this paper to categorize patients
The proposed paradigm in a single view. First, the gait cycle in the form of vertical Ground Reaction Force (vGRF) is extracted from some wearable sensors during subject’s walking. Afterward, the recorded vGRF signals are preprocessed. Next, clinical features are extracted from the recorded vGRF signals and the training samples are constructed. Using the training samples, an Interval Type-2 Fuzzy Neural Network is trained based on a batch learning algorithm and some fuzzy rules are extracted.

To have a common range of definition, considering different ranges for different features, we normalize all features to the range [0,1].

3.2 Interval type-2 fuzzy neural network

The general architecture of the proposed interval type-2 fuzzy neural network (IT2FNN) is shown in Fig. 4. The inputs to this network are the extracted features explained in Section 3.1, and its final output is the probability of belonging to the patient (positive) class, which is a value in the range [0,1]. Indeed, the output indicates that how much the subject is suspicious of having Parkinson’s disorder. This IT2FNN is built upon the *Mamdani Fuzzy Inference System* [67], using the Dot-Product function as the *T-Norm* operator. Furthermore, it is assumed that the fuzzy membership functions are *Gaussian* with uncertain width. Indeed, based on the proposed learning method (explained in Section 3.3.1) the centers of fuzzy sets are extracted, but there is no clue as the precise values of their width. To solve this problem, the width of fuzzy sets is considered.
Fig. 2 Comparing the gait cycle obtained from vGRF of a healthy subject (a) and a patient suffering from Parkinson’s Disease (b)

uncertain. This assumption improves the robustness of the model against uncertainty and sensor noisy measurements as investigated in Section 4. For type reduction, an adaptive version of the direct Biglarbegian-Melek-Mendel (BMM) method [68–71] is utilized. The details of each layer are explained as follows:

1. **Input Layer:** This layer encodes the input variable vector extracted form the gait cycle of the subject. The

| Feature | Extracted references |
|---------|----------------------|
| $z_1$   | Short swing time (average) [25] |
| $z_2$   | Long swing time (average) [25] |
| $z_3$   | Gait asymmetry (average) [25] |
| $z_4$   | Left foot swing time percentage (average) [24, 26] |
| $z_5$   | Right foot swing time percentage (average) [24, 26] |
| $z_6$   | Left foot swing time (average) [24, 65, 66] |
| $z_7$   | Left foot swing time (average) [24, 65, 66] |
| $z_8$   | Short swing time coefficients of variations [25] |
| $z_9$   | Long swing time coefficients of variations [25] |
| $z_{10}$| Gait asymmetry coefficients of variations [25] |
| $z_{11}$| Left foot swing time coefficients of variations [25] |
| $z_{12}$| Right foot swing time coefficients of variations [25] |
| $z_{13}$| Left foot stride time coefficients of variations [24] |
| $z_{14}$| Right foot stride time coefficients of variations [24] |

| Feature symbol | Feature value |
|----------------|--------------|
| $x_1$          | $z_1$        |
| $x_2$          | $z_2$        |
| $x_3$          | $z_3$        |
| $x_4$          | $z_4 + z_5$  |
| $x_5$          | $\frac{z_6 + z_7}{2}$ |
| $x_6$          | $z_8$        |
| $x_7$          | $z_9$        |
| $x_8$          | $\frac{z_{10} + z_{11}}{2}$ |
| $x_9$          | $\frac{z_{13} + z_{14}}{2}$ |
output of this layer is as follows:

$$X = [x_1, x_2, \cdots, x_{10}]^T$$

(1)

where, $x_1$ to $x_{10}$ are extracted features from the gait cycle introduced in Table 2.

2. **Fuzzification Layer:** There are two types of neurons in this layer to encode the interval type-2 fuzzy sets used for defining different fuzzy rules. For $i^{th}$ ($i = 1, 2, \ldots, R$) fuzzy rule, the interval type-2 fuzzy membership value of $j^{th}$ ($j=1,2,\ldots,10$) input variable is defined as follows:

$$\mu_{i,j}(x_j) = e^{\frac{-(x_j-c_{ij})^2}{\sigma^2}}$$

(2)

$$\overline{\mu}_{i,j}(x_j) = e^{\frac{-(x_j-c_{ij})^2}{\overline{\sigma}^2}}$$

(3)

where, $\mu_{i,j}$, and $\overline{\mu}_{i,j}(x_j)$ are lower and upper membership functions (LMF and UMF) of $j^{th}$ input variable to the $i^{th}$ fuzzy rule indicating the footprint of uncertainty (FOU), $c_{ij}$ is the center of the defined fuzzy set for $j^{th}$ input variable in the $i^{th}$ fuzzy rule, $\sigma$ is the width of lower membership functions, and $\overline{\sigma}$ is the width of upper membership functions.

3. **T-Norm Layer:** The interval of fuzzy rules’ firing strength are calculated by applying Dot-Product T-Norm operator in this layer as follows:

$$\phi_i = \prod_{j=1}^{n} \mu_{i,j}(x_j)$$

(4)

$$\overline{\phi}_i = \prod_{j=1}^{n} \overline{\mu}_{i,j}(x_j)$$

(5)

Fig. 3 Effect of applying preprocessing on vGRF signals of a subject. (a) before preprocessing, (b) after preprocessing
Fig. 4 The proposed interval type-2 fuzzy neural network architecture

where $\phi_i$ and $\bar{\phi}_i$ are lower and upper firing strength of the $i^{th}$ fuzzy rule.

4. **Normalization Layer**: To realize the BMM direct type reduction and defuzzification method, the extracted upper and lower firing strength of different fuzzy rules (output vector of the previous layer) are normalized by neurons of this layer as follows:

$$f_i = \frac{\phi_i}{\sum_{j=1}^{R} \phi_j} \quad (6)$$

$$\bar{f}_i = \frac{\bar{\phi}_i}{\sum_{j=1}^{R} \bar{\phi}_j} \quad (7)$$

where, $f_i$ and $\bar{f}_i$ are normalized upper and lower firing strength of the $i^{th}$ fuzzy rule.

5. **Aggregation Layer**: Neurons of this layer calculate the boundaries of the network’s output $[y_l, y_r]$ receiving the firing strength interval calculated in the T-Norm Layer as follows:

$$y_l = \sum_{i=1}^{R} y_l f_i$$

$$y_r = \sum_{i=1}^{R} y_r f_i$$

$$y = \frac{1}{1 + e^{-(m_l y_l + m_r y_r - \theta)}} \quad (9)$$

where, $m_l$ and $m_r$ are two weights for type reduction based on the BMM direct approach. If $y > 0.5$, the final decision would be patient, otherwise the input sample is classified as healthy.

It is worthy to mention that three final layers (normalization, aggregation and type-reduction layers) realize the direct BMM type-reduction and defuzzification with adaptive weights [68–71].

Usually, Interval Type-2 Fuzzy Neural Networks (IT2FNN) utilize the iterative Karnik-Mendel (KM) algorithms [72, 73] for Type Reduction [74–78]. Although the
iterative method is accurate, it is time-consuming. Recently, some IT2FNN [62, 68, 79] have been proposed based on using direct type reduction methods like the Nie-Tan and Begian-Melek-Mendel (BMM) operators. The BMM method was originally proposed for the Takagi-Sugeno fuzzy inference system (FIS) [80]. In this paper, considering the interpretability of the Mamdani FIS, an Interval Type-2 Fuzzy Neural Network (IT2FNN) based on the Mamdani [67] FIS using an adaptive version of BMM for Type Reduction is proposed. The extracted fuzzy rules of this IT2FNN can be interpreted in the form of:

\[
\text{Fuzzy Rule } i: IF \ x_1 \ \text{is} \ \tilde{A}_{i,1} \ \text{and} \ x_2 \ \text{is} \ \tilde{A}_{i,2} \ \text{and} \ldots \ \text{and} \ x_{10} \ \text{is} \ \tilde{A}_{i,10} \ \text{Then} \ y \ \text{is} \ y_i
\]

where, \( \tilde{A}_{i,j} \) (j = 1, 2, ..., 10) is a Gaussian interval type-2 fuzzy set with uncertain width (see Fig. 5) defined for \( i^{th} \) fuzzy rule and \( j^{th} \) variable (\( x_j \) in Table 2), y is the output of the FIS, and \( y_i \in [-1, 1] \) is the consequent part of \( i^{th} \) fuzzy rule. To report the fuzzy rules, it is sufficient to report the extracted centers for all interval type-2 fuzzy sets \( \tilde{A}_{i,j} \).

### 3.3 Proposed learning algorithm

The proposed learning algorithm consists of three phases. First, based on the interpretable architecture of a Fuzzy Neural Network (FNN), the different parameters of the antecedent and consequent parts of fuzzy rules are initialized with a density-based clustering algorithm. Next, the initialized parameters are fine-tuned using a novel entropy-aware-Levenberg-Marquardt, which is appropriate for classification problems. We name this method quasi-Levenberg-Marquardt (qLM). Finally, to provide the ability to adapt parameters based on new labeled instances, a complementary online learning rule is proposed.

#### 3.3.1 Initialization method

The introduced network in Section 3.3.2 has two categories of parameters we should determine to classify subjects properly: 1- Fuzzy rules antecedent parts’ parameters including the centers and width values of fuzzy sets, 2- Fuzzy rules consequent parts’ parameters indicating the decision of each fuzzy rule (\( y_i \)). To determine these parameters, based on the interpretability of the FNN, a clustering approach based on K-Nearest Neighbors is proposed and applied to the training dataset as follows:

1. Find the K nearest neighbors (KNN) of each training sample. Here, the desired output (label) which is +1 for patients and -1 for healthy subjects is considered as an input variable. Therefore, the \( k^{th} \) training sample \( X_k \) is a vector with 11 dimensions as follows:

\[
X_k^{\text{augmented}} = [x_{k,1}, x_{k,2}, \ldots, x_{k,10}, y^*_k]
\]

where \( x_{k,j} \) (j = 1, 2, ..., 10) are values of input variables for \( k^{th} \) sample and \( y^*_k \) is its desired output value. Using the output value along with input variables causes that the clustering approach considers the classes of training instances in composing clusters. Indeed, in constructing each cluster the class labels of different samples are considered.

2. Find the sample that represents the densest cluster. To do this, we calculate the Mean-Squared Distance (MSD) of each sample with its KNN. Next, the sample with the minimum value of MSD is chosen as a fuzzy rule’s center. Afterward, the sample with its KNN are removed from the dataset for choosing the next rule’s center. Since here the output value is considered as a new dimension, the purity of clusters regarding their labels is investigated for choosing a rule’s center.

3. After choosing center of \( i^{th} \) (i = 1, 2, ..., R) fuzzy rule (vector \( V_i \)), its first 10 elements are considered as the center of its fuzzy sets. Moreover, to determine the consequent part parameters \( y_i \) for \( i^{th} \) fuzzy rule, the average of desired outputs of training samples that belong to the \( i^{th} \) cluster, is calculated as follows:

\[
y_i = 1/K \sum_{k \in KNN(V_i)} y^*_k
\]

where \( y^*_k \) is the desired output value for \( k^{th} \) member of \( KNN(V_i) \).

4. To determine the fuzzy sets width, considering the uncertainty we assume that the width belongs to the range \([\sigma_1, \sigma_2]\) where \( \sigma_1 \) and \( \sigma_2 \) are two constant values lower than 1. Therefore, \( \sigma_1 \) is considered as the width of the lower membership functions, and \( \sigma_2 \) is considered as the width of the upper membership function as

![Gaussian Interval Type-2 Fuzzy Set with uncertain width](image)

Fig. 5 An example of an interval type-2 fuzzy set with uncertain width. UMF: Upper Membership Function, LMF: Lower Membership Function, FOU: Footprint of Uncertainty
follows:

\[ \sigma = \sigma_1 \]
\[ \overline{\sigma} = \sigma_2 \]  

(12)

The proposed learning algorithm is summarized in Algorithm 1.

---

**Algorithm 1** Initialization algorithm.

---

### 3.3.2 Fine-tuning method

After initializing the parameters, to improve the performance of the model, it is advantageous to apply a local search method to fine-tune them. Generally, local search methods like the well-known Stochastic Gradient Descent (SGD) approach suffer from the problem of getting stuck at a local optimum. Different improvements are applied to provide the ability to avoid getting stuck at local optima. A very successful iterative local search approach is the Levenberg-Marquardt (LM) algorithm, which considers a trust region for changing different parameters [57, 58, 62, 81–83]. It is possible to interpret this learning approach as a second-order hessian-free optimization method that approximates the Hessian matrix based on the Jacobian matrix.

\[
P = P + \Delta P
\]  

(13)

The LM optimization method is based on using the Taylor’s expansion of the network’s output and minimizing an augmented version of Mean-Squared-Error (MSE) considering a trust region for changing parameters (see more details in [58, 62]). Although MSE is an appropriate loss function for regression applications, it is not suitable for classification problems [84]. Considering the output of a classifier as the probability of belonging to a class, the Cross-Entropy loss function is an appropriate loss function for classification applications instead of the MSE [84].

In this paper, a novel optimization method inspired by the LM method and suitable for classification applications based on minimizing the Cross-Entropy loss function, instead of MSE, is presented (quasi-Levenberg-Marquardt (qLM)).

The proposed learning algorithm is an iterative method that updates the parameters of the model as follows:

\[
\text{Input:} \\
\quad \text{training samples } X \text{ along with their desired output values } y^* \\
\quad \text{name of rules } (R), \text{ number of samples } (N) \\
\quad \text{range of fuzzy sets width } (\sigma, \overline{\sigma}) \\
\quad \text{number of rules } (R), \text{ number of samples } (N) \\
\quad \text{range of fuzzy sets width } (\sigma_1, \sigma_2) \\
\quad \text{Output:} \\
\quad \text{initialized center of fuzzy sets } (c_{i,j}) \\
\quad \text{initialized consequent parts’ parameters } \gamma_i \\
\quad \text{initialized type reduction parameters } m_l \text{ and } m_r \\
\quad \text{initialized threshold } \theta \\
\quad \text{1: } \Delta P \leftarrow 0 \\
\quad \text{2: } \text{calculate } P \text{ based on (25)} \\
\quad \text{3: } \text{for iter } \leftarrow 1 \text{ to maxIter do} \\
\quad \text{4: for } k \leftarrow 1 \text{ to } N \text{ do} \\
\quad \text{5: } \phi_i \leftarrow 1, \phi_i \leftarrow 1 \\
\quad \text{6: for } i \leftarrow 1 \text{ to } R \text{ do} \\
\quad \text{7: for } j \leftarrow 1 \text{ to } n \text{ do} \\
\quad \text{8: calculate } \mu_{i,j}(X_k) \text{ and } \bar{\mu}_{i,j}(X_k) \text{ based on (2) and (3)} \\
\quad \text{9: } \bar{\phi}_i \leftarrow \bar{\phi}_i \times \bar{\mu}_{i,j}(X_k) \\
\quad \text{10: } \phi_i \leftarrow \phi_i \times \mu_{i,j}(X_k) \\
\quad \text{11: end for} \\
\quad \text{12: end for} \\
\quad \text{13: for } i \leftarrow 1 \text{ to } R \text{ do} \\
\quad \text{14: calculate } f_i(X_k) \text{ and } \bar{f}_i(X_k) \text{ based on (6) and (7)} \\
\quad \text{15: end for} \\
\quad \text{16: calculate the output boundaries } [y_l, y_u] \text{ based on (8)} \\
\quad \text{17: calculate the estimated output } y \text{ based on (9)} \\
\quad \text{18: calculate } J \text{ using (26) to (31)} \\
\quad \text{19: calculate } \Lambda \text{ and } \Lambda^\prime \text{ using (20) and (21)} \\
\quad \text{20: update } P \text{ using (32)} \\
\quad \text{21: end for} \\
\quad \text{22: end for} \\
\quad \text{Algorithm 2 Fine-tuning learning algorithm.}
\]
where vector $P$ contains all adjustable parameters of the method and $ΔP$ is the teaching signal. Our proposed quasi-Levenberg-Marquardt (qLM) method determines the teaching signal $ΔP$ based on minimizing the Cross-Entropy loss function. To extract this teaching signal, suppose that the input space has $n$ input variables and the training dataset consists of $N$ training instances in the form of $\{X_k, y_k\} (k = 1, 2, \ldots, N)$, where $X_k$ is the $k^{th}$ training input vector, and $y_k^*$ is its desired output (label). In our binary classification problem, $y_k^*$ is a value in the range $[0,1]$ that indicates the probability of belonging to the patient class. The estimated output value of the network for $k^{th}$ is indicated as $\hat{y}_k$. Now, we define $\hat{y} = [\hat{y}_1, \ldots, \hat{y}_k, \ldots, \hat{y}_N]^T$ as the vector containing network’s predictions and $\hat{Y}^* = [\hat{y}_1^*, \ldots, \hat{y}_k^*, \ldots, \hat{y}_N^*]^T$ as the desired output of them. The change of the network’s output vector ($\hat{Y}$) with respect to changes in networks’ parameter $P$ is estimated based on the Taylor’s expansion as follows:

$$
\begin{align*}
\hat{Y}^\text{new} & = \left[ F(X_1, P + ΔP), \ldots, F(X_N, P + ΔP) \right]^T \\
\hat{Y} & = \left[ F(X_1, P), \ldots, F(X_N, P) \right]^T \\
\hat{Y}^\text{new} & = \hat{Y} + J \times ΔP
\end{align*}
$$

(14)

where, $F$ is the network’s function, $\hat{Y}$ and $\hat{Y}^\text{new}$ are network’s prediction with parameters $P$ and $P + ΔP$, and $J$ is the Jacobian matrix defined as follows:

$$
J = \frac{∂\hat{Y}}{∂P} = \begin{bmatrix}
\frac{∂\hat{y}_1}{∂p_1} & \cdots & \frac{∂\hat{y}_1}{∂p_m} \\
\vdots & \ddots & \vdots \\
\frac{∂\hat{y}_N}{∂p_1} & \cdots & \frac{∂\hat{y}_N}{∂p_m}
\end{bmatrix}
$$

(15)

Here, $m$ is the number of parameters included in the vector $P$, and $p_i$ ($i=1,\ldots, m$) is the $i^{th}$ adjustable parameter.

Based on the Taylor’s expansion, the new network’s output vector $Y^\text{new}$ after applying the parameter change $ΔP$ on the parameter vector $P$ is approximated as follows:

$$
\hat{Y}^\text{new} = F(P + ΔP) \approx F(P) + J \times ΔP = \hat{Y} + J \times ΔP
$$

(16)

Therefore, the amount of parameter change is approximated by $J \times ΔP$.

We define the loss function $L$ as follows:

$$
L = -Y^{*T} \log(\hat{Y}^\text{new}) - (1 - Y^*)^T \log(1 - \hat{Y}^\text{new}) + η_j ΔP^T J^T J ΔP
$$

(17)

where, 1 is a vector contains 1s, $T$ is the symbol of transpose matrix, and $η$ is a constant positive penalty coefficient. This proposed loss function is composed of two parts: 1- the cross-entropy for a binary classification problem ($-Y^{*T} \log(\hat{Y}^\text{new}) - (1 - Y^*)^T \log(1 - \hat{Y}^\text{new})$), and 2- a regularizing term to limit the amount of change on network’s output ($η_j ΔP^T J^T J ΔP = η_j ∥J ΔP∥_2^2$). This penalty term tries to avoid applying dramatic changes on the parameters and network’s outputs.

Based on (16), the loss function $L$ (17) can be rewritten as follows:

$$
L = -Y^{*T} \log(\hat{Y}^\text{new}) - (1 - Y^*)^T \log(1 - \hat{Y}^\text{new}) + η\Delta P^T J^T J ΔP
$$

(18)

Moreover, by applying the Taylor’s expansion on the logarithm function, the loss function $L$ is rewritten as follows:

$$
L = -Y^{*T} (\log(\hat{Y}^\text{new}) + \Lambda^T J ΔP) - (1 - Y^*)^T (\log(1 - \hat{Y}^\text{new}) - \Lambda^T J ΔP) + η\Delta P^T J^T J ΔP
$$

(19)

where $\Lambda$ and $\Lambda'$ are two diagonal matrix as follows:

$$
\Lambda = \begin{bmatrix}
\frac{1}{y_1} & 0 & \cdots & 0 \\
0 & \frac{1}{y_2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{1}{y_N}
\end{bmatrix}
$$

(20)

$$
\Lambda' = \begin{bmatrix}
\frac{1}{1-y_1} & 0 & \cdots & 0 \\
0 & \frac{1}{1-y_2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{1}{1-y_N}
\end{bmatrix}
$$

(21)

Based on the optimality necessary condition [85], the optimal change parameter vector $ΔP$ is derived by solving the following equation:

$$
\frac{∂L}{∂ΔP} = 0
$$

(22)

Therefore, based on (21) and (22) we derive the optimal change parameter vector $ΔP$ as follows:

$$
\frac{∂L}{∂ΔP} = -J^T J ΔP + η J^T J ΔP = 0
\Rightarrow ΔP = \frac{1}{η} (J^T J)^{-1} J^T ((\Lambda + \Lambda') Y^* - \Lambda Y^*)
$$

(23)

To avoid singularity of the matrix $J^T J$, we can add a diagonal identity matrix $I$. Consequently, the final teaching signal $ΔP$ is derived as follows [57, 58, 62]:

$$
ΔP = \frac{1}{η} (J^T J + λI)^{-1} J^T ((\Lambda + \Lambda') Y^* - \Lambda Y^*)
$$

(24)

where, $λ$ is a positive scalar.

The adjustable parameters of the proposed model with $R$ fuzzy rules include center of fuzzy sets $c_{ij}$ ($i = 1, 2, ..., R$ and $j = 1, 2, ..., 10$), consequent parts parameter $y_i$ ($i = 1, 2, ..., R$), type-reduction parameters $m_j$ and $m_*$, and the threshold value $θ$. It is assumed that the upper and lower width parameters $σ$ and $π$ are constant and predefined. Therefore
the number of parameter is \( m = 10R + R + 2 + 1 = 11R + 3 \).
We define the parameter vector \( P \) as follows:

\[
P = [c_{1,1}, c_{1,2}, \cdots, c_{1,n}, c_{2,1}, \cdots c_{R,n}, y_1, \cdots y_R, m_1, m_r, \theta]^T
\]

(25)

where \( n \) is the number of input variables and in our problem \( n = 10 \). Consequently, the Jacobian matrix \( J \) defined in (15) is rewritten as follows:

\[
J = \frac{\partial \hat{Y}}{\partial P} = \begin{bmatrix}
\frac{\partial \hat{y}_1}{\partial c_{1,1}} & \frac{\partial \hat{y}_1}{\partial c_{1,2}} & \cdots & \frac{\partial \hat{y}_1}{\partial c_{1,n}} & \frac{\partial \hat{y}_1}{\partial c_{2,1}} & \cdots & \frac{\partial \hat{y}_1}{\partial c_{R,n}} & \frac{\partial \hat{y}_1}{\partial y_1} & \cdots & \frac{\partial \hat{y}_1}{\partial y_R} \\
\frac{\partial \hat{y}_2}{\partial c_{1,1}} & \frac{\partial \hat{y}_2}{\partial c_{1,2}} & \cdots & \frac{\partial \hat{y}_2}{\partial c_{1,n}} & \frac{\partial \hat{y}_2}{\partial c_{2,1}} & \cdots & \frac{\partial \hat{y}_2}{\partial c_{R,n}} & \frac{\partial \hat{y}_2}{\partial y_1} & \cdots & \frac{\partial \hat{y}_2}{\partial y_R} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial \hat{y}_n}{\partial c_{1,1}} & \frac{\partial \hat{y}_n}{\partial c_{1,2}} & \cdots & \frac{\partial \hat{y}_n}{\partial c_{1,n}} & \frac{\partial \hat{y}_n}{\partial c_{2,1}} & \cdots & \frac{\partial \hat{y}_n}{\partial c_{R,n}} & \frac{\partial \hat{y}_n}{\partial y_1} & \cdots & \frac{\partial \hat{y}_n}{\partial y_R}
\end{bmatrix}
\]

(26)

To complete the learning method and apply the extracted rule presented in (24), the following derivatives should be calculated: \( \frac{\partial \hat{y}}{\partial c_{i,j}}, \frac{\partial \hat{y}}{\partial y_i}, \frac{\partial \hat{y}}{\partial ml}, \frac{\partial \hat{y}}{\partial mr}, \) and \( \frac{\partial \hat{y}}{\partial \theta} \) for \( i = 1, 2, \ldots, R, j = 1, 2, \ldots, n, \) and \( k = 1, 2, \ldots, N. \) These derivatives are determined based on applying the chain rule as follows:

\[
\frac{\partial \hat{y}}{\partial \theta} = -\hat{y}(1 - \hat{y})
\]

(27)

\[
\frac{\partial \hat{y}}{\partial ml} = \hat{y}(1 - \hat{y})f_{i,j}y_i
\]

(28)

\[
\frac{\partial \hat{y}}{\partial mr} = \hat{y}(1 - \hat{y})f_{r,i}y_i
\]

(29)

\[
\frac{\partial \hat{y}}{\partial y_i} = \hat{y}(1 - \hat{y})\left( m_r f_{i} + m_l f_{l,i} \right)
\]

(30)

\[
\frac{\partial \hat{y}}{\partial c_{i,j}} = \hat{y}(1 - \hat{y})\left( m_r (y_i - y_{r,i}) f_{i,j} \sigma_x^2 + m_l (y_i - y_{l,i}) \frac{f_{i,j}}{\sigma_x^2} \right) (x_j - c_{i,j})
\]

(31)

Finally, inspired by the Momentum idea, to accelerate the learning method and increase its robustness against the local optima [84], our final learning law is proposed as follows:

\[
\Delta P = \beta \Delta P + \frac{1 - \beta}{\eta} (J^T J + \lambda I)^{-1} J^T ((A + \Lambda') Y^* - \Lambda' 1) \\
\]

\[
P = P + \Delta P
\]

(32)

where \( \beta \) is multiplier in the range \((0,1)\) for considering the effect of the previous changing vectors. The fine-tuning algorithm is presented in Algorithm 2.

### 3.3.3 Complementary online learning

The proposed learning method is a kind of batch learning approach based on clustering all samples, which assumes that all training instances are available at once. This is partially true, that we should have training examples for training the fuzzy neural network at first. However, it is possible that more studies may gather more examples in the future. Therefore, to avoid learning the network from scratch, here we propose a complementary learning approach to enable to handle the new training instances whenever it is necessary.

The proposed complementary online learning approach is based on investigating the amount of current fuzzy rules’ coverage encountered by new instances [59, 60]. If the coverage of existing fuzzy rules is not sufficient to cover a new example, a new fuzzy rule is added to the Fuzzy Neural Network to cover an area consisting of this new sample and future similar instances. Indeed, if the sum of fuzzy rules’ firing strength is lower than a predefined threshold, a new fuzzy rule based on this new sample is

---

**Algorithm 3** Complementary online learning algorithm.
Fig. 6 An illustrative example of adding a new fuzzy rule in a two-dimensional input space. First, the input vector $x_{\text{new}}$ is analyzed by two existing fuzzy rules with centers $c_1$ and $c_2$. The total coverage for the sample $x_{\text{new}}$ is lower than a predefined threshold $\theta_c$. Consequently, a new fuzzy rule, which its center is $x_{\text{new}}$, has been added to the network. The center of this newly added fuzzy rule would be the arriving new instance and the width would be lower than the previous fuzzy rules to decrease the conflict with previous fuzzy rules and only cover the region with a low amount of coverage. The consequent part of this new fuzzy rule is the desired output of the newly arrived instance. Figure 6 shows an illustrative example of adding a new fuzzy rule in a two-dimensional space and Algorithm 3 summarizes this complementary online learning. The paradigm of complementary online learning is shown in Fig. 7.

4 Experimental results

In this section, the performance of the proposed method is evaluated and compared to some previous methods. First, the utilized data is explained in Section 4.1. Next, the evaluation metrics used for assessing the performance of the method are introduced in Section 4.2. Next, the performance of the proposed paradigm is compared to the performance of some previous supervised and unsupervised machine learning approaches applied to the similar data in the literature in Section 4.3. Afterward, the effectiveness of the proposed complementary online learning method is evaluated in Section 4.4. To investigate the effectiveness of using Interval Type-2 Fuzzy Logic, the performance of the proposed method is compared with the performance of a similar type-1 fuzzy neural network encountering noisy data in Section 4.5. Moreover, the effectiveness of the proposed fine-tuning approach is investigated in Section 4.6. Next, the performance of the proposed method is compared with some recent deep learning approaches in Section 4.7. Afterward, the final fuzzy sets and rules extracted by the proposed method are reported in Section 4.8. These reported fuzzy rules can be used by experts to: 1- evaluate
the extracted fuzzy rules and fine-tune them, and 2- apply them to classify patients and healthy persons. At the end of this section, an illustrative example is presented to better show the workflow of the proposed method. We divide each experiment into two parts: first, the main purpose of performing the experiment is presented in part “Description and Aims”. Next, the results are analyzed through the “Results and Discussion”. All experiments have been conducted in MATLAB 2018b runtime environment on an Intel Core-i3 CPU with clock speed 2.1 GHz and 8 GB RAM running the Windows 10 professional operating system.

4.1 Data

We utilized the gait cycle data provided by PhysioNet. This data aggregates three similar sub-datasets obtained in different experimental studies from different subjects [24, 52, 53]. The dataset includes vGRF obtained from 16 sensors placed under the feet of control and patient subjects (8 per foot) as shown in Fig. 1. Overall, the data was recorded from 93 patients with PD and 72 healthy control subjects. The vertical ground reaction force was recorded for subjects as they walked at their usual, self-selected pace for approximately 2 minutes on level ground. Features are extracted from two sequences, indicating the average vGRF of each foot. The details of each dataset are outlined in Table 3.

4.2 Evaluation metrics

To evaluate the performance of the proposed method, the usual metrics utilized in the classification tasks are used. One of the most popular metrics in classification tasks is “Accuracy” that shows the average performance of the method to properly classify instances and is defined as follows:

\[
\text{Accuracy} = \frac{TP + TF}{TP + TN + FP + FN} \quad (33)
\]

where TP (“True Positive”) is the number of instances that belongs to the positive class (here PD patient) classified correctly, TN (True Negative) is the number of instances belongs to the negative class (here healthy persons) classified correctly, FP (False Positive) is the number of instances belongs to the negative class but classified wrongly, and finally, FN (False Negative) is the number of instances belonging to the positive class but classified wrongly.

In some applications like medical diagnosis, it is very important to avoid False Positives and also detect the patients correctly. Detecting a healthy person as a patient would lead to extra costs (e.g. additional experiments) and also disappointment for the person and his/her family. On the other hand, if the system detects a patient as a healthy person, it would be dangerous for his/her health. Therefore, to have a more precise evaluation, the following metrics are also utilized:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (34)
\]
\[
\text{Recall} = \frac{TP}{TP + FN} \quad (35)
\]
\[
F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (36)
\]

where, Precision evaluates the “False Positive” rate, Recall evaluates the ability of the method to classify patients correctly, and \(F_1\) score is a combination of Precision and Recall.

4.3 Comparison with previous methods

Description and aims In this experiment, we aim at comparing the performance of the proposed method with the some previous methods that utilize similar clinical features in [1]. Moreover, we show the effectiveness of the proposed fine-tuning method by comparing the performance of the proposed method before and after applying the proposed quasi-Levenberg-Marquardt (qLM) learning algorithm. Following [1], each sub-dataset is divided into training and testing sets according to the leave-one-out cross-validation procedure. For parameter setting, we should determine the number of fuzzy rules and the width of fuzzy sets. The width of fuzzy sets is considered in the range [0.1, 0.2]. The maximum number of iterations in the fine-tuning phase set to 100 and the learning rate parameters \(\eta\), \(\beta\), and \(\lambda\) are set to 1, 0.3, and 10 respectively. To determine the number of fuzzy rules, the performance of the method is evaluated for a different number of fuzzy rules for 10 independent runs and then we have calculated the mean and the standard deviation (STD) of the F1 score for a different number of fuzzy rules. Afterward, the lowest number of fuzzy rules with a higher mean and lower STD

| Data | Subjects | Total subjects | Female | Male |
|------|----------|----------------|--------|------|
| Ga [24] PD | 29 | 9 | 20 |
| | CO | 18 | 8 | 10 |
| Si [52] PD | 35 | 13 | 22 |
| | CO | 29 | 11 | 18 |
| Ju [53] PD | 29 | 13 | 16 |
| | CO | 25 | 14 | 12 |
is chosen for each dataset (see Fig. 8). Based on Fig. 8, the chosen numbers of fuzzy rules are listed in Table 4.

**Results and discussion** Tables 5, 6 and 7 evaluate the performance of the method based on different criteria in different datasets introduced in Table 3. According to the results reported in Tables 5 to 7, the proposed method outperforms all other supervised and unsupervised machine learning approaches trained with similar features in the sub-dataset Ga, and Si. In sub-dataset Ju, two supervised machine learning approaches, KNN and SVM, have better Accuracy, but the proposed method has better Precision, Recall, and F1 Score values. Indeed, the proposed method has a lower False Negative rate which is very important in a clinical application. Moreover, by comparing the performance of the proposed method before and after applying the fine-tuning (the first and the second columns in Tables 5, 6 and 7), the effectiveness of this novel learning method is shown.

Furthermore, the detailed performance of the proposed method is compared with the previous approaches in Tables 8, 9 and 10 by showing the “True Positive” and “True Negative” rates. The proposed method performs worse than KNN in the Ju sub-dataset in detecting healthy control subjects. The reason could be low coverage of healthy input space by the gathered data in the Ju sub-dataset. Therefore, there are not sufficient healthy instances to cover the input space efficiently. Subsequently, the KNN approach that saves all instances and votes in a neighborhood around the test example performs better than the proposed method in this sub-dataset.

Finally, the performance of the proposed method trained by using the whole dataset is compared with some other approaches in Table 11. Based on these results, the proposed method with the proposed learning approach outperforms the other models.

### 4.4 Effectiveness of the online learning

**Description and aims** The purpose of this experiment is to investigate the effectiveness of the proposed complementary online learning in the case of providing supplementary instances. To evaluate the effectiveness of the proposed complementary online learning, we assume that only one sub-set is available as the initial training dataset. After learning the Interval Type-2 Fuzzy Neural Network and extracting the fuzzy rules based on this available initial dataset, another sub-dataset is added as new training samples. We assume that the dataset Ga [24] is available, and after training, the samples of dataset Ju will be added. Here, the fine-tuning method is not applied. The threshold for investigating the amount of fuzzy rules’ coverage is set to 0.1. The $\epsilon$ parameter in Algorithm 3 is set to 1.

**Results and discussion** Table 12 of the proposed method before and after adding new instances. The first row in Table 12 (“Trained using Ga”) reports the number of fuzzy rules along with their performance in different sub-datasets after training based on sub-dataset Ga. The second row of Table 12, shows the changes after applying the complementary online learning based on the new sub-dataset Ju. It is shown that the number of fuzzy

---

**Table 4** Chosen number of fuzzy rules based on studies reported in Fig. 8

| Data | Ga | Ju | Si |
|------|----|----|----|
| Number of fuzzy rules | 8  | 4  | 3  |
### Table 5  Comparison of the performance of the proposed method with other related approaches studied in [1] in the case of the Yogev et al. (Ga) sub-datasets

| Performance metric | Propose method without fine-tuning | Propose method with fine-tuning | KNN [1] | CART [1] | RF [1] | NB [1] | SVM [1] | K-Means [1] | GMM [1] |
|--------------------|----------------------------------|--------------------------------|---------|----------|--------|--------|---------|-------------|---------|
| Accuracy           | 92.92                            | 98.23                          | 86.05   | 83.72    | 86.05  | 74.42  | 86.05   | 63.72       | 64.77   |
| Precision          | 92.40                            | 97.40                          | 84.89   | 82.86    | 85.07  | 74.73  | 84.90   | 64.34       | 62.63   |
| Recall             | 97.33                            | 100.00                         | 86.34   | 81.94    | 85.07  | 76.45  | 85.71   | 65.31       | 62.91   |
| F1 Score           | 94.80                            | 98.68                          | 85.61   | 82.40    | 85.07  | 75.58  | 85.30   | 64.82       | 62.77   |

The bold entries represents the best value in each comparison.

### Table 6  Comparison of the performance of the proposed method with other related approaches studied in [1] in the case of the Hausdordd et al. (Ju) sub-datasets

| Performance metric | Propose method without fine-tuning | Propose method with fine-tuning | KNN [1] | CART [1] | RF [1] | NB [1] | SVM [1] | K-Means [1] | GMM [1] |
|--------------------|----------------------------------|--------------------------------|---------|----------|--------|--------|---------|-------------|---------|
| Accuracy           | 85.83                            | 90.06                          | 90.91   | 84.30    | 87.60  | 77.69  | 90.08   | 55.12       | 65.12   |
| Precision          | 88.89                            | 92.63                          | 85.35   | 82.34    | 89.41  | 70.64  | 89.31   | 52.58       | 57.95   |
| Recall             | 93.62                            | 93.62                          | 88.35   | 64.96    | 71.48  | 78.54  | 78.96   | 53.91       | 61.29   |
| F1 Score           | 91.19                            | 93.27                          | 86.83   | 72.62    | 79.45  | 74.38  | 83.82   | 53.24       | 59.57   |

The bold entries represents the best value in each comparison.
Table 7  Comparison of the performance of the proposed method with other related approaches studied in [1] in the case of the Frenkel-Toledo et al. (Si) sub-datasets

| Performance metric | Propose method without fine-tuning | Propose method with fine-tuning | KNN [1] | CART [1] | RF [1] | NB [1] | SVM [1] | K-Means [1] | GMM [1] |
|--------------------|-----------------------------------|--------------------------------|---------|---------|-------|-------|---------|------------|--------|
| Accuracy           | 83.33                             | 93.33                          | 81.25   | 79.69   | 82.81 | 79.69 | 82.81   | 57.19      | 65.31  |
| Precision          | 85.71                             | 94.28                          | 81.43   | 79.56   | 83.10 | 79.69 | 82.81   | 61.07      | 64.95  |
| Recall             | 85.71                             | 94.28                          | 81.67   | 79.36   | 82.22 | 79.95 | 83.10   | 59.23      | 64.62  |
| F1 Score           | 85.71                             | 94.28                          | 81.55   | 79.46   | 82.65 | 79.82 | 82.96   | 60.14      | 64.78  |

The bold entries represents the best value in each comparison

Table 8  Global confusion matrix obtained using different classifiers in the case of the Yogev et al. (Ga) sub-dataset (H: Healthy, P: Patient)

| Obtained classes | Proposed method | KNN [1] | CART [1] | RF [1] | NB [1] | SVM [1] | K-means [1] | GMM [1] |
|------------------|-----------------|---------|---------|-------|-------|---------|------------|--------|
|                  | H               | P       | H       | P     | H     | P       | H          | P      |
| True             | H               | 94.74   | 5.26    | 87.50 | 12.50 | 75.00   | 25.00      | 81.25  |
| Classes          | P               | 0       | 100     | 14.81 | 85.19 | 11.11   | 88.89      | 31.48  |

The bold entries represents the best value in each comparison
Table 9  Global confusion matrix obtained using different classifiers in the case of the Hausdorff et al. (Ju) sub-dataset (H: Healthy, P: Patient)

| Obtained classes | Proposed method | KNN [1] | CART [1] | RF [1] | NB [1] | SVM [1] | K-Means [1] | GMM [1] |
|------------------|-----------------|---------|----------|--------|--------|---------|-------------|--------|
| True H           |                 | H       | P        | H      | P      | H       | P           | H      | P      | 73.08 | 26.92 | 84.00 | 16.00 | 32.00 | 68.00 | 44.00 | 56.00 | 80.00 | 20.00 | 60.00 | 40.00 | 51.84 | 48.16 | 54.76 | 45.24 |
| Classes P        |                 | 6.38    | 93.62    | 7.29   | 92.71  | 2.08    | 97.92      | 1.04   | 98.96 | 22.92 | 77.08 | 2.08  | 97.92 | 44.02 | 55.98 | 32.19 | 67.81 |

Table 10  Global confusion matrix obtained using different classifiers in the case of the Frenkel-Toledo et al. (Si) sub-dataset (H: Healthy, P: Patient)

| Obtained classes | Proposed method | KNN [1] | CART [1] | RF [1] | NB [1] | SVM [1] | K-Means [1] | GMM [1] |
|------------------|-----------------|---------|----------|--------|--------|---------|-------------|--------|
| True H           |                 | H       | P        | H      | P      | H       | P           | H      | P      | 92.00 | 8.00  | 86.21 | 13.79 | 75.86 | 24.14 | 75.86 | 24.14 | 82.76 | 17.24 | 86.21 | 13.79 | 80.97 | 19.03 | 57.24 | 42.76 |
| Classes P        |                 | 5.71    | 94.29    | 22.86  | 77.14  | 17.14   | 82.86      | 11.43  | 88.57 | 22.86 | 77.14 | 20.00 | 80.00 | 62.51 | 37.49 | 28.00 | 72.00 |
Table 11 Comparison of the final performance of the proposed method with some previous approaches

| Method | Proposed method without fine-tuning | Proposed method with fine-tuning | [86] | [87] | [50] | [20] |
|--------|-------------------------------------|----------------------------------|------|------|------|------|
| Accuracy | 88.74 | **97.61** | 88.89 | 84.48 | 89.92 | 91.53 |

The bold entries represent the best value in each comparison.

---

Rules is increased and the performance of the method is improved for all sub-datasets after receiving new instances.

### 4.5 Effectiveness of interval type-2 fuzzy system against noisy data

**Description and aims** One advantage of Type-2 fuzzy systems is their robustness against noisy measurements. Moreover, the underlying problem depends on sensor measurements that face noise and uncertainty. Therefore, in this experiment we aim at investigating the robustness of the proposed Type-2 fuzzy system against noisy measurements.

To investigate the effectiveness of the proposed Interval Type-2 Fuzzy system against noisy data, its performance is compared with a type-1 fuzzy neural network applied to noisy test samples. In this experiment, all samples from sub-datasets Ga, Ju, and Si are gathered together to form one dataset. Next, it is divided into training and testing sets according to the leave-one-out cross-validation procedure. We add a Gaussian noise with different width values of 0.1 and 0.3 to the test samples. The number of fuzzy rules is considered to be 10. To compare just the effect of using Interval Type-2 fuzzy sets instead of Type-1 ones, the fine-tuning approach is not applied.

**Results and discussion** Table 13 shows the effectiveness of the proposed Interval Type-2 Fuzzy Neural Network, comparing its performance with a Type-1 Fuzzy Neural Network. According to this table, by increasing the strength of the noise effect from 0.1 to 0.3, the difference between the performance of the Type-1 model and the Interval Type-2 method increases in terms of different criteria for the two datasets Si and Ju. However, the Type-1 model performs agreeable in some cases, but in most cases, the Interval Type-2 model outperforms the Type-1 method in noisy states.

### 4.6 Efficiency of the proposed learning method

**Description and aims** According to the reported results of Tables 5, 6 and 7 along with Table 11, the efficiency of the proposed fine-tuning approach is revealed by comparing the performance of the proposed method after and before applying it. However, these observations show the importance of applying the fine-tuning. In this experiment, we aim at studying the efficiency of the proposed qLM method by comparing its performance with other local search methods, including: 1-Levenberg-Marquardt (LM), 2-Stochastic Gradient Descent (SGD), and 3-Momentum approach.

**Results and discussion** Figure 9 shows the proposed learning approach (qLM) with other mentioned learning methods based on different criteria in three datasets Ga, Ju, and Si, separately. It is shown that in all cases, the performance of the proposed method (the blue bar) is better than the other methods. In dataset Ju, the recall of the SGD is higher than the recall of the proposed method. However,
Table 13  Evaluating the effectiveness of interval type-2 fuzzy neural network against noisy data

| Noise Number of fuzzy rules | Data | Method | Accuracy | Precision | Recall | F1 score |
|-----------------------------|------|--------|----------|-----------|--------|----------|
| 0.1 10                      | Ga IT2 | 91.96  | 93.42    | 94.66     | 94.04  |
|                             | T1    | 89.28  | 89.87    | 94.66     | 92.20  |
|                             | Ju IT2 | 89.74  | 91.83    | 95.74     | 93.75  |
|                             | T1    | 86.32  | 88.23    | 95.74     | 91.83  |
|                             | Si IT2 | 81.25  | 78.04    | 91.42     | 84.21  |
|                             | T1    | 81.25  | 76.74    | 94.28     | 84.61  |
| 0.3 10                      | Ga IT2 | 83.03  | 87.83    | 86.66     | 87.24  |
|                             | T1    | 84.82  | 88.15    | 89.33     | 88.74  |
|                             | Ju IT2 | 83.76  | 92.13    | 87.23     | 89.61  |
|                             | T1    | 80.34  | 87.36    | 88.29     | 87.83  |
|                             | Si IT2 | 76.56  | 79.41    | 77.14     | 78.26  |
|                             | T1    | 67.18  | 69.44    | 71.42     | 70.42  |

the performance of the proposed method is significantly better based on the other criteria. Thus, the SGD is stuck at a local optimum and has a drift to the positive class.

4.7 Comparison with deep learning-based methods

Description and aims Recently, different architectures of deep neural networks have been used for PD patients’ diagnosis based on analyzing their gait cycle [11, 13–15]. For a more comprehensive comparison, the final results of the proposed method are compared with those of some recently published deep-learning-based methods.

It is worthy to note that although these methods have used the same data for training to learn a huge number of their deep structures’ parameters, they have to increase the number of training instances by segmenting the vGRF sequences into very small pieces (10 to 20 steps). Since the main purpose of the proposed method is the interpretability and the input variables are high-level expert-understandable clinical features related to the statistics of the different stages of the gait (including stride, stance, and swing), it is not possible to segment the sequences into smaller pieces. Therefore, the comparison is not fair but could show the pros and cons of the proposed method.

Results and discussion Table 14 compares the final accuracy, the number of parameters (network’s size), and the number of learning epochs of the proposed method with some deep structures. It is shown that the accuracy of the proposed method after applying the fine-tuning is very close

![Fig. 9](image-url)  Comparison of the performance of the proposed learning method (qLM) with Levenberg-Marquardt (LM), Stochastic Gradient Descent (SGD), and the Momentum learning approaches
Table 14 Comparison between the proposed method and recent methods based on deep structures

| Method                        | Number of parameters | Number of training samples | F1     | accuracy | Interpretability | Number of training epochs | Online learning | Learning time  |
|-------------------------------|----------------------|---------------------------|--------|----------|------------------|---------------------------|----------------|---------------|
| 1D-CNN [13]                  | 857120               | 58023                     | 85.70% | 98.90%   | False            | 30 True                   | 60 False        | 60 False |
| CNN with LSTM [14] (in parallel) | 665057               | 29763                     | N/A    | N/A      | False            | 100 False                 | 300000 False    | 100 False |
| CNN with LSTM [15] (sequentially) | 89201292             | 19249                     | 98.61% | 99.22%   | False            | 100 True                  | True           | True |
| Proposed method               | 110                  | 165                       | 98.74% | 98.30%   | True             | 100 True                  | True           | True |
| Proposed method               | 110                  | 165                       | 98.74% | 98.30%   | True             | 100 True                  | True           | True |

Table 14: Comparison between the proposed method and recent methods based on deep structures.

The performance of the deep neural networks. However, it is shown that the methods using deep neural networks have a huge number of parameters that must be determined using a supervising learning approach based on a training dataset including lots of instances. Since the application is related to people suffering from a special disease (PD), providing such a dataset is too difficult. However, the proposed method has a few parameters that must be determined. The gait cycle of each person should be segmented into smaller pieces for these deep networks, but the proposed method assumes the whole sequence of walking belongs to each subject.

Furthermore, the represented knowledge in these deep structures cannot be expanded by gathering new instances, and it is necessary to redo the training from the scratch. However, the proposed method has the ability of online learning to improve the knowledge based on new training instances.

Finally, the proposed method is both explainable and interpretable. Therefore, the supporting reasoning for its decision is clear to an expert. Interpretability is important in many machine learning applications, including healthcare [88, 89]. This ability is helpful in medical decisions related to people’s healthcare.

4.8 Final extracted fuzzy rules

Description and aims As has been mentioned, one of our motivations for utilizing the Mamdani Interval Type-2 Fuzzy Neural Network is its interpretable structure. Indeed, this model extracts human-understandable rules that can be used by experts to validate its decisions. Moreover, the extracted rules can be evaluated and modified by experts. Therefore, in this experiment, we aim at studying the final extracted fuzzy rules. Here, all the available samples are fed to the proposed Interval Type-2 Fuzzy Neural Network to extract final fuzzy rules based on the currently available data. The number of fuzzy rules is set at 9. The width of fuzzy sets is considered in the range [0.01, 0.1]. Indeed, it is assumed that each fuzzy set could be very uncertain ($\sigma = 0.1$) or close to a precise value ($\sigma = 0.01$) for decision making.

Results and discussion Figure 10 shows the extracted fuzzy rules. In this figure, each row indicates a fuzzy rule and each column indicates a feature. The center of the fuzzy sets used for constructing each rule is shown in each cell, and the color of each cell represents the sense of the linguistic variable (higher values are near to red and lower values are near to white). The last column shows the consequent parts’ parameters, which is the proposed label of each fuzzy rule (+1’ for patients (colored red) and ‘−1’ for healthy subjects (colored blue). The explanations of features $x_1$ to
Fig. 10  Extracted fuzzy rules. Each row indicates a fuzzy rule ($R_1$ to $R_9$) and each column except the last indicates a feature ($x_1$ to $x_{10}$). The last column presents each fuzzy rule’s output value as the consequent part’s parameter (‘+1’ for patients colored red) and ‘−1’ for healthy subjects (colored blue). The center of fuzzy sets used for constructing each rule is shown in each cell and the color of each cell represents the sense of the linguistic variable (higher values are near to red and lower values are close to white).

$x_{10}$ have already been presented in Table 2. Studying this table can reveal the importance of different variables and their correlations. For example, comparing Rule “R3” with other rules indicates that a person with a high value (colored red) of “Short Swing Time” ($x_1$) along with a low value (colored yellow) of “Long Swing Time” ($x_2$) is classified as a “Patient”. These lingual values for these features describe a subject with very short steps, which can be a symptom of Parkinson’s Disease. However, rule “R7” describes a subject with high values (colored red) for short and long swing...
Table 15  Final performance of the proposed method trained using all available data

| Method                  | Accuracy | Precision | Recall | F1 Score |
|-------------------------|----------|-----------|--------|----------|
| Without fine-tuning     | 88.74    | 89.41     | 95.10  | 92.16    |
| After fine-tuning       | 97.61    | 97.58     | 99.02  | 98.30    |

The bold entries represent the best value in each comparison.

times \((x_1 \text{ and } x_2)\) along with variations in the swing time \((x_6)\). This subject, which is considered as a healthy person, indicates various types of paces, from short to long. An expert can interpret the extracted rules in a similar manner.

Moreover, Fig. 11 shows extracted fuzzy sets for a selected set of features (six out of ten). As this Figure shows, based on the available data and the importance of different ranges of values, the density of extracted fuzzy sets in different regions of features is varied.

Finally, the performance of the method trained by using the whole dataset is summarized in Table 15 in two modes: without applying the proposed fine-tuning approach, and with applying it. According to these results, the proposed learning approach applied as the fine-tuning method increases the performance of the proposed model significantly.

4.9 Illustrative example

To better show the workflow of the model, an illustrative example is presented to classify two subjects, a patient and a healthy person, based on the extracted fuzzy rules. Considering the great number of rules extracted in Section 4.8, an example is presented for the dataset Si with 3 fuzzy rules. Table 16 shows the antecedent and consequent parts’ parameters of the network learned based on this dataset. In this example, it is assumed that \(\sigma \in [0.1, 0.2]\) and \(m_r = m_I = \theta = 0.5\).

Two samples from Patient and Healthy classes are chosen as shown in Table 17:

After calculating the upper and lower membership values of different fuzzy sets based on (2) and (3), the normalized upper and lower firing strength values of different fuzzy rules are computed based on (6) and (7) as shown in Table 18:

Based on the reported results in Table 18, it is comprehended that the healthy instance \((s_1)\) majorly fires the second rule with consequence \(-1\) (interpreted as “is Healthy”) and the patient \((s_2)\) majorly fires two rules \(r_1\) and \(r_2\) with consequence \(+1\) (interpreted as “is Patient”).

To derive the final decision for these two instances, the boundaries of the network’s output \(([y_l, y_r])\) and its final output \((y)\) are calculated based on (8) and (9) as shown in Table 19:

We interpret the final decision for the first instance as “Healthy” because the network’s output is lower than 0.5 (0.32). The network’s decision on the second sample is interpreted as “Patient” because it is higher than 0.5 (0.66).

5 Conclusions

In this paper, a classifier using an explainable self-organizing interval type-2 fuzzy neural network is proposed to diagnose patients suffering from Parkinson’s Disease (PD). The proposed method analyzes the gait cycle of subjects recorded in the form of vertical Ground Reaction Force (vGRF) signals. To have an interpretable system, 10 clinical features are extracted from the vGRF signal. To overcome the uncertainty and noisy measurements of sensors, the fuzzy neural network is built upon interval type-2 fuzzy logic. Indeed, the benefits of using a self-organizing interval type-2 fuzzy neural network can be summarized as follows:

- Providing a transparent structure with interpretable rules based on fuzzy logic that is useful for clinical applications;
- Considering the effect of noise and uncertainty is beneficial when encountering noisy data measured by sensors;
- Presenting an explainable structure with clear reasoning to find why an individual is classified as a patient or a healthy person;
- Providing the learning ability of a neural network;

| Rule index | \(c_1\) | \(c_2\) | \(c_3\) | \(c_4\) | \(c_5\) | \(c_6\) | \(c_7\) | \(c_8\) | \(c_9\) | \(c_{10}\) | \(y\) |
|------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|------|
| r1         | 0.63   | 0.59   | 0.21   | 0.69   | 0.60   | 0.20   | 0.26   | 0.33   | 0.21   | 0.14    | +1   |
| r2         | 0.64   | 0.44   | 0.02   | 0.25   | 0.54   | 0.25   | 0.24   | 0.16   | 0.22   | 0.03    | −1   |
| r3         | 0.88   | 0.75   | 0.02   | 0.78   | 0.81   | 0.21   | 0.24   | 0.25   | 0.20   | 0.19    | +1   |
• Providing the ability to learn new instances provided by the progress of studies based on the self-organizing structure.

To extract fuzzy rules, two paradigms are presented. First, using the available training dataset, initial interpretable fuzzy rules are extracted. An online learning approach is proposed to provide the ability to learn from new instances. This complementary online learning method investigates the performance and coverage of the current rule base. In the cases of wrong classification and low coverage, a new fuzzy rule is added to the rule base.

Moreover, to fine-tune the initialized parameters of the interval type-2 fuzzy neural network, a novel quasi-Levenberg-Marquardt (qLM) learning method is proposed. This new learning approach minimizes the Cross-Entropy loss function based on a trust-region optimization method and has the ability to avoid getting stuck at local optima in comparison to the Stochastic Gradient Descent (SGD) method.

The performance of the proposed method is evaluated on three datasets obtained from different subjects [24, 52, 53] and compared with other supervised and unsupervised machine learning approaches using similar clinical features. Based on these results, the proposed method has the best performance.

Moreover, the performance of the model in the presence of noise is evaluated and compared with a Type-1 fuzzy neural network. Furthermore, the effectiveness of the online learning is investigated by adding new training instances after finishing the batch learning. This experiment shows an obvious improvement in the network’s performance after applying the proposed online learning.

In addition, the effect of the proposed qLM learning approach is studied by comparing the performance of the method after and before applying the proposed fine-tuning algorithm. Based on performed comparisons with other learning approaches, including the SGD, Levenberg-Marquardt (LM), and Momentum, it is revealed that the proposed method could better avoid getting stuck at local optima. It is worth mentioning that the proposed novel learning algorithm is a general-purpose algorithm that could be applied to other similar classification problems.

The performance and structure size of the proposed method are compared with those of some deep learning-based methods. Based on these comparisons, the proposed method’s performance is very close to that of these deep structures. However, its number of parameters is significantly lower than the deep architectures. Moreover, the proposed method utilizes human-understandable clinical features and forms interpretable rules while the deep structures are not explainable. Furthermore, the proposed method has the ability to expand its knowledge by encountering new instances.

Finally, the extracted fuzzy sets of different features along with the learned fuzzy rules are reported to show the interpretability of the model. Based on the extracted fuzzy rules, it is shown that the value of some features could better guide an expert to classify patients and healthy people. For example, if the average value of swing time percentage ($x_4$) is high or very low, the person would be a patient. It is reasonable because “very low” values show that the person’s step size is short and “very high” values reveal that the person is walking slowly. Indeed both of them are caused by slowness. Moreover, we can better study the correlations among input variables. For example, a high value of “Short Swing Time” ($x_1$) along with a low value of “Long Swing Time” ($x_2$) can determine a “Patient” with very short steps, which can be a symptom of Parkinson’s Disease.

### Table 17 Two samples chosen from Patient (labeled +1) and Healthy (labeled -1) classes

| Sample index | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | Label |
|--------------|------|------|------|------|------|------|------|------|------|---------|-------|
| s1           | 0.87 | 0.71 | 0    | 0.17 | 0.79 | 0.12 | 0.06 | 0.09 | 0.0528 | 0.1672  | −1    |
| s2           | 0.64 | 0.44 | 0.02 | 0.25 | 0.54 | 0.25 | 0.24 | 0.16 | 0.22  | 0.03    | +1    |

### Table 18 Normalized upper and lower firing strength values for different rules

| Sample index | $r_1$ | $r_2$ | $r_3$ |
|--------------|------|------|------|
|              | Upper | Lower| Upper | Lower| Upper | Lower|
| s1           | 0     | 0.31 | 1    | 0.89 | 0     | 0.07 |
| s2           | 0.99  | 0.81 | 0.002| 0.18 | 0     | 0.01 |
Table 19 Network’s output boundaries and final outputs for two instances

| Sample index | $y_l$ | $y_r$ | $y$  |
|--------------|-------|-------|-----|
| s1           | -0.8  | -0.6  | 0.32|
| s1           | 0.5   | 0.8   | 0.66|

In summary, the final Accuracy, Precision, Recall, and $F_1$ score of the proposed method on Ga dataset are 98.23%, 97.40%, 100.00%, 98.68%, and on Ju dataset are 90.06%, 92.63%, 93.62%, 93.27%, and on Si dataset are 93.33%, 94.28%, 94.28%, and 94.28% respectively.

For the future work of this study, the following trends are proposed:

- Presenting a multi-modal model to investigate different aspects of Parkinson’s disease related to different data modalities at the same time to improve the precision of the current model, including gait cycle changing (vGRF), general movement disorders (data from Inertial Measurement Unit (IMU)), and speech disorders (vocal impairment);
- Combining the current explainable method with a deep representation learning approach to extract new interpretable features from raw data;
- Expanding the proposed method from Interval Type-2 fuzzy system to a general Type-2 one to improve its robustness against noisy measurements;
- Using meta-heuristic optimization methods like Evolutionary Algorithms and Swarm Intelligence methods for fine-tuning.

References

1. Khoury N, Attal F, Amirat Y, Oukhellou L, Mohammed S (2019) Data-driven based approach to aid Parkinson’s disease diagnosis. Sensors 19(2):242
2. Hariharan M, Polat K, Sindhu R (2014) A new hybrid intelligent system for accurate detection of Parkinson’s disease. Comput Methods Programs Biomed 113(3):904–913
3. Pan S, Iplikci S, Warwick K, Aziz TZ (2012) Parkinson’s disease tremor classification—a comparison between support vector machines and neural networks. Expert Syst Appl 39(12):10764–10771
4. Goyal J, Khandnor P, Aseri TC (2020) Classification, prediction, and monitoring of Parkinson’s disease using computer assisted technologies: a comparative analysis. Eng Appl Artif Intell 96:103955. https://doi.org/10.1016/j.engappai.2020.103955
5. Hall JE (2010) Guyton and hall textbook of medical physiology. Guyton physiology. Elsevier Health Sciences
6. Salimi-Badr A, Ebadzadeh MM, Darlot C (2017) A possible correlation between the basal ganglia motor function and the inverse kinematics calculation. J Comput Neurosci 43(3):295–318
7. Salimi-Badr A, Ebadzadeh MM, Darlot C (2018) A system-level mathematical model of Basal Ganglia motor-circuit for kinematic planning of arm movements. Comput Biol Med 92:78–89
8. Das R (2010) A comparison of multiple classification methods for diagnosis of Parkinson disease. Expert Syst Appl 37(2):1568–1572
9. Åström F, Koker R (2011) A parallel neural network approach to prediction of Parkinson’s disease. Expert Syst Appl 38(10):12470–12474
10. Khoury N, Attal F, Amirat Y, Chibani A, Mohammed S (2018) CDTW-based classification for Parkinson’s disease diagnosis. In: ESANN
11. Salimi-Badr A, Hashemi M (2020) A neural-based approach to aid early Parkinson’s disease diagnosis. In: 2020 11th international conference on information and knowledge technology (IKT), pp 23–25
12. Farashi S (2021) Analysis of vertical eye movements in Parkinson’s disease and its potential for diagnosis. Appl Intell 51(11):8260–8270. https://doi.org/10.1007/s10489-021-02264-9
13. El Maachi I, Bilodeau GA, Bouachir W (2020) Deep 1D-convnet for accurate Parkinson disease detection and severity prediction from gait. Expert Syst Appl 143:113075. https://doi.org/10.1016/j.eswa.2019.113075
14. Zhao A, Qi L, Li J, Dong J, Yu H (2018) A hybrid spatio-temporal model for detection and severity rating of Parkinson’s disease from gait data. Neurocomputing 315:1–8. https://doi.org/10.1016/j.neucom.2018.03.032
15. Liu X, Li W, Liu Z, Du F, Zou Q, (2021) A dual-brancl model for diagnosis of Parkinson’s disease based on the independent and joint features of the left and right gait. Appl Intell 51(10):7221–7232. https://doi.org/10.1007/s10489-020-02182-5
16. Gunning D, Stefić M, Choi J, Miller T, Stumpf S, Yang GZ (2019) XAI–explainable artificial intelligence. Sci Robot 4(37):eaay7120
17. Arrienda AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A et al (2020) Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58:82–115
18. Tjoa E, Guan C (2020) A survey on explainable artificial intelligence (xai): toward medical xai. IEEE Trans Neural Netw Learn Syst 32(11):4793–4813
19. Lee SH, Lim JS (2012) Parkinson’s disease classification using gait characteristics and wavelet-based feature extraction. Expert Syst Appl 39(8):7338–7344
20. Nancy Jane Y, Khanna Nehemia H, Arputharaj K (2016) A q-backpropagated time delay neural network for diagnosing severity of gait disturbances in Parkinson’s disease. J Biomed Inform 60:169–176. https://doi.org/10.1016/j.jbi.2016.01.014
21. Litvan I, Goldman JG, Tröster AI, Schmand BA, Weintraub D, Petersen RC et al (2012) Diagnostic criteria for mild cognitive impairment in Parkinson’s disease: movement disorder society task force guidelines. Mov Disord 27(3):349–356
22. Marras C, Armstrong MJ, Meaney CA, Fox S, Rothberg B, Reginold W et al (2013) Measuring mild cognitive impairment in patients with Parkinson’s disease. Mov Disord 28(5):626–633
23. Baiano C, Barone P, Trojano L, Santangelo G (2020) Prevalence and clinical aspects of mild cognitive impairment in Parkinson’s disease: a meta-analysis. Mov Disord 35(1):45–54
24. Yogev G, Giladi N, Perez C, Springer S, Simon ES, Hausdorff JM (2005) Dual tasking, gait rhythmicity, and Parkinson’s disease: which aspects of gait are attention demanding? Eur J NeuroSci 22(5):1248–1256
25. Yogev G, Plotnik M, Perez C, Giladi N, Hausdorff JM (2007) Gait asymmetry in patients with Parkinson’s disease and elderly fallers: when does the bilateral coordination of gait require attention? Exp Brain Res 177(3):336–346
26. Hausdorff JM, Cudkowicz ME, Firtion R, Wei JY, Goldberger AL (1998) Gait variability and basal ganglia disorders: stride-to-stride variations of gait cycle timing in Parkinson’s disease and Huntington’s disease. Mov Disord 13(3):428–437
27. Pahwa R, Lyons KE (2013) Handbook of Parkinson’s disease. Crc Press, Boca Raton

28. Harel B, Cannizzaro M, Snyder P. J. (2004) Variability in fundamental frequency during speech in prodromal and incipient Parkinson’s disease: a longitudinal case study. Brain Cogn 56(1):24–29

29. Hartelius L, Svensson P (1994) Speech and swallowing symptoms associated with Parkinson’s disease and multiple sclerosis: a survey. Folia Phoniatr Logop 46(1):9–17

30. Jeon HS, Han J, Yi WJ, Jeon B, Park KS (2008) Classification of Parkinson gait and normal gait using spatial-temporal image of plantar pressure. In: 2008 30th annual international conference of the IEEE engineering in medicine and biology society. IEEE, pp 4672–4675

31. Ashhar K, Soh CB, Kong KH (2017) A wearable ultrasonic sensor network for analysis of bilateral gait symmetry. In: 2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, pp 4455–4458

32. Nieuwboer A, Dom R, De Weerdt W, Desloovere K, Janssens L, Stijn V (2004) Electromyographic profiles of gait prior to onset of freezing episodes in patients with Parkinson’s disease. Brain 127(7):1650–1660

33. Hong M, Perlmutter JS, Earhart GM (2009) A kinematic and electromyographic analysis of turning in people with Parkinson disease. Neurorehabil Neural Repair 23(2):166–176

34. Saito N, Yamamoto T, Sugiura Y, Shimizu S, Shimizu M (2004) Lifecorder: a new device for the long-term monitoring of motor activities for Parkinson’s disease. Intern Med 43(8):685–692

35. Salarian A, Russmann H, Vingerhoets FJ, Dehollain C, Blanc Y, Burkhard P et al (2004) Gait assessment in Parkinson’s disease: toward an ambulatory system for long-term monitoring. IEEE Trans Biomed Eng 51(8):1434–1443

36. Mariani B, Jiménez MC, Vingerhoets FJ, Aminian K (2012) On-shoe wearable sensors for gait and turning assessment of patients with Parkinson’s disease. IEEE Trans Biomed Eng 60(1):155–158

37. Latash ML, Aruin AS, Neyman I, Nicholas JJ (1995) Anticipatory postural adjustments during self inflicted and predictable perturbations in Parkinson’s disease. J Neurol Neurosurg Psychiatry 58(3):326–334

38. Cho CW, Chao WH, Lin SH, Chen YY (2009) A vision-based analysis system for gait recognition in patients with Parkinson’s disease. Expert Syst Appl 36(3):7033–7039

39. Pachoulakis I, Kouroulis K (2014) Building a gait analysis framework for Parkinson’s disease patients: motion capture and skeleton 3D representation. In: 2014 international conference on telecommunications and multimedia (TEMU). IEEE, pp 220–225

40. Galna B, Barry G, Jackson D, Mhiripiri D, Olivier P, Rochester L (2014) Accuracy of the microsoft kinect sensor for measuring movement in people with Parkinson’s disease. Gait & Posture 39(4):1062–1068

41. Dror B, Yani E, Frid A, Peleg N, Goldenthal N, Schlesinger I et al (2014) Automatic assessment of Parkinson’s disease from natural hands movements using 3D depth sensor. In: 2014 IEEE 28th convention of electrical & electronics engineers in israel (IEEEI). IEEE, pp 1–5

42. Dyshel M, Arkadir D, Bergman H, Weinshall D (2015) Quantifying levodopa-induced dyskinesia using depth camera. In: Proceedings of the IEEE international conference on computer vision workshops, pp 119–126

43. Antonio-Rubio I, Madrid-Navarro C, Salazar-López E, Pérez-Navarro M, Sáez-Zea C, Gómez-Milán E et al (2015) Abnormal thermography in Parkinson’s disease. Parkinsonism Relat Disord 21(8):852–857

44. Song J, Sigward S, Fisher B, Salem GJ (2012) Altered dynamic postural control during step turning in persons with early-stage Parkinson’s disease. Parkinson’s Disease 2012

45. Foreman K, Wisted C, Addison O, Marcus R, LaStayo P, Dibble L (2012) Improved dynamic postural task performance without improvements in postural responses; the blessing and the curse of dopamine replacement. Parkinson’s Disease 2012

46. Muniz A, Liu H, Lyons K, Pahwa R, Liu W, Nobre F et al (2010) Comparison among probabilistic neural network, support vector machine and logistic regression for evaluating the effect of subthalamic stimulation in Parkinson disease on ground reaction force during gait. J Biomech 43(4):720–726

47. Vaugoyeau M, Viallet F, Mesure S, Massion J (2003) Coordination of axial rotation and step execution: deficits in Parkinson’s disease. Gait & Posture 18(3):150–157

48. Su B, Song R, Guo L, Yen CW (2015) Characterizing gait asymmetry via frequency sub-band components of the ground reaction force. Biomed Signal Process Control 18:56–60

49. Zeng W, Liu F, Wang Q, Wang Y, Ma L, Zhang Y (2016) Parkinson’s disease classification using gait analysis via deterministic learning. Neurosci Lett 633:268–278

50. Daliri MR (2012) Automatic diagnosis of neuro-degenerative diseases using gait dynamics. Measurement 45(7):1729–1734

51. Joshi D, Khajuria A, Joshi P (2017) An automatic non-invasive method for Parkinson’s disease classification. Comput Methods Prog Biomed 145:135–145

52. Frenkel-Toledo S, Giladi N, Peretz C, Herman T, Gruendlinger L, Hausdorff JM (2005) Treadmill walking as an external pacemaker to improve gait rhythm and stability in Parkinson’s disease. Movement Disorders: Official Journal of the Movement Disorder Society 20(9):1109–1114

53. Hausdorff JM, Lowenthal J, Herman T, Gruendlinger L, Peretz C, Giladi N (2007) Rhythmic auditory stimulation modulates gait variability in Parkinson’s disease. Eur J NeuroSci 26(8):2369–2375

54. Jang JSR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst, Man, Cybern 23(3):665–685. https://doi.org/10.1109/21.256541

55. de Jesus Rubio J (2009) SOFMLS: online self-organizing fuzzy modified least-squares network. IEEE Trans Fuzzy Syst 17(6):139–153. https://doi.org/10.1109/TFUZZ.2009.2029569

56. Malek H, Ebadzadeh MM, Rahmati M (2012) Three new fuzzy neural networks learning algorithms based on clustering, training error and genetic algorithm. Appl Intell 37(2):280–289

57. Ebadzadeh MM, Salimi-Badr A (2015) CFNN: correlated fuzzy neural network. Neurocomputing 148:430–444. https://doi.org/10.1016/j.neucom.2014.07.021

58. Ebadzadeh MM, Salimi-Badr A (2018) IC-FNN: a novel fuzzy neural network with interpretable, intuitive, and correlated-contours fuzzy rules for function approximation. IEEE Trans Fuzzy Syst 26(3):1288–1302

59. Salimi-Badr A, Ebadzadeh MM (2022) A novel Self-Organizing fuzzy neural network to learn and mimic habitual sequential tasks. IEEE Trans Cybern 52(1):323–332

60. Salimi-Badr A, Ebadzadeh M, Darlot C (2017) Fuzzy neural model of motor control inspired by cerebellar pathways to online and gradually learn inverse biomechanical functions in the presence of delay. Biol Cybern 111(5-6):421–438. https://doi.org/10.1007/s00422-017-0735-9

61. Salimi-Badr A, Ebadzadeh MM (2022) A novel learning algorithm based on computing the rules’ desired outputs of a TSK fuzzy neural network with non-separable fuzzy rules. Neurocomputing 470:139–153. https://doi.org/10.1016/j.neucom.2021.10.103

62. Salimi-Badr A (2022) IT2CFNN: an interval type-2 correlation-aware fuzzy neural network to construct non-separable fuzzy rules with uncertain and adaptive shapes for nonlinear function approximation. Appl Soft Comput 115:108258. https://doi.org/10.1016/j.asoc.2021.108258
63. Bencherif A, Chouireb F (2019) A recurrent TSK interval-type-2 fuzzy neural networks control with online structure and parameter learning for mobile robot trajectory tracking. Appl Intell 49(11):3881–3893. https://doi.org/10.1007/s10489-019-01439-y

64. Eyoh I, John R, De Maere G, Kayacan E (2018) Hybrid learning for interval-type-2 intuitionistic fuzzy logic systems as applied to identification and prediction problems. IEEE Trans Fuzzy Syst 26(5):2672–2685. https://doi.org/10.1109/TFUZZ.2018.2803751

65. Hausdorff JM, Rios DA, Edelberg HK (2001) Gait variability and fall risk in community-living older adults: a 1-year prospective study. Arch Phys Med Rehabil 82(8):1050–1056

66. Frenkel-Toledo S, Giladi N, Peretz C, Herman T, Gruendlinger L, Hausdorff JM (2005) Effect of gait speed on gait rhythmicity in Parkinson’s disease: variability of stride time and swing time respond differently. J Neuroeng Rehabil 2(1):1–7

67. Mamdani EH (1997) Application of fuzzy logic to approximate reasoning using linguistic synthesis. IEEE T Computers (12):1182–1191

68. Biglarbegian M, Melek WW, Mendel JM (2011) Design of novel interval-type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments. IEEE Trans Ind Electr 58(4):1371–1384. https://doi.org/10.1109/TIE.2010.2049718

69. Biglarbegian M, Melek WW, Mendel JM (2011) On the robustness of type-1 and interval-type-2 fuzzy logic systems in modeling. Inf Sci 181(7):1325–1347

70. Khanesar MA, Mendel JM (2016) Maclaurin series expansion complexity-reduced center of sets type-reduction + defuzzification for interval-type-2 fuzzy systems. In: 2016 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp 1224–1231

71. Mendel JM (2017) Uncertain rule-based fuzzy systems. Introduction and new directions, p 684

72. Karnik NN, Mendel JM (2001) Centroid of a type-2 fuzzy set. Inform Sci 132(1):195–220. https://doi.org/10.1016/S0019-9551(00)00605-X

73. Wu D, Mendel JM (2009) Enhanced Karnik–Mendel algorithms. IEEE Trans Fuzzy Syst 17(4):923–934

74. Juang CF, Tsao YW (2008) A self-evolving interval-type-2 fuzzy neural network with online structure and parameter learning. IEEE Trans Fuzzy Syst. 16(6):1411–1424

75. Juang CF, Huang RB, Cheng WY (2010) An interval-type-2 fuzzy-neural network with support-vector regression for noisy regression problems. IEEE Trans Fuzzy Syst 18(4):686–699

76. Pratama M, L, Hausdorff JM (2015) Effect of gait speed on gait rhythmicity in Parkinson’s disease: variability of stride time and swing time respond differently. J Neuroeng Rehabil 2(1):1–7

77. Juang CF, Wang PH (2015) An interval-type-2 neural fuzzy classifier learned through soft margin minimization and its human posture classification application. IEEE Trans Fuzzy Syst 23(5):1474–1487. https://doi.org/10.1109/TFUZZ.2014.2362547

78. Bakhouri N, Abraham A, Alimi AM (2018) A beta basis function interval-type-2 fuzzy neural network for time series applications. Eng Appl Artif Intell 71:259–274. https://doi.org/10.1016/j.engappai.2018.03.006

79. Das AK, Subramanian K, Sundaram S (2015) An evolving interval-type-2 neurofuzzy inference system and its metacognitive sequential learning algorithm. IEEE Trans Fuzzy Syst 23(6):2080–2093. https://doi.org/10.1109/TFUZZ.2015.2403793

80. Takagi T, Sugen M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man SMC-15(1):116–132

81. Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. J Soc Ind Appl Math 11(2):431–441. https://doi.org/10.1137/0111030

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Armin Salimi-Badr received the B.Sc., M.Sc. and PhD degrees in Computer Engineering, all from Amirkabir University of Technology, Tehran, Iran in 2010, 2012, and 2018 respectively. He also obtained a PhD degree in Neuroscience from University of Burgundy, Dijon, France in 2019, where he was researching on presenting a computational model of brain motor control in the Laboratory 1093 CAPS (Cognition, Action, et Plasticité Sensorimotrice) of the Institut National de la Santé et de la Recherche Médicale (INSERM). He was a Postdoctoral Research Fellow at Bio-computing lab of Amirkabir University of Technology from October 2019 to September 2020. Currently, he is an Assistant Professor at Faculty of Computer Science and Engineering of Shahid Beheshti University, Tehran, Iran. He is also the founder and Chair of Robotics Intelligent Autonomous Agents (RollA) Lab in Shahid Beheshti University. He is also a Senior Member of IEEE and currently he is the Chair of Professional Activities Committee and a Board Member of Computer Society of IEEE Iran Section. His research interests include fuzzy neural networks, evolutionary algorithms, computational intelligence, machine learning, and robotics.
Mohammad Hashemi is currently a senior undergraduate computer engineering student at Shahid Beheshti University (National University), Tehran, Iran. His main research interests are Computer Vision, Deep Learning, and Machine Learning.

Hamidreza Saffari is currently a senior undergraduate computer engineering student at Shahid Beheshti University (National University), Tehran, Iran. His main research interests are Computer Vision, Deep Learning, and Social network analysis.