Journal of Biomedical Informatics 60 (2016) 169–176

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

A Q-backpropagated time delay neural network for diagnosing severity of gait disturbances in Parkinson's disease

Y. Nancy Jane [a], H. Khanna Nehemiah [a,*], Kannan Arputharaj [b]

[a] Ramanujan Computing Centre, Anna University, Chennai 600025, India
[b] Department of Information Science and Technology, Anna University, Chennai 600025, India

ARTICLE INFO

Article history:
Received 4 August 2015
Revised 22 December 2015
Accepted 23 January 2016
Available online 2 February 2016

Keywords:
Clinical decision-making system
Parkinson's disease
Gait
Time delay neural network
Backpropagation
Q-learning

ABSTRACT

Parkinson's disease (PD) is a movement disorder that affects the patient's nervous system and health-care applications mostly uses wearable sensors to collect these data. Since these sensors generate time stamped data, analyzing gait disturbances in PD becomes challenging task. The objective of this paper is to develop an effective clinical decision-making system (CDMS) that aids the physician in diagnosing the severity of gait disturbances in PD affected patients. This paper presents a Q-backpropagated time delay neural network (Q-BTDNN) classifier that builds a temporal classification model, which performs the task of classification and prediction in CDMS. The proposed Q-learning induced backpropagation (Q-BP) training algorithm trains the Q-BTDNN by generating a reinforced error signal. The network's weights are adjusted through backpropagating the generated error signal. For experimentation, the proposed work uses a PD gait database, which contains gait measures collected through wearable sensors from three different PD research studies. The experimental result proves the efficiency of Q-BP in terms of its improved classification accuracy of 91.49%, 92.19% and 90.91% with three datasets accordingly compared to other neural network training algorithms.

1. Introduction

Parkinson's disease (PD) affects the nervous system of the patients by destroying neurons in the brain that produce a chemical named dopamine. The dopamine is responsible for sending messages to the brain for movement co-ordination [1–3]. Thus, most of the PD affected patient's exhibit movement disorders resulting in the postural instability or walking disturbances [1,2]. The symptoms of PD include muscle stiffness, tremors, and changes in speech and gait [2]. Many research studies provide detailed investigations about the PD symptoms [3]. In general, a gait cycle also referred as stride contains one stance phase and one swing phase [1,2]. The stance phase represents the period at which the foot strikes on the ground. The swing phase represents the period at which the same foot lifts up the floor. A normal person's walking constitutes repetition of this gait cycle and a normal human being approximately takes 60% of stance phase and 40% of swing phase [4]. PD patients often show disturbances and variations in this gait cycle. This work analyses the gait disturbances to identify its severity in PD.

In [5–8] the authors have presented an experimental study that examines the associations between the walking speed and variations in the gait. The authors have observed that for PD patients, there is a decrease in the stride length and average swing time and an increase in the stride and swing time variations. The impact of PD in terms of movement disabilities is measured using several rating scales namely Unified Parkinson Disease (UPDRS), Hoehn and Yahr Scale, modified UPDRS [9–11]. In [12] the authors have evaluated the severity level of PD by characterizing the leg swiftness task. The authors have investigated an association between the angular amplitude and speed of thigh motion with UPDRS scores.

Though, there are several studies [5–14] done to analyze the movement disorders in PD patients there are still many challenging areas of research in this domain due to the time stamped nature of PD data recorded through most of the wearable sensors.

1.1. Outline of the paper

This paper aims in developing a clinical decision-making system (CDMS) that uses an effective classification model for diagnosing the severity of gait disturbances in PD. This work presents a Q-backpropagated time delay neural network (Q-BTDNN) classifier for building a classification model. Q-BTDNN is a dynamic feed

* Corresponding author. Tel.: +91 044 22358013.
E-mail address: nehemiah@annauniv.edu (H. Khanna Nehemiah).

http://dx.doi.org/10.1016/j.jbi.2016.01.014
1532-0464/

forward time delay neural network (TDNN) in which the learning is done using the proposed Q-learning induced backpropagation (Q-BP) technique. The proposed work is experimented with time stamped gait database acquired through wearable sensors. The experimental result, prove the effectiveness of the presented classifier in terms of its improved classification accuracy and reduced error rates.

2. Related works

In this section, few existing research work related to this paper is reviewed. Cole et al. [15] have presented dynamic machine-learning algorithms for monitoring the tremor severity and dyskinesia by analyzing signals collected from PD patients wearing small numbers of hybrid sensors. The authors have designed dynamic algorithms in pattern classification, neural networks, support vector machines and Hidden Markov Model (HMM). The experimental data were collected from eight PD patients and four healthy individuals through wearable sensors by allowing them to do unplanned and unrestricted daily living activities. The experimental results show that the performance of all the presented dynamic algorithms is equally effective in reducing the error rates.

Wangr and Jiang [16] have presented an incremental learning classification model based on fuzzy clustering algorithm and probabilistic neural networks for sensor-based human activity recognition. The presented classifier has the ability to adapt and incrementally learn from the training data. The system effectively performs classification with incremental data and proves its efficiency in terms of its learning ability and accuracy. Mazilu et al. [17] have investigated the performance of feature learning process for detecting freezing of gait (FOG) in PD patients. Three features namely statistical features, time-domain and unsupervised based on principal components analysis were considered. The experiments were conducted with acceleration data acquired from ankle of patients suffering with FOG. The experimental results prove that the statistical and time-domain features outperform the unsupervised feature extraction process.

Michael et al. [18] have presented an evolutionary algorithm, namely sliding window genetic programming (SGP) and Artificial biochemical networks (ABN) to classify the movement characteristics of Parkinson disease patients. SGP is used to capture movement patterns within a cycle. ABN is used to capture dynamical patterns occurring during time scales. Two-thirds of the clinical recordings are placed in a training set for fitness evaluation. The other third of the data is used to evaluate the classifier. Though, both the techniques effectively classify PD patients and control patients the SGP classifier outperforms ABN. The advantage of using evolutionary algorithms is to produce patterns that human might not notice.

Ene [19] have presented an application of probabilistic neural network (PNN) for classifying healthy and PD subjects. The PNN model based on three searches namely incremental search (IS), Monte Carlo search (MCS) and hybrid search (HS) were used in the classification process. The experimental study was conducted with biomedical voice measures data for twenty-five PD patients and six normal person obtained through UCI repository. From the experimental results, it can be inferred that there is no significant differences between three searches, however; the use of hybrid heuristic approach can improve the classification results.

Little et al. [20] have presented a classification technique named Kernel based Support Vector machine to diagnose the PD by identifying dysphonia. For experimentation, the authors used sustained phonations from 23 PD patients and 8-control person. It was observed that the new dysphonia measure introduced such as pitch period frequency along with another ten measures provides improved classification accuracy, which is recommended in many telemonitoring applications. Rigas et al. [21] have presented a study to illustrate that a hidden Markov model (HMM) is well suited for identifying tremors since they mostly represents temporal dependencies. They have experimented with ten patients and thirteen control subjects daily activity accelerometer data. Djuric-Jovicic et al. [22] have presented a thresholding technique and a neural network to classify PD patients based on their walking patterns. This distinguishes the normal walk and shuffling steps. For experimentation, the data were acquired using a set of six inertial measurement units attached to the subjects' legs (i.e. thigh and shin) as well as their feet. The movements of four patients for thirty minutes were collected and used to train a neural network. The error rate of the training process obtained depends on the choice of threshold.

Das [23] has presented a comparative study about various classification methods, namely Decision Tree, Neural Networks, DMneural and Regression for diagnosing PD disease. For experimentation, the authors have used biomedical voice measurements from PD patients who are suffering from speech disorder. From the experimental results, it was observed that neural network outperforms other classifiers in terms of its classification accuracy. Ahlrichs and Lawo [3] have presented a detailed review that discusses about various techniques used in diagnosing PD based on motor symptoms from times series data. The authors have provided detail descriptions about the accuracies and error rates with respect to the experimental data they have considered. Waibel [24] has proposed a time delay neural network for identifying the temporal relationships among the acoustic–phonetic features.

Comparing to the works discussed in the literature the proposed work is different in following ways: This work proposes a reinforced Q-learning backpropagation algorithm to train the TDNN in an incremental way. During the training process, the network weights are adjusted based on the reinforced backpropagated error signal. The temporal ordering among the observed gait patterns of each subjects are considered in diagnosing the severity conditions of the gait disturbances in PD.

3. Materials and methods

This section describes the dataset and methods used in the presented temporal data mining framework.

3.1. Dataset description

For experimentation, this work uses the PD gait database [25] that contains data collected in the Unit of the Tel-Aviv Sourasky Medical Center at the Laboratory for Gait & Neurodynamics, Movement Disorders. This database consists of three PD datasets used in research studies [5–8]. Totally, this database stores 93 PD subjects and 73 control subjects. Each person involved in the study is referred as a subject. In the data acquisition, a computerized force-sensitive wearable sensor from Ultraflex Computer Dyno Graphy, Infotronic Inc. [25] measures the stride-to-stride variations and gait of a subject. The wearable sensor consists of a pair of shoes each of which contains eight sensors that is placed in the insole. The subjects were asked to wear those shoes and walk using different styles such as treadmill walking, unassisted walking on a ground level, walking on a ground level using walker, dual-task walking. The vertical ground reaction force (VGRF) from each sensor measured in newtons is recorded in the attached memory card.

These walking (gait) patterns of the PD subjects and normal subjects were observed for 2 min. The sensor generates output for every 0.01 s and for each subject 12,000 observations were

Table 1
Dataset overview.

Dataset study	Subjects	Total subjects	Female	Male
Ga [6]	PD	29	9	20
	CO	18	8	10
Si [7]	PD	35	13	22
	CO	29	11	18
Ju [8]	PD	29	13	16
	CO	26	14	12

Table 2
Dataset description.

Column	Description	Units
1	Time	Seconds
2–9	Measured Vertical ground reaction force (VGRF) from each of eight sensors (L1–L8) in left leg	Newton
10–17	Measured Vertical ground reaction force (VGRF) from each of eight sensors (R1–R8) in Right leg	Newton
18	Total force under the left leg	Newton
19	Total force under the Right leg	Newton

considered. The detailed mode of the study and statistical analysis were discussed in [5–8] which provide detail descriptions of the PD data considered in this work. Table 1 describes an overview of the data that were used by the authors in their study [6–8]. Table 2 provides the descriptions of the attributes in the datasets.

The time stamped data in Table 2 and few non time-stamped data, such as the medical identity, gender, age, height (m), weight (kg), Hoehn Yahr were considered in this work for experimentation.

3.2. Methods

The PD dataset used in this work is a time series data that describes a temporal sequence of observations on each subject walking pattern. To build a classification model for this time series data, this work presents a Q-BTDNN classifier that is trained using the proposed Q-BP algorithm. The trained classification model is used in CDMS for predicting the gait disturbances in PD. The following sections provide a detail description about the presented Q-BTDNN structure, the proposed Q-BP algorithm and its application in predicting the severity of gait disturbances in PD.

3.2.1. Q-BTDNN structure and Q-BP learning algorithm

The traditional TDNN [24] is a biologically inspired neural network that is used to model time-series data. TDNN interprets the temporal sequence effectively by relating the inputs in different time points. Q-BTDNN presented in this work is a feed forward time delay neural network (TDNN) in which the training is done using the proposed Q-learning induced backpropagation (Q-BP) technique. Q-BP functions in an incremental way by combining the relative advantages of both the reinforced Q-learning [26] and backpropagation [27]. The network structure for Q-BTDNN shown in Fig. 1 includes three layers, namely tapped delay input layer, computational layer and reinforced feedback layer.

The tapped delay input layer feeds the inputs from a clinical time-series data. The input ($I_1, \ldots, I_y$) refers to a clinical examination (or an attribute) that describes the medical test taken from a person (subject) where I_1 refers to first input attribute and 'y' refers to the total number of attributes. Each input is observed successively at different time points ($t, t-d, t-2d, \ldots$), where 'd' refers to the delay between each time point.

The computational layer comprises of hidden and output layers. The configuration of this layer relies on the chosen application. The presented work adopts the computational layer of two hidden layer with 240 hidden nodes, 120 hidden nodes respectively and an output layer with four nodes. The error tracker keeps track of the weights for which minimum error is achieved. The reinforced feedback layer generates a reinforced error signal using the Q-learning technique, which is backpropagated on the network to adjust the network's weights.

The traditional backpropagation (BP) learning technique [27,28] minimizes the classification error by adjusting the network's weights through backpropagating the error obtained for each training input instance. This work presents a Q-BP learning algorithm to train the Q-BTDNN. The difference between the traditional BP and Q-BP lies in the mode of propagating the error and adjusting the weights. In Q-BP instead of directly backpropagating the error, we generate a reinforced error signal using Q-learning principle and then backpropagate. The weight updations are done based on this reinforced error signal. The proposed algorithm

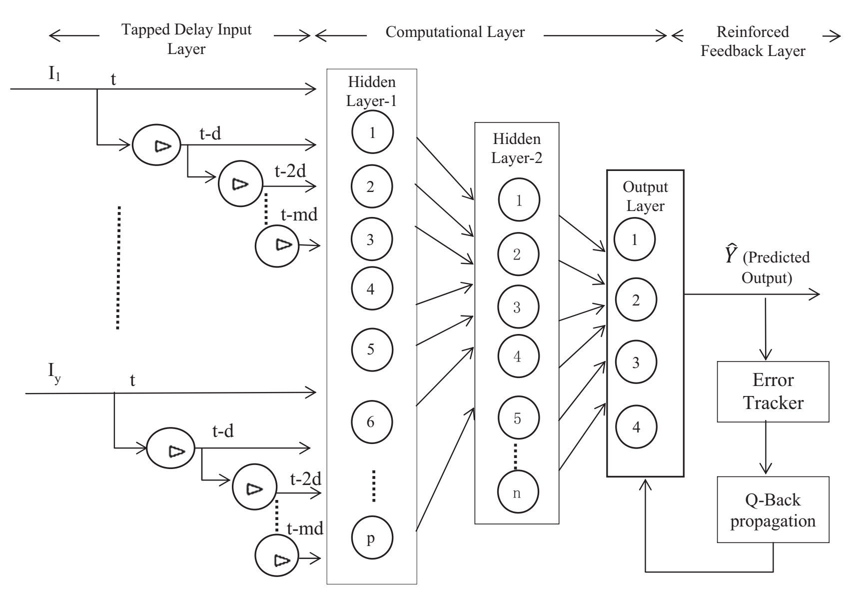

Fig. 1. Q-BTDNN structure for predicting gait disturbances in PD.

includes two major functionalities namely forward input propagation with delay, reinforced error backpropagation and weight updation.

In forward input propagation with delay the inputs are fed forward to the presented Q-BTDNN with the delay of '*d*'. The working principles of Q-BP algorithm are illustrated in the steps 1–9. The forward input propagation with delay includes the process involved in forwarding the temporal inputs in the forward direction which is summarized in the steps 3–4. The process involved in computing the reinforced error backpropagation and weight updation is summarized in the steps 5–8.

The following are the steps used in Q-BP algorithm for training the time delay neural network:

Step 1: Initialize the weights of the network randomly.
Step 2: Read the temporal sequence of each subject in TDNN from 16 sensors with a delay of 0.01 s.
// Forward input propagation with delay
Step 3: For every node in the input layer its output equals the inputs along with the time delay lines (TDL) given to it.
Step 4: For every node (j) in the hidden layer, compute its output using the Eq. (1).

$$O_j(t) = \varphi\left(\sum_{i=1}^{m}\sum_{g=0}^{l} RQw_{ij}x_i(t-g)\right) + \theta_j \quad (1)$$

where RQw_{ij} is the networks weight adjusted using Q-learning, $x_i(t-g)$ inputs at the time $(t-g)$ for ith node in the input layer, m refers to the number of nodes in the input layer, l refers to the delay lag for each input sensor observation. θ_j is the bias and φ denotes the activation. For every node (k) in the output layer, compute its output using the Eq. (2).

$$O_k(t) = \varphi\left(\sum_{j=1}^{h} RQw_{jk}O_j(t)\right) + \theta_k \quad (2)$$

where RQw_{jk} is the networks weight adjusted using Q-learning, $O_j(t)$ output from the jth hidden node, h refers to the number of hidden nodes, θ_k is the bias and φ denotes the activation.
// Reinforced Error backpropagation and Weight Updation (Q-Learning induced Error propagation).
Step 5: For each node in the output layer compute its error based on the three functions, namely Q-function, Reinforcement function and weight updation using the step 6, 7 and 8 through Q-learning method.
Step 6: Q-function is computed when every subject's observation is given to the network. The computed Q-value represents the utility function for the current patient state and the result of the action performed.

$$QW_{t+1}(S_t,a_t) = QW_t(S_t,a_t) + \alpha_t(S_t,a_t).(Ri_{t+1} + \gamma\max_a QW_t(S_t,a_t) - QW_t(S_t,a_t) \quad (3)$$

where $QW_t(S_t,a_t)$ is the old Q-value, S_t and a_t refers to state and action, $\alpha_t(S_t,a_t)$ is the learning rate, Ri_{t+1} is the reward value, γ is the discount factor, $\max_a QW_t(S_t,a_t)$ is the estimate of optimal value. In this work we refer three actions of adjusting weights namely Q-dependent, target value dependent, gradient descent, gradient descent with momentum. The estimate of optimal weights is identified by finding best minimal errors returned by taking any of these actions.
Step 7: The reinforcement signal (Ri), which represents the reward or penalty, is generated from the outcome of the error and is defined in the Eq. (4). A parameter named min_error-tracker (MinE) is maintained to keep track of the weights for which minimum error is achieved. Initially the min_error-tracker is assigned with first error generated with initialized random weights. As the network training process starts, it reassigns its value by tracking the latest local best optimal weight.

$$Ri = \begin{cases} 1, & \text{CurrE} < \text{MinE} \\ -1, & \text{CurrE} > \text{MinE} \end{cases} \quad (4)$$

The reinforcement signal assigns a reward (1) when it finds that the current predicted Error (CurrE) is less than MinE otherwise it sends a penalty signal (−1).
Step 8: Weight updations is done based on the generated reinforcement signal. If reward signal is activated then the network weights are adjusted by adding Q-value to the all current weights as defined in Eq. (5).

$$RQw_{ij} = RQw_{ij} + QW_{t+1}(S_t,a_t) \quad (5)$$

If the penalty is activated then the combination of Q-value and propagated error is averaged and used in the weight updation as defined in the Eq. (6).

$$RQw_{ij} = RQw_{ij} + (QW_{t+1}(S_t,a_t) + \text{CurrE}) \quad (6)$$

Step 9: The network training stops when the terminating conditions bounded with the iterations have reached.

3.2.2. Application of Q-BTDNN to diagnose gait disturbances in PD

The Q-BTDNN classifier is applied to diagnose the severity of gait disturbances in PD. The gait data is acquired from three PD studies [6–8]. The description of the data and its method of data acquisition are discussed in the earlier Section 3.1. The proposed work uses a network structure that includes one tapped delay input layer and computational layer comprising of two hidden layers and one output layer. The tapped delay input layer feeds the successive observation of each 16 input sensors (L1, L2, L3, L4, L5, L6, L7, L8, R1, R2, R3, R4, R5, R6, R7, R8) recorded from the right leg and left leg of the subjects at different time points (t, $t-d$, $t-2d$, ...), where 'd' refers to the delay between each time point. During the network training process, this works considers 12,000 time stamped observations from 16 sensors for each subject.

The computational layer comprises of two hidden layer with 240 hidden nodes and 120 hidden nodes respectively, and an output layer with four nodes. The reinforced feedback layer generates a reinforced error signal using the Q-learning technique, which is backpropagated on the network to adjust the network's weights. The input layer of Q-BTDNN forms a 16 × 12,000 vector. The first hidden layer consists of 240 nodes arranged in 2 × 120 vector. The second hidden layer consists of 120 nodes arranged in 2 × 60 vector. The output layer consists of four nodes arranged in 1 × 4 vector. Each 100 frames in the input layer are mapped to one frame in the first hidden layer. Every 2 frames in the first hidden layer are mapped to a node in the second hidden layer. Each 15 frame in the second hidden layer is connected to one frame in the output layer. For each subject instance, the network computes an output.

In forward input propagation with delay, the data from 16 sensors observed from the left and right legs of the subjects are fed forward to the time delay network with the delay of 0.01 s. These inputs are forwarded to the Q-BTDNN using steps 1–4. A reinforcement based Q-learning approach is used to generate and backpropagate the error signal using steps 5–8.

4. Experimental settings

This section discusses the experimental settings and evaluation criteria used to assess the effectiveness of the Q-BTDNN classifier with three PD study dataset. Experiments were conducted with 93 PD subjects and 73 normal subjects, each observed data record corresponds to the walking patterns observed in 2 min walk using

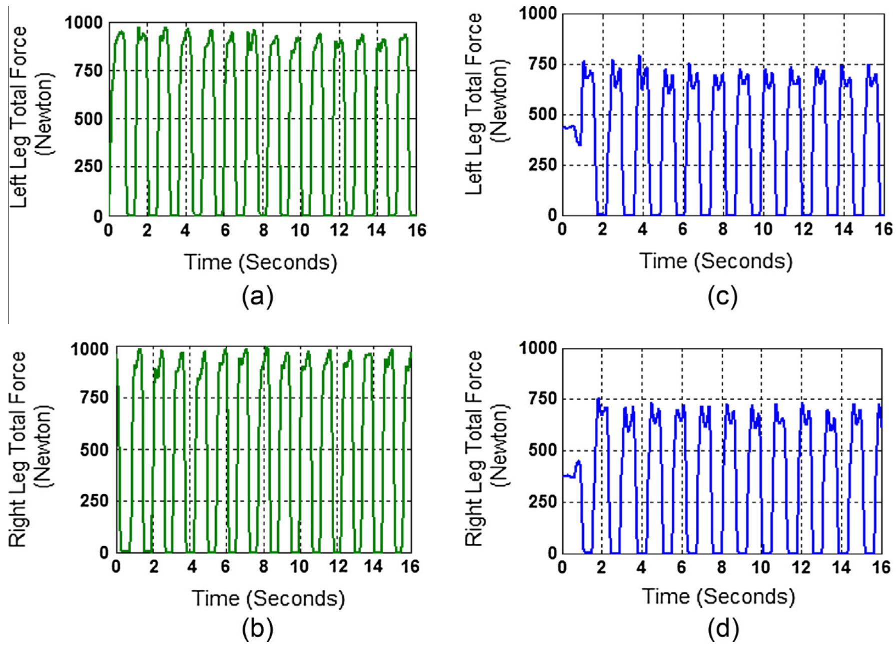

Fig. 2. (a and b) TF in left and right leg of PD patient and (c and d) TF in left and right leg of normal person.

wearable sensors. Fig. 2 shows the graphical plot that illustrates the total force recorded for 15 s from the 8 sensors in the left leg and 8 sensors from the right leg for a sample PD and normal subject. A stride measure refers to the period the foot strikes the ground goes to swing and the foot in the same leg strikes the ground. The Fig. 2 illustrates that when a person is in stance state the force recorded will be higher compared to the moment he or she is going to swing state. PD affected patients show a high increase in the average number of strides within an observed duration compared to the normal control patients. The variations in stance, swing and stride time of the PD patients show a high impact in identifying the severity of the disease.

The input data from 16 sensors are propagated in the Q-BTDNN with a delay of 0.01 s. The records were sampled at the rate of 100 samples per second that is observed for two minutes. The implementations and experimentations were carried out using the neural network tool box in MATLAB 2013 [29]. A 10-fold cross validation is used for experimental evaluation. A 10-Cross-validation model assessment technique randomly split the dataset into 10 subsets of approximately equal size. The 9 subsets are used for training and remaining one subset is used for testing. The entire process is repeated for 10 times. To make the evaluation unbiased it is required to make several runs of 10-fold cross validation. Thus, in this work the classification model was assessed using 10-fold cross validation repeated for 5 independent runs. To select the best fit parameter values for Q-BTDNN training, a series of experiments were conducted with different combinations of network parameters. The parameter for which the network in its training process gives a minimal RMSE were selected. Based on the outcome of these experiments the computational layer of Q-BTDNN is configured with two hidden layer and an output layer for constructing a trained classification model for PD gait database. The first hidden layer has 240 hidden nodes arranged in 2×120, the second hidden layer has 120 nodes arranged in 2×60 and an output layer with four hidden nodes. The four output nodes represent the class label minimal, mild, moderate and severe. These class labels in the training records describe the severity of gait disturbances in PD based on the Hoehn and Yahr Scale [11]**.** The Hoehn and Yahr Scale represent the PD stages in the scale of 1.5–2.5 based on the motor symptoms.

Accordingly, this works considers the class label minimal for scale rating 1–1.5, mild for 1.5–2, moderate for 2–2.5 and severe for more than 2.5. A learning rate of 0.01 and sigmoid activation function was choosen for Q-BTDNN training. The network output scales the value to be in the range of 0–1. The output in four nodes is interpreted as 1000 for minimal, 0100 for mild, 0010 for moderate and 0001 for severe. The realization of the presented classifier tested with a few different set of hidden units is discussed in this paper. The work was tested with one hidden layer with 240, 120, 80 hidden nodes. For the one hidden layer of 240 nodes a vector space of 2×120 was considered. Each 100 frames in the input layer were mapped to one frame in the hidden layer. Each 30 frame in hidden layer is mapped to one node in the output layer. For the one hidden layer of 120 nodes a vector space of 2×60 was considered. Each 200 frames in the input layer were mapped to one frame in the hidden layer. Each 15 frame in hidden layer is mapped to one node in the output layer. For the one hidden layer of 80 nodes a vector space of 2×40 was considered. Each 300 frames in the input layer were mapped to one frame in the hidden layer. Each 10 frame in hidden layer is mapped to one node in the output layer.

For two hidden layers of 240 and 120 nodes, the first hidden layer forms 2×120 vector and Second hidden layer forms 2×60 vector. Each 100 frames in input layer were mapped to one frame in the first hidden layer, each 2 frames in the first hidden layer is mapped to one frame in second hidden layer, each 15 frame in the second hidden layer is mapped to one node in output layer. For two hidden layers of 120 and 60 nodes, the first hidden layer forms 2×60 vector and Second hidden layer forms 2×30 vector. Each 200 frames in input layer is mapped to one frame in the first hidden layer, each 2 frames in first hidden layer is mapped to one frame in second hidden layer, each 13 frame in second hidden layer is mapped to one node in output layer. For two hidden layers of 80 and 40 nodes, the first hidden layer forms 2×40 vector and Second hidden layer forms 2×20 vector. Each 300 frames in input layer is mapped to one frame in first hidden layer, each 4 frames in first hidden layer is mapped to one frame in second hidden layer, each 5 frame in second hidden layer is mapped to one node in output layer. The experiments were conducted on a personal computer with Intel (R) Core (TM) i7-3770 processor with speed of

3.40 GHZ and 8 GB RAM. The programs were implemented using MATLAB 2013 computing platform.

5. Results and discussion

This section briefly discusses the results and findings obtained from the proposed Q-BTDNN classifier with the three PD study dataset. To select the best fit values for the network parameters in training process, a huge number of network architectures and experimentations were implemented. Table 3 shows a few results of this realization during the network construction and parameter selection process. Since, there were many experimental trails conducted to find the best fit values, results of few implementations were shown in the table. The root mean square error (RMSE) obtained at 100, 150 and 200 epochs was shown for the different combinations of the number of nodes in the computational layer of the network structure. The network stabilizes itself and RMSE value reaches minimum at 200 epochs for the hepatitis and thrombosis dataset and hence the training was stopped.

Table 3
RMSE values vs epochs observed for varied combinations in network structure.

Data set	Hidden layers	Hidden layer 1 nodes	Hidden layer 2 nodes	RMSE ($\times 10^{-1}$)		
				100 Epochs	150 Epochs	200 Epochs
Ga [6]	1	240	–	0.765	0.713	0.691
		120	–	0.791	0.726	0.611
		80	–	0.801	0.782	0.776
	2	240	120	0.414	0.358	0.279
		120	60	0.647	0.616	0.597
		80	40	0.698	0.679	0.623
Si [7]	1	240	–	0.703	0.696	0.615
		120	–	0.765	0.723	0.698
		80	–	0.799	0.781	0.786
	2	240	120	0.382	0.328	0.239
		120	60	0.567	0.534	0.512
		80	40	0.618	0.597	0.555
Ju [8]	1	240	–	0.798	0.711	0.701
		120	–	0.801	0.785	0.713
		80	–	0.823	0.811	0.791
	2	240	120	0.433	0.378	0.299
		120	60	0.667	0.613	0.612
		80	40	0.638	0.653	0.703

A 10-fold-cross-validation repeated for 5 runs generates test and training set for evaluating the classification model. The results obtained from each fold is averaged and considered for comparative analysis. Fig. 3 depicts the graphical plot for illustrating the RMSE value obtained for various iterations during Q-BTDNN training for the three datasets [6–8]. It can be observed that at 200 epochs the networks RMSE value stabilizes.

The classification results were evaluated with the following measures true positive rate (TPR), true negative rate (TNR), recognition rate (RR), misclassification rate (MR) and precision [28]. TPR refers to the proportion of subjects correctly classified as having PD. TNR refers to the proportion of subjects correctly classified, as not having PD. RR is the percentage of subjects correctly classified as PD or normal. MR is the percentage of subjects not correctly classified as PD or normal. Precision is the measure of percentage of records labeled as PD tuples are correctly as such. It is observed from figure (Fig. 3) that the RMSE value reaches minimum at 200 iterations and in this work the training terminates at this point. The performance of the proposed Q-BP algorithm for TDNN is compared with other training algorithms, namely Levenberg–Mar-

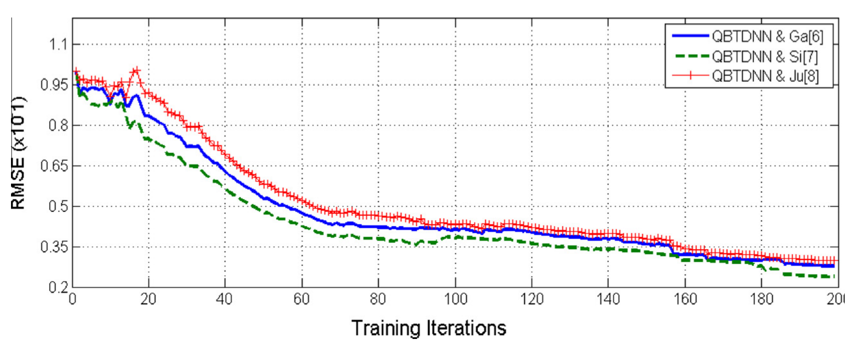

Fig. 3. Q-BTDNN training iterations vs root mean square error (RMSE).

Table 4
Comparison of classification results.

Dataset	Network learning techniques	TPR	TNR	RR	MR	Precision
Ga [6]	Q-BP *(Proposed)*	0.77	0.55	91.49	8.51	93.10
	Levenberg–Marquardt [30]	0.69	0.48	80.85	19.15	82.76
	Gradient descent [32]	0.60	0.41	70.12	29.79	72.41
	Gradient descent with momentum [32]	0.66	0.45	76.60	23.40	79.31
	Scaled Conjugate Gradient [31]	0.63	0.41	72.34	27.66	75.86
Si [7]	Q-BP *(Proposed)*	0.943	0.897	92.19	7.81	91.67
	Levenberg–Marquardt [30]	0.886	0.793	84.38	15.63	83.78
	Gradient descent [32]	0.829	0.724	78.13	21.88	78.38
	Gradient descent with momentum [32]	0.829	0.828	82.81	17.19	82.86
	Scaled Conjugate Gradient [31]	0.800	0.724	76.56	23.44	80.00
Ju [8]	Q-BP *(Proposed)*	0.74	0.83	90.91	9.09	89.66
	Levenberg–Marquardt [30]	0.71	0.72	83.64	16.36	83.33
	Gradient descent [32]	0.60	0.66	72.73	27.27	65.63
	Gradient descent with momentum [32]	0.66	0.76	81.82	18.18	79.31
	Scaled Conjugate Gradient [31]	0.60	0.69	74.55	25.45	70.00

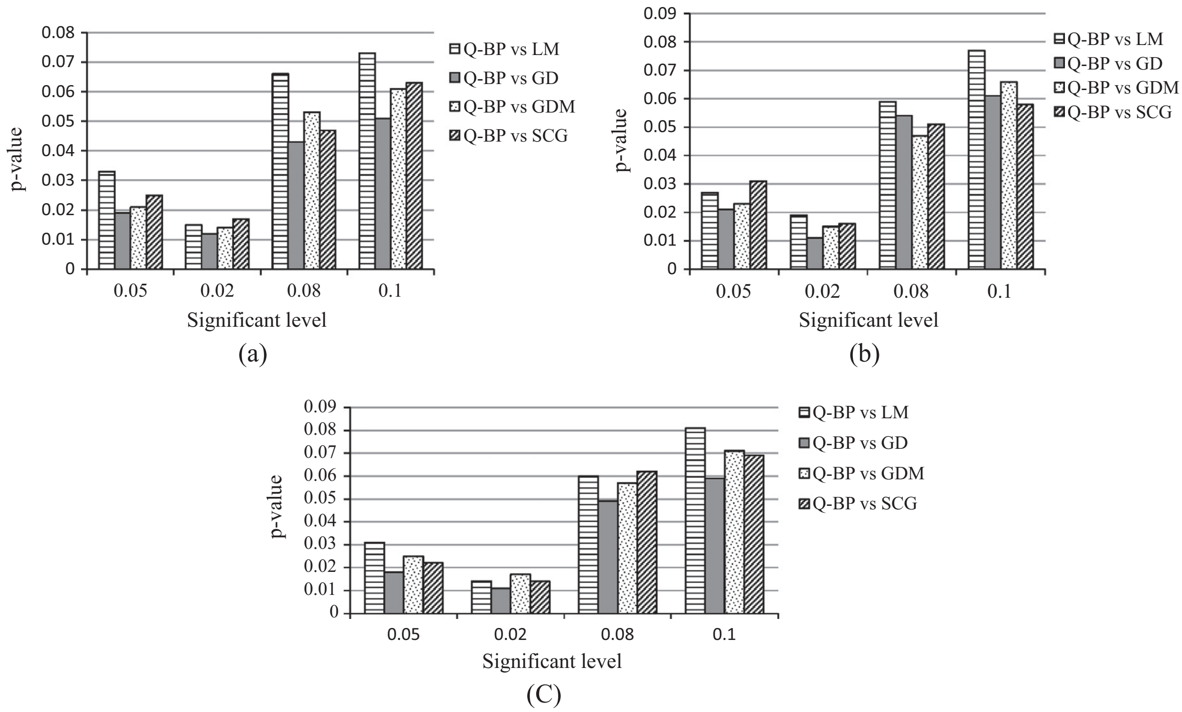

Fig. 4. Results of statistical paired *t*-test on classification accuracies (a) Dataset: Ga [6], (b) Dataset: Si [7] and (c) Dataset: Ju [8].

quardt (LM), Gradient descent (GD), Gradient descent with momentum (GDM) and Scaled Conjugate Gradient (SCG) learning algorithms [30–32]. Table 4 illustrates the classification results obtained from the proposed Q-BP and the other related learning algorithms.

The obtained classification accuracy of Q-BP with TDNN for the data gathered in the study [6] is 91.49% and for LM, GD, GDM and SCG are 80.85%, 70.12%, 76.6%, 72.34% respectively. For the data gathered in the study [7] the obtained classification accuracy is 92.19% and for LM, GD, GDM and SCG are 84.38%, 78.13%, 84.38%, 76.56% respectively. For the data gathered in the study [8] the obtained classification accuracy is 90.91% and for LM, GD, GDM and SCG are 83.64%, 72.73%, 81.82%, 74.55% respectively. From the classification results, it has been observed that TPR, FPR, RR, MR and precision of Q-BP with TDNN outperforms other TDNN training algorithms like LM, GD, GDM and SCG. In addition to the traditional classification metrics a statistical hypothesis testing was also carried out using paired *t*-test [33,34]. The test was carried out with significant level of 0.05 to identify whether there was any significant change in the error rates of Q-BP compared with Levenberg–Marquardt (LM), Gradient descent (GD), Gradient descent with momentum (GDM) and Scaled Conjugate Gradient (SCG) learning algorithms. The ρ value of less than 0.05 is obtained, which proves that there is a significant change in the error-rates of Q-BP compared to Levenberg–Marquardt (LM), Gradient descent (GD), Gradient descent with momentum (GDM) and Scaled Conjugate Gradient (SCG) learning algorithms. Thus, the evaluation results show that there is a significant improvement in the classification accuracy. The graphical illustration of the results of statistical assessment obtained for various significant level is shown in Fig. 4.

The computational time in seconds for training PD data [6] using Q-BP is 222, LM is 212, GD is 188, GDM is 179 and SCG learning is 175. The computational time in seconds for training PD data [7] using Q-BP is 231, LM is 219, GD is 198 and GDM is 195 and SCG learning is 193. The computational time in seconds for training PD data [8] using Q-BP is 227, LM is 216, GD is 192, GDM is 184 and Scaled Conjugate Gradient (SCG) learning is 197.

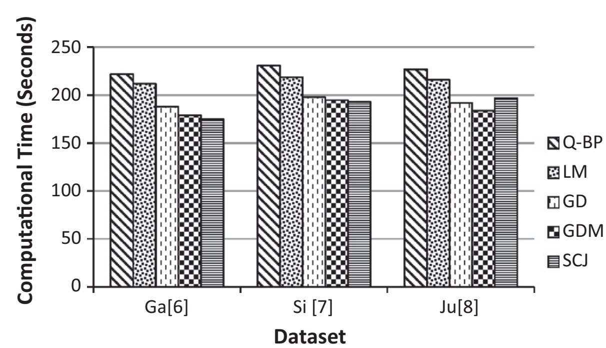

Fig. 5. Comparison of computational times.

Fig. 5 shows a comparison with computational times of Q-BP, LM, GD, GDM and SCG. Though, the computational time for the proposed Q-BP is found to be a bit higher compared to LM, GD, GDM and SCG; the high accuracy of Q-BP makes it effective in diagnosing the gait disturbances in PD.

This work has performed vast implementation of architectures to select the best fit parameter values for neural network learning. Hence, the computational cost of training the network is a bit higher compared to other classical approaches. However, this cost is associated only with the training process. The trained classification model is effective in diagnosing the gait disturbances in PD patients, which makes it recommended and applicable for practical usage. Thus, the trained classification model can be used in developing a clinical decision making system for assisting the clinician in diagnosing the severity of gait disturbances in PD affected patients.

6. Conclusion

Gait data acquired through wearable sensors for detecting gait disturbances in PD are time stamped and they characterize temporal related walking patterns. This work aims at presenting a Q-BTDNN classifier that monitors and predicts the severity of gait disturbances in PD affected patients by analyzing the instabilities in the postures and walking patterns. A Q-backpropagated (Q-BP) training algorithm is proposed to train the Time Delay Neural Network for building a classification model that is used to develop a clinical decision making system (CDMS). This CDMS is used to assist the physician in diagnosing the severity of gait disturbances in Parkinson's disease affected patients. The presented Q-BTDNN classifier is experimented with the Parkinson gait database collected from physionet [25], which includes data from three PD research studies. The experimental results shows, that the classification accuracy of Q-BTDNN for the three study datasets is 91.49%, 92.19% and 90.91% accordingly.

The proposed work demonstrates its effectiveness on three PD study data [6–8]. To prove the effectiveness and scalability, the system was also evaluated using large-scale clinical data which is not related to gait analysis. The experimental result showed improved classification accuracy and has proven the extendibility of the system with large-scale data. However, as a future work, the authors are investigating studies on extending the Q-BP algorithm to large-scale PD data analysis. There are still many challenging areas of research in handling the time stamped data for improving the classification accuracy and reducing the computational cost.

Conflict of interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] C.A. Davie, A review of Parkinson's disease, British Med. Bull. 86 (1) (2008) 109–127.
[2] Parkinson's disease Foundation, <http://www.pdf.org/en/about_pd> (2011.09.16).
[3] Claas Ahlrichs, Michael Lawo, Parkinson's disease motor symptoms in machine learning: a review, Health Inform. Int. J. 2 (4) (2013).
[4] Root2Being, <http://root2being.com/foot-shape-and-function/foot-function/>.
[5] S. Frenkel-Toledo, N. Giladi, C. Peretz, T. Herman, L. Gruendlinger, J.M. Hausdorff, Effect of gait speed on gait rhythmicity in Parkinson's disease: variability of stride time and swing time respond differently, J. Neuro Eng. Rehab. 2 (2005) 23.
[6] G. Yogev, N. Giladi, C. Peretz, S. Springer, E.S. Simon, J.M. Hausdorff, Dual tasking, gait rhythmicity, and Parkinson's disease: which aspects of gait are attention demanding, Eur. J. Neurosci. 22 (2005) 1248–1256.
[7] S. Frenkel-Toledo, N. Giladi, C. Peretz, T. Herman, L. Gruendlinger, J.M. Hausdorff, Treadmill walking as a pacemaker to improve gait rhythm and stability in Parkinson's disease, Movement Disorders 20 (9) (2005) 1109–1114.
[8] J.M. Hausdorff, J. Lowenthal, T. Herman, L. Gruendlinger, C. Peretz, N. Giladi, Rhythmic auditory stimulation modulates gait variability in Parkinson's disease, Eur. J. Neurosci. 26 (2007) 2369–2375.
[9] S. Fahn, R.L. Elton, Unified Parkinson's disease rating scale, in: S. Fahn, C.D. Marsden, M. Goldstein, D.B. Calne (Eds.), Recent Developments in Parkinson's Disease, vol. 2, Macmillan Healthcare Information, 1987.
[10] Movement disorder society task force on rating scales for Parkinson's. The unified Parkinson's disease rating scale (updrs): status and recommendations, Movement Disorders: Off. J. Movement Disorder Soc. 18(7) (2003) 738–750.
[11] M. Hoehn, M. Yahr, Parkinsonism: onset, progression and mortality, Neurology 17 (5) (1967) 427–442.
[12] Matteo Giuberti, Gianluigi Ferrari, Laura Contin, Veronica Cimolin, Corrado Azzaro, Giovanni Albani, Alessandro Mauro, Assigning UPDRS scores in the leg agility task of Parkinsonians: can it be done through BSN-based kinematic variables, IEEE Internet Things J. 2 (1) (2015) 41–51.
[13] M.F. Beal, Experimental models of Parkinson's disease, nature reviews, Neuroscience 2 (2001) 325–334.
[14] Cho, W.H. Chao, S.H. Lin, Y. Chen, A vision-based analysis system for gait recognition in patients with Parkinson's disease, Expert Syst. Appl. 36 (3) (2009) 7033–7039.
[15] B.T. Cole, S.H. Roy, C.J. De Luca, S.H. Nawab, Dynamical learning and tracking of tremor and dyskinesia from wearable sensors, IEEE Trans. Neural Syst. Rehab. Eng. 22 (5) (2014) 982–991.
[16] Zhelong Wangr, Ming Jiang, An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors, IEEE Trans. Inform. Technol. Biomed. 16 (4) (2012) 691–699.
[17] Sinziana Mazilu, Alberto Calatroni, Eran Gazit, Daniel Roggen, Jeffrey M. Hausdorff, Gerhard Tröster, Feature learning for detection and prediction of freezing of gait in Parkinson's disease, in: Machine Learning and Data Mining in Pattern Recognition Lecture Notes in Computer Science, vol. 7988, 2013, pp. 144–158.
[18] Michael A. Powell, Rahul R. Kaliki, User training for pattern recognition-based myoelectric prostheses: improving phantom limb movement consistency and distinguishability, IEEE Trans. Neural Syst. Rehab. Eng. 22 (3) (2014) 522–532.
[19] Marius Ene, Neural network-based approach to discriminate healthy people from those with Parkinson's disease, Ann. Univ. Craiova, Math. Comput. Sci. Ser. 35 (2008) 112–116.
[20] M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, L.O. Ramig, Suitability of dysphonia measurements for telemonitoring of Parkinson's disease, IEEE Trans. Biomed. Eng. 56 (4) (2008) 1015–1022.
[21] G. Rigas, A. Tzallas, M. Tsipouras, P. Bougia, E. Tripoliti, D. Baga, D. Fotiadis, S. Tsouli, S. Konitsiotis, Assessment of tremor activity in the Parkinson's disease using a set of wearable sensors, IEEE Trans. Inform. Technol. Biomed. 16 (3) (2012) 478–487.
[22] M. Djuric-Jovicic, N.S. Jovicic, I. Milovanovic, S. Radovanovic, N. Kresojevic, M. B. Popovic, Classification of walking patterns in Parkinson's disease patients based on inertial sensor data, Neural Network Appl. Elect. Eng. (2010) 3–6.
[23] Resul Das, A comparison of multiple classification methods for diagnosis of Parkinson disease, Expert Syst. Appl. 37 (2010) 1568–1572.
[24] Alexander Waibel, Phoneme recognition using time-delay neural networks, IEEE Trans. Acoust., Speech Signal Process. 37 (3) (1989) 328–339.
[25] Accessed from the site <http://physionet.org/pn3/gaitpdb/>.
[26] C.J.C.H. Watkins, Learning from delayed rewards, PhD thesis, Cambridge University, Cambridge, England.
[27] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams, Learning representations by back-propagating errors, Nature 323 (October) (1986) 533–536.
[28] J. Han, M. Kamber, Data Mining, Concepts and Techniques, second ed., Morgan Kaufmann, San Francisco, CA, 2001.
[29] H. Demuth, M. Beale, Neural network toolbox for use with MATLAB. Users guide. <http://www.mathworks.com> Edition, The Math Works Inc, 2013.
[30] M.T. Hagan, M. Menhaj, Training feed-forward networks with the Marquardt algorithm, IEEE Trans. Neural Networks 5 (6) (1999) 989–993.
[31] Moller, Neural Networks 6 (1993) 525–533.
[32] M.T. Hagan, H.B. Demuth, M.H. Beale, Neural Network Design, PWS Publishing, Boston, MA, 1996.
[33] S. Belciug, F. Gorunescu, Error-correction learning for artificial neural networks using the Bayesian paradigm. Application to automated medical diagnosis, J. Biomed. Inform. 52 (2014) 329–337.
[34] Donald W. Zimmerman, A note on interpretation of the paired-samples t test, J. Educat. Behav. Stat. 22 (3) (1997) 349–360.