Neural Comput & Applic (2017) 28:565–574
DOI 10.1007/s00521-015-2089-3

ORIGINAL ARTICLE

# Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach

**Vijay Bhaskar Semwal[1] · Kaushik Mondal[1] · G. C. Nandi[1]**

Received: 21 August 2015 / Accepted: 4 November 2015 / Published online: 17 November 2015


**Abstract** This current work describes human push recovery data classification using features that are obtained from intrinsic mode functions by performing empirical mode decomposition on different leg joint angles (hip, knee and ankle). Joint angle data were calculated for both open-eyes and closed-eyes subjects. Four kinds of pushes were applied (small, medium, moderately high, high) during the experiment to analyze the recovery mechanism. The classification was performed based on these different kinds of the pushes using deep neural network (DNN), and 89.28 % overall accuracy was achieved. The first classifier was based on artificial neural network on feed-forward back-propagation neural network (FF-BPNN), and second one was based on DNN. The proposed DNN-based classifier has been applied and evaluated on four types of pushes, i.e., small, medium, moderately high, high. The classification accuracy with a success of 88.4 % has been obtained using fivefold cross-validation approach. The analysis of variance has also been conducted to show the statistical significance of results. The corresponding strategies (hip, knee, and ankle) can be utilized once the categories of pushes (small, medium, moderately high, high) were identified accordingly push recovery (Semwal et al. in International conference on control, automation, robotics and embedded systems (CARE), pp 1–6, 2013).

**Keywords** Push recovery · IMF · EMD · DNN · Feature selection · Classification · ANOVA · FF-BPNN · Fivefold cross-validation

✉ Vijay Bhaskar Semwal
vsemwal@gmail.com

[1] Indian Institute of Information Technology, Allahabad, India

## 1 Introduction

Humanoid robot is designed for interacting with the physical environment, human tools, and many experimental purposes. This assiduity got succeed in various field, but still challenging to resemble that of the human body. Humanoid robots in the recent field of work have shown some excellent performance where the human cannot even reach. Human recovers from push-up to a certain extent. The ability of push recovery of human increases with age and also decreases with age. An adult person has better push recovery capability than a child, and elderly person has poor push recovery capability than younger [1]. In the field of robotics, the push recovery is still challenging because robot does not have that much physical strength and decision-making capability that human have. Humans are living infrastructures in the world that are already made for humankind [2]. Most of robots in the recent field of work are not exactly like human. So their working spaces are to be made differently with their comfort [3]. This causes additional amount of money. Our aim was to build up a robot that can use the same infrastructure in real time. Our experiment is a human-based experiment [4].

Many researchers are exploring the human gait and push recovery behavior for understanding the problem of elderly people [5] and understanding how human being do it very efficiently [6, 7]. To develop the computational model based on the joint trajectory value, we need features. To move toward more general artificial intelligence, we need the algorithms to automatically find the 'interesting' features that disentangle the data. Deep learning is a step toward identifying these 'interesting' representations. The goal is to automatically identify higher-level features from low-level features. Deep learning is used for the training of deep architectures such as multilayered perceptron which

has several hidden layers [8]. Deep architecture consists of multilevel nonlinear operations that transform the input data to better representation [9, 10]. Researchers also worked for the recognition of gait for different types of walk [11]. Hybrid automata-based model was used to describe the behavior of system which has two phases, continuous dynamics and discrete switching logic [12].

In this paper, we have analyzed the push recovery pattern. We extensively collected data using mobile-embedded accelerator with android application interface [4]. With the help of this android application, we have collected a huge number of data for 15 different subjects using our indigenously developed device. The push data were collected for applied push with different forces, and sometimes the process continues for more than one time for a particular subject [13, 14]. The leg joint angles were further calculated using inverse kinematics [15–17], and then classification was performed using features (Min, Max, RMS, Shannon entropy, log energy, entropy, zero-crossing rate) obtained from intrinsic mode functions (IMF) by performing empirical mode decomposition (EMD) [18, 19]. This analysis will help in the development of prosthetic leg for a paralyzed person, development of social robots, and understanding the problem of elderly people and the people who are suffering from gait problem who are deprived the sense of walking [20–22]. Then the collected data are classified using the artificial neural network (ANN) and DNN-based classifier, and after forming the fivefold cross-validation, it is obtained that DNN-based proposed classifier is better than ANN. The analysis of variance (ANOVA) has also been performed to show the statistical significance of results [23].

The organization of the paper is as follows: The second section is a methodology that starts with innovative data-capturing technique and the feature extraction for classification purpose (using DNN). The third section includes the result and its discussion. The calculations are also demonstrated to show the original output. In results section, a detailed theoretical analysis of classifier is done using fivefold cross-validation, and ANOVA is used to analyze statistical significance of results.

## 2 Methodology

### 2.1 Data capturing and feature extraction

Data-capturing technique changes with the available modern device in the field of research. Researchers used to use potentiometer and body-sensor-based human motion capture device (HMCD) [1] till 2014. The extracted data from that device were used to take several purifying techniques to get removed from the noise. The data-capturing technique proposed here is innovative and reliable. The software, named "physics tool box" powered by Google is used as an accelerometer sensor that works on android interface. Figure 1 shows the entire process of data collection. In this experiment, the subjects were selected of different age group, weight, and height. Four different kinds of pushes were applied with a known range of intensity from behind. The pushes are named as less, medium, high, and moderately high push [1, 24]. The range of less push starts with the 0 N. The highest value of moderately high push is the maximum value after that recovery is not possible which means zero sensing value. The data were collected for both left and right handed. Force-sensing resistor (FSR 3105) is used to measure the reading of the applied push on the back between spinal chord and the last rib of subjects [1]. The measured value is then converted into Newton using following formula.

$$\text{Force (N)} = f \times 9.8/100 \quad (1)$$

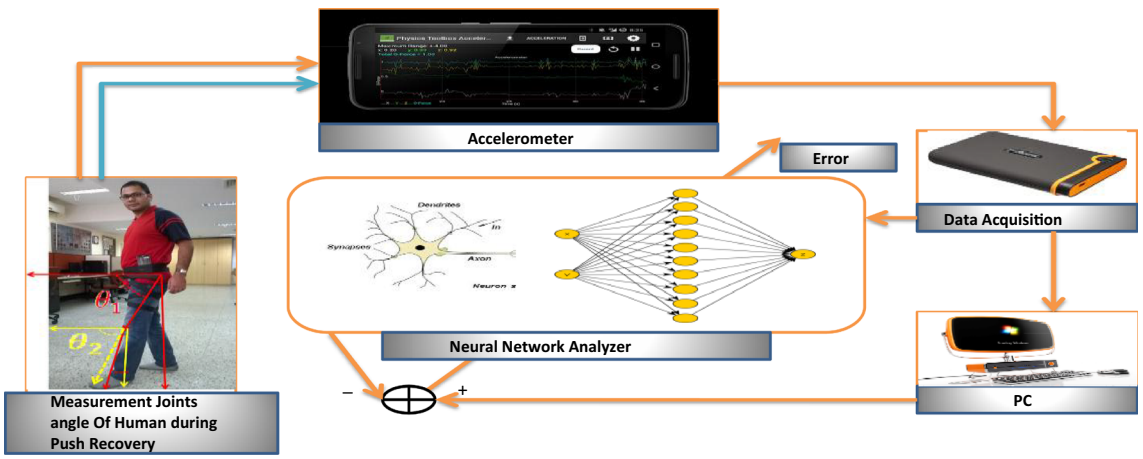


**Fig. 1** Extracting feature using accelerometer

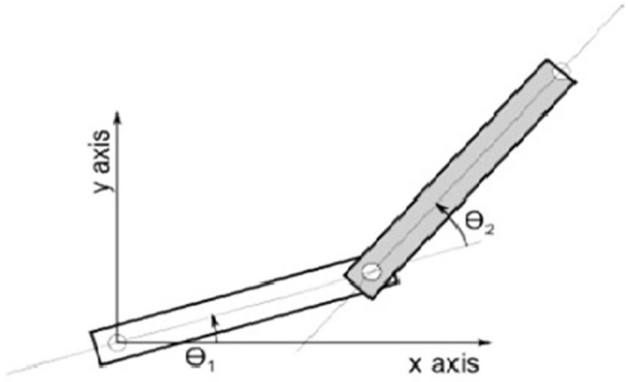


**Fig. 2** Diagram of a two-link manipulator

The forces were applied to apply the supervised learning with the main experimental data set. The push was applied up to few seconds to collect the data. With the known range of push, the data set is arranged separately for performing the next experiment. The data set is basically output of the "physics tool box" software, which is basically coordinate values of a moving two-link manipulator in a 2-D coordinate. Figure 2 is the representation of human leg as two-link manipulator. The android embedded sensor gives the output in a dot comma separated value (.csv) file. The inverse kinematics was solved to get the joint angles (hip, knee).

### 2.2 Joint angle calculation

Inverse kinematics [4] tells all the possible sets of joint angle and the geometries of link which can be used to get the orientation and position of the end effector from the given orientation and position of the link manipulator. First of all, we have to face the coordinate values from our extracted dot csv file, using that we will calculate the joint angle for two-link manipulator. Now using these coordinate values, inverse kinematics will be solved to get the corresponding joint angles. Figure 2 shows above a two-link manipulator with two different length and corresponding angle position.

The android application (physics tool box in Fig. 1) which is used accelerometer to collect the data of different joints. The application gives acceleration as well as the corresponding coordinates where the sensor is attached with respect to the origin (in our experiment it is hip-joint). The human's both the legs jointly can be considered as inverted pendulum [25, 26]. The sensor is attached at the hinge ankle. When the human body is in motion, then it looks like a two-link manipulator moving in 2-D coordinate with hip as origin. The hip and knee joint angles are calculated from the corresponding coordinate. The inverse kinematics was solved using the following Eqs. (2) and (3) to get the corresponding joint angles.

$$\theta_1 = a\tan 2(y, x) - a\tan 2(k_2, -k_1) \tag{2}$$

$$\theta_2 = a\tan 2\left(\pm\sqrt{1-\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right)^2}, \frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right) \tag{3}$$

where

$$k_1 = l_1 + l_2\cos\theta_2 \tag{4}$$

$$k_2 = l_2\sin\theta_2 \tag{5}$$

Here $l_1$ represents the length of the first link, and $l_2$ is the length of the second link, and in this experiment femur and fibula are shown in Fig. 1. Inverse kinematics tells all the possible sets of joint angle and the geometries of link which can be used to get the orientation and position of the end effector from the given orientation and position of the link manipulator.

The forces were applied in four different ways, i.e. small, medium, moderately high, and high from behind. FSR 3105 force-sensing resistor [1] was used to measure the intensity level of push. Here 0–3 N (unit of force) is treated as less push, 3–6 N as medium push, 6–9 as moderately high push, and 9–12 N as high push, and afterward 12 N there is no recover from push. This sensor is also used to measure ZMP-based stability [27] of biped robot. The extracted joint angles are then used in EMD process to get the intrinsic features [18].

### 2.3 Empirical mode decomposition

Empirical mode decomposition (EMD) is a very useful method for breaking down the data into finite and a very small number of components. EMD filters out the data set from a complete and orthogonal basis. The method of EMD shows the completeness and the way EMD is decomposed signify completeness [18, 19]. The obtained component after doing decomposition is called IMF that is intrinsic mode function.

IMF has following two properties:

1. Between zero-crossing, it has only one extreme
2. It has the mean value of zero

The Algorithm 1 shows how to calculate the IMF features of data.

```
Algorithm1: Shifting procedure
Begin
Identify all extrema of θ(t)
    Determine the upper envelop u(θ) from its local maxima
    Determine the lower envelop l(θ) from its local minima
Obtain the mean envelop, m(t) = [ u(θ) + l(θ)]/2
Subtract the mean from θ(t) , h(t) = θ(t) − m(t)
Check whether h(t) satisfies the properties of IMF
    If (Yes)
        The obtained h(t) is IMF
        Stop shifting
        Break
    Else
        θ(t) = h(t)
        Keep shifting
        Continue
End
```

Empirical mode decomposition (EMD) is a very useful method for breaking down the signal into finite and very small number of components. For the analysis of natural signal, this process is very useful, in which most of the cases are non-stationary and non-linear [20].

Empirical mode decomposition follows a shifting process to break down the signal into various intrinsic components, to extract IMF.

The shifting process:

Let the mean value of the original signal $x(t)$ of its upper envelop and lower envelop be $m_1$. Compute the first component $h_1$:

$$h_1 = x(t) - m_1 \tag{6}$$

Now in the second shift $h_1$ will be considered as a main data signal, and the mean of upper and lower envelop of $h_1$ be $m_{11}$, so the second component is:

$$h_{11} = h_1 - m_{11} \tag{7}$$

The shift process will be continued n times until $h_{1n}$ is an IMF component.

$$h_{1(n-1)} = h_{1n} - m_{1n} \tag{8}$$

Then it is called first IMF component as $k_1 = h_{1n}$ from the data, and it contains the shortest period component of main signal $x(t)$. We can extract and separate it from the remaining data $x(t) - k_1 = r_1$, this process will be repeated on $r_1$, $r_1 - k_1 = r_2,\ldots,r_{n-1} - k_n = r_n$.

Here IMF components have been computed for five times using EMD. The snapshot of the corresponding IMF has been illustrated below in Table 1.

### 2.4 The distribution of data

One famous technique is box plot, which graphically shows the distribution of a group of numerical data through their quartiles. The quartiles are the three points of data set that divide the data set into four groups. The middle number between the median of the data set and the smallest number is called first quartile, and the median of the data set is called the second quartile, and the middle number between the highest value of data set and the median of the data set is called third quartile. The middle number between third quartile and the first quartile is called interquartile range (IQR). The data in the range of more than three times of the IQR are called outliers, and data in the range of more than 1.5 times of the IQR are called suspected outliers.

A general figure of box plot that shows all the quartiles and outliers is shown in Fig. 3. Every IMF corresponding every joint angle has this kind of data distribution. If the data values are well synchronized, then that data set will be considered as best data set among all. Figure 4 illustrate the distribution of corresponding IMF. The next step is data classification. The classification is needed to get the exact result or the exact class for any unknown pattern as input from that analysis the robot configuration will be upgraded.

### 2.5 The statistical feature selection

The selection of feature is very important for a particular data set. The machine learning algorithm performs based on this statistical feature. Therefore, good quality of feature always gives the best result. The min is the minimum value of a particular data set.

$$d_{\min} = p \in x_i \tag{9}$$

The max of a data set gives the maximum value of that particular data set.

$$d_{\max} = q \in x_i \tag{10}$$

**Table 1** Corresponding IMF of every joint angle

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | −0.00033 | 0.000114 | 0.000416 | 0.000763 | 0.000377 | −0.00075 | 0.000695 | −0.00063 | 0.000568 |
| 2 | 0 | −0.00016 | 0.000304 | 7.72E−05 | −0.00055 | −0.00066 | 0.000199 | 0.000844 | −2.11E−05 | −0.00098 |
| 3 | 0 | −0.00033 | −0.00011 | 0.000423 | 0.000989 | 0.0001362 | 0.00145 | 0.001261 | 0.000815 | 0.000124 |
| 4 | 0 | −0.00018 | −0.00023 | −0.00018 | −5.53E−05 | 0.00012 | 0.000319 | 0.000516 | 0.000686 | 0.000802 |
| 5 | 0 | −0.00034 | −0.00054 | −0.00061 | −0.00057 | −0.00044 | −0.00025 | −4.72E−06 | 0.000274 | 0.000567 |

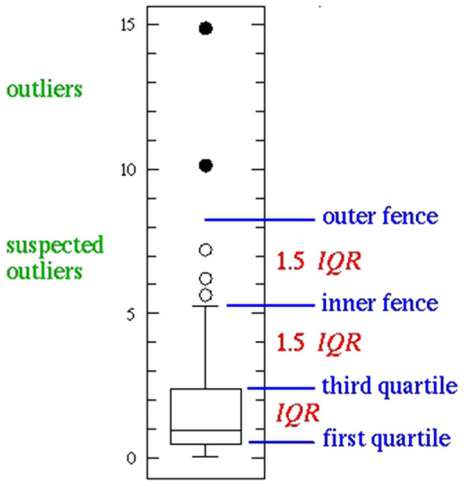


**Fig. 3** Generalize diagram of box plot

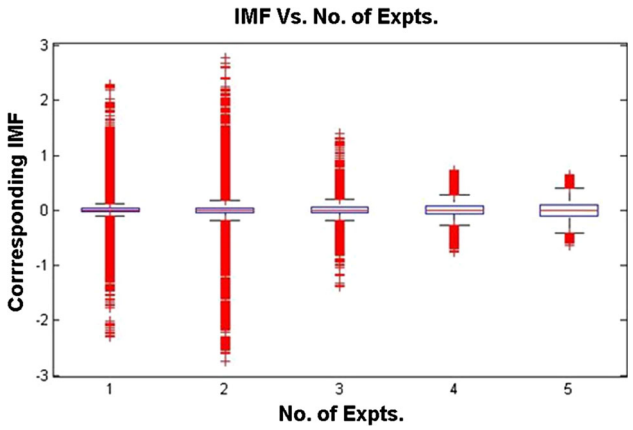


**Fig. 4** Box plot of corresponding IMF

The next selected feature is Shannon entropy, which is the most important feature of a particular data set. In Information theory the expected value (average value) is known as the entropy. Entropy always categorizes the uncertainty about any source about information. The Shannon entropy in following equation which gives the average information of the data set:

$$H(X) = -\sum_{i=1}^{N} p(s_i) \log_2 p(s_i) \tag{11}$$

The Log Energy Entropy of IMF is calculated as follows:

$$\text{Energy} = \sum_{i=1}^{N} \log\left(s_i^2\right) \tag{12}$$

The quadratic mean in statistics is known as the root mean square (RMS). It is the statistical measurement data that defined square root of the mean of squares of a sample.

$$f_{\text{rms}} = \lim_{T\to\infty} \sqrt{\frac{1}{T}\int_0^T f(t)^2 \mathrm{d}t} \tag{13}$$

Another important feature is a zero-crossing rate (ZCR). It is the number of the sign changes of a signal along a particular axis. It is shown in the following equation:

$$\text{ZCR} = \frac{1}{T-1}\sum_{t=1}^{T-1} \prod(s_t s_{t-1} < 0) \tag{14}$$

where $T$ is the length of the signal $s$. $\prod(A)$ is called the indicator function. So the indicator function is 1 if its argument $A$ is true and 0 if $A$ is false. In the field of music information retrieval and speech recognition, this feature is heavily and now it is the key feature to classify percussive sounds. In many experimental cases instead of all crossing, only the positive going and negative is counted. In our experiment, this feature is also important because of the gait cycle that oscillatory in nature. So based on the applied pushes, we can easily calculate that how frequently the gait cycle changes. So these are the statistical feature that is considered as main data set for the classification purpose. The ANN was introduced to classify the data set. These two features give the unique information of a particular data set. The remaining four features Min, Max, RMS, and ZCR were further calculated using simple MATLAB function.

### 2.6 Feature extraction

Extraction of unique feature always helps to enhance the classification performance. There are set of features extraction method has been proposed so far. In this study, we have used six basic statistical measurement of the signal and found unique patterns among the Users. As above-described EMD methodology for joint angle signals to split it into number of intrinsic mode, we have extracted statistical features of the IMFs. After decomposition of original signal into different number of IMFs, we have extracted the statistical features of all IMFs to give the input to classifier for subject identification. The above table is showing the extracted features name.

Algorithm 2 is our designed filter for data smoothing, which is inspired from moving average method

Algorithm 2: Using moving average and calculate the least square error up to=.0001 and number of step initial length/2

Input: InputData (x[n]), Length OfData ($l_e$), RootMeanSquare(rms)
Output: SmoothedData( $S_1$[n]); CalcRootMeanSquare(rms1)
Initial: $l_t \leftarrow x[n]$; rms1 = rms;

```
Begin
        l_e = Length(x[n])
    x = (x(1:l_e − 2) + x(2:l_e − 1) + x(3:l_e))/3;
      rms1=rms(x)
     While rms > 0.0001and length(x)> l_e/2
          X = (x(1:l_e − 2) + x(2:l_e − 1) + x(3:l_e))/3;
     rms = rms(x)
     End while
    x= Imresize (x, l_e);
End Begin
```

The performance of machine learning algorithms depends on the several factors. The selection of accurate features from right problem is very important for better performance. The DNN is one of the choices.

### 2.7 Deep learning process model

Multilayer feed-forward neural networks (MFNN) [28] designed for the problem uses multiple neurons interconnected in basic format of input layer, multiple hidden layers, and an output layer, respectively. The basic functionality of multilayer neural networks is the same as that of a traditional one-hidden-layer neural network [20]. The extracted features are fed into the input layer, each of the hidden layers processes the features and generates a set of projected features which are more compact and represent better attributes to classify the data. The final layer is the output layer which processes the data from the last hidden layer and classifies the data into one of the several possible classes.

In multilayer neural network, as the depth of network increases, the error gradients become very small for back propagation and diminish eventually. So it does not contribute the starting layer of the network. Due to poor learning of network, it leads to under fitting and as we increase the size of network the parameter, i.e., bias and weight also increases. So we explore the space of very complex function, which needs large amount of data for training else it will lead for over fitting (curse of dimensionality). The positive results are obtained only up to two or maximum three layers. More than three layers the network will give poor results. Then in 2006 Hinton et al. introduced deep belief networks (DBNs) which were trained greedily, i.e., one layer at a time in an unsupervised manner and with unlabelled data using the restricted Boltzmann machine (RBM). The Fig. 5 correspond to auto-encoder network and Fig. 6 represents the stochastic gradient decent process [10].

One of the fundamental differences between deep learning and a general multi-layered neural network is

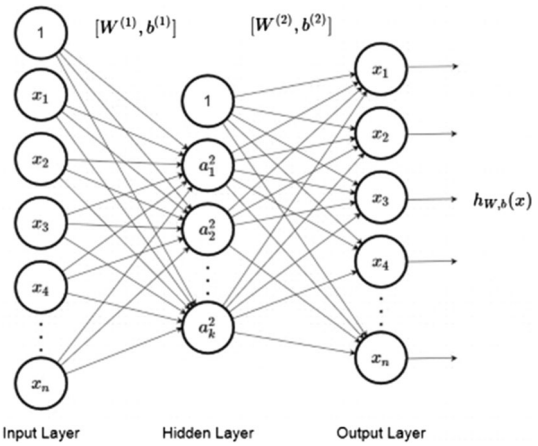


**Fig. 5** Auto-encoder

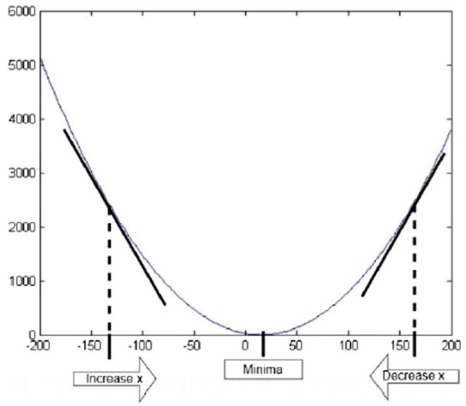


**Fig. 6** Stochastic gradient descent (SGD)

learning. In deep learning, the initial training is done layer-by-layer, starting from the first hidden layer to the last, ensuring that each layer represents vital features useful for classification. The weight adjustment and learning rate are adjusted with bias $b$ value. We need to minimize the errors Eq. (15) representing the loss function. Let us assume $\omega$ is collection of weight matrix $\{\omega_i\}$, $1 \leq i \leq N$, where $\omega_i$ represents weights connecting $i$ layer to $i+1$ and $N$ represents the number of layers; similarly $\beta$ a collection of biases $\{\beta_i\}$, $1 \leq i \leq N$, where $\beta_i$ is collection of biases at layer $i$. The loss function needs to be minimized given by Eq. (15)

$$\mathcal{L}(\omega, \mathcal{B}|j) \tag{15}$$

Deep learning is a hierarchical feature extraction architecture for multilayer basic framework having a major advantage that it can handle nonlinearity in data for training and testing purposes. We use stochastic gradient descent (SGD) which is memory efficient and very fast with HOGWILD [22] for supporting shared memory model, and therefore, training and testing data are able to execute on multinode system. The general procedure is shown in Algorithm 3. The architecture of deep learning is

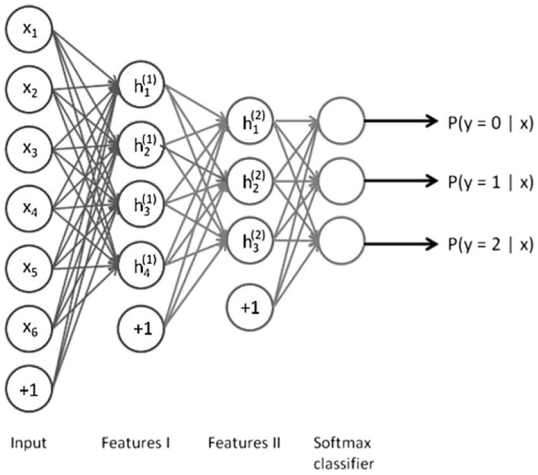


**Fig. 7** Deep learning architecture

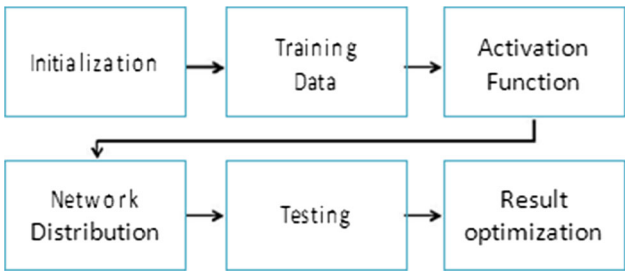


**Fig. 8** Deep learning process model

shown in Fig. 7. The general procedure of using DNN is shown in Fig. 8.

Algorithm 3:

Initialize $W, B$ for h data Load training data $T$ for all nodes not converge criteria satisfied

1. For each node $n$ training $T_k$, do parallel
2. $\omega_n, \beta_n \leftarrow$ global model (seed value, will update after )
3. 3. Select$T_{k\alpha} \subset F_k$ ($F_k$ is samples for all iteration)
4. Divide $T_{k\alpha}$ into $T_{k\alpha c}$ from $k_c$ (number of cores$k_c$)
5. For $T_{k\alpha c}$ on node $k$, do parallel
6. Initialize training $i \in T_{k\alpha c}$
7. Update $\omega_{ij} \in \omega_n$ and biases $b_{ij} \in \beta_n$
8. $\omega_{ij} := \omega_{ij} - \alpha \times \frac{\delta L(w,\beta|j)}{\delta \omega_{ij}}$
9. $b_{ij} := b_{ij} - \alpha \times \frac{\delta L(w,\beta|j)}{\delta b_{ij}}$
10. $W := Average_k \times W_k$
11. $B := Average_k \times B_k$

($Average_k$ for final local iteration averaging for each node $k$ step 1 to 9)

The classification is then performed using these statistical features using deep neural network (DNN) shown in Fig. 3. The network we proposed here having five hidden layer with 100, 50, 25, 10, 10 neurons in it. The input dropout ratio is set to 0.2, and hidden dropout ratio is set to 0.5 for every layer. The activation function is used here is tanh($x$). The network is trained in supervised manner, because the pushes were applied with known range of force parameter [24]. The training of the DNN architecture is

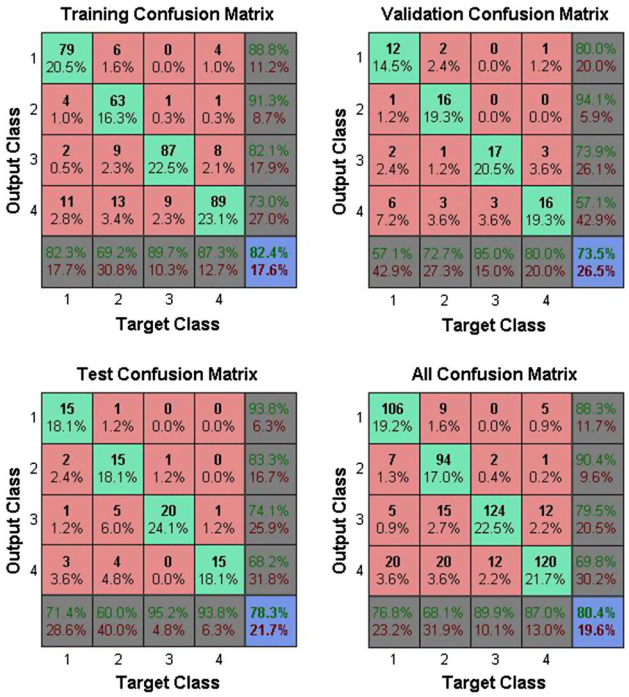


**Fig. 9** Confusion matrix

done by gradient decent method in Eq. (16). The back-propagation algorithm did not help because it does not work when the number of hidden layers is large. As the depth of the network increases, the error gradient that is back-propagated becomes very small and diminishes eventually.

$$\Delta\omega_{ij}(t+1) = \Delta\omega_{ij}(t) + \eta \frac{\delta C}{\delta \omega_{ij}} \tag{16}$$

where $\eta$ learning rate and $C$ is cost function. The cost function is chosen based on learning type (supervised or unsupervised) and the activation function.

## 2.8 Fivefold cross-validation

It is a technique for validation. In the next step, fivefold cross-validation was performed to analyze the validity and performance of the classifiers. During the fivefold cross-validation process, the total dataset was divided into five equal parts. Among the total five parts of the dataset, one part will be used as testing set and remaining will be used as training set in every phase of the experiment. Thus, a total of five experiments will be performed and accordingly five results will be generated. Finally, the average of the five results will provide us the final accuracy of the classifier used in our system.

Algorithm 4:

Begin

*Place the training sample into some random order.*

*Divied the smaples into* 5 *fold. i.e.* 5 *chunks of approminately* $n/5$ Size each

*for* $i = 1:5$

*Train the all sample which do not belong to* $i^{th}$ *fold*

*Test the classifer on all the smaple of* $i^{th}$ *fold.*

*Compute the* $n_i$ *, no of smaples which are misclafied into* $i^{th}$ *fold.*

*end for*

*Compute* $E = \frac{\sum_{i=1}^{5} n_i}{n}$

*To achieve the good accuracy of classifier, the* $5-$ *fold cross validation is run many times.*

*let* $E_1, E_2, \ldots, E_p$ *be the accuracy estimate obtained in p run*

$e = \frac{\sum_{j=1}^{p} E_j}{p}, V = \frac{\sum_{j=1}^{p}(E_j - e)^2}{p-1}, \sigma = \sqrt{v}$

*Where is e estimated error and* $\sigma$ *is stnadar deviation*

End Begin

## 3 Results and discussion

Ideally, the human joint angle oscillates sinusoidally with respect to the time when the body in motion. The joint angle (knee) obtained from the fundamental experiment is different than ideal one. Researchers already worked on the analysis of variation in leg joint angle in different environment using different data-capturing technique. The idea is proposed here is totally innovative and reliable. The confusion matrix obtained from the experiment as output is shown below in Fig. 9.

The accuracy of the individual class is calculated using following equation:

$$\text{Accuracy (ACC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{17}$$

Table 2 shows the accuracy of the individual class. Table 3 shows classification accuracy obtained using ANN on different chosen parameters. Table 3 gives the result of performance using ANN using different number of neuron and epoch and Table 4 gives the performance matrix using DNN.

The classification results are tabulated in Tables 5 and 6 using fivefold cross-validation for ANN and DNN classifier.

Table 4 shows classification accuracy obtained using DNN on different chosen parameters, and Table 7 gives the comparison of success rate by different classifier using fivefold cross-validation.

Table 7 shows the results in terms of the correct classification rates of the proposed technique against three existing classifiers. From results in Table 7, it can be concluded that the proposed technique achieves the highest classification rate of 89 % with only 130 misclassifications.

Tables 8 and 9 show the result obtained after performing the one-way ANOVA. The ANOVA is performed on the result obtained from fivefold cross-validation of both classifiers. As here, only two classifier so $t$ test and ANOVA will give same results. The DNN-based

**Table 2** Accuracy of the individual class

| | True positives (TP) | False positives (FP) | False negatives (FN) | True negative (TN) | Accuracy (ACC) (%) |
|---|---|---|---|---|---|
| Small class | 99 | 39 | 38 | 360 | 85.63 |
| Medium class | 90 | 48 | 6 | 389 | 89.86 |
| Moderate high class | 128 | 10 | 33 | 381 | 92.2 |
| High class | 106 | 32 | 52 | 362 | 84.7 |
| Overall accuracy (%) | 76.63 | | | | |

**Table 3** Overall confusion matrix of ANN using different number of neuron and epochs

| Hidden neurons | Epoch | Training accuracy (%) | Testing accuracy (%) | Overall accuracy (%) |
|---|---|---|---|---|
| 30 | 20 | 82.4 | 78.3 | 80.4 |
| 20 | 15 | 80.6 | 75.9 | 76.6 |
| 15 | 10 | 79.9 | 74.3 | 75.5 |
| 10 | 10 | 79.0 | 74.0 | 75.0 |

**Table 4** Performance matrix using deep neural network

| Hidden neurons | No. of hidden layer | Epoch | Training accuracy (%) | Testing accuracy (%) | Overall accuracy (%) |
|---|---|---|---|---|---|
| 150, 100, 60, 40, 20 | 5 | 20 | 90.6 | 85.71 | 87.4 |
| 100, 60, 40, 20 | 4 | 15 | 87.4 | 80.14 | 82.2 |
| 60, 40, 30 | 3 | 10 | 92.4 | 85.71 | 89.28 |
| 50, 30, 20 | 3 | 10 | 89 | 82.14 | 84.28 |

**Table 5** Fivefold cross-validation result using artificial neural network (ANN) classifier

| Exp# | Accuracy (%) |
|---|---|
| Subset 1 | 76 |
| Subset 2 | 82 |
| Subset 3 | 79 |
| Subset 4 | 81 |
| Subset 5 | 77 |
| Overall | 79 |

**Table 6** Fivefold cross-validation result using deep neural network (DNN) classifier using chosen parameter

| Exp# | Accuracy (%) |
|---|---|
| Subset 1 | 90 |
| Subset 2 | 87 |
| Subset 3 | 86 |
| Subset 4 | 89 |
| Subset 5 | 90 |
| Overall | 88.4 |

**Table 7** Comparison of success rate by different classifier using fivefold cross-validation

| Classifier | No. of sample | No. of misclassified | Success rate (%) |
|---|---|---|---|
| ANN classifier | 1000 | 170 | 83 |
| DNN classifier (proposed) | 1000 | 110 | 89 |

**Table 8** Analysis of variance (ANOVA)

| Group | Count | Sum | Average | Variance ($\sigma$) |
|---|---|---|---|---|
| ANN classifier | 5 | 395 | 79 | 6.5 |
| DNN classifier (proposed) | 5 | 434 | 86.8 | 3.3 |

**Table 9** ANOVA analysis details: ANOVA: single-factor group variation

| Source of variation | Sum of square | Degree of freedom | MS | $F$ | $p$ value | $f$ critic |
|---|---|---|---|---|---|---|
| Within | 20.43 | 1 | 20.43 | .89 | .67 | 4.2 |
| Between | 182.23 | 8 | 22.77 | | | |
| Total | 202.66 | 9 | | | | |

classification is compared with ANN in order to see where the difference is significant enough to reject hypothesis (H1). The null hypothesis will be (H0) there is no statistical variance between DNN and ANN classifier (H0). To verify the statistical significant improvement in hypothesis H1, an ANOVA is no statistical variance between DNN and ANN classifier (H0), whereas alternative hypothesis there is significant difference between classification accuracy (H1).

The value of $p = .67$ indicates that one should reject the null hypothesis in favor of the alternative. It shows there is significant difference as null hypothesis has rejected (H0) and alternative hypothesis accepted (H1).

## 4 Conclusions

Our experiment mainly focuses on two basic parameters hip, knee, and ankle, which are most important for any humanoid robot or elderly person for walking and push recovery. The EMD-based feature extraction technique had been shown to be effective for the classification of four different kinds of push recovery data. The experiment shows the features extracted from IMF do not degrade the accuracy of the system. These parameters are better optimized with DNN; hence, we are getting more than 89.92 % results. Based on the experimental results, we can conclude that the proposed technique is suitable for push recovery data classification and can achieve over 88.4 % accuracy. Extensive experiments using fivefold cross-validation show the effectiveness of the proposed technique. An ANOVA test was performed on fivefold cross-validation results also shows that performance of DNN-based classification is statistically significant rather than using ANN based. Once the pushes are classified, appropriate push recovery mechanism can be implemented on it accordingly [1].

**Acknowledgments** The authors would like to thank all the research scholars, M.Tech. students and technical staffs for their comments and suggestions. At the same time, our sincere thanks to IIIT, Allahabad, for providing us all the necessary facilities for research.

## References

1. Semwal VB, BhushanA, Nandi GC (2013) Study of humanoid Push recovery based on experiments. In: International conference on control, automation, robotics and embedded systems (CARE), pp 1–6
2. Torres C et al (2014) Stable optimal control applied to a cylindrical robotic arm. Neural Comput Appl 24(3–4):937–944
3. Nilakantan JM et al (2015) Bio-inspired search algorithms to solve robotic assembly line balancing problems. Neural Comput Appl 26:1379–1393. doi:10.1007/s00521-014-1811-x
4. Semwal VB, Nandi GC (2015) Toward developing a computational model for bipedal push recovery—a brief. Sens J IEEE 15(4):2021–2022
5. Nandi GC et al (2009) Development of adaptive modular active leg (AMAL) using bipedal robotics technology. Robot Auton Syst 57(6):603–616
6. Iqbal S, Zang X, Zhu Y, Saad HMAA, Zha J (2015) Nonlinear time-series analysis of different human walking gaits. In: 2015 IEEE international conference on electro/information technology, At Naperville, IL, USA

7. Semwal VB, Raj M, Nandi GC (2015) Biometric gait identification based on a multilayer perceptron. Robot Auton Syst 65:65–75
8. Iamsa-at S, Horata P (2013) Handwritten character recognition using histograms of oriented gradient features in deep learning of artificial neural network. In: 2013 international conference on IT convergence and security (ICITCS), pp 1–5
9. Baptista D, Morgado-Dias F (2013) A survey of artificial neural network training tools. Neural Comput Appl 23(3-4):609–615
10. Gao S, Zhang Y, Jia K, Lu J, Zhang Y (1999) Single sample face recognition via learning deep supervised auto-encoders. IEEE Trans Inf Forensics Secur PP(99):1
11. Semwal VB et al (2013) Biped model based on human gait pattern parameters for sagittal plane movement. In: IEEE international conference on control, automation, robotics and embedded systems (CARE)
12. Semwal VB et al (2015) Biologically-inspired push recovery capable bipedal locomotion modeling through hybrid automata. Robot Auton Syst 70:181–190
13. Mao W, Kim J-J, Lee J-J (2009) Continuous steps toward humanoid push recovery. Automation and logistics, 2009. In: IEEE international conference on ICAL'09, pp 7–12
14. Semwal VB, Chakraborty P, Nandi GC (2015) Less computationally intensive fuzzy logic (type-1)-based controller for humanoid push recovery. Robot Auton Syst 63:122–135
15. Chowdhury S, Verma B, Stockwell D (2015) A novel texture feature based multiple classifier technique for roadside vegetation classification. Expert Syst Appl 42(12):5047–5055
16. Iqbala S, Zang X-Z, Zhu Y-H, Bie D-Y, Wang X-L, Zhao J (2015) Nonlinear time-series analysis of human gaits in aging and Parkinson's disease. In: 2015 international conference on mechanics and control engineering (MCE 2015)
17. Zhou C et al (2013) Backward swimming gaits for a carangiform robotic fish. Neural Comput Appl 23(7–8):2015–2021
18. Lin L, Hongbing J (2009) Signal feature extraction based on an improved EMD method. Measurement 42(5):796–803
19. Ibrahim RK et al (2008) Gait pattern classification using compact features extracted from intrinsic mode functions. In: Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th annual IEEE international conference
20. Ben X, Zhang P, Yan R, Yang M, Ge G (2015) Gait recognition and micro-expression recognition based on maximum margin projection with tensor representation. Neural Comput Appl. doi:10.1007/s00521-015-2031-8
21. Yildirim Ş, Eski İ, Polat Y (2013) Design of adaptive neural predictor for failure analysis on hip and knee joints of humans. Neural Comput Appl 23(1):73–87
22. Kim S-Y, Yang L, Park IJ, Kim EJ, Park MS, You SH, Kim Y-H, Ko H-Y, Shin Y-I (2015) Effects of innovative WALKBOT robotic-assisted locomotor training on balance and gait recovery in hemiparetic stroke: a prospective, randomized, experimenter blinded case control study with a four-week follow-up. IEEE Trans Neural Syst Rehabil Eng 23(4):636–642
23. Procházka A et al (2014) Discrimination of axonal neuropathy using sensitivity and specificity statistical measures. Neural Comput Appl 25(6):1349–1358
24. Mao W, Qin G, Lee J-J (2009) Humanoid push recovery strategy for unknown input forces. In: Mechatronics and automation, 2009. International conference on ICMA 2009, pp 1904–1909
25. Tang Z, Er MJ, Chien C-J (2008) Analysis of human gait using an inverted pendulum model. In: IEEE international conference on fuzzy systems, 2008 (IEEE world congress on computational intelligence), pp 1174–1178
26. Li Y, Tong S, Li T (2013) Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven DC motor via backstepping. Nonlinear Anal Real World Appl 14(1):483–494
27. Sorao K, Murakami T, Ohnishi K (1997) A unified approach to ZMP and gravity center control in biped dynamic stable walking. In: Final program and abstracts. IEEE/ASME international conference on advanced intelligent mechatronics '97, pp 112
28. Kankal M, Yüksek Ö (2014) Artificial neural network for estimation of harbor oscillation in a cargo harbor basin. Neural Comput Appl 25(1):95–103