A Self-Monitoring Online Sequential Learning Mechanism for Feedfoward Neural Networks

Skanda S. Bharadwaj[1], R Chandan Kumar[1], B. N. Sumukha[1], and Koshy George[1,2]

[1]PES Centre for Intelligent Systems

[2]Department of Telecommunication Engineering

PES Institute of Technology (now, PES University)

100 Feet Ring Road, BSK 3rd Stage, Bangalore 560085, India

***Abstract*—The objective of online sequential learning is to make decisions on-the-fly. In this paper, we make a case for online sequential learning in the context of human activity recognition. Moreover, a mechanism to monitor learning online is introduced so as to avoid over-training and to reduce computational complexity. We consider a feedforward neural network with a single hidden layer for faster learning.**

***Keywords*—Feedforward neural networks, human activity recognition, online sequential learning.**

I. Introduction

Human Activity Recognition (HAR) is an active area of research in general. In particular, automatic recognition of activities has generated tremendous interest in the recent past. A contributing factor for an increased interest in HAR is perhaps the ubiquitous presence of 'smart' devices. Cognitive and interactive environments are relatively easier to develop using such devices. For example, human activity and motion disorder recognition is considered in [1]. Integration of the facets of human behaviour into devices such as smart phones results in better interaction with the environment.

Typical activity recognition activities are carried out with the information obtained from a camera [2], or with body-mounted sensors [3]. Vision-based techniques are based on a number of different methodologies; e.g., optical flow models in [4], and Kalman filtering and Hidden Markov Models (HMM) in [5]. Quite often, data mining, statistical modeling and machine learning techniques are used in sensor-based activity recognition. In this paper, we use the latter approach.

The majority of the work in activity recognition is based on supervised learning. Few exceptions are [6] and [7]. Whilst the former deals with semi-supervised learning, the latter deals with unsupervised learning. Different networks are also considered. For example, feed-forward neural networks are used in [8] and multi-class support vector machine in [9]. Other techniques include predictive models like binary decision trees and threshold based classifiers respectively in [10] and [11], and probabilistic models like HMM and Naïve Bayes' classifiers respectively in [12] and [13]. In the context of activity detection, a few authors have also worked on issues regarding implementations; e.g., [14].

Offline learning of information from data is the predominant method of training artificial neural networks with very little or no work reported that considers online sequential learning. We emphasise that the latter is more suitable for real-time applications. Further, offline methods are chosen by researchers as they are amenable for feature extraction. An added advantage is that of data compression. However, these are in direct contrast to the human cognitive process where it is primarily online and sequential, and decisions taken on-the-fly. The focus of this paper is to consider online sequential machine learning for human activity recognition. An implication of this is that data from the sensors are used directly without recourse to feature extraction.

The classical learning method for feed-forward artificial neural networks (FFNNs) is the back propagation algorithm (BPA). This method is applicable for networks of arbitrary complexity in terms of the number of hidden layers as well as neurons per layer. This algorithm can be used for online sequential supervised learning wherein the input-output pairs are shown one by one and then discarded. However, this method is notoriously slow and often gets trapped in a local minimum. In this paper, we consider FFNNs with a single hidden layer and consider a form of sequential training that can result in faster convergence.

In view of the aforementioned points, the goal of this paper is as follows: We propose a method for human activity recognition that does not require feature extraction, allows online sequential learning, and provides classification accuracy comparable to conventional techniques. This proposed technique also incorporates a scheme to monitor learning. Learning can be stopped when an 'optimal' level is achieved.

The methodology is discussed in Section II, and the learning method is explained in Section III. The results are presented in Section IV.

II. Methodolgy

We consider the open source UCI Machine Learning Repository for the database of signals provided by Smart Labs

978-1-5090-5256-1/16/$31.00 ©2016 IEEE

[15]. The signals correspond to a group of 30 subjects in the age group between 19 and 48. Of the six basic daily-life considered activities, three correspond to subjects with static posture – subjects standing, sitting or lying down – and three correspond to moving subjects – subjects walking, climbing up and down a staircase. Each subject wore a Samsung Galaxy SII smart-phone and followed a protocol of activities. Each subject stood for 30 seconds, sat for the next 15 seconds, and then stood for the next 15 seconds. This was followed by the subject lying down for 15 seconds. Subsequently, they sat for 15 seconds and lay down again for the next 15 seconds. After walking for the next 30 seconds, they alternated between climbing down and up a staircase for three cycles, with each activity lasting for 12 seconds. Between each task the subject rested for 5 seconds. This protocol of activities was repeated twice. The variation was in the location in the sensor. In the first set of activities, the smart phone was mounted on the waist, and in the second set, the location of the phone was left to the choice of the subject.

The total duration of measurement is 192 seconds. The specific measured signals were the outputs of the accelerometer and the gyroscope. The accelerometer provided signals along the three axes of an inertial frame of reference. The gyroscope measured the angular velocities along the same reference frame. With six signals per subject, there a total of 856 signals were extracted from the dataset. A sample of the raw signals acquired from the sensors for the different activities are shown in Figures 1–3. The sampling frequency is 50 Hz.

The objective in this paper is to differentiate and classify the above-mentioned activities on-the-go using FFNNs whilst simultaneously monitoring the training. The networks are trained sequentially. The network decides to stop training when an optimal level of learning is achieved. Thus, the proposed methodology closely resembles the working of the human cognitive process wherein the signals (e.g., speech, visual, smell) are processed sequentially without any explicit feature extraction. The proposed method has the ability to know how much of the data is required to attain a prescribed accuracy and thus the method has the potential to prevent unnecessary training as learning is monitored.

The proposed methodology is depicted in Fig. 4, and consists of three feed-forward neural networks (FFNNs), one for each axis; these are denoted $\boldsymbol{\mathcal{N}}_x$, $\boldsymbol{\mathcal{N}}_y$ and $\boldsymbol{\mathcal{N}}_z$. Classification and monitoring of learning is based jointly on the decisions provided by the three networks. Accordingly, a committee of networks is used to classify the human activities. Moreover, since learning is online sequential, pre-processing of signals and feature extraction is entirely avoided. Thus the inputs to the networks are the signals that have been acquired from the sensors. The monitoring of training is discussed in the sequel.

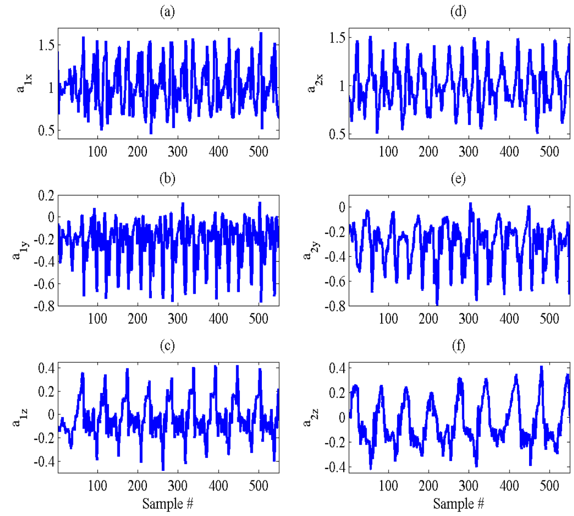

Fig. 1. Signals acquired from the sensors for different activities: (a) – (c) – Walking. (d) – (f) – Climbing up the stairs.

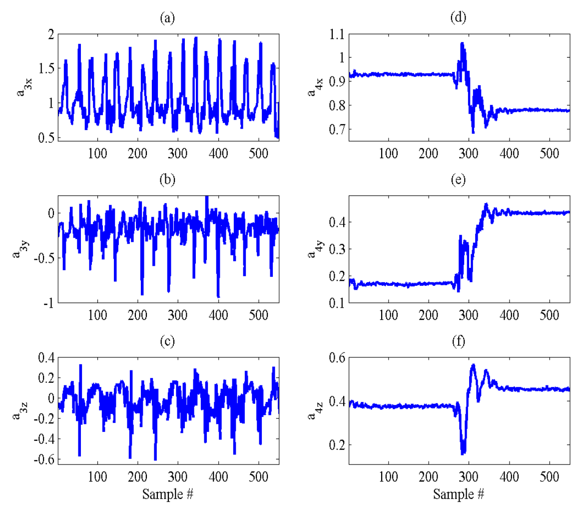

Fig. 2. Signals acquired from the sensors for different activities: (a) – (c) – Climbing down the stairs. (d) – (f) – Sitting.

Both present and previous values of the signals serve as inputs to the feedforward neural networks. This can be represented as a tapped delay line and depicted as the block TDL in Fig. 4. In this paper we consider a window (TDL) of length 20 and slide it by five in order to reduce the computational complexity. An overlap of 15 samples is created which is the block **SW** in Fig. 4. The length of the window or TDL and amount by which it slides are arrived at carefully by trial and error method.

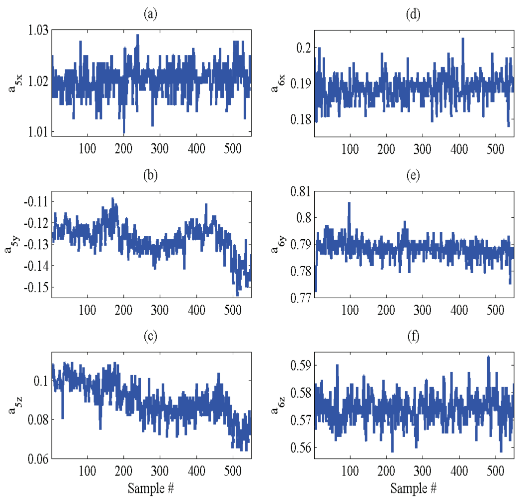

Fig. 3. Signals acquired from the sensors for different activities: (a) – (c) – Standing. (d) – (f) – Lying down.

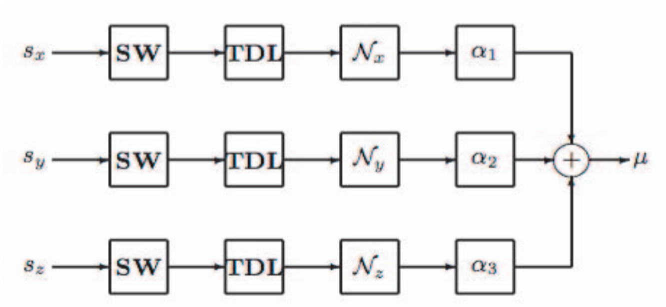

Fig. 4. The schematic for the proposed methodology. Here, **SW** is the sliding window, **TDL** is the tapped delay line, $\mathcal{N}_x$, $\mathcal{N}_y$ and $\mathcal{N}_z$ are the FFNNs for the *x*, *y* and *z*-axes, and α_1, α_2 and α_3 are constants.

The number of outputs of each network is six and corresponds to the number of classes. Accordingly, for every window of 20 samples, each network yields an output vector that belongs to $\mathbb{R}^6$. For a total of M windows, the outputs are concatenated into a matrix of dimensions $6 \times M$. The output matrices for the x, y and z axes are respectively denoted O_x, O_y and O_z. The class information is extracted from the averages of the six rows Thus,

$$\mu_a = \frac{1}{M}\begin{pmatrix} \sum_{j=1}^{M} O_{a,1,j} \\ \sum_{j=1}^{M} O_{a,2,j} \\ \vdots \\ \sum_{j=1}^{M} O_{a,6,j} \end{pmatrix}$$

where a is either x, y, or z. Obviously, the location of the maximum of the six elements of μ_a is the class to which the signal belongs. In this paper we consider a convex combination of the outputs of the three networks to determine the class. That is,

$$\mu = \alpha_1\mu_x + \alpha_2\mu_y + \alpha_3\mu_z$$

where the constants are such that $0 \leq \alpha_i \leq 1$, $\alpha_1 + \alpha_2 + \alpha_3 = 1$, and $\mu = (\mu_1 \; . \; . \; . \; \mu_6)^T$. The class is determined as follows:

$$c = \arg\max_{1 \leq i \leq 6} \mu_i$$

During the training phase, c is the desired class of the signal.

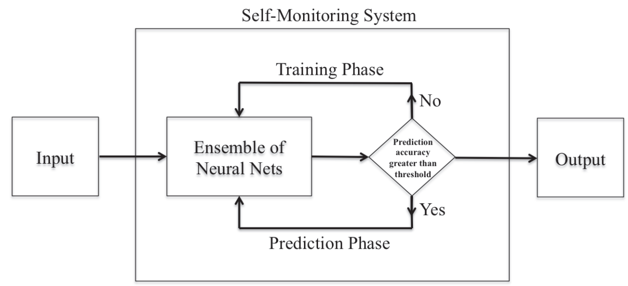

Fig. 5. FFNN with self – monitoring mechanism

The self-monitoring mechanism in the methodology is depicted in Fig. 5. As mentioned earlier, the set-up encompasses three feedforward neural networks, one for each axis (x, y and z), which we refer to as an ensemble of neural networks. The monitoring phase of the network includes a feedback of accuracy from the output of the system. The network uses the feedback accuracy for self-assessment. This is achieved by continuous monitoring of the classification. The network is designed in such a way that every incoming signal is first sent into classification. Obviously, the error is very high initially. Subsequent training will ensure the reduction in misclassification. Average accuracy is determined dynamically and continuously for every 100 signals. The average accuracy is computed by considering the predicted output against the ground-truth labels of the input data. This average accuracy is fed back to the network to take a decision on the training. As long as the average accuracy of the network is below a certain prescribed threshold the network continues to train itself with the incoming data. If the classification accuracy averaged over the 100 signals increases more than the prescribed threshold, the network takes the decision to stop training and enters into the prediction phase. At this point, the network just computes the output of the network for the input signal. Once the "YES" condition is met the network starts predicting the output. The training path is not traversed anymore. Also, the prediction phase is only traversed once. In the succeeding cycles the output of the network is obtained.

III. Sequential Learning of FFNN

For a feedforward neural network with one hidden layer suppose that the input layer has m_0 nodes, the hidden layer has m_1 neurons and the output layer has m_2 neurons. We denote this network as $\mathcal{N}_{m0:m1:m2}$. Whilst the hidden layer has a nonlinear activation function, the output layer is linear. For such an FFNN $\mathcal{N}_{m0:m1:m2}$, the matrix of synaptic weights that connects the inputs nodes to the hidden layer is denoted W_1, and the matrix of synaptic weights that connects the hidden layer to the output layer is denoted W_2. Clearly, $W_1 \in \mathbb{R}^{m_1 \times (m_0+1)}$ and $W_2 \in \mathbb{R}^{m_2 \times (m_1+1)}$, where the bias is also taken into account. The activation function of the hidden layer is $\Phi(v) = a \tanh(bv)$, where a and b are constants and v is the input vector to the hidden layer, and the output layer is a linear layer with unity gain. We denote the input to the network at instant k as x_k and the outputs of the hidden and output layer respectively denoted $y_{1,k}$ and $y_{2,k}$. The desired output at instant k is denoted $y_{d,k}$. Clearly, $x_k \in \mathbb{R}^{m_0}$, $y_{1,k} \in \mathbb{R}^{m_1}$ and $y_{2,k}$, $y_{d,k} \in \mathbb{R}^{m_2}$.

The weight matrix W_1 is randomly initialised. The outputs of each layer can clearly be computed as follows:

$$\begin{aligned}\mathbf{y}_{1,k} &= \phi(W_{1,k}\bar{\mathbf{y}}_{0,k}) \\ \mathbf{y}_{2,k} &= W_{2,k}\bar{\mathbf{y}}_{1,k}\end{aligned} \qquad (1)$$

where

$$\bar{\mathbf{y}}_{0,k} = \begin{pmatrix} 1 \\ \mathbf{x}_k \end{pmatrix}, \quad \bar{\mathbf{y}}_{1,k} = \begin{pmatrix} 1 \\ \mathbf{y}_{1,k} \end{pmatrix},$$

the weight matrix W_2 is updated as follows:

$$P_{k+1} = P_k - \frac{P_k\, y_{1,k+1}\, y_{1,k+1}^T\, P_k}{1 + y_{1,k+1}^T\, P_k\, y_{1,K+1}} \qquad (4)$$

$$W_{2,k+1} = W_{2,k} + \left(y_{d,k+1} - W_{2,k}\, y_{1,k+1}\right) y_{1,k+1}^T\, P_{k+1} \qquad (5)$$

with W_2 and P initialized as follows: $W_{2,0} = 0_{m2 \times (m1+1)}$, and $P_0 = (1/\lambda)\, I$.

Comments: (i) It can be shown that the weights corresponding to the output layer converges to the minimiser of the Tikhonov-Phillips functional [16], [17] $\|W_2 Y_1 - Y_d\| + \lambda\|W_2\|$. The parameter λ is the regularisation parameter that provides a trade-off between the classification error and the strength of the synaptic weights of the output layer. (ii) The weight update equation appears similar to that of the so-called Online Sequential Extreme Learning Algorithm (OSELM) proposed in [18]. It must be emphasised that the initialisations in this paper are very different compared to OSELM. Moreover, OSELM requires a chunk of data to be available a priori making the algorithm not truly sequential. Further, it requires an inversion of a related matrix whose condition number cannot be guaranteed to be low. It may be noted that OSELM is the sequential variant of the extreme learning machine proposed in [19]. (iii) The proposed learning algorithm provided better and satisfactory transient response in the adaptive control of nonlinear systems when compared to other training algorithms including BPA and OSELM [20].

IV. Results

As mentioned earlier, in this paper we consider classification of human activities using online sequential learning without pre-processing and feature extraction. In addition, we consider here a monitoring mechanism that controls the learning mechanism.

We consider here two different partitioning of data: 90:10 and 75:25. That is, in the entire dataset that is available we divide the dataset in the ratio 90:10 for training and testing for the first set of trails and also in the ratio 75:25 for second set of trails. For all experiments, the size of the networks is the same: The number of input nodes is 20 corresponding to the size of the window, the number of hidden neurons is 120 and the number of output neurons is 6 corresponding to the number of classes. As mentioned earlier, the six classes correspond to subjects walking, climbing up the stairs, climbing down the stairs, sitting, standing and lying down. These are respectively denoted C_1, C_2, C_3, C_4, C_5 and C_6.

When the data is partitioned using the 90:10 rule, the number of signals for training is 796 and the number of signals for testing is 60. The classification accuracies for training and testing for a particular trail is shown in Fig. 6 and Fig. 7 respectively. In these figures, the diagonal elements show the number and percentage of correct classification. The off-diagonal elements represent the incorrectly classified samples. Such trials are carried out several times and the average accuracies are considered. The averaged classification accuracies are as shown in Table I. The classification accuracy achieved here is comparable to that obtained in [15] using a multi-class support vector machine, where 96% accuracy was reported. It may be emphasised that the authors in [15] extract first a total of 561 features after the signal has been pre-processed. From the table is clear that class C_4 is the cause for the poorer results.

Similar results are obtained when the data is partitioned 75:25. The classification accuracies are as shown in Table II. On the training data, there is very little difference with a marginal reduction in accuracies with the testing data.

When monitoring is introduced, the computational complexity of the overall process is reduced. Indeed, by setting a classification accuracy threshold to 90%, the number of signals required for training is only 585 in contrast to 796 when there is no monitoring scheme. The achieved average classification accuracy was 88%. Similarly, when the data is partitioned 75:25, only 642 signals were required for training as opposed to 856 signals used in the absence of mechanism for monitoring learning. The proposed mechanism was able to obtain the comparable classification accuracies whilst using only a subset of the dataset.

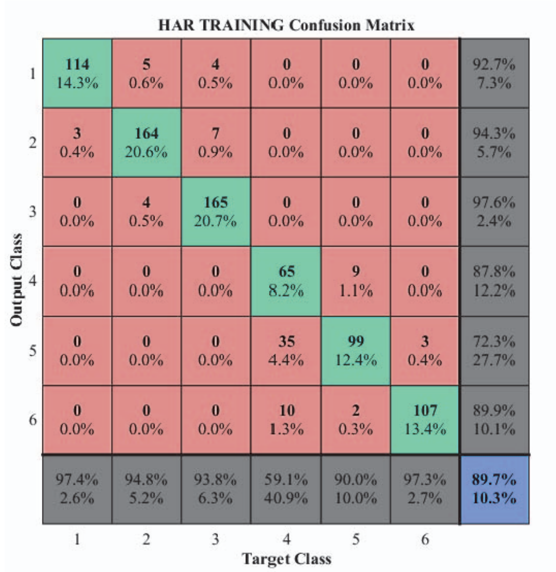

FIG 6. THE CONFUSION MATRIX CORRESPONDING TO TRAINING PHASE

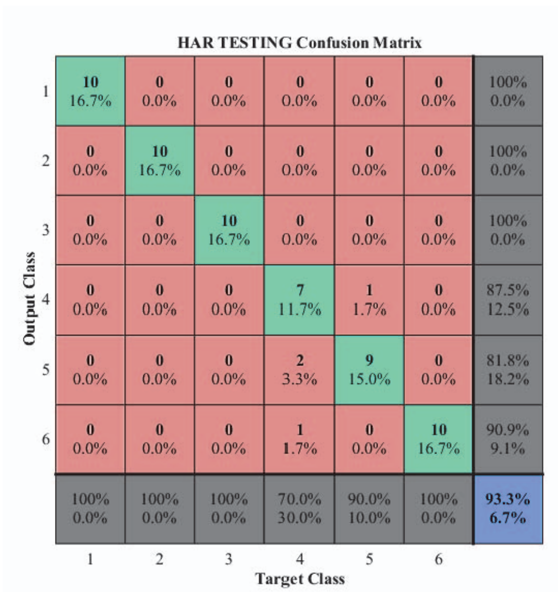

FIG 7. THE CONFUSION MATRIX CORRESPONDING TO TESTING PHASE

TABLE I
AVERAGED CLASSIFICATION ACCURACIES FOR TRAINING AND TESTING WITH 90:10 PARTITIONING OF DATA.

Class	Training Data	Testing Data
C_1	99.33	100.0
C_2	90.83	91.71
C_3	93.61	94.71
C_4	62.14	61.92
C_5	88.26	88.07
C_6	95.99	96.17
Overall	89.11	88.28

TABLE II
AVERAGED CLASSIFICATION ACCURACIES FOR TRAINING AND TESTING WITH 75:25 PARTITIONING OF DATA.

Class	Training Data	Testing Data
C_1	98.95	98.48
C_2	90.69	88.85
C_3	93.76	92.54
C_4	62.81	58.30
C_5	88.78	85.92
C_6	95.88	96.12
Overall	89.18	87.16

Comments: (i) Evidently, learning stops when the classification accuracy has reached an a priori specified threshold. This threshold introduces flexibility in the system. Depending on the application, users can provide different thresholds. (ii) The proposed method merely provides a faster technique for reasonable classification accuracy. Hence, an overall classification methodology can consist of the proposed faster technique to zero in on a class and to subsequently use a different computationally intensive and slower technique to fine-tune. (iii) The implication of a threshold is that the entire dataset is not required for a quick classification. (iv) The method could be used as a means to determine quickly whether or not an a priori specified classification accuracy is achievable with a given dataset. To substantiate the claim, the human activity recognition is performed on the dataset with the same configuration where the average accuracy is achieved using just 585 signals. The number of signals available for training is reduced to 450 signals. The dynamic threshold is set to the average accuracy achieved without the monitoring mechanism. The maximum achieved accuracy in the above scenario was not able to reach the average accuracy over several trials. This implies that the monitoring mechanism was able to understand that the dataset was insufficient to obtain the desired accuracy with available dataset in hand.

V. CONCLUSIONS

A methodology for classification of human activities from acquired signals with a monitoring mechanism for learning is proposed in this paper. A committee of three feedforward neural networks was used to classify six typical human activities. The method is closer to the human cognitive process in that it learns sequentially and provides decisions on-the-fly. The technique does not require pre-processing of signals and avoids feature extraction. The learning is online and sequential using a feedforward neural network with only a single hidden layer. The update equations are straightforward with simple initializations. The achieved classification accuracy is comparable with reduced computational complexity. By introducing a mechanism for monitoring learning, training is stopped when pre-specified classification accuracy is achieved. This avoids excessive learning and achieves a further reduction in the required computations.

References

[1] J. L. Reyes-Ortiz, A. Ghio, D. Anguita, X. Parra, J. Cabestany, and A. Catala, "Human activity and motion disorder recognition: Towards smarter interactive cognitive environments," in *Proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013),* Bruges, Belgium, April 2013, pp. 403–412.

[2] R. Poppe, "Vision-based human motion analysis," *Computer Vision and Image Understanding*, vol. 108, no. 1–2, pp. 4–18, 2007.

[3] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a tri-axial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.

[4] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009),* Miami, FL, USA, June 2009, pp. 1932–1939.

[5] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden markov models," *IEEE Transactions on Image Processing*, vol. 16, pp. 1912–1919, June 2007.

[6] D. Wyatt, M. Philipose, and T. Choudhury, "Unsupervised activity recognition using automatically mined common sense," in *Proceedings of the National Conference on Artificial Intelligence,* vol. 20, October 2005, pp. 21–27.

[7] M. Stikic, K. Van Laerhoven, and B. Schiele, "Exploring semi-supervised and active learning for activity recognition," in *Proceedings of the IEEE International Symposium on Wearable Computers*, October 2008, pp. 81–88.

[8] A. Khan, Y. K. Lee, S. Lee, and T. S. Kim, "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis," in *Proceedings of the 5th International Conference on Future Information Technology*, May 2010, pp. 1–6.

[9] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proceedings of the 4th International Workshop on Ambient Assisted Living (IWAAL 2012),* Vitoria-Gasteiz, Spain, December 2012, pp. 216–223.

[10] M. Ermes, J. Parkka, and L. Cluitmans, "Advancing from offline to online activity recognition with wearable sensors," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, August 2008, pp. 4451–4454.

[11] B. Coley, B. Najafi, A. Paraschiv-Ionescu, and K. Aminian, "Stair climbing detection during daily physical activity using a miniature gyroscope. Gait & posture," *Gait and Posture, Journal of Orthopedic Science*, vol. 22, pp. 287–294, December 2005.

[12] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *Proceedings of the 11th IEEE International Symposium on Wearable Computers*, vol. 22, October 2007, pp. 37–40.

[13] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2009, pp. 2154–2159.

[14] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," *Journal of Universal Computer Science*, vol. 19, pp. 1295–1314, May 2013.

[15] J. L. Reyes-Ortiz, A. Ghio, D. Anguita, X. Parra, J. Cabestany, and A. Catala, "A public domain dataset for human activity recognition using smartphones," in Proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013), Bruges, Belgium, April 2013, pp. 437–442.

[16] D. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *Journal of Association for Computing Machinery*, vol. 9, pp. 84–97, 1962.

[17] A. N. Tikonov, "On solving incorrectly posed problems and methods of regularization," *Doklady Akademii Nauk USSR*, vol.151, pp.501-504, 1963.

[18] N.-Y.Liang, G. -B.Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feed-forward networks," *IEEE Transactions on Neural Networks,* vol. 17, pp. 1411–1423, 2006.

[19] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feed-forward neural networks," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN2004),* Budapest, Hungary, July 2004, pp. 985–990.

[20] K. Subramanian, S. G. Krishnappa, and K. George, "Performance comparison of learning algorithms for system identification and control," in *Proceedings of the 12th IEEE India International Conference (INDICON 2015),* New Delhi, India, December 2015.