University of Reading

Temporal learning using echo state network for human activity recognition

Conference or Workshop Item

Accepted Version

Basterrech, S. and Ojha, V. ORCID: https://orcid.org/0000-0002-9256-1192 (2016) Temporal learning using echo state network for human activity recognition. In: 2016 Third European Network Intelligence Conference (ENIC), 5-7 Sep 2016, Wrocław, Poland, pp. 217-223. doi: https://doi.org/10.1109/ENIC.2016.039 Available at http://centaur.reading.ac.uk/93554/

Temporal Learning using Echo State Network for Human Activity Recognition

Sebastián Basterrech
Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VŠB-Technical University of Ostrava
Ostrava, Czech Republic
Sebastian.Basterrech.Tiscordio@vsb.cz

Varun Kumar Ojha
IT4Innovations
VŠB-Technical University of Ostrava
Ostrava, Czech Republic
varun.kumar.ojha@vsb.cz

***Abstract*—Several works have been applied non-temporal classification techniques in the Human Activity Recognition area. Instead of that, we present an approach for modelling the human activities using a temporal learning tool. Here, the activities are considered as time-dependent events, and we use a temporal learning method for their classification. We employ a well-known learning tool named Echo State Network (ESN). An ESN is a specific type of Recurrent Neural Networks, which has proven well performances for solving benchmark problems with sequential and time-series data. Another advantage is that the method is very robust and fast during the learning algorithm. Therefore, it is a good tool for being applied in real time contexts. We apply the proposed approach for analyzing a well-know benchmark dataset, and we obtain promising results.**

I. Introduction

Human Activity Recognition (HAR) is a relatively new research area that has been gaining relevance during the last decade [1]. The monitoring of human activities can be helpful for understanding the human behaviour. As a consequence, several scientific fields are pouring their interest into the HAR area, which include: sociology, psychology, human-computer interaction, and other field interested in the human behaviour. Nowadays, there is a huge number of smartwatch and mobile phone users, and this number is increasing with the time. Hence, there is a big opportunity to use the captured information through these devices for recognizing the human activities in real-time. Recently, a large public dataset that deals with the problems related to HAR in real-time contexts was introduced to the research communities [2]. The collected dataset has information of several subjects taken from several devices. The authors in [2] analyses the effect of measuring the human activities using different devices by using different recognition techniques. The learning techniques for recognizing activities that have been applied were non-temporal learning methods such as : K-NN, SVM, Random Forest, and C4.5. In the same way, several research works during the last decade have been applied only non-temporal classification tools. We present details about related works in the next section.

Our contribution in this paper consists in modeling and solving a HAR problem using a temporal learning tool, which it differs from the reported approaches in the literature. The human activities evolve in time. Therefore, a natural approach is to classify the output variables that evolve in time by applying classification techniques that use the temporal or sequential information of the dataset. We take account of the spatial information contained into the input features as well as the temporal information among the events. In particular, we used a specific type of *Recurrent Neural Network (RNN)* that was introduced at the beginning of the 2000's under the name of *Echo State Network (ESN)* [3]. In a learning procedure, the ESN tool has advantages for modeling sequential data by overcoming the limitations of RNNs. Another, good characteristic of the ESN is that it has been produced very good results in comparison to other techniques in many practical applications [4]. In addition, we are interested in studying the ESN technique on the HAR area because it has a robust and fast training process. Therefore, it is a very useful tool for learning or modeling datasets on real-time.

This paper is organized as follows. Section II presents a concise survey of the related works introduced during the past years by the research. In section III, we present the formalization of the temporal learning problem and an overview about the ESN model. The benchmark dataset and the results analysis are presented in section IV. We conclude this paper in section V with a precise discussion and an outlook for the future directions.

II. Related works on Human Activity Recognition

The information gathered from smartphones, smartwatches and any other wearable electronics can be used for analysing human behaviour. Data can be "artificially" generated in a laboratory or "naturally" collected data by placing sensors for registering human actions. At one hand, artificially generated data can suffer of over-simplicity and thus, can influence the activities of the subject [5]. On the other hand, data sets gathered from more natural environments suffer from the variations in movement patterns. Several works analysed the process of generation data and its properties [6], [7]. For example, in [6] the authors gathered activities of 20 humans during 20 days in semi-natural conditions. They placed five biaxial accelerometers on five different human body parts. Then, the authors applied machine learning methods for the activity recognition, they used: decision table, instance-based learning

(IBL), C4.5 decision tree, and Naive Bayes classifiers. The best performance was observed by the C4.5 decision tree that reaches a classification rate of 84%. Other similar works that have obtained good performance with C4.5 decision trees can be seen in [8], [9]. In [10], the authors collected data using a single triaxial accelerometer worn on the pelvic region of each subject, and they created a mapping between this data and the activities. Both activity-based computing and context-aware computing suggest the adaptation of user-wearable devices or mobile devices to adapt according to the state of the user activities [11]. In [12], authors applied classification techniques for recognizing the following activities: walking, climbing up, climbing down, standing up, sitting down, and falling, based on the data collected from a variety of accelerometer-based devices. Recent works on activity recognition witnessed classification of the human activities using SVM [13]–[15]. Traditional Neural Networks (NNs) was also applied [16], and in [17] a specific type of NN named Extreme Learning Machines was used.

A data from inertial embedded sensors in smartphones was made in [17]. The authors used an on-line sequential extreme learning machine for human activities recognition and compared their results with the obtained ones when SVM was used [7]. An overview about the implemented classification methods using on-line and off-line training algorithms on data from different platforms, phones, and sensors is available in [18]. In this survey, phones and sensors were used for the following tasks: on-line activity recognition, the resource consumption analysis, the real-time assistive feedback, the validation of on-line activity recognition, the orientation-independent activity recognition, the independent position activity recognition, fixed and adaptive sampling, and dynamic and adaptive sensor selection.

In recent work [19], authors designed a type of ensemble classifier algorithm where the classifiers have hierarchy, the outputs of the higher levels use the outputs of many other classifiers from the lower levels. At each level, different activities were detected, by using base classifiers. Hence, the output of one level passed to the higher level made the detection more specialized in the classification of activities. In [20], authors suggested us to perform feature selection from the extract features in the time domain (absolute mean, median absolute deviation, maximum, minimum, signal magnitude range, interquartile range, and power) and in the frequency domain (maximum, mean, kurtosis, skewness power). Then the authors applied kNN, SVM, kernel-extreme learning machine, and sparse representation classifier for the classification of the activities. Similarly, in [21] was proposed a method that follows speech processing strategies into a human activity recognition and segmentation (HARS) system. The recognition system was based on Hidden Markov Models (HMMs), for recognizing and segmenting six different physical activities: walking, walking upstairs, walking downstairs, sitting, standing and lying. Another helpful survey about HAR can be found in [18], and a very useful review of the data sources and the learning techniques applied on activity recognition can be seen in [2].

III. Supervised sequence labelling with Echo State Networks

This section contains two parts. First, we start by formalizing the mathematical problem studied in this paper. Second, we describe the ESN model and their properties.

A. Formalisation of a temporal learning problem

The goal of a classification problem consists of defining a model for labeling an outcome variable based on a set of input features [22]. Roughly speaking, the model is defined using a particular dataset, with the condition that the generated input-output mapping should be able of "well" mapping any other unknown dataset. A distinctive assumption for solving this task is considering the data points independent of each other. Instead of this, in the case of temporal learning, we can not assume independent patterns because of the both input and output are measured form strongly correlated sequences [23]. Therefore, in the case of temporal learning the model should be able of learning the geometric characteristics of the patterns as well as their serial order. Formally, we refer the distinct points of the input sequence as the time-steps [23]. Let $\mathbf{a}(t)$ be the input pattern collected at time step t, and let $\mathbf{y}(t)$ be its corresponded output. The dimensions of input and output space are given by N_{a} and N_{y}, respectively. The goal of modeling is consists of using a training set S composed by the pairs $(\mathbf{a}, \mathbf{y})$ for defining a parametric learning map $\varphi(\cdot)$ such that is able to label the sequences in a test set S' $(S \cap S' = \emptyset)$ as accurate as possible. For measuring the accuracy of the mapping, a distance in an arbitrary time range $[1, T]$ is used. In this paper, we use the following well-known quadratic distance:

$$E = \frac{1}{T} \sum_{t=1}^{T} \sum_{o=1}^{N_{\mathrm{y}}} (\hat{y_o}(t) - y_o(t))^2, \qquad (1)$$

where $\hat{y_o}(t)$ is the coordinate o of the model output at time t. Although, in our experimental part the output space is unidimensional ($N_{\mathrm{y}} = 1$).

B. Background on Echo State Networks

Recurrent Neural Networks (RNNs) are a type of Neural Networks (NNs) particularly suitable for solving benchmark problems with temporal and sequential data. They are bio-inspired distributed systems, where the nodes are simple processors (most often sigmoid functions) that transform and send information among them in a topology of graph (network-like structure). The graph has a weighted topology, and for the specific case of RNN at least one circuit is presented. This cyclical characteristic on the graph makes the RNN a special tool for processing sequential data. Due to the circuit, the model has the ability of memorize the information. However, the RNN has a limited popularity in the community, probably this is caused because is difficult to figure out an optimal topology [23]. More specifically, several optimization techniques that work well on NNs without recurrences can fail in the case of RNNs [24].

At the beginning of the 2000s, a new approach for designing and training a RNN was introduced in the community under the names of *Echo State Network (ESN)* [3] and *Liquid State Machine (LSM)* [25]. Both models have been developed in parallel and they are computationally equivalent. The first one was introduced by a research team of the Machine Learning area, and the second one comes from researchers on the area of Neurocomputing. The model consists of two well-differentiated structures. One is a RNN used for processing and memorizing the input sequence. Another one is a simple linear regression for generating the model outputs. Often, the RNN structure is randomly initialized by following some algebraic properties that we discuss below. Another characteristic of RNN is that its weights are fixed during the learning process. The role of the recurrent part, named *reservoir* [26], is to memories the serial order of the inputs and to expand the input space into a larger dimensional space (projection space). In the phase of linear regression, the points from the projected space are mapped into the output space. Only the weights of the linear regression are adjusted during the training algorithm. As a consequence, the learning is fast and robust. In addition, the model has proven to obtain very well performances in many real applications [4], [26].

The model is defined as follows. We follow previous mentioned notations where N_{a}, N_{x} and N_{y} are the number of input, hidden and output neurons, respectively. Let $\mathbf{w}^{\mathrm{in}}$ be a $N_{\mathrm{x}} \times N_{\mathrm{a}}$ matrix that collects the input-reservoir weights. Similarly, let $\mathbf{w}^{\mathrm{r}}$ be a $N_{\mathrm{x}} \times N_{\mathrm{x}}$ matrix that collects hidden-hidden weights and let $\mathbf{w}^{\mathrm{out}}$ be a $N_{\mathrm{y}} \times N_{\mathrm{x}}$ matrix with the projected space to the output space. The RNN on the reservoir is characterized at each time by the following high-dimensional state:

$$\mathbf{x}(t) = f_1(\mathbf{w}^{\mathrm{in}}\mathbf{a}(t) + \mathbf{w}^{\mathrm{r}}\mathbf{x}(t-1)). \tag{2}$$

The output of the model is given by:

$$\hat{\mathbf{y}}(t) = f_2(\mathbf{w}^{\mathrm{out}}\mathbf{x}(t)), \tag{3}$$

The functions $f_1(\cdot)$ and $f_2(\cdot)$ are two predefined coordinate-wise functions, most often they are a $\tanh(\cdot)$ function.

In this paper, we use a slight variation of the canonical ESN method that computes the reservoir state as follows:

$$\mathbf{x}(t) = (1-\alpha)\mathbf{x}'(t) + \alpha\mathbf{x}(t-1), \tag{4}$$

where the parameter $\alpha \in [0, 1)$ is called *leaky rate* and is used for controlling the reservoir state update. In addition, the regression also contains the input patterns. Then, the matrix $\mathbf{w}^{\mathrm{out}}$ is a $N_{\mathrm{y}} \times N_{\mathrm{a}} + N_{\mathrm{x}}$ dimensional matrix with the projected space to the output space. For the sake of notation simplicity, we omit the bias term (it is included in the weight matrices).

Due to the fact that the reservoir is fixed during the learning, the ESN initialization plays a very relevant role in the model. Basically, the ESN initialization must satisfy the following conditions [4], [27]. The size of the reservoir should be much larger than the dimensionality of the input space. This parameter impacts on the linear separability of the input data. Another relevant parameter is the input scaling factor what weights the input patterns, in our case we normalise the inputs, so all the inputs have equal relevance. The spectral radius of the reservoir matrix controls the stability and the memory capability of the RNN [26], [27]. Several techniques for setting the initial parameters have been introduced, for instance see [28], [29]. In order of having stable dynamics the spectral radius (we denote it by $\mathbf{w}^{\mathrm{r}}$) should satisfy $\rho(\mathbf{w}^{\mathrm{r}}) < 1$. Although, the stability can even happen when $\rho(\mathbf{w}^{\mathrm{r}}) >= 1$ [4]. Another parameter is the sparsity of the reservoir connections. We define reservoir weights with 20% non-zero values.

IV. Experimental Results

Here, we present the benchmark data used for This section starts with a description of the benchmark data. Next, we present our experimental results.

A. Benchmark dataset description

In this paper, we use a subset of the large dataset presented in [2] where, the authors were interested in figuring out the impact on the human activity recognition produced by data that is measured by several heterogeneous sensors. In our work, we are interested in evaluating the ability of temporal learning for modeling the human activity. In this way, we analyse a dataset collected with an embedded Gyroscope sensor, that is sampled at the highest frequency the respective device allows [2]. The tools used for collected the data are two LG smartwatches and two Samsung Galaxy Gears smartwatches. There are three subjects realizing the following activities (the output variable to be modeled): Biking, Sitting, Standing, Walking, Stair Up and Stair down, which are encoded by $0, 1/6, 1/3, \ldots, 1$, respectively. The input information are the coordinates x, y, z, the subject, the watch model, and the device. For more details about the collection of the data, see [2]. Figures 1, 2 and 3 show the activities of the subject 1, 2 and 3, respectively. From those graphics, we can see the different behavior of the subjects.

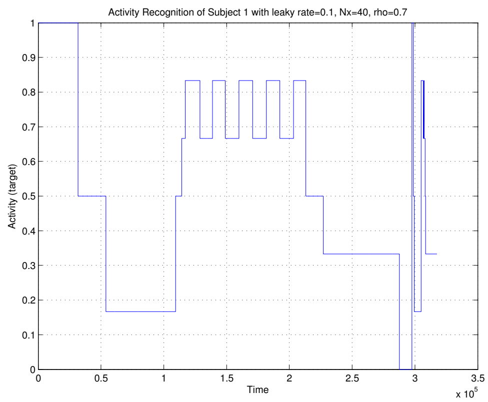

Figure 1: Activity of the subject 1 on the testing dataset.

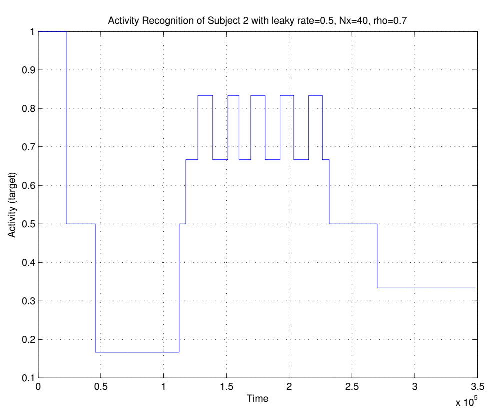

Figure 2: Activity of the subject 2 on the testing dataset.

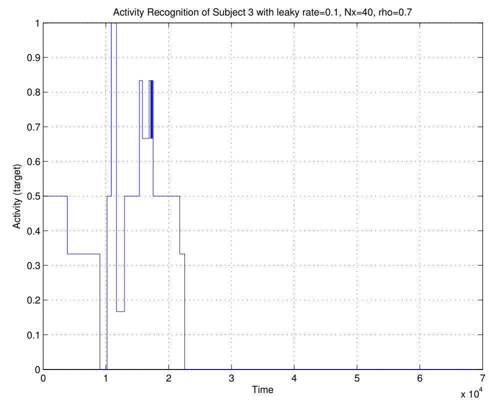

Figure 3: Activity of the subject 3 on the testing dataset.

In Figures 1, 2 and 3, the x-axis indicates time (time-step at which the activities were recorded), and the y-axis indicates the activity performed by the subject (e.g., Biking, walking, Standing, etc.). Our objective is to predict/classify these activities using the ESN and taking account of the time-sets of the activities.

B. Analysis of the results

The first step was setting up the global parameters of the ESN. Therefore, we evaluated the performance of the ESN for the following parameter stings: reservoir size N_{x} in $\{30, 40, 50, \ldots, 120\}$, spectral radius $\rho(\mathbf{w}^{\mathrm{r}})$ in $\{0.1, 0.2, \ldots, 0.9, 0.99\}$ and leaky rate α in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Then, we divide the dataset in two subsets, the first subset (80%) was used for the training of the model and the second subset (20%) was used for the testing of the model. We used off-line ridge regression for computing the output weights [30]. For the illustration of the impact of this parameter setting on the ESN model accuracy, we studied there logarithmic scale map (Figures 4, 5 and 6). Each of one is for different subject and for different leaky rate. Figure 4 shows the accuracy of the model (in logarithmic scale) according to the reservoir size and spectral radius when the ESN has leaky rate equal to 0.1. Figure 5 presents the accuracy (in logarithmic scale) when the leaky rate was 0.5, and Figure6 shows the logarithmic quadratic error for subject 3 and leaky rate 0.9. In Figures 4, 5 and 6, we observed the impact of the parameter N_{x} on the trade-offs between the training and the testing. It was found that the larger N_{x} values reduces the

Table I: Mean of the quadratic errors of the 30 experiment with different initialized ESN. The parameters were $N_{\mathrm{x}}^{*}=40$, $\rho^{*}=0.7$.

Leaky	Subject 1	Subject 2	Subject 3
0.1	0.238×10^{-4}	0.104×10^{-4}	0.012×10^{-1}
0.5	0.255×10^{-4}	0.095×10^{-4}	0.022×10^{-1}
0.9	0.331×10^{-4}	0.013×10^{-3}	0.035×10^{-1}

Table II: Mean of the standard deviations of the 30 experiment with different initialised ESN. The parameters were $N_{\mathrm{x}}^{*}=40$, $\rho^{*}=0.7$.

Leaky	Subject 1	Subject 2	Subject 3
0.1	0.132×10^{-4}	0.696×10^{-4}	0.010×10^{-1}
0.5	0.142×10^{-4}	0.749×10^{-4}	0.023×10^{-1}
0.9	0.192×10^{-4}	0.103×10^{-3}	0.030×10^{-1}

training error, but at the same time increase the testing error. In all the cases, the best values corresponds to $N_{\mathrm{x}}<50$. On the other hand, the impact of the spectral radius is not very clear although it seems that the best situations are produced when the parameter ρ belongs to $[0.5,0.8]$ Once we define the best ESN global parameters that we denote by N_{x}^{*}, ρ^{*}, and α^{*}, we run 30 random initialized ESN and their performances were evaluated. According to the obtained results, we consider that the best parameters are: $N_{\mathrm{x}}^{*}=40$ and $\rho^{*}=0.7$.

Table I presents the mean accuracy computed using (1) among the 30 experiments, with an ESN defined with $(N_{\mathrm{x}}^{*},\rho^{*})$ parameters according to the subjects and the leaky rate. A similar information is shown in Table II, in this case, the standard deviation among the 30 trials are presented. From Tables I and II, it is possible to statistically compare (using confidence intervals with Normal approximations) the impact of the leaky rate on the model accuracy. In addition, from the information of Table I, we can see that the best performance was produced by the temporal learning modeling. As an example, in Figure 7, we present the target and its prediction given by an ESN with the following parameter setting: $\alpha^{*}=0.5$, $N_{\mathrm{x}}^{*}=40$, $\rho^{*}=07$. In Figure 7, the red curve is the target and the blue one corresponds to the estimation. We can see that the estimation is very well, there is sometime a small delay when the person change of activity. From Table I, it is possible to see that the modeling accuracy for the subject 3 is less than the modeling accuracy for the other two subject. The possible cause for such difference in accuracy is because of the different length of the training set, each subject has different training set (subject 1 has 136083 points, subject 2 has 149204 points and subject 3 has 29285 points). The differences of the sizes were given by the original data set that has different size of data according the persons. Another cause of such difference, can be produced by the fact of that each subject has different behavior. Therefore, the training data can be very different from each other.

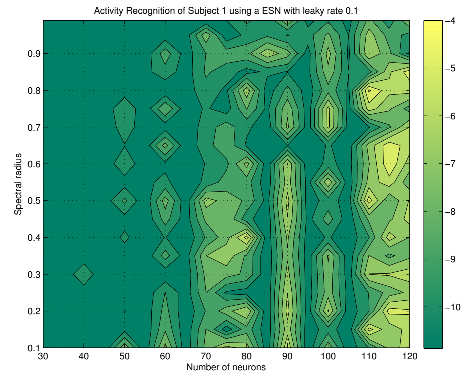

Figure 4: Model accuracy in logarithmic scale. The classification was for subject 1 with a ESN with leaky rate α equal to 0.1.

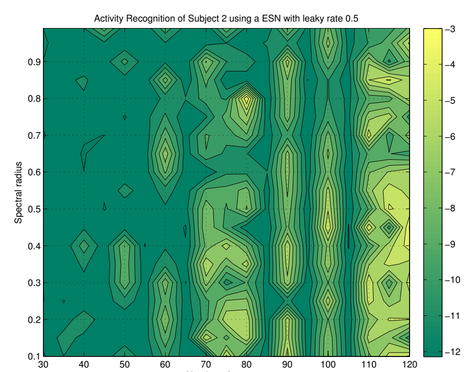

Figure 5: Model accuracy in logarithmic scale. The classification was for subject 2 with a ESN with leaky rate α equal to 0.5.

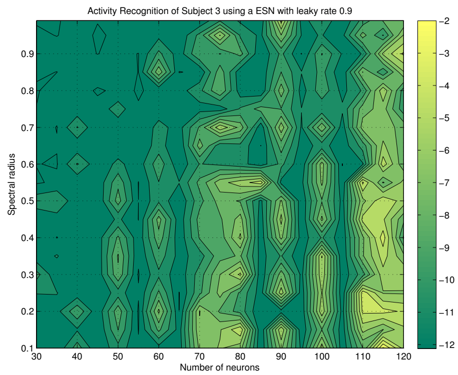

Figure 6: Model accuracy in logarithmic scale. The classification was for subject 3 with a ESN with leaky rate α equal to 0.9.

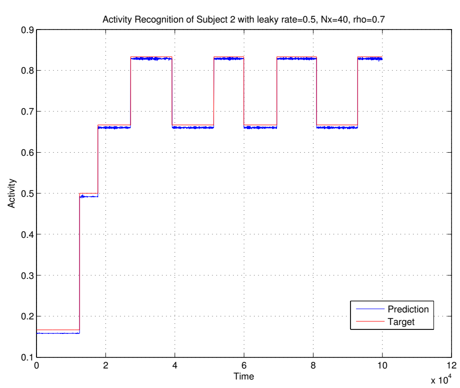

Figure 7: Example of the accuracy of the model. Activity of the subject 2 and its prediction given by an ESN with $\alpha^* = 0.5$, $N_{\mathrm{x}}^* = 40$, $\rho^* = 07$.

V. Conclusions and Future Work

We present an approach for Human Activity Recognition (HAR) that models the problem using temporal learning. In related works, most often the classifiers were built as non-temporal supervised learning techniques. The human activities evolve in time. Here, we consider the problem as a temporal supervised learning. Our hypothesis is that the activity presented at current time impacts in the next ones. We use a well-known learning tool for modelling temporal learning named Echo State Networks. The method has presented well-performances in several real-world problems. Another good property of the model is that is very robust and fast during the training, therefore it can be a good tool for real time contexts. Then, the model can be adapted in real-time devices producing new most appropriate activity estimations. The ESN has several parameters, we presented a sensitivity analysis of its parameters. We figure out that for this specific dataset, the model is very sensitive to the size of the hidden-hidden weights. On the other hand, the impact of the leaky rate and the spectral radius of the hidden-hidden matrix seem to have less relevance. The proposed method a very good accuracy, in the near future we are interested in analysing the problem with other type of biaxial or triaxial accelerometers, as well as with other supervised learning techniques.

Acknowledgement

This work was supported by Grant of SGS No. SP2016/146, VŠB-Technical University of Ostrava, Ostrava, Czech Republic.

References

[1] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1192–1209, 2013.

[2] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjaergaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition," in *In Proc. 13th ACM Conference on Embedded Networked Sensor Systems (SenSys 2015)*, Seoul, Korea, 2015.

[3] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," German National Research Center for Information Technology, Tech. Rep. 148, 2001.

[4] M. Lukoševičius and H. Jaeger, "Reservoir Computing Approaches to Recurrent Neural Network Training," *Computer Science Review*, vol. 3, pp. 127–149, 2009.

[5] K. Aminian, P. Robert, E. Jequier, and Y. Schutz, "Estimation of speed and incline of walking using neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 44, no. 3, pp. 743–746, 1995.

[6] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive computing*. Springer, 2004, pp. 1–17.

[7] J. L. R. Ortiz, "Human activity dataset generation," in *Smartphone-Based Human Activity Recognition*. Springer, 2015, pp. 59–78.

[8] M. Mathie, B. G. Celler, N. H. Lovell, and A. Coster, "Classification of basic daily movements using a triaxial accelerometer," *Medical and Biological Engineering and Computing*, vol. 42, no. 5, pp. 679–687, 2004.

[9] 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 triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.

[10] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *AAAI*, vol. 5, 2005, pp. 1541–1546.

[11] N. Davies, D. P. Siewiorek, and R. Sukthankar, "Activity-based computing," *IEEE Pervasive Computing*, no. 2, pp. 20–21, 2008.

[12] T. Brezmes, J.-L. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living*. Springer, 2009, pp. 796–799.

[13] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[14] 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 *Ambient assisted living and home care*. Springer, 2012, pp. 216–223.

[15] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[16] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern recognition letters*, vol. 29, no. 16, pp. 2213–2220, 2008.

[17] Y. Al Jeroudi, M. Ali, M. Latief, and R. Akmeliawati, "Online sequential extreme learning machine algorithm based human activity recognition using inertial data," in *Control Conference (ASCC), 2015 10th Asian*. IEEE, 2015, pp. 1–6.

[18] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.

[19] G. Filios, S. Nikoletseas, C. Pavlopoulou, M. Rapti, and S. Ziegler, "Hierarchical algorithm for daily activity recognition via smartphone sensors," in *IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015*. IEEE, 2015, pp. 381–386.

[20] Q. Zhu, Z. Chen, and Y. C. Soh, "Smartphone-based human activity recognition in buildings using locality-constrained linear coding," in *IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), 2015*. IEEE, 2015, pp. 214–219.

[21] R. San-Segundo, J. M. Montero, R. Barra-Chicote, F. Fernández, and J. M. Pardo, "Feature extraction from smartphone inertial signals for human activity segmentation," *Signal Processing*, vol. 120, pp. 359–372, 2016.

[22] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of Statistical Learning*, ser. Spring series in statistics. New York, USA: Springer-Verlag, 2001.

[23] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, 1st ed., ser. Studies in Computational Intelligence. Springer-Verlag Berlin Heidelberg, 2012, vol. 385.

[24] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

[25] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: a new framework for a neural computation based on perturbations," *Neural Computation*, vol. 14, pp. 2531–2560, november 2002.

[26] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Networks*, vol. 20, no. 3, pp. 287–289, 2007.

[27] J. B. Butcher, D. Verstraeten, B. Schrauwen, C. R. Day, and P. W. Haycock, "Reservoir Computing and Extreme Learning Machines for Non-linear Time-series Data Analysis," *Neural Networks*, vol. 38, pp. 76–89, feb 2013.

[28] S. Basterrech, C. Fyfe, and G. Rubino, "Self-organizing Maps and Scale-invariant Maps in Echo State Networks," in *11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Córdoba, Spain, November 22-24, 2011*, November 2011, pp. 94–99.

[29] ——, "Initializing Echo State Networks with topographic maps," *Proceeding in the Second International Conference on Morphological Computation (ICMC 2011)*, pp. 103–105, September 2011.

[30] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, Feb. 2002.