Inertia Based Recognition of Daily Activities with ANNs and Spectrotemporal Features

Ozsel Kilinc, Alexander Dalzell, Ismail Uluturk and Ismail Uysal
Electrical Engineering
RFID Lab for Applied Research
University of South Florida
Tampa, Florida, United States
ozsel@mail.usf.edu, adalzell@mail.usf.edu, uluturki@mail.usf.edu, iuysal@usf.edu

***Abstract*—As mobile and personal health devices gain in popularity, increasing amounts of data is collected via their embedded sensors such as heart rate monitors and accelerometers. Data analytics and more specifically machine learning algorithms can transform this data into actionable information to improve personal healthcare and quality of life. The main objective of this study is to develop an algorithmic classification framework using feed-forward multilayer perceptrons and statistically rich spectrotemporal features to recognize daily activities based on 3-axis acceleration data. A multitude of MLP topologies and setups, such as different numbers and sizes of hidden layers, supervised output structuring, etc. are tested to comprehensively analyze the clustering capabilities of the artificial neural network for a wide-range of settings. In addition, the contribution of subset of features to classification accuracy is studied to identify respective information potentials and further improve accuracy. Publicly available wrist-worn accelerometer dataset from University of California Irvine's machine learning repository is used for fair comparison with the most recent literature published using the same dataset. Results indicate a significant improvement in recognition rate where the overall accuracy over seven selected activity classes is 91% compared to 54% of the latest publication using the same dataset.**

Keywords—machine learning, accelerometer, inertia, classification, recognition, daily activity, health

I. Introduction

Healthcare is, without a doubt, one of the most important application areas of big data analytics, only poised to become more significant as more capable and personal data collection devices gain in popularity [1]. Groves et. al. suggest that the big data revolution is already in full force in healthcare and mobile health applications are one of its biggest sources of innovation [2]. According to a recent study by Allied Market Research, the mobile health market will expand at a compound annual growth rate of 33.5% between 2015 and 2020 [3]. A special report by the IEEE Spectrum magazine talks about how big data analytics are shaping human health in the 21st century [4]. As mobile and personal devices become more capable front-end tools to collect personalized data, machine learning algorithms are becoming more complex to accommodate pursuant analysis, classification and prediction tasks [5].

The objective of this paper is to analyze inertia data obtained via personal wrist-worn accelerometer sensors to develop an automatic classification algorithm to identify and recognize several unique classes of daily activities, such as walking, getting up from a bed, eating, climbing stairs, etc. Researchers have investigated this problem in the recent past using a wide variety of signal processing algorithms for feature extraction and classifiers ranging from nearest class centroids [6] to Gaussian mixture models [7] as well as looking at different combination of sensors to improve accuracy [8]. Some groups looked at applying artificial neural networks (ANN) as more sophisticated non-linear classifiers but only to recognize four distinct activity classes with non-publicly available datasets [9].

In this paper, we use a publicly available wrist-worn accelerometer dataset from University of California Irvine's machine learning repository, originally developed to support the research study of Bruno et. al. [7]. With the help of signal processing algorithms to obtain statistically rich spectrotemporal features from the raw acceleration data, we use a multitude of multilayer perceptron (MLP) topologies to fully analyze the classification capabilities of ANNs for this particular application. Furthermore, different combinations and subsets of features are fed into the network to measure their respective information potentials. A significant performance increase of approximately 70% is observed when compared to the latest publication to study the same dataset [7]. Finally, the results as well as complete training/testing procedures are presented in detail to allow meaningful comparisons for future competitive studies.

This paper is organized as follows: next section focuses on the extraction of features from the time-domain signal and evaluation of these features according to their effects on accuracy. In the third chapter, selection of hidden layer architecture is stated and a modification to the output layer is offered for further accuracy enhancement. Results are compared in the fourth chapter, followed by conclusions with final remarks and evaluation.

978-1-5090-0287-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICMLA.2015.220

II. Feature Selection

A. *Feature Extraction from Time-Domain Signal*

Accelerometers are electromechanical devices used for measuring proper acceleration. Since multi-axis accelerometers can be used to sense orientation, change in g-force per axis, vibration and shock, they are one of the essential sensors used in smart phones and other wearable technology devices on the market. By decent evaluation of the data obtained via accelerometers, systems may be able to recognize more complex motions besides sensing screen orientation and counting number of steps.

Fig. 1 shows an example of single-axis acceleration data obtained during the execution of 7 different human motion primitives. One way to introduce this data into an artificial neural network (ANN) is to assume each time-domain sample as a feature and create a network for which the input layer size equals the number of sampling points. This method forms a relatively larger ANN which needs bigger datasets for training and it might be more susceptible to noise. Another way is to extract differentiating features from whole time-domain signal and create an ANN with 10 – 20 inputs that can be adequately using a modest dataset.

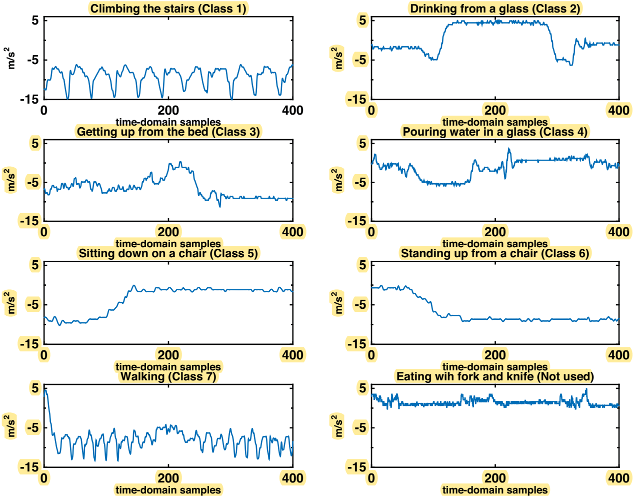

Fig. 1. An example of single-axis acceleration data for 7 different motion primitives

Mean and variance of a time-domain signal are two conventional and inexpensive features to examine the general attitude of the motion [10]. Values of these two features for all 700 time-domain recordings in the dataset are shown in Fig. 2 and Fig. 3. Since they are clustered according to motion class, we can make a quick evaluation of the distinguishing characteristics of these features. For example, it is easy to say that Class 2 and Class 4 can be clearly differentiated from other classes by using mean. By interpretation of variance, Class 1 and Class 7 can be diverged from Class 3, Class 5 and Class 6.

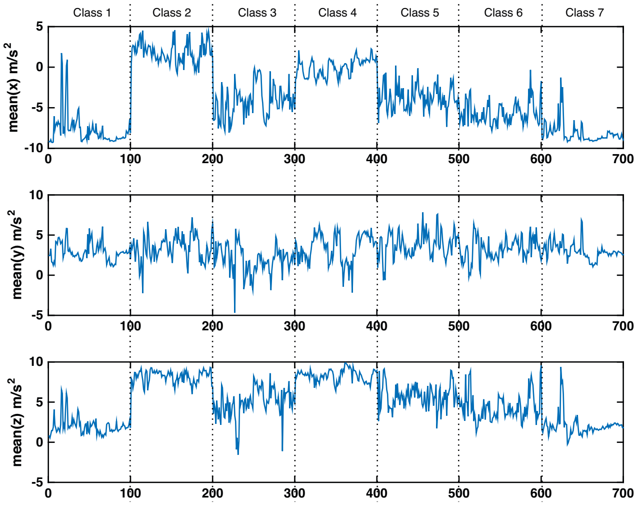

Fig. 2. Mean of three-axis acceleration data for whole dataset

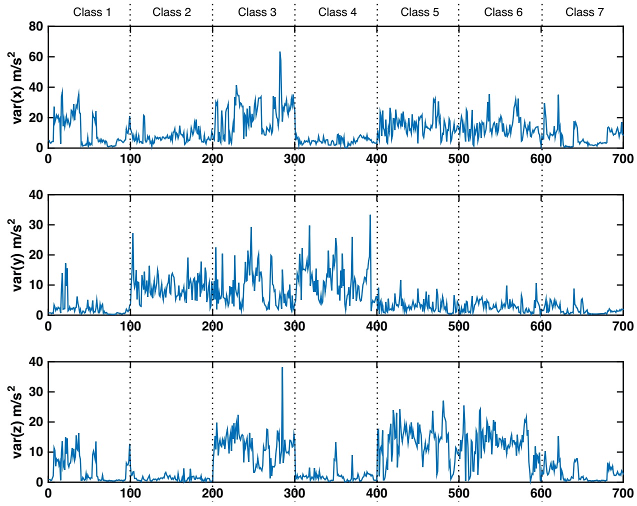

Fig. 3. Variance of three-axis acceleration data for whole dataset

In order to evaluate the orientation change during motion we can also calculate pitch, roll and yaw features which we define as in equations (1), (2) and (3) where *N* is the number of time-domain samples. The values of these features are given in Fig. 4. These features seem to be useful for separation of Class 5 from Class 3 and Class 6, which is unlikely to be managable using only mean and variance features.

$$pitch = \arctan\left(\frac{x_N}{\sqrt{y_N^2 + z_N^2}}\right) - \arctan\left(\frac{x_1}{\sqrt{y_1^2 + z_1^2}}\right) \quad (1)$$

$$roll = \arctan\left(\frac{y_N}{\sqrt{x_N^2 + z_N^2}}\right) - \arctan\left(\frac{y_1}{\sqrt{x_1^2 + z_1^2}}\right) \quad (2)$$

$$yaw = \arctan\left(\frac{z_N}{\sqrt{x_N^2 + y_N^2}}\right) - \arctan\left(\frac{z_1}{\sqrt{x_1^2 + y_1^2}}\right) \quad (3)$$

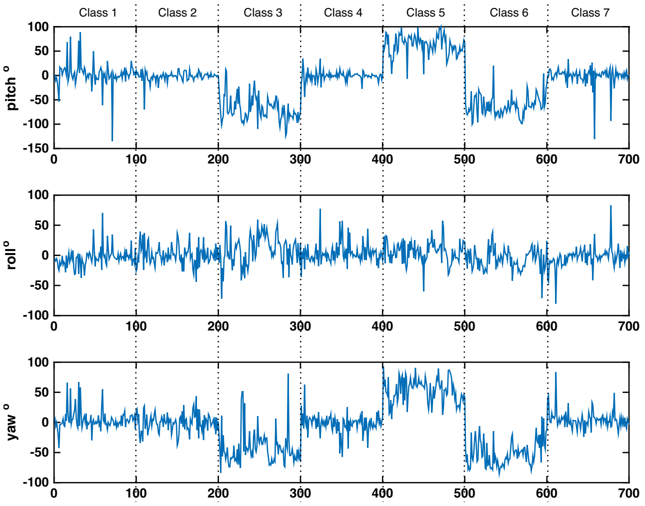

Fig. 4. Pitch, roll and yaw features for whole dataset

For all features we have already mentioned, while low frequency characteristics are preserved, high frequency characteristics are discarded. In order to regard these characteristics in an effective and inexpensive way, number of mean-crossings of time-domain-signals can be used as another feature [11]. Furthermore, first mean crossing index and longest period between mean crossings are robust features as well for interpreting high frequency characteristics. In Fig. 5, extraction of these features is shown on an example from Class 2. The values of these features calculated for x-axis is given in Fig. 6 for the whole dataset.

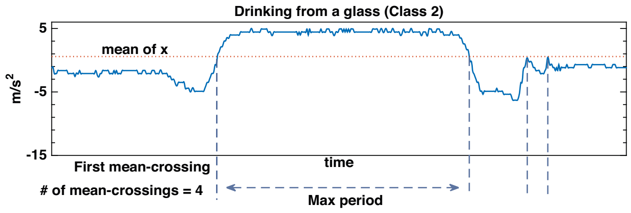

Fig. 5. Extraction of mean-crossing related features

Generally for all features, x-axis data constitute more distinct information about the motion. This is due to the orientation of accelerometers on wristband. In the natural standing position of human body, x-axis of these wristbands corresponds to the direction of gravitational force. Mean-crossing related features can also be observed on y-axis or z-axis. Some of them coincide with the ones observed on x-axis, but some of them are observed only on y-axis or z-axis and these features are too significant to be discarded. For the sake of keeping input size minimal, instead of adding 3 more features for both axes, we calculate the resultant of all 3 axes by using (4) and extract these features for *r*.

$$r = \sqrt{x^2 + y^2 + z^2} \qquad (4)$$

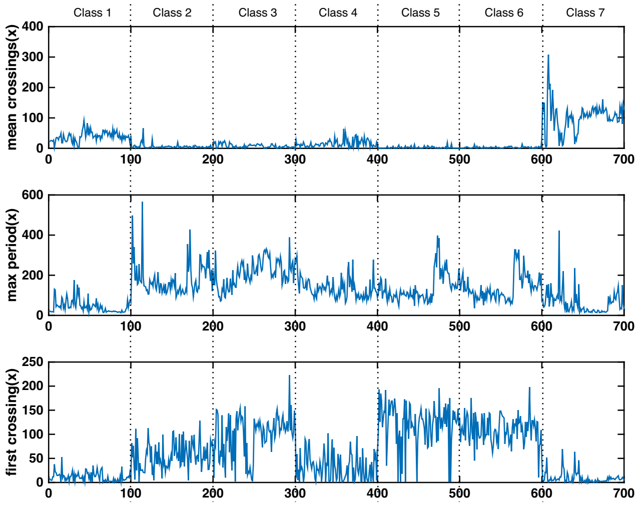

Fig. 6. Mean-crossing related features for whole dataset

B. Evaluation of Selected Features

We evaluate the features that we have extracted from time-domain acceleration signals using ANN with 2 hidden layers, each of which consists of 15 nodes as shown in Fig. 7. We use Scaled Conjugate Gradient Backpropagation method for training [12]. Results are shown in Table I where average accuracy is calculated on testing set, as the sum of correct classifications divided by the total number of classification.

Using only mean and variance of 3-axis data does not provide a decent classification performance. Introduction of pitch, roll and yaw (PRY) features improves accuracy dramatically. However, the most remarkable boosting is observed when mean-crossing related (MCR) features of x-axis are taken into account. Furthermore, by adding *r* to these features classification accuracy can be slightly enhanced.

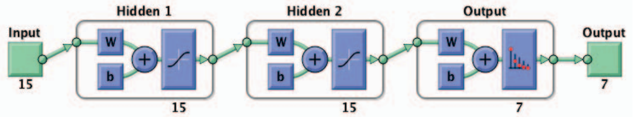

Fig. 7. Neural network used for feature evaluation

Table I. Effect of Selected Features on Overall Accuracy

Used Features	Accuracy %
Mean(x,y,z), Variance(x,y,z)	60.5
Mean(x,y,z), Variance(x,y,z), PRY	66.9
Mean(x,y,z), Variance(x,y,z), PRY, MCR(x)	81
Mean(x,y,z), Variance(x,y,z), PRY, MCR(x), MCR(r)	82.6

III. Further Accuracy Enhancement Via Modifying Network Size

A. Effect of ANN Size to Overall Accuracy

In order to observe the effect of ANN size on accuracy we perform two different experiments related to ANN size. First experiment focuses on the effect of number of hidden layers and second one focuses on the effect of number of nodes in each hidden layer.

All 15 features are used for both experiments. For the first experiment, 12 different ANNs with various number of hidden layers (1,2 and 3) and various number of nodes per hidden layer (5, 10, 15 and 20) are tested. As a result shown in Fig. 8, regardless of the number of nodes per hidden layer, ANNs having less hidden layers provide higher accuracy for our case. A single hidden layer ANN seems to be the best solution to use with 15 features that we have extracted from the dataset. Via Fig. 8, we can also observe that the increasing number of nodes per hidden layer increases the accuracy. In order to find out the limit of this increase, for a single hidden layer ANN we conducted a second experiment in which the number of nodes is varied from 5 to 200 in 12 steps. As we expected, increasing number of nodes in the hidden layer enhances the accuracy only up to a maximum point. As shown in Fig. 9, maximum accuracy (84.9%) is achieved with 60 nodes and it is settled approximately to this value for the next 2 steps. However, further increments beyond 100 nodes cause a drop of overall accuracy which might be due to the limitation of the size of the training dataset. As a result of these two experiments, we could determine the ideal size of the hidden layer of ANN trained using these 15 features to maximize the accuracy for our problem using the dataset provided by [7].

The first step was to tune up ANN parameters with feature selection affecting the input layer size. This was followed by evaluating several hidden layer sizes to improve accuracy. Before moving on to the comparison of our results with the ones found in [7], we will make a final attempt for further enhancement by actually modifying the output layer which will result in readjustment of the size of the hidden layer.

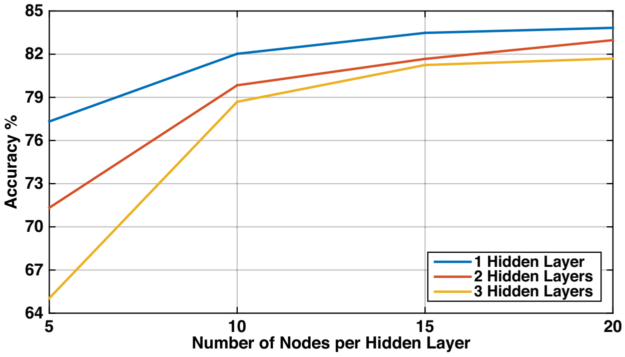

Fig. 8. Effect of Number of Hidden Layers on Overall Accuracy

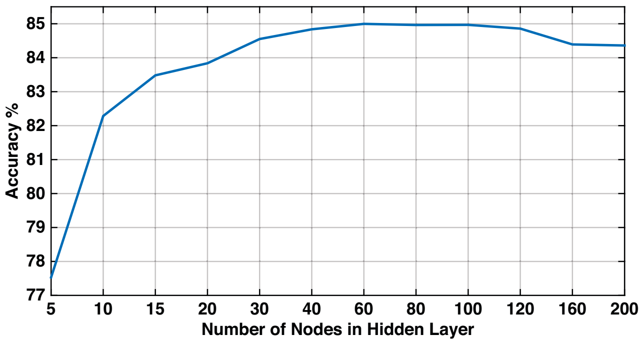

Fig. 9. Effect of Number of Nodes on Overall Accuracy

B. Modifying Output Layer

If we examine the extracted features given in Fig. 2, Fig. 3, Fig. 4 and Fig. 6, we can notice that it is almost impossible to constitute a single cluster for each class. Even if the average values of features are quite differentiated for classes, there are such examples that the value of a feature is much closer to another class average than its actual class average. This situation requires treating classes as formed by multi clusters of features. In fact, the hidden layer automatically carries out this task which is the reason behind improving accuracy with increasing number of hidden layer nodes. Besides this self-riding behavior of ANN we now force the network explicitly to form these clusters via modifying the output layer. In order to achieve this, we artificially increase the number of classes. Since number of nodes in output layer is directly equal to the number of classes, this modification increase the number of nodes in output layer.

Until now, we have used input-output pairs such as $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$ …… $(x^{(m)}, y^{(m)})$ where $x^{(i)} \in \mathfrak{R}^{15}$ and $y^{(i)} \in \mathfrak{R}^{7}$ in such a neural network that

$$h_{W,b}^{(i)} = net(W,b,x^{(i)}) \tag{5}$$

where *W, b* are fixed setting parameters of *net* after training and $h_{W,b}{}^{(i)}$ is hypothesis defined by *net* such that $h_{W,b}{}^{(i)} \in \mathfrak{R}^{7}$. In order to manipulate *net* to form multi-clustered classes we will use transformation matrix $T \in \mathfrak{R}^{7k \times 7}$ defined as

$$T = \begin{bmatrix} I_7 \\ I_7 \\ . \\ . \\ I_7 \end{bmatrix} \tag{6}$$

and used to create $\hat{y}^{(i)} \in \mathfrak{R}^{7k}$ such that

$$\hat{y}^{(i)} = Ty^{(i)} \tag{7}$$

This modification leads to a new hypothesis such as $\hat{h}_{W,b}^{(i)} \in \mathfrak{R}^{7k}$. We will use the transpose of T to reconstruct $h_{W,b}^{(i)}$ such that

$$h_{W,b}^{(i)} = T^{\mathrm{T}} \hat{h}_{W,b}^{(i)} \tag{8}$$

in order to compare it with $y^{(i)}$ and calculate accuracy. In other words, after this modification there are k clusters of each class. These clusters are merged to obtain the final result of classification.

In order to see the effect of output layer modification, we have tried three cases where k is equal to 2, 4 and 8. For all these cases we have also modified the number of nodes in hidden layer. We have added 3 more steps to previous 12 step modification and kept increasing hidden layer size until 400 nodes. For being a reference point, we have also added no modification case (k is equal to 1) shown in Fig. 9 to the comparison given in Fig. 10.

Before modification, best accuracy was achieved with 60 nodes hidden layer as 84.9%. This accuracy is outperformed by all three scenarios where $k > 1$ with same hidden layer size. Moreover, while accuracy was settling at this point before modification, now settling point moves to the right side of this point proportional to the increase of output layer size, where, further increments on hidden layer size start to improve accuracy. As a result, with this modification we have succeeded to enhance accuracy by incrementing both hidden layer size and output layer size. Best accuracy is obtained with 240 nodes in the hidden layer and 56 nodes in the output layer (which actually corresponds to 7 additional clusters for each of the 7 classes, 7 x 8 = 56) architecture as 86.3%. Output layer size may be extended over 56, however we expect progressively less increases in accuracy than the one obtained when it is increased from 28 to 56.

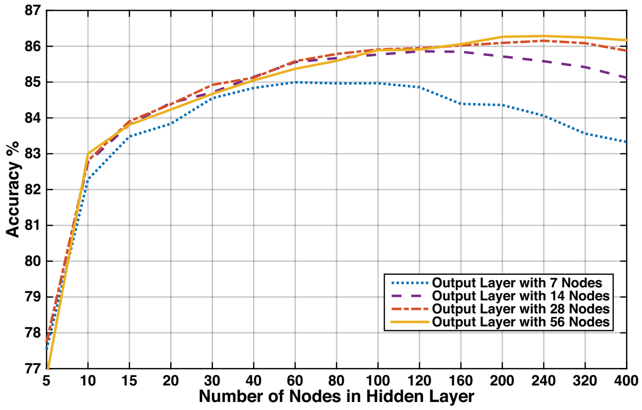

Fig. 10. Effect of Output Layer Modification

IV. Results

In previous two chapters, we have used overall accuracy as performance metric to observe the effect of selected features and to decide the best ANN sizes. Now we will compare two architectures (60 hidden, 7 output and 240 hidden, 56 output) in detail by using confusion matrices and confusion tables. We will also reference the results found in [7] for comparison.

In Table II and Table III, confusion matrices [13] for ANNs with 60 nodes hidden layer and 7 nodes output layer (60-7) and 240 nodes hidden layer and 56 nodes output layer (240-56) are given. Beside confusion matrices, performance metrics such as accuracy, True Positive Rate (TPR) also known as Sensitivity and Positive Predictive Value (PPV) also known as Precision are provided as well. Calculations of these metrics are shown in (9), (10) and (11). Accuracy metric is shown in the intersection of TPR and PPV in darker shaded cell. We can observe that output layer extension and following hidden layer modification have clearly enhanced sensitivity and precision for each class. At first glance, this enhancement can be underestimated. However, if we evaluate it in terms of error rate, this modification reduces the overall error rate about 9%, and class based error rate up to 18% (Class 4). On the other hand, Class 5 is the best-classified motion primitive for all cases and its accuracy is almost constant regardless of chosen architecture. So for cases where higher accuracy is required, we may prefer to use such extended ANN at the expense of some more training time and memory size requirements.

$$TPR = \frac{TP}{TP + FN} \tag{9}$$

$$PPV = \frac{TP}{TP + FP} \tag{10}$$

$$ACCURACY = \frac{TP + TN}{P + N} \tag{11}$$

Table II. Confusion Matrix for 60-7 Architecture using repeated random sub-sampling

		Predicted Class							
	%	1	2	3	4	5	6	7	TPR
Actual Class	1	90.10	1.61	0.85	0.08	1.04	0.15	6.18	90.10
	2	0.01	95.26	0.02	4.65	0.03	0.01	0.04	95.26
	3	0.34	0.72	84.90	0.48	0.22	13.02	0.33	84.90
	4	0.03	5.97	0.09	93.08	0.09	0.03	0.72	93.08
	5	1.51	0.51	0.04	0.16	97.56	0.07	0.15	97.56
	6	1.19	0.54	10.80	1.02	1.08	85.28	0.09	85.28
	7	11.24	0.00	0.57	1.12	0.67	0.93	85.47	85.47
	PPV	86.28	91.07	87.28	92.55	96.90	85.72	91.93	90.23

Table III. Confusion Matrix for 240-56 Architecture using repeated random sub-sampling

		Predicted Class							
	%	1	2	3	4	5	6	7	TPR
Actual Class	1	90.98	1.33	0.65	0.03	0.93	0.03	6.06	90.98
	2	0.00	95.80	0.01	4.17	0.00	0.00	0.02	95.80
	3	0.19	0.87	86.54	0.55	0.11	11.54	0.22	86.54
	4	0.00	4.92	0.01	94.34	0.04	0.05	0.65	94.34
	5	1.55	0.52	0.02	0.05	97.67	0.05	0.14	97.67
	6	1.27	0.56	9.68	1.00	1.40	85.95	0.15	85.95
	7	10.21	0.00	0.22	1.16	0.72	0.89	86.81	86.81
	PPV	87.31	92.12	89.11	93.13	96.84	87.27	92.31	91.15

In Table IV, final comparison is made between all results we have obtained and the one given in [7] as recognition accuracy based on Dynamic Time Warping (DTW) and Mahalanobis Distance (MD). In [7], since results are stated in terms of TPR i.e. sensitivity and True Negative Rate (TNR) i.e. specificity, we will also express our results using these two performance metrics defined as (9) and (12), respectively.

$$TNR = \frac{TN}{TN + FP} \tag{12}$$

Both 2 ANNs outperform the result obtained in [7] in terms of TNR. Although 100% TPR is acquired using DTW + MD comparison method for Classes 2 and 4 (for which we also achieve TPR about 95%), for other 5 classes both 2 ANNs are far more accurate than the method in [7]. In fact, 0% TPR obtained for Class 5 makes usage of this method as a 7-class classifier impractical. On the other hand, both 2 ANNs provide well-balanced and high accuracy classification.

Table IV. Comparison of results of two ANN architectures using repeated random sub-sampling and DTW + MD method

Class	**60-7, RRSS**		**240-56, RRSS**		**DTW + MD[7]**	
	TPR	**TNR**	**TPR**	**TNR**	**TPR**	**TNR**
1	90.10	97.61	90.98	97.81	20.00	93.34
2	95.26	98.44	95.80	98.64	100.00	83.34
3	84.90	97.94	86.54	98.24	60.00	66.67
4	93.08	98.75	94.34	98.85	100.00	80.00
5	97.56	99.48	97.67	99.48	0.00	93.34
6	85.28	97.63	85.95	97.92	60.00	83.34
7	85.47	98.75	86.81	98.80	40.00	70.00

V. Conclusions

In this study, we explored the application of a supervised machine learning algorithm to create meaningful information from raw acceleration data. Statistically rich spectrotemporal features from a publicly available wrist-worn accelerometer dataset [7] were extracted, followed by evaluation of an extensive list of different network topologies with several hidden layer architectures, output layer extensions, etc. to find the optimal supervised learning setup operating on these features. A remarkable accuracy increase of up to 91.15% is observed when compared to the latest publication studying the same dataset [7]. For more accurate evaluation, class base performances are also presented via confusion matrices which clearly demonstrate a high accuracy classifier for this 7-class inertia recognition problem. By presenting the results and methodology in detail, our goal is to ultimately compete with future studies using the same dataset.

References

[1] K.-Y. Chung, J. Yoo, and K. Kim, "Recent trends on mobile computing and future networks," *Pers. Ubiquitous Comput.*, vol. 18, no. 3, pp. 489–491, 2014.

[2] P. Groves, B. Kayyali, D. Knott, and S. Van Kuiken, "The 'big data'revolution in healthcare," *McKinsey Quarterly*, 2013.

[3] Allied Market Research, "Global mHealth Market (Devices, Services, Application, Stakeholders and Geography) - Industry Analysis, Market Size, Share, Growth, Company Profiles, Demand, Insights, Opportunities, Trends and Forecast, 2014 – 2020," Available: https://www.alliedmarketresearch.com/mhealth-2015-market. [Accessed: Jun. 15, 2015]

[4] "Hacking the human OS," *Spectrum, IEEE*, vol. 52, no. 6. p. 31, 2015.

[5] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA: MIT Press, 2014.

[6] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, 2014.

[7] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza, and R. Zaccaria, "Analysis of human behavior recognition algorithms based on acceleration data," *2013 IEEE Int. Conf. Robot. Autom.*, pp. 1602–1607, 2013.

[8] L. Gao, A. K. Bourke, and J. Nelson, "Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems.," *Med. Eng. Phys.*, vol. 36, no. 6, pp. 779–85, 2014.

[9] A. M. Khan, Y. K. Lee, and T. S. Kim, "Accelerometer signal-based human activity recognition using augmented autoregressive model coefficients and artificial neural nets," *Conf. Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 2008, pp. 5172–5175, 2008.

[10] T. Huynh and B. Schiele, "Analyzing Features for Activity Recognition," *Proc. 2005 Jt. Conf. Smart objects Ambient Intell. Innov. Context. Serv. usages Technol.*, no. october, pp. 159–163, 2005.

[11] W. Dargie, "Analysis of time and frequency domain features of accelerometer measurements," *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN*, no. 1, pp. 3–8, 2009.

[12] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[13] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sens. Environ.*, vol. 62, no. 1, pp. 77–89, 1997.