Feature Generation for Physical Activity Classification*

Abstract. The paper investigates the human physical activity classification problem. Time series from accelerometer of a wearable device procude a dataset. Due to high dimension of the object description and low computational resources one has to state a feature generation problem. The authors propose to use parameters of the local approximation models as informative features. The experiment is conducted on two datasets for human activity recognition using accelerometer: WISDM and USC-HAD. It compares several superpositions of various generation and classification models.

Keywords: wearable devices, accelerometer, time series, local approximation, classification

1 Introduction

The paper investigates the multiclass classification problem of objects with no explicit feature representation. This problem arises in analysing biological data [17], human behavior and social interactions [1]. It considers the problem of human activity recognition. The accelerometer time series [8, 15, 20] from smart phones serve to recognize human physical activity in the internet of things [2, 14]. Methods to solve this problem range from topological data analysis [18] to convolutional neural networks [6]. The extensive survey of methods and datasets for this problem is in [9].

In this work the dataset collects time series of acceleration from three axis, which is obtained from a mobile phone or another wearable device with accelerometer. These time series are of various sizes, not aligned and multiscaled [4]. The problem is to predict physical activity of a person. The list of activities includes walking, running, sitting or walking up/down stairs. In this setup the time series are treated as complex structured objects without explicit feature description. This assumption allows to propose a flexible technology of accelerometer time series modelling. The main problem to tackle is the lack of computational resources, memory and energy in wearable devices. This investigation proposes an approach to generate features of time series as complex structured objects. The generated features bring adequate quality of classification and require moderate resources.

The problem of classifying complex structured objects is split in two distinctive procedures. The first extracts informative features. The second one classifies

* This research is supported by RFBR project 16-07-01154

objects of these feature descriptions. This research focuses mainly on comparison of different methods of feature generation [11, 12]: expert-defined functions, autoregressive model and singular spectrum analysis. The expert-defined functions [13] include the average, standard deviation, mean absolute deviation and histogram. The autoregressive model [16] builds a parametric model for each time series and use parameters of the model as features for classification. The singular spectrum analysis [7] uses the eigenvalues of trajectory matrix as generated features.

The authors propose a new feature generation method. We approximate time series segments with cubic splines [3]. The spline approximates the 3-order piecewise curve at the given knots. The additional smoothess conditions makes the curve and its first and second derivatives continuous. The splines give a smooth curve and adequate quality of approximation.

The experiment was conducted on two accelerometer datasets: WISDM [21], USC-HAD [19]. We compared the performance of stated feature extraction methods, as well as different classification algorithms. The latter include logistic regression, random forest and SVM.

2 Problem Statement

The accelerometer time series is represented as a set $\mathcal{S}$ of segments s of fixed length T:

$$s = [x_1, \dots, x_T]^\mathsf{T} \in \mathbb{R}^T. \tag{1}$$

One has to find a classification model $f : \mathbb{R}^T \to Y$ between segments from the set $\mathcal{S}$ and class labels from a finite set Y. Denote by

$$\mathcal{D} = \{(s_i, y_i)\}_{i=1}^m \tag{2}$$

a given sample set, where $s_i \in \mathcal{S}$ and $y_i = f(s_i) \in Y$.

The authors propose to construct the model f as a superposition $f = f(\boldsymbol{g})$. Here $\boldsymbol{g} : \mathbb{R}^T \to \mathbb{R}^n$ is a map from the space $\mathbb{R}^T$ to the feature space $G \subset \mathbb{R}^n$. Given the feature map $\boldsymbol{g}$ transform the original sample set (2) to the new sample set

$$\mathcal{D}_G = \{(\boldsymbol{g}_i, y_i)\}_{i=1}^m,$$

where $\boldsymbol{g}_i = \boldsymbol{g}(s_i) \in G$.

The classification model $f = f(\boldsymbol{g}, \boldsymbol{\theta})$ has a vector of parameters $\boldsymbol{\theta}$. The optimal parameters $\hat{\boldsymbol{\theta}}$ are given by the classification error function

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D}_G, \boldsymbol{\mu}). \tag{3}$$

Here the vector $\boldsymbol{\mu}$ is external parameters of a particular classification model. The examples of these parameters and error functions for different classification models are given below.

To compare classification quality with results from [11, 12] use the accuracy score:

$$\text{accuracy} = \frac{1}{m} \sum_{i=1}^{m} \left[f\left(\boldsymbol{g}(s_i), \hat{\boldsymbol{\theta}} \right) = y_i \right]. \tag{4}$$

3 Feature Generation Functions

The main focus of this paper is to compare different approaches for feature generation. In this section we provide analysis and motivation for each of the methods.

Expert Functions. Use the expert-given feature set as the baseline for local approximation models. These functions are statistics g_j, where $g_j : \mathbb{R}^T \to \mathbb{R}$. The description $\boldsymbol{g}(s)$ of the object s is the value of these statistics on the object

$$\boldsymbol{g}(s) = [g_1(s), \dots, g_n(s)]^\mathsf{T}.$$

In paper [13] the authors proposed to use the expert functions listed in table 1. This feature generation procedure extracts the feature description of time series $\boldsymbol{g}(s) \in \mathbb{R}^{40}$.

Table 1: Expert functions

Function description	**Formula**
Mean	$\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t$
Standard deviation	$\sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t - \bar{x})^2}$
Mean absolute deviation	$\frac{1}{T}\sum_{t=1}^{T} \lvert x_t - \bar{x}\rvert$
Distribution	Histogram values with 10 bins

Autoregressive Model. The autoregressive model [16] of the order n generates features of time series s with model parameters. Each time series is approximated by a linear combination of its previous $n-1$ components

$$x_t = w_0 + \sum_{j=1}^{n-1} w_j x_{t-j} + \epsilon_t,$$

where ϵ_t is a residual. The optimal parameters $\hat{\boldsymbol{w}}$ of the autoregressive model are the features $\boldsymbol{g}(s)$. These parameters minimize the squared error between the time series s and its prediction

$$\boldsymbol{g}(s) = \hat{\boldsymbol{w}} = \mathop{\arg\min}_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=n}^{T} \|x_t - \hat{x}_t\|^2 \right). \tag{5}$$

The problem (5) is a linear regression problem. Hence, for each initial time series s we have to solve linear regression problem with n predictors. The example of approximation using autoregressive model is demonstrated on the Fig. 1.

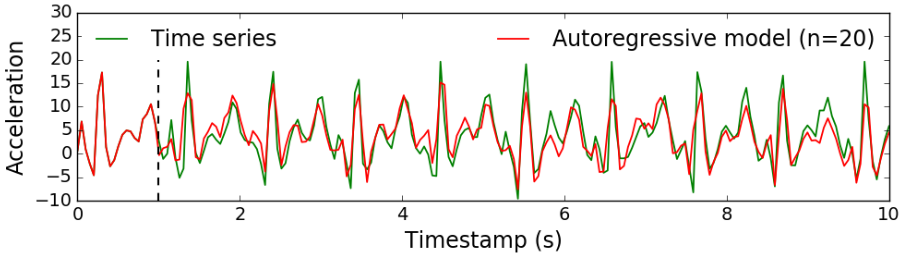


Fig. 1: Time series approximation using autoregressive model with order $n = 20$

Singular Spectrum Decomposition. Alternative hypothesis for generation of time series is Singular Spectrum Analysis (SSA) model [7]. We construct trajectory matrix for each time series s from the original sample $\mathcal{D}$:

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_3 & \dots & x_{n+1} \\ \dots & \dots & \dots & \dots \\ x_{T-n+1} & x_{T-n+2} & \dots & x_T \end{pmatrix}.$$

Here n is the window width, which is an external structure parameter. The singular decomposition [5] of the matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$:

$$\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\mathsf{T},$$

where $\mathbf{U}$ is a unitary matrix and $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ whose entries λ_i are eigenvalues of $\mathbf{X}^\mathsf{T}\mathbf{X}$. The spectrum of the matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$ is used as feature description of the object s:

$$\boldsymbol{g}(s) = [\lambda_1, \dots, \lambda_n]^\mathsf{T}.$$

Spline Approximation. The proposed method approximates time series with splines [3]. A spline is defined by its parameters: knots and coefficients. The set of knots $\{\xi_\ell\}_{\ell=0}^{M}$ are uniformly distributed over time series. The models, which are built on each the interval $[\xi_{\ell-1}; \xi_\ell]$, are given by the coefficients $\{\mathbf{w}_\ell\}_{\ell=1}^{M}$. Optimal spline parameters are solution of a system with additional constraints of equality of derivatives up to second order on the edges of intervals. Denote each spline segment as $p_i(t)$ $i = 1, \dots, M$ and spline as a whole as $S(t)$ and write these equations:

$$S(t) = \begin{cases} p_1(t) = w_{10} + w_{11}t + w_{12}t^2 + w_{13}t^3, & t \in [\xi_0, \xi_1], \\ p_2(t) = w_{20} + w_{21}t + w_{22}t^2 + w_{23}t^3, & t \in [\xi_1, \xi_2], \\ \dots & \dots \\ p_M(t) = w_{L0} + w_{M1}t + w_{M2}t^2 + w_{M3}t^3, & t \in [\xi_{M-1}, \xi_M], \end{cases}$$

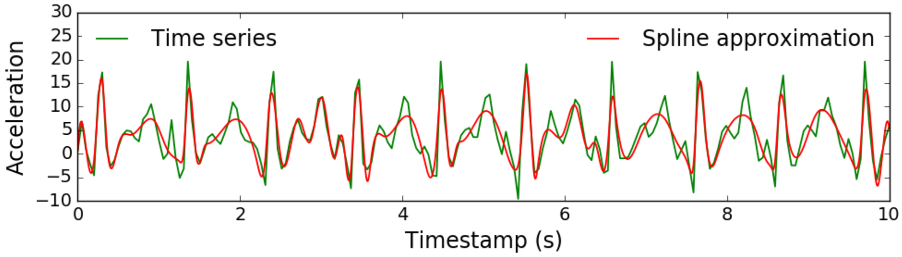


Fig. 2: Time series approximation using three order splines

For $S(t)$ to be an interpolatory cubic spline, we must also have conditions:

$$\begin{aligned} S(\xi_t) &= x_t, \quad t = 0, \dots, M, \\ p_i'(\xi_i) &= p_{i+1}'(\xi_i),\, p_i''(\xi_i) = p_{i+1}''(\xi_i), \quad i = 1, \dots, M-1, \\ p_i(\xi_{i-1}) &= x_{i-1},\, p_i(\xi_i) = x_i, \quad i = 1, \dots, M. \end{aligned}$$

The feature description of the time series could be assumed as a union of the spline parameters:

$$\boldsymbol{g}(s) = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_M]^{\mathsf{T}}.$$

Fig. 2 shows the time series approximation given by splines. Compared to the autoregressive model, the splines method gives smoother approximation using almost the same number of parameters.

4 Time Series Classification

Multiclass classification uses one-vs-rest approach to train binary classifiers for each class label and then, on the prediction step, classify new object according to the most confident classifier. Three classification models are used: logistic regression, SVM and random forest.

Regularized Logistic Regression. The optimal model parameters (3) is determined by minimising the error function

$$L(\boldsymbol{\theta}, \mathcal{D}_G, \mu) = \sum_{i=1}^{m} \log\bigl(1 + \exp(-y_i[\boldsymbol{w}^{\mathsf{T}}\boldsymbol{g}_i + b])\bigr) + \frac{\mu}{2}\|\boldsymbol{w}\|^2, \text{ where } \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}.$$

Thus, the optimal parameters $\hat{\boldsymbol{w}}, \hat{b}$ is given by (3).

The classification rule $f(\boldsymbol{g}, \boldsymbol{\theta})$ is given by sign of the linear combination for the object description $\boldsymbol{g}$ and parameters $\hat{\boldsymbol{\theta}}$

$$\hat{y} = f(\boldsymbol{g}, \hat{\boldsymbol{\theta}}) = \operatorname{sgn}(\boldsymbol{g}^{\mathsf{T}}\hat{\boldsymbol{w}} + \hat{b}).$$

SVM. The optimization problem is

$$\hat{\boldsymbol{\theta}} = \begin{pmatrix} \hat{\boldsymbol{w}} \\ \hat{b} \\ \hat{\boldsymbol{\xi}} \end{pmatrix} = \arg\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \mu \sum_{i=1}^{m} \xi_i, \text{ s.t. } y_i \left(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{g}_i + b\right) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \quad 1 \leq i \leq m.$$

The objective function corresponds to the classification error function $L(\boldsymbol{\theta}, \mathcal{D}_G, \mu)$. The prediction for new object is $\hat{y} = \text{sgn}(\boldsymbol{g}^{\mathsf{T}}\hat{\boldsymbol{w}} + \hat{b})$.

Random Forest. The random forest exploits the idea of bagging. This is an approach of building many random unstable classifiers and aggregating their predictions. This method works especially well if as base models we select models with low bias and high variance (due to aggregating variance is reduced). In case of random forest decision trees take the role of base models, also not only objects are used for bagging, but also features. In this case we make the prediction for each new object as the mean of the predictions of a single tree:

$$\hat{y} = \frac{1}{B}\sum_{i=1}^{B} \text{pred}(\boldsymbol{g}_i),$$

where B is an amount of trees used for bagging.

5 Experiment

In this paper we considered two different smart phone based datasets: WISDM [21] and USC-HAD [19]. The smart phone accelerometer measures acceleration along three axis with frequencies equal to 20 and 100 Hz. The WISDM dataset consists of 4321 time series. Each time series belongs to one of the six activities: Standing, Walking, Upstairs, Sitting, Jogging, Downstairs. The USC-HAD dataset contains 13620 time series with one of the twelve class labels: Standing, Elevator-up,Walking-forward, Sitting, Walking-downstairs, Sleeping, Elevator-down, Walking-upstairs, Jumping, Walking-right, Walking-left, Running. Table 2 shows the distributions of time series activities for each datasets. The length T of each time series equals 200. Fig. 3 plots the example of the time series for one activity of the specific person is given.

For each dataset apply feature generation procedures: expert functions, autoregressive model, SSA and splines. Three classification models for each generated feature description: logistic regression, support vector machine and random forest. The structure parameters: the length n for autoregressive model, the window width n for SSA and the number of splines knots M were tuned using K-fold cross validation, minimizing

$$CV(K) = \frac{1}{K}\sum_{k=1}^{K} L(f_k, \mathcal{D} \setminus \mathcal{C}_k), \tag{6}$$

Table 2: Distributions of the classes

(a) WISDM

	Activity	**# objects**	
1	Standing	229	5.30 %
2	Walking	1917	44.36 %
3	Upstairs	466	10.78 %
4	Sitting	277	6.41 %
5	Jogging	1075	24.88 %
6	Downstairs	357	8.26 %
	Total	4321	

(b) USC-HAD

	Activity	**# objects**	
1	Standing	1167	8.57 %
2	Elevator-up	764	5.61 %
3	Walking-forward	1874	13.76 %
4	Sitting	1294	9.50 %
5	Walking-downstairs	951	6.98 %
6	Sleeping	1860	13.66 %
7	Elevator-down	763	5.60 %
8	Walking-upstairs	1018	7.47 %
9	Jumping	495	3.63 %
10	Walking-right	1305	9.58 %
11	Walking-left	1280	9.40 %
12	Running	849	6.23 %
	Total	13620	

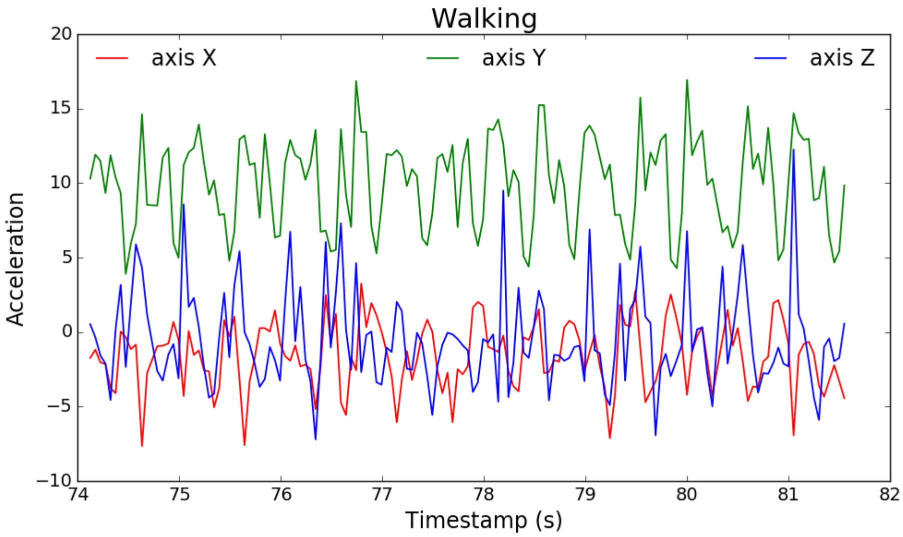


Fig. 3: Time series example

where C_k is a $\frac{K-1}{K}$ fraction of data, used for training model f_k. The hyperparameters $\boldsymbol{\mu}$ for classification models were also tuned using the same cross validation procedure.

The first approach for feature generation is expert functions. The main drawback of this approach is a restriction by the choice of the expert functions and these functions might be impossible to derive for some types of data.

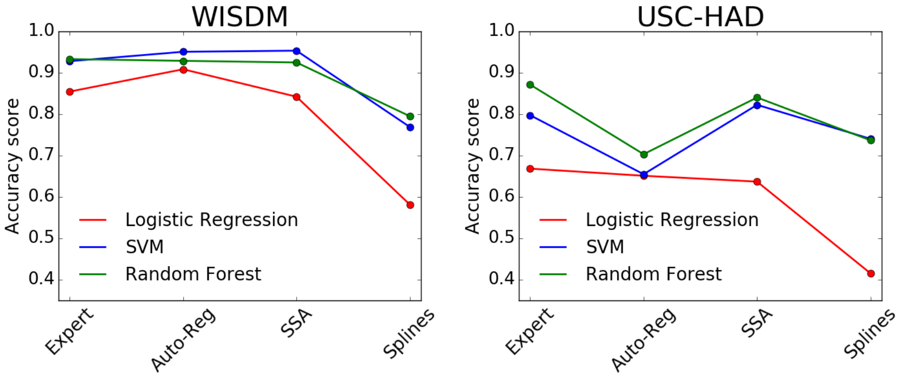


Fig. 4: Multiclass accuracy score

The autoregressive model was tuned to find the optimal length n. Cross validation procedure gives optimal value $n = 20$ for both dataset.

The singular spectrum analysis was tuned in the same way to find the optimal window width n. Similar to autoregressive model, cross validation procedure gives the same value $n = 20$.

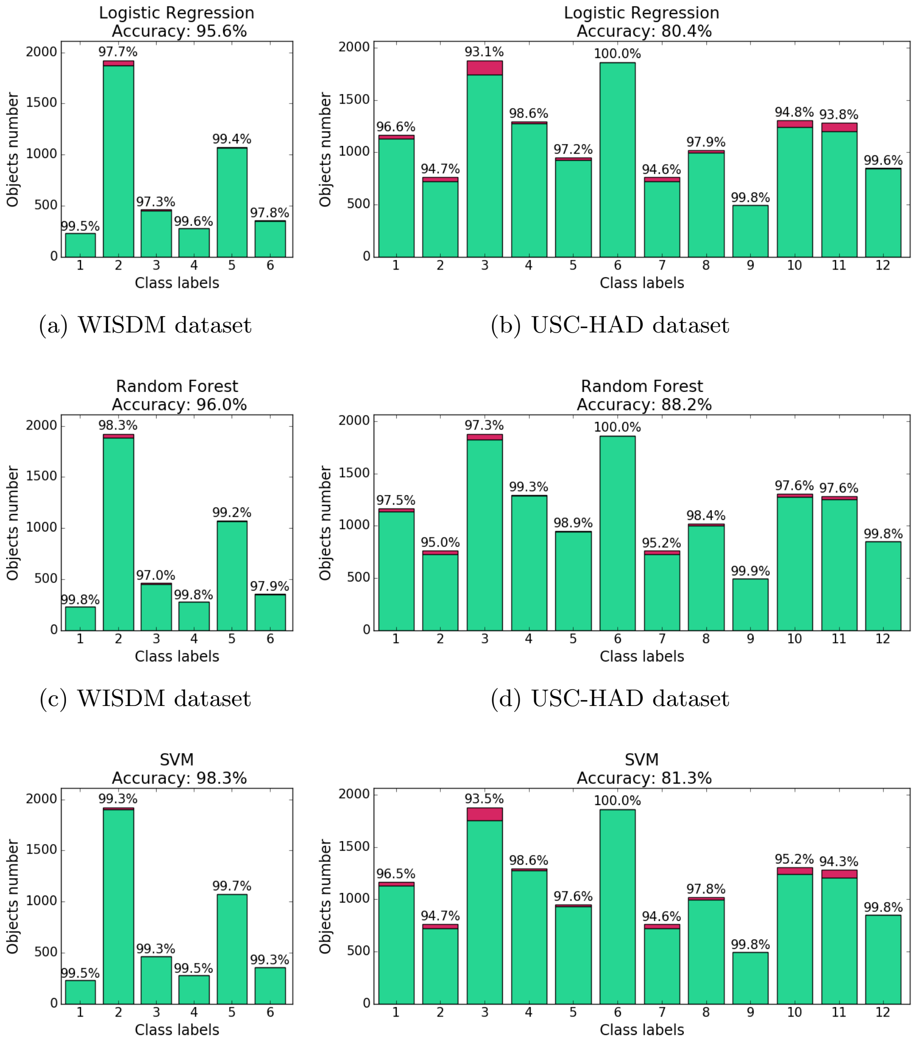
Fit cubic splines [3] for time series using *scipy* python library [10]. The knots $\{\xi_\ell\}_{\ell=1}^{M}$ for splines were distributed uniformly. Value of M was chosen with cross validation.

The feature extraction methods gives the following number of features for both datasets: expert features: 40; autoregressive model: 60; singular spectrum analysis: 60; splines: 33.

Table 3: Binary accuracy scores for WISDM using different feature generation methods: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines

	Logistic Regression				**Random Forest**				**SVM**			
	EX	AR	SSA	SPL	EX	AR	SSA	SPL	EX	AR	SSA	SPL
All	0.85	0.91	0.84	0.58	0.93	0.93	0.92	0.79	0.93	0.95	0.95	0.77
Standing	0.99	0.98	1.00	0.95	1.00	0.99	1.00	0.99	0.99	0.98	1.00	0.96
Walking	0.91	0.96	0.86	0.61	0.96	0.97	0.95	0.86	0.96	0.98	0.98	0.84
Upstairs	0.91	0.95	0.91	0.89	0.96	0.96	0.96	0.90	0.96	0.98	0.97	0.89
Sitting	0.99	0.98	1.00	0.99	1.00	0.99	1.00	1.00	0.99	0.98	1.00	1.00
Jogging	0.98	0.99	0.99	0.80	0.99	0.99	0.99	0.92	0.99	0.99	0.99	0.93
Downstairs	0.93	0.96	0.94	0.92	0.96	0.97	0.96	0.92	0.96	0.98	0.97	0.92

Fig. 4 presents the accuracy scores (4) of the experiments for the both datasets. For WISDM dataset the worst result was obtained with spline approximation. The results for expert functions, autoregressive model and SSA is



(a) WISDM dataset

(b) USC-HAD dataset

(c) WISDM dataset

(d) USC-HAD dataset

(e) WISDM dataset

(f) USC-HAD dataset

Fig. 5: Accuracy scores of classification of each class using all features

roughly identical. For USC-HAD dataset the results highly depend on the classification model. For both datasets logistic regression shows the worst quality, while the accuracy for support vector machine and random forest is almost the same. The spline approximation gives competitive result for USC-HAD dataset.

Table 3 and table 4 presents all results with classification accuracy scores (4) for each class. The first row of these tables introduces the multiclass accuracy score for each classification model and each feature extraction procedure. Next

Table 4: Binary accuracy scores for USC-HAD using different feature generation methods: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines

	Logistic Regression				**Random Forest**				**SVM**			
	EX	AR	SSA	SPL	EX	AR	SSA	SPL	EX	AR	SSA	SPL
All	0.67	0.65	0.64	0.41	0.87	0.70	0.84	0.74	0.80	0.65	0.82	0.74
Standing	0.94	0.94	0.92	0.89	0.98	0.94	0.97	0.98	0.95	0.94	0.97	0.96
Elevator-up	0.94	0.94	0.93	0.92	0.95	0.95	0.95	0.95	0.93	0.94	0.94	0.93
Walking-forward	0.87	0.87	0.89	0.70	0.97	0.89	0.96	0.88	0.95	0.87	0.97	0.91
Sitting	0.98	0.95	0.94	0.96	0.99	0.96	0.98	0.99	0.98	0.96	0.99	0.99
Walking-downstairs	0.95	0.93	0.93	0.90	0.99	0.96	0.98	0.95	0.98	0.93	0.98	0.96
Sleeping	1.00	0.98	0.99	1.00	1.00	0.98	1.00	1.00	1.00	0.98	1.00	1.00
Elevator-down	0.94	0.94	0.94	0.91	0.95	0.95	0.95	0.95	0.93	0.94	0.94	0.93
Walking-upstairs	0.94	0.95	0.93	0.92	0.98	0.95	0.98	0.96	0.98	0.95	0.98	0.96
Jumping	0.99	0.99	1.00	0.97	1.00	0.99	1.00	0.99	1.00	0.99	0.97	0.99
Walking-right	0.91	0.90	0.91	0.86	0.97	0.92	0.96	0.92	0.96	0.90	0.97	0.93
Walking-left	0.89	0.91	0.90	0.88	0.97	0.93	0.97	0.93	0.95	0.91	0.97	0.93
Running	0.99	0.99	0.99	0.92	1.00	0.99	1.00	0.97	1.00	1.00	0.95	0.98

rows are related to binary accuracy scores for each class. For WISDM dataset the best scores have the least active classes such as Standing and Sitting. For USC-HAD dataset all classes have the similar accuracy scores.

We also carried out the experiment for union of all 193 generated features. Fig. 5 demonstrates the results. Table 2 shows class labels, that are represented on the corresponding histograms. As expected, the accuracy scores for feature union are higher in all cases. All binary accuracy scores for WISDM dataset is higher than 97% for each classification model. These numbers for USC-HAD dataset is higher than 93%.

6 Conclusion

The paper investigates the problem of complex structured objects classification. The experiment compares various approaches of feature extraction, particularly the expert functions and local approximation models on data from smart phone accelerometer. The logistic regression, SVM and random forest are used for classification. The results show that obtained features recovers the class label with the high quality. The proposed spline method gives smooth approximation of time series. The number of splines parameters was less than for the other methods. The classification quality for splines are competitive with existing stated methods for both considered datasets. Stacking of all extracted features gives better performance.

References

1. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: International Conference on Pervasive Computing. pp. 1–17. Springer (2004)

2. Budnik, M., Gutierrez-Gomez, E.L., Safadi, B., Pellerin, D., Quénot, G.: Learned features versus engineered features for multimedia indexing. Multimedia Tools and Applications 76(9), 11941–11958 (2017)
3. De Boor, C.: A practical guide to splines, vol. 27. Springer-Verlag (1978)
4. Geurts, P.: Pattern extraction for time series classification. In: European Conference on Principles of Data Mining and Knowledge Discovery. pp. 115–127. Springer (2001)
5. Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. Numerische mathematik 14(5), 403–420 (1970)
6. Hammerla, N.Y., Halloran, S., Ploetz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv preprint arXiv:1604.08880 (2016)
7. Hassani, H.: Singular spectrum analysis: methodology and comparison. Journal of Data Science 5(2), 239–257 (2007)
8. Ignatov, A.D., Strijov, V.V.: Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. Multimedia tools and applications 75(12), 7257–7270 (2016)
9. Incel, O.D., Kose, M., Ersoy, C.: A review and taxonomy of activity recognition on mobile phones. BioNanoScience 3(2), 145–171 (2013)
10. Jones, E., Oliphant, T., Peterson, P.: SciPy: Open source scientific tools for Python (2001), http://www.scipy.org/, [Online; accessed today]
11. Karasikov, M., Strijov, V.: Feature-based time-series classification. Intelligence 24(1), 164–181 (2016)
12. Kuznetsov, M., Ivkin, N.: Time series classification algorithm using combined feature description. Machine Learning and Data Analysis 1(11), 1471–1483 (2015)
13. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. ACM SigKDD Explorations Newsletter 12(2), 74–82 (2011)
14. Lu, L., Qing-ling, C., Yi-Ju, Z.: Activity recognition in smart homes. Multimedia Tools and Applications pp. 1–18 (2016)
15. Lu, Y., Wei, Y., Liu, L., Zhong, J., Sun, L., Liu, Y.: Towards unsupervised physical activity recognition using smartphone accelerometers. Multimedia Tools and Applications 76(8)
16. Lukashin, Y.P.: Adaptive methods of short-term forecasting of time series. M.: Finance and statistics (2003)
17. Motrenko, A., Strijov, V.: Extracting fundamental periods to segment biomedical signals. IEEE journal of biomedical and health informatics 20(6), 1466–1476 (2016)
18. Umeda, Y.: Time series classification via topological data analysis. Transactions of the Japanese Society for Artificial Intelligence 32(3), D–G72_1 (2017)
19. The usc human activity dataset. http://www-scf.usc.edu/ mizhang/datasets.html
20. Wang, W., Liu, H., Yu, L., Sun, F.: Human activity recognition using smart phone embedded sensors: A linear dynamical systems method. In: Neural Networks (IJCNN), 2014 International Joint Conference on. pp. 1185–1190. IEEE (2014)
21. The wisdm dataset. http://www.cis.fordham.edu/wisdm/dataset.php