Metamodels for Complex Structured Objects Classification

Roman Isachenko[1] and Ilya Zharikov[2] Artem Bochkarev[3]

[1] Moscow Institute of Physics and Technology, Institutskiy lane 9, Dolgoprudny, 141700, Russia
Skolkovo Institute of Science and Technology, Nobel street 3, Moscow, 143026, Russia
isa-ro@yandex.ru
[2] Moscow Institute of Physics and Technology, Institutskiy lane 9, Dolgoprudny, 141700, Russia
Skolkovo Institute of Science and Technology, Nobel street 3, Moscow, 143026, Russia
ilya250894@gmail.com
[3] Moscow Institute of Physics and Technology, Institutskiy lane 9, Dolgoprudny, 141700, Russia
Skolkovo Institute of Science and Technology, Nobel street 3, Moscow, 143026, Russia
artem.bochkarev@phystech.edu

Abstract. The development and proliferation of various portable sensors poses new challenges for analyzing and finding meaning in this data. In our work we investigate classification of complex structured objects. One of the main problems in this task is to generate meaningful and relatively small set of features. We compare several approaches for feature extraction such as expertly defined features, autoregressive model and SSA. We propose a new feature generation algorithm, based on local spline approximation. The experiment is conducted on two datasets for human activity recognition using accelerometer.

Keywords: complex structured objects, time series, local approximation models

1 Introduction

This paper investigates the multiclass classification problem of complex structured objects (i.e. we don't have feature representation suitable for direct classification). The application is the human activity recognition. The accelerometer time series [7, 14, 18] from mobile phones serve to recognize people activity in the internet of things smart homes [1, 13]. New methods in this field range from topological data analysis [16] to convolutional neural networks [5]. The extensive survey of methods and datasets for this problem is in [8].

In our work the dataset collects time series of acceleration from three axis, which is obtained from the mobile phone or another wearable device with accelerometer. These time series are of different size, not aligned or multiscaled [3]. 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. First, we need to extract informative features, and then we use those features as input to some classifier to obtain final model. For simplicity, we assume that these two procedures can be built and analysed separately. In our project we focus mainly on comparing different methods of feature generation [10, 11]. Existing approaches include expert-defined functions [12], autoregressive model [15] and singular spectrum analysis [6].

Expert-defined functions allow to avoid the problem of feature extraction. These functions for time series task include average value, standard deviation, mean absolute deviation and distribution for each component. In autoregressive model we build parametric model for each time series and use parameters of the model as features for classification. Singular spectrum analysis proposes to use eigenvalues of trajectory matrix as features for building classifier.

The experiment was conducted on twoaccelerometer datasets: WISDM [19], USC-HAD [17]. 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

Let $\mathcal{S}$ be space of complex structured objects, Y is a finite set of class labels. We consider accelerometer time series as complex structured objects. Time series is represented as the vector with fixed length T:

$$s = [x_1, \ldots, x_T]^{\mathsf{T}} \in \mathcal{S}. \tag{1}$$

We assume that there is a hidden true dependence $f^* : \mathcal{S} \to Y$ between objects from the space $\mathcal{S}$ and their class labels from Y. Denote by $\mathcal{D} = \{(s_i, y_i)\}_{i=1}^m$ a given sample, where $s_i \in \mathcal{S}$ and $y_i = f^*(s_i) \in Y$. The problem is to recover the function f^*. We assume that the target function f^* can be approximated by some function $\hat{f}$ from the class of function compositions $f = g \circ \boldsymbol{h}$. Here $\boldsymbol{h} : \mathcal{S} \to H$ is a map from the original space $\mathcal{S}$ to the feature space $H \subset \mathbb{R}^n$, $g : H \times \Theta \to Y$ is a parametric map from the feature space H to the set of class labels Y. The function $\hat{g}$ corresponds to classification model which is parametrized by a vector parameter $\boldsymbol{\theta} \in \Theta$.

The determining of the approximation function $\hat{f}$ is equivalent to determining the optimal functions $\boldsymbol{h}$ and g. The function $\boldsymbol{h}$ corresponds to generating the appropriate feature space H. We consider different local approximation models

as the feature generation methods. In this case the features are the estimated parameters of the models.

Given appropriate feature space H and feature map $\boldsymbol{h}$ we transform our original sample $\mathcal{D} = \{s_i, y_i\}_{i=1}^m$ with complex structured objects to the new sample $\mathcal{D}_H = \{\mathbf{h}_i, y_i\}_{i=1}^m$, where $\mathbf{h}_i = \boldsymbol{h}(s_i) \in H$. The function $g(\mathbf{h}, \boldsymbol{\theta})$ is defined by its parameter vector $\boldsymbol{\theta} \in \Theta$. The optimal parameters $\hat{\boldsymbol{\theta}}$ are given by

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D}_H, \boldsymbol{\mu}), \tag{2}$$

where the function $L(\boldsymbol{\theta}, \mathcal{D}_H, \boldsymbol{\mu})$ is the classification error function. Here the vector $\boldsymbol{\mu}$ is a external parameters of the particular classification model. Examples of these parameters for different classification models are given below.

To evaluate the quality of our approximation we consider the accuracy score. This choice is based on our wish to compare our results with previous articles [10, **?**] and easy interpretation. Accuracy score is a relation between correctly classified objects and their total number in dataset:

$$\text{accuracy} = \frac{1}{m} \sum_{i=1}^{m} [y_i = \hat{y}_i],$$

where $\hat{y}_i = \hat{f}(s_i)$ is a prediction of the classifier.

3 Feature Generation

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

Expert Functions. We use the expert-given feature set as the baseline for local approximation models. These functions are statistics h_i, where $h_i : \mathcal{S} \to \mathbb{R}$. The description $\boldsymbol{h}(s)$ of the object s is the value of these statistics on the object

$$\boldsymbol{h}(s) = [h_1(s), \ldots, h_n(s)]^\mathsf{T}.$$

In paper [12] the authors proposed to use the expert functions listed in table 1. This feature generation procedure extracts the feature description of time series $\boldsymbol{h}(s) \in \mathbb{R}^{40}$.

Autoregressive Model. The autoregressive model [15] 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

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

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

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

$$\boldsymbol{h}(s) = \mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \left(\sum_{t=n}^{T} \|x_t - \hat{x}_t\|^2 \right). \tag{3}$$

The problem (3) 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.

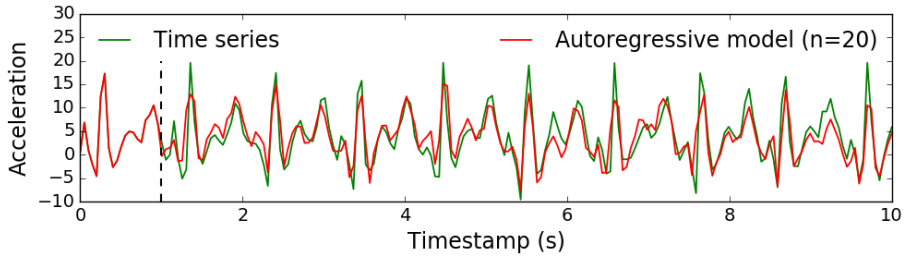


Fig. 1: Time series approximation using autoregressive model with order $n = 20$

Singular Spectrum Decomposition. Alternative hypothesis for generation of time series is SSA (Singular Spectrum Analysis) model [6]. 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 [4] of the matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$:

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

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

$$\boldsymbol{h}(s) = (\lambda_1, \ldots, \lambda_n).$$

Spline Approximation. The proposed method approximates time series with splines [2]. 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 $p_i(t)$ $i = 1, \ldots, M$ and spline as a whole as $S(t)$ and write these equations:

$$S(t) = \begin{cases} p_1(t) = w_{10} + w_{11}t + w_{12}t^2 + w_{13}t^3, & t \in [\xi_0, \xi_1], \\ p_2(t) = w_{20} + w_{21}t + w_{22}t^2 + w_{23}t^3, & t \in [\xi_1, \xi_2], \\ \ldots \\ p_M(t) = w_{L0} + w_{L1}t + w_{L2}t^2 + w_{L3}t^3, & t \in [\xi_{M-1}, \xi_M], \end{cases}$$

$$S(t) = x_t \quad t = 1, \ldots, T,$$

$$p_i'(\xi_i) = p_{i+1}'(\xi_i), \quad p_i''(\xi_i) = p_{i+1}''(\xi_i), \quad i = 1, \ldots, M-1.$$

The feature description of the time series could be assumed as a union of these parameters.

$$\boldsymbol{h}(s) = (\mathbf{w}_1, \ldots, \mathbf{w}_M).$$

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.

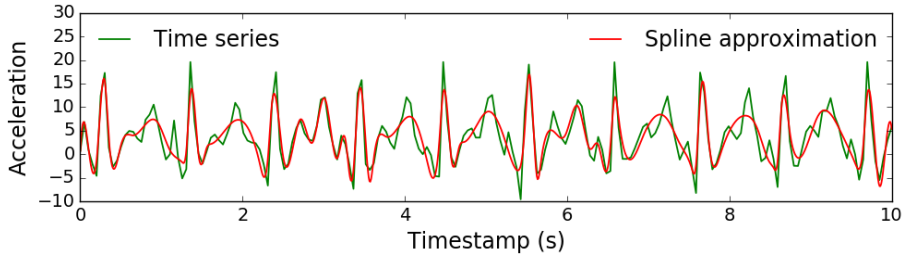


Fig. 2: Time series approximation using three order splines

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 (2) is determined by minimising the error function

$$L(\boldsymbol{\theta}, \mathcal{D}_H, \mu) = \sum_{i=1}^{m} \log\Big(1 + \exp(-y_i[\boldsymbol{w}^\mathsf{T}\mathbf{h}_i + b])\Big) + \frac{\mu}{2}\|\boldsymbol{w}\|^2,$$

where

$$\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{w} \\ b \end{pmatrix}.$$

Thus,

$$\boldsymbol{\theta}^* = \arg\min_{\theta} L(\boldsymbol{\theta}, \mathcal{D}_H, \mu).$$

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

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

SVM. The problem is

$$\begin{aligned} \boldsymbol{\theta}^* = \begin{pmatrix} \boldsymbol{w}^* \\ b^* \\ \boldsymbol{\xi}^* \end{pmatrix} = & \arg\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i, \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \mathbf{h}_i\rangle + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad 1 \leq i \leq m. \end{aligned}$$

The prediction for new object is

$$\hat{y} = \operatorname{sgn}(\mathbf{h}^\mathsf{T}\boldsymbol{w}^* + b^*).$$

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:

$$\hat{y} = \frac{1}{B} \sum_{i=1}^{B} g(\mathbf{h}_i),$$

where B is an amount of trees used for bagging.

5 Experiment

In this paper we considered two different smart phone based datasets: WISDM [19] and USC-HAD [17]. The smart phone accelerometer measures acceleration along three axis. Frequency ranges from 20 to 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. The distributions of time series activities for each datasets are presented in tfable 2. The length of each time series equals 200 which accounts 10 second. In the Fig. 3 the example of the time series for one activity of the specific person is given.

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	

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 L, were tuned using K-fold

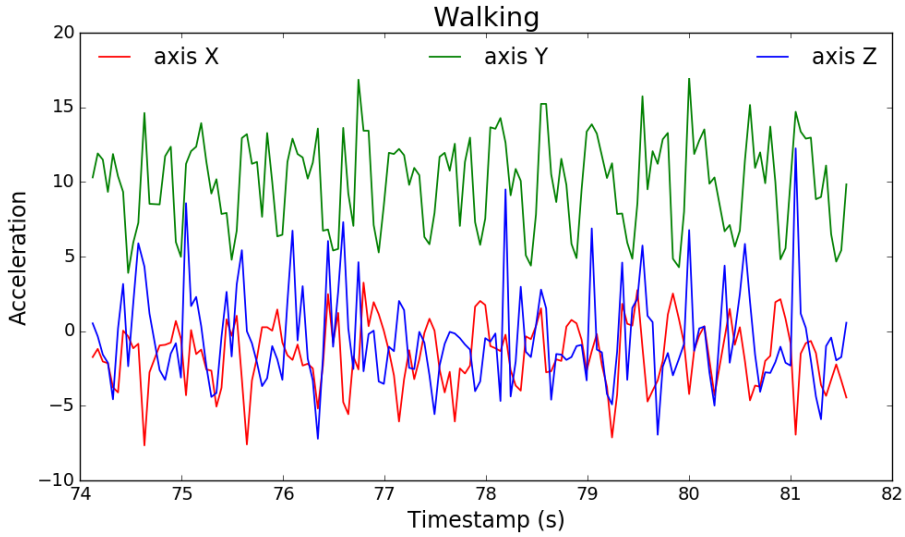


Fig. 3: Time series example

cross validation, minimizing

$$CV(K) = \frac{1}{K}\sum_{k=1}^{K} L(f_k, \mathcal{D} \setminus \mathcal{C}_k), \tag{4}$$

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 that we are restricted by our choice of the expert functions and these functions might be impossible to derive for some types of data.

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 $n = 20$.

We fit cubic splines [2] for time series using *scipy* python library [9]. 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: 63; singular spectrum analysis: 60; splines: 33.

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

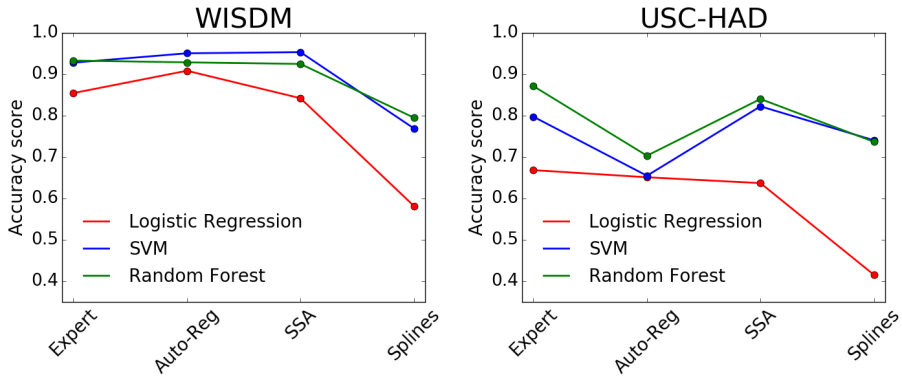


Fig. 4: Multiclass accuracy score

All results with classification accuracy scores for each class are represented in Table 3 and Table 4. The first row of these tables introduces the multiclass accuracy score for each classification model and each feature extraction procedure. Next 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.

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

We also carried out the experiment for union of all 196 generated features. The results are demonstrated on the Fig. 5. In the Table 2 one can see class labels, that are represented on the corresponding histograms. As expected, the

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

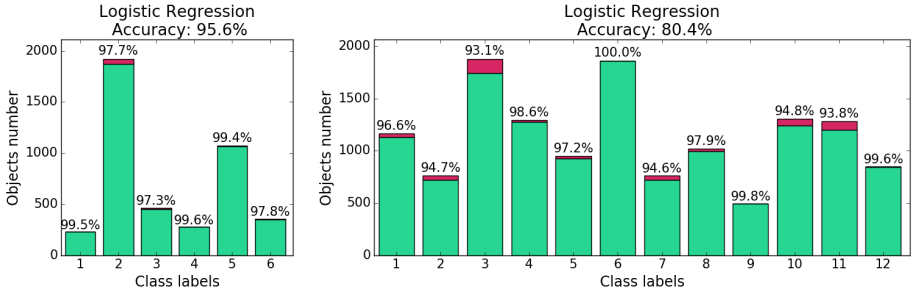
accuracy scores in this case are higher in all cases. All binary accuracy scores for WISDM datasets is larger than 97% for each classification model. These numbers for USC-HAD dataset is larger than 93%.

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. Logistic regression, SVM and random forest are used for classification. The results show that obtained features allows to recover the class label with the high quality.

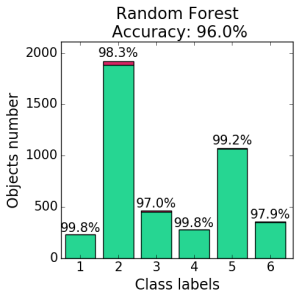
References

1. Budnik, M., Gutierrez-Gomez, E.L., Safadi, B., Pellerin, D., Quénot, G.: Learned features versus engineered features for multimedia indexing. Multimedia Tools and Applications pp. 1–18 (2016)
2. De Boor, C., De Boor, C., Mathématicien, E.U., De Boor, C., De Boor, C.: A practical guide to splines, vol. 27. Springer-Verlag New York (1978)
3. Geurts, P.: Pattern extraction for time series classification. In: European Conference on Principles of Data Mining and Knowledge Discovery. pp. 115–127. Springer (2001)
4. Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. Numerische mathematik 14(5), 403–420 (1970)
5. Hammerla, N.Y., Halloran, S., Ploetz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv preprint arXiv:1604.08880 (2016)

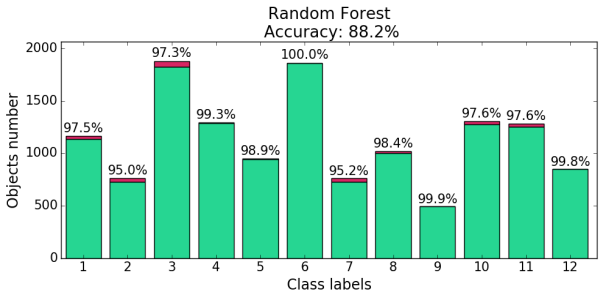


(a) WISDM dataset

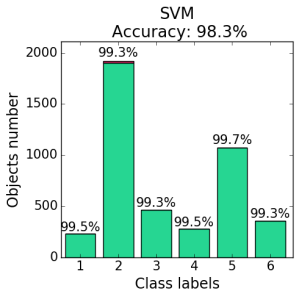
(b) USC-HAD dataset



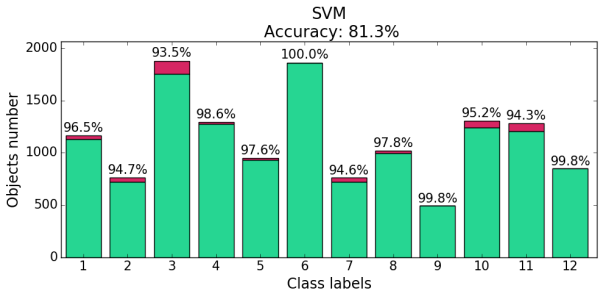
(c) WISDM dataset



(d) USC-HAD dataset



(e) WISDM dataset



(f) USC-HAD dataset

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

6. Hassani, H.: Singular spectrum analysis: Methodology and comparison. Journal of Data Science 5, 239–257 (2007)
7. 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)
8. Incel, O.D., Kose, M., Ersoy, C.: A review and taxonomy of activity recognition on mobile phones. BioNanoScience 3(2), 145–171 (2013)

9. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python. http://www.scipy.org/ (2001–)
10. Karasikov, M., Strijov, V.: Feature-based time-series classification. Intelligence 24(1), 164–181 (2016)
11. Kuznetsov, M., N.P., I.: Time series classification algorithm using combined feature description. Machine Learning and Data Analysis 1(11), 1471–1483 (2015)
12. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. ACM SigKDD Explorations Newsletter 12(2), 74–82 (2011)
13. Lu, L., Qing-ling, C., Yi-Ju, Z.: Activity recognition in smart homes. Multimedia Tools and Applications pp. 1–18 (2016)
14. 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)
15. Lukashin, Y.P.: Adaptive methods of short-term forecasting of time series. M.: Finance and statistics (2003)
16. Umeda, Y.: Time series classification via topological data analysis. Transactions of the Japanese Society for Artificial Intelligence 32(3), D–G72_1 (2017)
17. The usc human activity dataset. http://www-scf.usc.edu/ mizhang/datasets.html
18. 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)
19. The wisdm dataset. http://www.cis.fordham.edu/wisdm/dataset.php