is reasonable because we can't operate with original features as time series might be of different size, not aligned or even multiscaled [9].

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 analyzed separately. In our project we focused mainly on comparing different methods of feature generation [10, 11].

The first approach for feature generation is calculating expertly-defined functions of time series [12]. These functions include average value, standard deviation, mean absolute deviation and distribution for each component. We consider this approach a baseline, as it is the simplest method we use.

We compare baseline with more sophisticated parametric feature generation methods, in which we build approximation models and use their parameters as our final features for classification. In this paper we propose using local spline approximation for feature extraction. In this setup features are knots and parameters of optimal cubic splines approximating our data. We compare this method with other well-known methods for extracting features from time series. One of them is autoregressive model [13]. For each time series we build parametric model and use those parameters as features for classification. Another approach is the model of singular spectrum analysis of time series [14]. We use eigenvalues of trajectory matrix as features for building classifier.

The experiment was conducted on two real accelerometer datasets [15, 16]. We compared the performance of stated feature extraction methods, as well as different classification algorithms. The latter include logistic regression, random forest and SVM.

Problem Statement

Let $\mathcal{S}$ be a 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 $\mathfrak{D} = \{(s_i, y_i)\}_{i=1}^m$ 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 function $\hat{f}$ is equivalent to determining the functions $\hat{\boldsymbol{h}}$ and $\hat{g}$. The function $\hat{\boldsymbol{h}}$ 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 $\hat{\boldsymbol{h}}$ we transform our original sample $\mathfrak{D} = \{s_i, y_i\}_{i=1}^m$ with complex structured objects to the new sample $\mathfrak{D}_H = \{\mathbf{h}_i, y_i\}_{i=1}^m$, where $\mathbf{h}_i = \hat{\boldsymbol{h}}(s_i) \in H$. The function $\hat{g}(\mathbf{h}, \boldsymbol{\theta})$ is defined by its parameter vector $\boldsymbol{\theta} \in \Theta$. The optimal parameters $\hat{\boldsymbol{\theta}}$ are given by

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

where the function $L(\boldsymbol{\theta}, \mathfrak{D}_H, \boldsymbol{\mu})$ 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 [10,11] 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.

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 functions features as the baseline for local approximation models. Expert functions are based on prior knowledge of the original objects. These functions can be expressed as a set of statistics $\{h_i\}_{i=1}^n$, where $h_i : \mathcal{S} \to \mathbb{R}$. The description $\hat{\boldsymbol{h}}(s)$ of the object s is the value of these statistics on the object

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

Given a set of complex objects $\{s_i\}_{i=1}^m$ we extract features in a non-parametric way with a set of expert functions $\{h_j\}_{j=1}^n$. 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 $\hat{\boldsymbol{h}}(s) \in \mathbb{R}^{40}$.

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	Number of points in each histogram bin

Autoregressive model

In this method we assume autoregressive model [13] of the order n as a hypothesis for generation of time series s. Each component of the object s from (1) is a linear combination of the previous n components

$$x_t = w_0 + \sum_{j=1}^{n} w_j x_{t-j} + \varepsilon_t, \tag{3}$$

where ε_t is a random noise. Prediction of the autoregressive model is defined by

$$\hat{x}_t = w_0 + \sum_{j=1}^{n} w_j x_{t-j}. \tag{3}$$

The order of the autoregressive model n is a structural parameter.

The optimal parameters of autoregressive model $\mathbf{w}^* = \{w_j^*\}_{j=0}^n$ for time series s define the feature map $\boldsymbol{h}(s)$. These parameters minimize the squared error between the original object s and its prediction of the model (3)

$$\boldsymbol{h}(s) = \mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^{n+1}} \left(\sum_{t=n+1}^{T} \|x_t - \hat{x}_t\|^2 \right). \tag{4}$$

The problem (4) 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 Figure 1.

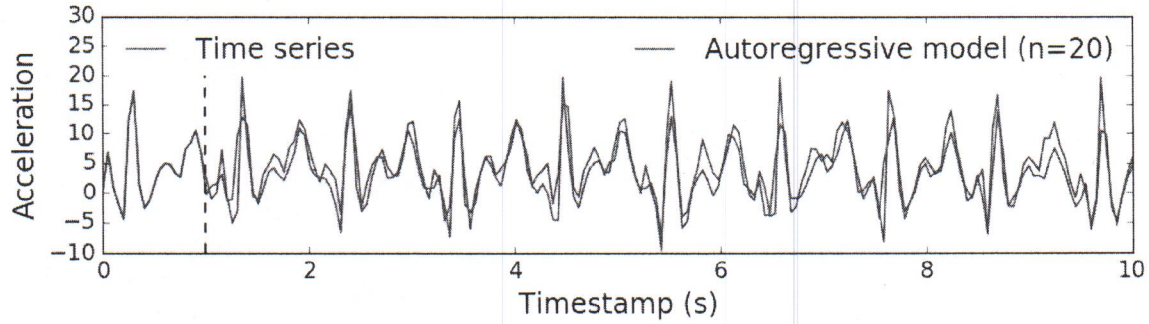


Figure 1: Time series approximation using autoregressive model with order $n = 20$

Singular spectrum decomposition

Alternative hypothesis for generation of time series is SSA (Singular Spectrum Analysis) model [14]. We construct trajectory matrix for each time series s from the original sample $\mathfrak{D}$:

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

Here n is a window width, which is an external structural parameter. The singular decomposition [17] of the matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$:

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

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

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

Splines

Approximation of time series can be done by splines [18]. The 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 $\{\mathbf{w}_\ell\}_{\ell=1}^{M}$.

In order to find optimal spline parameters $\mathbf{w}$, one need to solve system of equations with additional constraints of equality of derivatives up to second order on the edges of intervals. If we denote each spline segment as $p_i(t)$ $i = 1, \ldots, M$ and spline as a whole as $S(t)$, we can write these equations in a following way:

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

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

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

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

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

shows

In the Figure 2 one could find the result of time series approximation given by splines. Compared to the autoregressive model, the splines method gives smoother approximation using almost the same number of parameters.

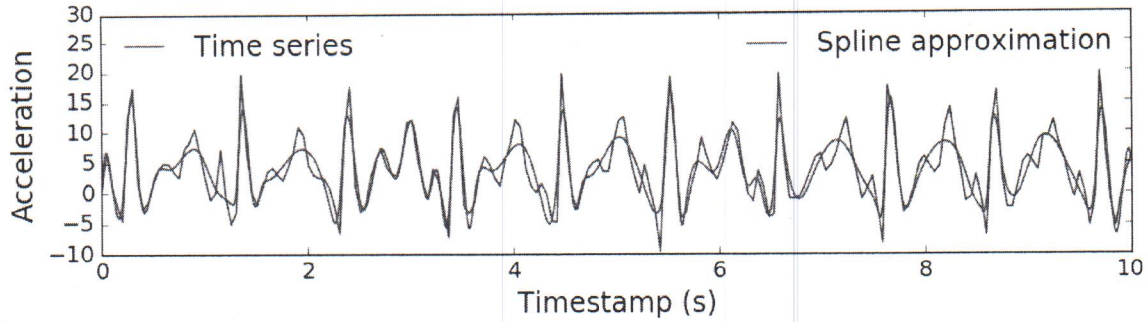


Figure 2: Time series approximation using three order splines

Time series

Classification

Multiclass classification

As we had numerous labels in our datasets we had to choose one of the multi-class approaches to classification. We decided to use one-vs-rest classification as a simple, yet effective approach. The main idea is that we train binary classifiers for each class label and then, on the prediction step, we classify new object according to the most confident classifier. In this section we will describe our approach to classification of time series using newly generated features. We use three different classification models: logistic regression, SVM and random forest.

Classification methods

Logistic regression

The first approach to classification is a regularized logistic regression model. The optimal model parameters (2) is determined by minimising the following error function

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

where

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

Thus,

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

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~~ of SVM model ~~can be formulated in a following way:~~

$$\boldsymbol{\theta}^* = \begin{pmatrix} \boldsymbol{w}^* \\ b^* \\ \boldsymbol{\xi}^* \end{pmatrix} = \arg\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i,$$

$$\text{subject to} \quad y_i(\langle \boldsymbol{w}, \mathbf{h}_i \rangle + b) \geqslant 1 - \xi_i,$$

$$\xi_i \geqslant 0, \quad 1 \leqslant i \leqslant m.$$

The prediction for new object is

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

Random Forest

The random forest ~~is an algorithm, which~~ 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.

Experiment

In this paper we considered two different smart phone based datasets: WISDM [15] and USC-HAD [16]. The smart phone accelerometer measures acceleration along three axis. Sample rate equals 50 ms. The WISDM dataset consists of 4321 objects and each time series belongs to one of the six activities : Standing, Walking, Upstairs, Sitting, Jogging, Downstairs. The USC-HAD dataset contains 13620 objects 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 Table 2. The length of each time series equals 200 which accounts 10 second. In the Figure 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	

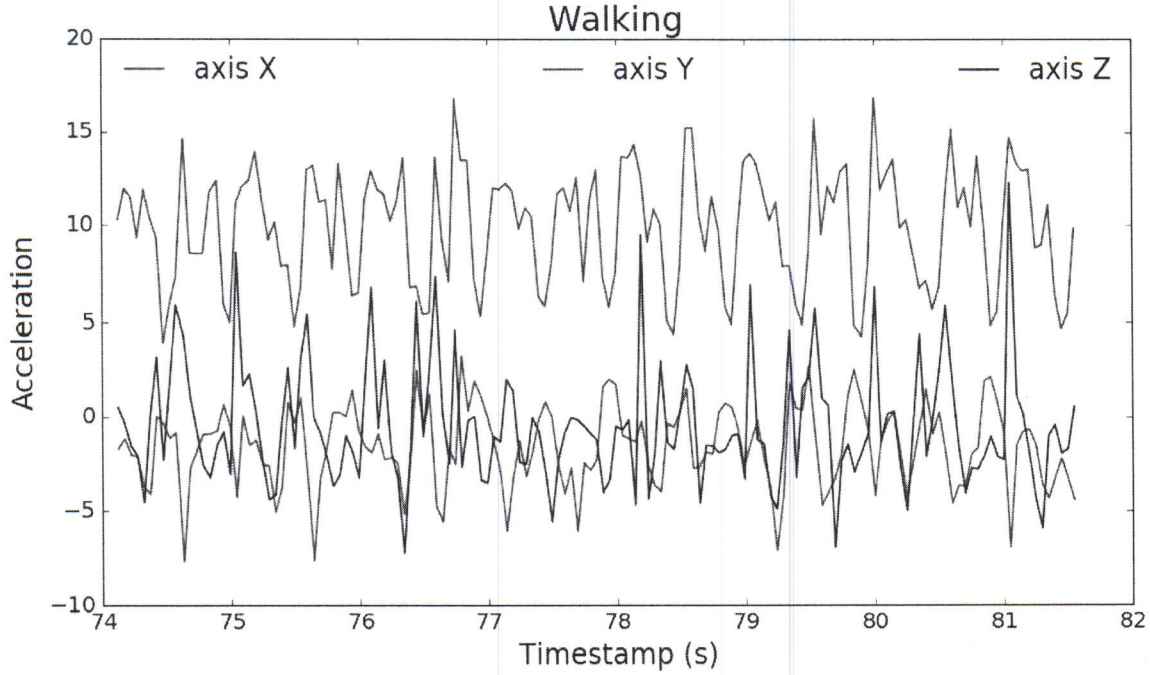


Figure 3: Time series example

For each dataset we applied feature generation approaches described above: expert functions, autoregressive model, SSA, splines. We used three classification model for each generated feature description: logistic regression, support vector machine and random forest. The external structural parameters: the length n for autoregression, the window width n for SSA and the number of splines knots L, were tuned using K-fold cross validation, i.e. minimizing

$$CV(K) = \frac{1}{K}\sum_{k=1}^{K} L(f_k, \mathfrak{D} \setminus \mathfrak{C}_k), \tag{5}$$

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 [18] for time series using *scipy* python library [19]. 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 Figure 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.

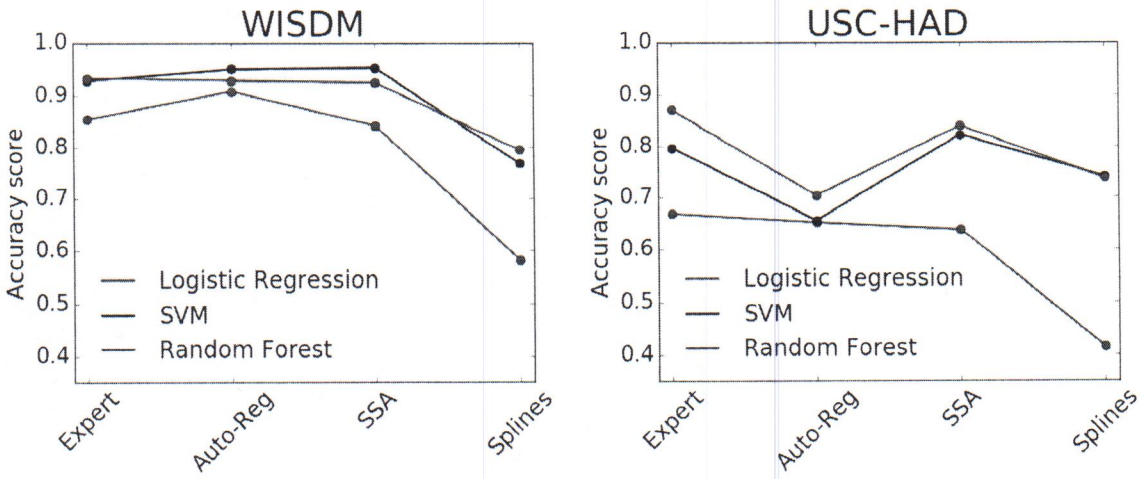


Figure 4: Multiclass accuracy score

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

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

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

Conclusion

The ~~research consider~~ the problem of complex structured objects classification. ~~We investigated~~ the ~~different~~ approaches of feature extraction, particularly the expert functions and ~~data generation hypothesis~~. The experiment on ~~the real~~ data from smart phone accelerometer ~~were carried out~~. ~~We~~ compared ~~different~~ feature descriptions ~~and different classification models~~. The results show that ob-

tained features allows to recover the class label with the high quality.

References

[1] Andrey D Ignatov and Vadim V Strijov. Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. *Multimedia tools and applications*, 75(12):7257–7270, 2016.

[2] Yonggang Lu, Ye Wei, Li Liu, Jun Zhong, Letian Sun, and Ye Liu. Towards unsupervised physical activity recognition using smartphone accelerometers. *Multimedia Tools and Applications*, pages 1–19, 2016.

[3] Wen Wang, Huaping Liu, Lianzhi Yu, and Fuchun Sun. Human activity recognition using smart phone embedded sensors: A linear dynamical systems method. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 1185–1190. IEEE, 2014.

[4] Mateusz Budnik, Efrain-Leonardo Gutierrez-Gomez, Bahjat Safadi, Denis Pellerin, and Georges Quénot. Learned features versus engineered features for multimedia indexing. *Multimedia Tools and Applications*, pages 1–18, 2016.

[5] Lu Lu, Cai Qing-ling, and Zhan Yi-Ju. Activity recognition in smart homes. *Multimedia Tools and Applications*, pages 1–18.

[6] Yuhei Umeda. Time series classification via topological data analysis. *Transactions of the Japanese Society for Artificial Intelligence*, 32(3):D–G72_1, 2017.

[7] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*, 2016.

[8] Ozlem Durmaz Incel, Mustafa Kose, and Cem Ersoy. A review and taxonomy of activity recognition on mobile phones. *BioNanoScience*, 3(2):145–171, 2013.

[9] Pierre Geurts. Pattern extraction for time series classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 115–127. Springer, 2001.

[10] ME Karasikov and VV Strijov. Feature-based time-series classification. *Intelligence*, 24(1):164–181.

[11] M.P. Kuznetsov and Ivkin N.P. Time series classification algorithm using combined feature description. *Machine Learning and Data Analysis*, 1(11):1471–1483.

[12] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[13] Yu P Lukashin. Adaptive methods of short-term forecasting of time series. *M.: Finance and statistics*, 2003.

[14] Hossein Hassani. Singular spectrum analysis: methodology and comparison. 2007.

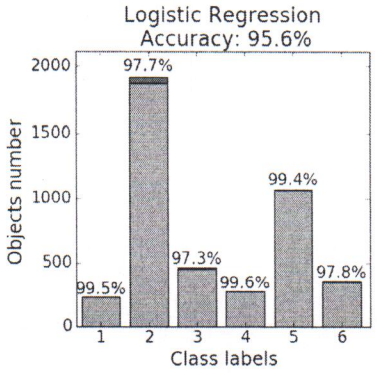
[15] Wisdm dataset. `http://www.cis.fordham.edu/wisdm/dataset.php`.

[16] The usc human activity dataset. `http://www-scf.usc.edu/~mizhang/datasets.html`.

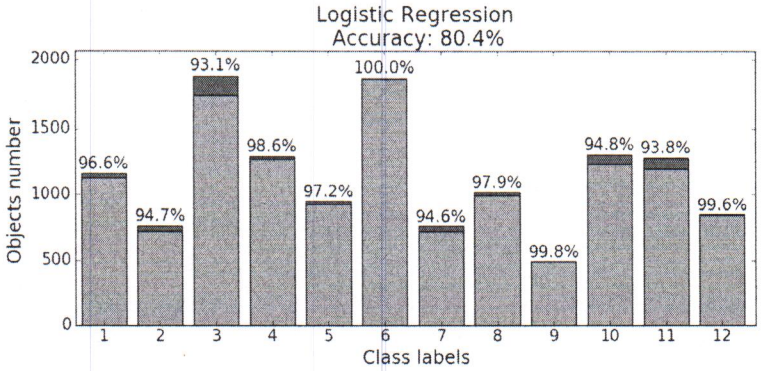
[17] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

[18] Carl De Boor, Carl De Boor, Etats-Unis Mathématicien, Carl De Boor, and Carl De Boor. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978.

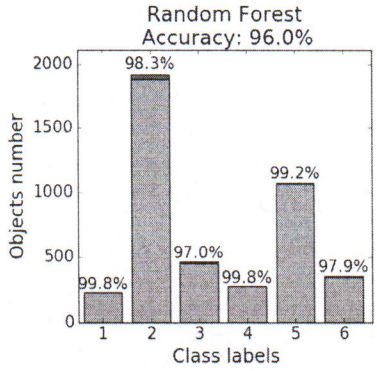
[19] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. `http://www.scipy.org/`, 2001–.



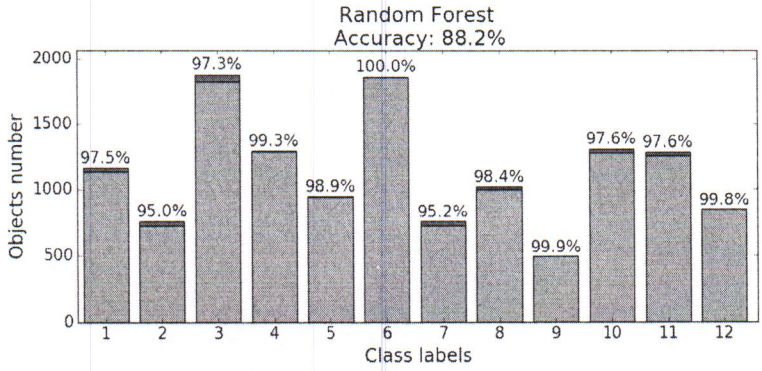
(a) WISDM dataset



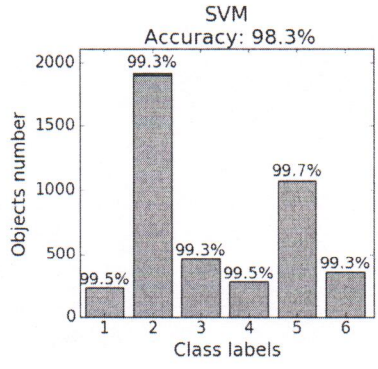
(b) USC-HAD dataset



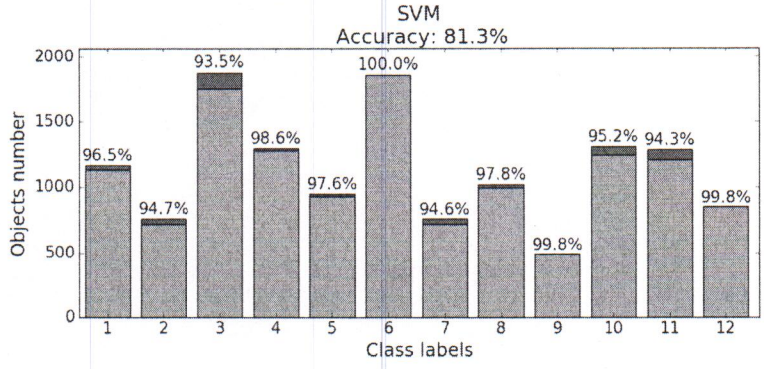
(c) WISDM dataset



(d) USC-HAD dataset



(e) WISDM dataset



(f) USC-HAD dataset

Figure 5: Accuracy scores of classification of each class using all features