# Metamodels for complex structured objects classification

Ilya Zharikov, Roman Isachenko, Artem Bochkarev

## Introduction

This project is dedicated to multiclass classification of complex structured objects (for which we don't have explicit features). The problem arises in many applications such as image recognition, signal processing or time series classification. We will focus on multiclass multivariate time series classification. In this setup time series are regarded as complex structured objects without explicit feature description. This is reasonable because we can't operate with original features as time series might be of different size, not aligned [1] or even multiscaled.

We investigate classification of accelerometer time series [2]. The data is time series of acceleration from three axis, which is sensed by mobile phone or other portable device with accelerometer. The task is to predict the activity a person is performing. List of activities might include walking, running, sitting or walking up/down the stairs.

In general the problem of classifying complex structured objects can be 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 [3, 4]. Extracted features can be later used for building classifiers and feature selection algorithms.

The first approach for feature generation is calculating expertly defined functions of time series [5]. 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. One of them is autoregressive model [6]. For each time series we build parametric model and use those parameters as features for classification. The next approach is the model of singular spectrum analysis of time series [7]. We use eigenvalues of trajectory matrix as features for building classifier. Finally, we consider the parameters of the splines as the feature description of time series.

In the first part of these report we give definitions and make problem statement. In the second part we describe all of the proposed approaches in more detail. Last, we make the experiment on real accelerometer datasets [8, 9], compare all methods and give conclusions and recommendations for practical use.

# Problem Statement

Let $\mathcal{S}$ be a space of complex structured objects, $Y$ is a finite set of class labels. Denote by $\mathfrak{D} = \{(s_i, y_i)\}_{i=1}^m$ a given sample, where $s_i \in \mathcal{S}$ and $y_i \in Y$.

We consider the problem of recovering the function $f : \mathcal{S} \to Y$

$$y = f(s).$$

Let $L(f, \mathfrak{D})$ be an error function which expresses the classification error of the function $f$ over the sample $\mathfrak{D}$. The goal is to determine function $f^*$ which minimizes the error

$$f^* = \arg\min_f L(f, \mathfrak{D}). \tag{1}$$

We assume that the target function $f^*$ belongs to the class of function compositions $f = g \circ \boldsymbol{h}$, where

- $\boldsymbol{h} : \mathcal{S} \to H$ 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 space of class labels $Y$. The function $g$ is parametrized by a vector parameter $\boldsymbol{\theta} \in \Theta$.

The determining of the function $f^*$ is equivalent to determining the functions $\boldsymbol{h}^*$ and $g^*$.

In this paper we consider the following ways of generating the feature space $H$:

- expert functions based on prior knowledge of the original objects. These functions can be expressed as a set of statistics $\{h_i\}_{i=1}^n$, where $h_i : \mathcal{S} \to \mathbb{R}$. Thus, the description $\boldsymbol{h}^*(s)$ of the object $s$ is the value of these statistics on the object

$$\boldsymbol{h}^*(s) = (h_1(s), \ldots, h_n(s)).$$

- hypothesis of data generation. In this case the features are the estimated parameters of the considered hypothesis. Let $S(s, \boldsymbol{h}, \boldsymbol{\lambda})$ be the error function which specifies the hypothesis, e.g. one could define the function $S$ as negative log-likelihood function [10]. The optimal feature map $\boldsymbol{h}^*(s)$ is obtained by

$$\boldsymbol{h}^*(s) = \arg\min_{\boldsymbol{h}} S(s, \boldsymbol{h}, \boldsymbol{\lambda}). \tag{2}$$

  The parameter $\boldsymbol{\lambda}$ is external structural parameter for the function $S$. The equation (2) determines the feature map $\boldsymbol{h}^*$ for each object $s \in \mathcal{S}$.

Given appropriate feature space $H$ and feature map $\boldsymbol{h}$ 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 $\mathbf{h}_i = \boldsymbol{h}(s_i) \in H$. The function $g(\mathbf{h}, \boldsymbol{\theta})$ is defined by its parameter vector $\boldsymbol{\theta} \in \Theta$. The optimal parameters $\boldsymbol{\theta}^*$ are given by

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathfrak{D}_H, \boldsymbol{\mu}), \tag{3}$$

where $L(\cdot, \cdot, \cdot)$ is an analogue of the function (1). Here the vector $\boldsymbol{\mu}$ is a external parameters of the particular classification model.

In our project we consider accelerometer time series as complex structured objects. Time series is represented in the following way:

$$s = (x_1, \ldots, x_T) \in \mathcal{S},$$

where $T$ denotes the length of time series. Now let us expand on different approaches for feature generation.

# Feature generation

### Expert functions

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 $\{h_j\}_{j=1}^n$. For time series, these functions could be mean, standard deviations, mean absolute deviations and distributions of the acceleration. 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.

### Autoregressive model

In this method we assume autoregressive model [6] of the order $n$ as a hypothesis for generation of time series $s$. Each component of the object $s$ is assumed as a linear combination of the previous $n$ components

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

where $\varepsilon_t$ is a random noise. Prediction of the autoregressive model is defined by

$$\hat{x}_t = w_0 + \sum_{j=1}^{n} w_j x_{t-j}. \tag{4}$$

For this method $n$ is a structural parameter and $\boldsymbol{\lambda} = n$.

Feature map $\boldsymbol{h}(s)$ is given by optimal parameters of autoregressive model $\mathbf{w}^* = \{w_j^*\}_{j=0}^n$ for time series $s$. The hypothesis error function (2) in this case is the squared error between the original object $s$ and its prediction of the model (4).

$$\boldsymbol{h}(s) = \mathbf{w}^* = \underset{\mathbf{w}\in\mathbb{R}^{n+1}}{\arg\min}\, S(s, \mathbf{w}, \boldsymbol{\lambda}) = \underset{\mathbf{w}\in\mathbb{R}^{n+1}}{\arg\min} \left( \sum_{t=n+1}^{T} \|x_t - \hat{x}_t\|^2 \right). \tag{5}$$

The problem (5) could be easily converted to the 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 demon-

strated on the Figure 1.

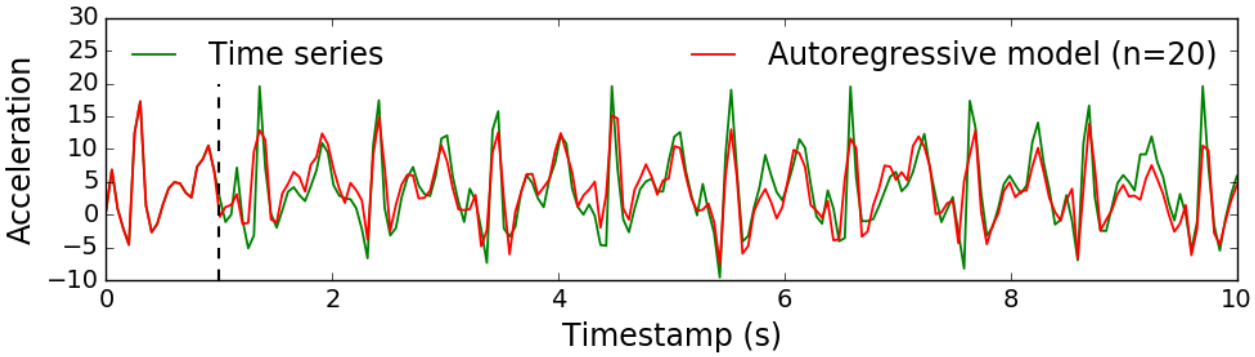



Figure 1: Time series approximation using autoregressive model with $n = 20$

### Singular spectrum analysis

Alternative hypothesis for generation of time series is SSA (Singular Spectrum Analysis) model [7]. We construct trajectory matrix for each time series $s = (x_1, \dots x_T)$:

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

Here $n$, called the window width, is an external structural parameter. Let find the singular decomposition [11] 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 $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ whose entries $\lambda_i$ 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 $s$

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

### Splines

One could approximate the time series by splines [12]. The spline is defined by its parameters:

- $\{\xi_\ell\}_{\ell=1}^{L}$ — the set of knots. To get the adequate result we normalized the knots for each time series.

- $\{\mathbf{w}_\ell\}_{\ell=1}^{L-1}$ — parameters of the models are built on the interval $[\xi_\ell; \xi_{\ell+1}]$. The dimension of the each parameter vector $\mathbf{w}_\ell$ depends on the spline order.

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

$$\boldsymbol{h}(s) = (\xi_1, \ldots, \xi_L, \mathbf{w}_1, \ldots, \mathbf{w}_{L-1}).$$

This approach gives another approximation of the time series. 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.

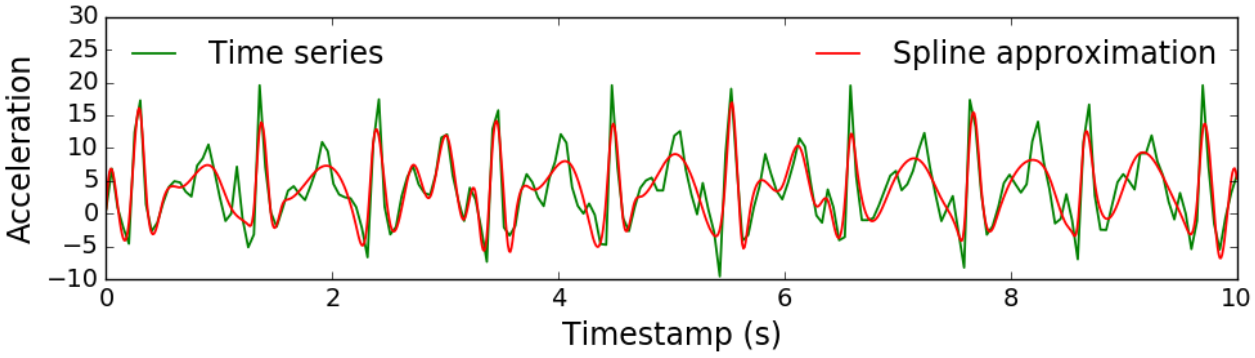



Figure 2: Time series approximation using splines

### Shape invariant model

All time series are shifted and have different magnitudes in time and acceleration. In order to find the same structure in every time series one could apply self model regression approach link. We assume that our time series $s$ is a function of time variable $t$ and this function can be represented as the shape invariant model

$$s(t) = \theta_0 + \theta_1 g\left(\frac{t - \theta_2}{\theta_4}\right).$$

The function $g(\cdot)$ represents the general characteristic shape of the time series. The coefficients $\theta_0, \theta_2$ are the corresponding shifts along acceleration and time

respectively. $\theta_1, \theta_4$ scale time series on acceleration and time axis. These four parameters are different for each time series, while the function $g$ is the same.

Self modelling regression is a problem of determining the optimal shape function $g$ and the coefficients $\{\theta_j\}_{j=1}^4$ for each time series.

The simplest way to solve this problem is to use one of the given time series as the function $g$. For example one could assume, $g(t) = s_1(t)$ and our model has the form

$$s(t) = \theta_0 + \theta_1 s_1\left(\frac{t - \theta_2}{\theta_4}\right).$$

More sophisticated way to solve the problem is to fix the family $G$ of the possible functions $g$. For example, we can find the best first order spline with fixed knots $(t_1, \ldots, t_L)$. Each this spline is determined by the values at the knots $(g_1, \ldots, g_L)$. Actually the spline is given by connecting the neighbouring points by straight lines. If we choose the spline knots as times given in the sample, we can formulate the iterative process.

Firstly, we initialize the vectors $\boldsymbol{\theta}_i \in \mathbb{R}^4$ for each time series $s_i$. Then we solve the linear regression problem

$$\sum_{i=1}^{L} (x_i - s(t_i))^2 \to \min_g .$$

The objective function depends on the function $g$ linearly. The output of this step is the optimal vector $(g_1, \ldots, g_T)$.

Then we want to update the parameters vectors $\boldsymbol{\theta}_i$. We minimize the same functional, but it depends on $\boldsymbol{\theta}_i$ non-linearly. To solve this problem we use non-linear regression using Newton-Raphson method.

These two steps repeat until convergence. The stopping criteria is the insignificant changes of the function $g$ between iterations.

# Classification

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.

### Quality measure

We consider the accuracy score as our main quality measure function. This choice is based on our wish to compare our results with previous articles [3, 4] and this measure is easy to interpret. Accuracy score is a relation correctly labeled objects and the total amount of objects in dataset:

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

where $\hat{y}_i$ is a prediction of the classifier. We calculated this measure for binary case, when we predict each class in particular, and in multiclass case.

### 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.

### Logistic regression

The first approach to classification we played with was regularized logistic regression model. The optimal model parameters (3) is determined by minimising the following error function

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

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

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

$$\hat{y} = g(\mathbf{h}, \boldsymbol{\theta}^*) = \text{sgn}\langle\boldsymbol{\theta}^*, \mathbf{h}\rangle$$

**SVM**

We also used binary SVM model. The problem in this case can be formulated in a following way:

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

For new objects we can make a prediction

$$\hat{y} = \text{sgn}(\hat{\mathbf{w}}^T\mathbf{h} + b)$$

**Random Forest**

Random forest is an algorithm which exploits the idea of bagging. This is an approach of building many random weak 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 single trees:

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

where $B$ is an amount of trees used for bagging.

# Experiment

In this paper we consider two different smart phone based datasets: WISDM [8] and USC-HAD [9]. Data from smart phone accelerometer consists of information about acceleration along each of three axis. Time difference between measure-

ments 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 1. 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 1: Activities distributions

(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 we applied the feature generation approaches described above: expert functions, autoregressive model, SSA, splines. We used three different widely used classification model for each generated feature description: logistic regression, support vector machine and random forest. The external structural parameters $\boldsymbol{\lambda}$ for feature generation procedures, such as the length $n$ for autoregression, the window width $n$ for SSA and the number of splines knots $L$, were tuned using 3-fold cross validation procedure. The hyperparameters $\boldsymbol{\mu}$ for classification models were also tuned using the same cross validation procedure.

In paper [5] the authors proposed to use the following expert functions for time series classification:

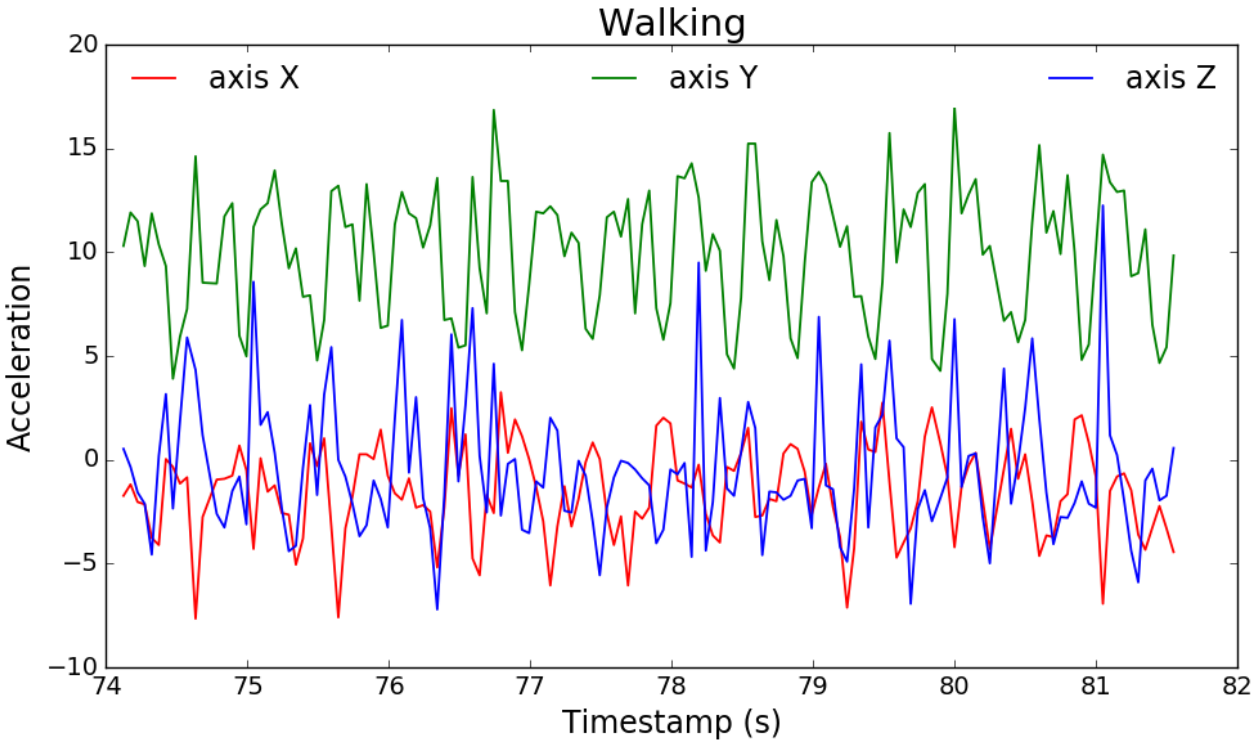



Figure 3: Time series example

- (3) average acceleration for each axis;
- (3) standard deviation for each axis;
- (3) average absolute difference for each axis;
- (1) average resultant acceleration;
- (30) values of histogram with 10 bins for each axis.

Hence feature extraction procedure gives us the feature description of time series $\boldsymbol{h}(s) \in \mathbb{R}^4 0$.

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

Singular spectrum analysis were tuned in the same way to find the optimal window width $n$. Analogously to autoregressive model the cross validation procedure gives the same value $n = 20$. Authors assume that it means that time series has memory of this size.

We fit splines for time series using *scipy* python library. This software fits 3-order B-splines [12]. The knots $\{\xi_\ell\}_{\ell=1}^{L}$ for splines were distributed uniformly.

The number $L$ were chosen implicitly by choosing the proper smoothing parameter $s$. The less the value of $s$, the larger the number of knots $L$. The fitting was constructed as follows. Firstly, there was the initialization step to find appropriate bounds for smoothing parameter. The next step is finding the smoothing parameter in this interval using bi-search approach.

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 is obtained by splines parameters. 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 are strongly correlated.

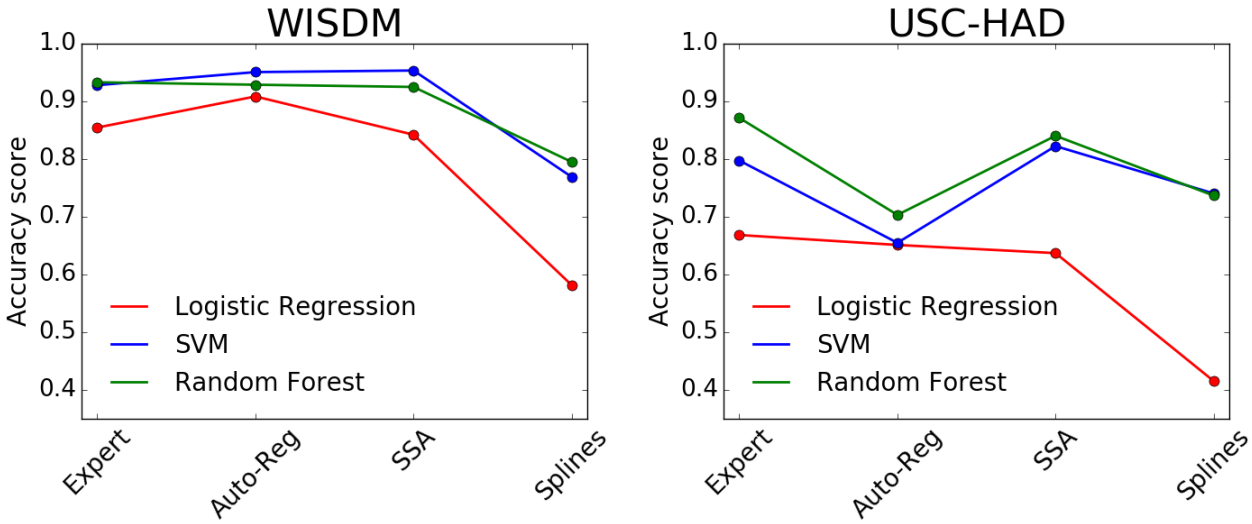



Figure 4: Multiclass accuracy score

All results with classification accuracy scores for each class are represented in Table 2 and Table 3. 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 2: 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 3: 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 1 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 problem of complex structured objects classification were considered. 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 obtained features allows to recover the class label with the high quality.

# References


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

[2] 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*, 1185–1190. 2014.

[3] Karasikov M.E. and Strijov V.V. Feature-based time-series classification. *Intelligence*, 24(1):164–181. 2016.

[4] Kuznetsov M.P. and Ivkin N.P. Time series classification algorithm using combined feature description. *Journal of Machine Learning and Data Analysis*, 1(11):1471–1483. 2015.

[5] 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.

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

[7] Hossein Hassani. Singular Spectrum Analysis: Methodology and Comparison. *Journal of Data Science*, 5(2):239–257. 2007.

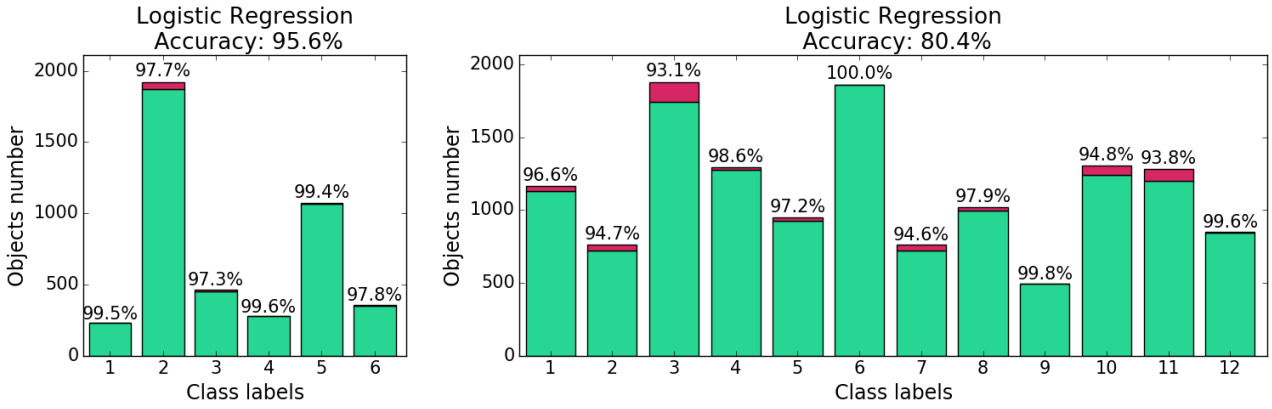
[8] The WISDM (Wireless Sensor Data Mining) dataset. http://www.cis.fordham.edu/wisdm/dataset.php.

[9] The USC-HAD (University of Southern California Human Activity Dataset). http://www-scf.usc.edu/~mizhang/datasets.html.

[10] Bishop, Christopher M. Pattern recognition. *Machine Learning*. 2006.

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

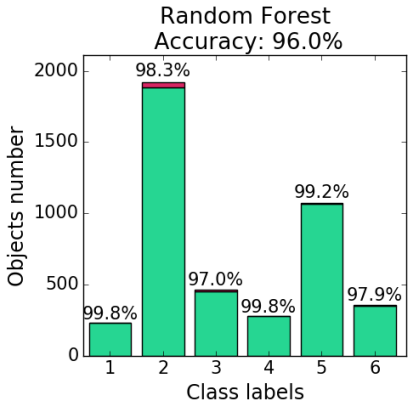
[12] De Boor C. et al. A practical guide to splines. *Springer-Verlag*. 1978.



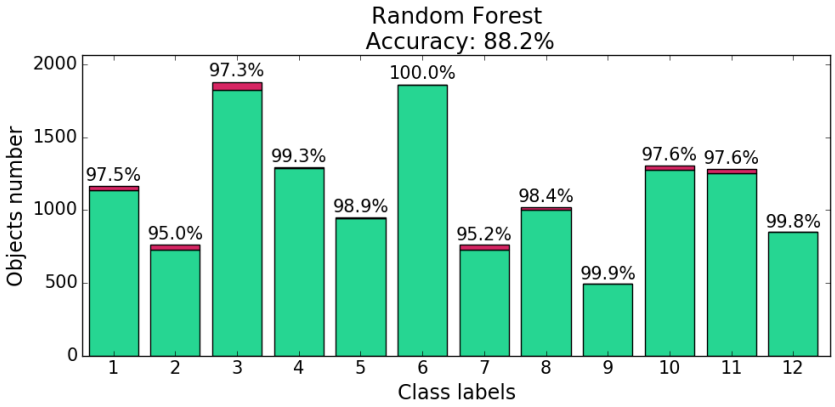

(a) WISDM dataset



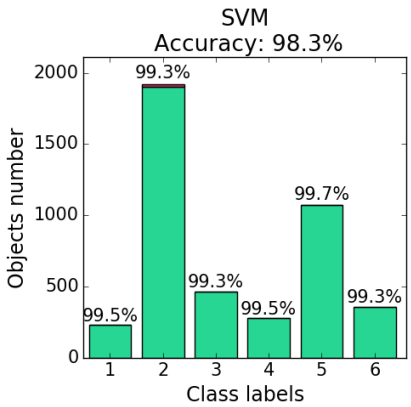

(b) USC-HAD dataset



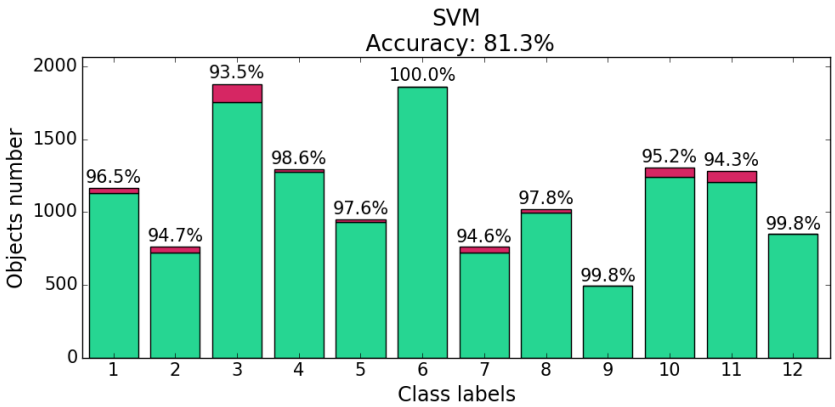

(c) WISDM dataset




(d) USC-HAD dataset




(e) WISDM dataset




(f) USC-HAD dataset

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