

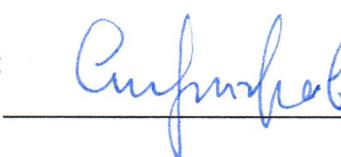
Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Московский физико-технический институт  
(государственный университет)»  
Физтех-школа прикладной математики и информатики  
Факультет управления и прикладной математики  
Кафедра «Интеллектуальные системы»

## **ВЫБОР МОДЕЛЕЙ ЛОКАЛЬНОЙ АППРОКСИМАЦИИ ДЛЯ КЛАССИФИКАЦИИ ВРЕМЕННЫХ РЯДОВ**

Выпускная квалификационная работа  
(бакалаврская работа)

Направление подготовки: 03.03.01 Прикладные математика и физика

Выполнил:  
студент 477 группы  Иванычев Сергей Дмитриевич

Научный руководитель:  
д.ф.-м.н.  Стрижов Вадим Викторович

Москва 2018

## Содержание

<b>1 Введение</b>	<b>4</b>
<b>2 Обзор литературы</b>	<b>6</b>
<b>3 Постановка задачи прогнозирования</b>	<b>7</b>
3.1 Комбинированное признаковое описание . . . . .	8
<b>4 Построение признакового пространства</b>	<b>10</b>
4.1 Локально-аппроксимирующие модели . . . . .	10
4.1.1 SEMOR . . . . .	10
4.1.2 AR-авторегрессия . . . . .	11
4.1.3 Сингулярный спектр . . . . .	11
4.1.4 Фурье-модель . . . . .	12
4.2 Построение пространства $Z$ . . . . .	12
<b>5 Алгоритм многоклассовой классификации</b>	<b>13</b>
5.1 Выбор моделей локальной аппроксимации . . . . .	13
<b>6 Генерация данных</b>	<b>15</b>
6.1 Сегментация данных . . . . .	15
6.2 Нормализация сегментов . . . . .	17
<b>7 Вычислительный эксперимент</b>	<b>19</b>
7.1 Анализ статистических свойств выборки . . . . .	19
7.2 Сложность модели . . . . .	21
7.3 Сравнение сложностей универсальной модели и GLE . . . . .	22
7.4 Универсальная модель. . . . .	23
<b>8 Выводы</b>	<b>25</b>

### Аннотация

**Ключевые слова:** *прогнозирование временных рядов; объекты сложной структуры.*

## 1 Введение

В работе рассматривается задача классификации временных рядов в задаче распознавания действий человека по временным рядам, порождаемым датчиками носимых устройств, например, с акселерометра, гироскопа, альтиметра смартфонов или умных браслетов. Существует несколько подходов к классификации временных рядов: среди них можно выделить машины опорных векторов [1, 2], рекурентные [3] и глубокие [4] нейронные сети или решающие деревья. Классификация временных рядов является частным случаем классификации объектов сложной структуры. Из-за того, что подобные задачи возникают во многих областях, например, в обработке сигналов, биологии, финансах, метеорологии, существует довольно много техник ее решения.

В нашей работе нас интересует решение задачи классификации временных рядов путем построения промежуточного признакового пространства [5]. Этот метод применим не только к задаче классификации рядов с носимых устройств, так как к объектам сложной структуры можно свести соответствующие ряды из других задач. В общем случае подход с промежуточным признаковым пространством разделим на два этапа.

- На первом этапе для сегментов временных рядов, которые выступают в роли объектов (которые, вообще говоря, могут быть различной длины и даже частоты дискретизации) вычисляются некоторые статистики или добываются некоторые экспертные оценки. В результате на каждый объект мы имеем некоторый набор численных показателей.
- Над вторичным пространством этих показателей (то есть преобразованными объектами) работает некоторый алгоритм классификации (например ...), который обучается на "вторичной" выборке.

Эти этапы зависят, так как классификатор, используемый во втором этапе может потребовать от обучающей выборки выполнимость некоторых гипотез и, в частности, гипотезы простоты выборки, что может быть обеспечено только корректным первым этапом. Выполнимость гипотезы простоты выборки, находящейся в промежуточном пространстве, необходима для корректной работы алгоритмов классификации.

В нашей работе мы рассматриваем при каких условиях отображение объектов сложной структуры порождает *простую* выборку, то есть случайную, однородную и независимую, а также предлагаем пути построения соответствующей выборки.

## 2 Обзор литературы

Алгоритм классификации временных рядов по их признаковому описанию а также базовые подходы к генерации признаковых описаний в задаче определения типа движения рассмотрены в работе [5]. Альтернативный подход к генерации признаковых описаний, основанный на описании сегментов оптимальными параметрами аппроксимирующих моделей описан в работе [6].

### 3 Постановка задачи прогнозирования

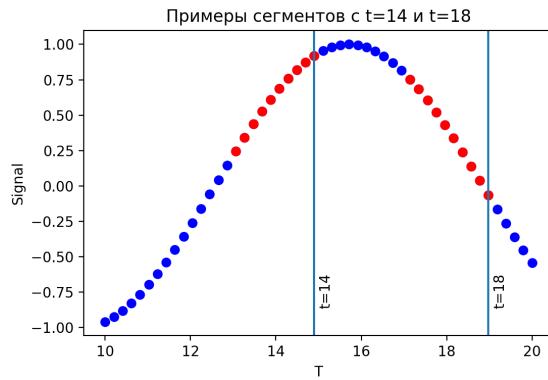
Рассмотрим некоторый временной ряд, то есть функцию определенную на множестве временных меток.

$$S : T \rightarrow \mathbb{R} \text{ где } T = \{t_0, t_0 + d, t_0 + 2d, \dots\}, |T| < \infty$$

Зададим некоторую ширину сегмента  $n \in \mathbb{N}$ , тогда объектом  $s_i$  мы назовем набор

$$\mathbf{x}_i = [S(t_i), S(t_i - d), S(t_i - 2d), \dots, S(t_i - (n-1)d)]^\top, \quad \mathbf{x}_i \in X \equiv \mathbb{R}^n$$

Рис. 1: Пример сегментов временного ряда с  $n = 10$ .



Необходимо восстановить зависимость  $y = f(\mathbf{x})$ ,  $f : X \rightarrow \{1, 2, \dots, K\}$ . Для этого задана обучающая выборка

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^l, \quad y_i \in \{1, 2, \dots, K\}$$

Где  $\{\mathbf{x}_i\}$  — набор сегментов данных акселерометра,  $\{y_i\}$  — метки классов движения, а также функция потерь (multinomial log-loss)

$$L(f(x_i), y) = \sum_{i=1}^l \sum_{k=1}^K [y_i = k] \log P(y_i = k | x_i, \theta) \tag{3.1}$$

Таким образом мы решаем задачу оптимизации

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \sum_{i=1}^l L(f(x_i), y_i)$$

### 3.1 Комбинированное признаковое описание

Пусть  $H$  — множество функций вида  $h : X \rightarrow \mathbb{R}^m$ , где  $m = m(g)$ , то есть это множество отображений пространства объектов сложной структуры в пространство действительных чисел некоторой размерности (для каждой функции размерность может быть своя). В  $H$  могут лежать например

- Множество моделей локальной аппроксимации сигнала
- Множество статистик
- Множество экспертных оценок каждого из сложных объектов

Возьмем конечный поднабор этих функций

$$\mathbf{h} = [\mathbf{h}_1 \dots \mathbf{h}_k], \{\mathbf{h}_1 \dots \mathbf{h}_k\} \subset H$$

Обозначим сумму размерностей образов функций из набора как

$$n_{\mathbf{h}} \triangleq \dim(\text{Im}(h_1)) + \dim(\text{Im}(h_2)) + \dots + \dim(\text{Im}(h_k))$$

Тогда  $\mathbf{h}$  индуцирует отображение  $\mathbf{h} : X \rightarrow Z \subset \mathbb{R}^{n_g}$ , причем в векторах образа первые  $\dim(\text{Im}(h_1))$  компонент соответствуют образу  $\mathbf{h}_1$ , следующие  $\dim(\text{Im}(h_2))$  соответствуют  $\mathbf{h}_2$  и так далее.  $Z$  называется *признаковым пространством* объектов сложной структуры  $X$ . Тогда, мы можем искать  $f$  в семействе суперпозиций  $a(\mathbf{h}(\cdot), \gamma)$ , где

$$T \rightarrow \mathbf{X} \xrightarrow{\mathbf{h}} \mathbf{Z} \xrightarrow{a} Y$$

- $\mathbf{h}$  — это признаковое отображение
- $a(\cdot, \gamma)$  — параметрическое отображение  $Z$  в  $\{1, 2, \dots, K\}$ , которое соответствует некоторому алгоритму машинного обучения, параметризованного вектором гиперпараметров  $\gamma$ .

В таком подходе функция потерь теперь определена на отображении  $a$ , то есть

$$L(f(\mathbf{x}), y) = L(a(g(\mathbf{x}), \gamma), y)$$

Итак, задача решается следующим образом:

- Поиск и вычисление отображения  $Z = \{\mathbf{h}(x_i)\}_{i=1}^l$  путем минимизации функционалов ошибок для каждой модели локальной аппроксимации и поиска оптимальных параметров

$$\arg \min_{\mathbf{w} \in W} L_g(\mathbf{X}, \mathbf{w}) = \arg \min_{\mathbf{w} \in W} \sum_{i=1}^l \sum_{k=1}^n \|g(\mathbf{w}, \mathbf{x}_i) - \mathbf{x}_i\|_2^2$$

- Оптимизация функции ошибки обобщенной линейной модели, которая в качестве обучающей выборки имеет  $(\mathbf{Z}, y)$ .

$$\arg \min_{\theta \in \Theta} L_a(\mathbf{Z}, \mathbf{y}, \theta) = \arg \min_{\theta \in \Theta} \left[ - \sum_{i=1}^l \sum_{k=1}^K [y_i = k] \log P(y_i = k | \mathbf{z}_i, \theta) \right]$$

Основное допущение, принимаемое в данном методе является допущение о том, что выборка в признаковом пространстве объектов является простой. В данной работе мы рассматриваем, для каких признаковых пространств это допущение справедливо, а также предлагаем способы построения таких выборок.

## 4 Построение признакового пространства

В качестве элементов из пространства функций  $H$  мы будем использовать локально-аппроксимирующие модели сегментов.

$$g(w, x) \in X, \text{ где } w \in \mathbb{R}^{n_g}$$

Тогда параметры настроенной модели будут являться образом отображения  $h$

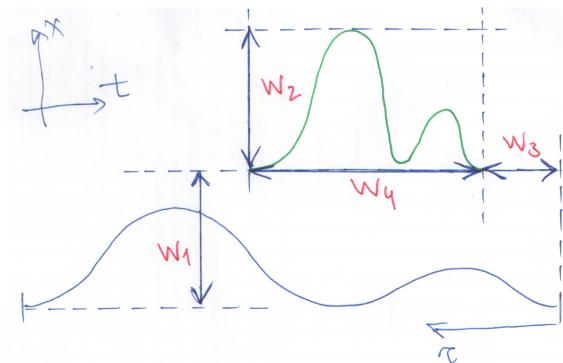
$$h(x) = \arg \min_{w \in \mathbb{R}^{n_g}} \rho(g(w, x), x)$$

### 4.1 Локально-аппроксимирующие модели

#### 4.1.1 SEMOR

Пусть наряду с сегментом  $x$  мы имеем некоторый временной ряд  $p$ , который содержит меньшее число элементов. Модель предполагает, что изгиб временного ряда  $p$  повторяет форму  $x$ , поэтому мы аппроксимируем временной ряд  $x$  сегментом  $p$ .

Рис. 2: Иллюстрация параметров метода SEMOR.



$$w = [w_1, w_2, w_3, w_4]$$

Модель Self-Modeling Regression описывается следующим выражением

$$g(x, w) = w_1 + w_2 p(w_3 + w_4 t)$$

Где в общем случае  $p$  — функция формы, однако в частном случае это может быть заранее заданный временной ряд. Параметры  $w_1, w_2$  находятся шагом двухпараметрического поиска.

раметрической линейной регрессии,  $w_3, w_4$  получаются минимизацией DTW, где из наклона пути, результирующего в minimum edit distance мы получаем  $w_4$ , а  $w_3$  — это смещение этой прямой относительно нуля. Следуя приведенной схеме мы получаем следующие оптимальные веса

$$w_{\text{SE MOR}} = [\hat{w}_1, \hat{w}_2, \hat{w}_3, \hat{w}_4, \rho]$$

#### 4.1.2 AR-авторегрессия

Является моделью авторегрессии порядка  $m$ .

$$g_{\text{AR}}(w, x) = [\hat{x}_1 \dots \hat{x}_n], \text{ где } \hat{x}_i = \begin{cases} x_k & \text{при } k \in [1, m] \\ w_0 + \sum_{i=1}^m w_i x_{k-i} & \text{при } k \in [m+1, n] \end{cases}$$

Оптимальные веса  $w$  определяются минимизацией некоторой функции ошибки, например MSE

$$w_{\text{AR}} = \arg \min_{w \in \mathbb{R}^m} \sum_{i=1}^n \|x_i - \hat{x}_i\|^2$$

#### 4.1.3 Сингулярный спектр

С помощью эмбеддинга построим траекторную матрицу  $S$  для сегмента  $x \in \mathbb{R}^n$

$$S = \begin{pmatrix} x_1 & x_2 & \dots & x_m \\ x_2 & x_3 & \dots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-m+1} & x_{n-m+2} & \dots & x_n \end{pmatrix}$$

Каждая следующая строка траекторной матрицы  $S$  получается сдвиганием окна длины  $m$  в сегменте на единицу.  $m$  — единственный параметр SSE. Применяем сингулярное разложение к матрице  $S^\top S$

$$S^\top S = V H V^\top$$

Где  $H = \text{diag}(\lambda_1 \dots \lambda_m)$  — диагональная матрица собственных значений. Вектор, образованный этими собственными значениями используется в качестве признакового описания сегмента  $x$ .

$$w_{\text{SSA}} = [\lambda_1 \dots \lambda_m]$$

#### 4.1.4 Фурье-модель

В качестве аппроксимирующей модели сегмента берем обратное дискретное преобразование Фурье, то есть признаковым описанием сегмента является прямое дискретное преобразование Фурье.

$$\begin{aligned} w_{2j} &= \operatorname{Re} \sum_{k=1}^n x_k \exp \left( -\frac{2\pi i}{n} k j \right), j = 1 \dots n \\ w_{2j+1} &= \operatorname{Im} \sum_{k=1}^n x_k \exp \left( -\frac{2\pi i}{n} k j \right), j = 1 \dots n \\ w_{\text{FFT}} &= [w_1 \dots w_{2n}] \in \mathbb{R}^{2n} \end{aligned}$$

В качестве признакового описания берем  $m$  частот из прямого преобразования Фурье с максимальной амплитудой, где  $m$  — параметр модели.

## 4.2 Построение пространства $Z$

Итак, набор  $g$  состоит из локально аппроксимирующих моделей, каждая из которых отображает сегмент  $x$  в оптимальные параметры модели. Имеем следующее отображение

$$[\mathbf{h}_1 \dots \mathbf{h}_k] : x \mapsto [\mathbf{w}_1^* \dots \mathbf{w}_k^*]$$

Взятые в совокупности эти отображения порождают промежуточное признаковое пространство  $Z$ , элементы которого являются конкатенацией векторов  $h_1(x), h_2(x) \dots h_k(x)$ .

$$g : \mathbf{X} \rightarrow \mathbf{Z}$$

## 5 Алгоритм многоклассовой классификации

Алгоритм обучения выглядит следующим образом. После преобразования пространства  $\mathbf{X}$  с  $l$  объектами в промежуточное пространство  $\mathbf{Z}$  с помощью  $l_h$  моделей локальной аппроксимации  $\mathbf{h}$ , мы применяем для многоклассовой классификации некий алгоритм машинного обучения. В нашем случае лучше всего подходит (для определения вероятностей принадлежности классу) многомерная логистическая регрессия.

---

### Алгоритм 5.1 Алгоритм обучения

---

**Вход:**  $\mathbf{X}, \mathbf{Y}, \mathbf{h}, l, l_h$ ;

**Выход:**  $\hat{\theta}$ ;

- 1: инициализировать  $\mathbf{u} := \mathbf{y}_1$  (вектор матрицы  $\mathbf{Y}$ )
  - 2:  $\mathbf{z} = []$
  - 3: **для**  $i = 1, \dots, l_h$   
     $\mathbf{z}[i] := \mathbf{h}[i](\mathbf{X})$
  - 4: инициализировать  $\mathbf{Z} = \mathbf{z}[1]$
  - 5: **для**  $i = 2, \dots, l_h$   
     $\mathbf{Z} := (\mathbf{Z}, \mathbf{z}[i])$
  - 6:  $\hat{\theta} = \arg \min(L(f(Z), Y))$
- 

Применение алгоритма к новым данным выглядит аналогично. Объекты переносятся в промежуточное пространство с помощью тех же моделей  $\mathbf{h}$ , а затем считаются вероятности принадлежности объектов из  $\mathbf{Z}$  каждому из классов с помощью оптимального вектора параметров  $\hat{\theta}$  из обученной логистической регрессии.

### 5.1 Выбор моделей локальной аппроксимации

Так как в классе моделей локальной аппроксимации могут присутствовать модели, образы которых от одинаковых объектов будут скоррелированы, качество классификации будет различаться от совокупности моделей в  $\mathbf{h}$ . Поэтому строит задача максимизации качества на скользящем контроле. Базовым алгоритмом для выбора промежуточного пространства параметров является минимизация мультиномиального log-loss на кросс-валидации.

---

**Алгоритм 5.2** Применение алгоритма

---

**Вход:**  $\mathbf{X}, \mathbf{h}, l, l_{\mathbf{h}}, \hat{\theta}$ ;

**Выход:**  $\hat{\mathbf{y}}$ ;

- 1: инициализировать  $\mathbf{u} := \mathbf{y}_1$  (вектор матрицы  $\mathbf{Y}$ )
  - 2:  $\mathbf{z} = []$
  - 3: **для**  $i = 1, \dots, l_{\mathbf{h}}$
  - 4:    $\mathbf{z}[i] := \mathbf{h}[i](\mathbf{X})$
  - 5: инициализировать  $\mathbf{Z} = \mathbf{z}[1]$
  - 6: **для**  $i = 2, \dots, l_{\mathbf{h}}$
  - 7:    $\mathbf{Z} := (\mathbf{Z}, \mathbf{z}[i])$
  - 8: инициализировать  $\hat{\mathbf{y}}$
  - 9: **для**  $i = 1, \dots, l$
  - 10:    $\hat{\mathbf{y}}[i] = \arg \max_{k \in K} P(f(\mathbf{X}[i]) = y_k | \hat{\theta})$
- 

$$CV(\mu, \mathbf{X}, \mathbf{Y}) = \frac{1}{|N|} \sum_{n \in N} Q_\mu(X_n^l, X_n^k)$$

Для отбора моделей локальной аппроксимации мы будем по очереди включать и выключать модель из вектора функций  $\mathbf{h}$ . То есть мы решаем задачу максимизации функционала качества на множестве подмножеств пространства моделей локальной аппроксимации.

$$CV(\mathbf{h}, \mathbf{X}, \mathbf{Y}) \rightarrow \max_{\mathbf{h} \subset \mathbf{h}_{\text{all}}}$$

Где под  $\mathbf{h}_{\text{all}}$  понимается вся совокупность моделей локальной аппроксимации, которые у нас есть. Если количество моделей не очень большое, то подойдет такой базовый алгоритм.

---

**Алгоритм 5.3** Отбор моделей локальной аппроксимации

---

**Вход:**  $\mathbf{X}, \mathbf{Y}, \mathbf{h}_{\text{all}}, l, l_{\mathbf{h}}, N$ ;

**Выход:**  $\mathbf{h}_{\text{best}}$ ;

```
1:  $\mathbf{h}_{\text{best}} := 0$ 
2:  $S_{\text{best}} := -\infty$ 
3: для  $\mathbf{h}$  в Combinations( $\mathbf{h}_{\text{all}}$ )
4:    $\mathbf{s} = []$ 
5:   для  $n = 1, \dots, N$ 
6:      $\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{test}} = \text{Split}(\mathbf{X}, n)$ 
7:      $\mathbf{Y}_{\text{train}}, \mathbf{Y}_{\text{test}} = \text{Split}(\mathbf{Y}, n)$ 
8:      $\theta = \text{Learn}(\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}}, \mathbf{h})$ 
9:      $\mathbf{Y}_{\text{predicted}} = \text{Predict}(\mathbf{X}_{\text{test}}, \mathbf{h}, \theta)$ 
10:     $\mathbf{s}[i] = \text{LogLoss}(\mathbf{Y}_{\text{predicted}}, \mathbf{Y}_{\text{test}})$ 
11:     $S_{\text{mean}} = \frac{1}{|N|} \sum \mathbf{s}[i]$ 
12:    если  $S_{\text{best}} < S_{\text{mean}}$  то
13:       $S_{\text{best}} := S_{\text{mean}}$ 
14:       $\mathbf{h}_{\text{best}} := \mathbf{h}$ 
```

---

## 6 Генерация данных

Для проведения вычислительного эксперимента были собраны данные с линейного акселерометра на мобильном телефоне Galaxy S7 с помощью приложения Physics Toolbox Suite. Данное приложение записывает показания акселерометра по трем осям с частотой 100 Hz, а также суммарную амплитуду. Экспериментатор по очереди ходил, бегал, ходил вверх и вниз по лестнице, в результате чего мы получили показания акселерометра по четырем видам активности.

Сырые данные, использованные для эксперимента доступны здесь.

### 6.1 Сегментация данных

Хотя для автоматической сегментации данных есть более научные подходы, в нашем случае мы используем ad-hoc подход, основанный на том, что во время проведения эксперимента частота действий в среднем была постоянной.

Зная временной ряд снятого действия  $S(t)$  и среднюю длительность этого действия этого действия в тиках  $F$ , мы в ряде находим индексы локальных максимумов

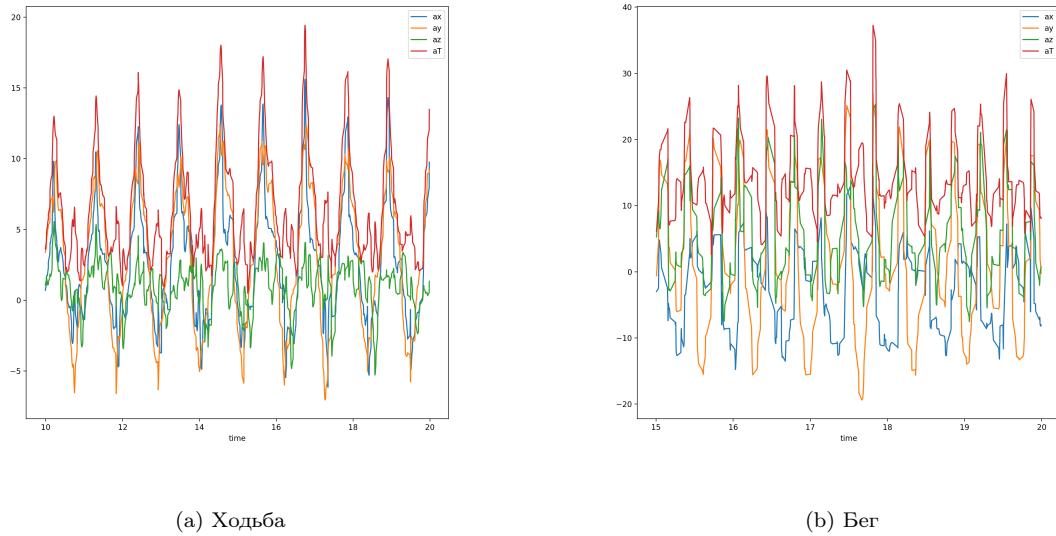


Рис. 3: Профиль данных 1.

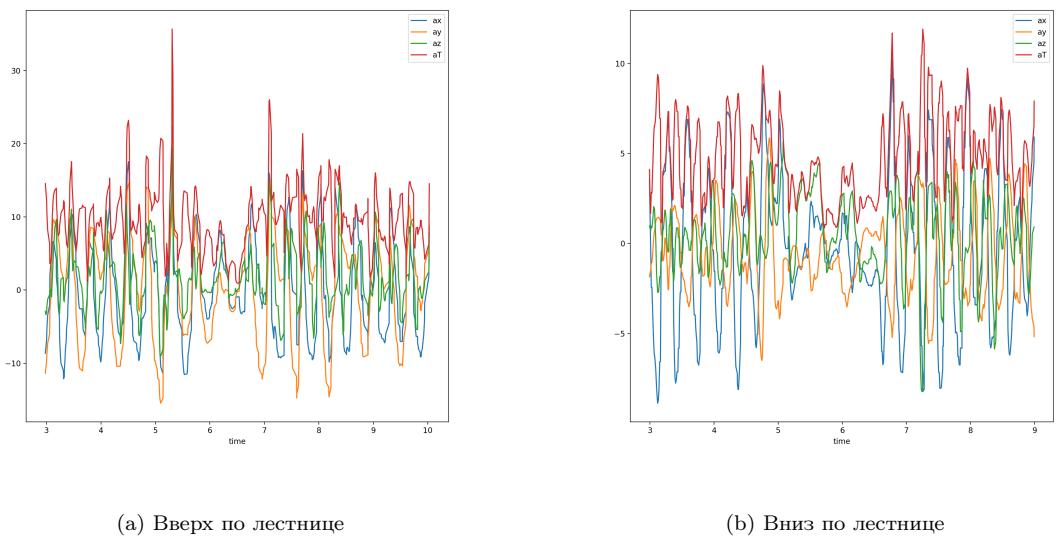


Рис. 4: Профиль данных 2.

радиуса  $F$ .

$$\mathbf{I} = \{i : S(i) = \max_{j=i-F \dots i+F} \{S(j)\}\}$$

Данные индексы будем считать началом сегмента, а также концом следующего перед ним. Полученные сегменты фильтруем на основании гистограммы их длины (например, отсекая правый и левый квантили).

---

#### Алгоритм 6.1 Сегментация временного ряда действия

---

**Вход:**  $\mathbf{S}, l, F$ ;

**Выход:**  $\mathbf{x}$ ;

- 1:  $\mathbf{I} = []$
  - 2:  $k = 0$
  - 3: **для**  $i \in 1 \dots l$
  - 4:   **если**  $S(i) = \max_{j=i-F \dots i+F} \{S(j)\}$  **то**
  - 5:      $\mathbf{I} = (\mathbf{I}, i)$
  - 6:      $k = k + 1$
  - 7: **для**  $i \in 1 \dots k - 1$
  - 8:      $\mathbf{x} = (\mathbf{x}, \mathbf{S}[\mathbf{I}[k] : \mathbf{I}[k + 1]])$
- 

## 6.2 Нормализация сегментов

После извлечения множества сегментов  $\mathbf{x}$  возникает проблема размера сегментов. В результате действия алгоритма 6.1, размеры сегментов будут различными. Чтобы их нормализовать, то есть привести их к одинаковой размерности, мы будем использовать кубические сплайны. Имея,  $n + 1$  значений  $(x_i, y_i)$  с различными  $\{x_i\}_{i=1 \dots n+1}$ , сплайн это функция  $S(x)$

- $S(x) \in C^2[\min\{x_i\}_{i=1 \dots n+1}, \max\{x_i\}_{i=1 \dots n+1}]$
- На каждом интервале  $(x_{i-1}, x_i)$ , сплайн это полином третей степени
- $S(x_i) = y_i$  для всех  $i = 1 \dots n + 1$ .

Тогда процедура нормализации сегментов выглядит следующим образом:

1. Узнаем минимальный размер сегмента  $m$
2. Для каждого сегмента  $\mathbf{x}_i$  генерируем сегмент размера  $\hat{\mathbf{x}}_i$  интраполяцией сегмента  $\mathbf{x}_i$ , а затем вычисления  $m$  равноудаленных значений сплайна внутри интервала  $1..len(\mathbf{x}_i)$ , так, что первая точка в 1, а последняя в  $len(\mathbf{x}_i)$ .

С помощью такой процедуры мы получили  $(\hat{\mathbf{x}}, Y)$  — обучающая выборка, используемая в эксперименте.

## 7 Вычислительный эксперимент

В вычислительном эксперименте использовались следующее семейство моделей локальной аппроксимации

- SSA с параметром 5.
- SSA с параметром 10.
- AR-авторегрессия с параметром 2.
- AR-авторегрессия с параметром 4.
- FFT с 2 старшими частотами.
- FFT с 5 старшими частотами.
- SEMOR с профилем ходьбы.
- SEMOR с профилем бега.
- SEMOR с профилем ходьбы вверх по лестнице.
- SEMOR с профилем ходьбы вниз по лестнице.

### 7.1 Анализ статистических свойств выборки

Имея выборку  $(\mathbf{X}, Y)$  нельзя доказать, что она является простой. В самом деле, одна и та же выборка может получиться в результате двух различных механизмов порождения данных, один из которых обеспечивает простоту а второй — нет. В качестве примера можно привести выборку со смесью двух нормальных распределений, где в первом случае их просто получили смесью двух выборок, а во втором она пораждалась из одного распределения (распределение бернулли и два нормальных распределения). Однако, можно проверить некоторые гипотезы, связанные с простой выборки. Напомним, что выборка называется *простой*, если она случайная, независимая и взятая из одного распределения. У нашей работе мы проверяли четыре гипотезы:

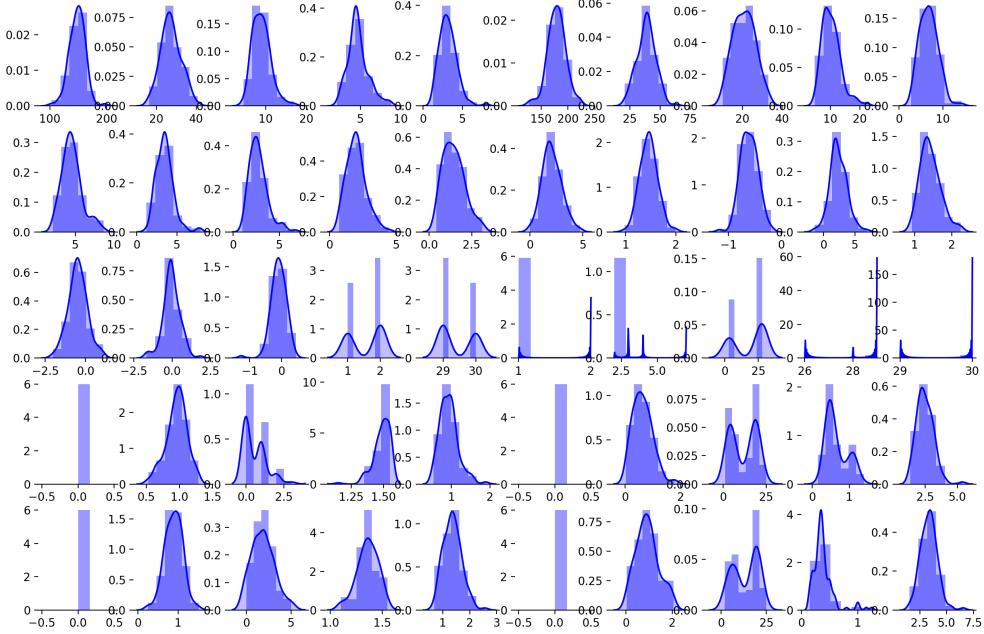
- Матожидание ошибки равна нулю, дисперсия ошибки постоянна, где под ошибкой понимается

$$\varepsilon_i = \hat{y}_i - y_i$$

Эти гипотезы проверялись вычислением ошибки и затем использования  $t$ -критерия Стьюдента.

- **Распределения признаков унимодальны.** Для проверки этой гипотезы были прямо построены распределения признаков.

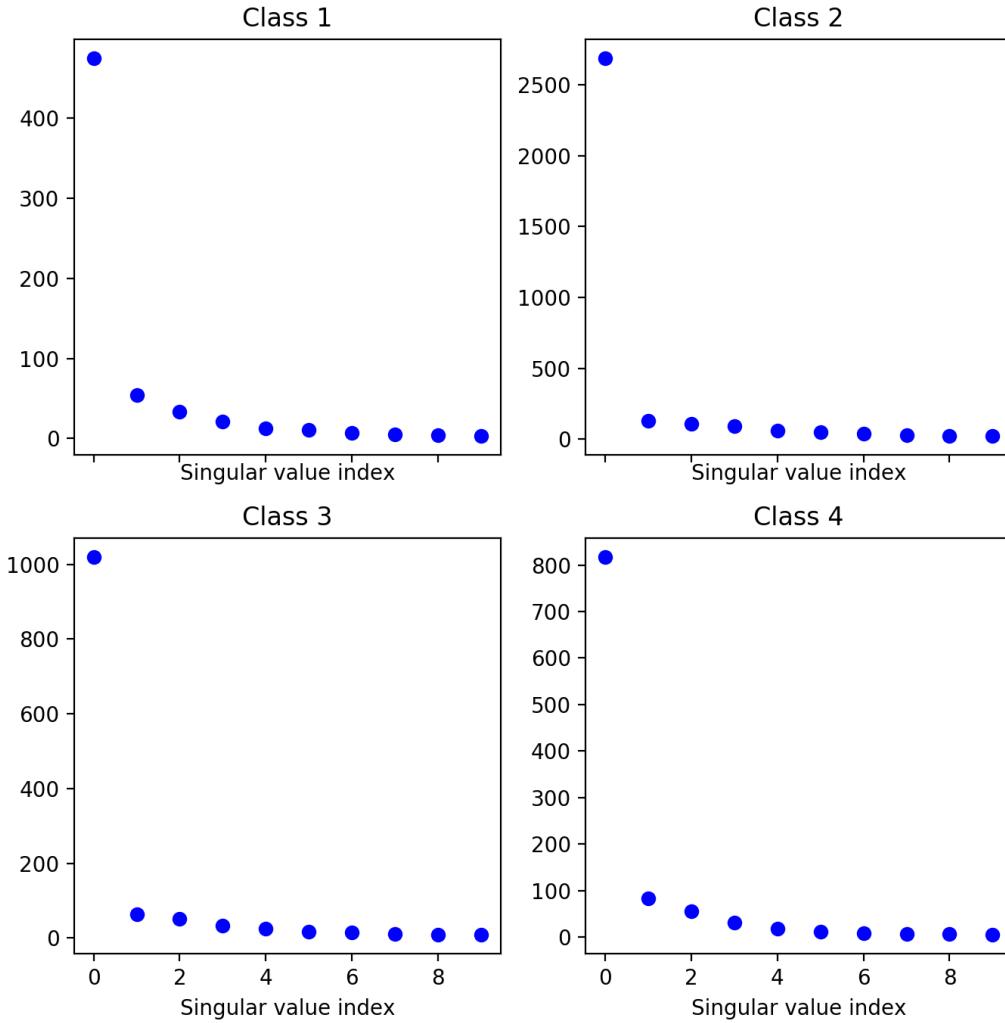
Рис. 5: Распределения признаков.



Мы видим, что большее количество признаков, построенных с помощью поиска оптимальных параметров моделей локальной аппроксимации имеют унимодальное распределение, что приводит нас к выводу о том, что выборка является почти простой.

- **Анализ спектра выборки.** Спектр выборки невырожден и имеет около одного собственного значения. Для этого мы для каждого класса  $y_i \in \{1 \dots k\}$  взяли соответствующие им объекты  $\mathbf{X}_i = \{\mathbf{x}_j : (\mathbf{x}_j, y_i) \in \mathfrak{D}\}$  и нашли спектр матрицы  $\mathbf{X}_i^\top \mathbf{X}_i$ .

Рис. 6: Спектральный анализ.



Мы видим, что спектры имеют одно выраженное собственное значение в случае каждой метки класса, что также непротиворечит простоте выборки.

## 7.2 Сложность модели

Простота (или почти простота) выборки в промежуточном пространстве дает нам основание полагать, что обобщенная линейная модель будет хорошо разделять распределения, имея при этом низкую сложность по сравнению с универсальными моделями, применяемыми к обучающей выборке  $(\mathbf{X}, y)$ . В нашей работе все рассматриваемые классификаторы имеют эквиваленты в нейронных сетях с одним скрытым

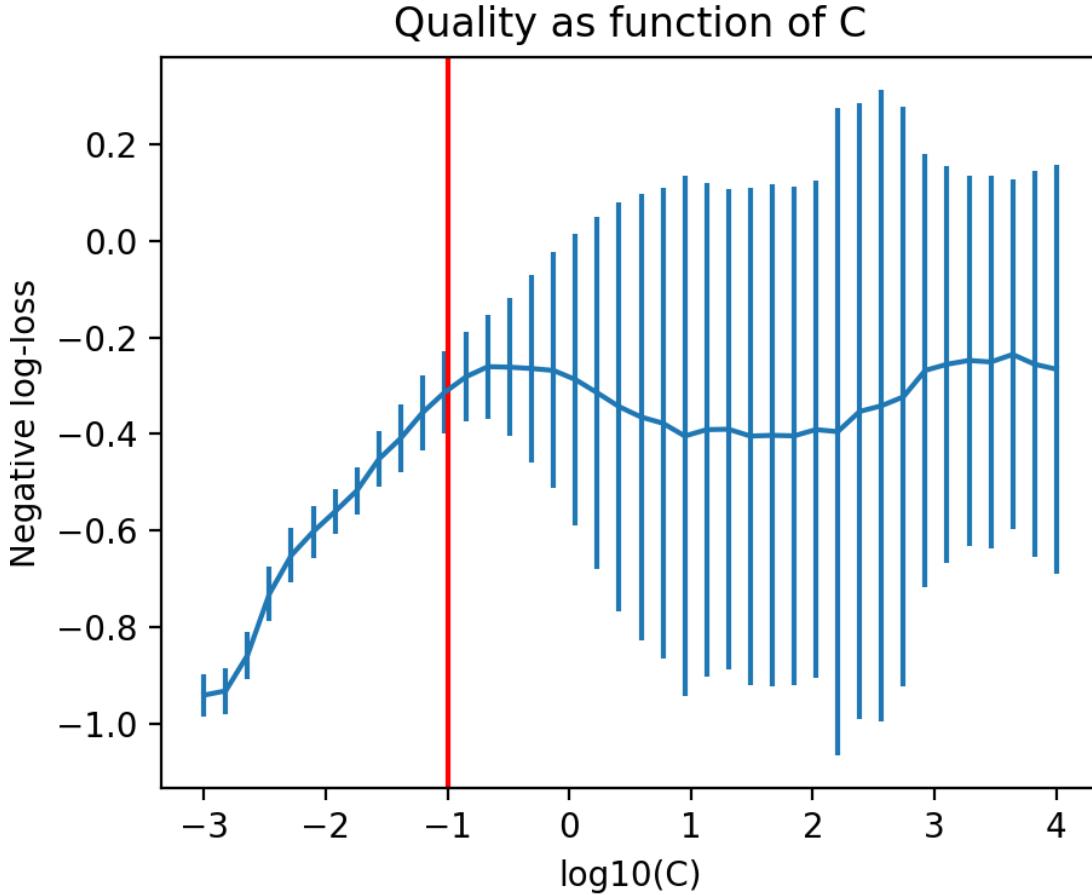
слоем. Тогда для подобных моделей имеет место следующее определение: определим *сложность модели* как количество нейронов на скрытом слое эквивалентной ей нейронной сети

$$\text{Comp}(\mu) = \#\text{neurons in the hidden layer}$$

### 7.3 Сравнение сложностей универсальной модели и GLE

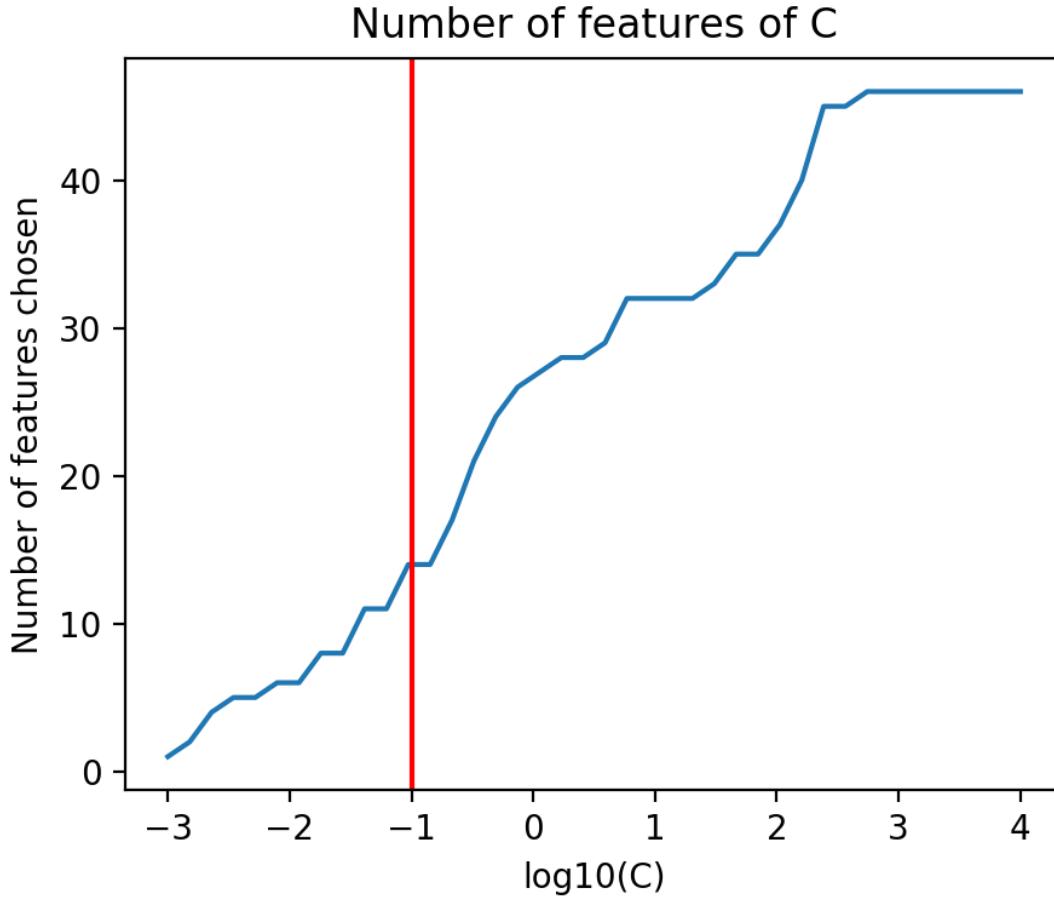
Для выборка признаков для обобщенной линейной модели (т.е. минимизации ее сложности) исследуем зависимость распределения функции ошибки на кроссвалидации в зависимости от параметра  $L_1$ -регуляризации, которая имеет свойства отбора признаков. Мы получили следующие зависимости:

Рис. 7: Функция ошибки от параметра регуляризации



Анализируя эти графики, мы выбрали параметр  $C = 0.1$ , который обеспечивает наименьшую ошибку (с положительным знаком), при которой среднее отклонение

Рис. 8: Количество признаков от параметра регуляризации



от нее не сравнимо с самой ошибкой. Этот параметр соответствует 14-ти признакам с ненулевыми коэффициентами в итоговом обученном GLE.

## 7.4 Универсальная модель.

Для сравнения с GLE на  $(\mathbf{X}, Y)$  мы обучили двуслойную нейронную сеть с различными значениями числа нейронов на скрытом слое. Ниже мы приводим зависимость функции ошибки и ее дисперсии от числа нейронов в скрытом слое.

$$L(\text{Comp}), D_L(\text{Comp}).$$

Мы видим, что с увеличением числа нейронов, а то есть сложности модели, модель начинает переобучаться. Сравним функции ошибки GLE и универсальной модели.

Красной линией обозначен фактор равный 1, то есть случай, когда сложности

Рис. 9: Распределения ошибки от сложности универсальной модели

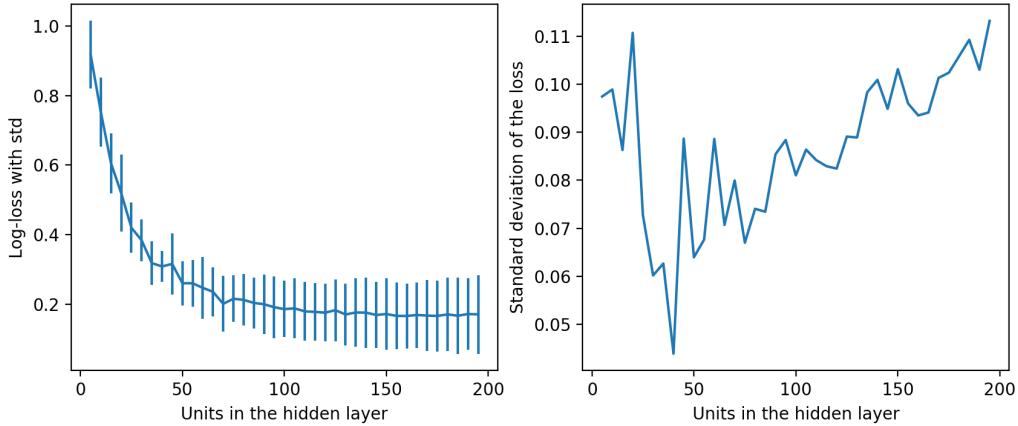
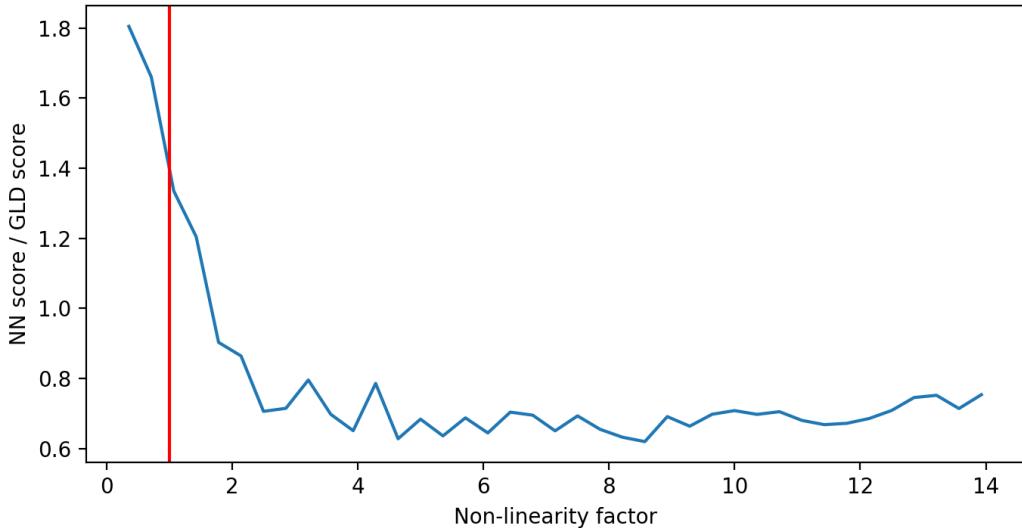


Рис. 10: Отношение ошибок от отношения сложностей



GLE и универсальной модели одинаковы. Мы видим, что в среднем ошибка универсальной модели при одинаковой сложности больше ошибки, показываемой GLE, в 1.4 раза, а для достижения отношения 1 необходимо увеличить число нейронов на скрытом слое в два раза.

## 8 Выводы

Вычислительные эксперименты показали, что в нашей задаче метод моделей локальной аппроксимации обеспечивает модель, которая адекватнее разделяет выборку чем универсальная модель и дает высокое качество классификации и низкую переобучаемость, тогда как универсальная модель обеспечивает либо первое (при высокой сложности), либо другое (при низкой сложности). Мы исследовали статистические свойства промежуточного пространства признаковых описаний временных рядов. Выборка в промежуточном пространстве простая, а аппроксимирующая ее линейная модель являются адекватной. Предложен и реализован способ построения набора моделей локальной аппроксимации для устойчивой классификации сигналов носимых устройств, тест простоты выборки в промежуточном пространстве признаковых описаний а также методика оценки ее обобщающей способности по сравнению с универсальными моделями.

## Список литературы

- [1] Argyro Kampouraki, George Manis, and Christophoros Nikou. Heartbeat time series classification with support vector machines. In *IEEE Transactions on Information Technology in Biomedicine*, volume 13, pages 512–518, 2009.
- [2] Damian Eads, Daniel Hill, Sean Davis, Simon Perkins, Junshui Ma, Reid Porter, and James Theiler. Genetic algorithms and support vector machines for time series classification. *Proceedings of SPIE - The International Society for Optical Engineering*, 4787:74–85, 2002.
- [3] Michael Hüskens and Peter Stagge. Recurrent neural networks for time series classification, 2003.
- [4] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J. Leon Zhao. Time series classification using multi-channels deep convolutional neural networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8485 LNCS, pages 298–310, 2014.
- [5] Кузнецов МП and Ивкин НП. Алгоритм классификации временных рядов акселерометра по комбинированному признаковому описанию. *Машинное обучение и анализ данных*, pages 1471–1483, 2015.
- [6] Карасикова МЕ and Стрижов ВВ. Классификация временных рядов в пространстве параметров порождающих моделей . *Информ. и её примен.*, page 121–131, 2016.
- [7] H. Hassani. Singular Spectrum Analysis: Methodology and Comparison. *Journal of Data Science*, 5:239–257, 2007.