

# Определение местоположения по сигналу акселерометра

*Zaynulina E. T., Kiseleva E. A., Protasov V. P., Fateev D. A.,  
Bozhedomov N., Tolkanev A. A., Nochevkin V., Ryabov A.*

October 2018

## 1 Краткий обзор

Данная статья посвящена использованию методов машинного обучения в задаче определения местоположения по показаниям носимых человеком сенсоров. Задача является актуальной и имеет такое применение, как, например, автоматическое включение/выключение энергозатратных сервисов при различном положении мобильного устройства. Поставленная задача решается по сигналам датчика телефона – акселерометра. Основной целью работы – это способ выбора и предобработки признаков, позволяющий уменьшить влияние шума на результат классификации и анализировать активность в независимости от пространственной ориентации мобильного устройства. Результаты, полученные в ходе вычислительного эксперимента, подтверждают применимость предложенного подхода.

Современные смартфоны обладают большим числом сенсоров и высокой вычислительной способностью. Так как в настоящее время почти каждый человек им обладает, то методы определения местоположения человека с использованием смартфонов получили наибольшее внимание со стороны исследователей. Среди этих методов - методы, основанные на беспроводных сигналах (WiFi, Bluetooth, UWB)[12][7], датчиках обзора (лазерный сканер, монокулярная и бинокулярная камера)[1], инерционных датчиках (акселерометр, гироскоп, магнитометр)[9][4][13][6]. Многие из предложенных методов локализации человека представляют собой комбинацию выше перечисленных для увеличения точности позиционирования[3][2][8]. Методы, основанные на беспроводных сигналах и датчиках обзора, помимо наличия смартфона требуют также введения дополнительного оборудования либо наличия дополнительных знаний, например карты помещения или базы данных силы сигнала (RSSI) WiFi точки в зависимости от координаты (WiFi fingerprint). Однако не всегда возможно предоставить карту помещения, например, в силу конфиденциальности; вспомогательное оборудование, в свою очередь, требует технического обслуживания и больших затрат. Что касается WiFi позиционирования, то при наличии существующей

базы данных WiFi fingerprint при некотором изменении среды, позиционирование будет неточным, поэтому база данных нуждается в постоянном обновлении[11].

Чтобы избежать данных проблем, предлагается метод, основанный на инерционных датчиках. В качестве базового алгоритма рассматривается pedestrian dead reckoning (PDR) [5]. По сравнению с методами, основанными на беспроводных сигналах и датчиках обзора, PDR рассчитывает относительно точное местоположение человека быстрее и потребляя меньше вычислительной мощности. Для фильтрации шума в данных используется фильтр Калмана [14]. Особенность данной работы состоит в том, чтобы восстанавливать траекторию не от точки к точке, а всю целиком. Для работы с полученным многомерным пространством предлагается использовать метод PLS[10].

**Новизна:** задача исследования ставится в терминах projection to the latent space (It is pretty much used that way in machine learning — you observe some data which is in the space that you can observe, and you want to map it to a latent space where similar data points are closer together.)

**Ключевые слова:** обработка сигналов; сенсоры; акселерометр; анализ данных; машинное обучение, инерциальные методы, позиционные методы

## 2 Введение

Задача определения местоположения произвольного телефона для любого пользователя является сложной по следующим причинам: манера движения, в частности походка, у людей сильно различается; характеристики одежды, карманов и сумок варьируются в широких пределах, ориентация телефона в пространстве может быть произвольной. Датчики мобильных устройств имеют значительный разброс параметров

## 3 Постановка задачи

При решении задачи используются данные, полученные с помощью инерционных датчиков. Эти данные представляются в виде многомерных временных рядов  $s(t) \in \mathbb{R}^N$ . Каждому временному ряду ставится в соответствие вектор признаков. Эти вектора образуют матрицу признаков  $X \in \mathbb{R}^{N \times T}$ . По данной матрице предсказывается матрица траекторий пешехода вида  $Y \in \mathbb{R}^{N \times T}$ , а строками данной матрицы являются временные ряды  $y(t)$ , демонстрирующие изменение положения по широте и долготе в течение времени. Модель имеет вид:

$$f : X \rightarrow Y$$

Используемые в задаче данные, были собраны с разных расположений датчиков: рюкзак, нога, рука, туловище. Поэтому задача разбивается на

две: определение класса расположения датчика ( $P$ ) и предсказание самой траектории перемещения на основе решения первой задачи:

$$f_1 : X \rightarrow P = \{0, 1, 2, 3\}$$

$$f_2 : X, P \rightarrow Y$$

Для решения задач используем метод опорных векторов (SVM) для случая линейной неразделимости классов, который минимизирует следующий функционал  $S(w|f, X, Y)$  и введём штраф за суммарную ошибку:

$$\min_{w, w_0} S(w, w_0) = \frac{1}{2} \|w\|^2 + C \sum_{i=1} \xi_i$$

где  $C$  - параметр настройки метода,  $\xi_i$  - набор дополнительных переменных характеризующих величину ошибки на объектах  $x_i$ , вектор  $w$  - перпендикулярен к разделяющей гиперплоскости.

При условиях:

$$y_i(w^T x_i + w_0) \geq 1 - \xi_i$$

$$\xi_i \geq 0 \quad \forall i$$

Для оценки качества модели используется критерий суммы квадратов отклонений предсказанных координат от истинных, а также корреляция между предсказанной и истинной траекториями пешехода.

Формально постановку задачи следует записать следующим образом:

$$w^* = \arg \min_w S(w|f, X, Y).$$

## 4 Базовый алгоритм

В ходе получения матрицы признаков  $X$  в данных подавляются шумы высокой частоты с помощью применения Гауссового сглаживания с разными параметрами для 6-ти каналов гиросtabilизатора (IMU channels) и 2-ух скоростных каналов. Преобразуем сглаженные угловое и линейные ускорения в вектор признаков.

В качестве базового алгоритма используется каскадная регрессия состоящая из модели вида:

- подаются данные на вход SVM классификатора, который уже их разделяет на 4 класса: нога, сумка, рука, тело
- далее полученные данные идут на вход на 2 SVR-регрессора для каждого класса, которые обучаются на тренировочных данных, которые в итоге выдают скорости передвижения человека для каждого временного блока. Именно 2 SVR-регрессора для предсказания двумерной скорости в IMU-стабилизированной системе координат, игнорируя вертикальную составляющую.

Но полученные векторы скоростей содержат ошибки, которые связаны с неточностями инерционных датчиков: системой ротации. Поэтому делается предположение о низкочастотных смещениях в линейном ускорении. Этот подход не имеет явных физических обоснований, но позволяет обойти явное моделирование шумов / смещений и сводит задачу к задаче минимизации:

$$\min_{\{x_I^1, x^5 1_I, \dots\}} V_{bias} = \min_{\{x_I^1, x^5 1_I, \dots\}} \sum_{f \in F_2} \|v_C^f - v_R^f\| + \lambda \sum_{f \in F_1} \|x_I^f\|^2,$$

$$v_C^f = R_{SW}^f \sum_{f'=1}^f R_{WI}^{f'} (a_I^{f'} + x_I^{f'}),$$

где  $f$  - единица блока выборки,  $F$  - блок выборки,  $v_C^f$  - скорректированное значение скорости,  $v_R^f$  - предсказанное значение скорости,  $I$  - система координат устройства,  $W$  - глобальная система координат,  $S$  - IMU-стабилизированная система координат,  $R_{AB}$  - матрица перехода из системы координат  $B$  в систему координат  $A$ .

Для каждого класса создается SVR-регрессор, предсказывающий угловые скорости пешехода в каждом временном блоке.

На контрольной выборке для SVM-классификатора и каждого SVR-регрессора подбираются оптимальные значения гиперпараметров.

По полученным значениям скоростей восстанавливается траектория пешехода.

После получения из данных матрицы признаков  $X$

Формально алгоритм описывается следующим образом:

**Вход:**  $X, Y_{class}, Y, X_{test}$

- 1: *initialize classifier\_options*
- 2: *classifier = SVMClassifier(classifier\_options);*
- 3: *classifier.fit(X, Y\_class)*
- 4: **для** *cls in classes:*
- 5:   *initialize regressor\_cls\_options*
- 6:   *regressor\_cls = SVRRRegressor(regressor\_cls\_options)*
- 7:   *regressor\_cls.fit(X[X[ind] ∈ cls], Y)[Y[ind] ∈ cls]*
- 8: *Y\_test-class = classifier.predict(X\_test)*
- 9: **для** *cls in classes:*
- 10:   *Velocity\_cls = regressor\_cls.predict(X\_test[Velocity\_class[ind] == cls])*
- 11:    $x_I^1, x^5 1_I, \dots = \arg \min_{\{x_I^1, x^5 1_I, \dots\}} V_{bias\_cls}$
- 12:   *Velocity\_cls =  $R_{SW}^f \sum_{f'=1}^f R_{WI}^{f'} (a_I^{f'} + x_I^{f'})$*
- 13:   *Trajectory\_cls recovery depending on Velocity\_cls*
- 14: **return** *Full\_trajectory*

## 5 Эксперимент

Цель эксперимента: найти параметры модели для более точного предсказания исходной траектории.

В ходе эксперимента используются данные в статье, исследуемой алгоритм RIDI[14]. Данные были собраны с помощью инерционных датчиков смартфона с разным расположением: в руке, на ноге, в сумке и на поясе. Выборки содержат траектории с временным блоком в 100 минут и частотой сигнала 200 Гц.

В качестве объекта рассматривается положение в определенный момент времени  $i$ . Признаками объекта являются угловые скорости и линейные ускорения в стабилизированной системе координат датчиков в моменты времени  $i - window\_size, \dots, i$ , где  $window\_size$  - размер окна (равен 200). Целевыми переменными являются метки классов, характеризующие то, в каком положении находился смартфон при получении определенных данных, а также скорости в данный момент времени  $i$ , которые вычисляются через координаты пешехода и прошедшее время. По полученным данным после уточнения скоростей с помощью оптимизации функции  $V_{bias}$  строится предсказанная траектория пешехода.

В ходе эксперимента исследовалась зависимость качества моделей на контрольной выборке в зависимости от параметров SVM-регрессоров. Во всех моделях в качестве ядер были выбраны радиальные базисные функции, подбирались такие параметры как коэффициент штрафа  $C$  и ядерный коэффициент  $\gamma$ . Качество измерялось с помощью кросс-валидации. Из результатов эксперимента следует, что для каждого расположения смартфона и каждого канала данных должны быть выбраны свои параметры модели. Это подтверждает разумность классификации типа расположения смартфона перед непосредственным предсказанием траектории.

Графики зависимости качества предсказания модели от параметров:

1. Выборка 1 состоит из 30742 объектов (8728 объектов класса рука, 6106 объектов класса нога, 7758 объектов класса тело, 8150 объектов класса сумка).
2. Выборка 2 состоит из 42731 объектов (13204 объектов класса рука, 8083 объектов класса нога, 11105 объектов класса тело, 10339 объектов класса сумка).
3. Выборка 3 состоит из 35892 объектов (9458 объектов класса рука, 7304 объектов класса нога, 13306 объектов класса тело, 5824 объектов класса сумка).

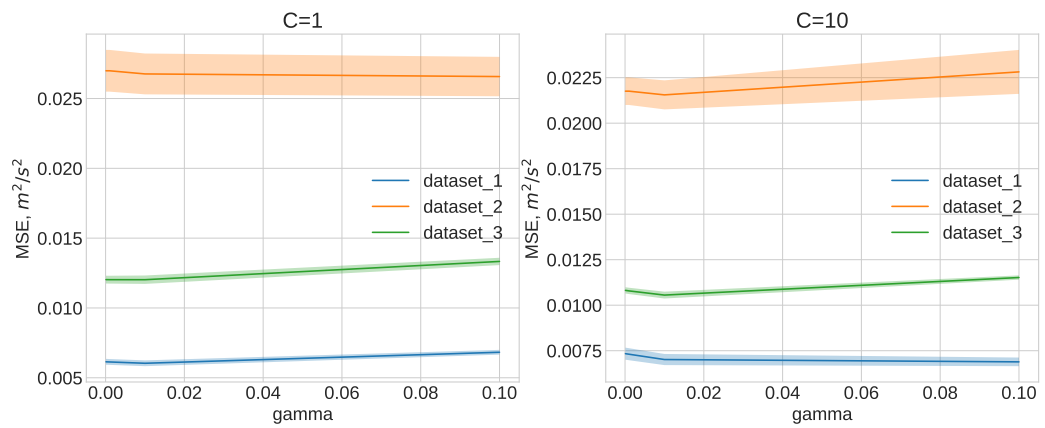


Рис. 1: Рука, канал 0

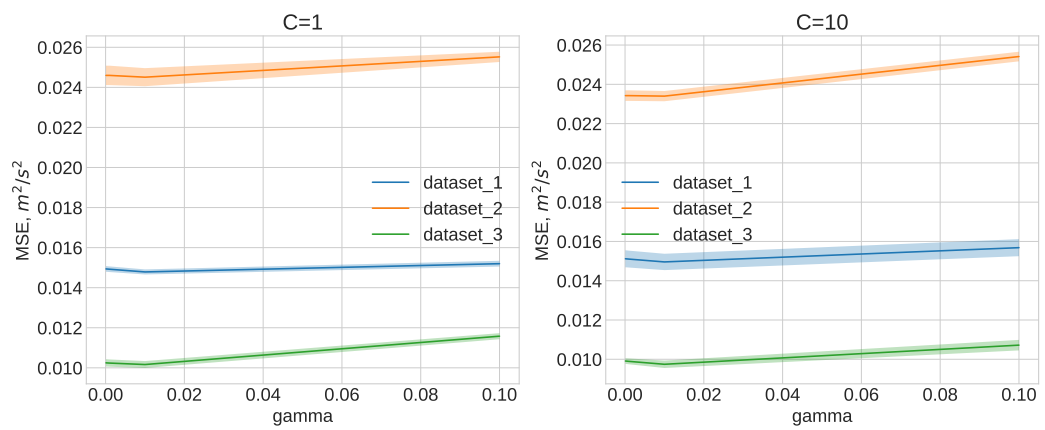


Рис. 2: Рука, канал 1

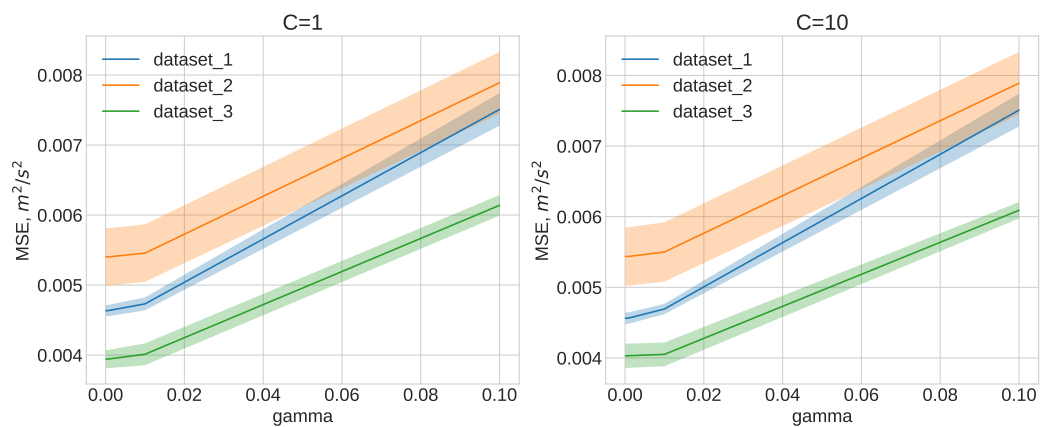


Рис. 3: Нога, канал 0

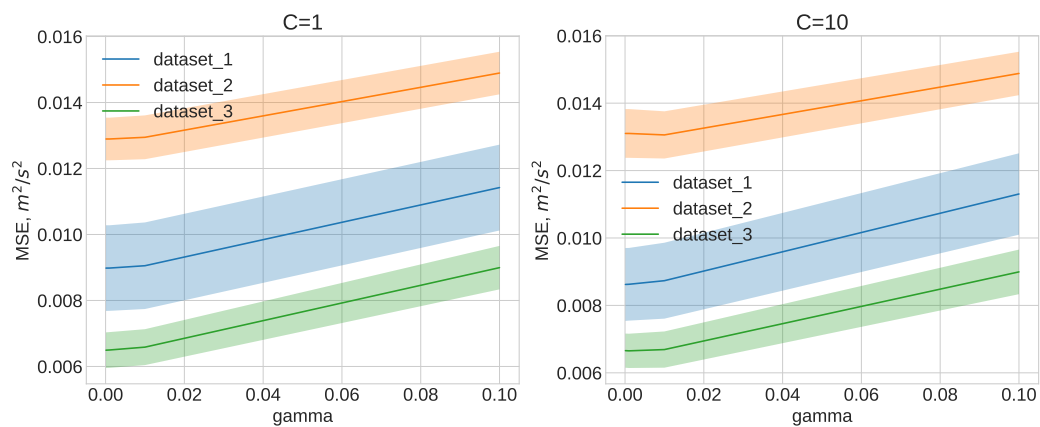


Рис. 4: Нога, канал 1

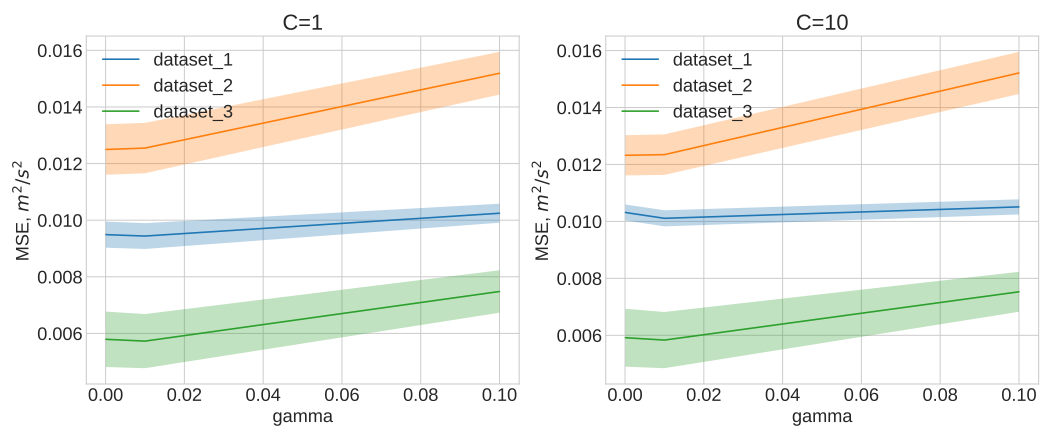


Рис. 5: Сумка, канал 0

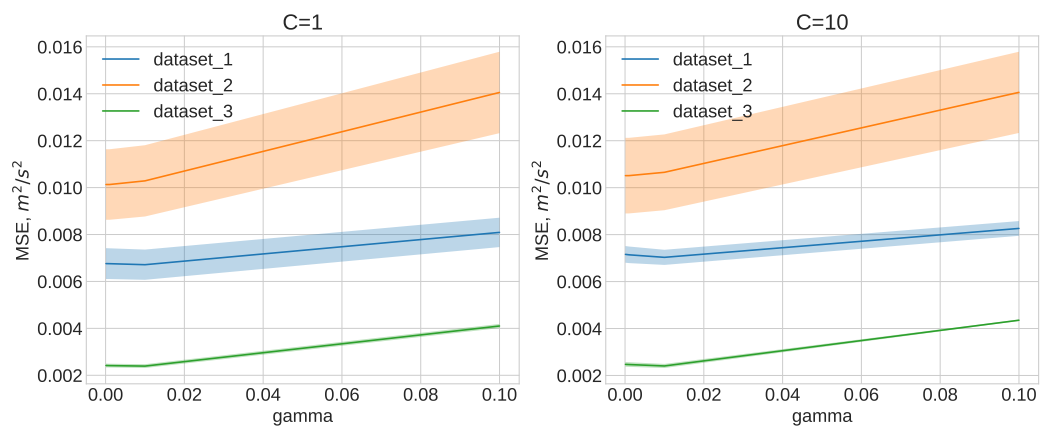


Рис. 6: Сумка, канал 1



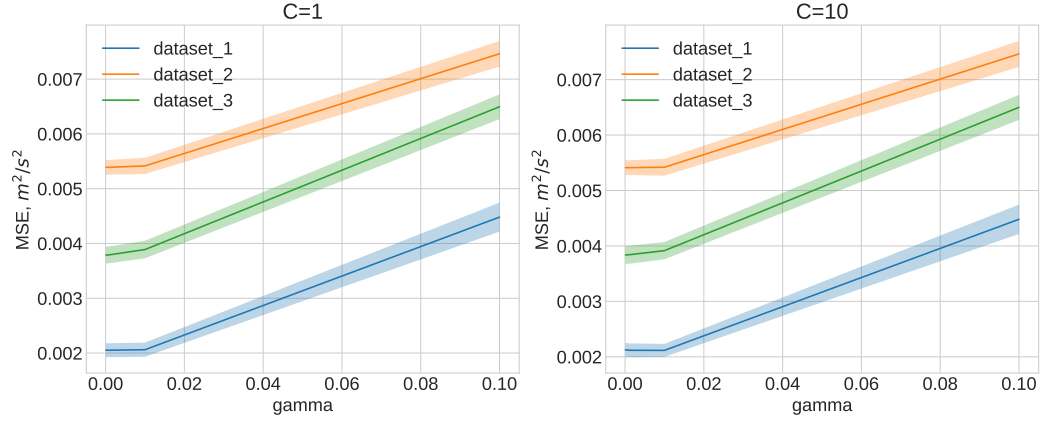


Рис. 7: Тело, канал 0

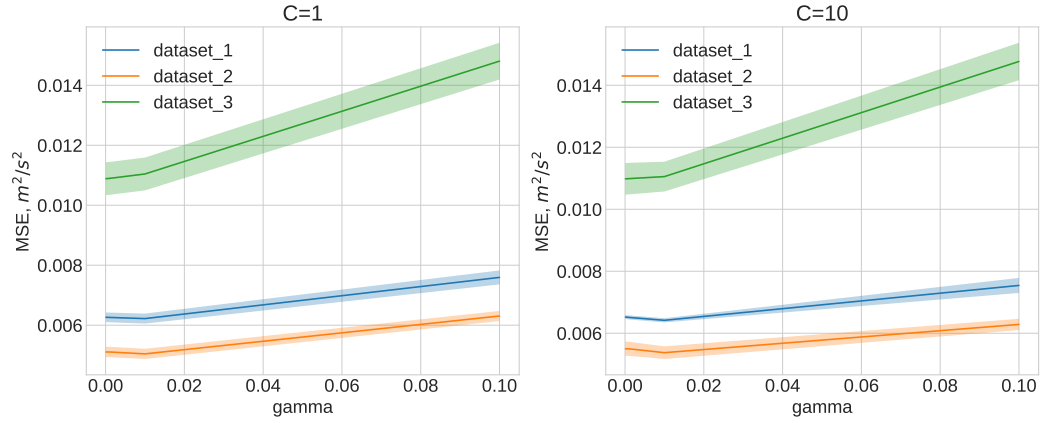


Рис. 8: Тело, канал 1

Для всех классов и выборок оптимальные значения параметра  $\gamma$  близки к 0.001, 0.01, поэтому при дальнейшем обучении моделей на большом количестве данных при заранее не заданных параметрах SVM-регрессоров, при поиске по сетке для параметра  $\gamma$  будут использоваться только эти значения. Тогда для построенных моделей оптимальными параметрами будут следующие:

	Рука	Нога	Сумка	Тело
C	10	1	1	1
$\gamma$	0.01	0.001	0.01	0.001

По полученным значениям ошибок на кросс-валидации были выбраны оптимальные модели. С помощью этих моделей были построены траектории для каждого класса расположения смартфона (в качестве тестовой выборки была использована выборка Zhicheng). При этом траектории были построены для случаев, когда дополнительная корректировка весов с помощью оптимизации  $V_{bias}$  не производилась (сиреневая линия) и когда производилась (синяя линия). Истинная траектория обозначена красным цветом.



Рис. 9: Траектории



Рис. 10: Траектории

## 6 Выводы

Путем изначального определения расположения смартфона у человека (класс в данной задаче), были подобраны более подходящие параметры для моделей, которые увеличили точность построенных траекторий.

В ходе данной работы были повторены результаты статьи для алгоритма RIDI [14]. При работе с данными и для их улучшения был использован фильтр Гаусса.

В дальнейшем планируется применить полученную модель для дополнительно собранных данных, а также улучшить методы обработки данных

для уменьшения шума (применение фильтра Калмана) и посмотреть другие способы оптимизации модели.

## 7 Приложения

Таблица 1: Зависимости MSE ( $m^2/s^2$ ) от параметров моделей для выборки 1

Регрессор	C=1			C=10		
	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 0.0001$	$\gamma = 0.01$
Сумка, 0	0.00949	0.00948	0.00944	0.01025	0.01029	0.01011
Сумка, 1	0.00676	0.00676	0.00671	0.00809	0.00714	0.00703
Тело, 0	0.00205	0.00205	0.00206	0.00448	0.00213	0.00212
Тело, 1	0.00626	0.00626	0.00622	0.00759	0.00651	0.00642
Рука, 0	0.00614	0.00613	0.00604	0.00683	0.00734	0.00702
Рука, 1	0.01494	0.01492	0.01479	0.0152	0.01512	0.01496
Нога, 0	0.00463	0.00464	0.00473	0.00751	0.00456	0.00469
Нога, 1	0.00898	0.00898	0.00905	0.01142	0.00862	0.00873

Таблица 2: Зависимости MSE ( $m^2/s^2$ ) от параметров моделей для выборки 2

Регрессор	C=1			C=10		
	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 0.0001$	$\gamma = 0.01$
Сумка, 0	0.0125	0.0125	0.01255	0.01519	0.01232	0.01234
Сумка, 1	0.01013	0.01013	0.01029	0.01406	0.01051	0.01065
Тело, 0	0.00205	0.00205	0.00206	0.00448	0.00213	0.00212
Тело, 1	0.00511	0.00511	0.00504	0.00631	0.0055	0.00537
Рука, 0	0.02699	0.02699	0.02676	0.02657	0.02176	0.02155
Рука, 1	0.0246	0.0246	0.02451	0.02552	0.02342	0.0234
Нога, 0	0.0054	0.0054	0.00546	0.00789	0.00544	0.0055
Нога, 1	0.01289	0.01289	0.01295	0.01489	0.0131	0.01306

Таблица 3: Зависимости MSE ( $m^2/s^2$ ) от параметров моделей для выборки 3

Регрессор	C=1					C=10				
	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 0.0001$	$\gamma = 0.01$
Сумка, 0	0.00579	0.00579	0.00573	0.00748	0.00592	0.00591	0.00583	0.00753	0.00591	0.00583
Сумка, 1	0.00242	0.00241	0.00239	0.0041	0.00247	0.00247	0.00241	0.00435	0.00247	0.00241
Тело, 0	0.00379	0.00379	0.00389	0.0065	0.00384	0.00384	0.00391	0.0065	0.00384	0.00391
Тело, 1	0.00511	0.00511	0.00504	0.00631	0.0055	0.0055	0.00537	0.00629	0.0055	0.00537
Рука, 0	0.02699	0.02699	0.02676	0.02657	0.02176	0.02176	0.02155	0.02282	0.02176	0.02155
Рука, 1	0.01025	0.01024	0.01016	0.01158	0.00991	0.00989	0.00975	0.01072	0.00991	0.00975
Нога, 0	0.00394	0.00395	0.00401	0.00614	0.00403	0.00403	0.00405	0.00609	0.00403	0.00405
Нога, 1	0.00649	0.0065	0.00659	0.009	0.00666	0.00665	0.00669	0.00899	0.00665	0.00669

## Список литературы

- [1] Erich Bruns and Oliver Bimber. Adaptive training of video sets for image recognition on mobile phones. *Personal and Ubiquitous Computing*, 13(2):165–178, 2009.
- [2] L. Chen, E. H. Wu, M. Jin, and G. Chen. Intelligent fusion of wi-fi and inertial sensor-based positioning systems for indoor pedestrian navigation. *IEEE Sensors Journal*, 14(11):4034–4042, Nov 2014.
- [3] Frédéric Evennou and François Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *EURASIP J. Adv. Sig. Proc.*, 2006, 2006.
- [4] Michael Hardegger, Daniel Roggen, and Gerhard Tröster. 3d actionslam: wearable person tracking in multi-floor environments. *Personal and Ubiquitous Computing*, 19(1):123–141, 2015.
- [5] R. Hostettler and S. Särkkä. Imu and magnetometer modeling for smartphone-based pdr. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Oct 2016.
- [6] W. Kang and Y. Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal*, 15(5):2906–2916, May 2015.
- [7] Soo-Cheol Kim, Young-Sik Jeong, and Sang Oh Park. Rfid-based indoor location tracking to ensure the safety of the elderly in smart home environments. *Personal and Ubiquitous Computing*, 17(8):1699–1707, 2013.
- [8] M. Kok, J. D. Hol, and T. B. Schön. Indoor positioning using ultrawideband and inertial measurements. *IEEE Transactions on Vehicular Technology*, 64(4):1293–1303, April 2015.
- [9] Kwanghyo Park, Hyojeong Shin, and Hojung Cha. Smartphone-based pedestrian tracking in indoor corridor environments. *Personal and Ubiquitous Computing*, 17(2):359–370, 2013.
- [10] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In Craig Saunders, Marko Grobelnik, Steve Gunn, and John Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection*, pages 34–51, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [11] Joaquín Torres-Sospedra and Adriano J. C. Moreira. Analysis of sources of large positioning errors in deterministic fingerprinting. *Sensors*, 17(12):2736, 2017.
- [12] Rodrigo Vera, Sergio F. Ochoa, and Roberto G. Aldunate. Edips: an easy to deploy indoor positioning system to support loosely coupled mobile work. *Personal and Ubiquitous Computing*, 15(4):365–376, 2011.

- [13] Boyuan Wang, Xuelin Liu, Baoguo Yu, Ruicai Jia, and Xingli Gan. Pedestrian dead reckoning based on motion mode recognition using a smartphone. *Sensors*, 18(6):1811, 2018.
- [14] Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. *CoRR*, abs/1712.09004, 2017.