

Санкт-Петербургский государственный университет

Программная инженерия

Байгельдин Александр Юрьевич

Система сбора и анализа показателей
жизнедеятельности на основе данных с
мобильных устройств

Выпускная квалификационная работа

Научный руководитель:
к. ф.-м. н., доцент Романовский К. Ю.

Рецензент:
Зам. ген. директора ООО “ПитерСофтвареХаус” Хитров Д. В.

Санкт-Петербург
2018

Saint Petersburg State University

Software Engineering

Aleksandr Baigeldin

System for collection and analysis of vital factors based on data from mobile devices

Graduation Thesis

Scientific supervisor:
assistant professor Konstantin Romanovsky

Reviewer:
Deputy director general of “PiterSoftwareHouse” Ltd. Denis Khitrov

Saint-Petersburg
2018

Оглавление

Введение	4
1. Постановка задачи	6
2. Обзор	7
2.1. Физиология стресса	7
2.2. Существующие решения	8
3. Архитектура	9
3.1. Функциональность приложения	9
3.2. Используемые технологии	9
3.3. Взаимодействие с сенсорами	10
3.4. Взаимодействие с моделью	12
4. Мобильное приложение	15
4.1. Интерфейс	15
4.2. Особенности реализации	15
5. Модель машинного обучения	17
5.1. Сбор данных	17
5.2. Выбор модели	17
5.3. Оценка эффективности	17
Заключение	18
Список литературы	19

Введение

Стресс — это совокупность неспецифических (т.е. независимых от типа стрессора) адаптационных реакций организма на воздействие различных факторов (физических или психологических), нарушающих его гомеостаз (стабильное, равновесное состояние) [1]. В современной медицине принято разделять понятие положительного стресса (эустресса) и отрицательного стресса (дистресса) [2]. В результате положительного стресса повышается функциональный резерв организма, происходит его адаптация к стрессовому фактору и ликвидация самого стресса. Однако, когда организм постоянно подвергается стрессу или же стрессор слишком сильный, защитные силы организма истощаются и он становится не в состоянии самостоятельно справиться со стрессом. От такого стресса страдает иммунная система, он подрывает здоровье человека и способствует развитию тяжелых заболеваний, таких как депрессивное расстройство, диабет и даже рак [3]. В связи с этим, получили широкое развитие различные методы управления стрессом, которые помогают предупреждать его отрицательное воздействие. Однако, зачастую человек не замечает или не осознает того, что подвергается воздействию стресса. Поэтому перспективной областью исследования является автоматическое отслеживание стресса в повседневной жизни в реальном времени.

Поскольку основное воздействие стресс оказывает на нервную и эндокринную системы организма, то для его определения логичным является поиск соответствующих паттернов в работе этих систем. Например, анализ крови может выявить повышенное содержание кортизола (глюкокортикоидного “гормона стресса”) в крови. Однако, инвазивные методы не подходят для непрерывного отслеживания стресса. В связи с этим, особый интерес вызывает реакция нервной системы организма на стресс, а если точнее, то реакция симпатического отдела автономной нервной системы, который отвечает за мобилизацию сил организма в экстренных ситуациях. Симпатическая нервная система оказывает влияние на частоту сердцебиения и дыхания, кровяное давление, элект-

трическую активность кожи и другие показатели. Поэтому диапазон медицинских сенсоров, с помощью которых можно в той или иной мере определять стресс, довольно обширен: пульсометры, тонометры, GSR сенсоры, и т.д. Тем не менее, наиболее перспективным типом сенсоров для задачи отслеживания стресса в реальном времени кажутся именно пульсометры, т.к. несмотря на небольшую цену, они обладают необходимой мобильностью и предоставляют возможность высчитывать один из самых важных показателей активности симпатической нервной системы — вариабельность сердечного ритма [4].

Однако, симпатическая нервная система реагирует даже на небольшие стрессоры, которые нет смысла учитывать в статистике, но которые при этом оказывают влияние на вариабельность сердечного ритма. Например, даже при медленной ходьбе вариабельность сердечного ритма отличается от сидячего положения [5], хотя нельзя назвать ходьбу стрессом в отрицательном смысле. Поэтому учет физической активности (например, на основе данных с акселерометра) является хорошим способом отфильтровать ложные срабатывания отслеживающей стресс системы. Для комбинации показателей физической активности и показателей активности симпатической нервной системы при определении стресса можно применить популярное на сегодняшний день в медицине машинное обучение.

Таким образом, для задачи автоматического отслеживания стресса в реальном времени требуется система, которая бы определяла стресс на основе данных с пульсометра и акселерометра и обладала бы достаточной мобильностью для того, чтобы применять ее в повседневной жизни.

1. Постановка задачи

Целью данной работы является создание прототипа мобильного приложения, отслеживающего человеческий стресс в реальном времени, определяя его на основе данных полученных с акселерометра мобильного телефона и внешнего пульсометра, применяя для этого машинное обучение.

Для достижения этой цели были поставлены следующие задачи:

- Ознакомиться с природой человеческого стресса и изучить публикации на тему определения стресса на основе медицинских данных.
- Спроектировать архитектуру приложения и его взаимодействия с медицинскими сенсорами и моделью машинного обучения.
- Написать мобильное приложение для сбора данных и выделения из них признаков, полезных для определения стресса.
- Выбрать способ сбора данных, обучить модель на собранных данных, оценить ее эффективность и интегрировать ее в приложение, чтобы достичь определения стресса в реальном времени.

2. Обзор

2.1. Физиология стресса

Каким бы не был стресс, эмоциональным или физическим, организм всегда реагирует на него одинаково. Именно поэтому стресс определяется как неспецифическая реакция организма. Это значит, что вне зависимости от того, положительное ли событие произошло (например, победа в соревновании) или отрицательное (например, получение физической травмы), в организме активируются одни и те же механизмы. Разница в положительном и отрицательном стрессе заключается только в том, смог ли организм с ним справиться до того, как истощились его ресурсы, т.е. чем чаще организм подвергается конкретному стрессору и чем он продолжительнее и интенсивнее, тем больше вероятность, что данный стрессор отрицательно влияет на организм. Этот процесс состоящий из трех стадий (реакция тревоги, стадия сопротивления и стадия истощения) получил название общего адаптационного синдрома.

Главными системами организма, которые отвечают на стресс, являются гипоталамо-гипофизарно-надпочечниковая (НРА) система и вегетативная нервная система. При активации НРА в кровь выделяются глюкокортикоидные гормоны (в частности кортизол), повышается содержание глюкозы в крови, кровяное давление и тонус мышц. Поэтому анализ крови может достаточно точно определить уровень стресса. Также уровень стресса сильно коррелирует с содержанием альфа-амилазы в слюне [6].

Вегетативная нервная система — это отдел нервной системы, регулирующий деятельность внутренних органов, который в свою очередь делится на симпатический и парасимпатический отделы. Парасимпатическая стимуляция одних органов оказывает тормозное действие, а других — возбуждающее. То же касается и симпатической стимуляции, но в большинстве случаев действие парасимпатической и симпатической систем противоположно. В частности, парасимпатическая система

отвечает за приведение тела в гомеостаз (т.е. равновесное состояние): уменьшает частоту сердцебиения, но при этом активизирует органы отвечающие за переваривание пищи. При активации симпатической системы все происходит наоборот: увеличивается частота дыхания и сердцебиения, выделяется пот, изменяется электрическая проводимость кожи. В целом, баланс активности симпатической и парасимпатической систем определяет уровень стресса.

Одним из самых важных показателей баланса вегетативной нервной системы является вариабельность сердечного ритма (HRV) — изменчивость промежутков между ударами сердца, т.е. изменчивость R-R интервалов. Чем выше HRV, тем более активен парасимпатический отдел нервной системы. Соответственно, чем ниже HRV, тем активнее симпатический отдел.

HRV можно применять для определения как хронического, так и ситуативного стресса. Однако, он сильно зависит от возраста, пола и других факторов, поэтому зачастую некорректно сравнивать свой HRV с HRV другого человека. Правильнее будет сравнивать его с базовыми значениями, которые лучше измерять в сразу после пробуждения.

2.2. Существующие решения

В работе “Stress Detection Using Low Cost Heart Rate Sensors” [7] была достигнута точность 75% в определении стресса по данным пульсометра. Однако, в данной работе не была учтена физическая активность. Данные собирались в строго сидячем положении, любое движение считалось нарушением протокола.

В работе “Modeling perceived stress via HRV and accelerometer sensor streams” [8] была достигнута точность 87%. В ней была учтена физическая активность, однако способ ее вычисления не идеален — он зависит от поворота мобильного телефона в пространстве и не учитывает ошибку акселерометра. Другим важным недочетом является то, что HRV не было нормализовано относительно базовых значений для конкретного человека.

3. Архитектура

3.1. Функциональность приложения

В приложении была реализована следующая функциональность:

- Подключение медицинских сенсоров (в частности, пульсометров) по протоколу bluetooth и получение с них данных о сердцебиении.
- Отображение последних значений главных показателей (HRV, сердечный ритм и физическая активность) и состояния стресса в реальном времени.
- Отображение тенденций главных показателей в виде графиков и возможность смотреть статистику по прошедшему времени.
- Вычисление базовых значений сердцебиения и ошибки акселерометра для калибровки модели.
- Возможность указывать уровень воспринимаемого стресса в версии приложения для разработчика для разметки обучающей выборки.

3.2. Используемые технологии

При написании приложения были использованы следующие технологии:

- MedM DeviceKit SDK — библиотека для iOS и Android для подключения медицинских сенсоров по протоколу bluetooth.
- React Native — фреймворк для разработки нативных приложений для iOS и Android с использованием библиотеки React для построения пользовательских интерфейсов.
- TypeScript — язык программирования со статической типизацией, который является надмножеством JavaScript и позволяет писать более удобный в сопровождении код.

- MobX — библиотека для управления состоянием приложения в реактивном стиле.
- D3 — набор инструментов для визуализации данных.
- Для обучения модели была использована библиотека для машинного обучения SciKit-Learn и язык программирования Python.
- Языки программирования Kotlin и Swift использовались для написания адаптера MedM DeviceKit SDK для React Native.

3.3. Взаимодействие с сенсорами

Взаимодействие приложения с акселерометром и пульсометром заключается в сборе с них сырых данных и последующего извлечения из них признаков для обучения модели. Общая схема взаимодействия представлена на рис. 1.

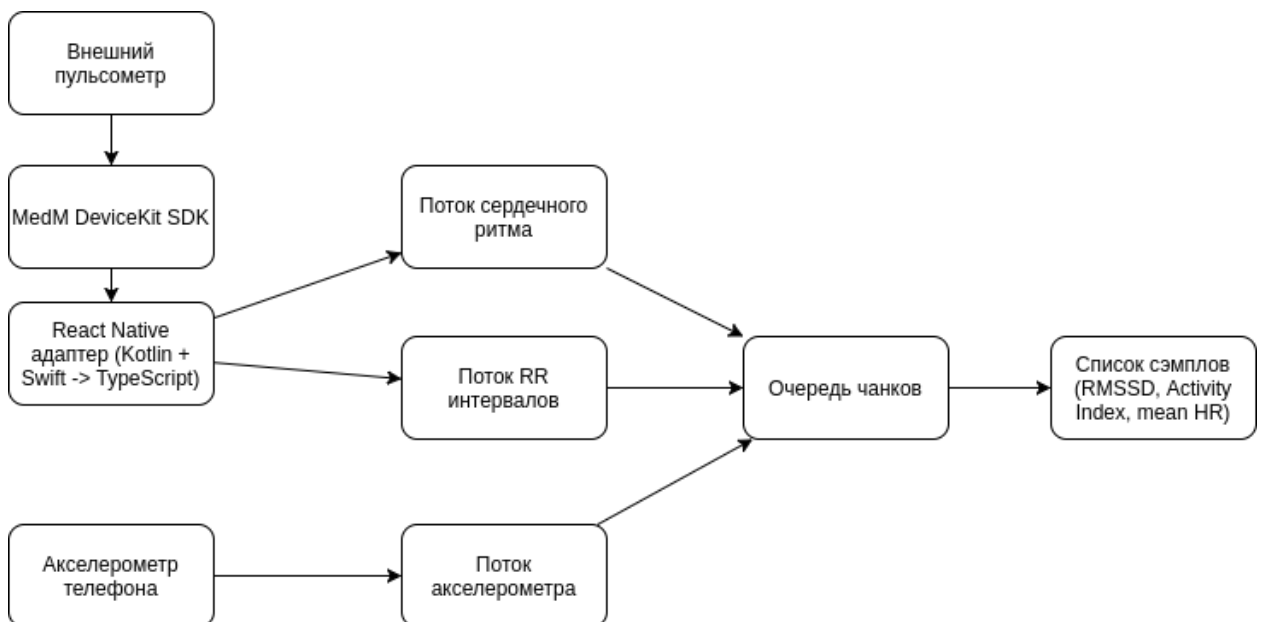


Рис. 1: Взаимодействие с сенсорами

Необработанные данные представляют собой три потока из R-R интервалов, сердечного ритма и показателей ускорения по трем осям мобильного телефона. Данные из этих потоков записываются в соответствующие буферы, которые периодически очищаются через фиксиро-

ванный промежуток времени, а полученные данные записываются в новый чанк (атомарный кусок данных, полученных с сенсоров), который в свою очередь записывается в двустороннюю очередь фиксированного размера. При поступлении нового чанка из очереди удаляется самый старый чанк, а через фиксированный шаг в несколько чанков по основе данных, содержащихся в очереди, считается новое измерение, которое также является элементом выборки для модели машинного обучения. Такая архитектура сбора данных позволяет разбить непрерывный поток данных на пересекающиеся отрезки фиксированного размера и эффективно использовать память, храня только ту часть данных, которая требуется для вычисления следующего измерения.

Перед подсчетом нового измерения данные сортируются по времени, т.к. вычисляемые признаки зависят от их хронологии и существует вероятность, что порядок пакетов данных полученных с пульсометра может быть нарушен из-за плохой связи с сенсором. После сортировки вычисляются главные признаки: вариабельность сердечного ритма, средний сердечный ритм и индекс физической активности. Для подсчета вариабельности сердечного ритма используется среднеквадратичная последовательная разница R-R интервалов (RMSSD), показанная на рис. 2. Для вычисления уровня физической активности используется формула индекса физической активности [9], который считается на основе значений ускорения по трем осям мобильного телефона за промежуток времени (см. рис. 3).

$$\sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (RR_{i+1} - RR_i)^2} \quad (1)$$

Рис. 2: Root Mean Square of Successive Differences

Все признаки вычисляются на основе данных за одинаковый промежуток длиной в 5 минут, что является стандартным размером для вычисления вариабельности сердечного ритма по RMSSD. Стоит заметить, что имеет смысл экспериментировать с более короткими проме-

$$\sqrt{\max(\frac{1}{3}\{\sum_{i=1}^3 \sigma_i^2 - \bar{\sigma}^2\}, 0)} \quad (2)$$

Рис. 3: Индекс физической активности

жутками времени для вычисления других признаков (например, показателя физической активности), но в данной работе признаки считаются на основе всех данных очереди.

3.4. Взаимодействие с моделью

Взаимодействие приложения с моделью машинного обучения заключается в сборе обучающей выборки и последующей адаптации обученной модели для использования во время выполнения приложения. Общая схема процесса обучения модели и его взаимодействия показана на рис. 4.

С целью упрощения работы с обучающей выборкой были написаны следующие скрипты: “pull-samples”, “calc-features-meta”, “regenerate-samples”, “train-model” и “serialize-model”. Чтобы понять, как они встраиваются в процесс взаимодействия с моделью, нужно подробнее рассмотреть сам процесс.

Версия приложения для разработчика предусматривает возможность разметки обучающей выборки и сохранения собранных данных в постоянном хранилище мобильного телефона. Каждая попытка сбора данных порождает отдельную обучающую выборку и сохраняет ее вместе с метками и сырыми данными в отдельную директорию, название которой является идентификатором выборки. После того, как данные были собраны, с помощью скрипта “pull-samples” они загружаются на компьютер, где будет проходить обучение модели.

Стоит заметить, что на этом этапе уже не требуется предварительная обработка (включая стандартизацию) обучающей выборки, т.к. она была выполнена еще во время сбора данных и реализована на TypeScript,

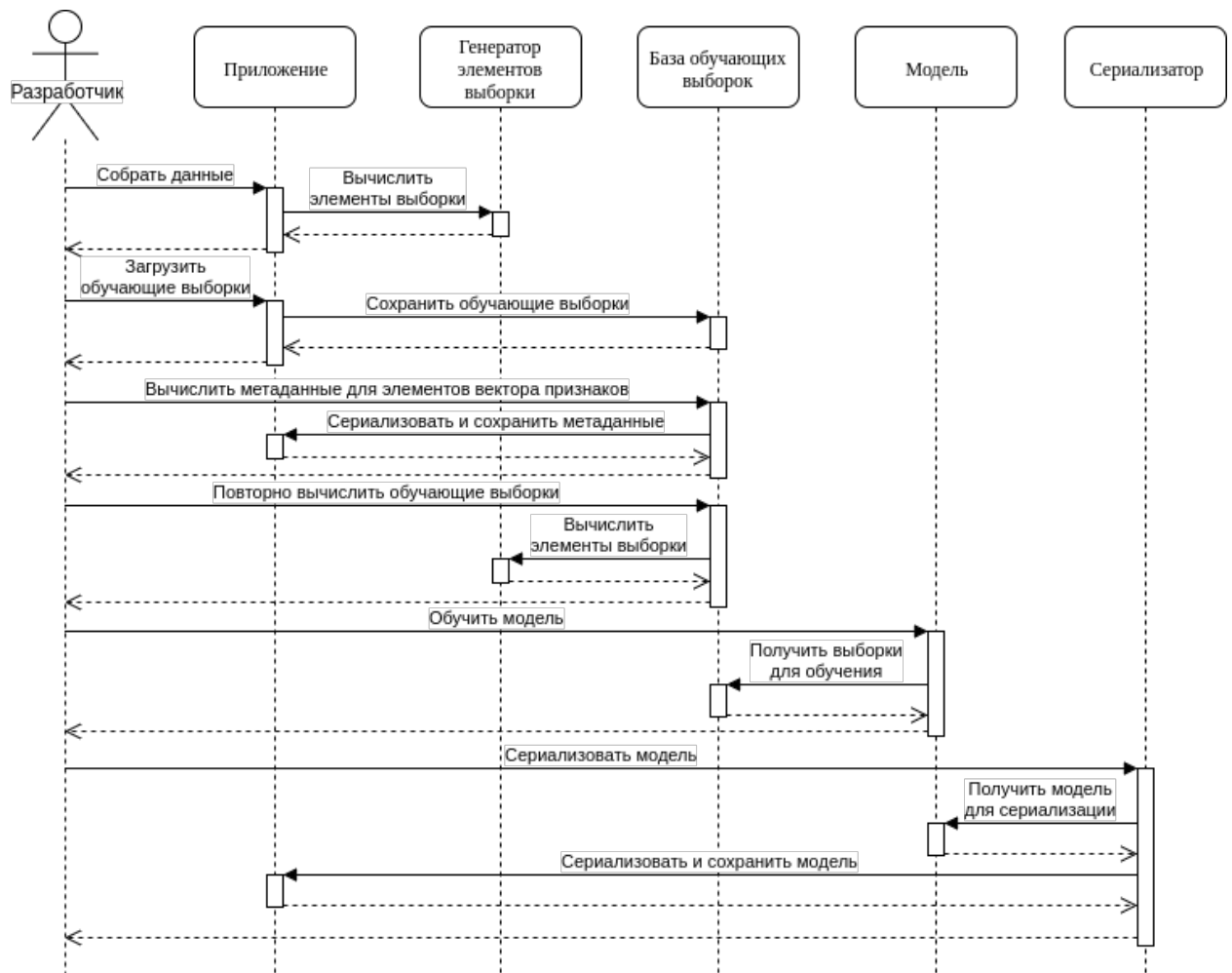


Рис. 4: Взаимодействие с моделью

чтобы не дублировать эту логику в процессе обучения и непосредственного использования модели. Предварительная обработка обучающей выборки состоит в составлении вектора признаков для каждого элемента выборки. Он составляется на основе вычисленных показателей сердцебиения и физической активности и базовых значений, полученных на этапе калибровки, и состоит из отношения HRV к базовому HRV, отношения среднего сердечного ритма к базовому сердечному ритму и индекса физической активности с учетом ошибки акселерометра. Благодаря учету базовых значений достигается большая инвариантность модели к особенностям сердечного ритма конкретного человека.

Далее модель обучается с помощью скрипта “train-model”, передав ему идентификаторы желаемых выборок, и сериализуется в JSON при помощи скрипта “serialize-model” для последующего использования в

приложении. В самом приложении во время выполнения используется не исходная модель, а ее портированный аналог, который настраивается на основе сериализованных параметров модели.

Перед подачей вектора признаков в модель он стандартизируется через стандартизованную оценку (z-score), которая считается на основе стандартных отклонений и средних значений элементов вектора. Эти значения вычисляются с помощью скрипта “calc-features-meta”, который также принимает на вход идентификаторы выборок и сериализует полученные значения в JSON.

Отдельно стоит отметить скрипт “regenerate-samples”, который пересчитывает значения обучающих выборок на основе сохраненных сырых данных и базовых значений. Он переиспользует всю логику вычисления обучающей выборки, благодаря чему он крайне удобен в тех случаях, когда данные уже собраны, но выяснилось, что для эффективной тренировки модели не хватает дополнительных признаков или в вычислении признаков были ошибки (т.е. его можно использовать для генерации обучающей выборки постфактум).

4. Мобильное приложение

4.1. Интерфейс

Интерфейс приложения состоит из трех основных и двух модальных экранов:

- Главный экран используется для приветствия, отображения прогресса сбора первичных данных, необходимых для отображения статистики, и самой статистики (рис. 5(a)). Последние значения главных показателей и состояния стресса находятся в верхней части экрана. При нажатии на значения показателей на графике отрисовывается его тенденция. Ползунок на графике можно использовать для просмотра статистики (главных показателей и продолжительности стресса) по прошедшему времени.
- На экране настроек (рис. 5(b)) отображаются текущий сенсор, базовые значения сердцебиения и ошибка акселерометра. Также через него можно отсоединить текущий сенсор, сбросить значения калибровки и вызвать модальные окна (рис. 5(c)) управления сенсорами и калибровки.
- Экран разработчика (рис. 5(d)) содержит низкоуровневую статистику для отладки приложения и кнопки с уровнями стресса для разметки обучающей выборки.

4.2. Особенности реализации

Несмотря на то, что экраны визуально не связаны между собой, они тесно связаны на уровне данных. Например, главный экран и экран разработчика в реальном времени показывают статистику, вычисленную по одним и тем же данным, а при калибровке и подсчете статистики используются одни и те же механизмы (буферы потоков и вычисление

значений). Поэтому было принято решение разрабатывать пользовательский интерфейс в реактивном стиле, что существенно упростило разработку приложения.

Также для ускорения разработки была добавлена возможность сборки приложения с ускоренным режимом (для этого нужно установить переменную среды “ACCELERATED_MODE”). В этом режиме нет необходимости подключать медицинский сенсор, т.к. он заменяется генератором, который создает все данные автоматически. В этом режиме промежуток времени между вычислениями нового измерения сокращен до минимума, а этап сбора первичных данных вовсе отсутствует.

Версия приложения для разработчика, которая используется для сбора данных и содержит экран разработчика, и версия для пользователя собираются из одной кодовой базы в зависимости от выставленной переменной среды “NODE_ENV”, которая может равняться значениям “production” и “development”.

5. Модель машинного обучения

[Комментарий: эта часть работы еще не доделана, поэтому пока я описал только общий план.]

5.1. Сбор данных

The Trier social stress test (TSST).

Четыре уровня стресса при бинарной классификации.

5.2. Выбор модели

Классификация: SVM, RandomForest, Naive Bayes.

Без меток: K-means, OneClassSVM.

Сведение к регрессии по уровням стресса.

5.3. Оценка эффективности

Статистика по false positive и false negative в задаче классификации.

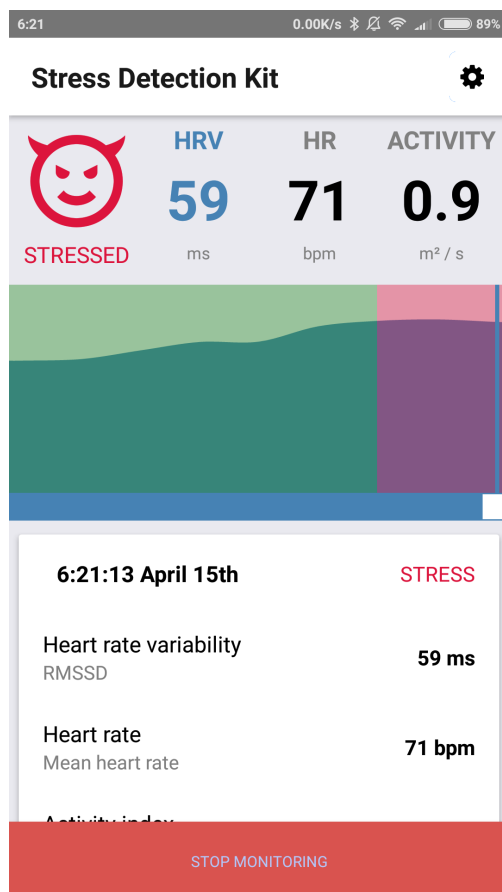
Заключение

В рамках данной работы был создан прототип мобильного приложения для автоматического отслеживания человеческого стресса в реальном времени. Показано, что на основе данных с пульсометра и акселерометра можно эффективно определять стресс в повседневной жизни. Более детально, были получены следующие результаты:

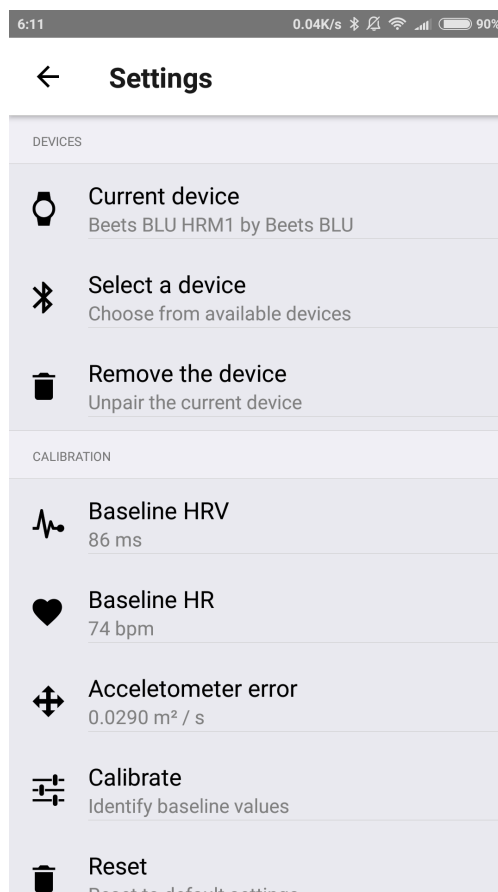
- Проанализированы существующие публикации на тему определения стресса на основе медицинских данных, выявлены недостатки в их подходах и выбраны наиболее важные показатели для определения стресса — вариабельность сердечного ритма и индекс физической активности.
- Спроектирована архитектура приложения, позволяющая переиспользовать логику обработки данных с медицинских сенсоров во время выполнения и в процессе обучения модели, что в свою очередь помогает последовательно и независимо друг от друга разрабатывать мобильное приложение и модель машинного обучения.
- Написано мобильное приложение для сбора данных с возможностью подключить пульсометр по bluetooth и отслеживать изменения в сердцебиении и физической активности на графике в реальном времени.
- Собраны данные для обучения модели и на их основе обучена модель, которая была портирована в TypeScript, чтобы иметь возможность вызывать ее из кода приложения во время его выполнения, не отправляя данные во внешнюю систему.

Список литературы

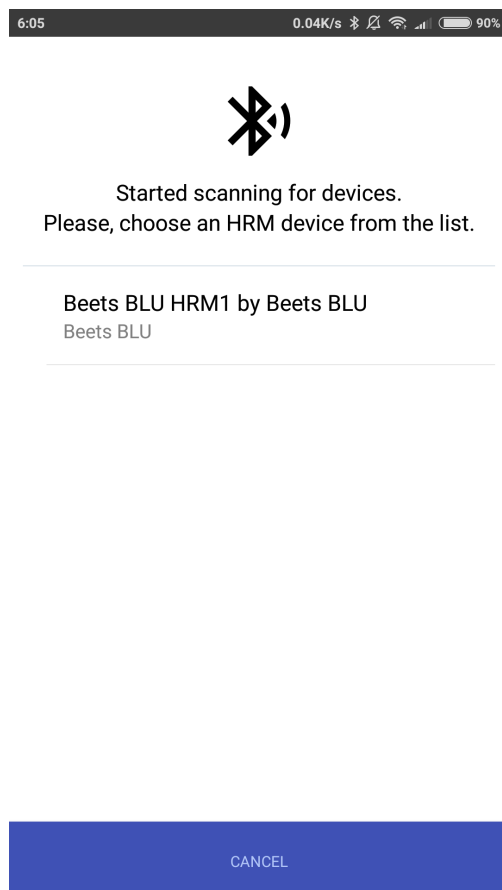
- [1] Selye Hans. The stress of life. McGraw-Hill New York, 1976.
- [2] Dr. Hans Selye M.D. Ph.D. D.Sc. Confusion and Controversy in the Stress Field // Journal of Human Stress. 1975. Т. 1. С. 37–44.
- [3] Razali Salleh Mohd. Life Event, Stress and Illness // The Malaysian journal of medical sciences : MJMS. 2008. Т. 15. С. 9–18.
- [4] Influence of Mental Stress on Heart Rate and Heart Rate Variability / Joachim Taelman, S. Vandeput, A. Spaepen [и др.]. Springer Berlin Heidelberg, 2009. С. 1366–1369.
- [5] Mesquita Ricardo, Kyröläinen Heikki, Schäfer Olstad Daniela. Reliability and validity of time domain heart rate variability during daily routine activities - An alternative to the morning orthostatic test? 2017. 04. Т. 9. С. 64–68.
- [6] Stress-induced changes in human salivary alpha-amylase activity—associations with adrenergic activity / Urs Markus Nater, Roberto La Marca, Ladina Florin [и др.] // Psychoneuroendocrinology. 2006. Т. 31. С. 49–58.
- [7] Salai Mario, Vassányi István, Kósa István. Stress Detection Using Low Cost Heart Rate Sensors // Journal of Healthcare Engineering. 2016. 06. Т. 2016. С. 1–13.
- [8] Modeling perceived stress via HRV and accelerometer sensor streams / M. Wu, H. Cao, H. L. Nguyen [и др.] // 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2015. С. 1625–1628.
- [9] An Activity Index for Raw Accelerometry Data and Its Comparison with Other Activity Metrics / Jiawei Bai, Chongzhi Di, Luo Xiao [и др.] // PLOS ONE. 2016. 08. Т. 11, № 8. С. 1–14. URL: <https://doi.org/10.1371/journal.pone.0160644>.



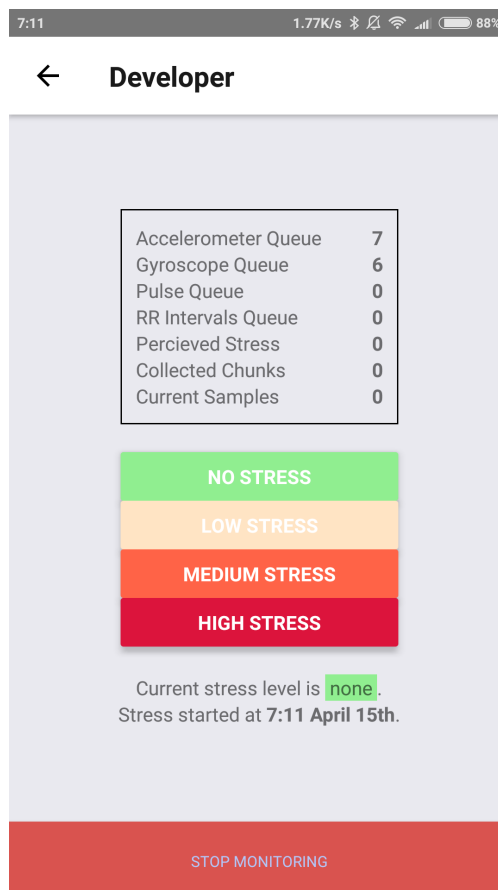
(a) Экран статистики



(b) Экран настроек



(c) Экран управления сенсорами



(d) Экран разработчика