**Sleet Monitor – система оценки заледенелости улиц**

Оглавление

[Описание проекта 1](#_Toc438818050)

[Техническое описание 1](#_Toc438818051)

[Текущий статус 2](#_Toc438818052)

[На данный момент реализовано 2](#_Toc438818053)

[Дальнейшие планы 2](#_Toc438818054)

[Приложение Sleet Monitor Data для сбора данных 2](#_Toc438818055)

[Как выглядят данные? 2](#_Toc438818056)

[Как собирать данные с помощью приложения? 2](#_Toc438818057)

[Рассмотрим процесс записи падения. 4](#_Toc438818058)

[Запись не падения 4](#_Toc438818059)

[Эксперименты 4](#_Toc438818060)

[Приложение 5](#_Toc438818061)

# Описание проекта

**Sleet Monitor** – система оценки заледенелости улиц.

Со множества пользователей, имеющих мобильное приложение Sleet Monitor, собираются данные в реальном времени (показания акселерометра и местоположение) и отправляются в облако Windows Azure. В облаке происходит следующая обработка этих данных - имеется заранее натренированная с помощью алгоритма машинного обучения модель, способная предсказывать по новым данным, свойственны такие показания акселерометра падению человека на льду или нет.

Таким образом, зная информацию о падениях людей, проходящих по той или иной улице в городе, можно поставить оценку этой улице по балловой шкале гололедицы. Люди, находясь дома и планируя выходить куда-то, запускают приложение и выбирают наиболее безопасный маршрут для перемешения из пункта A в пункт B.

# Техническое описание

Для создания системы оценки заледенелости улиц необходимо реализовать следующее:

1. **Облачную алгоритмическую часть**. Это сервер, запущенный в облаке Microsoft Azure (бесплатно по подписке BizSpark). На сервере исполняется алгоритм классификации приходящих с различных устройств данных в реальном времени и устанавливается оценка тому или иному участку дорог в городе по некоторой шкале заледенелости.
2. **Мобильное приложение Sleet Monitor**, которое отображает карту с оценками улиц и отправляет в облако показания акселерометра, собирающиеся во время движения человека по улицам.
3. **Дополнительное приложение для сбора данных**, которые используются для тренировки и тестирования алгоритма машинного обучения (классификатор падений-непадений).
4. **Система проведения экспериментов** для исследования лучшего представления данных (изменение кол-ва осей, по которым собираются показания акселерометра, изменение частоты опрашивания акселерометра, изменение длины фрейма и др.). Цель: выбрать лучший вариант представления данных, находя компромисс между точностью классификации и объемом передаваемых на сервер данных.

# Текущий статус

## На данный момент реализовано

1. Мобильное приложение для сбора данных для проведения экспериментов. Приложение разработано на платформе Android (см. подробное описание ниже)
2. Система проведения экспериметов на базе Python-библиотеки Scikit-Learn (см. описание ниже)

## Дальнейшие планы

1. Провести эксперименты с различными представлениями данных.
2. Реализовать сервер и запустить его в облаке Microsoft Azure.
3. Реализовать пользовательское мобильное приложение Sleet Monitor.

# Приложение Sleet Monitor Data для сбора данных

**Sleet Monitor Data** - это мобильное приложение, задачей которого является сбор тренировочных и тестовых данных для построения и тестирования алгоритма детектирования падений человека на скользкой поверхности.

## Как выглядят данные?

Данные - это показания акселерометра по трем осям с минимально возможным для конкретного устройства SENSOR\_DELAY (задержка прихода новых данных). Показания акселерометра пишутся в файл `Download\SleetMonitorData\data.txt` на флеш-карте устройства (если таковой нет, то во внутреннем хранилище).

Данные представляют из себя строки вида *time: x\_value y\_value \_z\_value ... time: x\_value y\_value \_z\_value is\_fall*, где:

* *time* - время в миллисекундах с момента начала записи состояния
* *x\_value, y\_value, z\_value* - показания акселерометра по соответствующим осям, нормированные по максимально возможному для конкретного типа акселерометра значению
* *is\_fall* - значение true, если в данный момент записывалось падение, false – не падение.

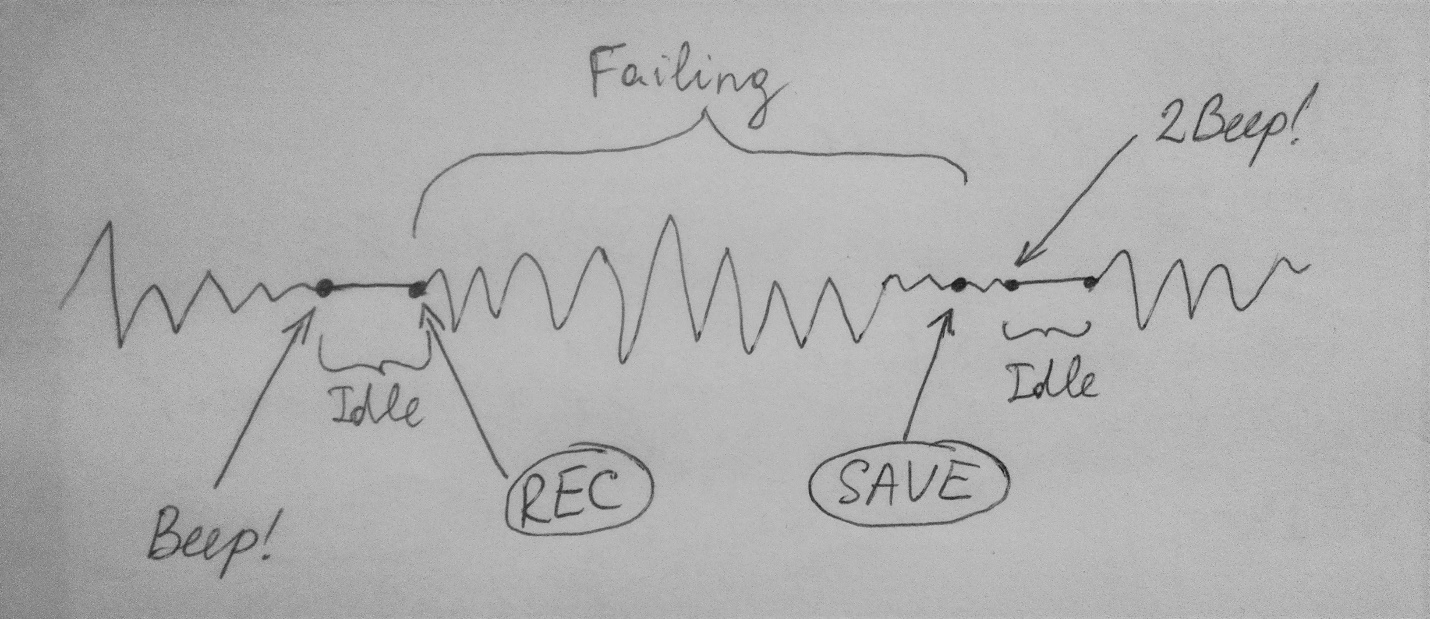
Строка может иметь разную длину на разных устройствах в зависимости от минимального SENSOR\_DELAY на нем, но она всегда ограничена по времени измерения 1.5 сек. Таким образом, если взять устройство Nexus 5, у которого минимальный SENSOR\_DELAY акселерометра равен 5 мсек, то за 1.5 секунды мы получим 300 показаний по трем осям (итого 900 значений для одного измерения). Вообще такая частота прихода новых данных очень благоприятна, так как близка к минимально возможной на всех Android устройствах на данный момент времени. Исходя из этого, сбор данных будет производиться на Nexus 5, хотя может быть сделан и на любом другом смартфоне.

## Как собирать данные с помощью приложения?

Так выглядит основной экран приложения:



После запуска приложения можно выбрать режим записи - либо записывается падение (кнопка **TRACK FALL**), либо непадение (**TRACK NON-FALL**). Чтобы начать запись, нужно нажать на кнопку **PLAY**. Система отслеживания движений включится. Весь цикл работы системы изображен на следующем рисунке.



### Рассмотрим процесс записи падения.

После нажатия на кнопку **PLAY** система ждет, когда колевания смартфона станут незначительными - человек держит телефон в руке, либо кладет его в карман перед началом записи. Как только колебания стали незначительными, система понимает, что человек готов к записи, и она тоже готова - издает коротки звук Beep.

Дальше идет состояние *Idle* (безделие), которое длится до тех пор, пока человек не начнет двигаться, либо не начнет колебать смартфон в руке. Тогда система начинает записывать данные (на рисунке эта стадия называется *Failing*). Запись происходит на протяжении 1.5 секунды и завершается длинным сигналом Beep (система записала данные в файл).

Пока человек лежит, он не двигается (опять стадия *Idle*). Как только человек начал двигаться, а вместе с ним и смартфон в кармане, система понимает, что человек поднимается. С этого момента цикл повторяется. Для записи непадения все работает точно так же.

### Запись не падения

Для переключения на другой режим записи нужно просто нажать на соответствующую кнопку (например, **TRACK-NON-FAIL**). Но перед этим рекомендуется нажать на кнопку **PAUSE**, которая остановит жизненный цикл трекера и отбросит его в начало, а также запретит запись данных до тех пор, пока не будет нажата кнопка **PLAY**.

Вообще кнопка **PAUSE** может использоваться всегда, когда человек считает, что сейчас пишутся некорректные данные (не соответствуют режиму записи). Предусмотрена также система избавления от некорректных данных - кнопки **1**, **2**, **3** предназначены для добавления в конец файла метки *Label1*, *Label2*, *Label3*, сигнализирующих о том, что последние измерения 1, 2 или 3 соответственно некорректны. В дальнейшем исследователь удалит эти данные из файла. Также существует отдельная метка *GOOD*, которую человек может поставить, чтобы сказать, что все предыдущие измерения гарантированно корректны.

Еще на экране имеются поля **Falls** и **Non-Falls**, соответствующие кол-ву записанных измерений в данной сессии приложения по каждому типу состояния. При новом запуске приложения система запишет в файл строчку *DataWriter was initialized*, которая позволит исследователю увидеть новую сессию приложения и в случае ошибок в данных сузить область поиска для их удаления.

# Эксперименты

На данный момент разработана система проведения экспериментов (язык Python). На вход системе подается файл, записанный с помощью приложения Sleet Monitor Data. Система считывает данные и собирает их в виде двух списков: список измерений акселерометра и список меток состояния для всех фреймов. По этим данным производится обучение и тестирование классификатора.

Для классификации был выбран алгоритм **Random Forest Classifier**, реализация которого входит в библиотеку Scikit-Learn.

Все множество данных делится на две части в соотношении 80:20, т.е. 80% данных поступают в классификатор для тренировки, остальные 20% идут на тестирование. Код системы можно увидеть в репозитории проекта (см. Приложение).

На данный момент в качестве характеристики оценки алгоритма используется простое отношение правильно классифицированных фреймов показаний акселерометра к общему числу фреймов.

# Приложение

Общая ссылка на репозиторий проекта: <https://github.com/VladVin/SleetMonitor>

Ссылка на приложение Sleet Monitor Data для сбора данных: <https://github.com/VladVin/SleetMonitor/tree/master/SleetMonitorData>

Ссылка на систему проведения экспериментов: <https://github.com/VladVin/SleetMonitor/tree/master/ml>