**DEBUGGERS**

**Team Members:**

1. **Lakshmipriya R**

1.1. Roll number: 2127200501080

1.2. Branch and Section: CSE - B

1.3. Year: 2nd year

1.4. Contact no.:  9344749849

1.5. Email id: [2020cs0005@svce.ac.in](mailto:2020cs0005@svce.ac.in)

2. **Aathan A**

2.1. Roll number: 2127200501001

2.2. Branch and Section: CSE - A

2.3. Year: 2nd year

2.4. Contact no.:  9080151829

2.5. Email id: [2020cs0167@svce.ac.in](mailto:2020cs0167@svce.ac.in)

3. **Hakash M P**

3.1. Roll number: 2127200501047

3.2. Branch and Section: CSE - A

3.3. Year: 2nd year

3.4. Contact no.: 9363206992

3.5. Email id: [2020cs0263@svce.ac.in](mailto:2020cs0263@svce.ac.in)

4. **Lakshmi Sruthi K**

4.1. Roll number: 2127200501079

4.2. Branch and Section: CSE - B

4.3. Year: 2nd year

4.4. Contact no.: 9176645479

4.5. Email id: [2020cs0165@svce.ac.in](mailto:2020cs0165@svce.ac.in)

**PROBLEM STATEMENT:**

Build a Fake Step Detection System (FSDS) that detects when a user gets steps in illegal ways.

**SUMMARY:**

Step detection and counting has become a challenging task today. There are multiple ways these days where smartphones can be held and used, leading to easier chance of miscount of steps. The smartphone's position and placement might cause jitter during the human movement, which leads to the false counting of the step. Therefore, this work aims to design a robust algorithm for the step counting with high accuracy.

The pre-existing step counting models have chances of miscounting the steps due to unavoidable movements of smartphone even when the user isn’t actually walking. The various other ways in which users try to cheat the algorithm include fixing it with a movable household device, tying it to animals or making it count when moving in vehicles etc. This establishes the validity of the problem statement, i.e., finding out the steps generated illegally.

This fake step detection model detects steps with an existing efficient step detection algorithm and then trains a Machine learning model with the dataset generated while step calculation. Then it undergoes another phase of illegal step filtration which will consider all existing ways in which steps counting algorithm can be manipulated into miscalculation of steps and will rectify it using the datasets derived from the machine learning algorithm.

After the detection of the step, the steps are classified and differentiated as steps counted while travelling in a vehicle, Fake steps (emulated steps) and Legit steps (actual steps counted while walking).

**APPROACH:**

**Data collection phase:**

During this phase the data from accelerometer and gyroscope is collected for the three axes x, y and z. The live location of the user is collected using GPS.

**Filtering phase:**

The derived data is filtered in order to exclude the noise and outlier values. Smoothing the data helps in improving the accuracy of the system as the accelerometer sensor is very sensitive to external movements.

**Step detection:**

The collected data is represented in the form of graphs with respect to dynamic time windows. In the graph, the crest between 2 troughs in the graph is calculated as one step. So, the algorithm keeps track of 3 values, i.e., the start(trough), the end(trough) and the peak(crest) in the graph. When the start and end values are significantly lower than the crest and if the crest is above a threshold value, then the algorithm counts it as a step. After calculating and filtering the steps based on values from accelerometer and gyroscope, the next process is calculating the actual step.

**Fake Step Detection:**

Our aim is to identify actual steps from fake steps which are generated illegally.

**Training ML model:**

With the help of the processed graph, an ANN (Artificial Neural Network) model is trained i.e., classification and regression in the accelerometer graph is analyzed after filtration. The ANN model produces a result in the form of Boolean value (T or F) for each step. Now the accumulated result of the ANN model is taken and compared with the normally detected steps result. With this result the threshold is recalculated, and hence increasing the sustainability of the model. Thus, the model is made prone to data drift and model drift.

**Filtration of miscalculated steps:**

Further the following methods are also integrated to increase the accuracy of the step counts.

* When a person walks, a slight oscillating movement and a vibration is detected in the body of the user. This is detected using the values generated by the gyroscope (x, y, z orientation) and is plotted in a graph. This vibration differs substantially when the user is not walking. So, this graph is compared with already provided graph and with the datasets generated by just swinging the smartphone in hand and not by walking. So, this discards the steps taken without actually walking.
* When the user is travelling in a vehicle, with the help of the data from GPS, the user’s current location can be accessed and the speed at which the device (smartphone) moves is obtained. In case if it is too fast when compared to an average human walking or jogging, the counted steps are discarded and will be counted as the “steps generated due to the motion of vehicle”.
* If the graphs generated using the step detection algorithm has a pattern which is too monotonous to be human-generated, the model declares it as machine-generated steps (fake steps/illegally generated) and discards them. The graphs generated in a duration of time is compared with existing datasets to identify the pattern. The steps must have a gradual increase and decrease of pace at the start and end of the activity to be considered as human-made.
* Any steps generated due to lateral rotation of smartphones will also be discarded since a whole lot of gadgets and users rotate phones in order to generate illegal steps.

**DESIGN AND IMPLEMENTATION:**

Initially, the user’s fitness details such as height, weight and also gender is obtained. The users first few days of walking/running/ jogging is taken to set threshold values and on further use, the algorithm is gradually customized for the user.

* The algorithm first detects the steps and categorizes it based on the position at which the smartphone is held by the user.
* The various user positions considered by the algorithm include holding the phone at hand/wrist watches, phone in shirt/jacket pockets, phone in pant pockets, phone in armbands, phone in purses/handbags, neck pouches while walking, jogging or running.
* The steps are detected using a normal step detection algorithm and then it is used to train an ML model to filter the steps.
* GPS is used to find the speed of the user moving if there is a periodic change in location while detecting steps.
* The machine learning model will study patterns of the person walking and also customize it for each user.

The users shall be advised to keep the smartphones at the optimal positions relative to their body (optimal positions include placing the phone in pockets in an upright manner without rotating it).

**RISKS:**

This model customizes the algorithm based on the walking patterns of the individual users. So, multiple users using a same smartphone with the same account might face a slight reduction in the accuracy in step detection. It is also a case of concern when users give incorrect data about their physical fitness which might influence the customization of the algorithm (in case of calculating step length).

The second case of risk is when a user rotates the phone in hand while walking. This illegal step filtration algorithm filters steps generated by rotation since it can be classified as machine-generated. So, users might face a miscalculation of steps when they rotate their smartphones while walking.

**RECOMMENDED TECHNOLOGIES:**

*i)* **React Native**: It is a JavaScript-based mobile app framework that allows you to build natively-rendered mobile apps for iOS and Android.

*ii)* **Scikit-learn**: Scikit-learn a free software machine learning library for the Python programming language.

*iii)* **TensorFlow**: TensorFlow is a free and open-source software library for machine learning and artificial intelligence.

*iv)* **Node.js**: Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

*v)* **Matplotlib**: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

*vi)* **Pandas**: Pandas a software library written for the Python programming language for data manipulation and analysis.

**PHASE 2 PROCESS:**

In phase 2, the app with Fake Step Detection System (FSDS) will be developed using Machine learning and regression/classification algorithm. The Machine Learning Model is trained with various kinds of data sets in such a way that the model is able to recognize certain types of patterns from the processed graphs derived from the sensors. Therefore, the trained model will be able to differentiate between the steps counted while travelling, Fake steps (manually/illegally generated steps) and Legit steps (actual steps generated by walking).

The steps detected will be filtered in such a way that, when the user tries to trick the step count with the help of external tools, it will be displayed clearly under “Fake Steps”. If the step doesn’t fall under these categories and if it satisfies all the possible conditions of a normal step, then it will be counted in the “Legit Steps”.

However, the problem statement is simple with its requirements, the ways in which it can be compromised is more and hence it is made non vulnerable to fake motion patterns which are very similar to a usual step pattern. As it is a Machine Learning Model, every time when there is a fake step, it is identified and with the help of machine learning algorithm, when the same pattern is observed again, it is counted as a fake step with ease. Therefore, more the step the user walks, more the accuracy is.

The solution is usable across various platforms and various devices. Hence all resources used to implement the solution will be available in almost all the devices and can be accessible in all available platforms. Further with respect to app it can be utilized in various platforms like iOS, Android and WearOS (Android Wear).