Android Application to Detect Texting While Driving (TwD)

sdmay19-16

Client/Advisor: Daji Qiao, Iowa State Professor

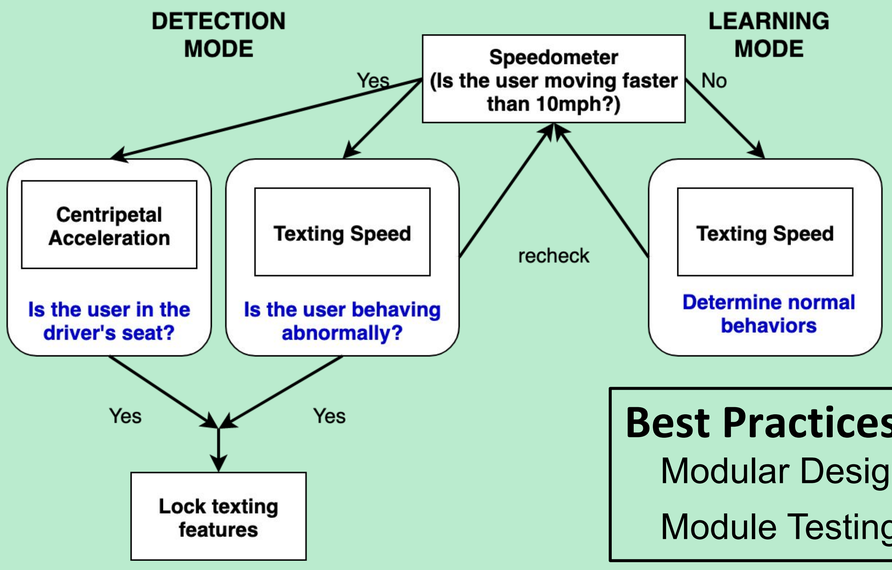
Ryan Baker, Derek Clayton, Lucas Golinghorst, Andrew Knaack, Sara Mace, Kristina Robinson

Project Description:

This project was begun to design and implement an innovative solution to detect and prevent texting while driving. Built-in sensors and APIs in Android phones were used to create 3 modules:

1. Centripetal Acceleration - determines which side of the vehicle the user is sitting in
2. Texting Speed - calculates average texting speed to see if the user has unexpected behaviors
3. location services to determine if the phone is moving fast enough to imply the user is in a vehicle

The goal of this project was to look into new ways to detect TwD



Best Practices:

Modular Design

Module Testing

System Architecture and Design:

We have 3 main tests to see if the phone user is driving.

1. The first thing the application checks for is the speed that the phone is going. This determines if the user is driving (going over 10 mph) or not. If the user is driving, then additional testing can occur.
2. The texting speed text is based off the assumption that the user will be multitasking when texting and driving. If the user's texting speed falls below the average by 70% of normal speed, the application will conclude that the user is texting and drive.
3. The centripetal acceleration test determines the user's location in the car based off how the user is accelerated around turns when driving.

Module Testing:

Speedometer

- Programmed the app to keep a printed log of all recorded speeds
- Took a second person driving (to avoid being too distracted) and monitored the readout, comparing it to live speeds
- Test zones included downtown Ames, Des Moines, and I-35

Texting Speed

- Used Mario Kart 8 DX for Nintendo Switch as a substitute for actually texting and driving. Slower carts and the most level, plain map was used to approximate realism.
- Attempted to text predetermined messages without crashing
- Data resulted in the "distraction threshold" speed of 70%

Centripetal Acceleration:

- Tested gyroscope reading and magnetometer readings
- Selected a route of four turns, passenger held the car either on passenger side or driver side while driver traveled the route multiple times to produce at least ten turns for one trial.
- Conducted multiple trials to establish thresholds for determining whether user is driver or passenger.
- The application succeeded in determining left or right turns nearly 100% of the time.

Functional Requirements:

1. The application prevents drivers from sending texts if certain dangerous conditions are met.
2. The false positive rate and false negative rate should not be greater than 10%.
3. The application should run on Android OS 6.0 and newer.
4. The app does not use an Internet connection (it is a standalone app). This is necessary if service is poor and to avoid breaking any applicable data handling laws with outside data storage.

NonFunctional Requirements:

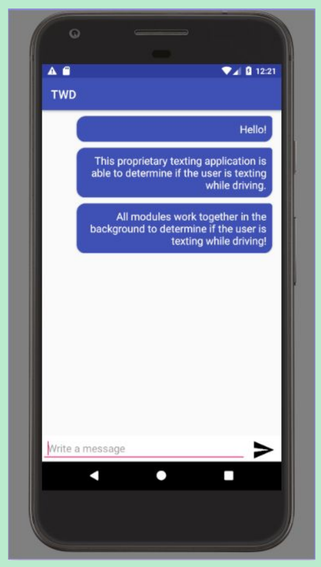
1. The app itself will be easy to set up and run.
2. The app must be reliable at least 95% of the time, meaning that it runs consistently without crashing.
3. The app must comply with applicable sensitive data laws (presumably resolved by making it a standalone app).

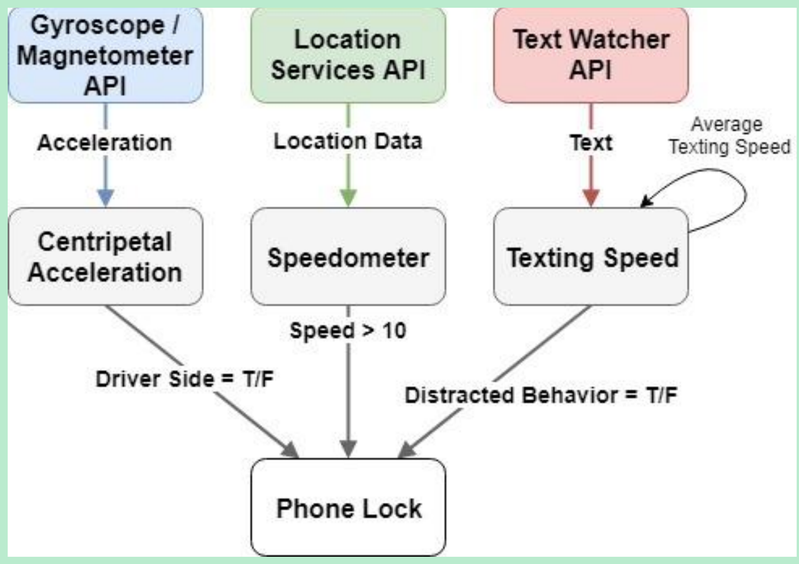
System Implementation:

1. Development Environment: Android Studio
2. **Hardware**
   a. Mobile Phone: Our application uses the gyroscope and magnetometer embedded on the mobile phone.
3. **Software**
   a. Location Services: Our application uses location services to determine the current speed of the phone. Speed is not accurately provided directly from the services, so its other data is used to calculate a speed instead.
   b. Textwatcher: This API is used to help determine the texting speed of the user by monitoring the textbox into which users type their text messages.

Standards:

1. **IEEE 16085-2006 -ISO/IEC 16085:2006, Standard for software engineering- software life cycle processes -risk management**
2. **IEEE Standard Specification Format Guide and Test Procedure for Non-gyroscopic Inertial Angular Sensors: Jerk, Acceleration, Velocity, and Displacement**
3. **29119-3-2013 - ISO/IEC/IEEE International Standard - Software and systems engineering -- Software testing --Part 3: Test documentation**





Integration and System Testing:

Integration

- Checked that each module individually would block the user from texting when integrated into the texting application.
- All modules were able to block texting functionality.

System

- Once all modules were integrated, we went on test drives to make sure that when the appropriate conditions were met, the application would block texting.
- All modules working together in the background successfully distinguished between passenger and driver and blocked texting capabilities for driver.

Conclusion:

This project found that it is possible to determine that someone is texting while driving exclusively using available Android sensors and APIs. For future improvements to this project we would make the modules implement more machine learning techniques. Machine learning techniques would take into account different phones and different users. The app *does* successfully and consistently determine if a user is on the passenger or driver side of the car, their texting behavior, and their speed. In future we would add in the ability to determine if a user is sitting in the back or the front. Through all our testing and our original plan of having 6 modules we determined that only 3 modules were feasible in this project, as the other 3 were not as effective as we hoped they would be.