



Boston University
Electrical & Computer Engineering
EC464 Senior Design Project
Final Prototype Testing Plan



by
Team 8
BikeGuard

Beren Donmez: bydonmez@bu.edu

Margherita Piana: mpiana@bu.edu

Marissa Ruiz: mbruiz@bu.edu

Bennett Taylor: betaylor@bu.edu

Albert Zhao: albertz@bu.edu

Required Materials

Hardware	Software
<p>Onboard:</p> <ul style="list-style-type: none">• Raspberry Pi 4 (with 32GB SanDisk SDHC Class 10 card)• Piezo Buzzer 110 db• IRLZ44N Mosfet• MPU-6050 (Accelerometer)• INIU BI-B61 Portable Charger 22.5W 10000mAh Power Bank• Raspberry Pi Camera Module V3 NoIR Wide• 12mm On/Off Metal Key Lock Switch• TP-Link Router• Small breadboard• Electrical Tape• Jumper Cable• USB-C to micro USB cable• USB C Extension Cable 1ft, 1-Pack, USB C 3.2 Male to USB C Female Cable• 2-Pack 1.5ft Short Braided USB 3.0 A to A Cable - Male to Male• Small heat sinks• 3D Printed Enclosure + lid <p>Remote:</p> <ul style="list-style-type: none">• Laptop	<p>Raspberry Pi:</p> <ul style="list-style-type: none">• Raspberry Pi OS 64 Bit, Bookworm• Python3 -> accelerometer reading, writing to CSV file, sending camera stream to flask server, sending post notification to backend, thread management, setting off buzzer• Flask -> pi camera stream <p>Front End:</p> <ul style="list-style-type: none">• Javascript and React• CSS for formatting• Flask for connecting to the backend <p>Back End:</p> <ul style="list-style-type: none">• Flask• SQL for data storage <p>Machine Learning Model:</p> <ul style="list-style-type: none">• Logistic regression model• Detect theft based on the accelerometer data that Assign labels “0” for normal and “1” for theft• Extract_features() method• Train_test_split() method• StandardScaler() for better performance• Train model with LogisticRegression.• True Positives (TP): 49, False Positives (FP): 3, False Negatives (FN): 6, True Negatives (TN): 18 -> Accuracy = ≈ 0.8816

Set Up:

Our hardware setup includes a Key Lock Switch, a Raspberry Pi 4, a corresponding Raspberry Pi Camera module V3 NoIR Wide, an accelerometer, and a buzzer. We use a router to establish our network within the broader BU wifi. We connected the Raspberry Pi to the network and are hosting all our servers on the network. We use SSH to access the Raspberry Pi remotely. We are powering the Raspberry PI with a portable power bank that is intended to charge phones and tablets. We additionally attached small heat sinks to our Raspberry Pi to prevent it from overheating. Once the Raspberry Pi is powered on, a `boot_try.py` script is automatically run. The `boot_try.py` script listens for GPIO signals from a Key Lock switch, which activates all of our software once the Key Lock is turned to the active position. This was implemented to save power. Our components are housed in a 3D printed enclosure attached to the bike with clamps. The accelerometer's data are saved in a CSV file using python libraries. In this manner, the data can be used to train the machine learning model to recognize possible bike theft. Once the pitch and roll reach a certain threshold while shaking, the buzzer goes off. Once the buzzer goes off it can be stopped through a button on the website. The backend file `app.py` receives constant information from the raspberry pi `accelerometer.py` folder. If the accelerometer detects motion greater than a pre-set threshold, it sends a push request to the backend and stores the message in the SQL database. Once the front end detects new changes in the database, it displays the most recent message on our website. As for the camera live view, the Raspberry Pi already has a module to convert real video feed into mpng (different format of png), then using flask we can stream the video feed directly to the front end.

Pre-testing Setup Procedure:

Raspberry Pi:

1. Connect and turn on the router.
2. Switch key switch to ON (green color)

Server/Web UI:

3. Enable virtual environment for new systems
4. Make sure system is connected to router
5. Go to back end directory and start flask “python3 app.py”
6. Go to the front end folder and run “npm start”

Testing Procedure:

1. Mount the enclosure to your bike
2. To activate the theft detection system, simply turn the Key Lock switch to the 'on' position.
3. The camera is being consistently streamed on the front-end website
4. Once the bike gets shaken (is getting stolen) the buzzer goes off and a notification will be seen on the website
5. Once the bike is not shaken anymore the buzzer should stop immediately.
6. All the data are collected and saved in a CSV file that can be used in the machine learning model.
7. The machine learning algorithm utilizes linear classification to train the model and correctly classify new data

Measurable Criteria:

The criteria for a successful running are the following:

1. The enclosure is resistant and does not obstruct the normal function of the bike.
2. When the bike is shaken the buzzer goes off immediately.
3. The buzzer can be turned off through a button on the website.
4. When the buzzer goes off a message is printed on the UI.
5. Video stream is correctly visible on our website.
6. Once the machine learning model is trained a low error is expected as well as a good confusion matrix, meaning that possible theft is correctly classified.

Hardware Pins:

Raspberry Pi 4 pins	Usage/Description
GPIO 2 Serial Data(I2C) Pin# : 3	SDA pin used for I2C communication between accelerometer (MPU) and raspberry pi
GPIO 3 Serial Data(I2C) Pin# : 5	SCL pin used for I2C communication between accelerometer (MPU) and raspberry pi
Power Pin (3V3) Pin# :1	Powers the accelerometer
Ground Pin Pin# :9	Ground for accelerometer

