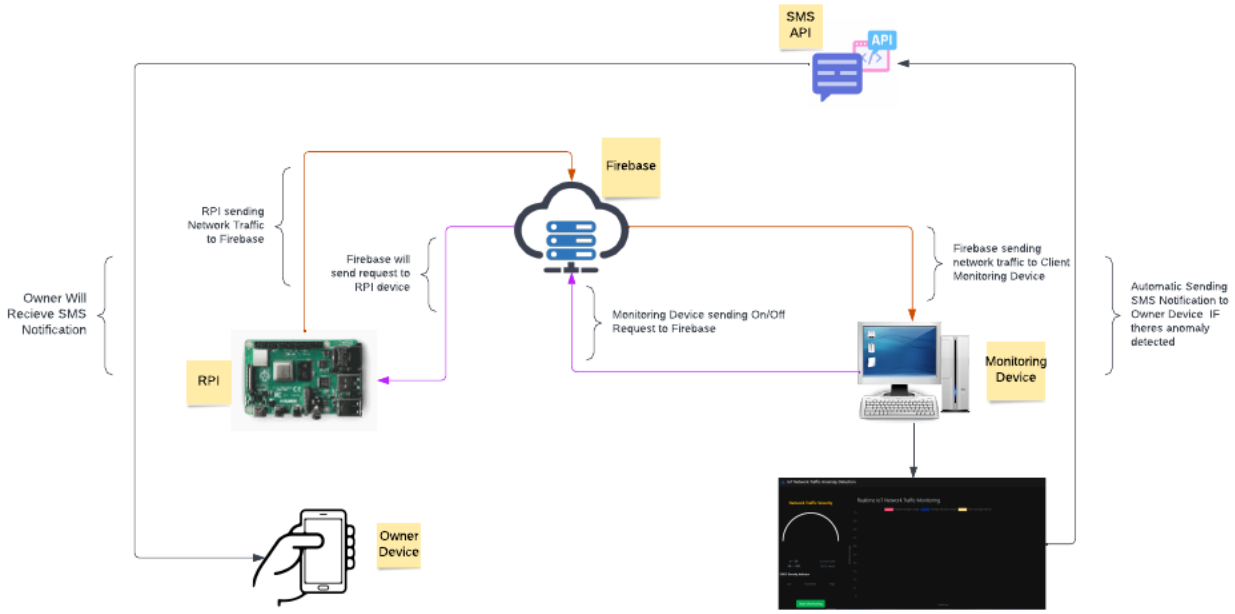
The diagram below shows how the entire system simulation works. The technique below is solely intended to demonstrate how simple it is to use Firebase as a Realtime cloud database for Realtime services. I also included the RPI and python files for the web application.



Conceptual Framework Process Overview

Web Application Source Code

Ignore the security warnings because they want me to use a newer version of Pillow for model integration, but the current version only works with the joblib file. Also, that version isn't compatible with the Python version I'm using, so you'll have to upgrade at your own risk.

https://github.com/Senpaixyz/AnomalyDetection-WebApplication

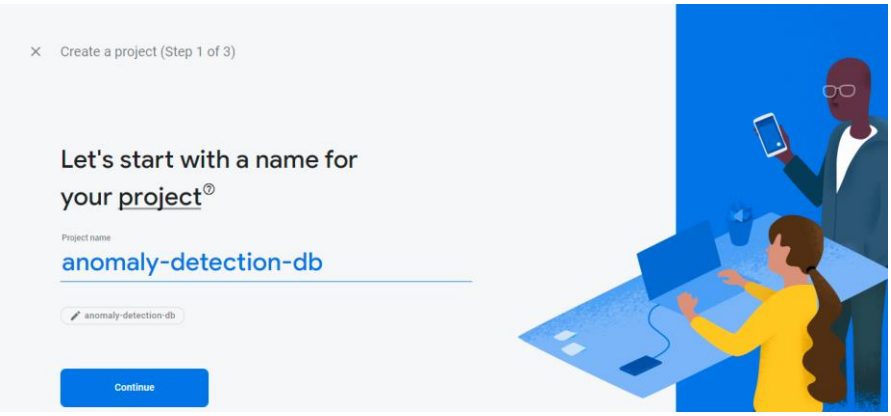
Monitoring Mode Source Code

https://github.com/Senpaixyz/Python-3-6-8-RPI.git

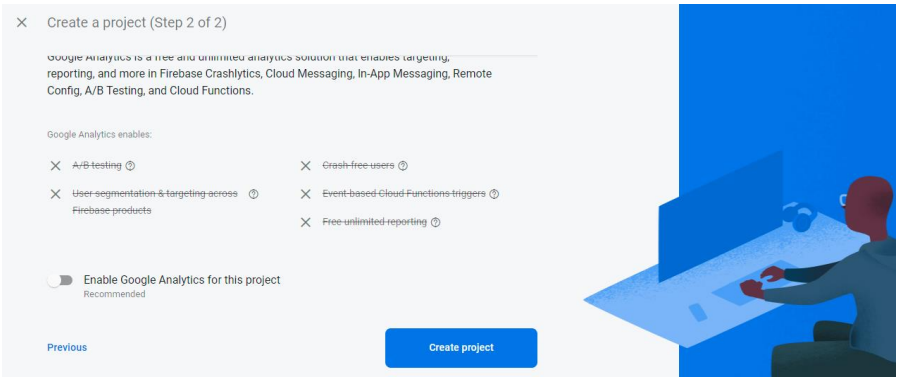
Setting Up Firebase Realtime Database

Create a new Firebase account with your own Gmail or Email address, and then click the 'Add Project' button to start building your own database.

Enter the name of your project and double-check that the database you're about to create isn't the same as any of your prior databases.



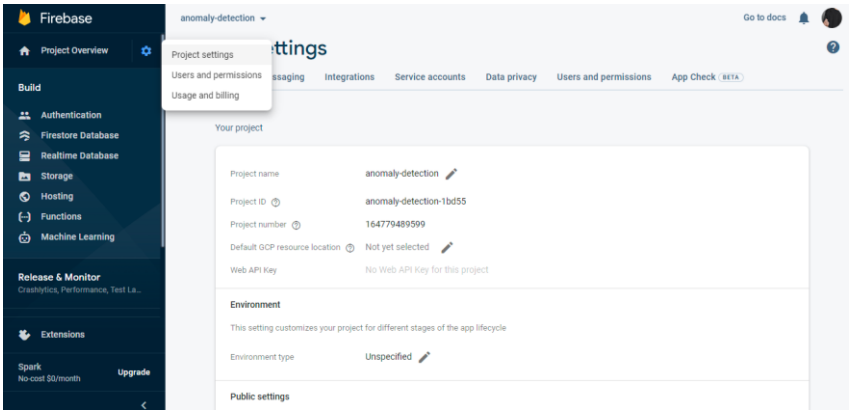
Turned off Google Analytics and pressed the Create Project option.



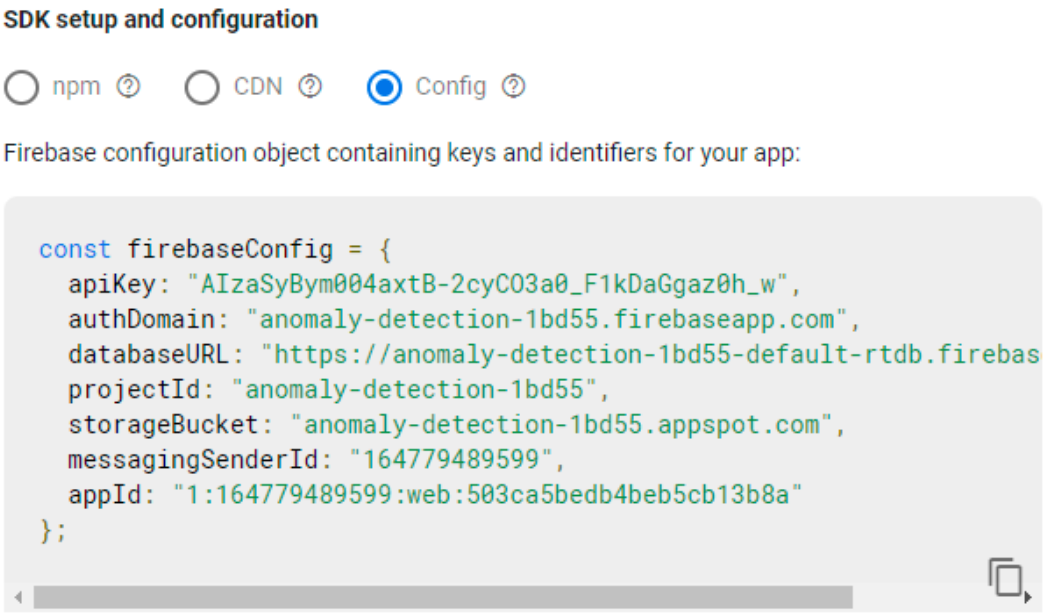
Modify the object key that is required in this project inside the Realtime Database tab. Verify that the object key corresponds to the image below. You simply need to make three objects, and you can ignore all of the object values that were displayed. Only the **Network-Active**, **Network-Connection**, and **Network-Traffic** object keys are required.

RPI- Network Traffic Sender Configuration

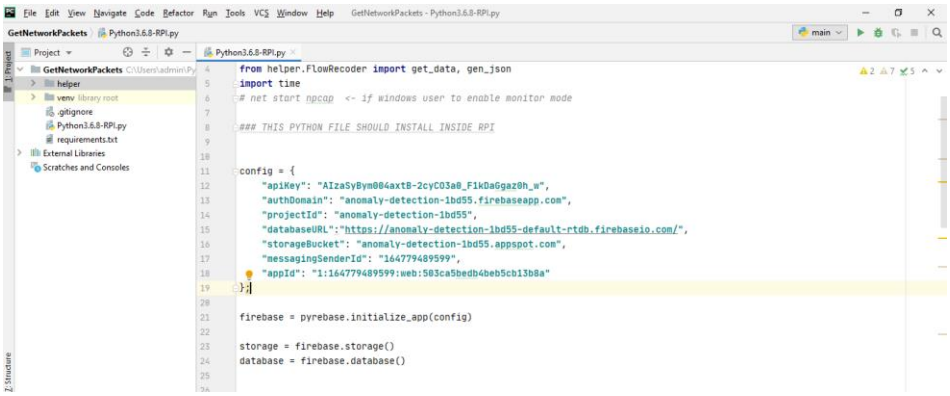
Select the Project Settings by clicking the 'Gear Favicon.' Scroll down to the Apps Configuration section.



Select Config from SDK Setup Configuration. Go to Python3.6.8-RPI.py and paste the firebase API configuration. Your Python3.6.8-RPI.py file enables monitoring mode on your RPI device. Furthermore, please avoid putting API keys into the python code, as I did, and instead use environment variables from the cloud hosting server for future features of this system.



Paste your Firebase Configuration into the Python file. Ensure that firebaseConfig is set to config. Use environment variables for confidential keys, as seen in the image below, for best practices or the future of this system.



RPI- Network Traffic Python Dependencies Installation.

You have two choices: establish a new virtual environment or use the device's default environment. Additionally, ensure that your RPI has root access. To make this task easier, I generated the requirements.txt file, which you can use to install all of the required dependencies. By the way, the version of Python I'm using to run the system is Python 3.6.8. All you have to do is run the command below.

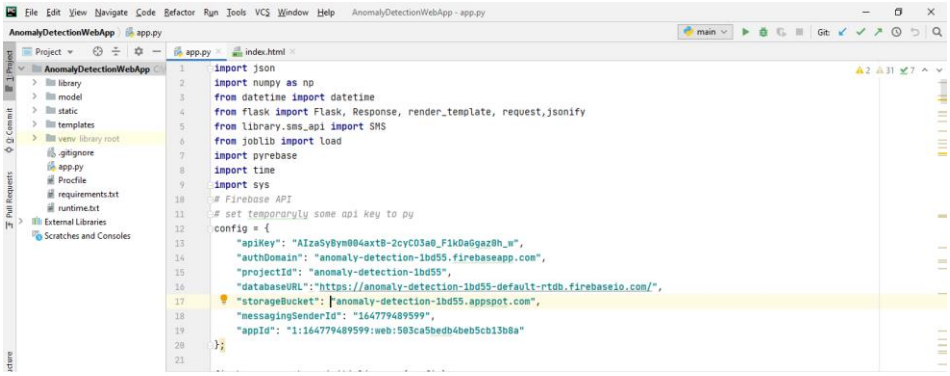
pip3 install -r requirements.txt

Web Application Python Dependencies Installation

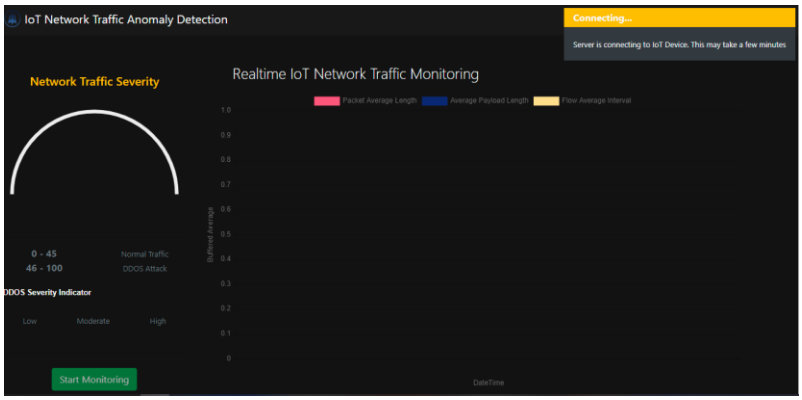
Kindly remind to rename firebaseConfig to config. Install all of the essential libraries using the terminal.

pip3 install -r requirements.txt

Copy and paste your Firebase Configuration into the Python file as you did before.

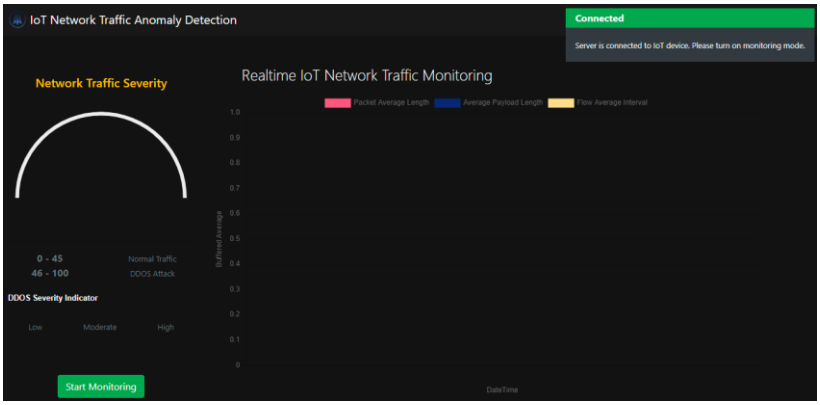


Execute the web application using the command "python app.py" in the terminal to test network traffic anomaly detection. By opening your browser, go to http://127.0.0.1:5000/. Following that, the web application will attempt to connect to the RPI device over the cloud.



You must run the python file from the RPI device in order for the network monitoring tools to function. To avoid an unexpected error, go to Python3.6.8-RPI.py and ensure that you installed all of the dependencies and configuration that are required. After opening the terminal and entering the command "python Python3.6.8-RPI.py," the device will take a few seconds to boot and monitor.

When you return to your web application after that successful configuration, the status will have changed to Connected. Press the Start Monitoring button to begin the monitoring simulation.



If the device detects an anomaly, the toast notification will automatically notify you. The network notification changed from Yellow to Red depending on the severity of the attack. If the attack is deemed to be 80 percent severe, the application will automatically send a notification to the Owner's device.