Detecting Abnormal Behaviors in Smart Home
Code Instruction

Group 2

Crawler for Collecting Data

This script is written with python and Scrapy framework.

Before running this, you need to have python 2.7 and Scrapy framework installed on your device.

Then, you should edit the *base_url* variable to the address of your console's event list page.

When you have all things above ready, cd to where you store this script and use *scrapy crawl event -o [filename]* to run this script and save collected data into your .csv file.

Classifier Model

One-class SVM

The one-class SVM model is in the source file oneclasssvm.py. The machine learning library sklearn is necessary to run the code, and a trained model will be saved into model.p. A trained model could be loaded and make prediction using the class OneClassSVM_Ab in file ab_one_class_svm.py

Autoencoder

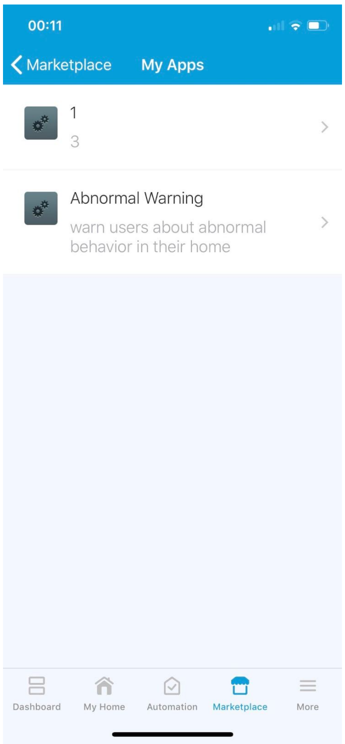
The autoencoder model is in the source file autoencoder.py. We use the normal training data to build the autoencoder model. For this we make use of Keras with tensorflow backend, Pandas and Numpy. We train our autoencder model, set the threshold on a hold-out validation set, and then build our classifier based on this trained model and the threshold.

SmartApp

The SmartApp is deployed in the SmartThings web console. Currently, to simplify the process, we only published our SmartApp to our own Samsung account, which means that only the app login with our account has the access to the SmartApp and install it. Thus, if you want to successfully replay the real situation experiment, you have to set up all the IoT sensors and hub, install the app, create your account, and use our code to publish the SmartApp to your account and install it.

At first, you need to set up the SmartThings IoT sensors and hub in your room. And then, you can download and install SmartThings control app from IOS or Android store. You can install "SmartThings (Samsung Connect)" or "SmartThings Classic". Both of them can configure your devices. But only the latter one can install SmartApp plugin. Then you can create your account and configure it. After you configure the app, you need to make sure the hub can connect to the Internet and your sensors. You can also monitor all of them in the

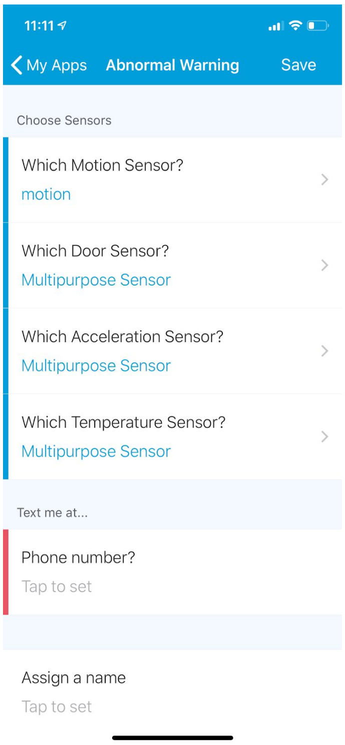
web console, https://graph-na04-useast2.api.smartthings.com. It contains all the information about your account, including the devices, location scenarios, SmartApps, and data log.
The SmartApp installation process is like below:



1
3

Abnormal Warning
warn users about abnormal behavior in their home

Figure 1



Choose Sensors

Which Motion Sensor?
motion

Which Door Sensor?
Multipurpose Sensor

Which Acceleration Sensor?
Multipurpose Sensor

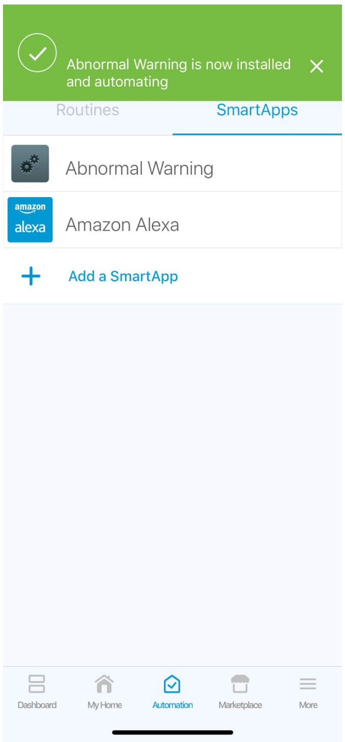
Which Temperature Sensor?
Multipurpose Sensor

Text me at...

Phone number?
Tap to set

Assign a name
Tap to set

Figure 2



Abnormal Warning is now installed and automating

Abnormal Warning
Amazon Alexa

Figure 3

Figure 1: find the SmartApp in the marketplace, “Abnormal Warning”.
Figure 2: choose the proper sensors and provide the mobile phone number.
Figure 3: the SmartApp is installed successfully.

Cloud API

The cloud API is built using Python’s Flask web framework. The source is in the file named app.py. In this file, we expose an HTTP POST API handler named get_status which accepts POST requests sent from the smartapp, and calls our model on the data point to get the prediction (normal or abnormal). We send this prediction to the smartapp as our response to the POST request.

To run the app, please install the dependencies using:
pip install -r requirements.txt

Then run the app using:
python app.py