OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FACULTY OF COMPUTER SCIENCE

# IoT Based Smart Home Environment using SCION

## Burglar detection with motion sensors using MQTT, data analysis and machine learning over SCION

Guided by:
Prof. David Hausheer

Team
Abhijith Remesh (221424)
Manish RamaChandran (221423)

## Table of Contents

## Use Case Context

Current Scenario:

Most of the Burglar detection systems employs the use of camera which causes

- Insecurity to the users
- Privacy of the users disturbed.

Proposed Idea :

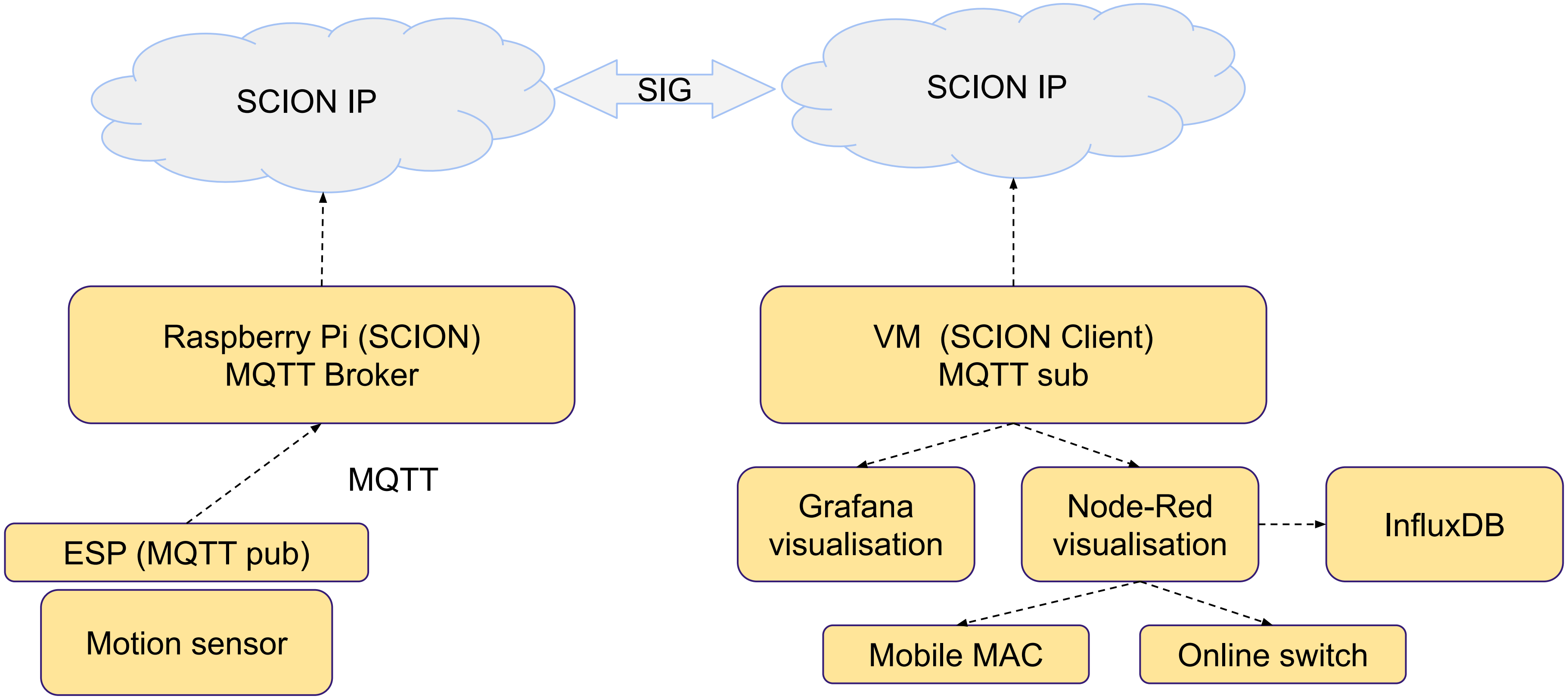
To identify the presence of a intruder in our home  from a remote location without the use of camera, rather using motion sensors.

The significant data components includes

- Motion detections and respective timestamps
- Online Switch value configurable by the user.
- Mobile MAC value ( additional)

Merits : Privacy not disturbed !!!!

## Overview : Application Schema



## Overview : Key Components

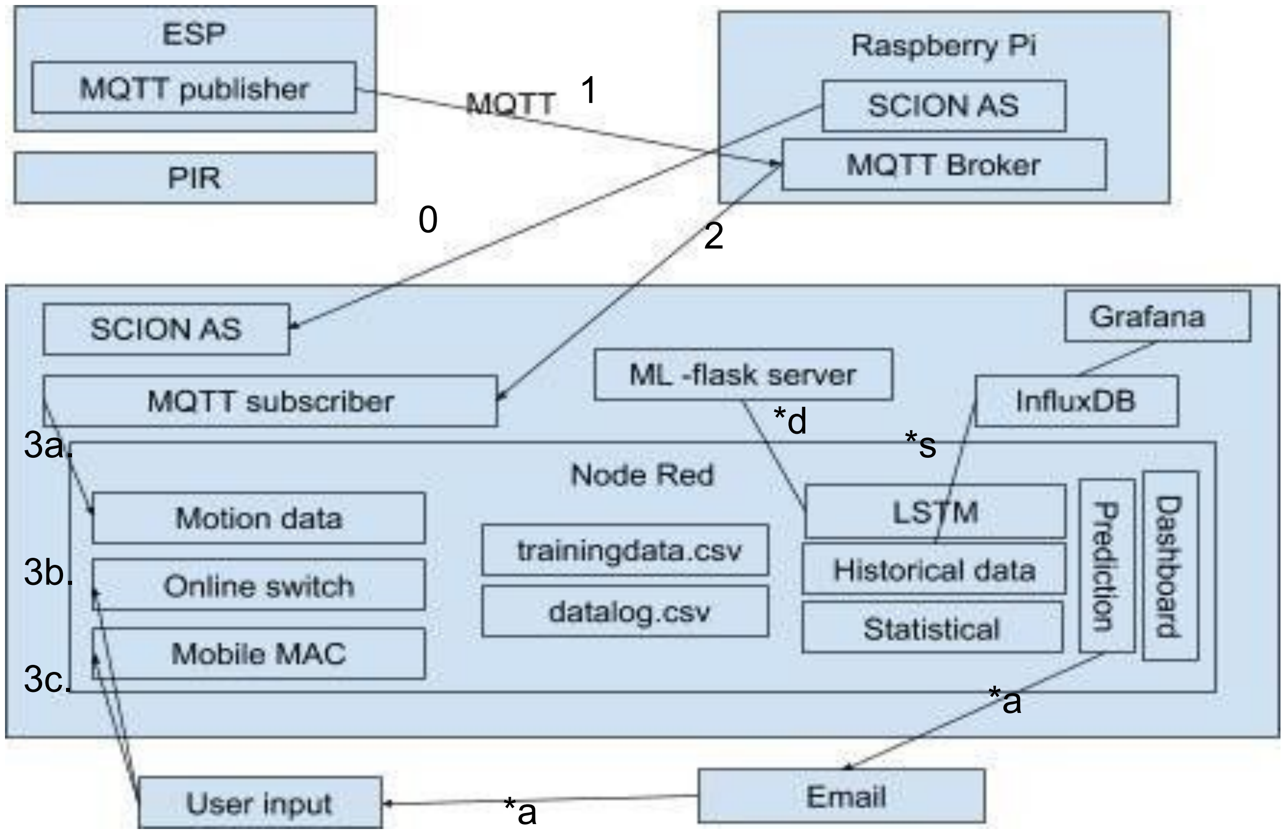
- Motion sensor coupled with ESP8266 ( Smart Battery management, WiFi manager) which acts as the MQTT pub and sends ‘Motion/Voltage’ on every detection.

- Raspberry Pi with SCION AS and a MQTT broker installed which forwards the detection message to the client SCION AS (VM) through the SIG ( Scion Internet Gateway).

- The client SCION AS as a MQTT sub subscribes on the same topic and receives the message, undergo several data analysis and machine learning approaches and provides Node-RED visualisation.

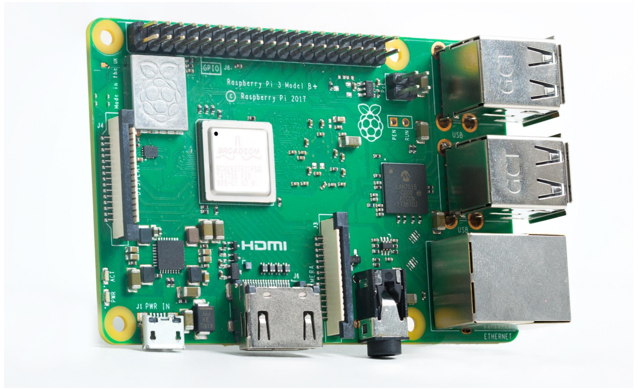
- The Node-RED dashboard provides the online switch functionality and also the mobile MAC presence identification functionality.

- The Node-RED uploads the data logged as csv onto the InfluxDB instance.



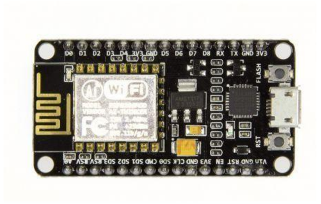
*d : detections
*s : schedule
*a : abnormal

## Hardware and Software Utilities


Raspberry Pi 3 B+ Model

SCION

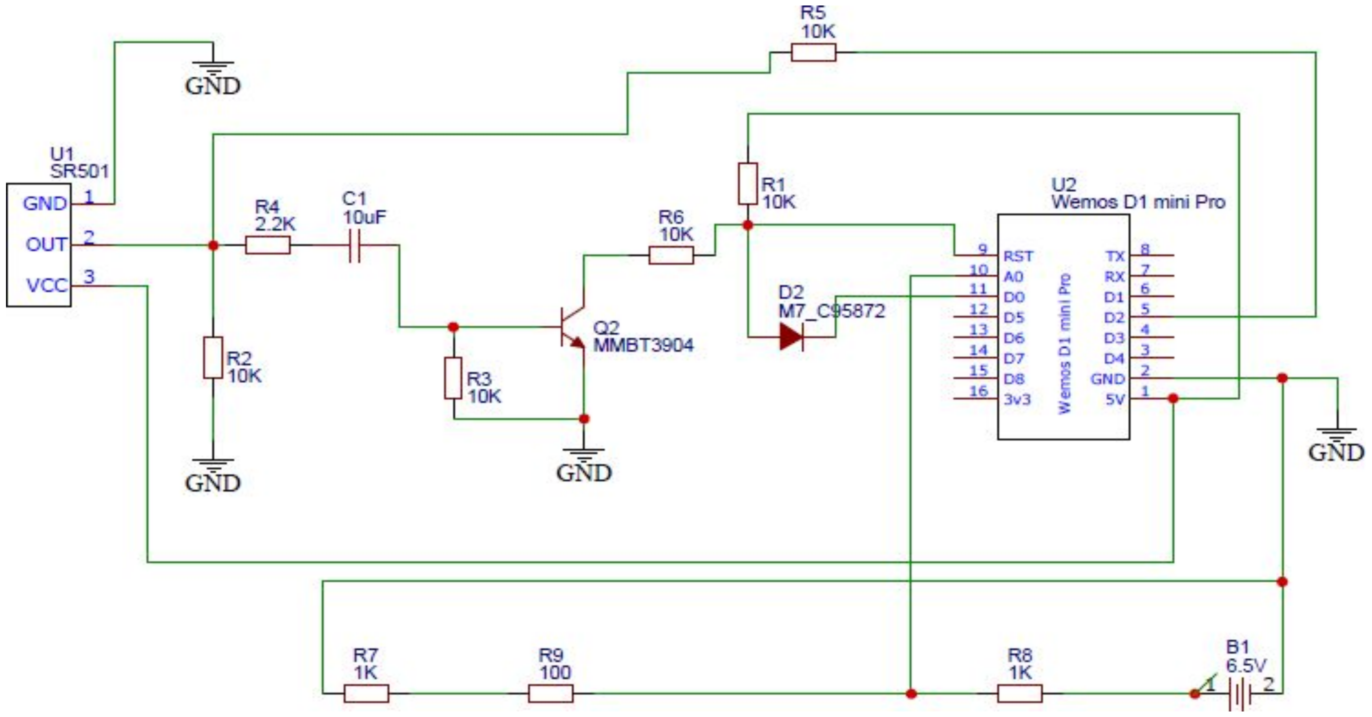
HC-SR501


ESP 8266

influxdb

Node-RED

## Hardware Interfacing



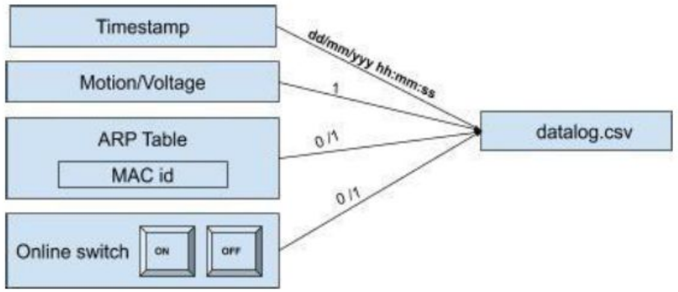
- The current PCB circuit puts the ESP in deep sleep mode when not in use as a part of Battery Management.
- The module wakes with an interrupt, publishes a MQTT message and goes back to sleep.
- The easy set up of WiFi connection with a 2 way switch  in ESP module using WifiManager library to avoid repetitive programming when we detect new WiFi connection.
- The ESP can be connected to a battery setup or direct power.
- The ESP can last for 8 days if there is 80 motion in average  in an hour and if the battery capacity is around 2900mAh or more as per our calculations in Testing phase.
- Because the MQTT message from ESP is “Motion/(batteryvoltage)”, the user in the client side get continuous notification about the battery health via Node-RED UI. Alerts is also set when the battery voltage steps down below 4.2V.

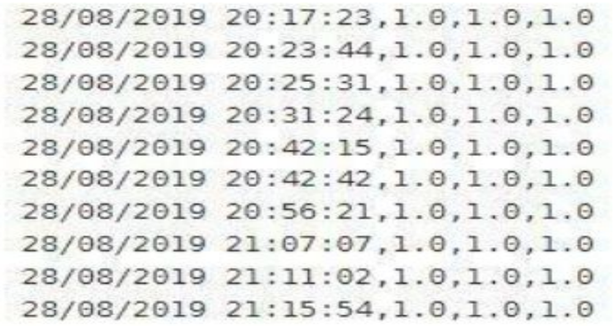
## Data Analysis Approaches for Burglar Detection

- Historical data analysis
- Artificial Intelligence methods
- Statistical methods

## Historical data analysis

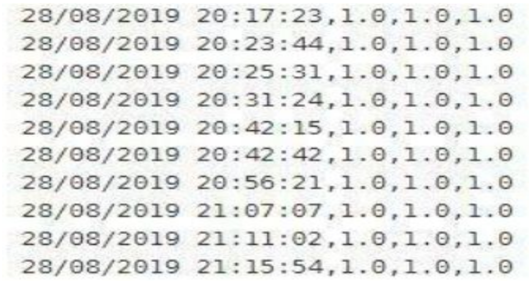
- Logging Data to a local csv file (datalog.csv)
    - Timestamp, Motion value , online switch value, mobile mac value
    - It gets refreshed weekly.



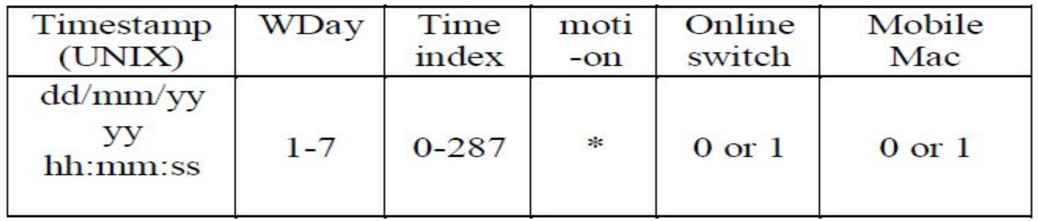


```
28/08/2019 20:17:23,1.0,1.0,1.0
28/08/2019 20:23:44,1.0,1.0,1.0
28/08/2019 20:25:31,1.0,1.0,1.0
28/08/2019 20:31:24,1.0,1.0,1.0
28/08/2019 20:42:15,1.0,1.0,1.0
28/08/2019 20:42:42,1.0,1.0,1.0
28/08/2019 20:56:21,1.0,1.0,1.0
28/08/2019 21:07:07,1.0,1.0,1.0
28/08/2019 21:11:02,1.0,1.0,1.0
28/08/2019 21:15:54,1.0,1.0,1.0
```

- Logged data upload to the InfluXDB ‘historicaldb’
  - Periodically : Crontab job scheduled every Sunday(weekly) at 23:20:00
  - Retention policy of 6 months on ‘historicaldb’
  - Data format transformed over the fly into a five minute format.

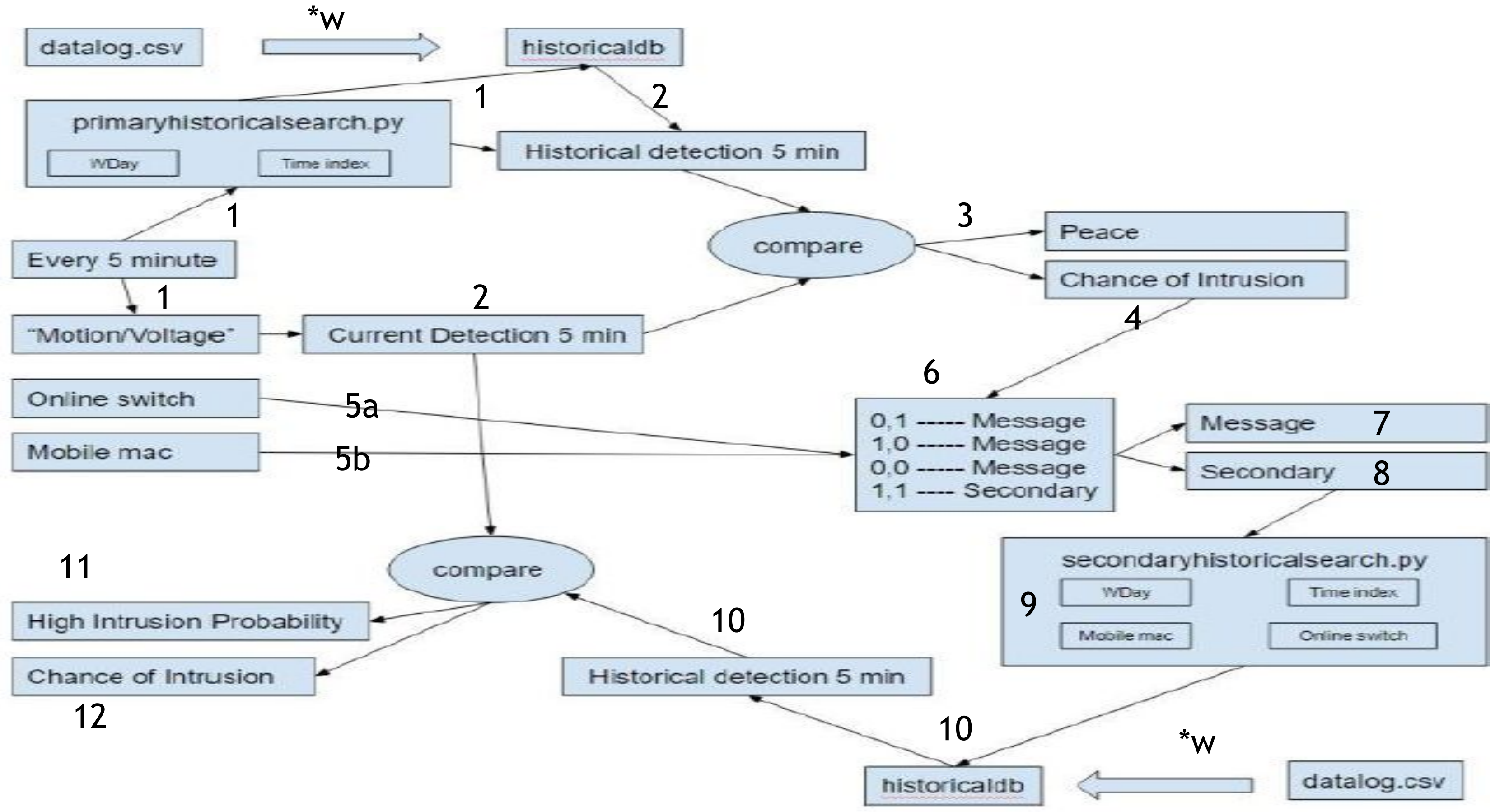


```
28/08/2019 20:17:23,1.0,1.0,1.0
28/08/2019 20:23:44,1.0,1.0,1.0
28/08/2019 20:25:31,1.0,1.0,1.0
28/08/2019 20:31:24,1.0,1.0,1.0
28/08/2019 20:42:15,1.0,1.0,1.0
28/08/2019 20:42:42,1.0,1.0,1.0
28/08/2019 20:56:21,1.0,1.0,1.0
28/08/2019 21:07:07,1.0,1.0,1.0
28/08/2019 21:11:02,1.0,1.0,1.0
28/08/2019 21:15:54,1.0,1.0,1.0
```

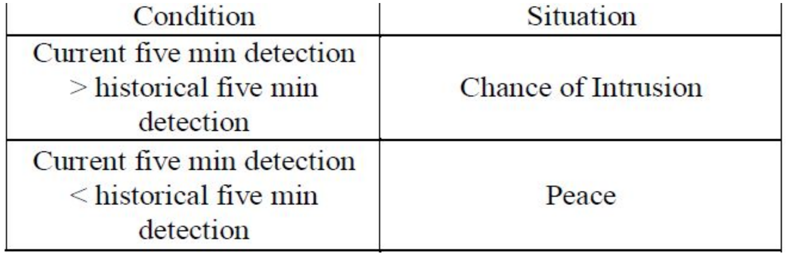


| Timestamp (UNIX) | WDay | Time index | moti -on | Online switch | Mobile Mac |
|---|---|---|---|---|---|
| dd/mm/yy yy hh:mm:ss | 1-7 | 0-287 | * | 0 or 1 | 0 or 1 |

- Applying the historical search logic
  - Done in every 5 min interval
  - SELECT Query is passed to return the number of detections for every 5 min interval on  the historical data.
  - In every 5 min, SELECT num_motion WHERE WDay = (1..7) & TimeIndex = (0….287)

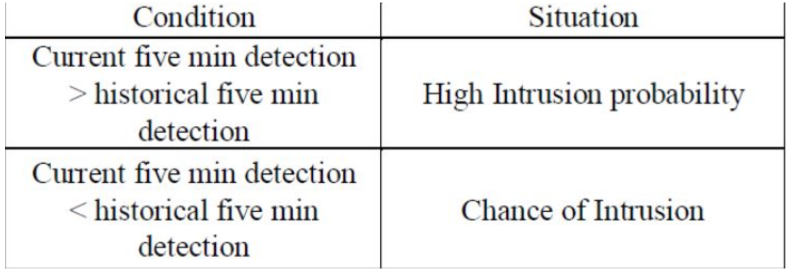

*w
datalog.csv
historicaldb
1
2
primaryhistoricalsearch.py
WDay
Time index
Historical detection 5 min
1
Every 5 minute
1
"Motion/Voltage"
2
Current Detection 5 min
compare
3
Peace
Chance of Intrusion
4
6
Online switch
5a
Mobile mac
5b
0,1 ------ Message
1,0 ------ Message
0,0 ------ Message
1,1 ---- Secondary
Message
7
Secondary
8
secondaryhistoricalsearch.py
9
WDay
Time index
Mobile mac
Online switch
11
compare
High Intrusion Probability
Chance of Intrusion
12
10
Historical detection 5 min
10
historicaldb
*w
datalog.csv

## 1st Compare:



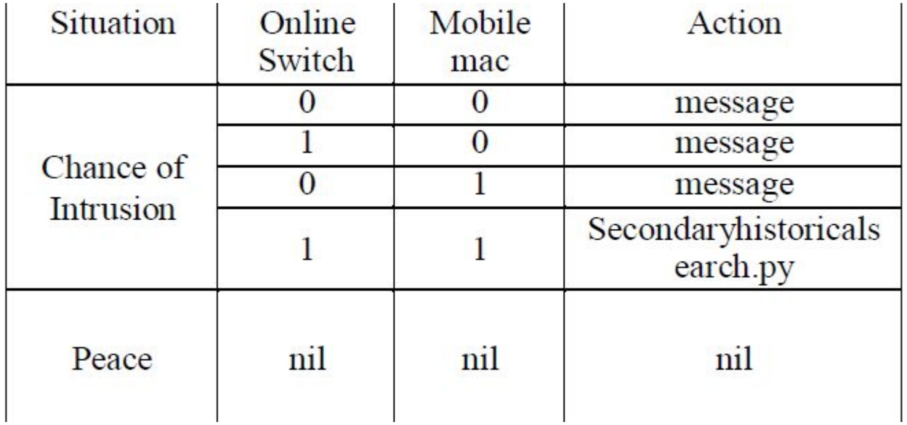
| Condition | Situation |
|---|---|
| Current five min detection > historical five min detection | Chance of Intrusion |
| Current five min detection < historical five min detection | Peace |

## 2nd Compare:



| Condition | Situation |
|---|---|
| Current five min detection > historical five min detection | High Intrusion probability |
| Current five min detection < historical five min detection | Chance of Intrusion |

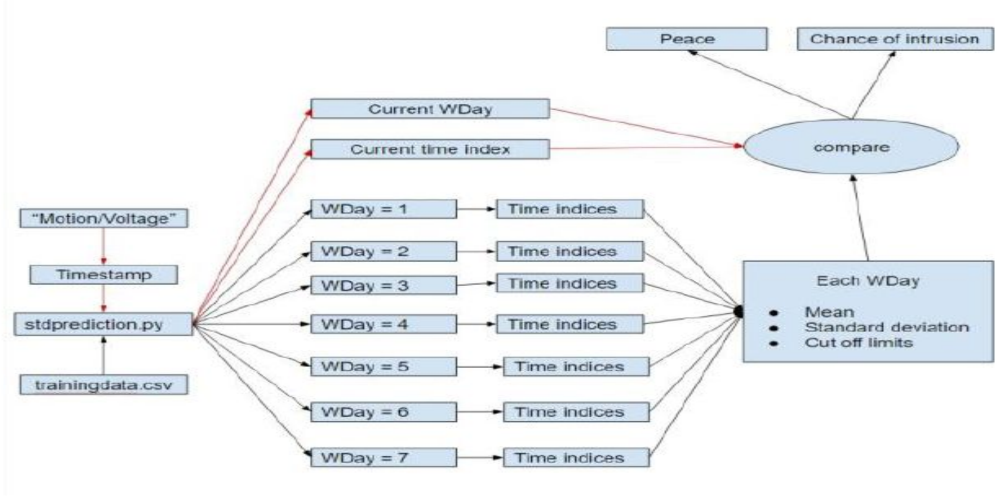
## After 1st Compare:



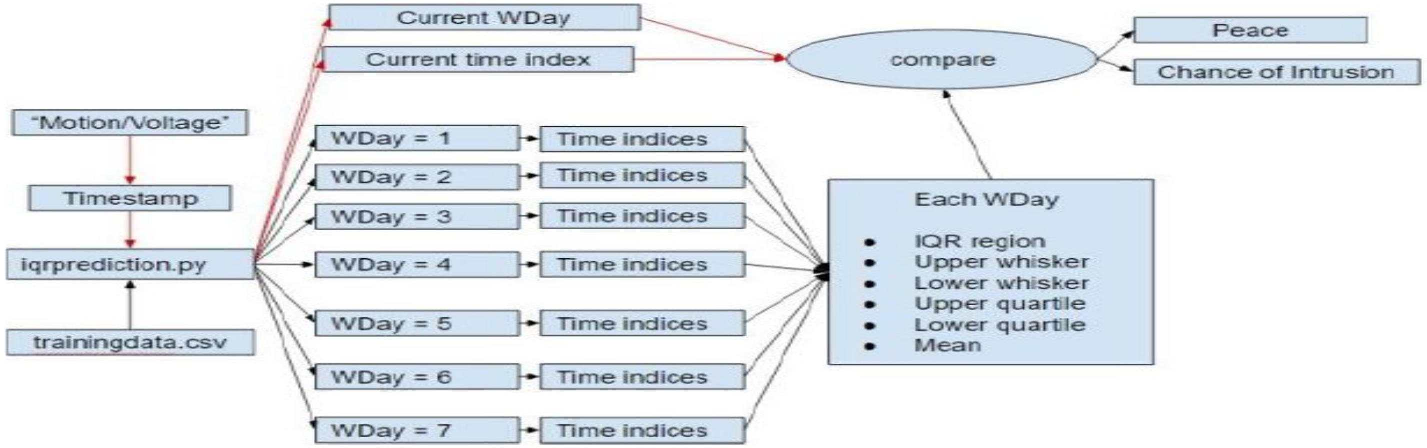
| Situation | Online Switch | Mobile mac | Action |
|---|---|---|---|
| Chance of Intrusion | 0 | 0 | message |
| | 1 | 0 | message |
| | 0 | 1 | message |
| | 1 | 1 | Secondaryhistoricalsearch.py |
| Peace | nil | nil | nil |

## Statistical methods

- Standard deviation method
  - Source file - logged csv data ( Timestamp, motion values, online switch, mobile mac)
  - Data converted in five minute format as WDay, Time Index.
  - Only considers data happened at user's presence ( online switch: 1 and mobile mac: 1)
  - Collects/Groups the motion detection caused respective time indices of each WDay
  - Calculates Mean, Standard deviation parameters for Each WDay
  - On every real time detection, the time index is checked if it is deviating more than the cut offs.



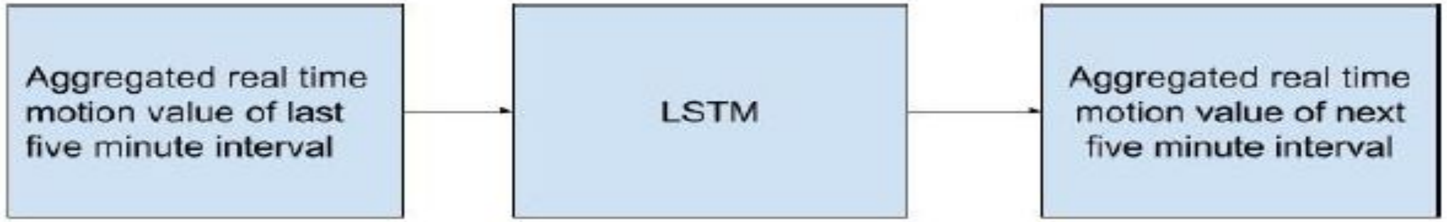
- InterQuartile region
  - Source file - logged csv data ( Timestamp, motion values, online switch, mobile mac)
  - Data converted in five minute format as WDay, Time Index.
  - Only considers data happened at user's presence ( online switch: 1 and mobile mac: 1)
  - Collects/Groups the motion detection caused respective time indices of each WDay
  - Calculates the IQR, whisker and quartile regions
  - On every real time detection, the time index is checked if it is inside/outside the regions.

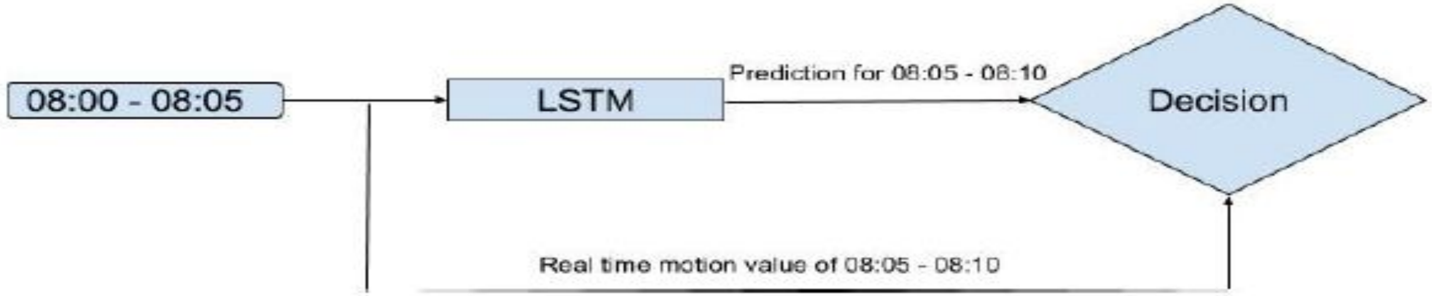


## Artificial Intelligence Approach

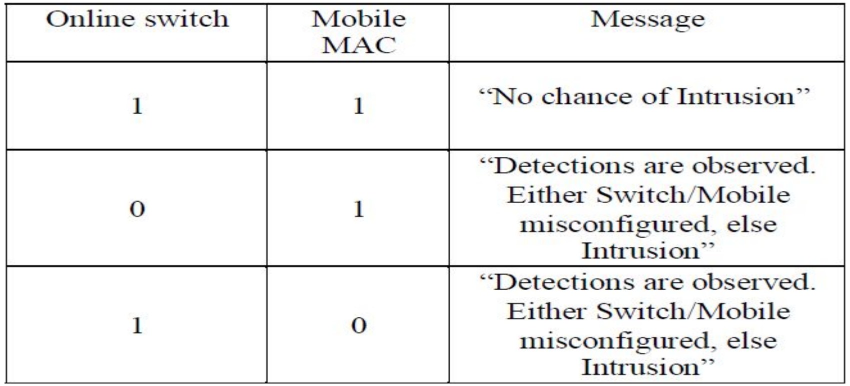
- Application deals with motion and time series, best suited is the Time series forecasting model.
- Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning.
  - LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).
- The lack of many training parameters, only time and motion values are only the parameters that are passed to the machine learning model. LSTM is preferred in such univariate time cases with plethora amount of data.
- LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.
- Whenever there is a motion detected, from the sensor, the data is recorded in the client side (Node-RED) as Timestamp : mm/dd/yy HH:mm:SS and motion value as “1”. This is the data format which is fed into our machine learning model as the input for training.

- The machine learning model was built with 67 percent of motion data (timestamp aggregated as 5min duration and the corresponding motion values for the duration is added.) as training set and remaining set of test data was used to validate the same. The other parameters used to train model was
- Batch Size : 1 Epochs : 1500 Neurons : 4 Loss : mean_squared_error Optimizer: adam
- The LSTM model obtained was quite satisfactory with the RMSE score of 1.6 as per test data validation given below where the orange plot refers to the predictions.
- The model was run on the machine with Flask application (framework)  which is based on Python.The python script in Node-Red would take the motion value as its input argument and make an API call to our Flask application server where the model is loaded and running. The predicted value is fed back as the output of the script.


Aggregated real time motion value of last five minute interval
LSTM
Aggregated real time motion value of next five minute interval


08:00 - 08:05
LSTM
Prediction for 08:05 - 08:10
Decision
Real time motion value of 08:05 - 08:10

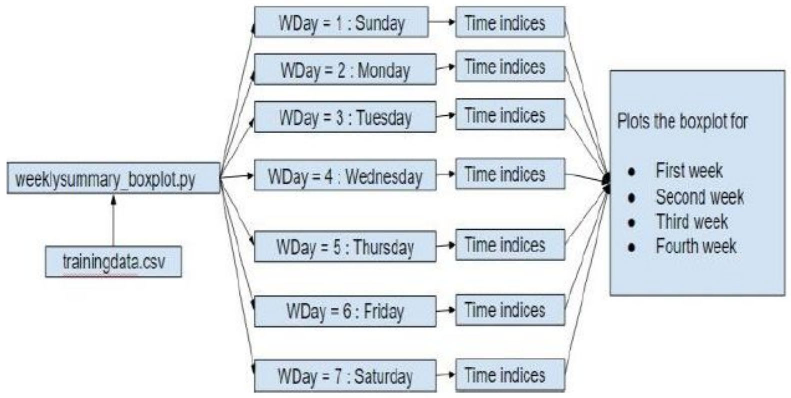
- After comparison of real time and predicted values, a difference of these values are calculated.
- If the difference is greater than 5 motions (which is a customization parameters and could vary with different applications and users), the final decision from the Machine learning side as Intrusion.
- The output decision/response again depends on the value of ARP and online switch obtained. Again this decisive part depends on person to person who can customize the algorithm as per their choice.
- If the output from LSTM is in a considerable range (threshold 5 motions) it is concluded as “Peace” ignoring switch and mac values.
- However if the output from LSTM is considerably deviated from the normal range, then the decision is based on the obtained switch and mac values. The user is notified by an alert mail sent to their respective mail id. This is done using the mail node in Node-RED.

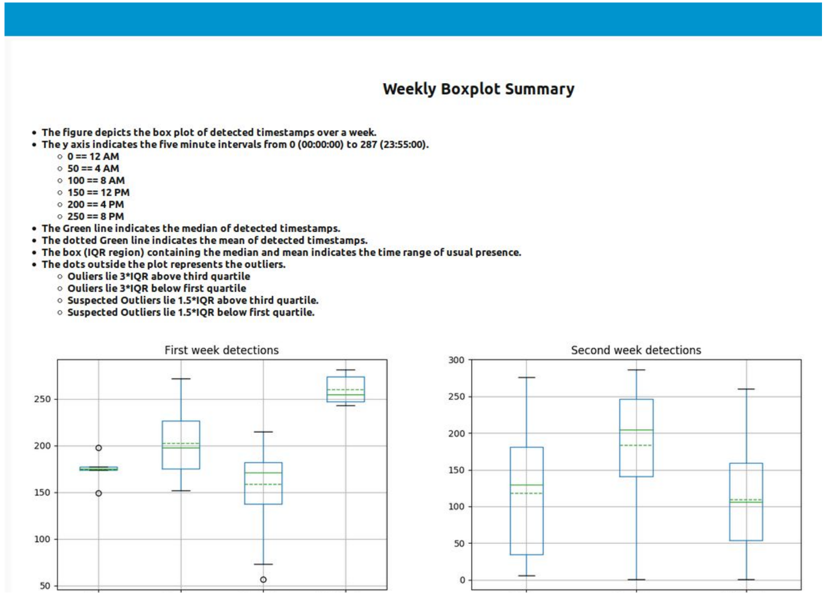


| Online switch | Mobile MAC | Message |
|---|---|---|
| 1 | 1 | “No chance of Intrusion” |
| 0 | 1 | “Detections are observed. Either Switch/Mobile misconfigured, else Intrusion” |
| 1 | 0 | “Detections are observed. Either Switch/Mobile misconfigured, else Intrusion” |

## Reports

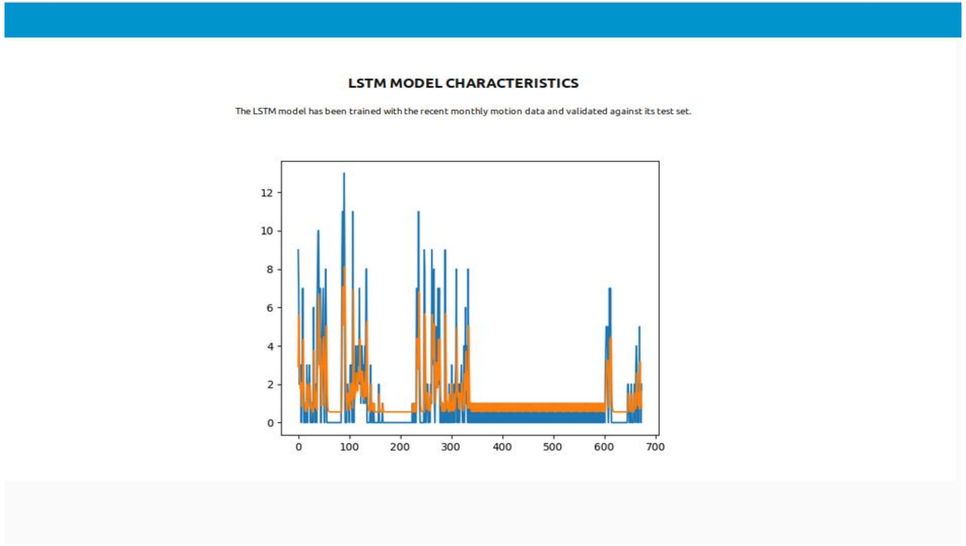
- Weekly detection summary report
  - Source file : trainingdata.csv ( entire data logged)




**Weekly Boxplot Summary**

- The figure depicts the box plot of detected timestamps over a week.
- The y axis indicates the five minute intervals from 0 (00:00:00) to 287 (23:55:00).
  - 0 == 12 AM
  - 50 == 4 AM
  - 100 == 8 AM
  - 150 == 12 PM
  - 200 == 4 PM
  - 250 == 8 PM
- The Green line indicates the median of detected timestamps.
- The dotted Green line indicates the mean of detected timestamps.
- The box (IQR region) containing the median and mean indicates the time range of usual presence.
- The dots outside the plot represents the outliers.
  - Ouliers lie 3*IQR above third quartile
  - Ouliers lie 3*IQR below first quartile
  - Suspected Outliers lie 1.5*IQR above third quartile.
  - Suspected Outliers lie 1.5*IQR below first quartile.

- Model training report
    - The model training report which tells the accuracy of the model trained to the users.



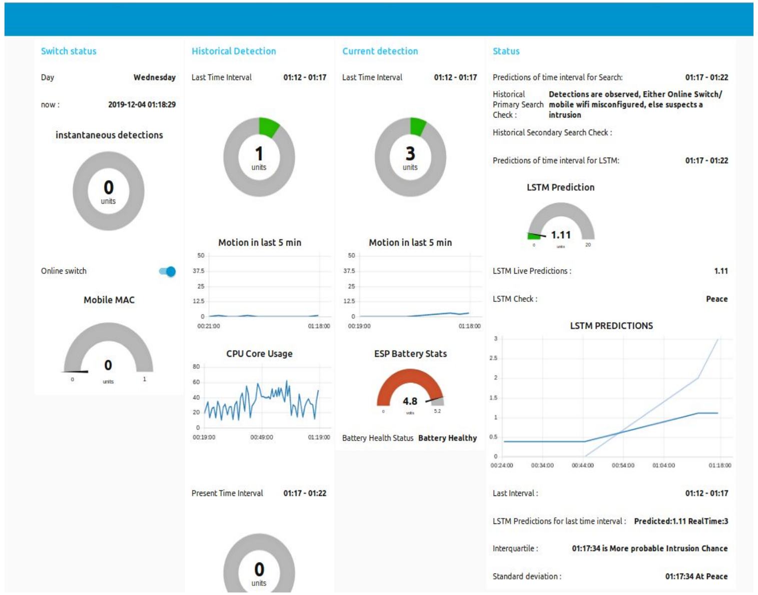
LSTM MODEL CHARACTERISTICS

The LSTM model has been trained with the recent monthly motion data and validated against its test set.

## Automation - Services and Cron Jobs

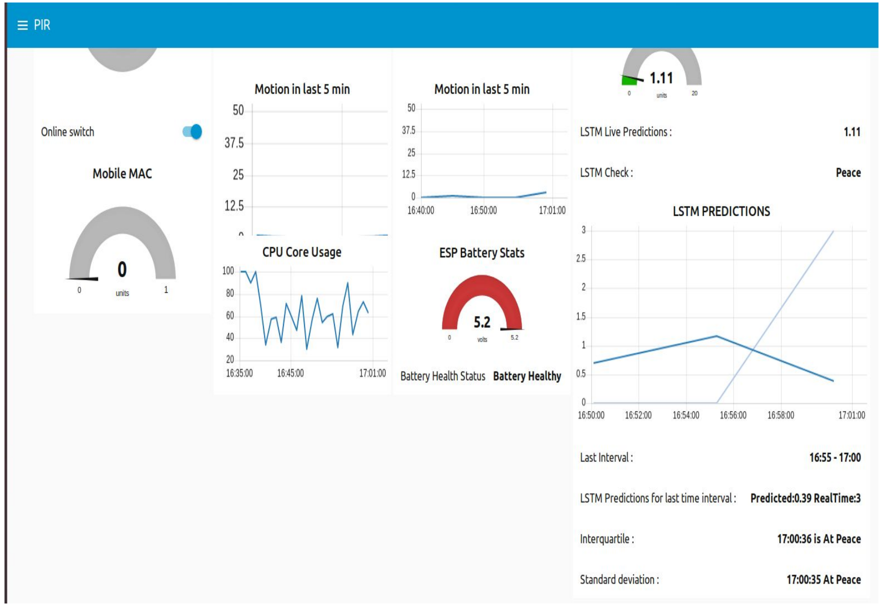
- Service - modelstart.service auto starts the Flask ML Server
- Service - NodeRed Service which will be running in the device.
- CronJob - The weekly summary boxplot is scheduled to run every day at 9pm.
- CronJob - The weekly influx load is scheduled to run every weekend to load the fresh motion data to influx database.
- CronJob - The training lstm is scheduled to run every month to recreate a new efficient ml model based on the latest motion data.
- CronJob - The refresh_training script is run after the above step to reset the monthly motion data.

## NODE-RED GUI dashboards



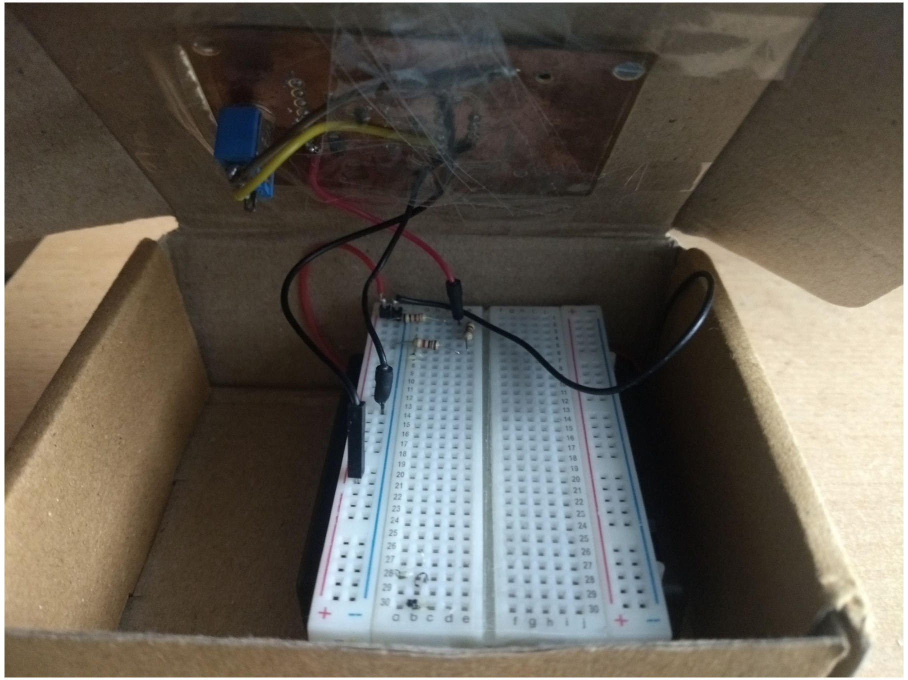
| | |
|---|---|
| Day | Wednesday |
| now : | 2019-12-04 01:18:29 |
| Last Time Interval | 01:12 - 01:17 |
| Present Time Interval | 01:17 - 01:22 |
| Battery Health Status | Battery Healthy |

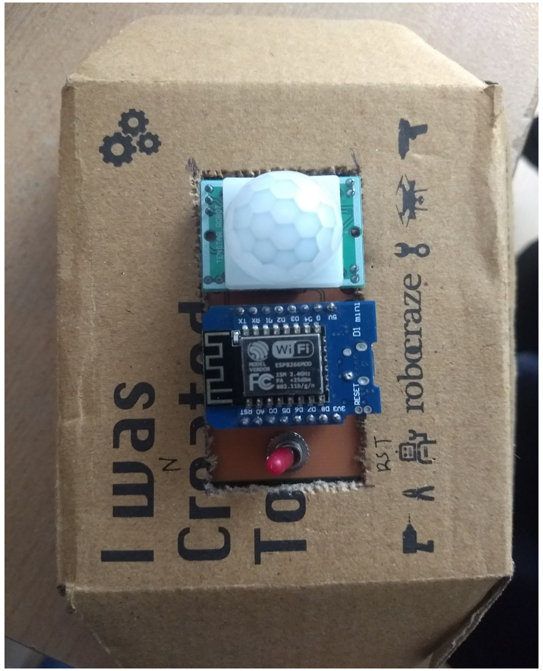
| | |
|---|---|
| Predictions of time interval for Search: | 01:17 - 01:22 |
| Historical Primary Search Check : | Detections are observed, Either Online Switch/ mobile wifi misconfigured, else suspects a intrusion |
| Historical Secondary Search Check : | |
| Predictions of time interval for LSTM: | 01:17 - 01:22 |
| LSTM Live Predictions : | 1.11 |
| LSTM Check : | Peace |
| Last Interval : | 01:12 - 01:17 |
| LSTM Predictions for last time interval : | Predicted:1.11 RealTime:3 |
| Interquartile : | 01:17:34 is More probable Intrusion Chance |
| Standard deviation : | 01:17:34 At Peace |



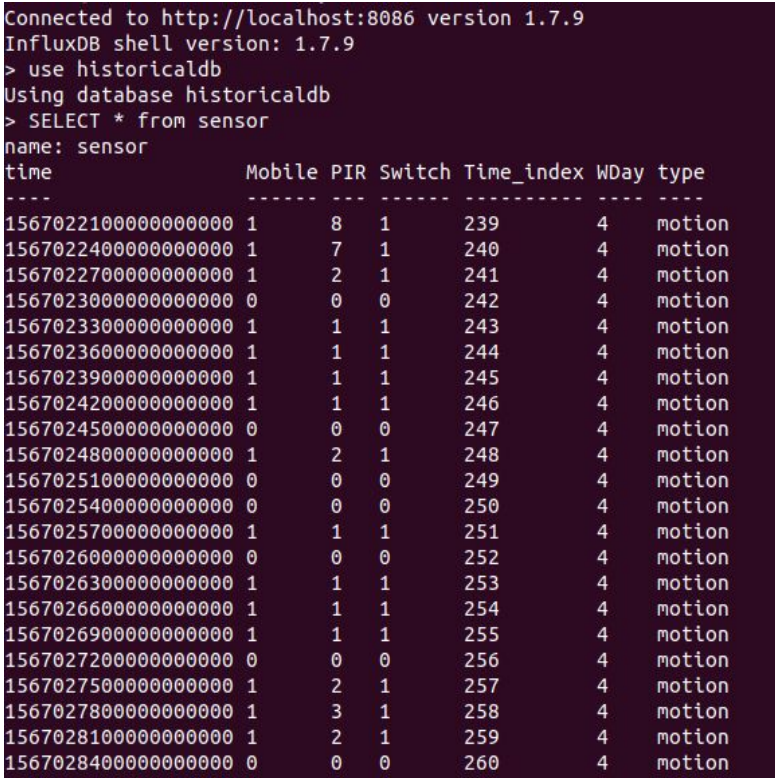
| | |
|---|---|
| Battery Health Status | Battery Healthy |
| LSTM Live Predictions : | 1.11 |
| LSTM Check : | Peace |
| Last Interval : | 16:55 - 17:00 |
| LSTM Predictions for last time interval : | Predicted:0.39 RealTime:3 |
| Interquartile : | 17:00:36 is At Peace |
| Standard deviation : | 17:00:35 At Peace |

## PIR+ESP Assembly

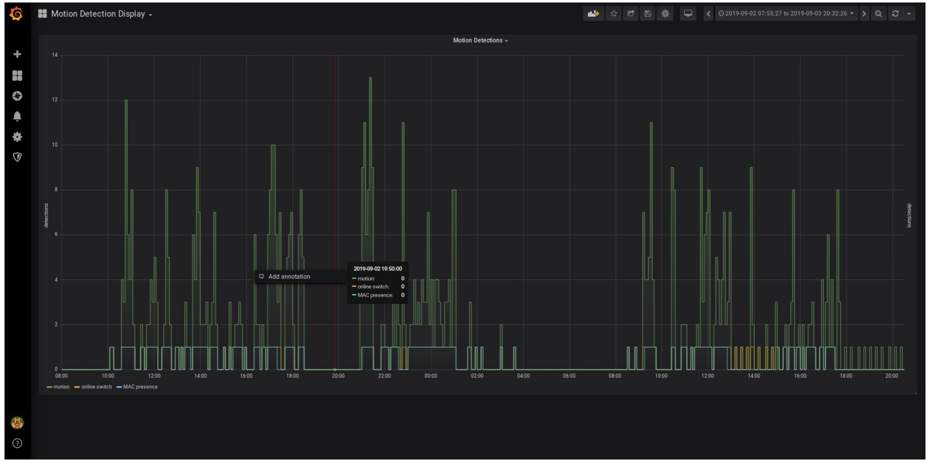


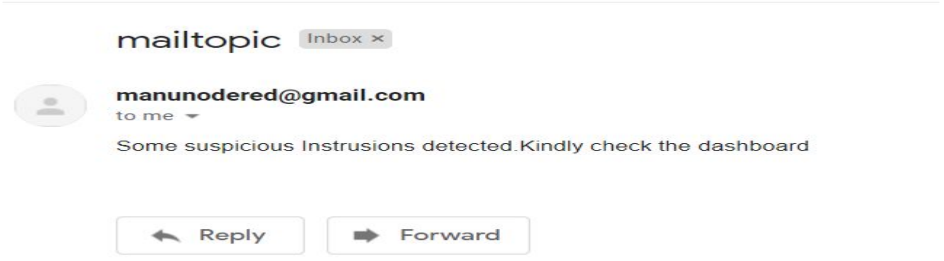


## User notification, InfluxDB Uploads & Grafana Visualisation



```
Connected to http://localhost:8086 version 1.7.9
InfluxDB shell version: 1.7.9
> use historicaldb
Using database historicaldb
> SELECT * from sensor
name: sensor
time                Mobile PIR Switch Time_index WDay type
----                ------ --- ------ ---------- ---- ----
1567022100000000000 1      8   1      239        4    motion
1567022400000000000 1      7   1      240        4    motion
1567022700000000000 1      2   1      241        4    motion
1567023000000000000 0      0   0      242        4    motion
1567023300000000000 1      1   1      243        4    motion
1567023600000000000 1      1   1      244        4    motion
1567023900000000000 1      1   1      245        4    motion
1567024200000000000 1      1   1      246        4    motion
1567024500000000000 0      0   0      247        4    motion
1567024800000000000 1      2   1      248        4    motion
1567025100000000000 0      0   0      249        4    motion
1567025400000000000 0      0   0      250        4    motion
1567025700000000000 1      1   1      251        4    motion
1567026000000000000 0      0   0      252        4    motion
1567026300000000000 1      1   1      253        4    motion
1567026600000000000 1      1   1      254        4    motion
1567026900000000000 1      1   1      255        4    motion
1567027200000000000 0      0   0      256        4    motion
1567027500000000000 1      2   1      257        4    motion
1567027800000000000 1      3   1      258        4    motion
1567028100000000000 1      2   1      259        4    motion
1567028400000000000 0      0   0      260        4    motion
```





mailtopic Inbox ×

**manunodered@gmail.com**
to me

Some suspicious Instrusions detected.Kindly check the dashboard

Reply Forward

## Conclusion & Further works

- Increase the number of motion sensors used.
- Optimize the placement of the motion sensors installed.
- Can infer the direction of motion and the speed of the motion happening.
- Can infer the type of object - small, big, medium
- Analysing the direction of motion - desirable direction or not.
- Sequence of activation of the different motion sensors placed at different locations of the home.
- If privacy is not a concern, employ cameras to confirm by taking a snapshot in case of abnormal situations which can be further used in image processing.

# Thanks for your attention