Anomaly detection in IoT devices and visualization of results

K V Bhavana, 1PE15CS063
Srinivas Vamshi, 1PE15CS077
Prakruth Nagraj, 1PE15CS104
Vikil S R, 1PE16CS433

Ms Shanthala
Batch No. - 16

May 6, 2019

Problem Statement / Definition

- **Domain:** Machine Learning
- **What:** Anomaly detection in large datasets
- **How:** One-class SVM for anomaly detection
- **Data:** Operational sensor dataset

Overview

Motivation

- IoT devices may not be in working condition
- Final result may vary if an IoT device fails
- Loss of money and time if project fails only because of failure of an IoT device

We want to build a project that alleviates these issues by better detecting and analyzing the anomalies

What are we doing?

- Implement an ML-based solution for detection of anomalies and also show the anomalies detected.
- **Idea:** Our initial focus is to detect anomalies in IoT data. We then want output the health of the IoT device.

Overview

Literature Survey

- Anomaly detection with event data in the Internet of Things :- Multidimensional scaling algorithm used to detect anomalies
- Anomaly detection and monitoring in Internet of Things communication :- A multi-platform monitoring and anomaly detection system that supports heterogeneous devices.
- Anomaly detection using machine learning using a case study :- Performance criteria used in anomaly detection based on mathematical statistics to specify boundaries in emerging applications used in the world.
- Fog-Empowered Anomaly detection in IoT using Hyperellipsoidal clustering :- Hyperellipsoidal clustering to detect anomalies
- Detecting malicious anomalies in IoT :- Performance of ensemble learners on incomplete IoT intrusion datasets, represented by point anomalies
- Information Visualization and Visual Data Mining :- Several algorithms to visualize data

Advantages and Disadvantages

Advantages :

- A non-technical person can still identify the health of an IoT device irrespective of his knowledge in IoT devices
- While ML and AI can help to make sense of data, it still requires an analyst

Disadvantages :

- Lot of historic operational data is required

Overview

Methodology

2 phases of our project:

- Detection of anomalies : Anomaly detection done using historic operational data using One-class SVM
- Visualization of the results : The anomalies are displayed in the form of a scatter plot

Overview

High Level design

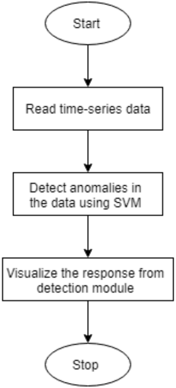


Figure: Data Flow Diagram

High Level design

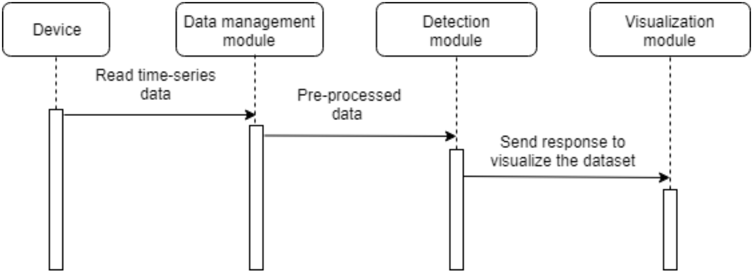


Figure: Sequence Diagram

Overview

Dataset

Beach Name	Measurement Timestamp	Water Temperature	Turbidity	Transducer Depth	Wave Height	Wave Period	Battery Life	Measurement Timestamp Label	Measurement ID
Montrose Beach	08/30/2013 08:00:00 AM	20.3	1.18	0.891	0.08	3	9.4	8/30/2013 8:00 AM	MontroseBeach201308300800
Ohio Street Beach	05/26/2016 01:00:00 PM	14.4	1.23		0.111	4	12.4	05/26/2016 1:00 PM	OhioStreetBeach201605261300
Calumet Beach	09-03-2013 16:00	23.2	3.63	1.201	0.174	6	9.4	09-03-2013 16:00	CalumetBeach201309031600
Calumet Beach	05/28/2014 12:00:00 PM	16.2	1.26	1.514	0.147	4	11.7	5/28/2014 12:00 PM	CalumetBeach201405281200
Montrose Beach	05/28/2014 12:00:00 PM	14.4	3.36	1.388	0.298	4	11.9	5/28/2014 12:00 PM	MontroseBeach201405281200

Figure: Dataset

Overview

Implementation

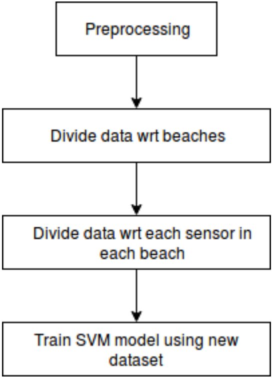


Figure: Flowchart

One-class SVM

- Unsupervised algorithm, that learns the decision function to find anomalies
- This function is used to classify new data, whether it belongs to or is different from the dataset
- Tries to fit a hyper-sphere, that includes most of the training samples
- Main issue: The dataset has to be free from outliers

Implementation - Code snippets

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import OneClassSVM
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.covariance import EllipticEnvelope
from IPython.display import display

#method 1 to detect outliers
def ellipticCurve(dataset):
    classifier = EllipticEnvelope(contamination = outlierFraction)
    classifier.fit(dataset)
    predScore = classifier.decision_function(dataset)
    pred = classifier.predict(dataset)
    outlierRows = [i for i in range(len(pred)) if pred[i]==-1]
    return predScore, outlierRows

#method 2 to detect outliers
def oneClassSVM(dataset):
    classifier = OneClassSVM(nu = outlierFraction, gamma = 0.03)
    classifier.fit(dataset)
    predScore = classifier.decision_function(dataset).T[0]
    pred = classifier.predict(dataset)
    outlierRows = [i for i in range(len(pred)) if pred[i]==-1]
    return predScore, outlierRows

df = pd.read_csv("./preprocessed.csv")
beaches = list(df["BeachName"].unique())
numBeaches = len(beaches)

cols = list(df.columns)
colSize = len(cols)
noStrCols = cols
del(noStrCols[1])
del(noStrCols[6])
```

Implementation - Code snippets

```python
for beach in beaches:
    csvName = "Beach"+str(beach)+".csv"
    dfDic[beach] = pd.read_csv(csvName)

colsToAnalyze = noStrCols
numRows = {}
for i in range(0, numBeaches, 1):
    numRows[i] = dfDic[i].shape[0]

outlierFraction = 0.01
ran = np.random.RandomState(123)
#anomalyList = ["ellCurve", "svm"]
anomalyList = ["svm"]

#dfDic[beach]['Turbidity']

predictions = {}
for beach, data in divideByBeachName.items():
    predictions[beach] = {}


    for x in noStrCols:
        s, o = oneClassSVM(np.reshape(dfDic[beach][x], (-1, 1)))
        predictions[beach][x] = {"ScorePred":s, "outliers":o}

statsCols = ["BeachName", "dataSize", "normals", "anomalies", "anomaliesRate"]
outliers = {}
for index in beaches:
    outliers[index] = {}

for x in noStrCols:
    dataSize = []
    oks = []
    ngs = []
    ngRate = []
    for i in range(0, len(beaches)):
```

Implementation - Dataset anomalies

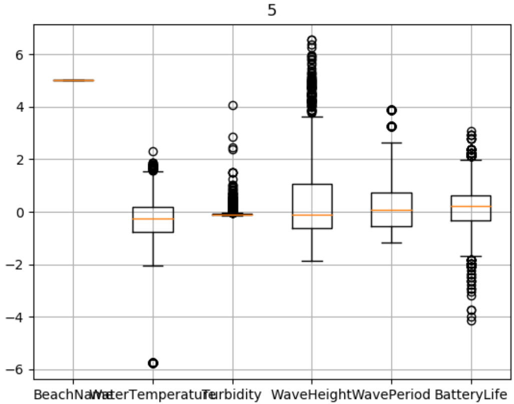
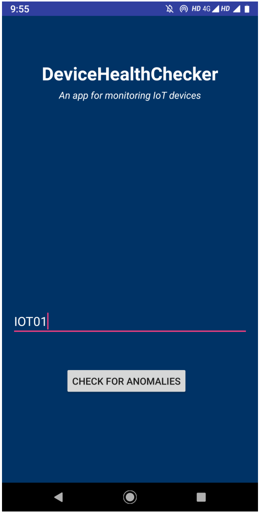


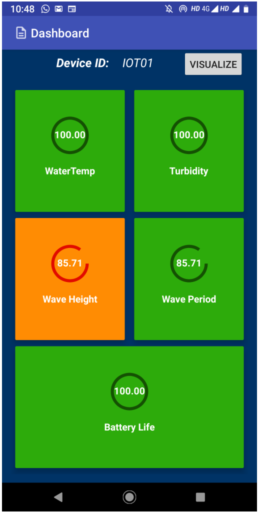
Figure: Anomalies detected

Overview

Results

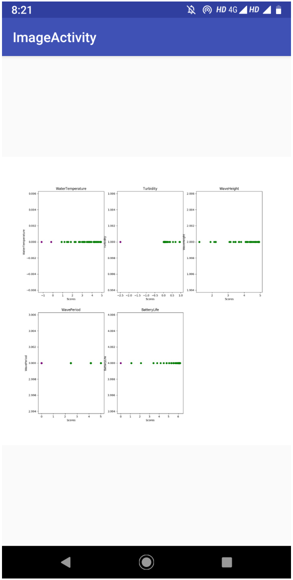


(a) Input



(b) Output

Results


8:21
ImageActivity
WaterTemperature
Turbidity
WaveHeight
WavePeriod
BatteryLife
Scores

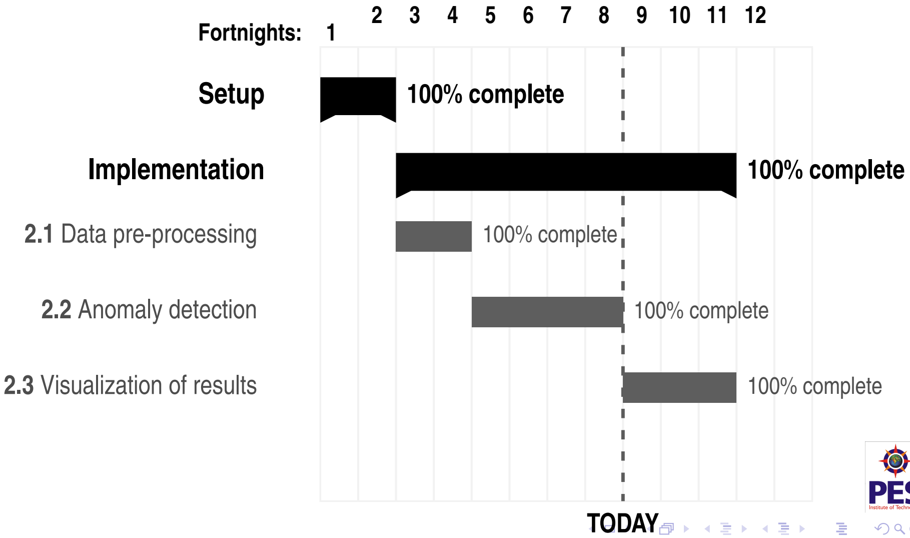
Overview

Hardware and Software Requirements

- Mobile phone for android app
- Flask
- Android Studio 3.4

Overview

Time line of completion of project from Sept 2018-April 2019(Gantt Charts).



Overview

References

Lingjuan Lyu, et al., (2017)
Fog-Empowered Anomaly Detection in IoT Using Hyperellipsoidal Clustering
IEEE Internet of Things Journal (Volume: 4, Issue: 5, Oct. 2017)

Software Innovations - Bosch IoT suite (2016)
Anomaly detection with event data in the Internet of Things
Internet of Things white paper series by Bosch, 1-19.

Deris Stiawan, et al., (2016)
Anomaly detection and monitoring in Internet of Things communication
2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)

Igor Franc, et al., (2016)
Detecting Malicious Anomalies in IoT: Ensemble Learners and Incomplete Datasets
The Eight International Conference on Business Information Security (BISEC), At Metropolitan University, Belgrade, 44-49.

References

Goverdhan Reddy Jidiga et al., (2014)
Anomaly detection using machine learning using a case study
IEEE International Conference on Advanced Communications, Control and Computing Technologies, 2014).

Daniel A. Keim (2002)
Information Visualization and Visual Data Mining
IEEE transactions on visualization and computer graphics, vol. 7, no. 1, January-March 2002, 100 – 107.

The End