Summer Project Report

B.tech - Part-III

2017-18

Probabilistic Approach Based Anomaly Detection Techniques for Real Time Systems

Submitted By :-

Shikhar Gupta

B.Tech Part III

15035040

Ceramic Eng.

Supervised By :-

Dr. Hari Prabhat Gupta

Assistant Professor

Dept. Of Computer Science

<u>Probabilistic Approach Based Anomaly Detection Techniques for Real Time Systems</u>

I acknowledge that Mr. Shikhar Gupta (15035040, B.tech-III, Ceramic Eng.) has completed a project on above topic during Summers 2017-18 under my supervision.

(Signature of the Supervisor)

Dr. Hari Prabhat Gupta
Assistant Professor
Dept. of Computer Sciences

OVERVIEW :-

Anomalies are strange data points; they usually represent an unusual occurrence. Anomaly detection is presented from the perspective of Wireless sensor networks. Different approaches have been taken in the past, as we will see, not only to identify outliers, but also to establish the statistical properties of the different methods. In a data-driven culture, companies approach decision-making from a quantitative point of view and rely less on gut feeling when making major decisions. Lack of accurate, timely or relevant data from across the business is also a major concern among companies with primitive or basic capabilities.

In statistics, an outlier is an observation point that is distant from other observations. An outlier can cause serious problems in statistical analyses. Outliers can occur by chance in any distribution, but they often indicate either measurement error or that the population has a heavy-tailed distribution.

ANOMALY DETECTION PROBLEMS :-

In data mining, anomaly detection is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.Typically the anomalous items will translate to some kind of problem such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions.

In particular, in the context of abuse and network intrusion detection, the interesting objects are often not *rare* objects, but unexpected bursts in activity. This pattern does not adhere to the common statistical definition of an outlier as a rare object, and many outlier detection methods (in particular unsupervised methods) will fail on such data, unless it has been aggregated appropriately. Instead, a cluster analysis algorithm may be able to detect the micro clusters formed by these patterns.

PURPOSE :-

The overall purpose of this study is to find a way for identifying incorrect data in statistics about contributions. The goal is to build an algorithm that determines if a contribution has a risk of being inaccurately coded, based on anomaly detection methods. It is desirable to investigate if descriptions about the contributions, in the form of plain text, can be useful in this problem. In order to obtain the goal, the following questions will be answered :-

- ➢What kind of informative statistical data is important to include in the study in order to build an anomaly detector?
- ➢Which processes and algorithms are preferable to use for this anomaly detection problem?

OUR APPROACH :-

We are using linear regression technique for anomaly detection in real time system data. linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression.

Simple linear regression is useful for finding relationship between two continuous variables. One is predictor or independent variable and other is response or dependent variable. It looks for statistical relationship but not deterministic relationship. Relationship between two variables is said to be deterministic if one variable can be accurately expressed by the other. For example, using temperature in degree Celsius it is possible to accurately predict Fahrenheit. Statistical relationship is not accurate in determining relationship between two variables.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications.This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

DATASET INFORMATION :-

The data (test signal) used in this report is **Single Chest-Mounted Accelerometer Dataset.** Dataset involves two csv files, one with **training data** and other with **testing data** namely anomaly_detection_train.csv & anomaly_detection_test.csv.

<u>Data Set Attribute Information:</u>

- ➢ The dataset collects data from a wearable accelerometer mounted on the chest
- ➢ Sampling frequency of the accelerometer: 52 Hz
- ➢ Accelerometer Data are Uncalibrated
- ➢ User Activity: Walking
- ➢ Data Format: CSV
- ➢ Training Instance = 25000
- ➢ Testing Instance = 100
- ➢ **Attribute Information:**-sequential number, x acceleration, y acceleration, z acceleration

DATASET PREVIEW :-

anomaly_detection_train.csv		anomaly_detection_test.csv	
1.	**1895,2390,2024**	**1.**	**1928,2634,2028**
2.	**1889,2389,2022**	**2.**	**1928,2298,2026**
3.	**1886,2383,2027**	**3.**	**1928,2290,2031**
-----------------------------		------------------------------	
-----------------------------		------------------------------	
-----------------------------		------------------------------	
24998.	**1928,2363,2070**	**98.**	**1907,2496,2070**
24999.	**1928,2317,2055**	**99.**	**1923,2399,2099**
25000.	**1928,2302,2035**	**100.**	**1941,2373,3406**

IMPLEMENTATION :-

Implementation is done in Python with linear Regression technique used from Linear model module imported from sklearn Package. For better visualisation of anomalous data points pyplot from matplotlib is used. For better 3D visualisation of output result of the Implementation ggplot style of matplotlib is also used.

The Python code implementation is as :-

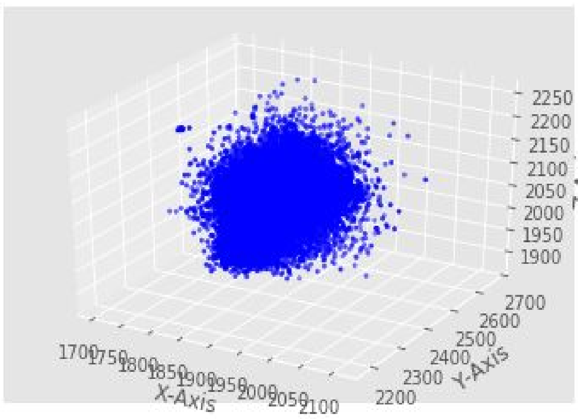
```
from sklearn import linear_model
import matplotlib
import matplotlib.pyplot as plt
from pandas import read_csv
from mpl_toolkits.mplot3d import Axes3D


filename = "anomaly_detection_train.csv"
names = ["x-axix","y-axis","z-axis"]
dataset = read_csv(filename,names = names)
array = dataset.values


X = []
Y = []
Z = []
for i in array:
   X.append([i[0]])
   Y.append([i[1]])
   Z.append([i[2]])


model_X = linear_model.LinearRegression()
model_X.fit(X,Y)
```

```
linear_model.LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1,
normalize=False)
model_Y = linear_model.LinearRegression()
model_Y.fit(Y,Z)
linear_model.LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1,
normalize=False)
model_Z = linear_model.LinearRegression()
model_Z.fit(Z,X)
linear_model.LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1,
normalize=False)


test_filename = "anomaly_detection_test.csv"
test_dataset = read_csv(test_filename,names = names)
test_array = test_dataset.values

outlier_list = []
n=1
count = 0
for instance in test_array:
#    print(instance)
    result_Y = model_X.predict(instance[0])
    result_Z = model_Y.predict(instance[1])
    result_X = model_Z.predict(instance[2])
    if result_X > 2110*(.92) or result_X < 1704*(1.05):
        print("Outlier Detected at instance ",n," in value",instance[2]," of Z-axis
")
        #print("Expected X at",result_X)
        count+=1
        outlier_list.append(n)
    if result_Y > 2678*(.92) or result_Y < 2179*(1.05):
        print("Outlier Detected at instance ",n," in value",instance[0]," of X-axis
")
        #print("Expexted Y at",result_Y)
        count+=1
        outlier_list.append(n)
    if result_Z > 2248*(.92) or result_Z < 1878*(1.05):
        print("Outlier Detected at instance ",n," in value",instance[1]," of Y-axis
")
        #print("Expexted Z at",result_Z)
        count+=1
        outlier_list.append(n)
    n += 1

if count>0:
    print(count," anomalies detected")
if n>25000:
    print("No anomaly detected")

matplotlib.style.use('ggplot')
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.set_xlabel('X-Axis')
ax.set_ylabel('Y-Axis')
ax.set_zlabel('Z-Axis')

ax.scatter(X,Y,Z, c='b', marker='.')

X_test = []
Y_test = []
Z_test = []
for i in test_array:
    X_test.append([i[0]])
    Y_test.append([i[1]])
    Z_test.append([i[2]])

matplotlib.style.use('ggplot')
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.set_xlabel('X_test-Axis')
ax.set_ylabel('Y_test-Axis')
ax.set_zlabel('Z_test-Axis')
ax.scatter(X_test,Y_test,Z_test, c='r', marker='.')
plt.show()
```

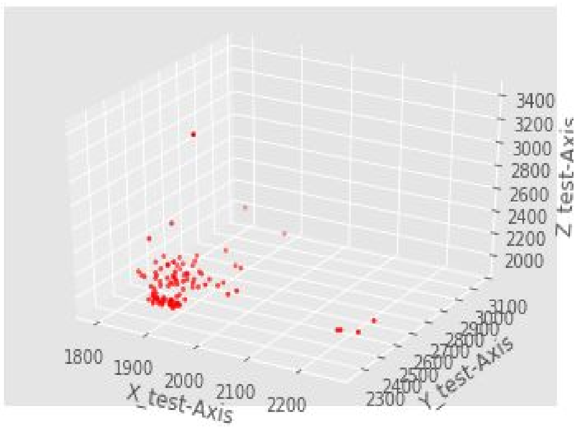
CODE OUTPUT :-

```
Outlier Detected at instance  1  in value 2634  of Y-axis
Outlier Detected at instance  5  in value 2403  of Z-axis
Outlier Detected at instance  10  in value 2255  of X-axis
Outlier Detected at instance  14  in value 3020  of Y-axis
Outlier Detected at instance  17  in value 2465  of Z-axis
Outlier Detected at instance  25  in value 2227  of X-axis
Outlier Detected at instance  30  in value 2619  of Y-axis
Outlier Detected at instance  37  in value 2409  of Z-axis
Outlier Detected at instance  59  in value 2743  of Y-axis
Outlier Detected at instance  66  in value 2253  of X-axis
Outlier Detected at instance  80  in value 2220  of X-axis
Outlier Detected at instance  94  in value 3070  of Y-axis
Outlier Detected at instance  100  in value 3406  of Z-axis
13  anomalies detected
```

Visualisation Result :-



Training Data



Testing Data

RESULT :-

To test the working capacity of Implementation Algorithm, Test data was modified such that, out of 100 data points in test data 15 data points were deliberately edited with 10-15% of random increase or decrease in their respective values.

These unedited-edited values duo and their locations in dataset were recorded for later comparisons with the functioning of algorithm.

Of the 15 edited values 13 values were correctively caught by algorithm as outliers.