

**School of Computer Science and Engineering**

**INTRUSION DETECTION IN RENEWABLE ENERGY CYBER PHYSICAL SYSTEM USING MACHINE LEARNING**

*A project submitted*

*in partial fulfillment of the requirements for the degree of*

*Bachelor of Technology (CSE)*

By

**Sanjay M S (15BCE0517)**

**Course:**

**CSE4020-Machine learning**

**E2 SLOT**

**Course Instructor**

Dr. Balakrishnan S

**April 2018**

**INDEX:**

**TOPIC PAGE**

Abstract 1

Scope 1

Introduction 1

Literature review 5

Dataset 7

Procedure 15

Code 17

Documentation 27

Screenshot 29

Result 31

References 39

**ABSTRACT:**

Intrusion detection is an important problem that has been researched within diverse research areas and application domains. Many detection techniques have been specifically developed for certain application domains, while others are more generic.In this project anomaly detection and time series analysis are considered.

**SCOPE:**

The scope of the project is to detect intrusion in cyber physical system(solar energy) using machine learning techniques namely anomaly detection and time series analysis.

**INTRODUCTION**:

RENEWABLE ENERGY:

Renewable energy is energy that is collected from renewable resources, which are naturally replenished on a human timescale, such as sunlight, wind,rain,tides,waves and geothermal heat  Renewable energy often provides energy in four important areas: electricity generation, air  and water heating/cooling, transportation, and rural (off-grid) energy services.

SOLAR ENERGY:

Solar energy is radiant light and heat from the Sun that is harnessed using a range of ever-evolving technologies such as solar heating, [photovoltaics](https://en.wikipedia.org/wiki/Photovoltaics" \o "Photovoltaics), solar thermal energy, solar architecture, molten salt power plants and artificial photosynthesis

It is an important source of renewable energy and its technologies are broadly characterized as either passive solar or active solar depending on how they capture and distribute solar energy or convert it into solar power. Active solar techniques include the use of [photovoltaic systems](https://en.wikipedia.org/wiki/Photovoltaic_system), concentrated solar power and solar water heating to harness the energy. Passive solar techniques include orienting a building to the Sun, selecting materials with favorable thermal mass or light-dispersing properties, and designing spaces that naturally circulate air.

ANOMALY DETECTION:

In machine learning , 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.

Three broad categories of anomaly detection techniques exist.

Unsupervised anomaly detection techniques detect anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal by looking for instances that seem to fit least to the remainder of the data set.

Supervised anomaly detection techniques require a data set that has been labeled as "normal" and "abnormal" and involves training a classifier (the key difference to many other statistical classification problems is the inherent unbalanced nature of outlier detection).

 Semi-supervised anomaly detection techniques construct a model representing normal behavior from a given normal training data set, and then testing the likelihood of a test instance to be generated by the learnt model.

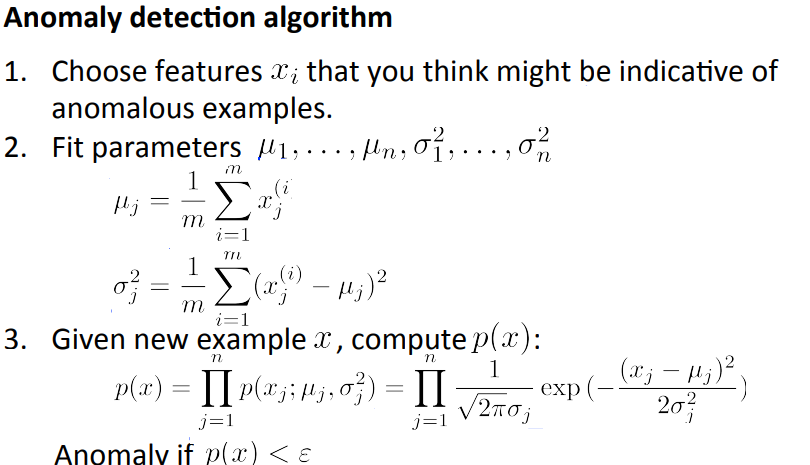
This project used unsupervised anomaly detection technique

**ANAMOLY DETECTION USING GAUSSIAN DISTRIBUTION**

A normal distribution is a very common probability distribution. In real life the normal distribution approximates many natural phenomena.

A data set is known as “normally distributed” when most of the data aggregate around it mean, on a symmetric fashion. Also, it extreme values get less and less likely to append.

When a system has been proven normally distributed, it follows a set of rules. Those rules become the model representing the normal behaviour of the metric. Under normal conditions, upcoming values will match the normal distribution and the model will be followed. Anomaly occurs when these rules get broken



**ANOMALY DETECTION USIGN TIME SERIES ANALYSIS:**

Anomalies in time series cannot be always detected by outlier analysis .So we need a separate technique to handle time series data. Generally, time series algorithms consider the adjacent values i.e. some values before and after the specified time.This project uses one such time series algorithm calles seasonal ESD algorithm.

ESD algorithm:

The generalized (extreme Studentized deviate) ESD test is used to detect one or more [outliers](http://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm) in a univariate data set that follows an [approximately normal distribution](http://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm#Normality).

The generalized ESD test requires an upper bound for the suspected number of outliers to be specified. The original data may contain lesser number of anomalies too.

Given the upper bound, r, the generalized ESD test essentially performs r separate tests: a test for one outlier, a test for two outliers, and so on up to r outliers.

The hypothesis of the generalized ESD test is:

|  |  |
| --- | --- |
| H0: | There are no outliers in the data set |
| Ha: | There are up to *r* outliers in the data set |
| Test Statistic: | Compute  Ri=maxi|xi−x¯| / s  with x¯ and ***s*** denoting the sample mean and sample standard deviation, respectively.  Remove the observation that maximizes |xi−x¯| and then recompute the above statistic with *n* - 1 observations. process is repeated until *r* observations have been removed. This results in the *r* test statistics *R1*, *R2*, ..., *Rr*. |

LEVEL OF SIGNIFICANCE:

Corresponding to the r test statistics, we compute the following r critical values:



where tp,ν is the 100p percentage point from the [t distribution](http://www.itl.nist.gov/div898/handbook/eda/section3/eda3664.htm) with ν degrees of freedom and



The number of outliers is determined by finding the largest i such that Ri > λi.

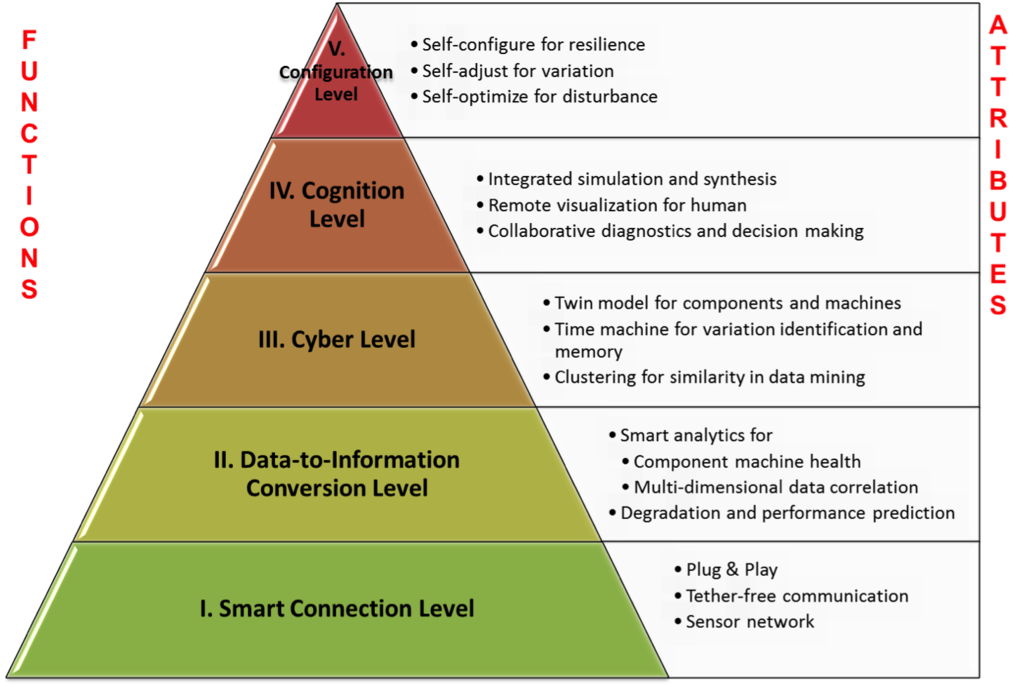
CYBER PHYSICAL SYSTEM:

A cyber-physical (also styled cyberphysical) system (CPS) is a mechanism that is controlled or monitored by computer-based algorithms, tightly integrated with the Internet and its users. In cyber-physical systems, physical and software components are deeply intertwined, each operating on different spatial and temporal scales, exhibiting multiple and distinct behavioral modalities, and interacting with each other in a myriad of ways that change with context. Examples of CPS include smart grid, autonomous automobile systems, medical monitoring, process control systems, robotics systems, and automatic pilot avionics.

CPS involves trans disciplinary approaches, merging theory of cybernetics, mechatronics, design and process science.The process control is often referred to as embedded systems. In embedded systems, the emphasis tends to be more on the computational elements, and less on an intense link between the computational and physical elements. CPS is also similar to the Internet of Things (IoT), sharing the same basic architecture; nevertheless, CPS presents a higher combination and coordination between physical and computational elements

Design:

Designing and deploying a cyber-physical production system can be done based on the 5C architecture (connection, conversion, cyber, cognition, and configuration). In the "Connection" level, devices can be designed to self-connect and self-sensing for its behavior. In the "Conversion" level, data from self-connected devices and sensors are measuring the features of critical issues with self-aware capabilities, machines can use the self-aware information to self-predict its potential issues. In the "Cyber" level, each machine is creating its own "twin" by using these instrumented features and further characterize the machine health pattern based on a "Time-Machine" methodology. The established "twin" in the cyber space can perform self-compare for peer-to-peer performance for further synthesis. In the "Cognition" level, the outcomes of self-assessment and self-evaluation will be presented to users based on an "infographic" meaning to show the content and context of the potential issues. In the "Configuration" level, the machine or production system can be reconfigured based on the priority and risk criteria to achieve resilient performance



**LITERATURE REVIEW:**

1. IJARCSSE 2012 Data Mining Techniques, Kalyani M Raval [1]

This paper defined data mining and explained the knowledge discovery database (KDD) process. Listed the data mining techniques which is: Association, Classification, Clustering, Prediction and sequential patterns

1. ACM 2009 Anomaly Detection: A Survey, Varun Chandola, Arindam Banerjee, And Vipin Kumar [2]

This paper provides structured and comprehensive overview of research on anomaly detection. It includes the definition, challenges, related work, various phases of anomaly detection problem, applications; several types of techniques etc. in short all about of anomaly detection.

1. ACM 2012 Time-Series Data Mining Philippe Esling And Carlos Agon [3]

This paper present the purpose of time-series data mining which is to try to extract all meaningful knowledge from the shape of data.

4 . ACM 2009 Detecting Anomalies in a Time Series Database, Varun Chandola, Deepthi Cheboli, Vipin [4]

This paper presents a comprehensive evaluation of semi-supervised anomaly detection techniques for time series data. The techniques can be grouped into four categories, i.e., kernel, window, predictive, and segmentation- based techniques.

5. Artificial Intelligence Review 2004 A Survey of Outlier Detection Methodologies, Victoria J. Hodge & Jim Austin [5]

This paper presents the methodology of the outlier detection and different approaches (supervised, semisupervised and unsupervised learning) of the same.

6. IEEE 2016 Anomaly Detection In Aircraft Data Using Recurrent Neural Networks (RNN), Anvardh Nanduri and Lance Sherry [6]

This paper describes the application of Recurrent Neural Networks (RNN) for effectively detecting anomalies in flight data.

7. International Journal of Computer Applications 2016 Anomaly based IDS using Backpropagation Neural Network, Vrushali D. Mane and S.N. Pawar [7]

This paper presents the Anomaly Intrusion Detection System that can detect various network attacks. The goal is to identify those attacks with the support of supervised neural network that is. back propagation artificial neural network algorithm and make complete data safe

8. IEEE 2011 Neural Network Approach to Real-Time Network Intrusion Detection and Recognition, Pavel Kachurka and Vladimir Golovko [8]

This paper introduced recirculation neural network based approach which detects previously undetected attack types in real-time mode and to more correct recognition of these types. The tests detained on both KDD data and real network traffic data prove that this approach can be used in host-based anomaly and misuse detectors.

9. IEEE 2009 Host Based Intrusion Detection Using RBF Neural Networks, Usman Ahmed and Asif Masood [9]

This paper presents a novel approach of host based intrusion detection that uses Radial basis Functions Neural Networks (RBFNNs) as profile containers.

10. IEEE 2016 Fuzzy Logic Inference for Unsupervised Anomaly Detection, Tetiana Gladkykh, Taras Hnot and Volodymyr Solskyy [10]

This paper introduced the solution for unsupervised anomaly detection i.e., to detect unexpected activity of user or network equipment, based on the analysis of mutual dependencies of the separate slices of network activity.

11. IEEE 2013 Anomaly Detection in Time Series Data using a Fuzzy C-Means Clustering, Hesam Izakian and Witold Pedrycz [11]

This paper presents anomalies in time series which are divided into two categories: amplitude anomalies and shape anomalies. A unified framework sustaining the detection of both types of anomalies is introduced.

12. Proceedings of the Second International Symposium on Networking and Network Security (ISNNS) 2010

An Anomaly Detection Method Based on Fuzzy C-means Clustering Algorithm, Linquan Xie, Ying Wang, Liping Chen, and Guangxue Yue [12]

This paper indicates the fuzzy C-means clustering (FCM) algorithm which applied to detect abnormality which based on network flow.

**DATASET:**

Daraset consists of 68 attributes .

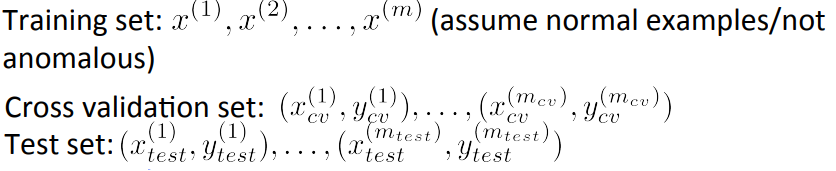
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Element** | | | | | **Unit or Range** | | | | **Resolution** | | | | | | **Description** | | |
| 1 | Date | | | | | MM/DD/YYYY | | | | -- | | | | | | Date of data record | | |
| 2 | Time | | | | | HH:MM | | | | -- | | | | | | Time of data record (local standard time) | | |
| 3 | Hourly extraterrestrial radiation on a horizontal surface | | | | | Watt-hour per square meter | | | | 1 Wh/m2 | | | | | | Amount of solar radiation received on a horizontal surface at the top of the atmosphere during the 60-minute period ending at the timestamp | | |
| 4 | Hourly extraterrestrial radiation normal to the sun | | | | | Watt-hour per square meter | | | | 1 Wh/m2 | | | | | | Amount of solar radiation received on a surface normal to the sun at the top of the atmosphere during the 60-minute period ending at the timestamp | | |
| 5 | Global horizontal irradiance | | | | | Watt-hour per square meter | | | | 1 Wh/m2 | | | | | | Total amount of direct and diffuse solar radiation received on a horizontal surface during the 60-minute period ending at the timestamp | | |
| 6 | Global horizontal irradiance source flag | | | | | 1-2 | | | | -- | | | | | | Table 1.1 | | |
| 7 | Global horizontal irradiance uncertainty | | | | | Percent | | | | 1% | | | | | | Uncertainty based on random and bias error estimates – see NSRDB User’s Manual (Wilcox, 2007b) | | |
| 8 | Direct normal irradiance | | | | | Watt-hour per square meter | | | | 1 Wh/m2 | | | | | | Amount of solar radiation (modeled) received in a collimated beam on a surface normal to the sun during the 60-minute period ending at the timestamp | | |
| 9 | Direct normal irradiance source flag | | | | | 1-2 | | | | -- | | | | | | Table 1.1 | | |
| 10 | Direct normal irradiance uncertainty | | | | | Percent | | | | 1% | | | | | | Uncertainty based on random and bias error estimates – see NSRDB User’s Manual (Wilcox, 2007b) | | |
| 11 | Diffuse horizontal irradiance | | | | | Watt-hour per square meter | | | | 1 Wh/m2 | | | | | | Amount of solar radiation received from the sky (excluding the solar disk) on a horizontal surface during the 60-minute period ending at the timestamp | | |
| 12 | Diffuse horizontal irradiance source flag | | | | | 1-2 | | | | -- | | | | | | Table 1.1 | | |
| 13 | Diffuse horizontal irradiance uncertainty | | | | | Percent | | | | 1% | | | | | | Uncertainty based on random and bias error estimates – see NSRDB User’s Manual (Wilcox, 2007b) | | |
| 14 | Global horizontal illuminance | | | | | Lux | | | | 100 lx | | | | | | Average total amount of direct and diffuse illuminance received on a horizontal surface during the 60-minute period ending at the timestamp | | |
| 15 | Global horizontal illuminance source flag | | | | | 1-2 | | | | -- | | | | | | Table 1.1 | | |
| 16 | Global horizontal illuminance uncertainty | | | | | Percent | | | | 1% | | | | | | Uncertainty based on random and bias error estimates – see section 2.10) | | |
| 17 | Direct normal illuminance | | | | | Lux | | | | 100 lx | | | | | | Average amount of direct normal illuminance received within a 5.7° field of view centered on the sun during 60-minute period ending at the timestamp | | |
| 18 | Direct normal illuminance source flag | | | | | 1-2 | | | | -- | | | | | | Table 1.1 | | |
| 19 | Direct normal illuminance uncertainty | | | | | Percent | | | | 1% | | | | | | Uncertainty based on random and bias error estimates – see section 2.10) | | |
| 20 | Diffuse horizontal illuminance | | | | | Lux | | | | 100 lx | | | | | | Average amount of illuminance received from the sky (excluding the solar disk) on a horizontal surface during the 60-minute period ending at the timestamp | | |
| 21 | Diffuse horizontal illuminance source flag | | | | | 1-2 | | | | -- | | | | | | Table 1.1 | | |
| 22 | Diffuse horizontal illuminance uncertainty | | | | | Percent | | | | 1% | | | | | | Uncertainty based on random and bias error estimates – see section 2.10) | | |
| 23 | Zenith luminance | | | | | Candela per square meter | | | | 10 cd/m2 | | | | | | Average amount of luminance at the sky's zenith during the 60-minute period ending at the timestamp | | |
| 24 | Zenith luminance source flag | | | | | 1-2 | | | | -- | | | | | | Table 1.1 | | |
| 25 | Zenith luminance uncertainty | | | | | Percent | | | | 1% | | | | | | Uncertainty based on random and bias error estimates – see section 2.10) | | |
| 26 | Total sky cover | | | | | Tenths of sky | | | | 1 tenth | | | | | | Amount of sky dome covered by clouds or obscuring phenomena at the time indicated | | |
| 27 | | | | Total sky cover flag (source) | | | | | | | | | Table 1.2 | | | | |
| 28 | | | | Total sky cover flag (uncertainty) | | | | | | | | | Table 1.3 | | | | |
| 29 | | Opaque sky cover | | | | | Tenths of sky | | | | 1 tenth | | | | Amount of sky dome covered by clouds or obscuring phenomena that prevent observing the sky or higher cloud layers at the time indicated | | |
| 30 | | | | Opaque sky cover flag (source) | | | | | | | | | Table 1.2 | | | | |
| 31 | | | | Opaque sky cover flag (uncertainty) | | | | | | | | | Table 1.3 | | | | |
| 32 | | Dry-bulb temperature | | | | | Degree C | | | | 0.1° | | | | Dry-bulb temperature at the time indicated | | |
| 33 | | | | Dry-bulb temperature flag (source) | | | | | | | | | Table 1.2 | | | | |
| 34 | | | | Dry-bulb temperature flag (uncertainty) | | | | | | | | | Table 1.3 | | | | |
| 35 | | Dew-point temperature | | | | | Degree C | | | | 0.1° | | | | Dew-point temperature at the time indicated | | |
| 36 | | | | Dew-point temperature flag (source) | | | | | | | | | Table 1.2 | | | | |
| 37 | | | | Dew-point temperature flag (uncertainty) | | | | | | | | | Table 1.3 | | | | |
| 38 | | Relative humidity | | | | | Percent | | | | 1% | | | | Relative humidity at the time indicated | | |
| 39 | | | | Relative humidity flag (source) | | | | | | | | | Table 1.2 | | | | |
| 40 | | | | Relative humidity flag (uncertainty) | | | | | | | | | Table 1.3 | | | | |
| 41 | | Station pressure | | | | | Millibar | | | | 1 mbar | | | | Station pressure at the time indicated | | |
| 42 | | | | Station pressure flag (source) | | | | | | | | | Table 1.2 | | | | |
| 43 | | | | Station pressure flag (uncertainty) | | | | | | | | | Table 1.3 | | | | |
| 44 | | Wind direction | | | | | Degrees from north (360° = north; 0° = undefined, calm) | | | | 10° | | | | Wind direction at the time indicated | | |
| 45 | | | | Wind direction flag (source) | | | | | | | | | Table 1.2 | | | | |
| 46 | | | | Wind direction flag (uncertainty) | | | | | | | | | Table 1.3 | | | | |
| 47 | | Wind speed | | | | | Meter/second | | | | 0.1 m/s | | | | Wind speed at the time indicated | | |
| 48 | | | | Wind speed flag (source) | | | | | | | | | Table 1.2 | | | | |
| 49 | | | | Wind speed flag (uncertainty) | | | | | | | | | Table 1.3 | | | | |
| 50 | | Horizontal visibility | | | | | Meter\* | | | | 1 m | | | | Distance to discernable remote objects at the time indicated (7777 = unlimited) | | |
| 51 | | | | Horizontal visibility flag (source) | | | | | | | | | Table 1.2 | | | | |
| 52 | | | | | | | | | Horizontal visibility flag (uncertainty) | | | | | | | | |
| 53 | | | Ceiling height | | | | | Meter\* | | | | 1 m | | | | | Height of the cloud base above local terrain (77777 = unlimited) | | |
| 54 | | | | | Ceiling height flag (source) | | | | | | | | | Table 1.2 | | | | | |
| 55 | | | | | Ceiling height flag (uncertainty) | | | | | | | | | Table 1.3 | | | | | |
| 56 | | | Precipitable water | | | | | Centimeter | | | | 0.1 cm | | | | | The total precipitable water contained in a column of unit cross section extending from the earth's surface to the top of the atmosphere | | |
| 57 | | | | | Precipitable water flag (source) | | | | | | | | | Table 1.2 | | | | | |
| 58 | | | | | Precipitable water flag (uncertainty) | | | | | | | | | Table 1.3 | | | | | |
| 59 | | | Aerosol optical depth, broadband | | | | | [unitless] | | | | 0.001 | | | | | The broadband aerosol optical depth per unit of air mass due to extinction by the aerosol component of the atmosphere | | |
| 60 | | | | | Aerosol optical depth, broadband flag (source) | | | | | | | | | Table 1.2 | | | | | |
| 61 | | | | | Aerosol optical depth, broadband flag (uncertainty) | | | | | | | | | Table 1.3 | | | | | |
| 62 | | | Albedo | | | | | [unitless] | | | | 0.01 | | | | | The ratio of reflected solar irradiance to global horizontal irradiance | | |
| 63 | | | | | Albedo flag (source) | | | | | | | | | Table 1.2 | | | | | |
| 64 | | | | | Albedo flag (uncertainty) | | | | | | | | | Table 1.3 | | | | | |
| 65 | | | Liquid precipitation depth | | | | | Millimeter\* | | | | 1 mm | | | | | The amount of liquid precipitation observed at the indicated time for the period indicated in the liquid precipitation quantity field | | |
| 66 | | | Liquid precipitation quantity | | | | | Hour\* | | | | 1 hr | | | | | The period of accumulation for the liquid precipitation depth field | | |
| 67 | | | | | Liquid precipitation depth flag (source) | | | | | | | | | Table 1.2 | | | | | |
| 68 | | | | | Liquid precipitation depth flag (uncertainty) | | | | | | | | | Table 1.3 | | | | | |

|  |  |
| --- | --- |
| **Table 1.1. Solar radiation and illuminance source flags Flag** | **Definition** |
| 1 | Data modeled using METSTAT or from 1961-1990 NSRDB solar fields |
| 2 | Data modeled using SUNY Satellite model (time shifted) |

|  |  |
| --- | --- |
| **Table 1.2. Meteorological source flags Flag** | **Definition** |
| A | Data as received from NCDC, converted to SI units |
| B | Linearly interpolated |
| C | Non-linearly interpolated to fill data gaps from 6 to 47 hours in length |
| D | Not used |
| E | Modeled or estimated, except: precipitable water, calculated from radiosonde data; dew point temperature calculated from dry bulb temperature and relative humidity; and relative humidity calculated from dry bulb temperature and dew point temperature |
| F | Precipitable water, calculated from surface vapor pressure; aerosol optical depth, estimated from geographic correlation |
| ? | Source does not fit any of the above. Used mostly for missing data |

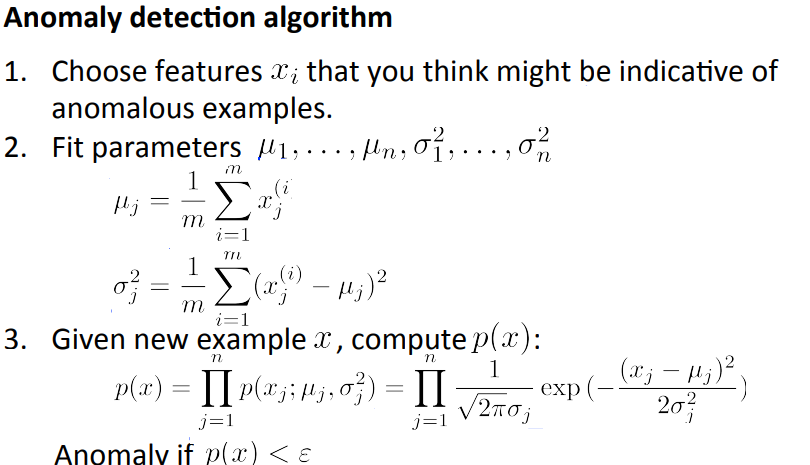
|  |  |
| --- | --- |
| **Table 1.3. Meteorological uncertainty flags Flag** | **Definition** |
| 1 – 6 | Not used |
| 7 | Uncertainty consistent with NWS practices and the instrument or observation used to obtain the data |
| 8 | Greater uncertainty than 7 because values were interpolated or estimated |
| 9 | Greater uncertainty than 8 or unknown |
| 0 | Not definable |

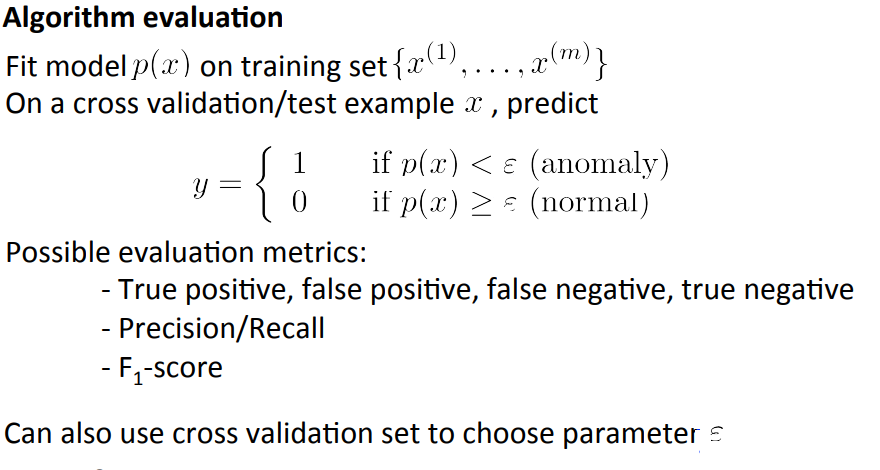
**Distribution of data**:



**PROCEDURE:**

**METHODOLOGY1:**





**METHODOLOGY2:**

**Algorithm : S-ESD Algorithm**

Input: X = A time series

n = number of observations in X

k = max anomalies (iterations in ESD)

Output: XA = An anomaly vector wherein each element is a tuple (timestamp, observed value)

Require: k ≤ (n × .49)

1. Extract seasonal component SX using STL Variant

2. Compute median X˜ /\* Compute residual \*/

3. RX = X − SX − X˜ /\* Detect anomalies vector XA using ESD \*/

4. XA = ESD(R, k)

return

**CODE**:

Initially data obtained from two data files are cleaned ,transformed and combined to get a single usable data set in MS Excel.

**EXCEL MACRO FOR SPLITTING DATASET BASED ON STATION NUMBER:**

Public Sub SplitToFiles()

' Loops through a specified column, and split each distinct values into a separate file by making a copy and deleting rows below and above.

' Files are saved in a "Split" subfolder from the location of the source workbook, and named after the section name.

Dim osh As Worksheet ' Original sheet

Dim iRow As Long ' Cursors

Dim iCol As Long

Dim iFirstRow As Long ' Constant

Dim iTotalRows As Long ' Constant

Dim iStartRow As Long ' Section delimiters

Dim iStopRow As Long

Dim sSectionName As String ' Section name (and filename)

Dim rCell As Range ' current cell

Dim owb As Workbook ' Original workbook

Dim sFilePath As String ' Constant

Dim iCount As Integer ' # of documents created

iCol = Application.InputBox("Enter the column number used for splitting", "Select column", 2, , , , , 1)

iRow = Application.InputBox("Enter the starting row number (to skip header)", "Select row", 5, , , , , 1)

iFirstRow = iRow

Set osh = Application.ActiveSheet

Set owb = Application.ActiveWorkbook

iTotalRows = osh.UsedRange.Rows.Count

sFilePath = Application.ActiveWorkbook.Path

If Dir(sFilePath + "\Split", vbDirectory) = "" Then

MkDir sFilePath + "\Split"

End If

'Turn Off Screen Updating Events

Application.EnableEvents = False

Application.ScreenUpdating = False

Do

' Get cell at cursor

Set rCell = osh.Cells(iRow, iCol)

sCell = Replace(rCell.Text, " ", "")

If sCell = "" Or (rCell.Text = sSectionName And iStartRow <> 0) Or InStr(1, rCell.Text, "total", vbTextCompare) <> 0 Then

' Skip condition met

Else

' Found new section

If iStartRow = 0 Then

' StartRow delimiter not set, meaning beginning a new section

sSectionName = rCell.Text

iStartRow = iRow

Else

' StartRow delimiter set, meaning we reached the end of a section

iStopRow = iRow - 1

' Pass variables to a separate sub to create and save the new worksheet

CopySheet osh, iFirstRow, iStartRow, iStopRow, iTotalRows, sFilePath, sSectionName, owb.fileFormat

iCount = iCount + 1

' Reset section delimiters

iStartRow = 0

iStopRow = 0

' Ready to continue loop

iRow = iRow - 1

End If

End If

' Continue until last row is reached

If iRow < iTotalRows Then

iRow = iRow + 1

Else

' Finished. Save the last section

iStopRow = iRow

CopySheet osh, iFirstRow, iStartRow, iStopRow, iTotalRows, sFilePath, sSectionName, owb.fileFormat

iCount = iCount + 1

' Exit

Exit Do

End If

Loop

'Turn On Screen Updating Events

Application.ScreenUpdating = True

Application.EnableEvents = True

MsgBox Str(iCount) + " documents saved in " + sFilePath

End Sub

Public Sub DeleteRows(targetSheet As Worksheet, RowFrom As Long, RowTo As Long)

Dim rngRange As Range

Set rngRange = Range(targetSheet.Cells(RowFrom, 1), targetSheet.Cells(RowTo, 1)).EntireRow

rngRange.Select

rngRange.Delete

End Sub

Public Sub CopySheet(osh As Worksheet, iFirstRow As Long, iStartRow As Long, iStopRow As Long, iTotalRows As Long, sFilePath As String, sSectionName As String, fileFormat As XlFileFormat)

Dim ash As Worksheet ' Copied sheet

Dim awb As Workbook ' New workbook

' Copy book

osh.Copy

Set ash = Application.ActiveSheet

' Delete Rows after section

If iTotalRows > iStopRow Then

DeleteRows ash, iStopRow + 1, iTotalRows

End If

' Delete Rows before section

If iStartRow > iFirstRow Then

DeleteRows ash, iFirstRow, iStartRow - 1

End If

' Select left-topmost cell

ash.Cells(1, 1).Select

' Clean up a few characters to prevent invalid filename

sSectionName = Replace(sSectionName, "/", " ")

sSectionName = Replace(sSectionName, "\", " ")

sSectionName = Replace(sSectionName, ":", " ")

sSectionName = Replace(sSectionName, "=", " ")

sSectionName = Replace(sSectionName, "\*", " ")

sSectionName = Replace(sSectionName, ".", " ")

sSectionName = Replace(sSectionName, "?", " ")

' Save in same format as original workbook

ash.SaveAs sFilePath + "\Split\" + sSectionName, fileFormat

' Close

Set awb = ash.Parent

awb.Close SaveChanges:=False

End Sub

**Principal Component Analysis(R):**

station1=read.csv("F:/Split\_for\_pca/690190.csv")

for(i in 1:ncol(station1)){ station1[is.na(station1[,i]), i] <- mean(station1[,i], na.rm = TRUE) }

> which(apply(station1, 2, var)==0)

RHum.uncert..code. Pwat.uncert..code. AOD.uncert..code. Alb.uncert..code.

11 23 25 27

> station1\_ok=station1[ , apply(station1, 2, var) != 0]

> dim(station1\_ok)

[1] 8760 40

> dim(station1)

[1] 8760 44

> station1pca=prcomp(station1\_ok,center=TRUE,scale.=TRUE)

> write.csv(file="F:/Split\_for\_pca/pca.csv",station1pca$x)

**Time series analysis (R )**

>library(AnomalyDetection)

> pca1=station1pca$x[,c(1)]

> pca1anomvec=AnomalyDetectionVec(pca1, max\_anoms=0.03, direction='both', plot=TRUE,period=100)

> pca1anomvec$plot

dim(pca1anomvec$anoms)

[1] 262 2

**3)MULTIVARIATE GAUSSIAN DISTRIBUTION (OCTAVE):**

**Main function:**

X=csvread("F:/Split\_for\_pca/pca\_rem.csv");

npca=14

X=X(:,1:npca);

% Estimate my and sigma2

[mu sigma2] = estimateGaussian(X);

% random initialization

A = 0; B = 1;

randomArray = (A-1) + (B-(A-1))\*rand(1,size(X)(1));

yval = floor(randomArray) + 1;

yval=yval';

for i=1:size(X)(1)

if X(i,1)> 0

yval(i)=1;

else

yval(i)=0;

end

end

% yval based on pooled mean and pooled vaiance.overwriting previous values

threshmean=mean(mu);

threshstd=sqrt(mean(sigma2));

ulim=threshmean+0.25\*threshstd;

llim=threshmean-0.25\*threshstd;

for i=1:size(X)(1)

if (llim<=mean(X(i))<=ulim)

yval(i)=0;

else

yval(i)=1;

end

end

pval = multivariateGaussian(X, mu, sigma2);

[epsilon F1] = selectThreshold(yval, pval);

fprintf('Best epsilon found using cross-validation: %e\n', epsilon);

fprintf('Best F1 on Cross Validation Set: %f\n', F1);

% Find the outliers in the training set

outliers = (pval < epsilon);

sum(outliers)

outliers;

**estiamteGaussian():**

function [mu sigma2] = estimateGaussian(X)

[m, n] = size(X);

mu = zeros(n, 1);

sigma2 = zeros(n, 1);

for i=1:n

mu(i,:)=mean(X(:,i));

sigma2(i,:)=var(X(:,i));

end

mu

sigma2

end

**multivariateGaussian():**

function p = multivariateGaussian(X, mu, Sigma2)

k = length(mu);

if (size(Sigma2, 2) == 1) || (size(Sigma2, 1) == 1)

Sigma2 = diag(Sigma2);

end

X = bsxfun(@minus, X, mu(:)');

p = (2 \* pi) ^ (- k / 2) \* det(Sigma2) ^ (-0.5) \* ...

exp(-0.5 \* sum(bsxfun(@times, X \* pinv(Sigma2), X), 2));

p

end

**selectThreshold():**

function [bestEpsilon bestF1] = selectThreshold(yval, pval)

bestEpsilon = 0;

bestF1 = 0;

F1 = 0;

stepsize = (max(pval) - min(pval)) / 1000;

for epsilon = min(pval):stepsize:max(pval)

predictions = (pval < epsilon);

tp = sum((predictions == 1) & (yval == 1));

fp = sum((predictions == 1) & (yval == 0));

fn = sum((predictions == 0) & (yval == 1));

prec = tp / (tp + fp);

rec = tp / (tp + fn);

F1 = 2 \* prec \* rec / (prec + rec);

if F1 > bestF1

bestF1 = F1;

bestEpsilon = epsilon;

end

end

end

**DOCUMENTATION:**

The original data contains 2 excel files,one for solar panel data and another for station data(latitude,longtitude,elevation etc.). These two files are merged using Station number as key.Some attributes like data source are deleted as they are of no use. Instrument value and uncertainity(%) and these are converted to the range of values using simple formula:

value-(uncertainity/100\*value) , value+(uncertainity/100\*value)

This is done in excel using simple math formula.This procedure is repeated for all attributes of solar panel data.

Also time and date which are two separate columns are merged to one column to get the timestamp.

The dataset is too big to load everything so it needed to be split according to station number.Also the timestamps for two station also overlap.This is done using macro program.

Now the dataset for each station is used separately. PCA is done individually for each station data and necessary components are taken. This value is saved.The value of the most significant PCA(upto 95% cumulative variance) is taken for time series analysis(Done using Twitter anomaly detection package which uses Seasonal Hybrid ESD algorithm).

The PCA values are then also used for multivariate Gaussian distribution .pooled mean and pooled variance of the PCA components are used to detect outliers .

Excel macro:

This macro will go through a specified column, top down, and split to a new file whenever a new value is encountered. Blanks or repeated values are kept together (as well as total rows).

The macro starts by prompting the user for the column to process, as well as the row number at which to start - that is to skip the headers, and goes from there.

When a section is identified, rather than copying those values to another sheet, the entire worksheet is copied to a new workbook and all rows below and above the section are deleted. This allows to keep any printing setup, conditional formatting, charts or whatever else useful, as well as keeping the header in each split file which is useful when distributing these files.

Files are saved in a \Split\ subfolder with the cell value as the filename

estimateGaussian**:**

This function returns mean (mu) and variance (sigma2) of columns of input matrix X(STEP 2 OF ALGORITHM)

multivariateGaussian**:**

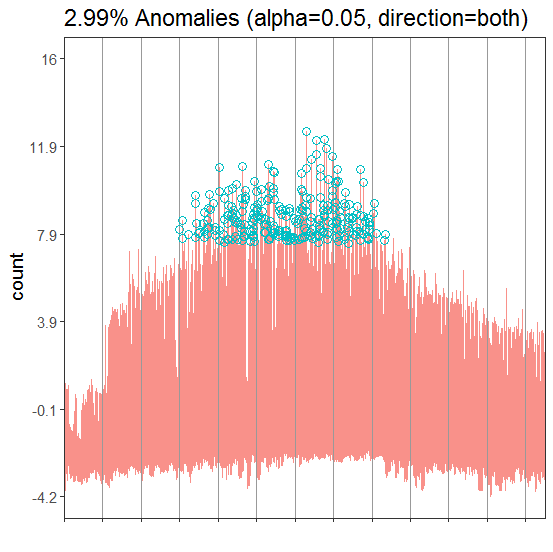
This function calculates the probability value p (STEP 3 OF ALGORITHM) using the given formula.

selecthreshold**:**

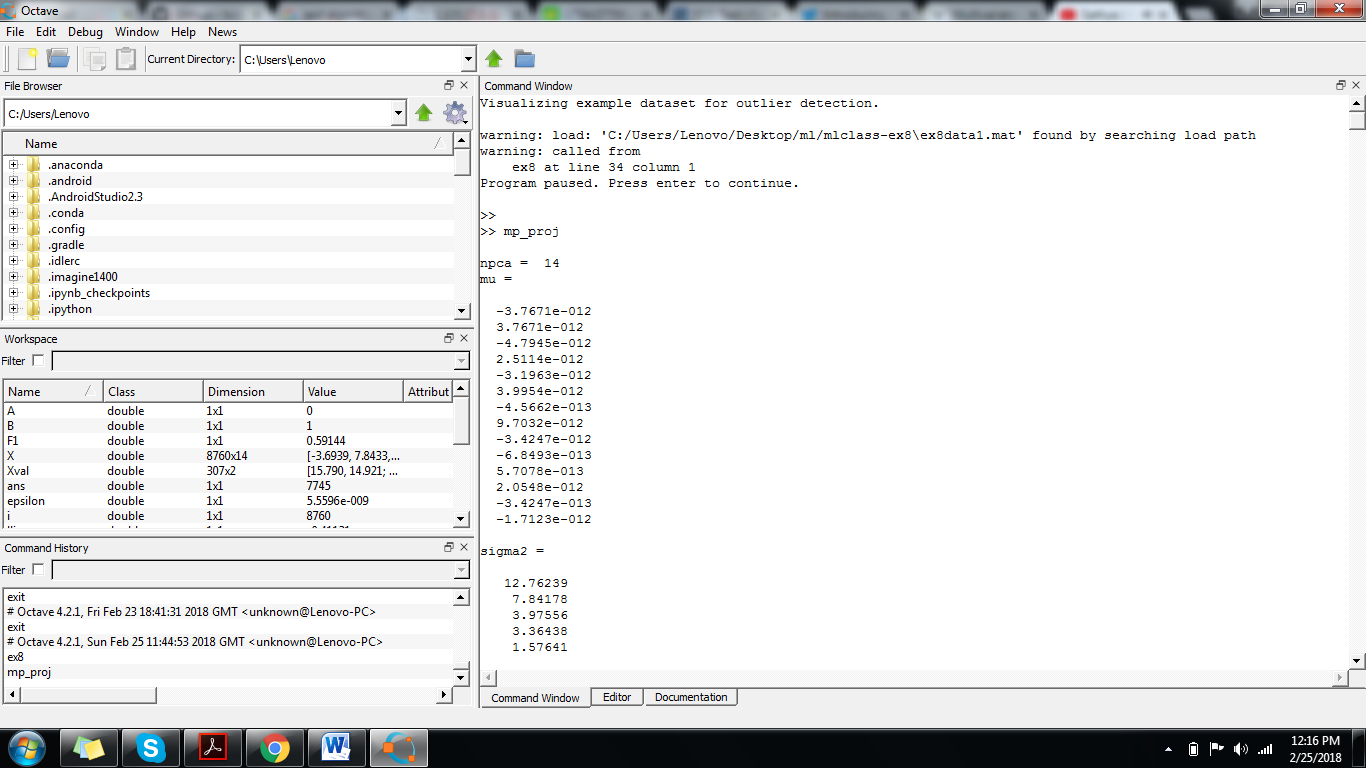
This function selects the epsilon value ‘e’ for cutoff value. This functions iterates over possible epsilon values((max(p)-min(p))/1000) and chooses the one with the best F score.

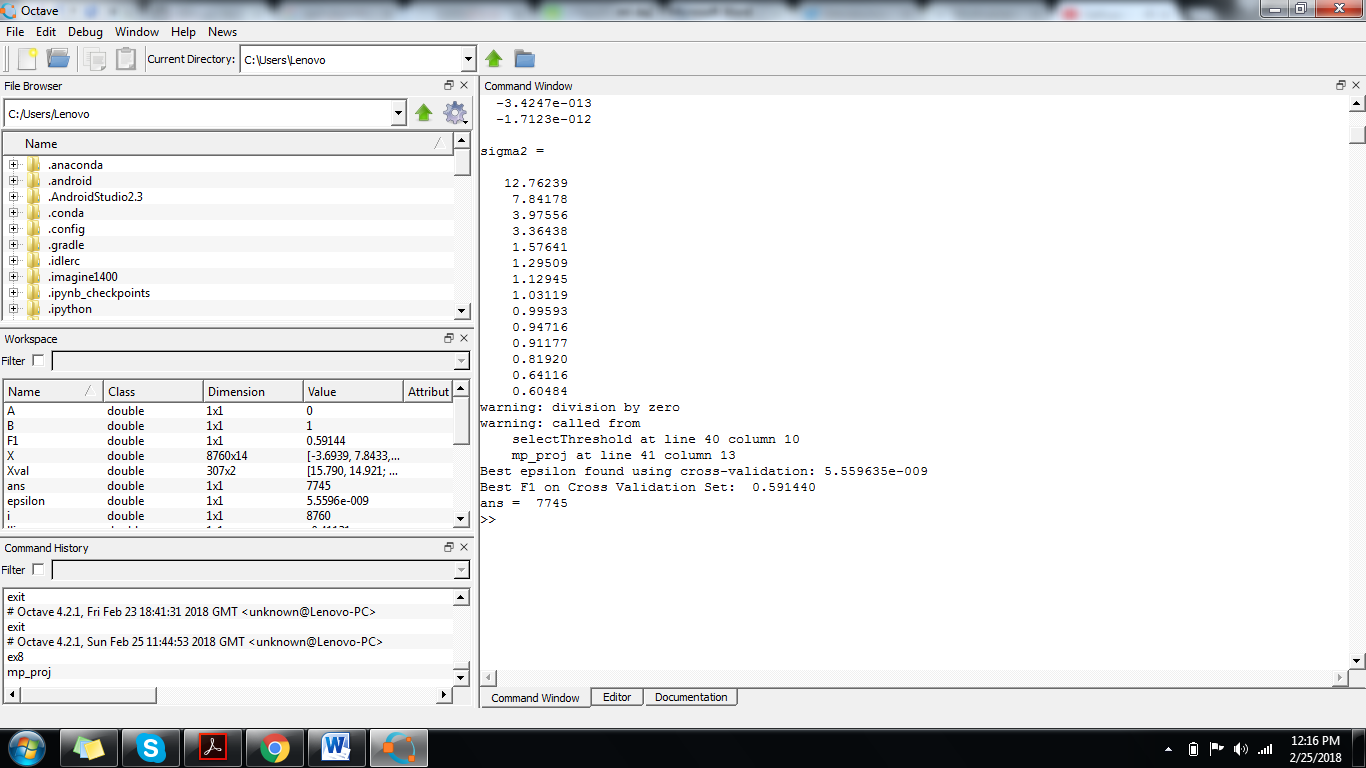
**SCREENSHOT**

TIME SERIES ANALYSIS:



Gaussian distribution:



**{\displaystyle {\boldsymbol {\Sigma }}}**

**RESULTS:**

The above mentioned procedure is done for 10 stations and the result of each station is as follows:

**Anomaly detection using gaussian distribution:**

***STATION1****:*

Best epsilon found using cross-validation: 8.176023e-009

Best F1 on Cross Validation Set: 0.617692

ans = 4421

ulim=threshmean+.625\*threshstd;

llim=threshmean-.625\*threshstd;

***STATION2:***

Best epsilon found using cross-validation: 4.345753e-008

Best F1 on Cross Validation Set: 0.547565

ans = 4756

ulim=threshmean+.59\*threshstd;

llim=threshmean-.59\*threshstd;

***STATION3:***

Best epsilon found using cross-validation: 9.201478e-009

Best F1 on Cross Validation Set: 0.630972

ans = 4014

ulim=threshmean+.625\*threshstd;

llim=threshmean-.625\*threshstd;

***STATION4:***

Best epsilon found using cross-validation: 4.452383e-009

Best F1 on Cross Validation Set: 0.585647

ans = 3791

ulim=threshmean+.627\*threshstd;

llim=threshmean-.627\*threshstd;

***STATION5:***

䢐Best epsilon found using cross-validation: 1.889349e-008

Best F1 on Cross Validation Set: 0.589333

ans = 4760

ulim=threshmean+.559\*threshstd;

llim=threshmean-.559\*threshstd;

***STATION6:***

堊Best epsilon found using cross-validation: 4.494104e-009

Best F1 on Cross Validation Set: 0.634697

ans = 3665

ulim=threshmean+.667\*threshstd;

llim=threshmean-.667\*threshstd;

***STATION7:***

Best epsilon found using cross-validation: 1.114201e-009

Best F1 on Cross Validation Set: 0.587305

ans = 2718

ulim=threshmean+.566\*threshstd;

llim=threshmean-.566\*threshstd;

***STATION8:***

Best epsilon found using cross-validation: 2.916088e-009

Best F1 on Cross Validation Set: 0.652585

ans = 3716

ulim=threshmean+.647\*threshstd;

llim=threshmean-.647\*threshstd;

***STATION9:***

Best epsilon found using cross-validation: 1.889349e-008

Best F1 on Cross Validation Set: 0.589333

ans = 4760

ulim=threshmean+.559\*threshstd;

llim=threshmean-.559\*threshstd;

***STATION10:***

Best epsilon found using cross-validation: 3.857624e-008

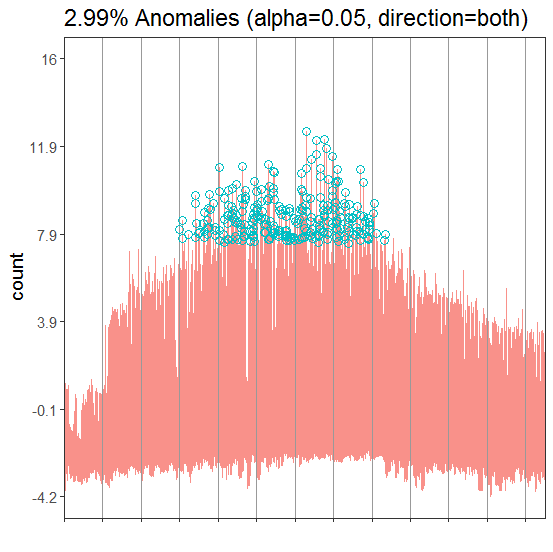
Best F1 on Cross Validation Set: 0.595404

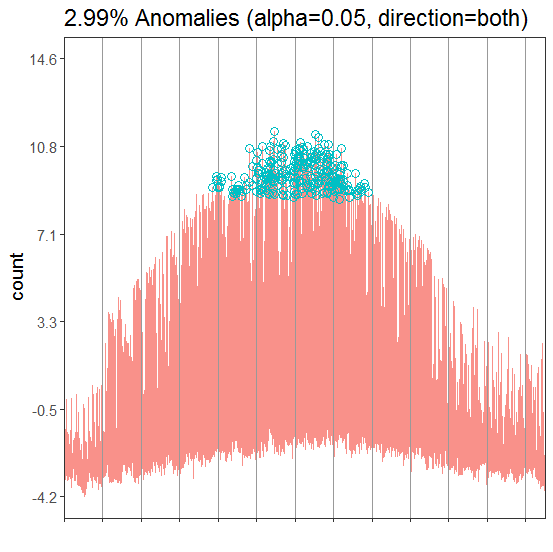
ans = 4735

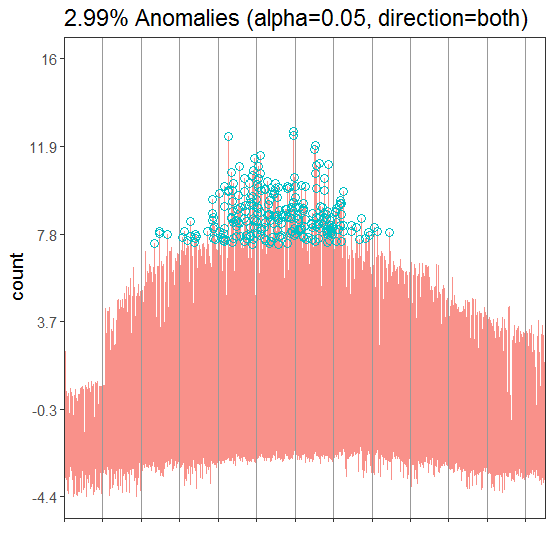
ulim=threshmean+.610\*threshstd;

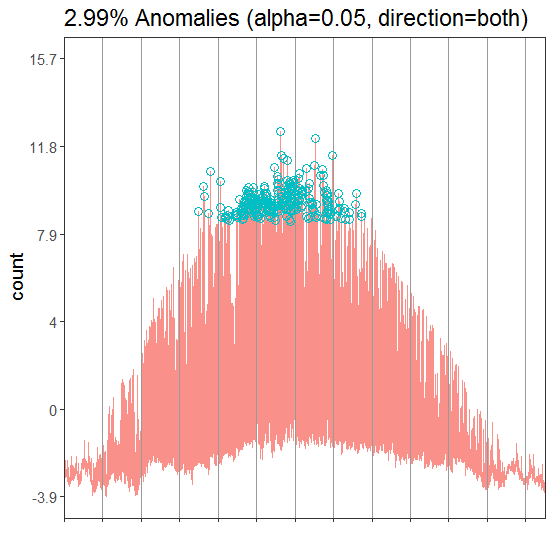
llim=threshmean-.610\*threshstd;

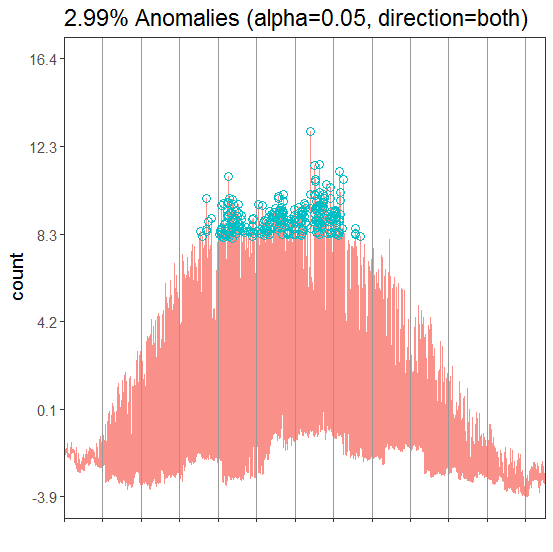
**TIME SERIES ANALYSIS:**

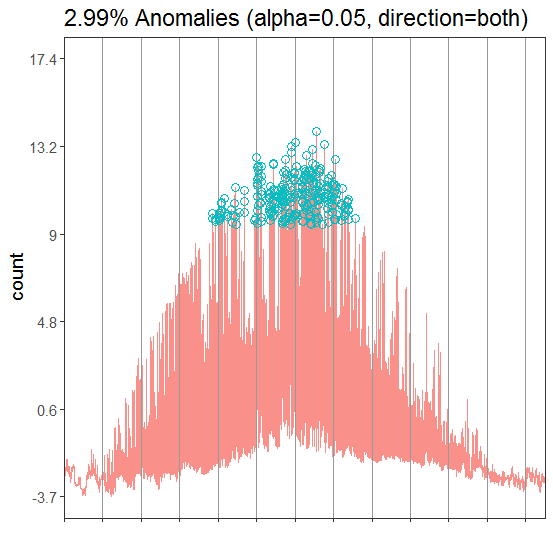


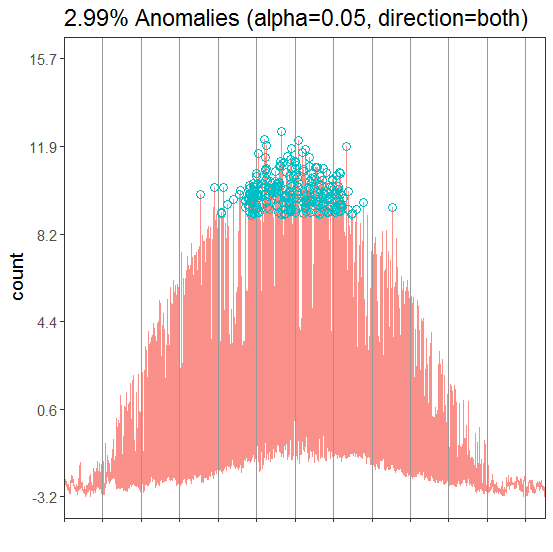


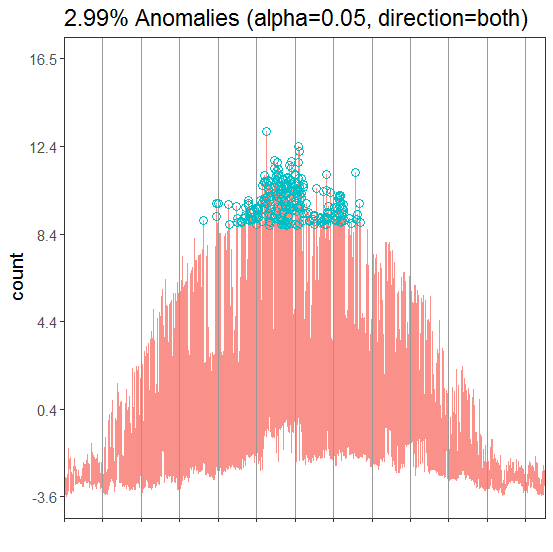


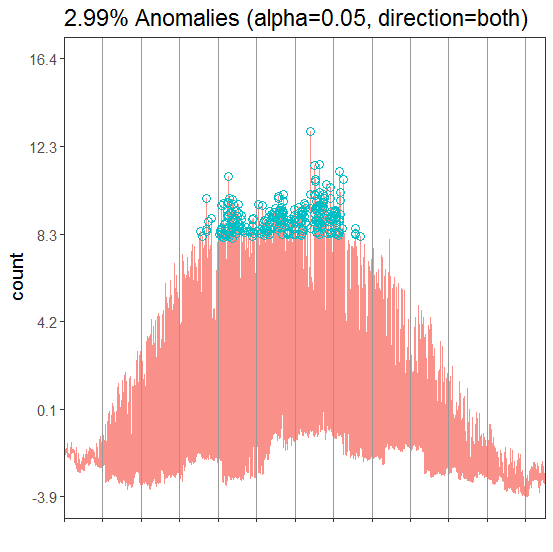


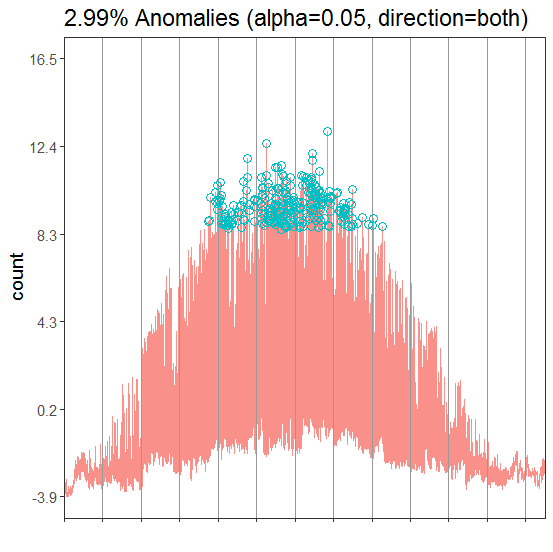












**REFERENCES:**

[1] J. Huysmans, B. Baesens, D. Martens, K. Denys And J. Vanthienen, New Trends in Data Mining, Tijdschrift voor Economie en Management, Vol. L, 4, 2005: 1-14.

[2] Varun Chandola, Arindam Banerjee and Vipin Kumar, Anomaly Detection: A Survey, ACM Computing Surveys, Vol. 41, No. 3, Article 15, 2009: 1-58.

[3] Animesh Patcha, Jung-Min Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, ScienceDirect 2007.

[4] Kalyani M Raval, Data Mining Techniques, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Volume 2, Issue 10, 2012: 439-442.

[5] Philippe Esling and Carlos Agon, Time-Series Data Mining, ACM Computing Surveys, Volume 45, No. 1, Article 12 (2012),: 1- 34.

[6] Varun Chandola, Deepthi Cheboli, and Vipin Kumar, Detecting Anomalies in a Time Series Database, ACM, Technical Report (2009).

[7] Victoria J. Hodge & Jim Austin, A Survey of Outlier Detection Methodologies, Artificial Intelligence Review 22 (2004): 85–126.

[8] Anvardh Nanduri and Lance Sherry, Anomaly Detection In Aircraft Data Using Recurrent Neural Networks (RNN), IEEE Integrated Communications Navigation and Surveillance (ICNS) Conference, 5C2-8(2016):19-21.

[9] Vrushali D. Mane and S.N. Pawar, Anomaly based IDS using Backpropagation Neural Network, International Journal of Computer Applications (0975 – 8887) Volume 136 – No.10 (2016):29-34.

[10] Pavel Kachurka and Vladimir Golovko, Neural Network Approach to Real-Time Network Intrusion Detection and Recognition, The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 15-17 (2011): 393-397.

[11] Usman Ahmed and Asif Masood, Host Based Intrusion Detection Using RBF Neural Networks, IEEE 2009 International Conference on Emerging Technologies (2009): 48-51.

[12] Tetiana Gladkykh, Taras Hnot and Volodymyr Solskyy, Fuzzy Logic Inference for Unsupervised Anomaly Detection, IEEE First International Conference on Data Stream Mining & Processing 23-27 (2016): 42-47.