T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

Kritik altyapılara ait sezgisel anomali tespiti
(Critical infrastructure anomaly detection using machine learning)

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

B151210554 - Konbil William Chol Deng

Bölüm : BİLGİSAYAR MÜHENDİSLİĞ

Danışman : Dr. Kevser Ovaz Akpınar

2020-2021 Güz Dönemi

Acknowledgments

I would like to appreciate:

Dr. Kevser Ovaz Akpinar,my supervisor for guiding me throughout this project.

Table of Contents

ABSTRACT(ÖZET)

Today,network intrusions are increasing on the daily.To be able to detect and tackle these suspicious network traffic from unlabelled data,it is very vital to understand network communication protocols.

In this report we present the performance of machine learning algorithms to analyse and identify outliers in the dataset without previous knowledge.

The procedures that we employ in this report include data extraction and Network Intrusion Detection.

The software applications used for these experiments are mainly Wireshark and RapidMiner. Wireshark is used here to view, filter and extract the required data.

This extracted data is then used as an input for the unsupervised learning algorithm for intrusion detection.

The techniques we have chosen for our implementation are Local Outlier Factor and K-means clustering.

Keywords:
Critical Infrastructure, Network Intrusion Detection,GOOSE,MMS,Machine Learning,Blaq_0,RapidMner,Unsupervised Learning,Local Outlier Factor,K-means clustering.

Chapter 1: INTRODUCTION(GİRİŞ)

Today, a number of people and various institutions across the globe communicate over the Internet.As 0f october 2020, approximately 4.66 billion people were active on the internet, that is 59 percent of the world's population.With all the technological advancements, cyber attacks have drastically increased.

The current intrusion and anomaly detection systems don't keep up with these threats.Hence the need to employ infrastructure intrusion detection algorithms to cut or reduce the impacts imposed by these attacks.In the electric substation setting,the standard IEC-61850 allows several Intelligent Electrical Devices (IEC) to communicate with each other.Despite the multiple advantages,researchers have discovered a couple of weaknesses ranging from lack of encryption in the GOOSE protocols due to firewalls not being implemented and absence of intrusion detection systems in the IEC-61850.

Our main task here is to be able to detect anomalies in our dataset.However our data is not ready to be fed into intrusion detection algorithms.

To be able to accomplish our task for the reports;

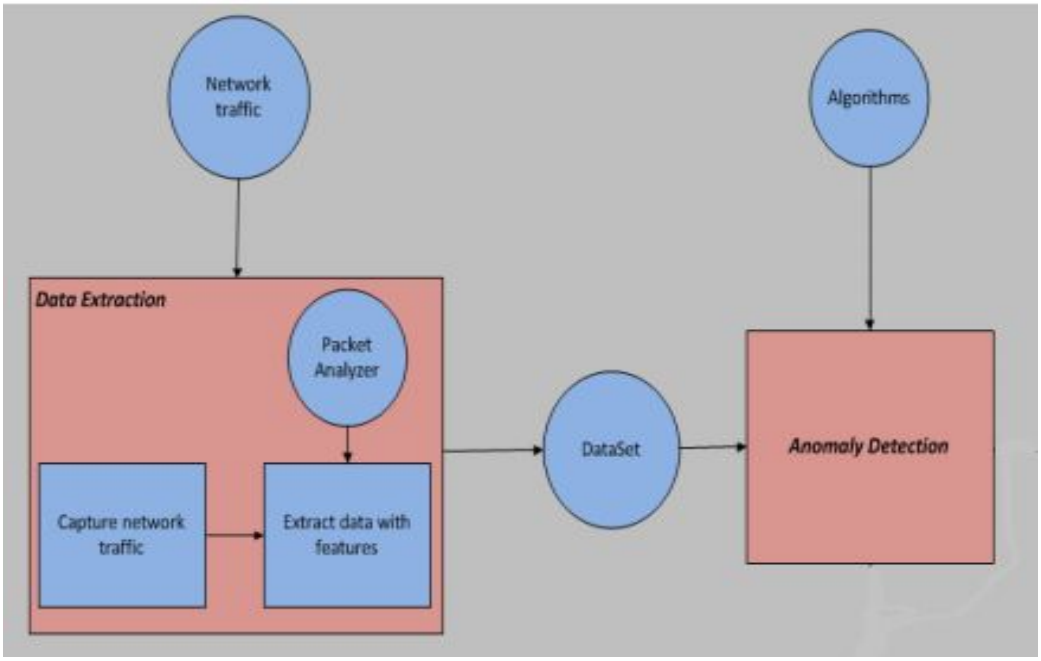
- We need to analyze and understand network traffic data (filter Generic Object-Oriented Substation Event (GOOSE) and Manufacturing Message Specification(MMS) supported by IEC-61850 protocols with the necessary features with the help of wireshark).
- Extract the data and normalize to a manner suitable for analysis.
- Determine the Machine Learning algorithm that can be used to detect outliers.
- Find the success level by comparing the performance of the results obtained in it with the studies previously conducted in this area of study.

OBJECTIVES:

The below points below are the targets to be realised for this report;

- Come up with the appropriate algorithms after carrying out extensive research on machine learning methodologies.
- Examine the IEC-61850 communication protocols.

The figure below is the overview illustration,we explain all the steps in detail.



CHAPTER 2:THE DATASET(VERİ SETİ)

In network anomaly detection,datasets which are both normal and anomalous are required for training and testing in the learning learning algorithms that we employ.

For our study,we will use blaq_0 dataset.This was organised for Singapore University of Technology and Design(SUTD) undergraduate students in Jan 2018.Independent attack teams

design and launch attacks on the EPIC dataset in which Network ‘pcapng’ for three days were collected.

This dataset was obtained from attacks launched on EPIC testbed.It is available inform of packet captured files with the extension ‘pcapng’

Details of the blaq_0 datasets can be seen below;

Team No.	Sub_dataset names.	Sub_dataset file	Pcapng size
1	Hammer Hackers	Hammer hackers	23,860kKB
2	Running	Part1 running Part2 running	3,737KB 14,565KB
3	Rainbow_rocket	Rainbow rocket	11,134KB
4	Spacehack_I	Empty	Empty

Chapter 3: DATA FILTERING AND EXTRACTION(VERİ ÇIKARMA)

Data extraction is the process of retrieving raw information out of data sources for further processing. Data extraction is mainly followed by data transformation to select features relevant for further procedures in the workflow.

From our PCAPs files,there are several network communication protocols.These include TCP,HTTP,GOOSE,MMS among others.In this dataset, we extract GOOSE AND MMS packets.

Filtering GOOSE AND MMS packets

In this section, we will have to analyze the pcaps files and filter GOOSE and MMS packets with the help of the Wireshark application.
But before we dive into our task,we will have to highlight briefly what wireshark is.It is a network packet analyzer that presents captured packet data in a much detailed manner. You can picture a network packet analyzer as a device for measuring and testing what's occurring inside a network cable, just like a doctor uses a thermometer for testing what's the temperature contained in an object.

In the past years, a tool like that was either very costly, proprietary, or the two. Though,with the advent of Wireshark, that is different. Wireshark is open source,available for free and regarded as one of the top packet analyzers available today.Wireshark can save network captured packets in various formats, even those that are used by other capture programs.

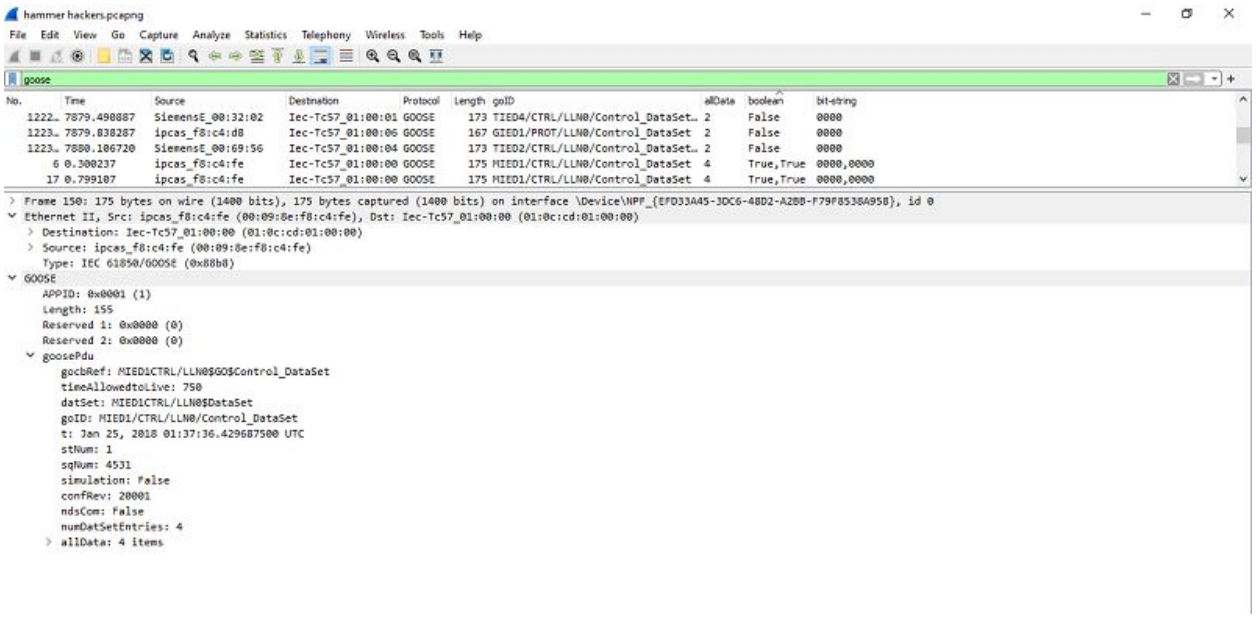
Filtering Generic Object-Oriented Substation Event (GOOSE) packets from the dataset:

The GOOSE model provides fast and reliable extensive distribution of input and output values. These messages are embedded via ethernet multicast.
From the goose dataframe,we filter the necessary features that will help us to train and test our algorithms to find out the anomalies.
Some of the features that we will extract with the help of wireshark from the goose dataframe parts of Ethernet,Goose pdus include the following;

- No.
- Time
- Source
- Destination

- Protocol
- Length
- goID
- Cmd(Command)
- AllData
- Boolean
- Bitstring

The figure below shows the GOOSE frame in wireshark captured.It indicates all the parts of the GOOSE protocol frame.



Screenshot of GOOSE message frame.

In GOOSE pdus,tag,length and data is their arranged order,you can be able to observe that you take a closer look at the figure above.

The goose pdu data is encoded using the abstract syntax notation one.

The information represented by the data is shown by Tag.

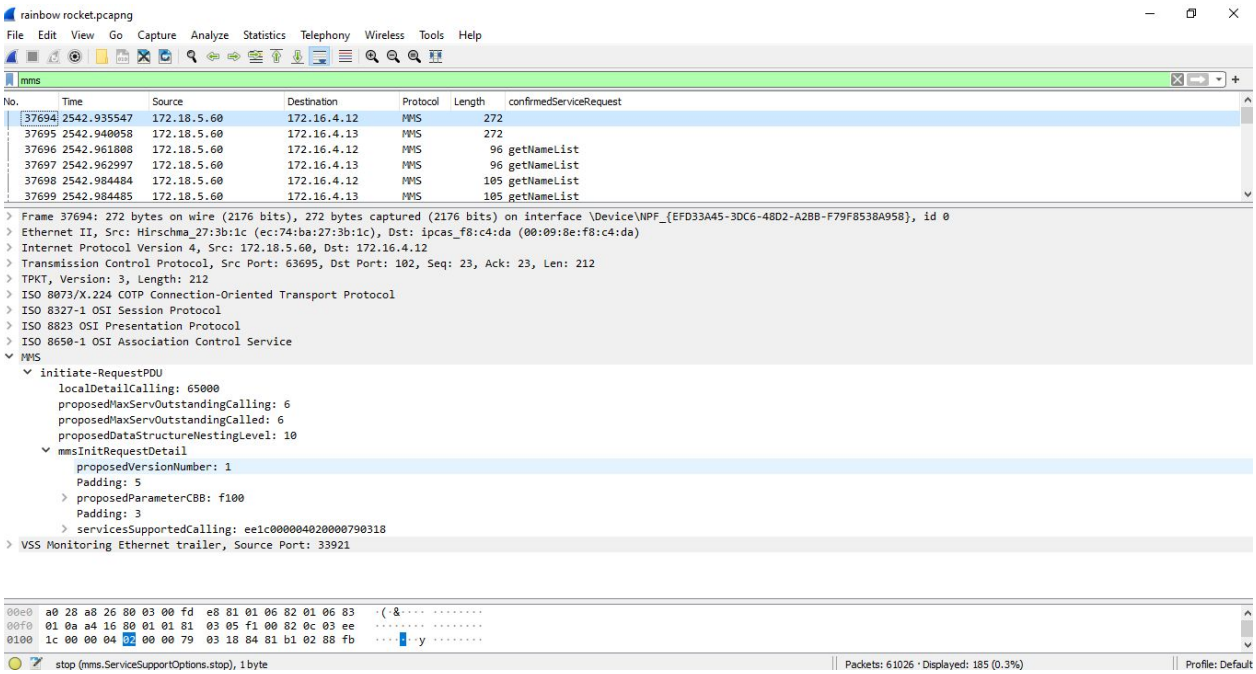
- Length shows the number of bytes by data
- gocbRef: names the GOOSE control block..

- timeAllowedtoLive: After a packet is being transmitted,it shows the time taken.
- dataSet: It references the dataset whose member's values are transmitted.
- goID: It indicates the GOOSE ID and the size is 7 bytes.
- timestamp: Timestamp for each GOOSE message (8 bytes).
- stNum: As goose messages are generated as a result in the change of events,this number is assigned.
- sqNum: This number is assigned to the re-transmitted messages in increasing order (1 byte).
- Test: When in the test mode,this bit is set.
- ConfRev: Indicates the version of Intelligent electronics device(IED).
- alldata: the data of all the elements in the data set.

Filtering Manufacturing Message Specification (MMS) packets with wireshark from the dataset.

From the blaq_0 dataset, with the help of wireshark,we filter MMS packets which are specifically found in the 'rainbow rocket ' pcapng file, a subsection of our main dataset.

Below is a figure showing the MMS packets filtration frame in a wireshark.

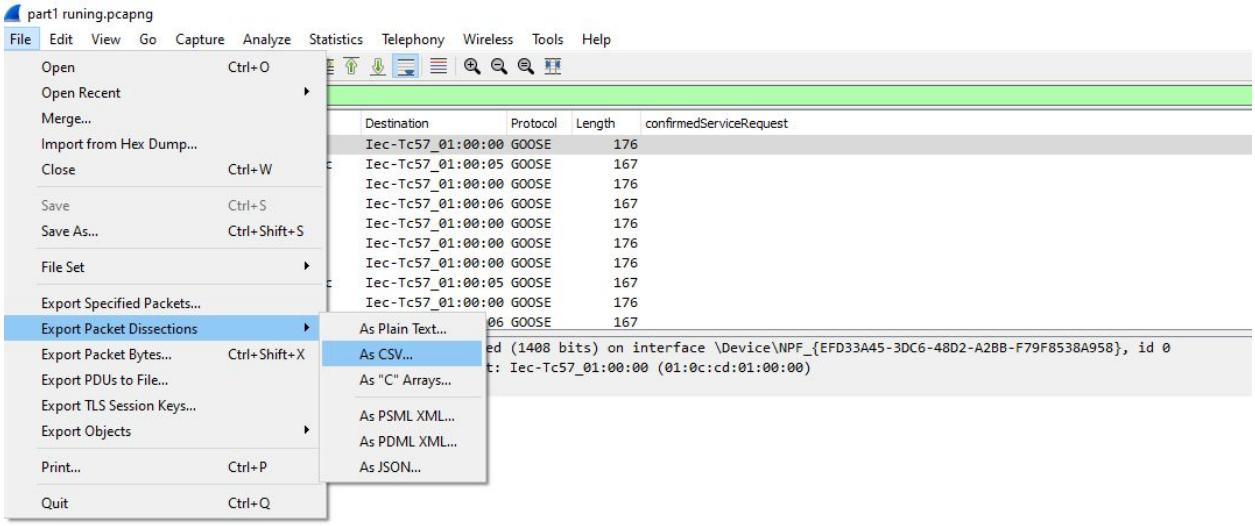


The Summary of filtered GOOSE and MMS protocol

In the process of filtering our dataset, we found out in which subsets of the data are the GOOSE and MMS protocol available.The Files of the datasets which contain these protocols are presented as below;

- Hammer hacker file has GOOSE protocol including other protocols except MMS.It contains 54,832 goose packets.
- Running file.This file has GOOSE protocols except MMS.It contains 1668 and 34796 in ‘running 1’ and ‘running 2’ goose packets respectively.
- Rainbow Rocket file.This file has both GOOSE and MMS packets among others.It contains 28978 GOOSE and 185 MMS packets.
- Spacehack_I is an empty file.

When done with the filtration of the packets we export with the .CSV extension.This is done as shown in the figure below.

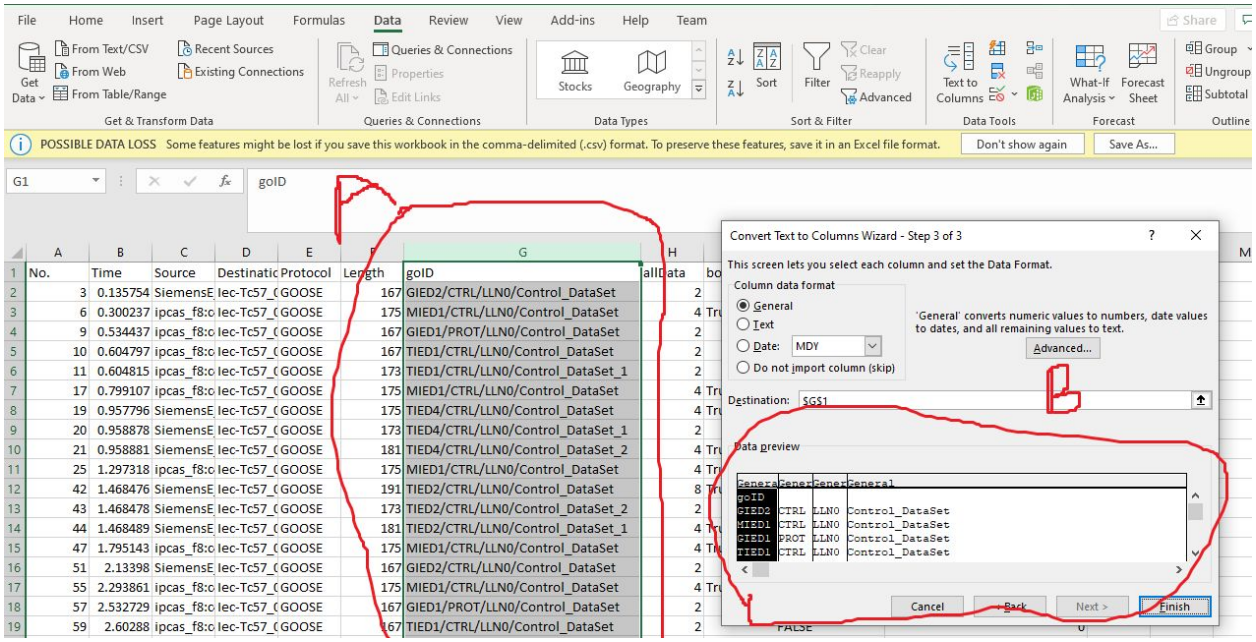


Our mission of exporting the packets as CSV is accomplished.

Parsing the dataset.

Paring is splitting a column into multiple.

If you look at the exported CSVs dataset files carefully,you will notice that some of the columns contain more than a data and these data is separated by a comma(,),forward slash(/),sharp(#) or what character might have been used for the separation.So here, we will have to split the columns such that we can get our data in an independent column.



The figure above shows the process of splitting the data as seen from part A to B.

The goose packet features that we split are the goID,boolean,and bitstring.

We came up with our names for the features.

These names are as follow;

For goID:

- goID
- Cmd
- llno
- parameter

For boolean:

- Item_1
- Item_2
- Item_3
- Item_4

For bitstring:

- Bit-str1
- Bit-str2
- Bit-str3

- Bit-str4

The features of the MMS packet that we split are;

confirmedServiceRequest divided into:

- confirmedServiceRequest1
- confirmedServiceRequest2

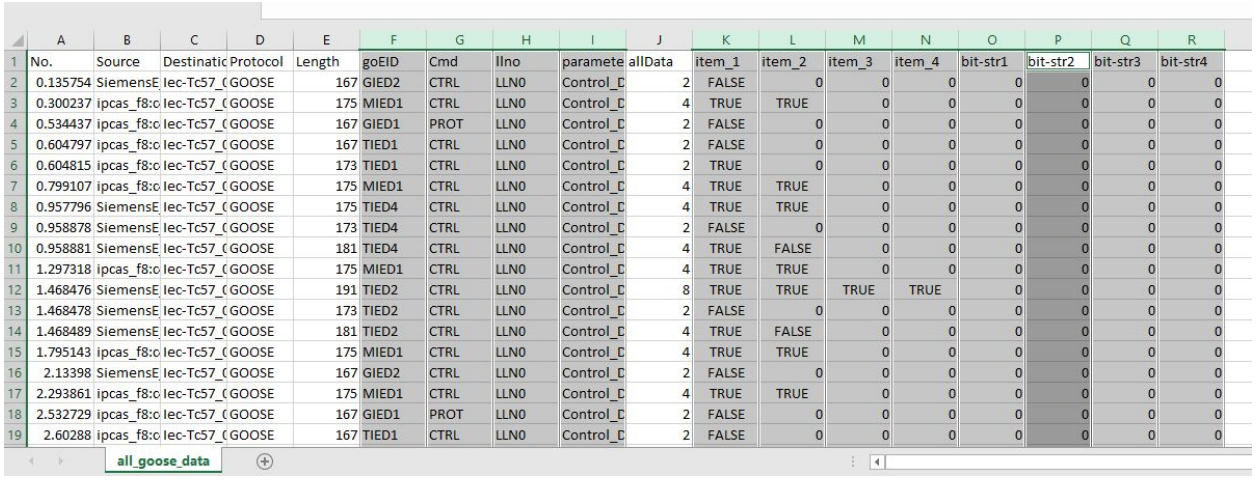
specificationWithResult parsed into:

- specificationWithResult1
- specificationWithResult2

And the last parsed feature happen to be Info divided into:

- Info1
- Info2
- Info3
- Info4

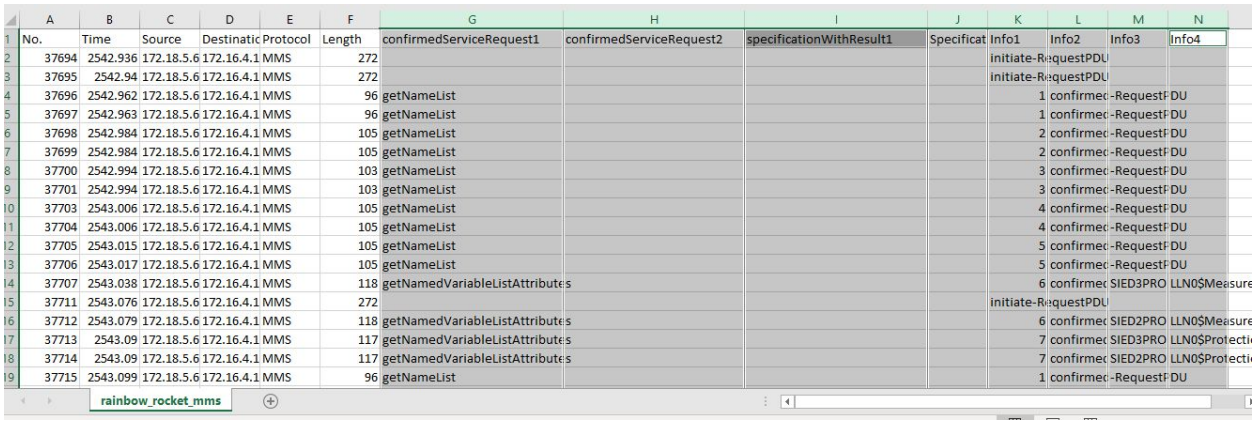
The final GOOSE and MMS packets in CSV with the relevant features will appear as below;



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	No.	Source	Destinatic	Protocol	Length	goEID	Cmd	llno	paramete	allData	item_1	item_2	item_3	item_4	bit-str1	bit-str2	bit-str3	bit-str4
2	0.135754	SiemensE	Iec-Tc57_(	GOOSE	167	GIED2	CTRL	LLN0	Control_D	2	FALSE	0	0	0	0	0	0	0
3	0.300237	ipcas_f8:c	Iec-Tc57_(	GOOSE	175	MIED1	CTRL	LLN0	Control_D	4	TRUE	TRUE	0	0	0	0	0	0
4	0.534437	ipcas_f8:c	Iec-Tc57_(	GOOSE	167	GIED1	PROT	LLN0	Control_D	2	FALSE	0	0	0	0	0	0	0
5	0.604797	ipcas_f8:c	Iec-Tc57_(	GOOSE	167	TIED1	CTRL	LLN0	Control_D	2	FALSE	0	0	0	0	0	0	0
6	0.604815	ipcas_f8:c	Iec-Tc57_(	GOOSE	173	TIED1	CTRL	LLN0	Control_D	2	TRUE	0	0	0	0	0	0	0
7	0.799107	ipcas_f8:c	Iec-Tc57_(	GOOSE	175	MIED1	CTRL	LLN0	Control_D	4	TRUE	TRUE	0	0	0	0	0	0
8	0.957796	SiemensE	Iec-Tc57_(	GOOSE	175	TIED4	CTRL	LLN0	Control_D	4	TRUE	TRUE	0	0	0	0	0	0
9	0.958878	SiemensE	Iec-Tc57_(	GOOSE	173	TIED4	CTRL	LLN0	Control_D	2	FALSE	0	0	0	0	0	0	0
10	0.958881	SiemensE	Iec-Tc57_(	GOOSE	181	TIED4	CTRL	LLN0	Control_D	4	TRUE	FALSE	0	0	0	0	0	0
11	1.297318	ipcas_f8:c	Iec-Tc57_(	GOOSE	175	MIED1	CTRL	LLN0	Control_D	4	TRUE	TRUE	0	0	0	0	0	0
12	1.468476	SiemensE	Iec-Tc57_(	GOOSE	191	TIED2	CTRL	LLN0	Control_D	8	TRUE	TRUE	TRUE	TRUE	0	0	0	0
13	1.468478	SiemensE	Iec-Tc57_(	GOOSE	173	TIED2	CTRL	LLN0	Control_D	2	FALSE	0	0	0	0	0	0	0
14	1.468489	SiemensE	Iec-Tc57_(	GOOSE	181	TIED2	CTRL	LLN0	Control_D	4	TRUE	FALSE	0	0	0	0	0	0
15	1.795143	ipcas_f8:c	Iec-Tc57_(	GOOSE	175	MIED1	CTRL	LLN0	Control_D	4	TRUE	TRUE	0	0	0	0	0	0
16	2.13398	SiemensE	Iec-Tc57_(	GOOSE	167	GIED2	CTRL	LLN0	Control_D	2	FALSE	0	0	0	0	0	0	0
17	2.293861	ipcas_f8:c	Iec-Tc57_(	GOOSE	175	MIED1	CTRL	LLN0	Control_D	4	TRUE	TRUE	0	0	0	0	0	0
18	2.532729	ipcas_f8:c	Iec-Tc57_(	GOOSE	167	GIED1	PROT	LLN0	Control_D	2	FALSE	0	0	0	0	0	0	0
19	2.60288	ipcas_f8:c	Iec-Tc57_(	GOOSE	167	TIED1	CTRL	LLN0	Control_D	2	FALSE	0	0	0	0	0	0	0

all_goose_data

Figure:GOOSE packets.



No.	Time	Source	Destinatic	Protocol	Length	confirmedServiceRequest1	confirmedServiceRequest2	specificationWithResult1	Specificat	Info1	Info2	Info3	Info4
37694	2542.936	172.18.5.6	172.16.4.1	MMS	272					initiate-R	equestPDU		
37695	2542.94	172.18.5.6	172.16.4.1	MMS	272					initiate-R	equestPDU		
37696	2542.962	172.18.5.6	172.16.4.1	MMS	96	getNameList				1	confirmed	-RequestP	DU
37697	2542.963	172.18.5.6	172.16.4.1	MMS	96	getNameList				1	confirmed	-RequestP	DU
37698	2542.984	172.18.5.6	172.16.4.1	MMS	105	getNameList				2	confirmed	-RequestP	DU
37699	2542.984	172.18.5.6	172.16.4.1	MMS	105	getNameList				2	confirmed	-RequestP	DU
37700	2542.994	172.18.5.6	172.16.4.1	MMS	103	getNameList				3	confirmed	-RequestP	DU
37701	2542.994	172.18.5.6	172.16.4.1	MMS	103	getNameList				3	confirmed	-RequestP	DU
37703	2543.006	172.18.5.6	172.16.4.1	MMS	105	getNameList				4	confirmed	-RequestP	DU
37704	2543.006	172.18.5.6	172.16.4.1	MMS	105	getNameList				4	confirmed	-RequestP	DU
37705	2543.015	172.18.5.6	172.16.4.1	MMS	105	getNameList				5	confirmed	-RequestP	DU
37706	2543.017	172.18.5.6	172.16.4.1	MMS	105	getNameList				5	confirmed	-RequestP	DU
37707	2543.038	172.18.5.6	172.16.4.1	MMS	118	getNamedVariableListAttributes				6	confirmed	SIED3PRO	LLN0$Measure
37711	2543.076	172.18.5.6	172.16.4.1	MMS	272					initiate-R	equestPDU		
37712	2543.079	172.18.5.6	172.16.4.1	MMS	118	getNamedVariableListAttributes				6	confirmed	SIED2PRO	LLN0$Measure
37713	2543.09	172.18.5.6	172.16.4.1	MMS	117	getNamedVariableListAttributes				7	confirmed	SIED3PRO	LLN0$Protecti
37714	2543.09	172.18.5.6	172.16.4.1	MMS	117	getNamedVariableListAttributes				7	confirmed	SIED2PRO	LLN0$Protecti
37715	2543.099	172.18.5.6	172.16.4.1	MMS	96	getNameList				1	confirmed	-RequestP	DU

rainbow_rocket_mms

Figure:MMS packets.

Merging the files:

In the Blaq_0 dataset,we have a few subsets as specified in the dataset background.So after filtration of each sub_dataset according to our required communication protocols.
We discovered that MMS packets are only found in a single file unlike GOOSE packet filtered from subsets of the dataset and need to be merged to a single file.
Since our GOOSE packets are CSV files ,we place all the GOOSE CSV files in one folder.
(In the root directory e.g C:\User\Desktop\GOOSE).
The next thing is opening the command prompt and moving to the GOOSE file directory.Make sure that the required files are available.
Type “copy *.csv goose_dataset csv in the command prompt.This merges our csv files.

A figure showing the above merging procedures.

```
Command Prompt
Microsoft Windows [Version 10.0.190  .68. ]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\           >cd C:\Users\\          \Desktop\GOOSE

C:\Users\ :         \Desktop\GOOSE>copy *.csv goose_dataset.csv
hammer_hackers.csv
part1_running.csv
part2_running.csv
rainbow_rocket_goose.csv
        1 file(s) copied.
```

Chapter 4 :ANOMALY DETECTION

Anomaly Detection model operates by identifying malicious activity by observing network behaviours which deviate from normal traffic norm.

As far as our approach is concerned,we apply unsupervised anomaly detection since we are working with data which is unlabelled.This algorithm seems to be the most flexible setup technique for intrusion detection.Here there is no previous knowledge about the data needed and it does not have distinct training and test dataset to obtain results.

Below is how this method works,the unlabeled data is given to the algorithm as input to score our goal.

Before we go further,we highlight that Unsupervised Learning detection depends on the selection of features.

In the previous section, we successfully selected our features by filtration in wireshark. Therefore, our extracted dataset with the relevant features is the input for identifying anomalies. In the process of anomaly detection,the values of the features belonging to the dataset are used in statistical measure calculations.The calculations measure range from one method of anomaly

detection to another.These malicious behaviours are referred to as anomalies(outliers).In the dataset,outliers differ from the normal network traffic significantly hence raises suspicions.So in the intrusion detection system,the suspicions are the results of an anomaly.

The decision of the results depend on the model of intrusion detection chosen.And the algorithms have varied models of calculation of anomaly(outlier) score.Outlier score can be defined as a ranking score of the data owing to the anomaly detection methodology.It is important to note that the threshold decided plays a big role in detecting anomalies.Low accuracy and ratio of high false positive may portray threshold poorly chosen.

Outlier score referred to as the recent dataset with new attributes for every instance as a result of the anomaly detection technique processing all the instances of the dataset.We will probably have some alerts since some instances of the outlier score will be higher in regards to the baseline.Let's take instance that the outlier score range is(0.1-5),and make instances greater than number one anomalies.In the example,number one is the baseline which means outlier score instances less than one are considered benign.

Whatever machine learning algorithm for intrusion detection that we work with,the beginning will be our filtered dataset and of course end up with a new dataset made up of alerts and outlier score.

We will now have to define the different algorithms that can fit for intrusion detection next.

Methods of Anomaly detection

In network intrusion detections,machine learning algorithms are mainly used for anomaly detection.There are several approaches to solve outliers issues.Unsupervised algorithms do not require labelled data.

Anomaly detection techniques could fall under either global or local.Algorithms in which anomaly score depends on each instance in coordination with the whole dataset is referred to as a global approach.While the data points directly to its neighbourhood,the outlierness of anomaly score is represented with local techniques.In the scenario of densities varied within the dataset,local techniques can identify outliers that can not be detected by global approaches.

Anomaly detection algorithms are distinguished into two approaches,that is clustering and nearest neighbour based algorithms.Clustering algorithms is faster since it works on the results of

the output.Nearest neighbour algorithms assume that anomalies are far from their nearest neighbours and are found in the sparse neighbourhoods.

Below we describe,the extension of these algorithms:

A. Clustering-based algorithms
    a. Cluster based Local Outlier Factor (CBLOF):here the dataset is used as an input and output cluster model depending on a clustering algorithm.Outliers score calculation is done in relation to the size of the cluster and the distance to the vast cluster's centroid.
    b. Local Density Cluster based Outlier factor (LDCOF):just like CBLOF,outliers are calculated by using how far the nearest big cluster is divided by the average cluster distance of the huge cluster.
B. Nearest-Neighbor based algorithms
    a. K-NN Global Anomaly Score:it calculates the nearest average distance to identify outliers score.We have an anomaly if the outlier score is high.
    b. Connectivity based Outlier Factor (COF) :it has the ability to handle outliers as an outcome of low density patterns.
    c. Local Outlier Probability (LoOP) :this algorithm does not ignore any distribution for the dataset hence the calculation technique of the statistical algorithms.
    d. Influenced Outlierness (INFLO):while calculating the density at a specific point,outliers are detected by considering the neighbors and the reversed neighbors.
    e. Local Correlation Integral (LOCI):this determines outliers by employing automatic statistical reject mechanisms.

Algorithms

Local outlier Factor(LOF) and K-means clustering are the anomaly detection algorithms we use in our exercises.Let's now dive into more details about the algorithms.

Local Outlier Factor(LOF)

We define outlier as a data point which is distinct from the other data points.The main question at the back of our minds is simply how we can detect these outliers in the dataset?

The unsupervised anomaly detection techniques which identifies these outliers available in the dataset.

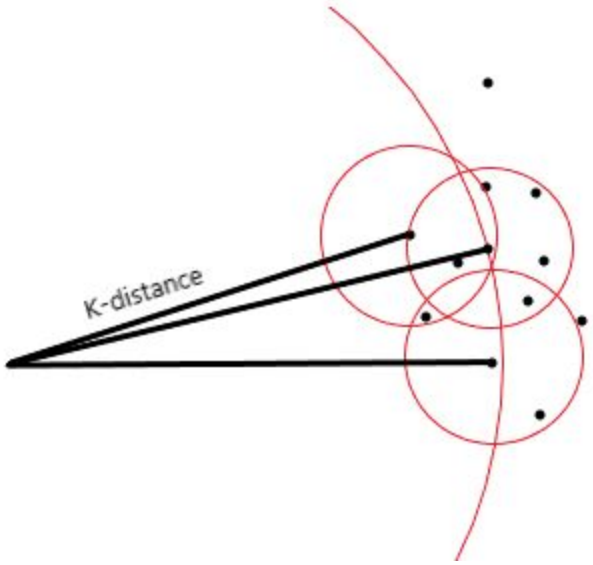
Local outlier is when a data point is based on its local neighborhood.

LOF takes into consideration the density of the neighborhood inorder to identify anomalies.In Case the density of the data is not similar in the entire dataset,the performance seems to be better.

Local density can be calculated for each data point.We can examine which data points got related densities and which have lesser density compared to its neighbours.The lesser densities are regarded as outliers.

The distances between data points that are calculated for each point to find out their k-nearest neighbors is referred to as k-distances.

Below is an image which represents k-distances of various neighbors in the clusters.



Reachability distance(RD) can be computed using the distance shown in the figure above.we refer to RD as the maximum distance between two data points and the k-distance of that point. The formula here;

RD(Xi,Xj) = max(K-distance(Xj), distance (Xi,Xj))

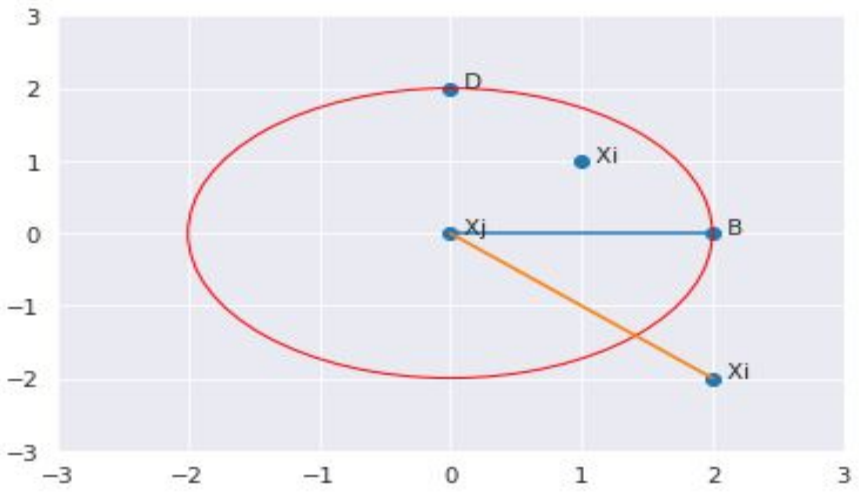


Illustration of reachability distance with K=2

In the figure above,for points enclosed in the circle,k-distance is taken into consideration and for the exterior points,the points not inside the cluster,the distance range of the point is valued for computing.

RD to the k-nearest neighbours of the data points need to be calculated to find the Local Reachability Density(LRD) of that point.

$$LRD_k(A) = \frac{1}{\sum_{Xj \in N_k(A)} \frac{\mathrm{RD}(A,Xj)}{\|N_k(A)\|}}$$

Above is the LRD equation.The inverse of the average reachability distance of data point A from its neighbors is called LRD.

The more the ratio of average reachability distance to that density of point found in a certain point as far as the LRD formula is used.

Values of LRD that are low indicate that the nearest cluster is farther from the point.

To compute LOF,the ratio of the average of the LRD of some number of k-neighbors of a point to that of the LRD of the same point.

In the cluster,if the density of the neighbors and data point is more compared to the density of the point,we can proudly then say the point is outlier.

Below is the LOF formula for appropriate outliers results;

$$\mathrm{LOF}_k(A) := \frac{\sum_{B \in N_k(A)} \frac{\mathrm{lrd}_k(B)}{\mathrm{lrd}_k(A)}}{|N_k(A)|} = \frac{\sum_{B \in N_k(A)} \mathrm{lrd}_k(B)}{|N_k(A)| \cdot \mathrm{lrd}_k(A)}$$

K-means clustering

This unsupervised machine learning algorithm is widespread and powerful.It aids in solving several critical unsupervised machine learning problems.

This algorithm monitors data points and groups those that are similar into clusters.

This method follows the following procedures to work;

- Predefine k values.
- Let the iteration of 1 to K continue until clusters no longer deviate.
- Calculate the centroid of the cluster.
- Choose the group and calculate the average.

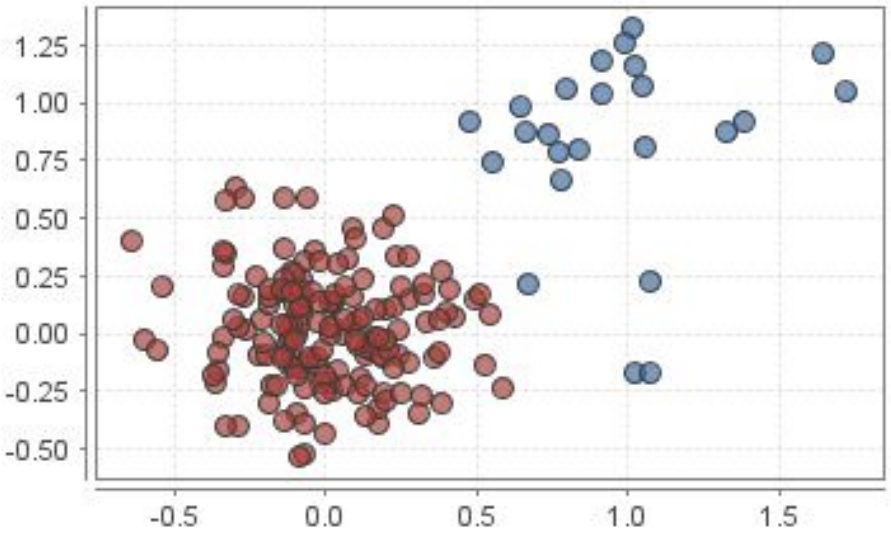


Figure showing K-means with k=2

Chapter 5: RapidMiner

In this chapter,we use Rapidminer tool for the implementation of our algorithms addressed in the previous part.RapidMiner contains various algorithms for anomaly detection.As a reminder,we apply unsupervised learning algorithms for anomaly detection.Python is the language in which these algorithms are built.

Here we will also have to look at the architecture of RapidMiner including the process for detection of anomalies.

Architecture of RapidMiner environment

RapidMiner is known as a commercial open-source software that provides a meshed environment that is used for data mining,machine learning,predictive analysis among others.
It is in this environment that we input our extracted dataset for anomaly detection.
The four major views available in RapidMiner are Design,Results,Turbo Prep,Auto Model views among others.
The following appear on clicking the view tabs mentioned above;

- Design view:Repository,Operators,Processes,Parameters,Help.
- Results:Result history
- Turbo Prep:Option to load dataset appears.
- Auto Model:Shows automatic procedures for anomaly detection.

RapidMiner has many data loading,modeling,processing and visualization methods that make it stand out as a choice for our objective.This prevents the issue of preprocessing the datasets and helps in visualization of the results.It's also has a friendly graphical user interface(GUI) for better modeling of complicated processes.

Terms in RapidMiner:

There are a few common terminologies used in RapidMiner.They include the following;

Operator:they are the building blocks for RapidMiner since it includes all the operators and extensions that we download.

Repository:it is where we upload data,running process and store data.

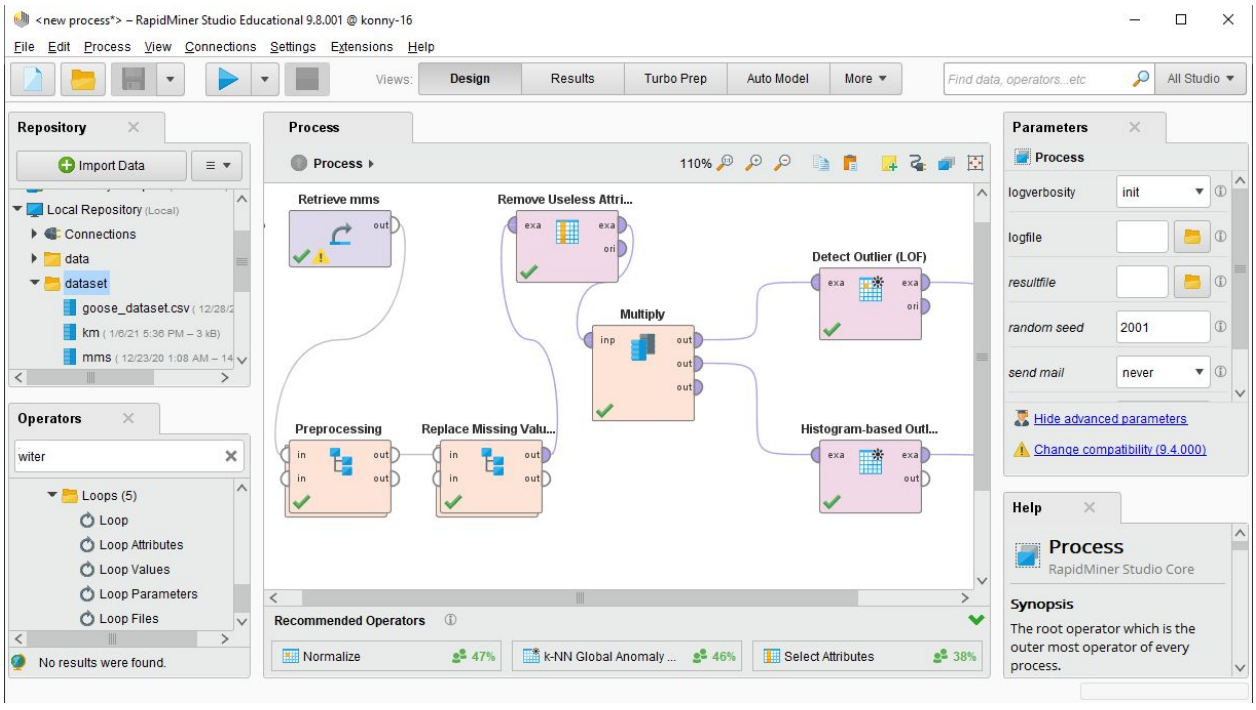
Special attribute:these are attributes used for identification.For example the id,label,cluster among others.

LOF Implementation

Here we will now have to detect anomalies from our dataset that we previously filtered and successfully extracted with the help of wireshark.

steps

- We open RapidMiner and then click on the design view tab.
- In the design view,we create a dataset and process repository.
- Import our CSV data into the repository we just created.
- We then start building our intrusion detection process.



Anomaly detection process in rapidMiner

As shown in the process design view above,we do the following;

- Read the mms csv file of the dataset filter from blaq_0 dataset.
- Preprocess the data
- Remove useless attributes
- Copy input objects in case there are various outputs connected with the help of multiply.
- Compute outlier score using Local Outlier Factor.

Attributes selected for MMS outlier detection

The attributes selected for the LOF algorithms are;

- Time
- Length
- Info3
- ConfirmedServiceRequest
- Destination
- Info1
- Outlier(new special attribute from LOF algorithm).

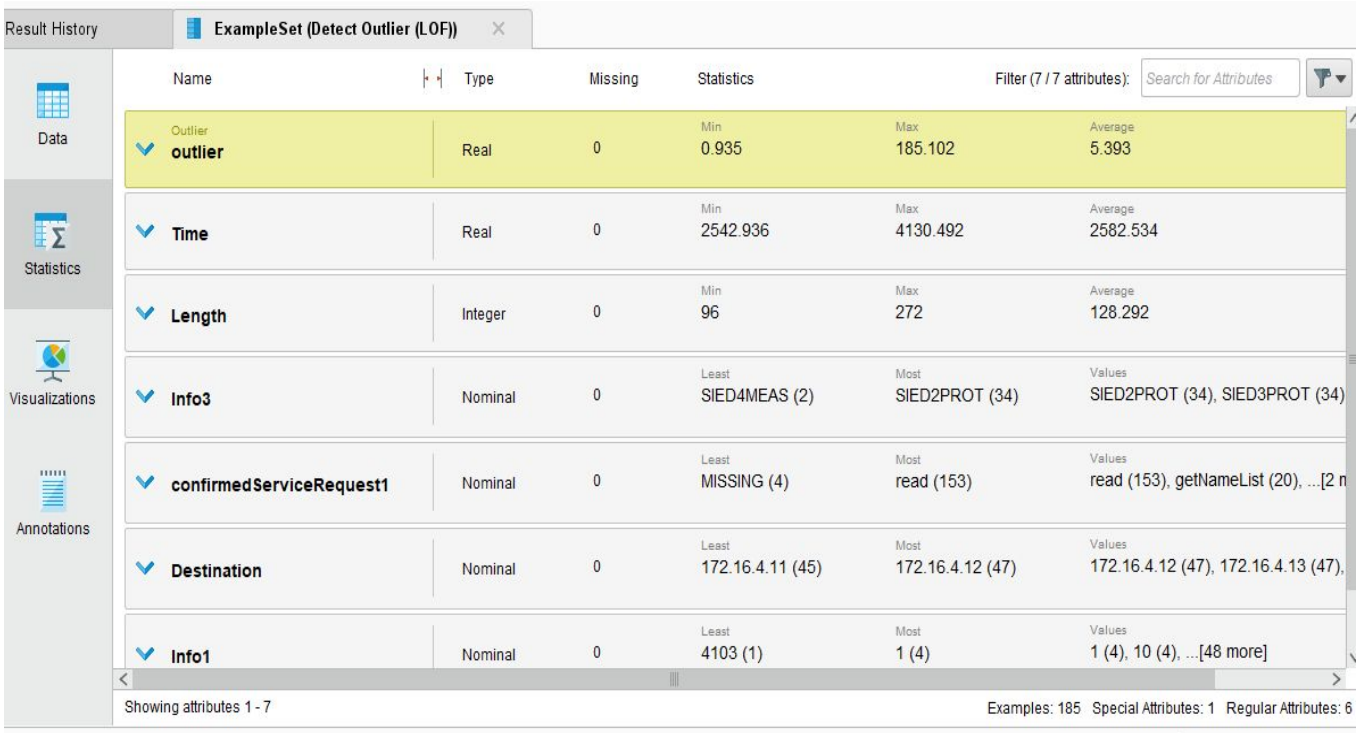
In the observation we can see that we left default settings for the parameters of Local Outlier Factor(LOF).

We calculate the outlier score based on the distance in the LOF algorithm.

The results of the LOF algorithm in the RapidMiner are as shown below.

Local Outlier Factors - Outlier Data

Row No.	outlier	confirm...	Destina...	Info1	Info3	Length	Time
1	1.974	read	172.16.4...	4103	SIED3P...	191	4130.4
2	1.972	read	172.16.4...	4079	SIED1P...	191	4130.4
3	1.972	read	172.16.4...	4082	SIED4P...	191	4130.4
4	1.462	read	172.16.4...	10	SIED4M...	134	2543.4
5	1.455	read	172.16.4...	10	SIED2M...	134	2543.1
6	1.455	read	172.16.4...	10	SIED3M...	134	2543.1
7	1.439	read	172.16.4...	34	SIED2DR	132	2553.1
8	1.439	read	172.16.4...	34	SIED3DR	132	2553.1



Screen of part LOF final dataset

K-means Clustering implementation

The goose packet is a very large dataset hence took a long time to run on the LOF algorithm without yielding any results.So we decided to apply K-means clustering for anomaly detection in the Automodel.

Steps

The main steps in clustering algorithm of RapidMiner include the following;

- Load a dataset into the process and perform a few preprocessing for which the model will work on.
- Normalize the dataset.
- Remove the unnecessary features in the dataset for better performance.
- Then create visualization for the model
- After running the process,and executing the results,export the process to our desired director.

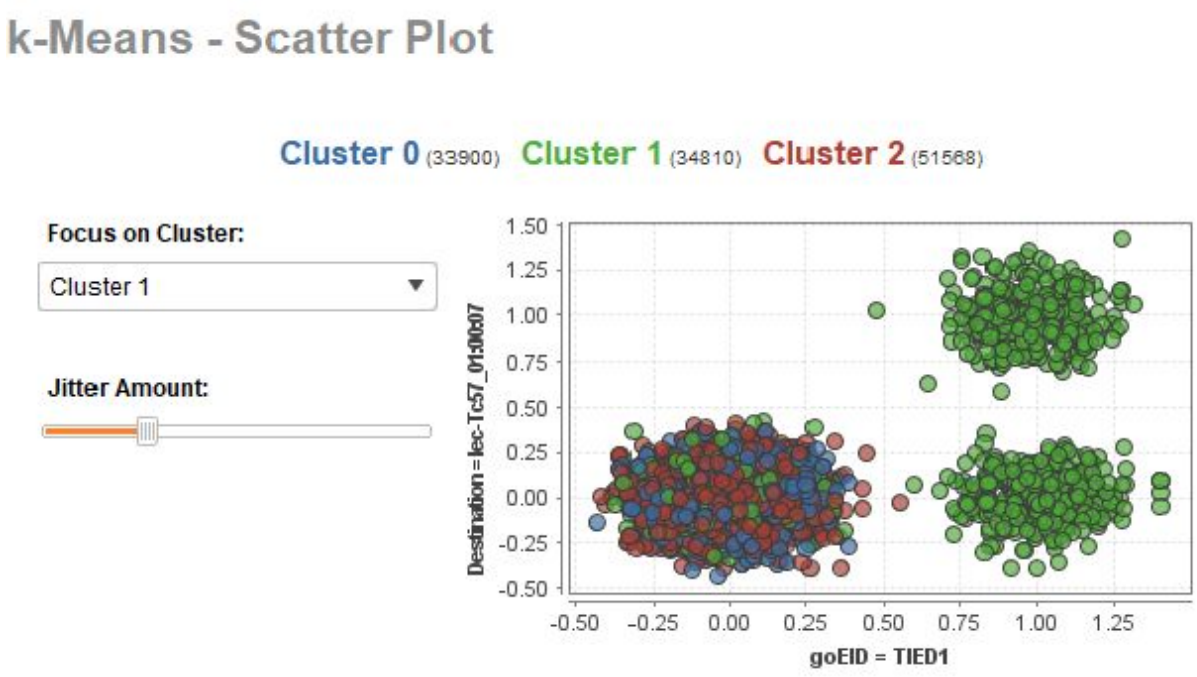


Figure showing the clusters

Observations

From the figure above,we can see that see we have chosen the value k=3.Our process 10 seconds to execute results successfully.

CONCLUSION

In conclusion,we filtered and extracted the necessary data protocol packets from the blaq_0 dataset with the help of wireshark.

The final datasets after merging was done according to the protocols were GOOSE and MMS csv file.

We then looked at the unsupervised methods of intrusion detection in unlabelled data of which we chose Local Outlier Factor and K-means clustering to solve our problem.

Anomaly detection was successfully performed by the usage of RapidMiner.In the LOF algorithm,outliers were detected by looking at their local density lower than that of their neighbours.To detect anomaly,we look at the LOF,if higher than 1 then there is an outlier.

In the K-means clustering algorithm,the major troubling task is choosing the value of k.

In cases where both local and global outliers are available,outliers under the radar might be passed and put into a cluster.

Graphs of the clustering algorithms help in the visualization of anomalies.

All in all,some outliers identified using the K-means clustering algorithm in actual sense might not be a malicious activity.

RESOURCES(KAYNAKLAR)

[1] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one, 11(4):e0152173, 2016.

[2] Richard Heady, George F Luger, Arthur Maccabe, and Mark Servilla. The architecture of a network level intrusion detection system. University of New Mexico. Department of Computer Science. College of Engineering, 1990.

[3] Adepu, S., Mathur, A.: An investigation into the response of a water treatment system to cyber attacks. In: Proceedings of the 17th IEEE High Assurance Systems Engineering Symposium, Orlando (January 2016)

[4] Adepu, S., Mathur, A.: Generalized attacker and attack models for Cyber-Physical Systems. In: Proceedings of the 40th Annual International Computers, Software & Applications Conference, Atlanta, USA. pp. 283–292. IEEE (June 2016)

[5] SP NIST. 800-94, guide to intrusion detection and prevention systems (idps). Information Technology Laboratory, National Institute of Standards and Technology, USA, 2007

[6] IEC-61850 Protocol Analysis and Online Intrusion Detection System for SCADA Networks using Machine Learning by Shivam Patel Bachelor of Engineering, Gujarat Technological University, 2014

[7] Nearest-Neighbor and Clustering based Anomaly Detection Algorithms for RapidMiner Mennatallah Amer1 and Markus Goldstein2

[8] Unsupervised anomaly detection and access control on network traffic Dimakogiannis, M.M. Award date: 2017

(CURRICULUM VITAE)ÖZGEÇMİŞ

Konbil William Chol DENG, 21.07.1994 de Pongborong,Güney Sudanda'da doğdu. İlk, orta ve lise eğitimini Uganda'da tamamladı. 2014 yılında St.Joseph's College Lisesinden mezun oldu. 2015 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nü kazandı.

2015 yılında başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği lisans eğitimi sürdürmektedir.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : Kritik altyapılara ait sezgisel anomali tespiti
ÖĞRENCİLER (B151210554/Konbil William Chol/DENG):

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			

Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN :Dr. Kevser Ovaz Akpınar
DANIŞMAN IMZASI: