Can Artificial Intelligence Secure your Infrastructure ‘?’

A. S. M. Shamim Reza

Deputy Manager, NOC
Link3 Technologies Limited

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
EUROPE

October 28 - 30, 2019
Lyon Convention Centre
Lyon, France

I propose to consider the question, "Can machines think?"

– Alan Turing 1950

[~]$whoami

– 10+ years, working for *Link3 Technologies Limited*
– InfoSec Professional

EC-Council Certified Security Analyst

sohag.shamim@gmail.com
shamimreza@link3.net

@asmshamimreza on **Linkedin**
@shamimrezasohag on **Twitter**

Agenda

The research work was basically motivated to detect Anomaly in DNS traffic, from NetFlow data, incorporating Machine Learning models.

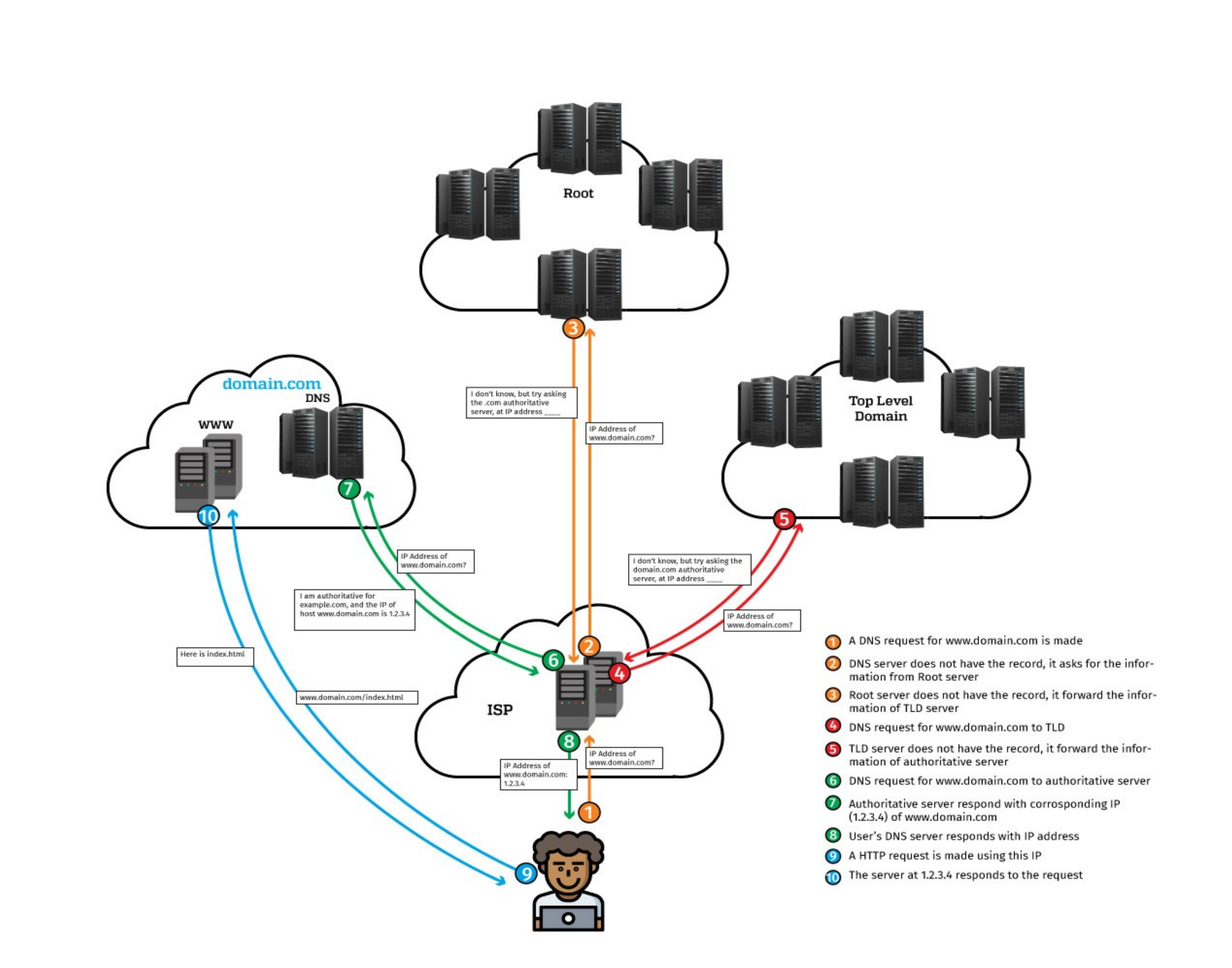
- What type of attacks that use DNS protocol or target DNS server?
- What are the characteristics of these attacks?
- How NetFlow conversation tells a story?
- Which Machine Learning model helps detect these attacks and how ?
- Which Machine Learning model can detect strange activities ?
- How the data has to be processed ?
- How the ML Model can be build ?

“it is better and more useful to meet a problem in time than to seek a remedy after the damage is done”

A Latin Proverb, found in 13th century

DNS	NetFlow	Anomaly
“The Domain Name Server (**DNS**) is the Achilles heel of the Web. The important thing is that it's managed responsibly.”	“It is network protocol developed by Cisco to collect and monitor network traffic flow data.”	“Anomaly detection, is the process of finding data objects with behaviors that are very different from expectation. Such objects are called **anomalies**.”
Tim Berners-Lee	**wikipedia**	***Data Mining. Concepts and Techniques* by Jiawei Han**

How DNS Works?



Attack Definition in-terms of NetFlow

Attacks that Target DNS Server

- Recursive Query Attacks
- Cache Poisoning Attacks
- Buffer overflow
- Port Scan

Attacks using DNS server –

- Reflection Attack
- DNS Tunneling

– It is very hard, to give an conclusive answer for a particular attack, if the amount of captured packet is low.

– For **DoS**, the amount of traffic can be high, but if the DNS package is vulnerable, only few packets can perform a successful attack.

– Increase amount of query request to a DNS server can be counted as, **Cache-Poisoning**

– **Buffer overflow**, packet size have to be large enough and it has to be manipulated

– But, a small amount of packet info can show that **port scan** is running

– **DNS tunneling**, could be the same as DoS, due to increase of traffic, larger packets, and using of BASE64 & BASE32 character encoding, generate the queries which is strange.

Attack Definition in-terms of NetFlow

Attacks that Target DNS Server

- Recursive Query Attacks
- Cache Poisoning Attacks
- Buffer overflow
- Port Scan

Attacks using DNS server –

- Reflection Attack
- DNS Tunneling

Amplification Attack	(proto udp and src port 53 and packets > 1000) or (proto udp and src port 53 and packets < 1001 and bytes > 900)
Scan Attack/Scanning	proto tcp and dst port 53 and packets => 1 and bytes == 58
Zero DNS Packet	proto udp/tcp and src/dst port 53 and packets => 1 and bytes <= 40
Acting Open-resovlers	proto udp/tcp and src port 53 and packets => 1 and bytes == 45
DNS Tunneling	proto udp/tcp and src/dst port 53 and packets => 1 and (151 => bytes <= 160)

The NetFlow Data Conversation

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2019-08-20 11:57:20.410	7.040	UDP	10.160.252.254:55427	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:20.430	7.040	UDP	10.160.252.254:55429	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:51601	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:56961	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-07-10 22:52:31.540	158.330	UDP	10.175.8.138:16664	->	192.168.0.254:53	......	0	111	7509	0	379	67	1
2019-07-10 22:52:26.570	199.400	UDP	10.175.8.78:1031	->	192.168.0.254:53	......	0	90	6123	0	245	68	1
2019-07-10 22:54:01.770	147.120	UDP	10.175.21.234:7623	->	192.168.0.254:53	......	0	104	7526	0	409	72	1
2019-07-10 22:54:25.950	133.830	UDP	10.175.24.160:41530	->	192.168.0.254:53	......	0	117	8018	0	479	68	1
2019-07-10 22:52:24.280	280.700	UDP	10.175.22.226:1029	->	192.168.0.254:53	......	0	117	8836	0	251	75	1

2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:62415	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:64165	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:63760	->	192.168.0.254:53	......	0	2	116	0	0	58	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:49521	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:61534	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50694	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50557	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-09-01 00:38:35.710	5.000	UDP	10.160.252.205:21343	->	192.168.0.254:53	......	0	4	1256	0	2009	314	1
2019-09-01 00:38:40.660	5.010	UDP	10.160.252.205:2500	->	192.168.0.254:53	......	0	4	976	0	1558	244	1
2019-09-01 00:38:45.710	7.960	UDP	10.160.252.205:29718	->	192.168.0.254:53	......	0	4	1256	0	1262	314	1
2019-09-01 00:38:50.670	6.440	UDP	10.160.252.205:5733	->	192.168.0.254:53	......	0	4	960	0	1192	240	1

The NetFlow Data Conversation

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2019-08-20 11:57:20.410	7.040	UDP	10.160.252.254:55427	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:20.430	7.040	UDP	10.160.252.254:55429	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:51601	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:56961	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-07-10 22:52:31.540	158.330	UDP	10.175.8.138:16664	->	192.168.0.254:53	......	0	111	7509	0	379	67	1
2019-07-10 22:52:26.570	199.400	UDP	10.175.8.78:1031	->	192.168.0.254:53	......	0	90	6123	0	245	68	1
2019-07-10 22:54:01.770	147.120	UDP	10.175.21.234:7623	->	192.168.0.254:53	......	0	104	7526	0	409	72	1
2019-07-10 22:54:25.950	133.830	UDP	10.175.24.160:41530	->	192.168.0.254:53	......	0	117	8018	0	479	68	1
2019-07-10 22:52:24.280	280.700	UDP	10.175.22.226:1029	->	192.168.0.254:53	......	0	117	8836	0	251	75	1
2019-09-08 19:06:41.210	0.000	UDP	10.160.252.188:61902	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-09-08 21:08:20.570	0.000	UDP	10.160.252.212:40960	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:62415	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:64165	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:63760	->	192.168.0.254:53	......	0	2	116	0	0	58	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:49521	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:61534	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50694	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50557	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-09-01 00:38:35.710	5.000	UDP	10.160.252.205:21343	->	192.168.0.254:53	......	0	4	1256	0	2009	314	1
2019-09-01 00:38:40.660	5.010	UDP	10.160.252.205:2500	->	192.168.0.254:53	......	0	4	976	0	1558	244	1
2019-09-01 00:38:45.710	7.960	UDP	10.160.252.205:29718	->	192.168.0.254:53	......	0	4	1256	0	1262	314	1
2019-09-01 00:38:50.670	6.440	UDP	10.160.252.205:5733	->	192.168.0.254:53	......	0	4	960	0	1192	240	1

The NetFlow Data Conversation

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2019-08-20 11:57:20.410	7.040	UDP	10.160.252.254:55427	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:20.430	7.040	UDP	10.160.252.254:55429	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:51601	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:56961	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-07-10 22:52:31.540	158.330	UDP	10.175.8.138:16664	->	192.168.0.254:53	......	0	111	7509	0	379	67	1
2019-07-10 22:52:26.570	199.400	UDP	10.175.8.78:1031	->	192.168.0.254:53	......	0	90	6123	0	245	68	1
2019-07-10 22:54:01.770	147.120	UDP	10.175.21.234:7623	->	192.168.0.254:53	......	0	104	7526	0	409	72	1
2019-07-10 22:54:25.950	133.830	UDP	10.175.24.160:41530	->	192.168.0.254:53	......	0	117	8018	0	479	68	1
2019-07-10 22:52:24.280	280.700	UDP	10.175.22.226:1029	->	192.168.0.254:53	......	0	117	8836	0	251	75	1
2019-09-08 19:06:41.210	0.000	UDP	10.160.252.188:61902	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-09-08 21:08:20.570	0.000	UDP	10.160.252.212:40960	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-08-26 21:44:18.720	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	28	0	0	28	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:62415	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:64165	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:63760	->	192.168.0.254:53	......	0	2	116	0	0	58	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:49521	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:61534	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50694	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50557	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-09-01 00:38:35.710	5.000	UDP	10.160.252.205:21343	->	192.168.0.254:53	......	0	4	1256	0	2009	314	1
2019-09-01 00:38:40.660	5.010	UDP	10.160.252.205:2500	->	192.168.0.254:53	......	0	4	976	0	1558	244	1
2019-09-01 00:38:45.710	7.960	UDP	10.160.252.205:29718	->	192.168.0.254:53	......	0	4	1256	0	1262	314	1
2019-09-01 00:38:50.670	6.440	UDP	10.160.252.205:5733	->	192.168.0.254:53	......	0	4	960	0	1192	240	1

The NetFlow Data Conversation

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2019-08-20 11:57:20.410	7.040	UDP	10.160.252.254:55427	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:20.430	7.040	UDP	10.160.252.254:55429	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:51601	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:56961	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-07-10 22:52:31.540	158.330	UDP	10.175.8.138:16664	->	192.168.0.254:53	......	0	111	7509	0	379	67	1
2019-07-10 22:52:26.570	199.400	UDP	10.175.8.78:1031	->	192.168.0.254:53	......	0	90	6123	0	245	68	1
2019-07-10 22:54:01.770	147.120	UDP	10.175.21.234:7623	->	192.168.0.254:53	......	0	104	7526	0	409	72	1
2019-07-10 22:54:25.950	133.830	UDP	10.175.24.160:41530	->	192.168.0.254:53	......	0	117	8018	0	479	68	1
2019-07-10 22:52:24.280	280.700	UDP	10.175.22.226:1029	->	192.168.0.254:53	......	0	117	8836	0	251	75	1
2019-09-08 19:06:41.210	0.000	UDP	10.160.252.188:61902	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-09-08 21:08:20.570	0.000	UDP	10.160.252.212:40960	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-08-26 21:44:18.720	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	28	0	0	28	1
2019-08-26 21:45:11.320	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	40	0	0	40	1
2019-08-26 21:47:33.480	0.000	TCP	88.6.232.5:42671	->	192.16.204.219:53	....S.	40	1	40	0	0	40	1
2019-08-26 22:18:58.290	0.000	TCP	88.6.232.5:52561	->	192.16.204.213:53	....S.	40	1	40	0	0	40	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:62415	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:64165	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:63760	->	192.168.0.254:53	......	0	2	116	0	0	58	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:49521	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:61534	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50694	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50557	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-09-01 00:38:35.710	5.000	UDP	10.160.252.205:21343	->	192.168.0.254:53	......	0	4	1256	0	2009	314	1
2019-09-01 00:38:40.660	5.010	UDP	10.160.252.205:2500	->	192.168.0.254:53	......	0	4	976	0	1558	244	1
2019-09-01 00:38:45.710	7.960	UDP	10.160.252.205:29718	->	192.168.0.254:53	......	0	4	1256	0	1262	314	1
2019-09-01 00:38:50.670	6.440	UDP	10.160.252.205:5733	->	192.168.0.254:53	......	0	4	960	0	1192	240	1

The NetFlow Data Conversation

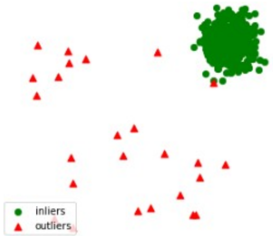
Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2019-08-20 11:57:20.410	7.040	UDP	10.160.252.254:55427	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:20.430	7.040	UDP	10.160.252.254:55429	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:51601	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:56961	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-07-10 22:52:31.540	158.330	UDP	10.175.8.138:16664	->	192.168.0.254:53	......	0	111	7509	0	379	67	1
2019-07-10 22:52:26.570	199.400	UDP	10.175.8.78:1031	->	192.168.0.254:53	......	0	90	6123	0	245	68	1
2019-07-10 22:54:01.770	147.120	UDP	10.175.21.234:7623	->	192.168.0.254:53	......	0	104	7526	0	409	72	1
2019-07-10 22:54:25.950	133.830	UDP	10.175.24.160:41530	->	192.168.0.254:53	......	0	117	8018	0	479	68	1
2019-07-10 22:52:24.280	280.700	UDP	10.175.22.226:1029	->	192.168.0.254:53	......	0	117	8836	0	251	75	1
2019-09-08 19:06:41.210	0.000	UDP	10.160.252.188:61902	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-09-08 21:08:20.570	0.000	UDP	10.160.252.212:40960	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-08-26 21:44:18.720	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	28	0	0	28	1
2019-08-26 21:45:11.320	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	40	0	0	40	1
2019-08-26 21:47:33.480	0.000	TCP	88.6.232.5:42671	->	192.16.204.219:53	....S.	40	1	40	0	0	40	1
2019-08-26 22:18:58.290	0.000	TCP	88.6.232.5:52561	->	192.16.204.213:53	....S.	40	1	40	0	0	40	1
2019-09-12 19:54:04.130	52.830	TCP	10.160.68.26:44628	->	192.168.0.254:53	.AP...	0	319	349200	6	52879	1094	1
2019-09-12 20:04:27.090	1426.890	TCP	10.160.68.26:44628	->	192.168.0.254:53	.AP...	0	245007	349.3 M	171	2.0 M	1425	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:62415	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:64165	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:63760	->	192.168.0.254:53	......	0	2	116	0	0	58	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:49521	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:61534	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50694	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50557	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-09-01 00:38:35.710	5.000	UDP	10.160.252.205:21343	->	192.168.0.254:53	......	0	4	1256	0	2009	314	1
2019-09-01 00:38:40.660	5.010	UDP	10.160.252.205:2500	->	192.168.0.254:53	......	0	4	976	0	1558	244	1
2019-09-01 00:38:45.710	7.960	UDP	10.160.252.205:29718	->	192.168.0.254:53	......	0	4	1256	0	1262	314	1
2019-09-01 00:38:50.670	6.440	UDP	10.160.252.205:5733	->	192.168.0.254:53	......	0	4	960	0	1192	240	1

The NetFlow Data Conversation

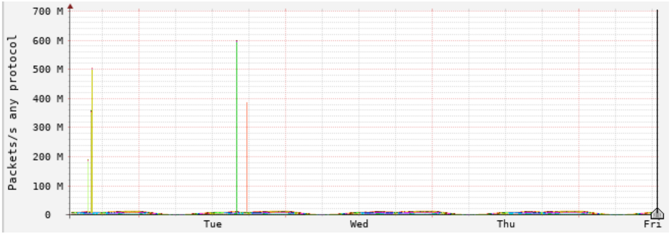
Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2019-08-20 11:57:20.410	7.040	UDP	10.160.252.254:55427	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:20.430	7.040	UDP	10.160.252.254:55429	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:51601	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:56961	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-07-10 22:52:31.540	158.330	UDP	10.175.8.138:16664	->	192.168.0.254:53	......	0	111	7509	0	379	67	1
2019-07-10 22:52:26.570	199.400	UDP	10.175.8.78:1031	->	192.168.0.254:53	......	0	90	6123	0	245	68	1
2019-07-10 22:54:01.770	147.120	UDP	10.175.21.234:7623	->	192.168.0.254:53	......	0	104	7526	0	409	72	1
2019-07-10 22:54:25.950	133.830	UDP	10.175.24.160:41530	->	192.168.0.254:53	......	0	117	8018	0	479	68	1
2019-07-10 22:52:24.280	280.700	UDP	10.175.22.226:1029	->	192.168.0.254:53	......	0	117	8836	0	251	75	1
2019-09-08 19:06:41.210	0.000	UDP	10.160.252.188:61902	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-09-08 21:08:20.570	0.000	UDP	10.160.252.212:40960	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-08-26 21:44:18.720	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	28	0	0	28	1
2019-08-26 21:45:11.320	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	40	0	0	40	1
2019-08-26 21:47:33.480	0.000	TCP	88.6.232.5:42671	->	192.16.204.219:53	....S.	40	1	40	0	0	40	1
2019-08-26 22:18:58.290	0.000	TCP	88.6.232.5:52561	->	192.16.204.213:53	....S.	40	1	40	0	0	40	1
2019-09-12 19:54:04.130	52.830	TCP	10.160.68.26:44628	->	192.168.0.254:53	.AP...	0	319	349200	6	52879	1094	1
2019-09-12 20:04:27.090	1426.890	TCP	10.160.68.26:44628	->	192.168.0.254:53	.AP...	0	245007	349.3 M	171	2.0 M	1425	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:62415	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:64165	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:63760	->	192.168.0.254:53	......	0	2	116	0	0	58	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:49521	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:61534	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50694	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50557	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-09-01 00:38:35.710	5.000	UDP	10.160.252.205:21343	->	192.168.0.254:53	......	0	4	1256	0	2009	314	1
2019-09-01 00:38:40.660	5.010	UDP	10.160.252.205:2500	->	192.168.0.254:53	......	0	4	976	0	1558	244	1
2019-09-01 00:38:45.710	7.960	UDP	10.160.252.205:29718	->	192.168.0.254:53	......	0	4	1256	0	1262	314	1
2019-09-01 00:38:50.670	6.440	UDP	10.160.252.205:5733	->	192.168.0.254:53	......	0	4	960	0	1192	240	1
2019-10-19 00:09:22.800	0.000	UDP	10.111.186.85:42531	->	192.168.0.254:53	......	192	1	160	0	0	160	1
2019-10-19 00:09:22.800	0.000	UDP	10.111.186.85:49651	->	192.168.0.254:53	......	192	1	160	0	0	160	1
2019-10-19 00:09:22.800	0.000	UDP	10.111.186.85:35340	->	192.168.0.254:53	......	192	1	160	0	0	160	1

Classification of Anomaly in Netflow

Point Anomaly – A data point that differ from the rest of the data set



Contextual Anomaly – If an observation is strange because of the context of the observation.



Collective Anomaly – A collection of data instances are strange in observation.

One should look for a possible alternative, and provide against it.
It is the first rule of criminal investigation.

Sherlock Holmes, the adventure of black peter

The Process for AI – standard procedure

Data Collection → Data Processing → The Algorithm → Training & Testing → Evaluation

– Data quality & quantity is very important

– we have collected data from 109 routers

– the volume is about 75TB

(*though we have worked only on few selected events*)

The Process for AI – standard procedure

Data Collection → **Data Processing** → **The Algorithm** → **Training & Testing** → **Evaluation**

– Data quality & quantity is very important

– we have collected data from 109 routers

– the volume is about 75TB

– manual processed the data

– randomized it

– labeled it accordingly

– used libraries to normalize & label

(*though we have worked only on few selected events*)

The Process for AI – standard procedure

Data Collection → **Data Processing** → **The Algorithm** → **Training & Testing** → **Evaluation**

– Data quality & quantity is very important

– we have collected data from 109 routers

– the volume is about 75TB

– manual processed the data

– randomized it

– labeled it accordingly

– used libraries to normalize & label

– as our work is to predict, kind of YES/NO, so we moved for Supervised learning and choose classification algorithm

(*though we have worked only on few selected events*)

The Process for AI – standard procedure

Data Collection → **Data Processing** → **The Algorithm** → **Training & Testing** → **Evaluation**

Data Collection

– Data quality & quantity is very important

– we have collected data from 109 routers

– the volume is about 75TB

Data Processing

– manual processed the data

– randomized it

– labeled it accordingly

– used libraries to normalize & label

The Algorithm

– as our work is to predict, kind of YES/NO, so we moved for Supervised learning and choose classification algorithm

Training & Testing

– started with small amount of data

– it has 5k of event

– split it for test & train

– again used 80K event sample for test & train

(*though we have worked only on few selected events*)

The Process for AI – standard procedure

Data Collection → **Data Processing** → **The Algorithm** → **Training & Testing** → **Evaluation**

Data Collection

– Data quality & quantity is very important

– we have collected data from 109 routers

– the volume is about 75TB

Data Processing

– manual processed the data

– randomized it

– labeled it accordingly

– used libraries to normalize & label

The Algorithm

– as our work is to predict, kind of YES/NO, so we moved for Supervised learning and choose classification algorithm

Training & Testing

– started with small amount of data

– it has 5k of event

– split it for test & train

– again used 80K event sample for test & train

Evaluation

– primarily worked with a single algorithm

– for anomaly check, tested two different algorithm

– tested with define sample % with different contamination ratio

(*though we have worked only on few selected events*)

The Process for AI – standard procedure, *“strategies to collect data”*

How is the data ?
– The data has multiple features – Too much noise is there

The Process for AI – standard procedure, *“strategies to collect data”*

How is the data ?
– The data has multiple features – Too much noise is there

So how we start working on the DATA ?
– Separated Open DNS Resolver to reduce data noise – Collected the data that are directed to Anycast DNS server – Collected the data that are directed to DNS server outside of link3 cloud – Started with “GO with the Book” formula

The Process for AI – standard procedure, *"raw data"*

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2019-08-20 11:57:20.410	7.040	UDP	10.160.252.254:55427	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:20.430	7.040	UDP	10.160.252.254:55429	->	192.168.0.254:53	......	0	4	280	0	318	70	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:51601	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-08-20 11:57:21.500	7.050	UDP	10.160.252.254:56961	->	192.168.0.254:53	......	0	4	276	0	313	69	1
2019-07-10 22:52:31.540	158.330	UDP	10.175.8.138:16664	->	192.168.0.254:53	......	0	111	7509	0	379	67	1
2019-07-10 22:52:26.570	199.400	UDP	10.175.8.78:1031	->	192.168.0.254:53	......	0	90	6123	0	245	68	1
2019-07-10 22:54:01.770	147.120	UDP	10.175.21.234:7623	->	192.168.0.254:53	......	0	104	7526	0	409	72	1
2019-07-10 22:54:25.950	133.830	UDP	10.175.24.160:41530	->	192.168.0.254:53	......	0	117	8018	0	479	68	1
2019-07-10 22:52:24.280	280.700	UDP	10.175.22.226:1029	->	192.168.0.254:53	......	0	117	8836	0	251	75	1
2019-09-08 19:06:41.210	0.000	UDP	10.160.252.188:61902	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-09-08 21:08:20.570	0.000	UDP	10.160.252.212:40960	->	192.168.0.254:53	......	0	1	45	0	0	45	1
2019-08-26 21:44:18.720	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	28	0	0	28	1
2019-08-26 21:45:11.320	0.000	UDP	142.93.132.42:55433	->	192.16.204.217:53	......	72	1	40	0	0	40	1
2019-08-26 21:47:33.480	0.000	TCP	88.6.232.5:42671	->	192.16.204.219:53	....S.	40	1	40	0	0	40	1
2019-08-26 22:18:58.290	0.000	TCP	88.6.232.5:52561	->	192.16.204.213:53	....S.	40	1	40	0	0	40	1
2019-09-12 19:54:04.130	52.830	TCP	10.160.68.26:44628	->	192.168.0.254:53	.AP...	0	319	349200	6	52879	1094	1
2019-09-12 20:04:27.090	1426.890	TCP	10.160.68.26:44628	->	192.168.0.254:53	.AP...	0	245007	349.3 M	171	2.0 M	1425	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:62415	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.650	0.000	UDP	10.160.252.254:64165	->	192.168.0.254:53	......	0	2	202	0	0	101	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:63760	->	192.168.0.254:53	......	0	2	116	0	0	58	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:49521	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.660	0.000	UDP	10.160.252.254:61534	->	192.168.0.254:53	......	0	2	198	0	0	99	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50694	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-08-20 12:07:59.670	0.000	UDP	10.160.252.254:50557	->	192.168.0.254:53	......	0	2	192	0	0	96	1
2019-09-01 00:38:35.710	5.000	UDP	10.160.252.205:21343	->	192.168.0.254:53	......	0	4	1256	0	2009	314	1
2019-09-01 00:38:40.660	5.010	UDP	10.160.252.205:2500	->	192.168.0.254:53	......	0	4	976	0	1558	244	1
2019-09-01 00:38:45.710	7.960	UDP	10.160.252.205:29718	->	192.168.0.254:53	......	0	4	1256	0	1262	314	1
2019-09-01 00:38:50.670	6.440	UDP	10.160.252.205:5733	->	192.168.0.254:53	......	0	4	960	0	1192	240	1
2019-10-19 00:09:22.800	0.000	UDP	10.111.186.85:42531	->	192.168.0.254:53	......	192	1	160	0	0	160	1
2019-10-19 00:09:22.800	0.000	UDP	10.111.186.85:49651	->	192.168.0.254:53	......	192	1	160	0	0	160	1
2019-10-19 00:09:22.800	0.000	UDP	10.111.186.85:35340	->	192.168.0.254:53	......	192	1	160	0	0	160	1

The Process for AI – standard procedure. *“formatted data”*

```
7.04,4,280,0,318,70,1,normalDNSFlow
7.04,4,280,0,318,70,1,normalDNSFlow
7.05,4,276,0,313,69,1,normalDNSFlow
7.05,4,276,0,313,69,1,normalDNSFlow
7.04,4,280,0,318,70,1,normalDNSFlow
7.04,4,280,0,318,70,1,normalDNSFlow
7.05,4,304,0,344,76,1,normalDNSFlow
7.05,4,304,0,344,76,1,normalDNSFlow
214.29,98,6475,0,241,66,1,dnsflood
226.31,173,10892,0,385,62,1,dnsflood
99.86,73,4560,0,365,62,1,dnsflood
63.67,129,8750,2,1099,67,1,dnsflood
222,18661,1.4M,84,50696,75,1,dnsflood
959.21,66129,5.8M,68,48762,88,1,dnsflood
437.5,84479,71.1M,193,1.3M,841,1,dnsflood
1712.01,99985,73.8M,58,344723,737,1,dnsflood
556.03,81319,81.7M,146,1.2M,1004,1,dnsflood
1799.38,250593,258.3M,139,1.1M,1030,1,dnsflood
481.86,90507,93.0M,187,1.5M,1027,1,dnsflood
```

The Process for AI – algorithm we choose to work with

Density Based
- DBSCAN
- LOF

Proximity

Distance Based
- K-NN
- K-MEANS

Parametric
- GMM
- One-Class SVMs

Probabilistic
- Angle-Based Outlier Detection (ABOD)
- FastABOD

– Proximity-based techniques define a data point as an anomaly when its locality is sparsely populated or very small in amount.

The Process for AI – algorithm we choose to work with

Density Based
- DBSCAN
- LOF

Distance Based
- **K-NN**
- K-MEANS

Proximity

Parametric
- GMM
- One-Class SVMs

Probabilistic
- Angle-Based Outlier Detection (ABOD)
- **FastABOD**

We choose to work on Supervised Learning

- In Supervised Learning, data has to be labeled
- Training data set has to be there

The Process for AI – algorithm we choose to work with, *kNN*

Density Based
- DBSCAN
- LOF

Distance Based
- **K-NN**
- K-MEANS

Parametric
- GMM
- One-Class SVMs

Probabilistic
- Angle-Based Outlier Detection (ABOD)
- FastABOD

For an observation, its distance to its kth nearest neighbor could be viewed as the outlying score. It could be viewed as a way to measure the density

```
method='largest',
algorithm='auto',
metric='minkowski',
p=2  (euclidean distance)
```

The Process for AI – algorithm we choose to work with, *ABOD*

Density Based
- DBSCAN
- LOF

Distance Based
- K-NN
- K-MEANS

Parametric
- GMM
- One-Class SVMs

Probabilistic
- Angle-Based Outlier Detection (ABOD)
- **FastABOD**

ABOD class for Angle-base Outlier Detection. For an observation, the variance of its weighted cosine scores to all neighbors could be viewed as the outlying score.

```
method='fast'
```

The Process for AI – GO with the Book; *“working with ML Model”*

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn import neighbors, preprocessing
from sklearn import metrics
```

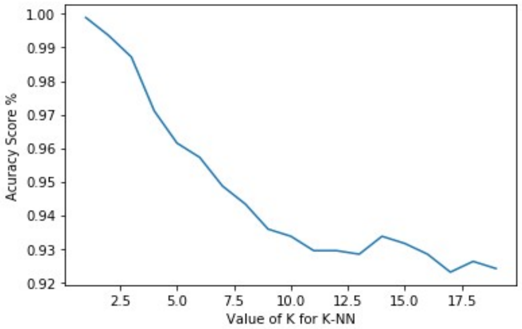
Data Normalization has already been done

Now Let's train & test the Model

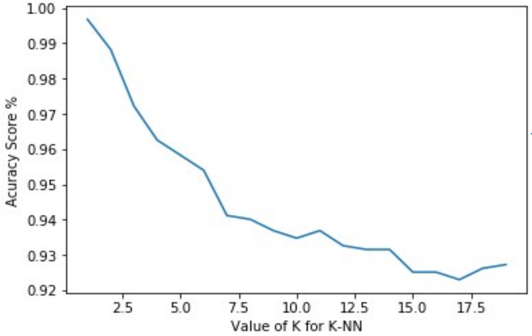
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(normalized_train_sample,train_labels,test_size=0.20)

from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 1)
classifier.fit(X_train,y_train)
pred=classifier.predict(normalized_test_sample)

scores=[]
scores.append(metrics.accuracy_score(test_labels, pred))
```

The Process for AI – GO with the Book; *“the test statistics”*

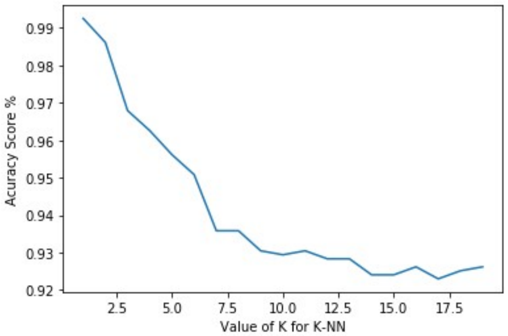
10%



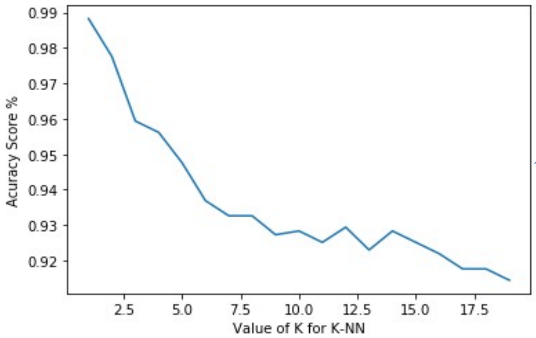
20%



30%



40%



The Process for AI – *"working with **PyOD**"*

"**PyOD** (*Python Outlier Detection*) is a comprehensive and scalable Python toolkit for detecting outlying objects in multivariate data."

– Zhao, Yue and Nasrullah, Zain and Li, Zheng

The Process for AI – *"working with **PyOD**"*

"**PyOD** (*Python Outlier Detection*) is a comprehensive and scalable Python toolkit for detecting outlying objects in multivariate data."

– Zhao, Yue and Nasrullah, Zain and Li, Zheng

** *For **Anomaly Detection,** data was not manually labeled, rather the raw data has been used.*

The Process for AI – *"working with **PyOD**"*

"**PyOD** (*Python Outlier Detection*) is a comprehensive and scalable Python toolkit for detecting outlying objects in multivariate data."

– Zhao, Yue and Nasrullah, Zain and Li, Zheng

** *For **Anomaly Detection,** data was not manually labeled, rather the raw data has been used.*

```
import matplotlib.pyplot as plt
import matplotlib.font_manager

from scipy import stats

from pyod.models.knn import KNN
from pyod.models.abod import ABOD
from pyod.utils.data import generate_data, get_outliers_inliers
```

Imported modules

The Process for AI – *“working with **PyOD**,* **kNN**”

Sample data – 10%
Contamination – 5%, 10%, 20%

```
On Training Data:
KNN ROC:0.9919, precision @ rank n:0.936

On Test Data:
KNN ROC:0.9998, precision @ rank n:0.96
```

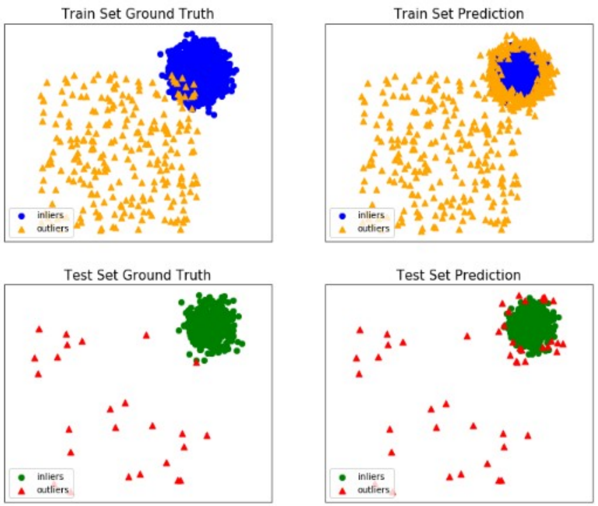
```
On Training Data:
KNN ROC:0.9906, precision @ rank n:0.932

On Test Data:
KNN ROC:0.9845, precision @ rank n:0.91
```

```
On Training Data:
KNN ROC:0.9869, precision @ rank n:0.937

On Test Data:
KNN ROC:0.9817, precision @ rank n:0.915
```

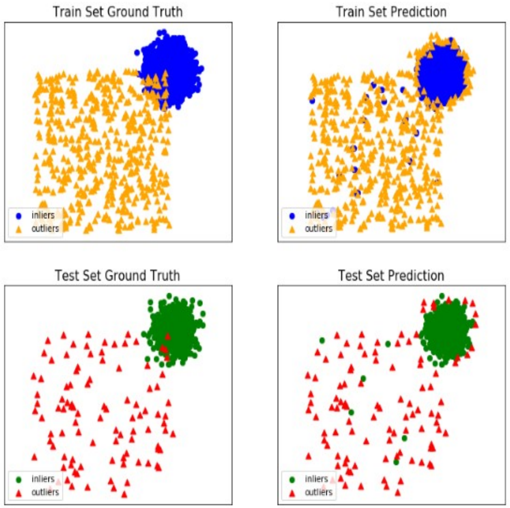
The Process for AI – *"working with **PyOD**,* **ABOD**"

Sample data – 10%
Contamination – 5%, 10%, 20%

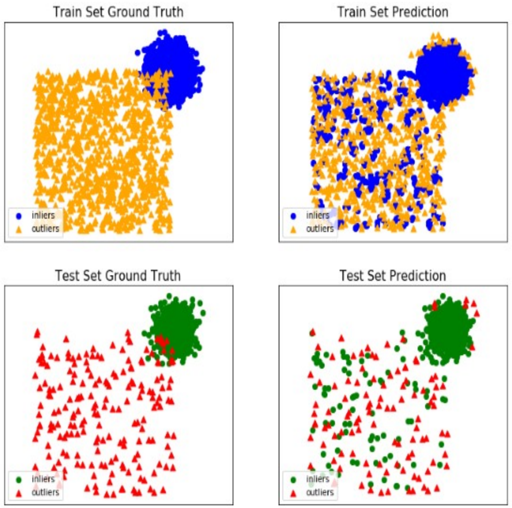
```
On Training Data:
ABOD ROC:0.9893, precision @ rank n:0.872

On Test Data:
ABOD ROC:0.9997, precision @ rank n:0.96
```




```
On Training Data:
ABOD ROC:0.987, precision @ rank n:0.872

On Test Data:
ABOD ROC:0.9826, precision @ rank n:0.89
```



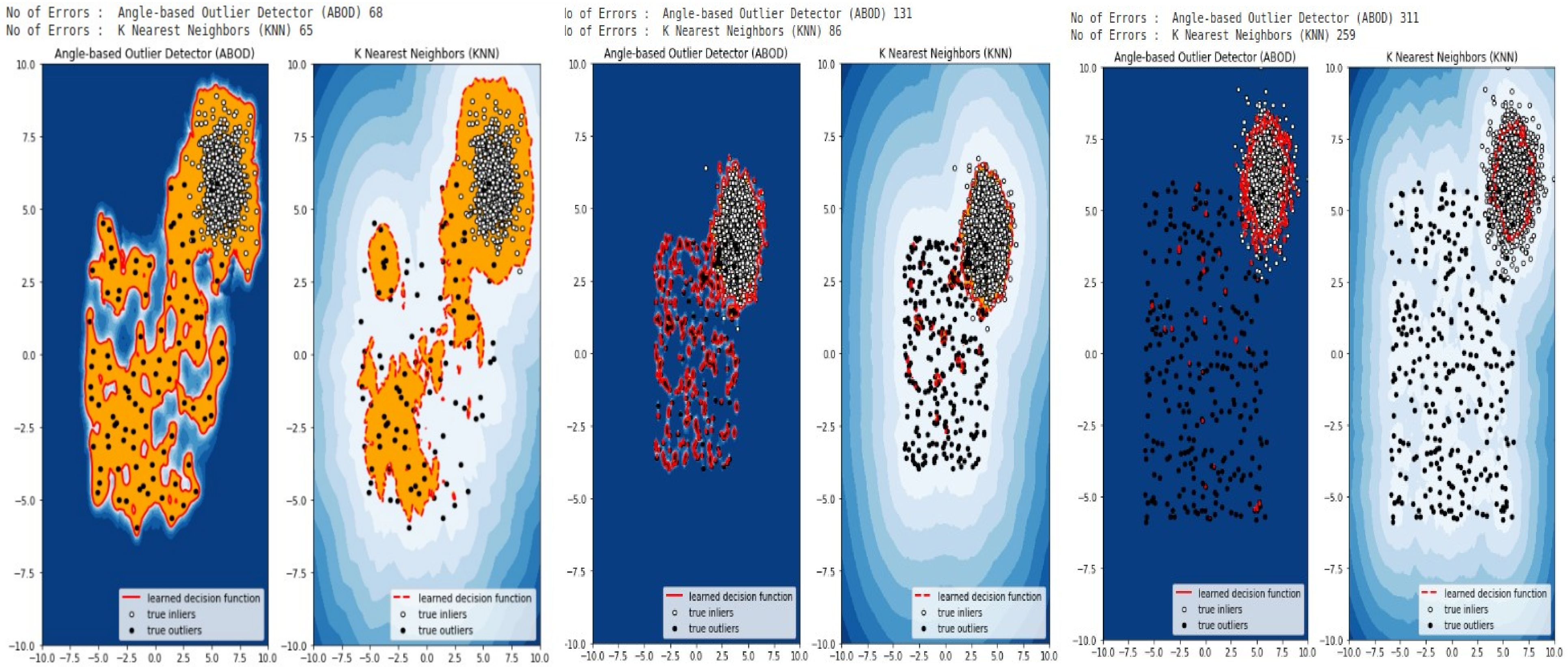

```
On Training Data:
ABOD ROC:0.9778, precision @ rank n:0.893

On Test Data:
ABOD ROC:0.9748, precision @ rank n:0.89
```



The Process for AI – *"working with **PyOD**"*

```
plt.figure(figsize=(10, 10))

for i, (clf_name,clf) in enumerate(classifiers.items()) :
    clf.fit(X_train)

    scores_pred = clf.decision_function(X_train)*-1
    y_pred = clf.predict(X_train)

    n_errors = (y_pred != Y_train).sum()
    print('No of Errors : ',clf_name, n_errors)

    threshold = stats.scoreatpercentile(scores_pred,100 *outlier_fraction)
    Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()]) * -1
    Z = Z.reshape(xx.shape)

    subplot = plt.subplot(1, 2, i + 1)
    subplot.contourf(xx, yy, Z, levels = np.linspace(Z.min(), threshold, 10),cmap=plt.cm.Blues_r)
    a = subplot.contour(xx, yy, Z, levels=[threshold],linewidths=2, colors='red')
    subplot.contourf(xx, yy, Z, levels=[threshold, Z.max()],colors='orange')
    b = subplot.scatter(X_train[:-n_outliers, 0], X_train[:-n_outliers, 1], c='white',s=20, edgecolor='k')
    c = subplot.scatter(X_train[-n_outliers:, 0], X_train[-n_outliers:, 1], c='black',s=20, edgecolor='k')
    subplot.axis('tight')

    subplot.legend(
        [a.collections[0], b, c],
        ['learned decision function', 'true inliers', 'true outliers'],
        prop=matplotlib.font_manager.FontProperties(size=10),
        loc='lower right')

    subplot.set_title(clf_name)
    subplot.set_xlim((-10, 10))
    subplot.set_ylim((-10, 10))
plt.show()
```

The Process for AI – *"working with **PyOD**,* **error rate"**



The Process for AI – "What are the lessons*?*"

– The difference between noise and anomalies
  – Noise often outnumber anomalies
– How to make use of label information/domain knowledge when available
– Fast runtime: can scale up to large datasets and high dimensional datasets
– Known behaviors under different data properties
– Can deal with different types of anomalies
– Its ability to deal with high dimensional problems

Reference

• Hoai-Vu Nguyen and Yongsun Choi, “Proactive Detection of DDoS Attacks Utilizing k-NN Classifier in an Anti-DDos Framework ” (World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering, Vol:4, No:3, 2010)

• Przemys ław Berezinski, Bartosz Jasiul, and Marcin Szpyrka; “ aw Berezinski, Bartosz Jasiul, and Marcin Szpyrka; “ An Entropy-Based Network Anomaly Detection Method”, Entropy 2015, 17, 2367-2408 (https://www.mdpi.com/journal/entropy)

• Das, S., Islam, M.R., Jayakodi, N.K. and Doppa, J.R., “Active Anomaly Detection via Ensembles: Insights, Algorithms, and Interpretability.” arXiv preprint arXiv:1901.08930. 2019.

• Ro, K., Zou, C., Wang, Z. and Yin, G., “Outlier detection for high-dimensional data. Biometrika” 2015.

• Suri, N.R. and Athithan, G., “Research Issues in Outlier Detection. In Outlier Detection: Techniques and Applications”, 2019.

• Milan Cermak, Pavel Celeda, Jan Vykopal, “Detection of DNS Traffic Anomalies in Large Networks”, Masaryk University, 2014 (https://www.researchgate.net/publication/290882059)

• Zdrnja, B., Brownlee, N., Wessels, D., “Passive Monitoring of DNS Anomalies. In:Detection of Intrusions and Malware, and Vulnerability Assessment”, 2007.

• Manasrah, A.M., Hasan, A., Abouabdalla, O.A., Ramadass, S., “Detecting Bot-net Activities Based on Abnormal DNS Traffic”, (IJCSIS) International Journal of Computer Science & Information Security, 2009.

• Zhao, Y., Nasrullah, Z. and Li, Z., 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. Journal of machine learning research (JMLR), 20(96), pp.1-7.

• Takashi Washio, Osaka University; IEEE International Conference on Data Mining, Singapore "Which Anomaly Detector should I use?", 2018

• LAKSHAY ARORA, NCU (THE NORTHCAP UNIVERSITY);Anomaly detection using PyOD, Analytics Vidhaya, 2019

Thank You

for your attention