BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Machine learning based Network Traffic Analysis

Desu Srinaveen
2017HT12532
F5Networks

Network Traffic Analysis

- Network Traffic
  - Passive
  - Active

- Network Traffic classification
  - Streaming traffic
  - Dynamic requests
  - Static requests
- Intrusions/Attacks
- Traffic analysis
  - Normal
  - Anamoly

F5 Networks

- Load Balancer
- Security Manager
- Policy manger
- Traffic Management System
- IP Intelligence

Projected Area of Concentration

- Intrusion : Attempting to break into or misuse your system.
- Intrusion Detection : Look for attack signatures, which are specific patterns that usually indicate malicious or suspicious intent.
- User Types: Intruders may be from outside the network or legitimate users of the network.
- Location Types: Intrusion can be a physical, system or remote intrusion.
- Anomaly based System : models the normal usage of the network as a noise characterization. Anything distinct from the noise is assumed to be an intrusion activity. This is the model that we will be implementing.
- Other systems are host based, network based , signature based etc

Building Prototype for analyzing network traffic

Step1: Data Capturing

- Data capture : The process of collecting data from network packets (pcap file).
- TCPDUMP : The command/tool is used to capture packets that is transferred or received over the network.
- WIRESHARK

Step 1 : Data Capture

The Captured Packets

```
root@blue-linux: ~
root@blue-linux:~#
root@blue-linux:~# tcpdump -i ens160
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), capture size 262144 bytes
20:19:52.200282 IP blue-linux.ssh > hyd-l-00047334.olympus.f5net.com.56633: Flag
s [P.], seq 306544290:306544498, ack 295643637, win 269, length 208
20:19:52.201156 IP6 blue-linux.55517 > ns1.pdsea.f5net.com.domain: 30238+ [1au]
PTR? 36.204.18.172.in-addr.arpa. (55)
20:19:52.202079 IP6 ns1.pdsea.f5net.com.domain > blue-linux.55517: 30238 1/0/1 P
TR hyd-l-00047334.olympus.f5net.com. (101)
20:19:52.202601 IP6 blue-linux.38455 > ns1.pdsea.f5net.com.domain: 46280+ [1au]
PTR? 145.74.145.10.in-addr.arpa. (55)
20:19:52.203337 IP6 ns1.pdsea.f5net.com.domain > blue-linux.38455: 46280 NXDomai
n* 0/1/1 (145)
20:19:52.203446 IP6 blue-linux.38455 > ns1.pdsea.f5net.com.domain: 46280+ PTR? 1
45.74.145.10.in-addr.arpa. (44)
20:19:52.204021 IP6 ns1.pdsea.f5net.com.domain > blue-linux.38455: 46280 NXDomai
n* 0/1/0 (134)
20:19:52.204699 IP blue-linux.ssh > hyd-l-00047334.olympus.f5net.com.56633: Flag
s [P.], seq 208:416, ack 1, win 269, length 208
20:19:52.209020 IP blue-linux.ssh > hyd-l-00047334.olympus.f5net.com.56633: Flag
s [P.], seq 416:1264, ack 1, win 269, length 848
20:19:52.209219 IP blue-linux.ssh > hyd-l-00047334.olympus.f5net.com.56633: Flag
s [P.], seq 1264:1456, ack 1, win 269, length 192
```

Step 2: Data Extraction

- Definition: Data extraction involves gathering the data used for model training and analysis by reducing the numbers of resources and eliminate redundant data from the raw data.

- X,Y variables : The extracted resources will act as input features for the machine learning models.

- SCAPY: Pythons Scapy module for reading pcap files and extracting the required features.

Step 2: Scapy code to extract features

```python
#!/usr/bin/python3
import os
import sys
import time
import csv
import scapy.all

resultFile = 'out.csv'

def usage():
	print('Usage: %s <pcap>' %(sys.argv[0]))

def main():
	if (len(sys.argv) < 2):
		usage()
		return(False)
	pcapFile = sys.argv[1]
	fd = open(resultFile, 'w')
	csvFd = csv.writer(fd)
	csvFd.writerow(['COUNT', 'DMAC', 'SMAC', 'DST-IP', 'SRC-IP', 'DPORT',\
		 'SPORT', 'PAYLOAD', 'IP-PAYLOAD','TIME'])
	packets = scapy.all.rdpcap(pcapFile)
	for (index, p) in enumerate(packets):
#		print('==========: packet [%d] :=============' %(index))
#		print(p.show())
		row = [index, p.dst, p.src]
		if (scapy.all.IP in p):
			row.extend([p[scapy.all.IP].dst, p[scapy.all.IP].src])
			ipPayloadLen = len(p[type(p[scapy.all.IP].payload)].payload)
		else:
			row.extend(['', ''])
			ipPayloadLen = 0
		if (scapy.all.TCP in p):
			row.extend([p[scapy.all.TCP].dport, p[scapy.all.TCP].sport])
		else:
			row.extend(['', ''])
		row.extend([len(p.payload), ipPayloadLen,p.time])
		csvFd.writerow(row)
	fd.close()
	return(True)
```

Step 2: Output generated in a csv format

	A	B	C	D	E	F	G	H	I	J
1	COUNT	DMAC	SMAC	DST-IP	SRC-IP	DPORT	SPORT	PAYLOAD	IP-PAY	TIME
2	0	00:00:00:(	00:00:00:00:00:00					82	0	1538631023
3	1	fa:16:3e:3	fa:16:3e:17:	10.0.0.23	10.0.0.1	20335	47302	60	0	1538631030
4	2	fa:16:3e:1	fa:16:3e:33:	10.0.0.1	10.0.0.23	47302	20335	60	0	1538631030
5	3	fa:16:3e:3	fa:16:3e:17:	10.0.0.23	10.0.0.1	20335	47302	52	0	1538631030
6	4	fa:16:3e:3	fa:16:3e:17:	10.0.0.23	10.0.0.1	20335	47302	82	30	1538631030
7	5	fa:16:3e:1	fa:16:3e:33:	10.0.0.3	10.0.0.2	20205	47302	60	0	1538631030

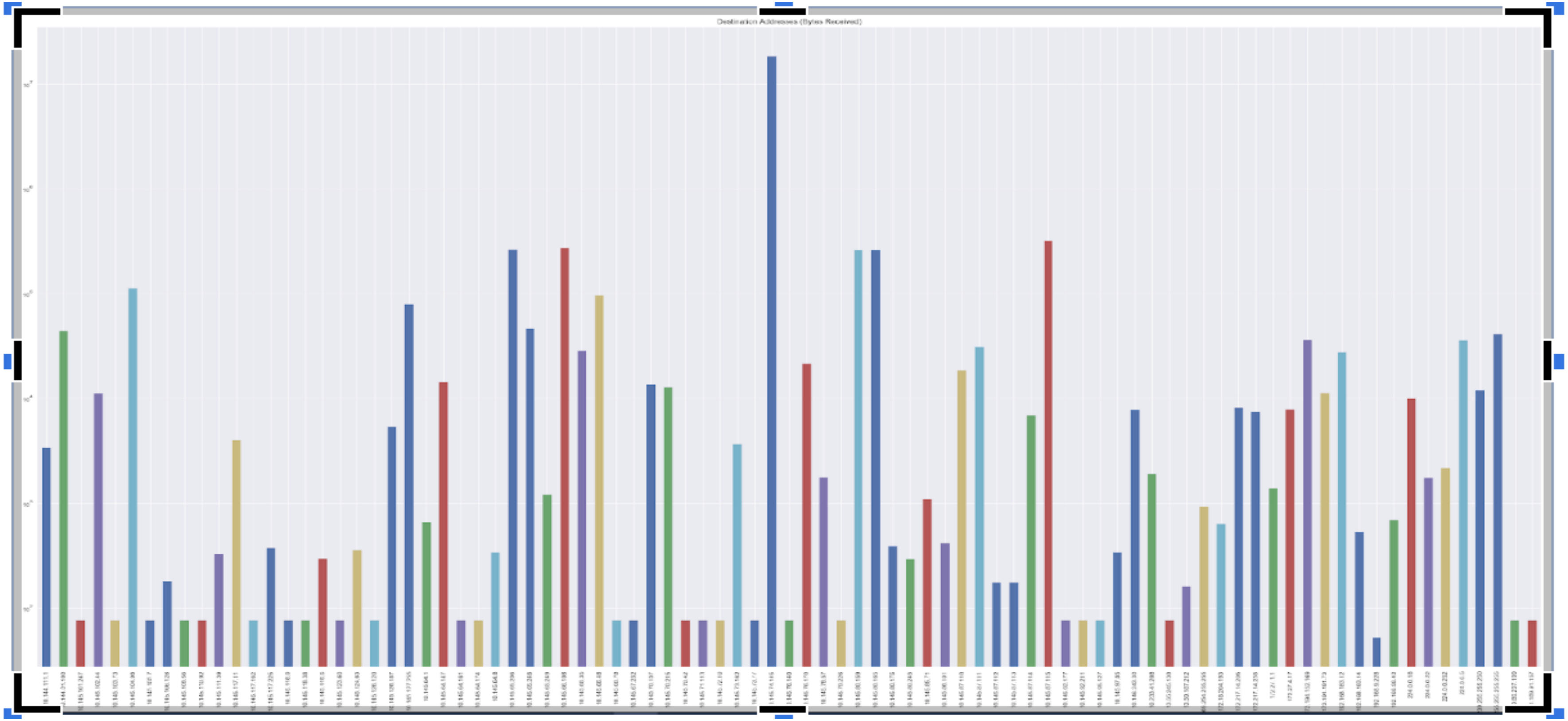
Step 3: Data Visualization

- Definition: Data visualization is used for representing the assembled data in the form of histogram or bar charts.

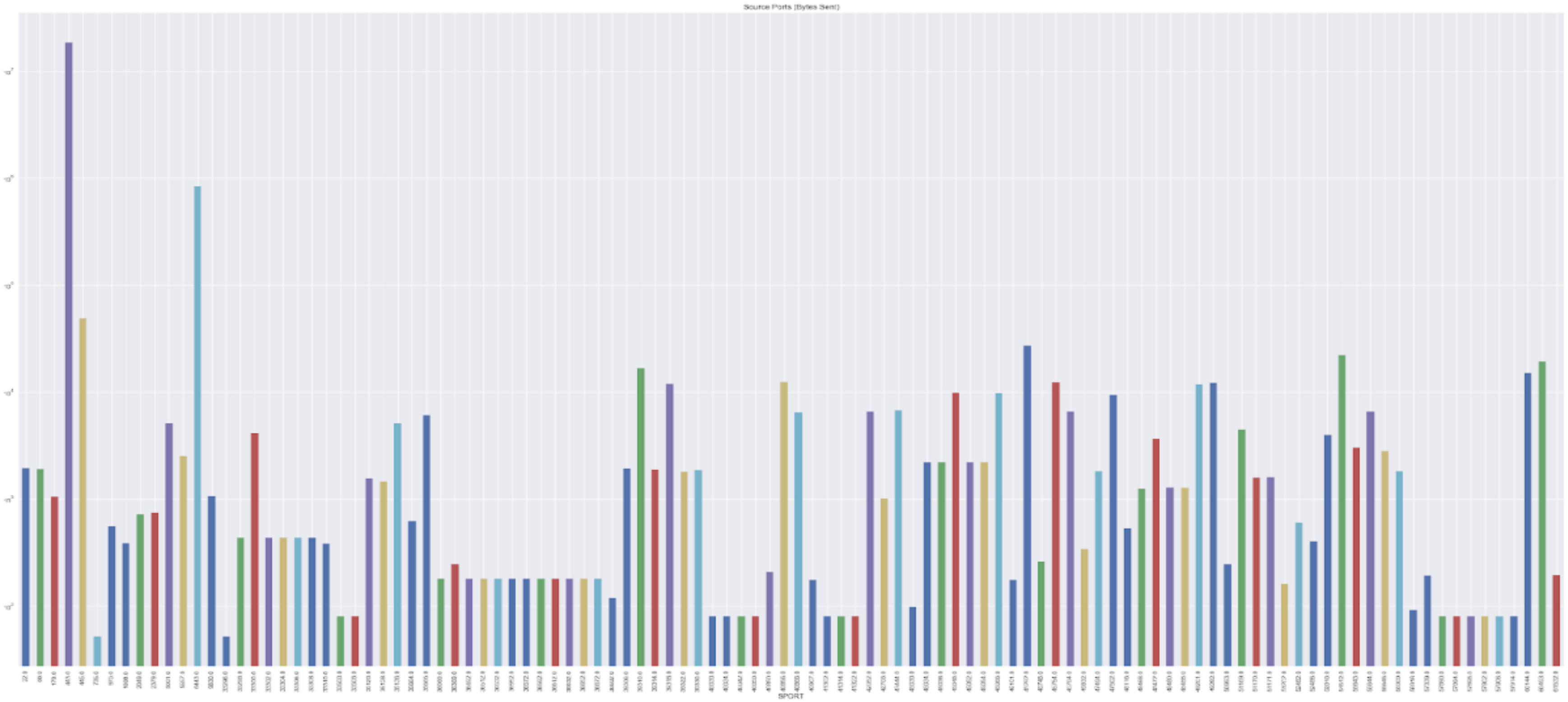
- Libraries: Pythons Pandas,matplotlib and Seaborn libraries to create a dataframe and visualize the data for underation the trends and patterns.

- Visual Treat of Analytics

Step 3: Bar plot depicts destination IP Vs bytes received



Step 3: Bar plot depicts source IP Vs bytes received



Step 4: Implementing network traffic analysis Models

- The network traffic types:
  - Time-based traffic
  - Host based traffic
- Host based traffic features
  - Content features
  - Traffic features

Step 4: Basic TCP features

Feature name	Description	Type
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of ``wrong" fragments	continuous
urgent	number of urgent packets	continuous

Step 4 : Content Features

	A	B	C
1	**Feature name**	**Description**	**Type**
2	hot	number of ``hot" indicators	continuous
3	num_failed_logins	number of failed login attempts	continuous
4	logged_in	1 if successfully logged in; 0 otherwise	discrete
5	num_compromised	number of ``compromised" conditions	continuous
6	root_shell	1 if root shell is obtained; 0 otherwise	discrete
7	su_attempted	1 if ``su root" command attempted; 0 otherwise	discrete
8	num_root	number of ``root" accesses	continuous
9	num_file_creations	number of file creation operations	continuous
10	num_shells	number of shell prompts	continuous
11	num_access_files	number of operations on access control files	continuous
12	num_outbound_cmds	number of outbound commands in an ftp session	continuous
13	is_hot_login	1 if the login belongs to the ``hot" list; 0 otherwise	discrete
14	is_guest_login	1 if the login is a ``guest"login; 0 otherwise	discrete

Step 4: Traffic Features

	A	B	C
1	**Feature name**	**Description**	**Type**
2	count	number of connections to the same host as the current connection in the past two seconds	continuous
3		*Note: The following features refer to these same-host connections.*	
4	serror_rate	% of connections that have ``SYN" errors	continuous
5	rerror_rate	% of connections that have ``REJ" errors	continuous
6	same_srv_rate	% of connections to the same service	continuous
7	diff_srv_rate	% of connections to different services	continuous
8	srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
9		*Note: The following features refer to these same-service connections.*	
10	srv_serror_rate	% of connections that have ``SYN" errors	continuous
11	srv_rerror_rate	% of connections that have ``REJ" errors	continuous
12	srv_diff_host_rate	% of connections to different hosts	continuous

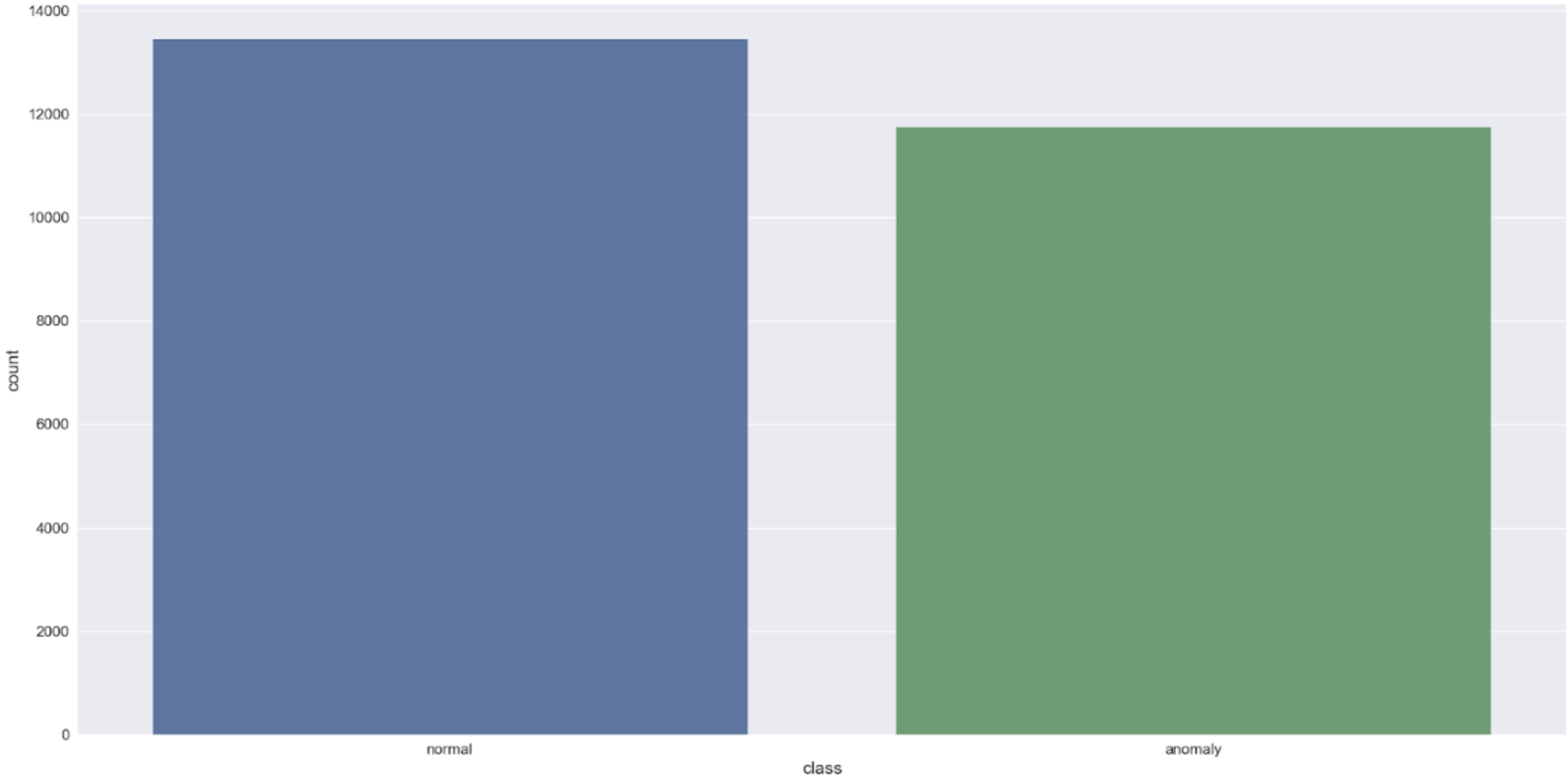
Step 5: Data Learning and Preprocessing

- Definition: Learning and preprocessing is to understand the data and remove the features that doesn't add value to the analysis/modeling.
- Describe(): Remove the columns with zero score
- Segregation of Categorical features
- Encoding features

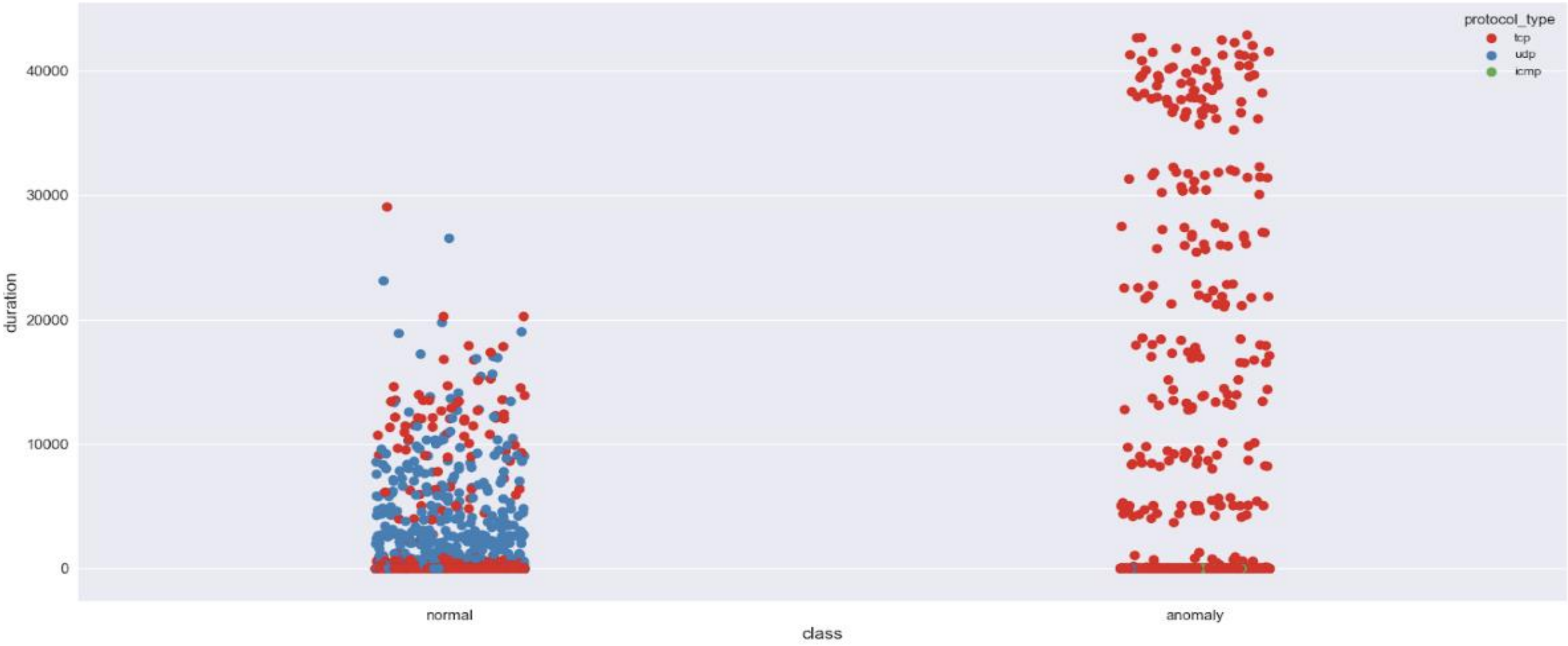
Step 5 : Data Preprocessing

	num_file_creations	num_shells	num_access_files	num_outbound_cmds
count	25192.000000	25192.000000	25192.000000	25192.0
mean	0.014727	0.000357	0.004327	0.0
std	0.529602	0.018898	0.098524	0.0
min	0.000000	0.000000	0.000000	0.0
25%	0.000000	0.000000	0.000000	0.0
50%	0.000000	0.000000	0.000000	0.0
75%	0.000000	0.000000	0.000000	0.0
max	40.000000	1.000000	8.000000	0.0

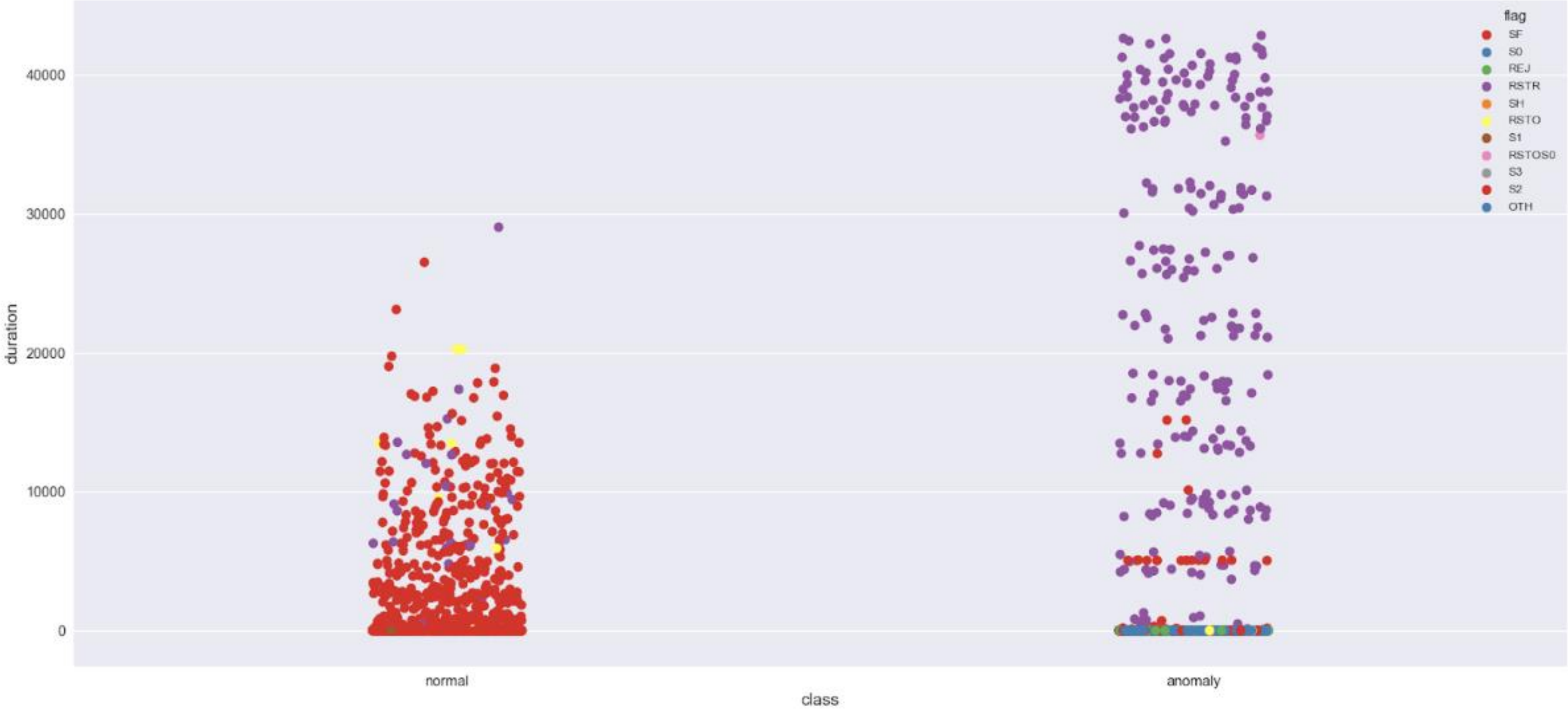
Step 5 : Visualization of normal and anomaly packets in our dataset



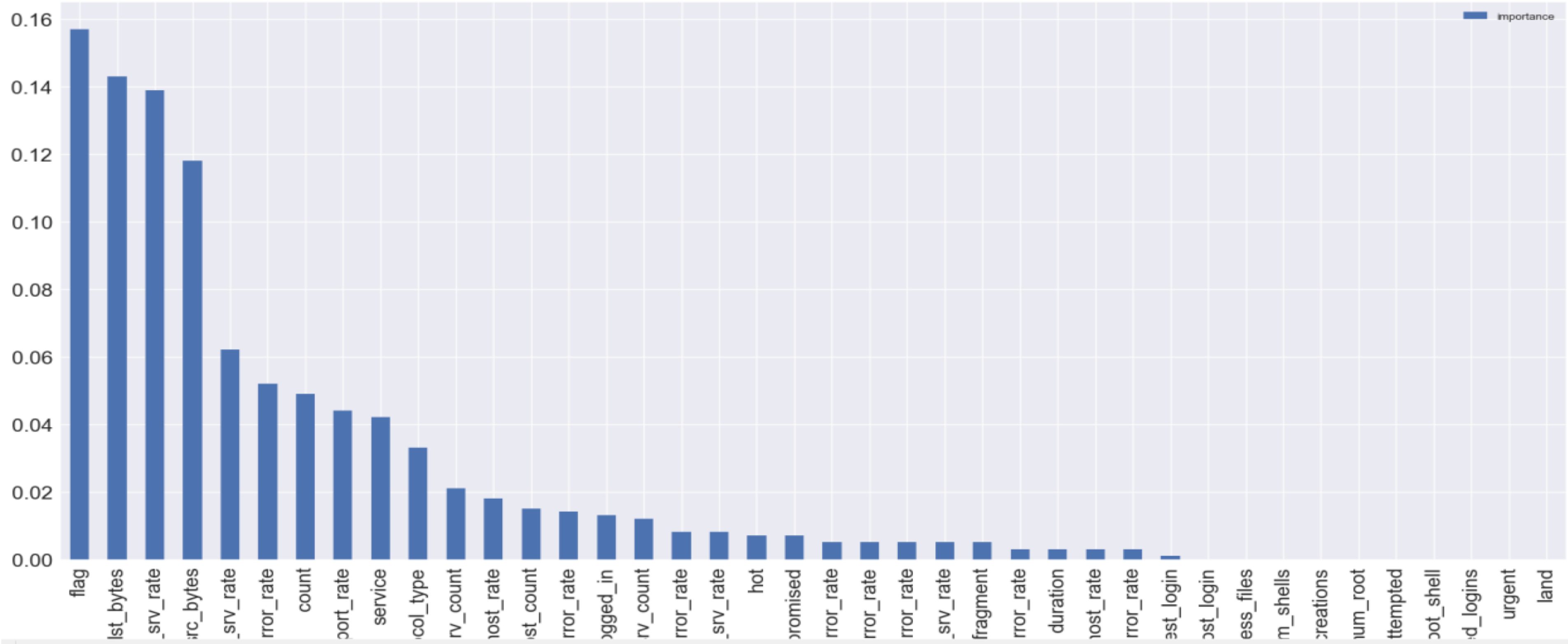
Step 5 : Class column visualized against protocol type



Step 5 : Class column visualized against flags in connection



Step 5: *Weights/importance of features*



Step 6: Data Fitting and modeling

- Random Forest Classifier
- Support Vector Classifier
- Extreme Gradient Decent Classifier
- K-Nearest Neighbors classifier
- Logistic Regression
- LGBM classifier
- Gaussian Naive Bayes Model Classifier
- Decision Tree Model Classifier

Step 7 : Model Evaluation

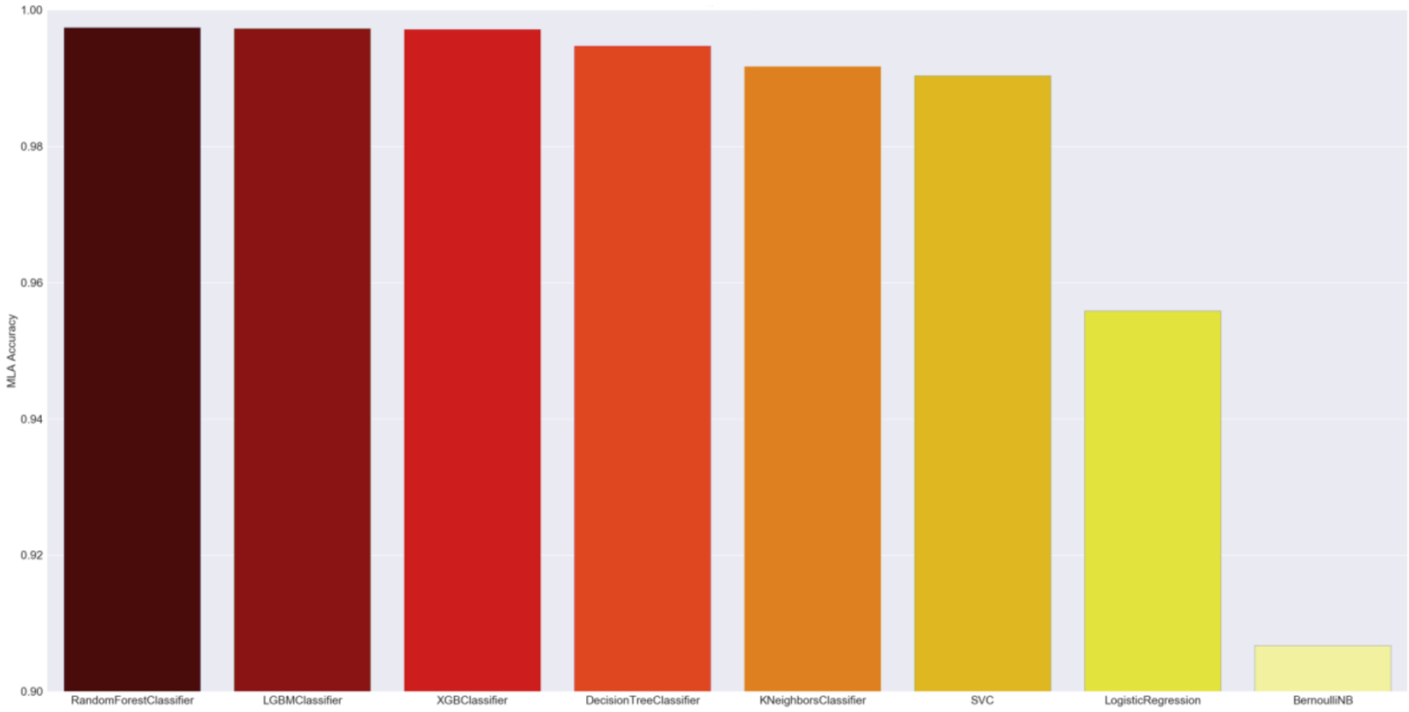
```
======== Logistic Regression Model Evaluation ==========

Cross Validation Mean Score:
 0.9538961919964779

Model Accuracy:
 0.954633095157083

Confusion matrix:
 [[7756  489]
 [ 311 9078]]

Classification report:
              precision    recall  f1-score   support

     anomaly       0.96      0.94      0.95      8245
      normal       0.95      0.97      0.96      9389

   micro avg       0.95      0.95      0.95     17634
   macro avg       0.96      0.95      0.95     17634
weighted avg       0.95      0.95      0.95     17634
```

Step 7 : Model Evaluation

```
============================== LGBM Classifier Model Evaluation

Cross Validation Mean Score:
 0.9976181792868687

Model Accuracy:
 1.0

Confusion matrix:
 [[8245    0]
 [   0 9389]]

Classification report:
               precision    recall  f1-score   support

     anomaly       1.00      1.00      1.00      8245
      normal       1.00      1.00      1.00      9389

   micro avg       1.00      1.00      1.00     17634
   macro avg       1.00      1.00      1.00     17634
weighted avg       1.00      1.00      1.00     17634
```

Step 8: Algorithms performance box comparison

Step 8: Algorithms performance bar comparison



Summary

- Classification
- Comparison of models
- Work environment usage
- Scalabilty
- Performance tuning

Thank You