FEATURE SELECTION AND CLASSIFICATION METHODS USING DEEP LEARNING IN VANET

LOAD DATASET

In [1]:
```
import pandas as pd
import sys
import numpy as np
cols=["srcip","sport","dstip","dsport","proto","state","dur","sbytes","dbytes","s
      "Dload","Spkts","Dpkts","swin","dwin","stcpb","dtcpb","smeansz","dmeansz","
      "Stime","Ltime","Sintpkt","Dintpkt","tcprtt","synack","ackdat","is_sm_ips_p
      "is_ftp_login","ct_ftp_cmd","ct_srv_src","ct_srv_dst","ct_dst_ltm","ct_src_
      "ct_dst_src_ltm","attack_cat","Label"]
# UNSW-NB15
unsw_train = pd.read_csv("E:/final_prj/dataset/UNSW-NB15/UNSW-NB15_1.csv",names=c
unsw_test = pd.read_csv("E:/final_prj/dataset/UNSW-NB15/set/UNSW_NB15_testing-set
```

In [2]:
```
unsw_train.head(3)
```

Out[2]:

	srcip	sport	dstip	dsport	proto	state	dur	sbytes	dbytes	sttl	...	ct_ftp_
0	59.166.0.0	1390	149.171.126.6	53	udp	CON	0.001055	132	164	31	...	
1	59.166.0.0	33661	149.171.126.9	1024	udp	CON	0.036133	528	304	31	...	
2	59.166.0.6	1464	149.171.126.7	53	udp	CON	0.001119	146	178	31	...	

3 rows × 49 columns

In [3]:
```
unsw_test.head(3)
```

Out[3]:

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_lt
0	1	0.121478	tcp	-	FIN	6	4	258	172	74.087490	...	
1	2	0.649902	tcp	-	FIN	14	38	734	42014	78.473372	...	
2	3	1.623129	tcp	-	FIN	8	16	364	13186	14.170161	...	

3 rows × 45 columns

Loading [MathJax]/extensions/Safe.js

```
In [4]: unsw_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700001 entries, 0 to 700000
Data columns (total 49 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   srcip             700001 non-null  object 
 1   sport             700001 non-null  object 
 2   dstip             700001 non-null  object 
 3   dsport            700001 non-null  object 
 4   proto             700001 non-null  object 
 5   state             700001 non-null  object 
 6   dur               700001 non-null  float64
 7   sbytes            700001 non-null  int64  
 8   dbytes            700001 non-null  int64  
 9   sttl              700001 non-null  int64  
 10  dttl              700001 non-null  int64  
 11  sloss             700001 non-null  int64  
 12  dloss             700001 non-null  int64  
 13  service           700001 non-null  object 
 14  Sload             700001 non-null  float64
 15  Dload             700001 non-null  float64
 16  Spkts             700001 non-null  int64  
 17  Dpkts             700001 non-null  int64  
 18  swin              700001 non-null  int64  
 19  dwin              700001 non-null  int64  
 20  stcpb             700001 non-null  int64  
 21  dtcpb             700001 non-null  int64  
 22  smeansz           700001 non-null  int64  
 23  dmeansz           700001 non-null  int64  
 24  trans_depth       700001 non-null  int64  
 25  res_bdy_len       700001 non-null  int64  
 26  Sjit              700001 non-null  float64
 27  Djit              700001 non-null  float64
 28  Stime             700001 non-null  int64  
 29  Ltime             700001 non-null  int64  
 30  Sintpkt           700001 non-null  float64
 31  Dintpkt           700001 non-null  float64
 32  tcprtt            700001 non-null  float64
 33  synack            700001 non-null  float64
 34  ackdat            700001 non-null  float64
 35  is_sm_ips_ports   700001 non-null  int64  
 36  ct_state_ttl      700001 non-null  int64  
 37  ct_flw_http_mthd  700001 non-null  int64  
 38  is_ftp_login      700001 non-null  int64  
 39  ct_ftp_cmd        700001 non-null  int64  
 40  ct_srv_src        700001 non-null  int64  
 41  ct_srv_dst        700001 non-null  int64  
 42  ct_dst_ltm        700001 non-null  int64  
 43  ct_src_ ltm       700001 non-null  int64  
 44  ct_src_dport_ltm  700001 non-null  int64  
 45  ct_dst_sport_ltm  700001 non-null  int64  
 46  ct_dst_src_ltm    700001 non-null  int64  
 47  attack_cat        22215 non-null   object 
 48  Label             700001 non-null  int64  
```

Loading [MathJax]/extensions/Safe.js

```
dtypes: float64(10), int64(31), object(8)
memory usage: 261.7+ MB
```

In [5]:
```
print("UNSW NB-15")
print(unsw_train.shape)
print(unsw_test.shape)
```

```
UNSW NB-15
(700001, 49)
(175341, 45)
```

Loading [MathJax]/extensions/Safe.js

In [6]:
```
unsw_train.isnull().sum()
```

Out[6]:
```
srcip                    0
sport                    0
dstip                    0
dsport                   0
proto                    0
state                    0
dur                      0
sbytes                   0
dbytes                   0
sttl                     0
dttl                     0
sloss                    0
dloss                    0
service                  0
Sload                    0
Dload                    0
Spkts                    0
Dpkts                    0
swin                     0
dwin                     0
stcpb                    0
dtcpb                    0
smeansz                  0
dmeansz                  0
trans_depth              0
res_bdy_len              0
Sjit                     0
Djit                     0
Stime                    0
Ltime                    0
Sintpkt                  0
Dintpkt                  0
tcprtt                   0
synack                   0
ackdat                   0
is_sm_ips_ports          0
ct_state_ttl             0
ct_flw_http_mthd         0
is_ftp_login             0
ct_ftp_cmd               0
ct_srv_src               0
ct_srv_dst               0
ct_dst_ltm               0
ct_src_ ltm              0
ct_src_dport_ltm         0
ct_dst_sport_ltm         0
ct_dst_src_ltm           0
attack_cat          677786
Label                    0
dtype: int64
```

Loading [MathJax]/extensions/Safe.js

In [7]:
```
unsw_train.attack_cat = unsw_train.attack_cat.fillna('Normal')
unsw_train.head()
```

Out[7]:

	srcip	sport	dstip	dsport	proto	state	dur	sbytes	dbytes	sttl	...	ct_ftp_
0	59.166.0.0	1390	149.171.126.6	53	udp	CON	0.001055	132	164	31	...	
1	59.166.0.0	33661	149.171.126.9	1024	udp	CON	0.036133	528	304	31	...	
2	59.166.0.6	1464	149.171.126.7	53	udp	CON	0.001119	146	178	31	...	
3	59.166.0.5	3593	149.171.126.5	53	udp	CON	0.001209	132	164	31	...	
4	59.166.0.3	49664	149.171.126.0	53	udp	CON	0.001169	146	178	31	...	

5 rows × 49 columns

In [8]:
```
unsw_test['attack_cat'].value_counts()
```

Out[8]:
```
Normal            56000
Generic           40000
Exploits          33393
Fuzzers           18184
DoS               12264
Reconnaissance    10491
Analysis           2000
Backdoor           1746
Shellcode          1133
Worms               130
Name: attack_cat, dtype: int64
```

Loading [MathJax]/extensions/Safe.js

In [9]:
```
# UNSW NB15 SET
# training set
print("training set for unsw")
for col_name in unsw_train.columns:
    if unsw_train[col_name].dtypes == 'object' :
        unique_cat = len(unsw_train[col_name].unique())
        print("Feature '{col_name}' has {unique_cat} categories".format(col_name=

print()
print('Distribution of categories in service:')
print(unsw_train['service'].value_counts().sort_values(ascending=False))

# testing set
print()
print("testing set for unsw")
for col_name in unsw_test.columns:
    if unsw_test[col_name].dtypes == 'object' :
        unique_cat = len(unsw_test[col_name].unique())
        print("Feature '{col_name}' has {unique_cat} categories".format(col_name=
```

```
training set for unsw
Feature 'srcip' has 40 categories
Feature 'sport' has 64541 categories
Feature 'dstip' has 44 categories
Feature 'dsport' has 62222 categories
Feature 'proto' has 135 categories
Feature 'state' has 16 categories
Feature 'service' has 13 categories
Feature 'attack_cat' has 10 categories

Distribution of categories in service:
-           430656
dns         121170
http         55858
ftp-data     37305
smtp         23588
ftp          16531
ssh          14636
pop3           206
ssl             20
snmp            14
radius           7
dhcp             7
irc              3
Name: service, dtype: int64

testing set for unsw
Feature 'proto' has 133 categories
Feature 'service' has 13 categories
Feature 'state' has 9 categories
Feature 'attack_cat' has 10 categories
```

ONE-HOT ENCODING

Loading [MathJax]/extensions/Safe.js

In [10]:
```
from sklearn.preprocessing import LabelEncoder,OneHotEncoder

# for unsw
categorical_columns=['proto', 'state', 'service','attack_cat']
unsw_train_cat_values = unsw_train[categorical_columns]
unsw_test_cat_values =unsw_test[categorical_columns]
print(unsw_train_cat_values.head(2))
print(unsw_test_cat_values.head(2))
```

```
  proto state service attack_cat
0   udp   CON     dns     Normal
1   udp   CON       -     Normal
  proto state service attack_cat
0   tcp   FIN       -     Normal
1   tcp   FIN       -     Normal
```

In [11]:
```
# changing categorical values to numeric for unsw
# train set
unsw_train_values_enc=unsw_train_cat_values.apply(LabelEncoder().fit_transform)
print(unsw_train_cat_values.head(5))
print('--------------------')
print(unsw_train_values_enc.head(3))

# test set
print()
unsw_test_values_enc=unsw_test_cat_values.apply(LabelEncoder().fit_transform)
print(unsw_test_values_enc.head(3))
```

```
  proto state service attack_cat
0   udp   CON     dns     Normal
1   udp   CON       -     Normal
2   udp   CON     dns     Normal
3   udp   CON     dns     Normal
4   udp   CON     dns     Normal
--------------------
   proto  state  service  attack_cat
0    120      2        2           6
1    120      2        0           6
2    120      2        2           6

   proto  state  service  attack_cat
0    113      2        0           6
1    113      2        0           6
2    113      2        0           6
```

Loading [MathJax]/extensions/Safe.js

In [12]:
```
#trianing set for unsw
protocol=sorted(unsw_train.proto.unique())
unique_proto=[x for x in protocol]
print(unique_proto)

service=sorted(unsw_train.service.unique())
unique_service=[ x for x in service]
print(unique_service)

state=sorted(unsw_train.state.unique())
unique_state=[ x for x in state]
print(unique_state)

attack=sorted(unsw_train.attack_cat.unique())
unique_attack = [ x for x in attack]
print(unique_attack)

unswtraincols=unique_proto + unique_service + unique_state+unique_attack
len(unswtraincols)
```

```
['3pc', 'a/n', 'aes-sp3-d', 'any', 'argus', 'aris', 'arp', 'ax.25', 'bbn-rcc',
'bna', 'br-sat-mon', 'cbt', 'cftp', 'chaos', 'compaq-peer', 'cphb', 'cpnx', 'cr
tp', 'crudp', 'dcn', 'ddp', 'ddx', 'dgp', 'egp', 'eigrp', 'emcon', 'encap', 'es
p', 'etherip', 'fc', 'fire', 'ggp', 'gmtp', 'gre', 'hmp', 'i-nlsp', 'iatp', 'i
b', 'icmp', 'idpr', 'idpr-cmtp', 'idrp', 'ifmp', 'igmp', 'igp', 'il', 'ip', 'ip
comp', 'ipcv', 'ipip', 'iplt', 'ipnip', 'ippc', 'ipv6', 'ipv6-frag', 'ipv6-no',
'ipv6-opts', 'ipv6-route', 'ipx-n-ip', 'irtp', 'isis', 'iso-ip', 'iso-tp4', 'kr
yptolan', 'l2tp', 'larp', 'leaf-1', 'leaf-2', 'merit-inp', 'mfe-nsp', 'mhrp',
'micp', 'mobile', 'mtp', 'mux', 'narp', 'netblt', 'nsfnet-igp', 'nvp', 'ospf',
'pgm', 'pim', 'pipe', 'pnni', 'pri-enc', 'prm', 'ptp', 'pup', 'pvp', 'qnx', 'rd
p', 'rsvp', 'rtp', 'rvd', 'sat-expak', 'sat-mon', 'sccopmce', 'scps', 'sctp',
'sdrp', 'secure-vmtp', 'sep', 'skip', 'sm', 'smp', 'snp', 'sprite-rpc', 'sps',
'srp', 'st2', 'stp', 'sun-nd', 'swipe', 'tcf', 'tcp', 'tlsp', 'tp++', 'trunk-
1', 'trunk-2', 'ttp', 'udp', 'udt', 'unas', 'uti', 'vines', 'visa', 'vmtp', 'vr
rp', 'wb-expak', 'wb-mon', 'wsn', 'xnet', 'xns-idp', 'xtp', 'zero']
['-', 'dhcp', 'dns', 'ftp', 'ftp-data', 'http', 'irc', 'pop3', 'radius', 'smt
p', 'snmp', 'ssh', 'ssl']
['ACC', 'CLO', 'CON', 'ECO', 'ECR', 'FIN', 'INT', 'MAS', 'PAR', 'REQ', 'RST',
'TST', 'TXD', 'URH', 'URN', 'no']
[' Fuzzers', 'Analysis', 'Backdoors', 'DoS', 'Exploits', 'Generic', 'Normal',
'Reconnaissance', 'Shellcode', 'Worms']
```

Out[12]:
```
174
```

Loading [MathJax]/extensions/Safe.js

In [13]:
```
# test set for unsw
t_protocol=sorted(unsw_test.proto.unique())
t_unique_proto=[x for x in t_protocol]
print(t_unique_proto)

t_service=sorted(unsw_test.service.unique())
t_unique_service=[ x for x in t_service]
print(t_unique_service)

t_state=sorted(unsw_test.state.unique())
t_unique_state=[ x for x in t_state]
print(t_unique_state)

t_attack=sorted(unsw_test.attack_cat.unique())
t_unique_attack = [ x for x in t_attack]
print(t_unique_attack)

unswtestcols=t_unique_proto + t_unique_service + t_unique_state+t_unique_attack
len(unswtestcols)
```

```
['3pc', 'a/n', 'aes-sp3-d', 'any', 'argus', 'aris', 'arp', 'ax.25', 'bbn-rcc',
'bna', 'br-sat-mon', 'cbt', 'cftp', 'chaos', 'compaq-peer', 'cphb', 'cpnx', 'cr
tp', 'crudp', 'dcn', 'ddp', 'ddx', 'dgp', 'egp', 'eigrp', 'emcon', 'encap', 'et
herip', 'fc', 'fire', 'ggp', 'gmtp', 'gre', 'hmp', 'i-nlsp', 'iatp', 'ib', 'icm
p', 'idpr', 'idpr-cmtp', 'idrp', 'ifmp', 'igmp', 'igp', 'il', 'ip', 'ipcomp',
'ipcv', 'ipip', 'iplt', 'ipnip', 'ippc', 'ipv6', 'ipv6-frag', 'ipv6-no', 'ipv6-
opts', 'ipv6-route', 'ipx-n-ip', 'irtp', 'isis', 'iso-ip', 'iso-tp4', 'kryptola
n', 'l2tp', 'larp', 'leaf-1', 'leaf-2', 'merit-inp', 'mfe-nsp', 'mhrp', 'micp',
'mobile', 'mtp', 'mux', 'narp', 'netblt', 'nsfnet-igp', 'nvp', 'ospf', 'pgm',
'pim', 'pipe', 'pnni', 'pri-enc', 'prm', 'ptp', 'pup', 'pvp', 'qnx', 'rdp', 'rs
vp', 'rtp', 'rvd', 'sat-expak', 'sat-mon', 'sccopmce', 'scps', 'sctp', 'sdrp',
'secure-vmtp', 'sep', 'skip', 'sm', 'smp', 'snp', 'sprite-rpc', 'sps', 'srp',
'st2', 'stp', 'sun-nd', 'swipe', 'tcf', 'tcp', 'tlsp', 'tp++', 'trunk-1', 'trun
k-2', 'ttp', 'udp', 'unas', 'uti', 'vines', 'visa', 'vmtp', 'vrrp', 'wb-expak',
'wb-mon', 'wsn', 'xnet', 'xns-idp', 'xtp', 'zero']
['-', 'dhcp', 'dns', 'ftp', 'ftp-data', 'http', 'irc', 'pop3', 'radius', 'smt
p', 'snmp', 'ssh', 'ssl']
['CON', 'ECO', 'FIN', 'INT', 'PAR', 'REQ', 'RST', 'URN', 'no']
['Analysis', 'Backdoor', 'DoS', 'Exploits', 'Fuzzers', 'Generic', 'Normal', 'Re
connaissance', 'Shellcode', 'Worms']
```

Out[13]:
```
165
```

Loading [MathJax]/extensions/Safe.js

In [14]:
```
## one hot encoding
# train for unsw
enc = OneHotEncoder()
unsw_train_values_hotenc = enc.fit_transform(unsw_train_values_enc)
unsw_train_cat_data = pd.DataFrame(unsw_train_values_hotenc.toarray(),columns=uns
unsw_train_cat_data.head(2)
```

Out[14]:

	3pc	a/n	aes-sp3-d	any	argus	aris	arp	ax.25	bbn-rcc	bna	...	Fuzzers	Analysis	Backdoors	Do
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.

2 rows × 174 columns

In [15]:
```
#  one hot encoding
# test for unsw
unsw_test_values_hotenc = enc.fit_transform(unsw_test_values_enc)
unsw_test_cat_data = pd.DataFrame(unsw_test_values_hotenc.toarray(),columns=unswt
unsw_test_cat_data.head()
```

Out[15]:

	3pc	a/n	aes-sp3-d	any	argus	aris	arp	ax.25	bbn-rcc	bna	...	Analysis	Backdoor	DoS	Exploits
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0

5 rows × 165 columns

NORMALIZATION USING MIN-MAX TECHNIQUE

In [16]:
```python
# unsw train data set
unsw_train_data= pd.concat([unsw_train, unsw_train_cat_data],axis=1)
print(unsw_train_data.head(3))
unsw_train_data.drop(columns=['proto','service','state','attack_cat'],inplace=Tru
unsw_train_data
```

```
        srcip  sport          dstip dsport proto state       dur  sbytes  \
0  59.166.0.0   1390  149.171.126.6     53   udp   CON  0.001055     132   
1  59.166.0.0  33661  149.171.126.9   1024   udp   CON  0.036133     528   
2  59.166.0.6   1464  149.171.126.7     53   udp   CON  0.001119     146   

   dbytes  sttl  ...   Fuzzers  Analysis  Backdoors  DoS  Exploits  Generic  \
0     164    31  ...       0.0       0.0        0.0  0.0       0.0      0.0   
1     304    31  ...       0.0       0.0        0.0  0.0       0.0      0.0   
2     178    31  ...       0.0       0.0        0.0  0.0       0.0      0.0   

   Normal  Reconnaissance  Shellcode  Worms  
0     1.0             0.0        0.0    0.0  
1     1.0             0.0        0.0    0.0  
2     1.0             0.0        0.0    0.0  

[3 rows x 223 columns]
```

Out[16]:

	srcip	sport	dstip	dsport	dur	sbytes	dbytes	sttl	dttl	sloss	...	F
0	59.166.0.0	1390	149.171.126.6	53	0.001055	132	164	31	29	0	...	
1	59.166.0.0	33661	149.171.126.9	1024	0.036133	528	304	31	29	0	...	
2	59.166.0.6	1464	149.171.126.7	53	0.001119	146	178	31	29	0	...	
3	59.166.0.5	3593	149.171.126.5	53	0.001209	132	164	31	29	0	...	
4	59.166.0.3	49664	149.171.126.0	53	0.001169	146	178	31	29	0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
699996	59.166.0.8	12520	149.171.126.6	31010	0.020383	320	1874	31	29	1	...	
699997	59.166.0.0	18895	149.171.126.9	80	1.402957	19410	1087890	31	29	2	...	
699998	59.166.0.0	30103	149.171.126.5	5190	0.007108	2158	2464	31	29	6	...	
699999	59.166.0.6	30388	149.171.126.5	111	0.004435	568	304	31	29	0	...	
700000	59.166.0.0	6055	149.171.126.5	54145	0.072974	4238	60788	31	29	7	...	

700001 rows × 219 columns

Loading [MathJax]/extensions/Safe.js

In [17]:
```
# unsw test data set
unsw_test_data= pd.concat([unsw_test,unsw_test_cat_data],axis=1)
print(unsw_test_data.head(3))
unsw_test_data.drop(columns=['proto','service','state','attack_cat'],inplace=True
unsw_test_data
```

```
   id       dur proto service state  spkts  dpkts  sbytes  dbytes       rate  \
0   1  0.121478   tcp       -   FIN      6      4     258     172  74.087490   
1   2  0.649902   tcp       -   FIN     14     38     734   42014  78.473372   
2   3  1.623129   tcp       -   FIN      8     16     364   13186  14.170161   

   ...  Analysis  Backdoor  DoS  Exploits  Fuzzers  Generic  Normal  \
0  ...       0.0       0.0  0.0       0.0      0.0      0.0     1.0   
1  ...       0.0       0.0  0.0       0.0      0.0      0.0     1.0   
2  ...       0.0       0.0  0.0       0.0      0.0      0.0     1.0   

   Reconnaissance  Shellcode  Worms  
0             0.0        0.0    0.0  
1             0.0        0.0    0.0  
2             0.0        0.0    0.0  

[3 rows x 210 columns]
```

Out[17]:

	id	dur	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	..
0	1	0.121478	6	4	258	172	74.087490	252	254	1.415894e+04	..
1	2	0.649902	14	38	734	42014	78.473372	62	252	8.395112e+03	..
2	3	1.623129	8	16	364	13186	14.170161	62	252	1.572272e+03	..
3	4	1.681642	12	12	628	770	13.677108	62	252	2.740179e+03	..
4	5	0.449454	10	6	534	268	33.373826	254	252	8.561499e+03	..
...	...	...	...	...	...	...	...	...	...	...	..
175336	175337	0.000009	2	0	114	0	111111.107200	254	0	5.066666e+07	..
175337	175338	0.505762	10	8	620	354	33.612649	254	252	8.826286e+03	..
175338	175339	0.000009	2	0	114	0	111111.107200	254	0	5.066666e+07	..
175339	175340	0.000009	2	0	114	0	111111.107200	254	0	5.066666e+07	..
175340	175341	0.000009	2	0	114	0	111111.107200	254	0	5.066666e+07	..

175341 rows × 206 columns

Loading [MathJax]/extensions/Safe.js

In [18]:
```
# normalization
# selecting numeric attributes columns from train data
num_col_t = list(unsw_train_data.select_dtypes(include='number').columns)
print(num_col_t)
```

```
['dur', 'sbytes', 'dbytes', 'sttl', 'dttl', 'sloss', 'dloss', 'Sload', 'Dload',
'Spkts', 'Dpkts', 'swin', 'dwin', 'stcpb', 'dtcpb', 'smeansz', 'dmeansz', 'tran
s_depth', 'res_bdy_len', 'Sjit', 'Djit', 'Stime', 'Ltime', 'Sintpkt', 'Dintpk
t', 'tcprtt', 'synack', 'ackdat', 'is_sm_ips_ports', 'ct_state_ttl', 'ct_flw_ht
tp_mthd', 'is_ftp_login', 'ct_ftp_cmd', 'ct_srv_src', 'ct_srv_dst', 'ct_dst_lt
m', 'ct_src_ ltm', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm', 'L
abel', '3pc', 'a/n', 'aes-sp3-d', 'any', 'argus', 'aris', 'arp', 'ax.25', 'bbn-
rcc', 'bna', 'br-sat-mon', 'cbt', 'cftp', 'chaos', 'compaq-peer', 'cphb', 'cpn
x', 'crtp', 'crudp', 'dcn', 'ddp', 'ddx', 'dgp', 'egp', 'eigrp', 'emcon', 'enca
p', 'esp', 'etherip', 'fc', 'fire', 'ggp', 'gmtp', 'gre', 'hmp', 'i-nlsp', 'iat
p', 'ib', 'icmp', 'idpr', 'idpr-cmtp', 'idrp', 'ifmp', 'igmp', 'igp', 'il', 'i
p', 'ipcomp', 'ipcv', 'ipip', 'iplt', 'ipnip', 'ippc', 'ipv6', 'ipv6-frag', 'ip
v6-no', 'ipv6-opts', 'ipv6-route', 'ipx-n-ip', 'irtp', 'isis', 'iso-ip', 'iso-t
p4', 'kryptolan', 'l2tp', 'larp', 'leaf-1', 'leaf-2', 'merit-inp', 'mfe-nsp',
'mhrp', 'micp', 'mobile', 'mtp', 'mux', 'narp', 'netblt', 'nsfnet-igp', 'nvp',
'ospf', 'pgm', 'pim', 'pipe', 'pnni', 'pri-enc', 'prm', 'ptp', 'pup', 'pvp', 'q
nx', 'rdp', 'rsvp', 'rtp', 'rvd', 'sat-expak', 'sat-mon', 'sccopmce', 'scps',
'sctp', 'sdrp', 'secure-vmtp', 'sep', 'skip', 'sm', 'smp', 'snp', 'sprite-rpc',
'sps', 'srp', 'st2', 'stp', 'sun-nd', 'swipe', 'tcf', 'tcp', 'tlsp', 'tp++', 't
runk-1', 'trunk-2', 'ttp', 'udp', 'udt', 'unas', 'uti', 'vines', 'visa', 'vmt
p', 'vrrp', 'wb-expak', 'wb-mon', 'wsn', 'xnet', 'xns-idp', 'xtp', 'zero', '-',
'dhcp', 'dns', 'ftp', 'ftp-data', 'http', 'irc', 'pop3', 'radius', 'smtp', 'snm
p', 'ssh', 'ssl', 'ACC', 'CLO', 'CON', 'ECO', 'ECR', 'FIN', 'INT', 'MAS', 'PA
R', 'REQ', 'RST', 'TST', 'TXD', 'URH', 'URN', 'no', ' Fuzzers', 'Analysis', 'Ba
ckdoors', 'DoS', 'Exploits', 'Generic', 'Normal', 'Reconnaissance', 'Shellcod
e', 'Worms']
```

Loading [MathJax]/extensions/Safe.js

In [19]:
```
# selecting numeric attributes columns from test data
num_col = list(unsw_test_data.select_dtypes(include='number').columns)
num_col.remove('id')
num_col.remove('label')
print(num_col)
```

```
['dur', 'spkts', 'dpkts', 'sbytes', 'dbytes', 'rate', 'sttl', 'dttl', 'sload',
'dload', 'sloss', 'dloss', 'sinpkt', 'dinpkt', 'sjit', 'djit', 'swin', 'stcpb',
'dtcpb', 'dwin', 'tcprtt', 'synack', 'ackdat', 'smean', 'dmean', 'trans_depth',
'response_body_len', 'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm', 'ct_src_dport_
ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm', 'is_ftp_login', 'ct_ftp_cmd', 'ct_f
lw_http_mthd', 'ct_src_ltm', 'ct_srv_dst', 'is_sm_ips_ports', '3pc', 'a/n', 'ae
s-sp3-d', 'any', 'argus', 'aris', 'arp', 'ax.25', 'bbn-rcc', 'bna', 'br-sat-mo
n', 'cbt', 'cftp', 'chaos', 'compaq-peer', 'cphb', 'cpnx', 'crtp', 'crudp', 'dc
n', 'ddp', 'ddx', 'dgp', 'egp', 'eigrp', 'emcon', 'encap', 'etherip', 'fc', 'fi
re', 'ggp', 'gmtp', 'gre', 'hmp', 'i-nlsp', 'iatp', 'ib', 'icmp', 'idpr', 'idpr
-cmtp', 'idrp', 'ifmp', 'igmp', 'igp', 'il', 'ip', 'ipcomp', 'ipcv', 'ipip', 'i
plt', 'ipnip', 'ippc', 'ipv6', 'ipv6-frag', 'ipv6-no', 'ipv6-opts', 'ipv6-rout
e', 'ipx-n-ip', 'irtp', 'isis', 'iso-ip', 'iso-tp4', 'kryptolan', 'l2tp', 'lar
p', 'leaf-1', 'leaf-2', 'merit-inp', 'mfe-nsp', 'mhrp', 'micp', 'mobile', 'mt
p', 'mux', 'narp', 'netblt', 'nsfnet-igp', 'nvp', 'ospf', 'pgm', 'pim', 'pipe',
'pnni', 'pri-enc', 'prm', 'ptp', 'pup', 'pvp', 'qnx', 'rdp', 'rsvp', 'rtp', 'rv
d', 'sat-expak', 'sat-mon', 'sccopmce', 'scps', 'sctp', 'sdrp', 'secure-vmtp',
'sep', 'skip', 'sm', 'smp', 'snp', 'sprite-rpc', 'sps', 'srp', 'st2', 'stp', 's
un-nd', 'swipe', 'tcf', 'tcp', 'tlsp', 'tp++', 'trunk-1', 'trunk-2', 'ttp', 'ud
p', 'unas', 'uti', 'vines', 'visa', 'vmtp', 'vrrp', 'wb-expak', 'wb-mon', 'ws
n', 'xnet', 'xns-idp', 'xtp', 'zero', '-', 'dhcp', 'dns', 'ftp', 'ftp-data', 'h
ttp', 'irc', 'pop3', 'radius', 'smtp', 'snmp', 'ssh', 'ssl', 'CON', 'ECO', 'FI
N', 'INT', 'PAR', 'REQ', 'RST', 'URN', 'no', 'Analysis', 'Backdoor', 'DoS', 'Ex
ploits', 'Fuzzers', 'Generic', 'Normal', 'Reconnaissance', 'Shellcode', 'Worm
s']
```

Loading [MathJax]/extensions/Safe.js

In [20]:
```
# train normalise
# using minmax scaler for normalizing data
print("data before normalisation")
print(unsw_train_data.head())
print("\n\n")

from sklearn.preprocessing import MinMaxScaler

minmax_scale = MinMaxScaler(feature_range=(0, 1))
def normalization(df,col):
  for i in col:
    arr = df[i]
    arr = np.array(arr)
    df[i] = minmax_scale.fit_transform(arr.reshape(len(arr),1))
  return df

print("data after normalisation")
train = normalization(unsw_train_data,num_col_t)
print(train.head(2))
```

```
data before normalisation
        srcip  sport          dstip dsport       dur  sbytes  dbytes  sttl  \
0  59.166.0.0   1390  149.171.126.6     53  0.001055     132     164    31   
1  59.166.0.0  33661  149.171.126.9   1024  0.036133     528     304    31   
2  59.166.0.6   1464  149.171.126.7     53  0.001119     146     178    31   
3  59.166.0.5   3593  149.171.126.5     53  0.001209     132     164    31   
4  59.166.0.3  49664  149.171.126.0     53  0.001169     146     178    31   

   dttl  sloss  ...  Fuzzers  Analysis  Backdoors  DoS  Exploits  Generic  \
0    29      0  ...      0.0       0.0        0.0  0.0       0.0      0.0   
1    29      0  ...      0.0       0.0        0.0  0.0       0.0      0.0   
2    29      0  ...      0.0       0.0        0.0  0.0       0.0      0.0   
3    29      0  ...      0.0       0.0        0.0  0.0       0.0      0.0   
4    29      0  ...      0.0       0.0        0.0  0.0       0.0      0.0   

   Normal  Reconnaissance  Shellcode  Worms  
0     1.0             0.0        0.0    0.0  
1     1.0             0.0        0.0    0.0  
2     1.0             0.0        0.0    0.0  
3     1.0             0.0        0.0    0.0  
4     1.0             0.0        0.0    0.0  

[5 rows x 219 columns]



data after normalisation
        srcip  sport          dstip dsport           dur    sbytes    dbytes  \
0  59.166.0.0   1390  149.171.126.6     53  1.200687e-07  0.000010  0.000011   
1  59.166.0.0  33661  149.171.126.9   1024  4.112267e-06  0.000039  0.000021   

       sttl      dttl  sloss  ...  Fuzzers  Analysis  Backdoors  DoS  \
0  0.121569  0.114173    0.0  ...      0.0       0.0        0.0  0.0   
1  0.121569  0.114173    0.0  ...      0.0       0.0        0.0  0.0   

   Exploits  Generic  Normal  Reconnaissance  Shellcode  Worms  
0       0.0      0.0     1.0             0.0        0.0    0.0  
```

```
1       0.0      0.0     1.0            0.0        0.0    0.0

[2 rows x 219 columns]
```

Loading [MathJax]/extensions/Safe.js

In [21]:
```
# test normalise
# using minmax scaler for normalizing data
print("data before normalisation")
print(unsw_test_data.head(3))
print("\n\n")

from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

minmax_scale = MinMaxScaler(feature_range=(0, 1))
def normalization(df,col):
  for i in col:
    arr = df[i]
    arr = np.array(arr)
    df[i] = minmax_scale.fit_transform(arr.reshape(len(arr),1))
  return df

print("data after normalisation")
test = normalization(unsw_test_data,num_col)
print(test.head(3))
```

```
data before normalisation
   id       dur  spkts  dpkts  sbytes  dbytes       rate  sttl  dttl  \
0   1  0.121478      6      4     258     172  74.087490   252   254   
1   2  0.649902     14     38     734   42014  78.473372    62   252   
2   3  1.623129      8     16     364   13186  14.170161    62   252   

          sload  ...  Analysis  Backdoor  DoS  Exploits  Fuzzers  Generic  \
0  14158.942380  ...       0.0       0.0  0.0       0.0      0.0      0.0   
1   8395.112305  ...       0.0       0.0  0.0       0.0      0.0      0.0   
2   1572.271851  ...       0.0       0.0  0.0       0.0      0.0      0.0   

   Normal  Reconnaissance  Shellcode  Worms  
0     1.0             0.0        0.0    0.0  
1     1.0             0.0        0.0    0.0  
2     1.0             0.0        0.0    0.0  

[3 rows x 206 columns]



data after normalisation
   id       dur     spkts     dpkts    sbytes    dbytes      rate      sttl  \
0   1  0.002025  0.000520  0.000364  0.000018  0.000012  0.000074  0.988235   
1   2  0.010832  0.001352  0.003463  0.000054  0.002867  0.000078  0.243137   
2   3  0.027052  0.000728  0.001458  0.000026  0.000900  0.000014  0.243137   

       dttl         sload  ...  Analysis  Backdoor  DoS  Exploits  Fuzzers  \
0  1.000000  2.364553e-06  ...       0.0       0.0  0.0       0.0      0.0   
1  0.992126  1.401989e-06  ...       0.0       0.0  0.0       0.0      0.0   
2  0.992126  2.625704e-07  ...       0.0       0.0  0.0       0.0      0.0   

   Generic  Normal  Reconnaissance  Shellcode  Worms  
0      0.0     1.0             0.0        0.0    0.0  
1      0.0     1.0             0.0        0.0    0.0  
2      0.0     1.0             0.0        0.0    0.0  
```

Loading [MathJax]/extensions/Safe.js

```
[3 rows x 206 columns]
```

In [ ]:

In [ ]:

Loading [MathJax]/extensions/Safe.js