Anomaly Detection

Done by Rishabh Rakesh

Business Understanding

Designing intrusion detection systems using machine learning and data mining algorithms.

Globally, Security of computers and the networks that connect them is increasingly becoming of great significance. Machine learning and data mining algorithms play important roles in designing intrusion detection systems.

In the anomaly detection approach, on the other hand, anomalous states in a system are identified based on a significant difference in the state transitions of the system from its normal states.

Ideas

Link:- https://arxiv.org/ftp/arxiv/papers/1610/1610.04306.pdf

Table2. Network packets database

ID	service	src_bytes	dst_bytes	duration	…
r1	telnet	100	2000	13	…
r2	ftp	200	300	2	…
r3	smtp	250	300	1	…
r4	telnet	200	12100	60	…
r5	smtp	200	300	1	…
…	…	…	…	…	…

Perform Discretization to convert given table into-->

Table 3. Discretization result of network packets database

ID	service	src_bytes	dst_bytes	duration	…
r1	A	D	E	G	…
r2	B	D	F	H	…
r3	C	D	F	H	…
r4	A	D	E	G	…
r5	C	D	F	H	…
…	…	…	…	…	…

A set of all items in Table 3 is I={A, B, C, D, E, F, G, H}, where A: [f_1=telnet], B: [f_1=ftp], C: [f_1=smtp], D: [f_2= src_bytes $\leq$ 300], E: [f_3= dst_bytes > 1000], F: [f_3= dst_bytes $\leq$ 1000], G: [f_4= duration > 10], H: [f_4= duration $\leq$ 10].

Data Acquisition

- Data was Provided By the Institute, is similar to well known datasets in the field (for reference : https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-)
- Usually The data is created by collecting Packets Flowing through a computer Network, Allowing one to get varied Amount of data based on different kinds of user present in Data. Usually Some=sort of Wireshark or similar tool is used to collect this data.

Understanding

By Utilizing the Common use case of MArket BAsket Analysis and IDS, we came up with following Differences in functionality:

(1) While the database for market basket analysis is a transaction database in which each transaction has different length (i.e. the number of data items in a transaction), the database of for intrusion detection is a relational database of which record length is same.

(2) In a realistic case, there can be many hundreds or even many thousands of products (data items) in database for market basket analysis. In contrast to this, network audit databases face tens of attributes.

(3) For market basket analysis, an association rule is the implication X→Y, where X and Y are itemsets like {I1, I2, …, In}. But, for intrusion detection, X and Y are itemsets like {f1=q1, f2=q2, …, fn=qn}, where fk(k=1, 2, …, n) is item name (field name) and qk (k=1, 2, …, n) is a value of item.

Wrangling 1

```
In [115]: factor = pd.qcut(dataset['src_bytes'],[0,0.5,1])
          pd.value_counts(factor)

Out[115]: (-0.001, 54.0]          11308
          (54.0, 62825648.0]      11236
          Name: src_bytes, dtype: int64
```

Utilizing the pd.cut/ pd.qcut for Wrangling and Discretizing the features

Wrangling 2

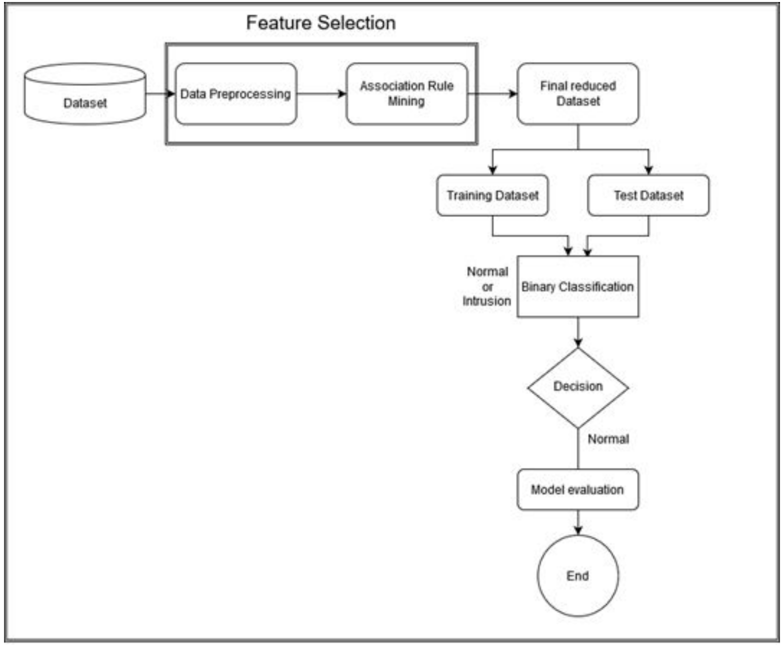
```
In [9]: for col in discrete_col:
            thresholder(col)

duration threshold =  28857.5
src_bytes threshold =  31412824.0
dst_bytes threshold =  672963.5
land threshold =  0.5
wrong_fragment threshold =  1.5
hot threshold =  50.5
num_failed_logins threshold =  2.0
logged_in threshold =  0.5
num_compromised threshold =  398.0
root_shell threshold =  0.5
su_attempted threshold =  1.0
num_root threshold =  439.0
num_file_creations threshold =  50.0
num_shells threshold =  2.5
num_access_files threshold =  2.0
num_outbound_cmds threshold =  0.0
is_guest_login threshold =  0.5
count threshold =  255.5
srv_count threshold =  255.5
serror_rate threshold =  0.5
srv_serror_rate threshold =  0.5
rerror_rate threshold =  0.5
srv_rerror_rate threshold =  0.5
same_srv_rate threshold =  1.0
diff_srv_rate threshold =  0.5
srv_diff_host_rate threshold =  0.5
dst_host_count threshold =  255.0
dst_host_srv_count threshold =  255.0
dst_host_same_srv_rate threshold =  1.0
dst_host_diff_srv_rate threshold =  0.5
dst_host_same_src_port_rate threshold =  0.5
dst_host_srv_diff_host_rate threshold =  0.5
dst_host_serror_rate threshold =  0.5
dst_host_srv_serror_rate threshold =  0.5
dst_host_rerror_rate threshold =  0.5
dst_host_srv_rerror_rate threshold =  0.5
```

Various Thresholds for 36 Numerical Columns And Converting it to⟶ Which was suitable For Rules Association Mining

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	hot	num_failed_logins	...	dst_host_srv_count	dst_host_same_srv_r
0	duration < 28857.5	tcp	private	REJ	src_bytes < 31412824.0	dst_bytes < 672963.5	land < 0.5	wrong_fragment < 1.5	hot < 50.5	num_failed_logins < 2.0	...	dst_host_srv_count < 255.0	dst_host_same_srv_r <
1	duration < 28857.5	tcp	private	REJ	src_bytes < 31412824.0	dst_bytes < 672963.5	land < 0.5	wrong_fragment < 1.5	hot < 50.5	num_failed_logins < 2.0	...	dst_host_srv_count < 255.0	dst_host_same_srv_r <
2	duration < 28857.5	tcp	ftp_data	SF	src_bytes < 31412824.0	dst_bytes < 672963.5	land < 0.5	wrong_fragment < 1.5	hot < 50.5	num_failed_logins < 2.0	...	dst_host_srv_count < 255.0	dst_host_same_srv_r <
3	duration < 28857.5	icmp	eco_i	SF	src_bytes < 31412824.0	dst_bytes < 672963.5	land < 0.5	wrong_fragment < 1.5	hot < 50.5	num_failed_logins < 2.0	...	dst_host_srv_count < 255.0	dst_host_same_srv_r >=
4	duration < 28857.5	tcp	telnet	RSTO	src_bytes < 31412824.0	dst_bytes < 672963.5	land < 0.5	wrong_fragment < 1.5	hot < 50.5	num_failed_logins < 2.0	...	dst_host_srv_count < 255.0	dst_host_same_srv_r <

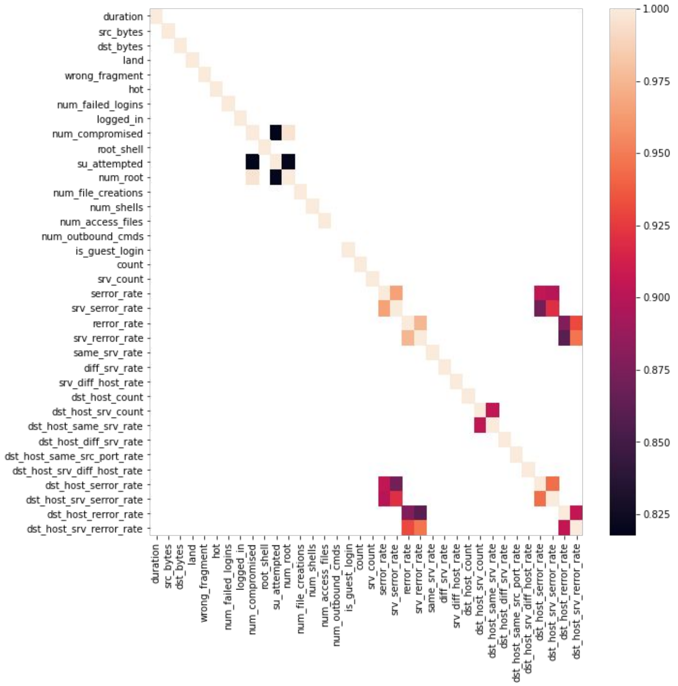
5 rows × 40 columns

Feature Selection



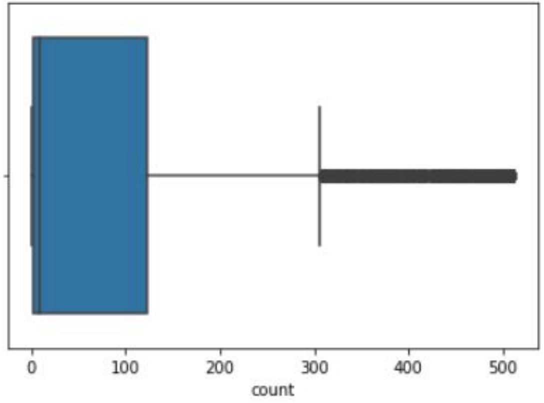
System Architecture

Results of FE technique 1



- Utilizing Correlation Matrix to filter Features
- Creating Filtered Correlation Plot.
- Creating Box Plot to understand the Distribution of the Dataset.
- Creating Density Plots

Results of FE technique 2

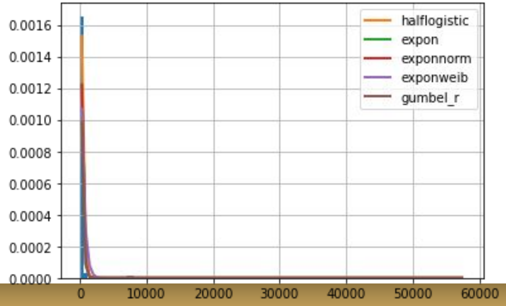


In [71]:
```
data[].groupby(['service','flag','protocol_type'])[discrete_col].sum()
```

Out[71]:

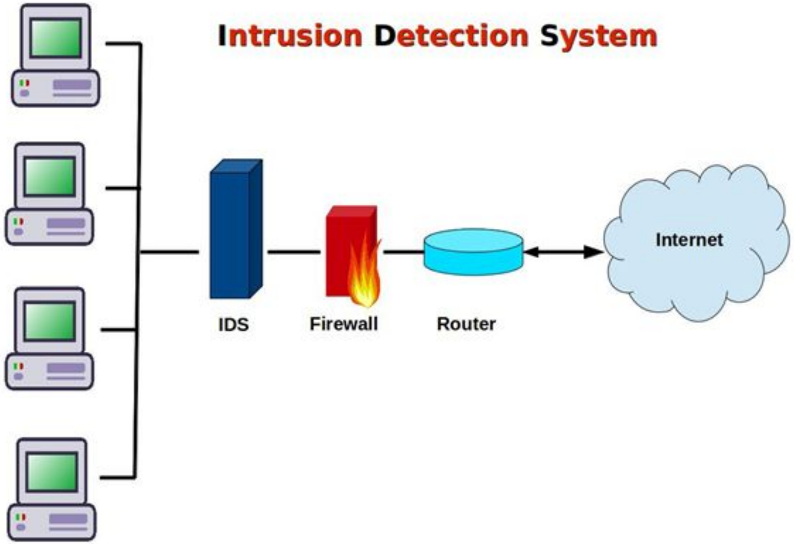
service	flag	protocol_type	duration	src_bytes	dst_bytes	land	wrong_fragment	hot	num_failed_logins	logged_in	num_compromised	root_shell	...
IRC	REJ	tcp	0	0	0	0	0	0	0	0	0	0	...
	RSTO	tcp	4560	938	4725	0	0	0	0	0	0	0	...
	RSTR	tcp	50537	9625	56479	0	0	0	0	0	0	0	...
	SF	tcp	134	342	1011	0	0	0	0	0	0	0	...
X11	S1	tcp	0	314868	415220	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...
uucp_path	S0	tcp	0	0	0	0	0	0	0	0	0	0	...
vmnet	REJ	tcp	0	0	0	0	0	0	0	0	0	0	...
	S0	tcp	0	0	0	0	0	0	0	0	0	0	...
whois	REJ	tcp	0	0	0	0	0	0	0	0	0	0	...
	S0	tcp	0	0	0	0	0	0	0	0	0	0	...

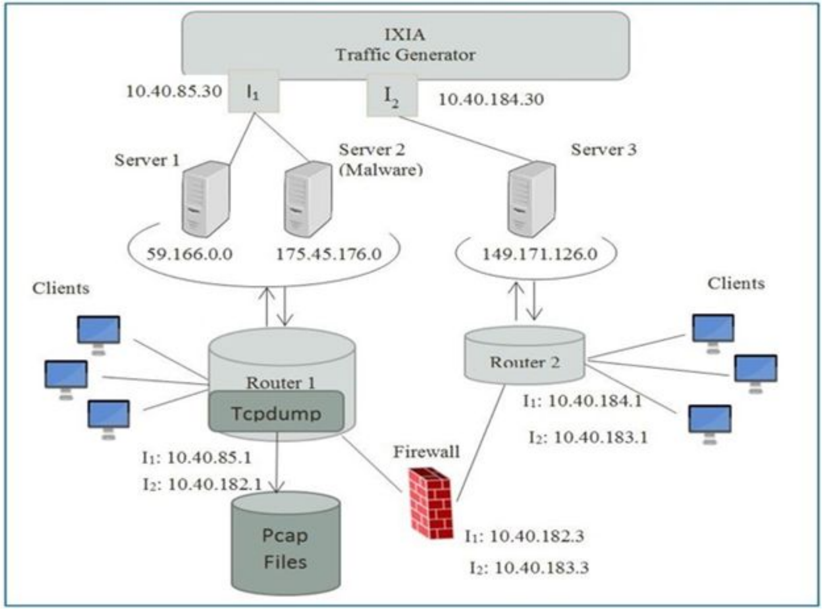
189 rows × 36 columns



Modeling

“The goal is to turn **data** into information, and information into insight.” – Carly Fiorina





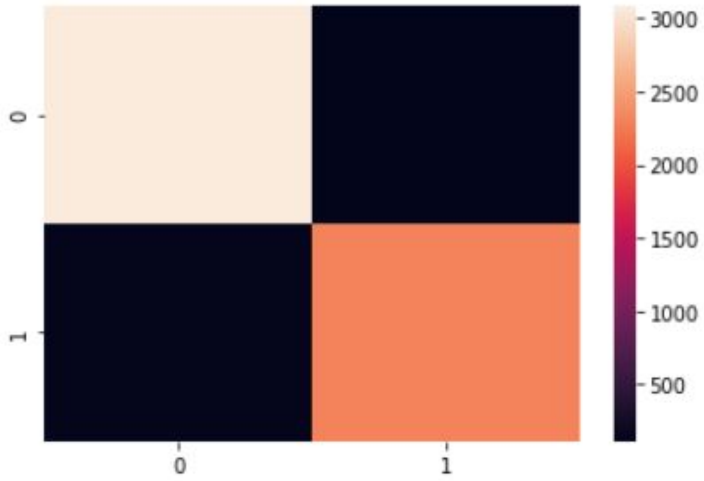
Results of ML technique 1

Association Rules Mining

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(_)	(e)	0.001464	0.001286	0.001242	0.848485	659.594566	0.001240	6.591510
1	(e)	(_)	0.001286	0.001464	0.001242	0.965517	659.594566	0.001240	28.957550
2	(o)	(_)	0.001286	0.001464	0.001153	0.896552	612.480669	0.001151	9.652517
3	(_)	(o)	0.001464	0.001286	0.001153	0.787879	612.480669	0.001151	4.708221
4	(r)	(_)	0.001153	0.001464	0.001065	0.923077	630.601399	0.001063	12.980971
5	(_)	(r)	0.001464	0.001153	0.001065	0.727273	630.601399	0.001063	3.662438
6	(s)	(_)	0.001331	0.001464	0.001242	0.933333	637.608081	0.001240	14.978043
7	(_)	(s)	0.001464	0.001331	0.001242	0.848485	637.608081	0.001240	6.591217
8	(t)	(_)	0.001375	0.001464	0.001242	0.903226	617.040078	0.001240	10.318207
9	(_)	(t)	0.001464	0.001375	0.001242	0.848485	617.040078	0.001240	6.590924
10	(s)	(e)	0.001331	0.001286	0.001109	0.833333	647.816092	0.001107	5.992282
11	(e)	(s)	0.001286	0.001331	0.001109	0.862069	647.816092	0.001107	7.240352
12	(t)	(e)	0.001375	0.001286	0.001020	0.741935	576.765295	0.001018	3.870015
13	(e)	(t)	0.001286	0.001375	0.001020	0.793103	576.765295	0.001018	4.826687
14	(t)	(o)	0.001375	0.001286	0.001153	0.838710	651.995551	0.001152	6.192024
15	(o)	(t)	0.001286	0.001375	0.001153	0.896552	651.995551	0.001152	9.653374
16	(t)	(r)	0.001375	0.001153	0.001065	0.774194	671.285360	0.001063	4.423464
17	(r)	(t)	0.001153	0.001375	0.001065	0.923077	671.285360	0.001063	12.982124
18	(s)	(t)	0.001331	0.001375	0.001065	0.800000	581.780645	0.001063	4.993125
19	(t)	(s)	0.001375	0.001331	0.001065	0.774194	581.780645	0.001063	4.422678
20	(s, _)	(e)	0.001242	0.001286	0.001065	0.857143	666.325123	0.001063	6.990995
21	(s, e)	(_)	0.001109	0.001464	0.001065	0.960000	655.825455	0.001063	24.963405
22	(_, e)	(s)	0.001242	0.001331	0.001065	0.857143	644.114286	0.001063	6.990685

Results of ML technique 2

Random Forest Classifier




```
[[3082  112]
 [ 137 2305]]
```

	precision	recall	f1-score	support
anomaly	0.96	0.96	0.96	3194
normal	0.95	0.94	0.95	2442
accuracy			0.96	5636
macro avg	0.96	0.95	0.95	5636
weighted avg	0.96	0.96	0.96	5636

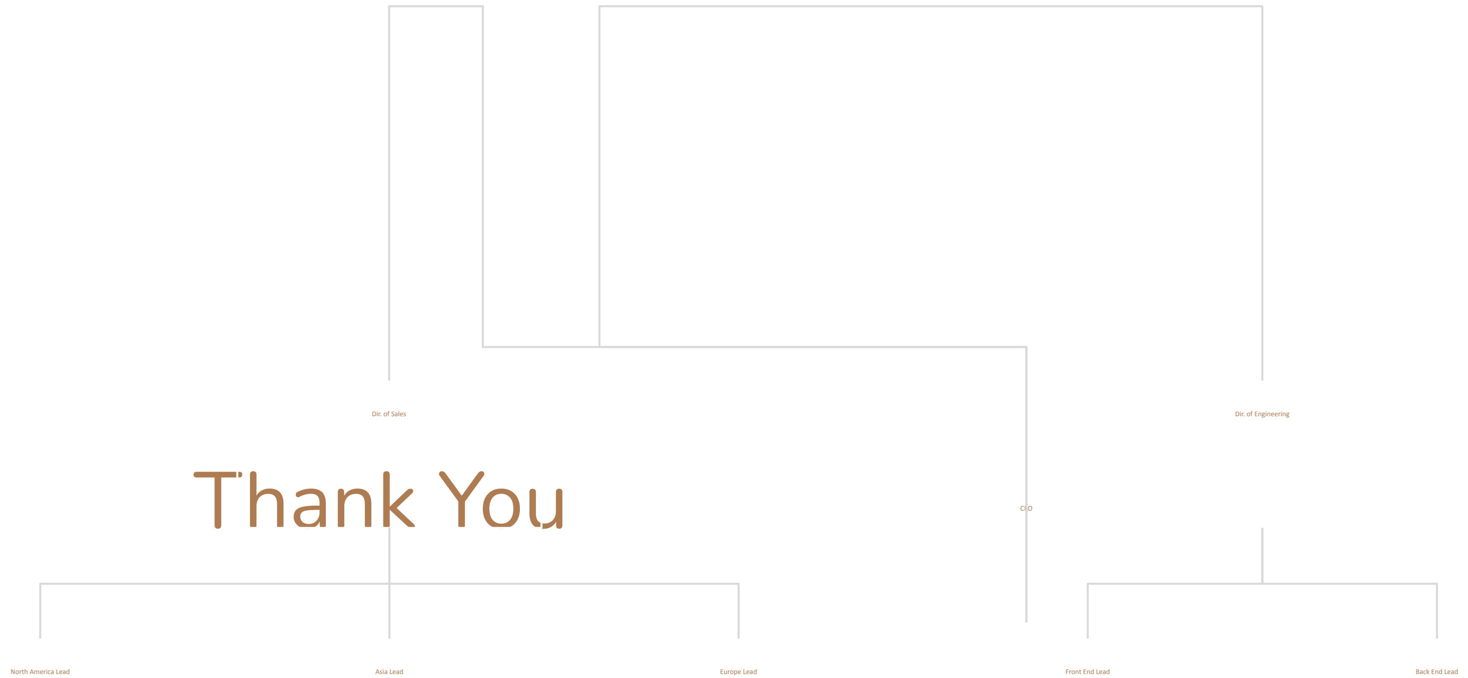
Conclusion

Association Rule Mining Could be used to provide explainability to Random Forest Ensembles.

LIME is one popular Technique which utilizes Association Rule Mining to Create the Trees in Random Forest Which can than be explained Factor By Factor.

SHAP is another popular Technique Though it uses different Mining Technique.

Thank You


Dir. of Sales
Dir. of Engineering
CEO
North America Lead
Asia Lead
Europe Lead
Front End Lead
Back End Lead