DATA MINING FOR INTRUSION DETECTION

Introduction

As a part of cybersecurity efforts, we studied the effectiveness of using data mining techniques in detecting computer network intrusions. By dividing the network connections into good =”normal” and bad =”attack”, the classification technique is used to develop models that can predict network connections that are intrusive.

In the present study, we use four classification algorithms – KNN (K nearest neighbor), TAN (Tree Augmented Naïve Bayesian), CART (Classification and Regression and Trees) and C5.0 (Decision Trees) and compare their predictive accuracies for intrusion. In addition we also use the C5.0 algorithm as a profiling tool to characterize the four different categories of attack, DOS (denial of service), R2L (unauthorized access from remote machine), U2R (unauthorized root privileges) and Probing (port scanning).

The data contains 43 different predictors for 98,327 records including broadly the duration, protocol, connection error rates and classification of the target variable connection into normal and the 22 different attack types. Since we have a large data set, 70% : 30% ratio is used for partitioning data into training and testing sets.

Exploratory Data Analysis

Data auditing shows no missing values.

DISTRIBUTION OF CONNECTION TYPE

Value	Proportion	%	Count
back.		0.46	448
buffer_overflow.		0.01	6
ftp_write.		0.0	1
guess_passwd.		0.01	14
imap.		0.0	2
ipsweep.		0.25	244
land.		0.0	3
loadmodule.		0.0	1
multihop.		0.0	1
neptune.		22.02	21654
nmap.		0.04	43
normal.		19.53	19207
perl.		0.0	1
phf.		0.0	1
pod.		0.06	60
portsweep.		0.21	209
rootkit.		0.0	2
satan.		0.32	314
smurf.		56.65	55704
teardrop.		0.21	206
warezclient.		0.21	202
warezmaster.		0.0	3

Only 19.53% of records are normal. i.e. good. About 80.47% of records are examples of intrusive connections. Comparing to the list of 22 attack types we see that the data does not

contain any spy attacks. A majority of attacks, about 56.65% are Smurf type, followed by Neptune at 22.02%. All other attack types account for are less than 1% of the attacks

DISTRIBUTION OF PROTOCOL TYPE

Value	Proportion	%	Count
icmp		57.2	56240
tcp		38.66	38017
udp		4.14	4069

The data contains only 3 protocol types shown above. Over 50% are the icmp type.

DISTRIBURION OF SERVICE TYPES

Value	Proportion	%	Count
eco_i		0.32	318
ecr_i		56.77	55824
efs		0.02	18
exec		0.02	23
finger		0.14	139
ftp		0.16	155
ftp_data		0.94	924
gopher		0.02	21
hostnames		0.02	21
http		12.87	12656
http_443		0.02	15
imap4		0.03	30
IRC		0.01	13
iso_tsap		0.03	28
klogin		0.02	17
kshell		0.02	21
ldap		0.02	17
link		0.02	22
login		0.02	15
mtp		0.02	17
name		0.02	23
netbios_dgm		0.02	17
netbios_ns		0.02	21
netbios_ssn		0.02	18
netstat		0.02	16
nnsp		0.01	12
nntp		0.02	21
ntp_u		0.09	93
other		1.46	1436
pop_2		0.02	23
pop_3		0.04	39
printer		0.03	25
private		22.79	22412
red_i		0.0	1
remote_job		0.03	26

Data includes many different service types. 56.8% are of ecr_i type, 22.8% private and http at 12.9 %. All other service types account for less than 1%.

DISTRIBUTION OF FLAG

Value	Proportion	%	Count
REJ		5.47	5380
RSTO		0.12	118
RSTOS0		0.0	3
RSTR		0.18	181
S0		17.95	17646
S1		0.01	7
S2		0.01	6
S3		0.0	3
SF		76.24	74962
SH		0.02	20

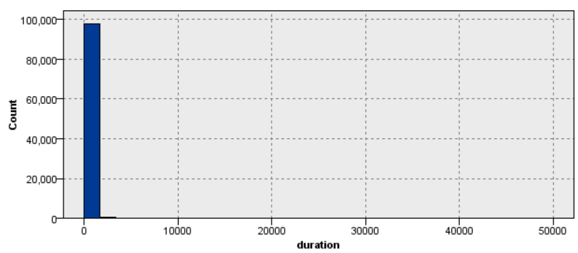
The Flag variable shows the status of the connections. Data mostly has SF type at 76% followed by S0 type at 18% and REJ type at 5.5%.

Although the following predictors have outliers (number of outliers listed below), we do not remove them to ensure that the model can be trained with all possible situations.

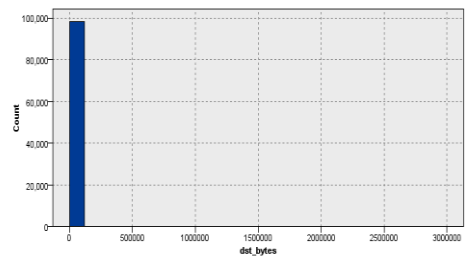
Duration (382), dst_bytes (17), num_root (8), rerror_rate (5673), srv_rerror_rate (5644), diff_srv_rate (94),srv_diff_host_rate (672), dst_host_count (5420) ,dst_host_diff_srv_rate (277) ,dst_host_srv_diff_host_rate (548) ,dst_host_rerror_rate (5575), dst_host_srv_rerror_rate (5559).

Histograms of a few of a few predictors are displayed to show the outliers.

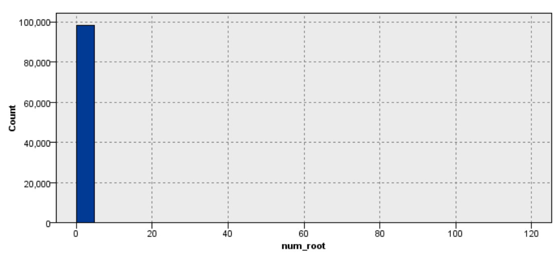
HISTOGRAM OF DURATION



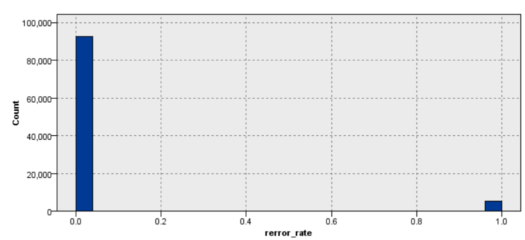
HISTOGRAM OF DST_BTYE



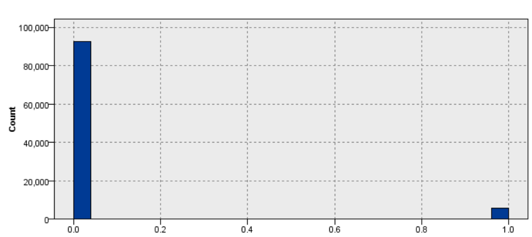
HISTOGRAM OF NUM_ROOT



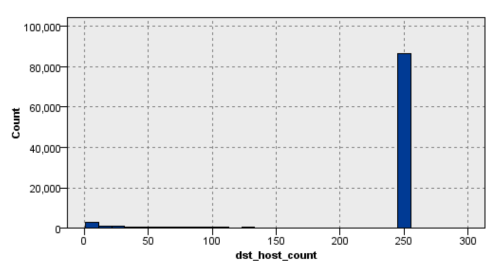
HISTOGRAM OF RERROR_RATE



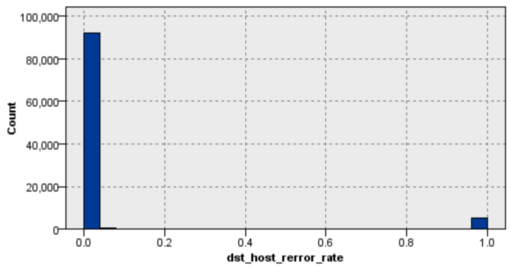
HISTOGRAM OF SRV_RERROR_RATES

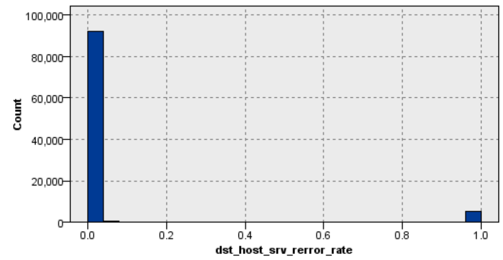


HISTOGRAM OF DST_HOST_COUNT



HISTOGRAM OF DST_HOST_RERROR RATE **HISTOGRAM OF DST_HOST_SRV_REEROR_RAT**

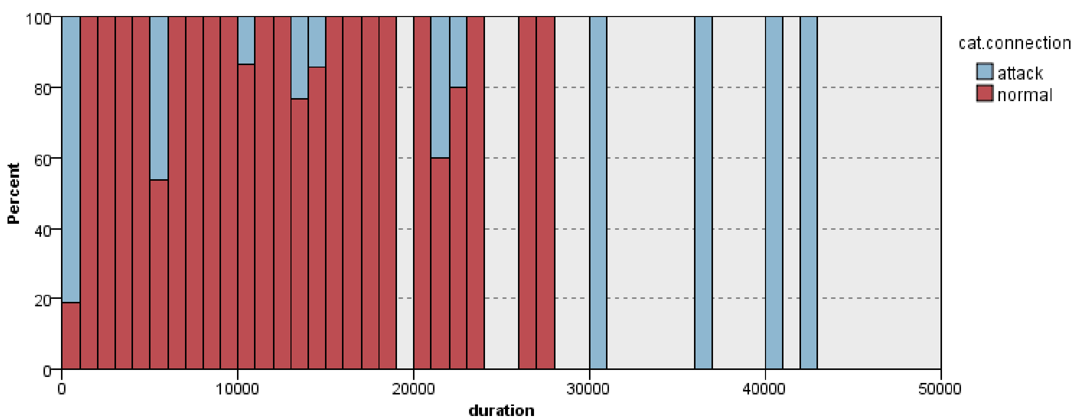




After reclassifying all the 22 target variable connection types into normal and attack categories, data audit confirms about 80:20 ratio of bad to good connections.

Value	Proportion	%	Count
normal		19.53	19207
attack		80.47	79119

DISTRIBUTION OF DURATION WITH CONNECTION TYPE OVERLAY



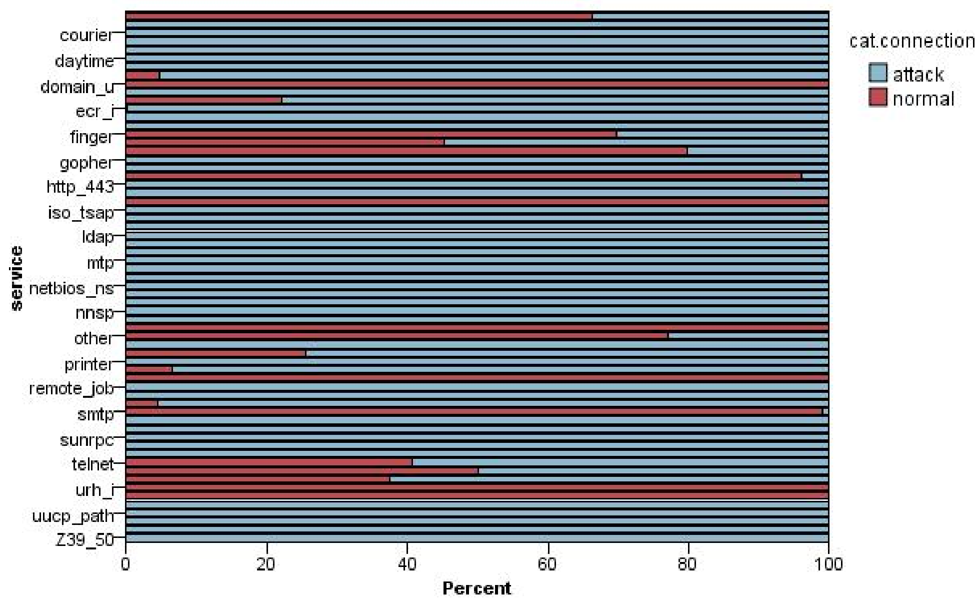
This distribution, shows that above ~30,000, the connections are of the attack type.

DISTRIBUTION OF PROTOCOL TYPE WITH CONNECTION TYPE OVERLAY

Value	Proportion	%	Count
icmp		57.2	56240
tcp		38.66	38017
udp		4.14	4069

The protocol type icmp which account for 57% of all connections, are all attack type. Tcp connections are almost equal proportion normal and attack connections. Udp connections are mostly good, but they account for only 4 % of all connection types

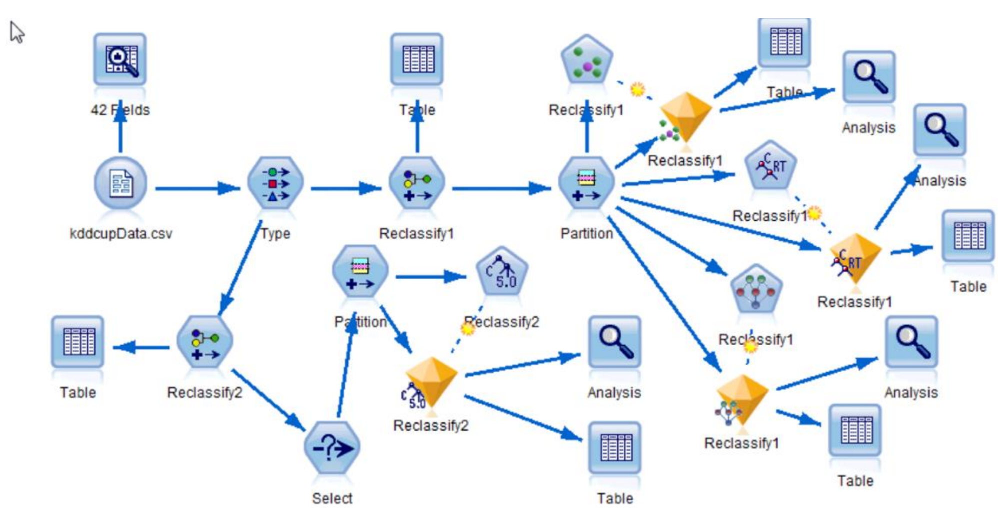
DISTRIBURION OF SERVICE TYPES WITH CONNECTION TYPE OVERLAY



Comparing to previous graphs we see by overlaying connection types that service ecr_i , is 100% bad connection. About 95% of http are good connections and about 30% of private connections are bad connection.

Predictive Models

With a good idea of the variables involved, we now proceed to build different predictive models for computer network connections. The modeling stream is shown below.



KNN-Intrusion Analysis

Results for output field Reclassify1

Comparing $KNN-Reclassify1 with Reclassify1

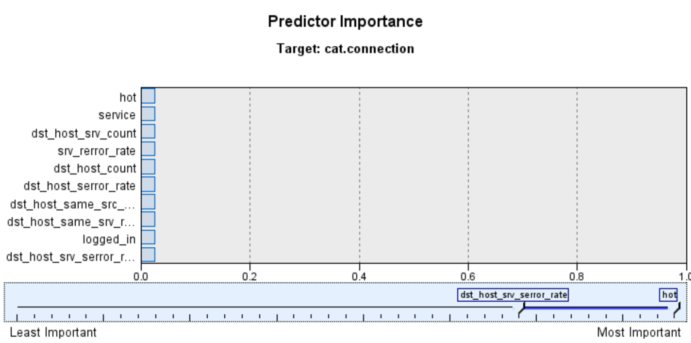
'Partition'	1_Training		2_Testing	
Correct	68,894	99.94%	29,374	99.94%
Wrong	39	0.06%	19	0.06%
Total	68,933		29,393	

Coincidence Matrix for $KNN-Reclassify1 (rows show actuals)

'Partition' = 1_Training	attack.	normal.
attack.	55,411	23
normal.	16	13,483

'Partition' = 2_Testing	attack.	normal.
attack.	23,672	13
normal.	6	5,702

KNN model build took 23 minutes to complete. The analysis results are shown below.



Many predictors, hot, service, logged_in and traffic features are important predictors as shown in the graph.

Testing data	(Attack)	(Normal)
(Attack)	TP =23,672	FN= 13
(Normal)	FP =6	TN =5,702

Accuracy for the test data is 99.94%.

Since the distribution of attack to normal connection types is about 24%, it is somewhat skewed. We therefore look at other metrics of the KNN classification.

Recall = TP/(TP + FN) = 23,672/(23672+13) = 99.94%

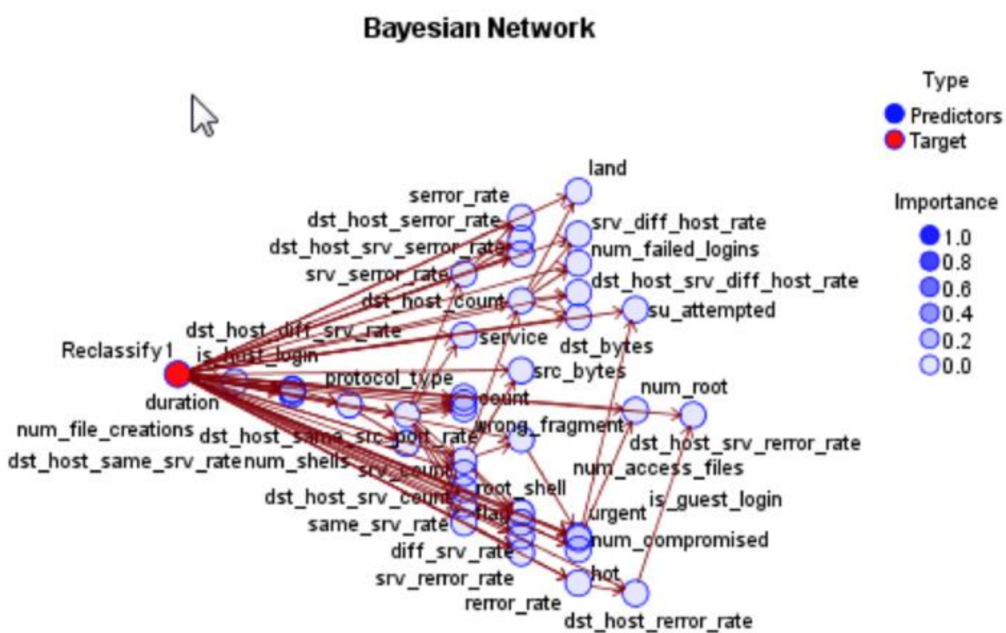
Precision = TP/(TP +FP) = 23672/(23,672+ 6) = 99.97%

1-Specificity = 1- TN/(TN +FP) = 1- TN/(TN+FP) = 0.001

Recall indicates that 99.4% of instances are correctly identified as attack connections. Precision indicates that 99.9.7% of instances that were classified as attack are actually so.1-Specificity indicates that there only 2.8% of false alarms i.e, only 0.1% of attack connections were wrongly classified as normal connections.

TAN (Tree Augmented Naive Bayesian) - Intrusion Analysis

The Bayesian network model below shows that the most important predictor is the variable dst_host_diff_srv_rate – a traffic feature signifying the percentage of the connections to different hosts.



TAN Confusion Matrix:

Results for output field Reclassify1

Comparing $R-Reclassify1 with Reclassify1

'Partition'	1_Training		2_Testing	
Correct	67,546	97.99%	28,772	97.89%
Wrong	1,387	2.01%	621	2.11%
Total	68,933		29,393	

Coincidence Matrix for $R-Reclassify1 (rows show actuals)

'Partition' = 1_Training	attack.	normal.
attack.	54,214	1,220
normal.	167	13,332

'Partition' = 2_Testing	attack.	normal.
attack.	23,141	544
normal.	77	5,631

The Accuracy of TAN model is 97.99%.

Since there are many predictors in the data, we display the conditional probabilities of only the important predictors shown in the Bayesian network diagram.

Conditional Probabilities of dst_host_diff_srv_rate

Parents		Probability				
duration	Reclassify1	<= 0.2	0.2 ~ 0.4	0.4 ~ 0.6	0.6 ~ 0.8	> 0.8
<= 8,489.6	normal.	0.94	0.01	0.01	0.02	0.03
8,489.6 ~ 16,979.2	attack.	0.29	0.00	0.00	0.71	0.00
8,489.6 ~ 16,979.2	normal.	0.16	0.01	0.16	0.36	0.31
16,979.2 ~ 25,468.8	attack.	0.00	0.00	0.00	1.00	0.00
16,979.2 ~ 25,468.8	normal.	0.00	0.00	0.11	0.19	0.70
25,468.8 ~ 33,958.4	attack.	0.00	0.00	0.00	1.00	0.00
25,468.8 ~ 33,958.4	normal.	0.33	0.00	0.00	0.67	0.00
> 33,958.4	attack.	0.33	0.00	0.67	0.00	0.00

Conditional Probabilities of Reclassify1

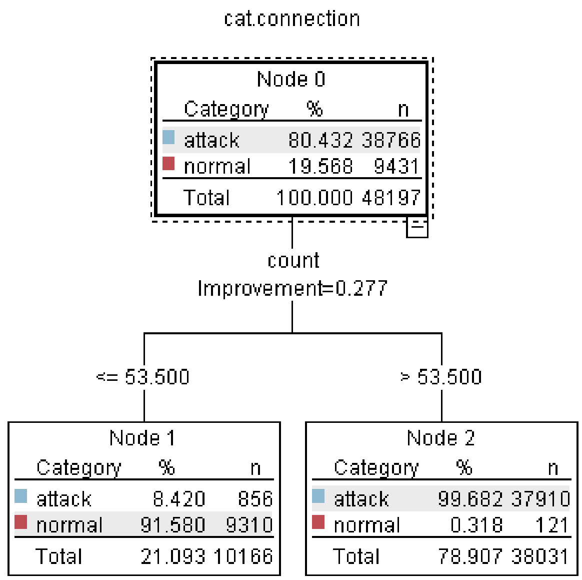
Probability	
attack.	normal.
0.80	0.20

The prior probabilities shown above are as expected from the initial data exploration.

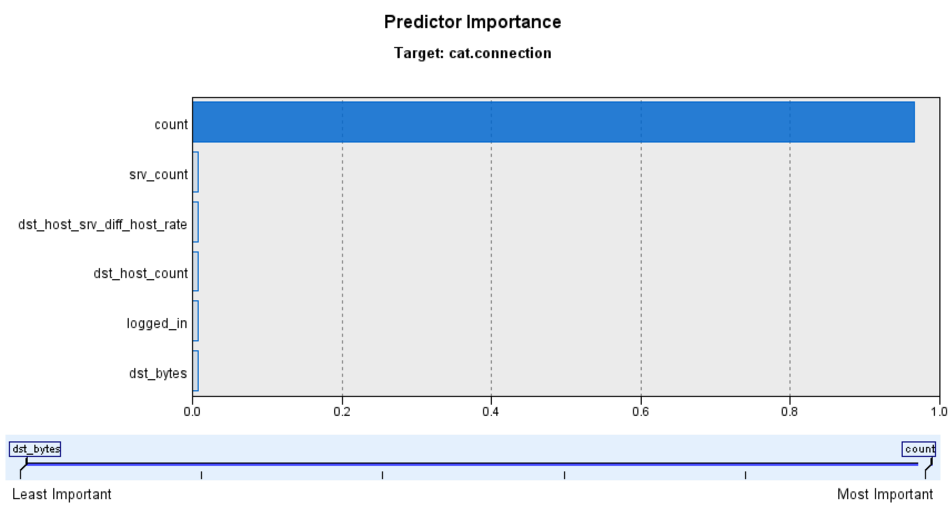
The conditional probabilities are Bayesian classification for intrusive connections.

CART Intrusion Analysis

CART Decision Tree



Only the predictor variable count - the number of connections to the same host as the current connection in the past 2 seconds - seems to be the most important predictor. This is depicted in the bar graph below.



CART Decision Rules:

count <= 53.500 [Mode: normal] ⇨ **normal** (10,166; 0.916)

count > 53.500 [Mode: attack] ⇨ **attack** (38,031; 0.997)

The CART model show that only count- the number of connections to the same host as the current connection in the past 2 seconds is important in deciding whether the connection is normal or attack type.

If count < 53,500 the connection is normal type with 10,166 reported instances and confidence level of 91.6%.

If count >53,500 the connection is intrusive with 38,1031 reported instances and confidence level of 99.7%.

CART Confusion matrix:

Results for output field Reclassify1

Comparing $B-Reclassify1 with Reclassify1

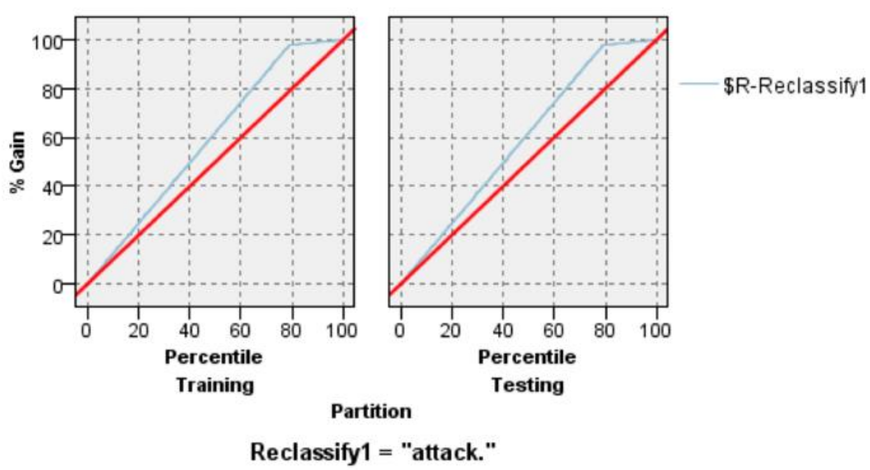
'Partition'	1_Training		2_Testing	
Correct	68,324	99.12%	29,092	98.98%
Wrong	609	0.88%	301	1.02%
Total	68,933		29,393	

Coincidence Matrix for $B-Reclassify1 (rows show actuals)

'Partition' = 1_Training	attack.	normal.
attack.	54,834	600
normal.	9	13,490
'Partition' = 2_Testing	**attack.**	**normal.**
attack.	23,385	300
normal.	1	5,707

The predictive accuracy is 98.98%

CART Gain Curves:



The gain curves show the percentage improvement in classification of an attack compared to random chance strategy. At 80% of the sample we attain the largest improvement with respect to random chance strategy.

C5.0-Intrusion Pruned Analysis

The results for C5.0 classifier are shown below.

C5.0 Confusion Matrix:

Results for output field Reclassify1

Comparing $C-Reclassify1 with Reclassify1

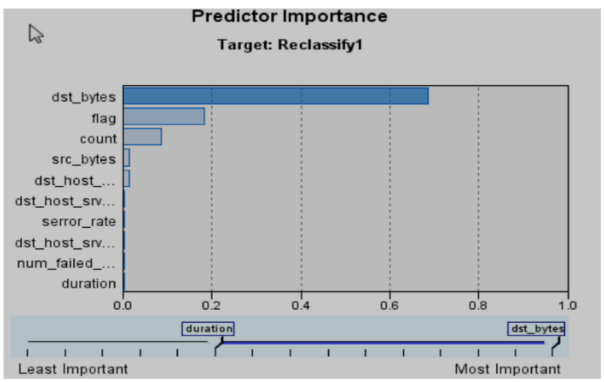
'Partition'	1_Training		2_Testing	
Correct	68,917	99.98%	29,383	99.97%
Wrong	16	0.02%	10	0.03%
Total	68,933		29,393	

Coincidence Matrix for $C-Reclassify1 (rows show actuals)

'Partition' = 1_Training	attack.	normal.
attack.	55,422	12
normal.	4	13,495

'Partition' = 2_Testing	attack.	normal.
attack.	23,682	3
normal.	7	5,701

The confusion matrix for the C5.0 classifier has a 99.97 % accuracy.



The most important predictor is dst_bytes - the number of data bytes from destination to source.

C5.0 Decision Rules:

Rules for attack - contains 11 rule(s)
- Rule 1 for **attack** (15,129; 1.0)
- Rule 2 for **attack** (2,779; 1.0)
- Rule 3 for **attack** (12,500; 1.0)
- Rule 4 for **attack** (39,089; 1.0)
- Rule 5 for **attack** (12,372; 1.0)
- Rule 6 for **attack** (300; 0.997)
- Rule 7 for **attack** (54,381; 0.997)
- Rule 8 for **attack** (195; 0.995)
- Rule 9 for **attack** (116; 0.992)
- Rule 10 for **attack** (181; 0.989)
- Rule 11 for **attack** (35; 0.973)

Rules for normal - contains 3 rule(s)
- Rule 1 for **normal** (11,531; 0.998)
- Rule 2 for **normal** (1,070; 0.993)
- Rule 3 for **normal** (14,552; 0.916)

Default: attack

Descision Rules for normal connections with support and confidence levels.

Rules for normal - contains 3 rule(s)

- Rule 1 for **normal** (11,531; 0.998)
  - if src_bytes <= 40,494
  - and dst_bytes > 1
  - and hot <= 24
  - and same_srv_rate > 0.190
  - and dst_host_diff_srv_rate <= 0.930
  - then **normal**
- Rule 2 for **normal** (1,070; 0.993)
  - if src_bytes > 1,114
  - and src_bytes <= 40,494
  - and wrong_fragment <= 0
  - and hot <= 24
  - and count <= 53
  - then **normal**
- Rule 3 for **normal** (14,552; 0.916)
  - if count <= 53
  - then **normal**

Descision Rules for attack connections with support and confidence levels.

- Rule 1 for **attack** (15,129; 1.0)
  - if src_bytes <= 6
  - and same_srv_rate <= 0.190
  - then **attack**
- Rule 2 for **attack** (2,779; 1.0)
  - if service = private
  - and flag = REJ
  - then **attack**
- Rule 3 for **attack** (12,500; 1.0)
  - if dst_host_same_srv_rate <= 0.100
  - and dst_host_serror_rate > 0.020
  - then **attack**
- Rule 4 for **attack** (39,089; 1.0)
  - if src_bytes > 327
  - and dst_bytes <= 1
  - and dst_host_same_src_port_rate > 0.990
  - then **attack**
- Rule 5 for **attack** (12,372; 1.0)
  - if dst_host_serror_rate > 0.960
  - then **attack**
- Rule 6 for **attack** (300; 0.997)
  - if src_bytes > 40,494
  - and src_bytes <= 54,540
  - then **attack**
- Rule 7 for **attack** (54,381; 0.997)
  - if count > 53
  - then **attack**
- Rule 8 for **attack** (195; 0.995)
  - if wrong_fragment > 0
  - then **attack**
- Rule 9 for **attack** (116; 0.992)
  - if flag = RSTR
  - and src_bytes <= 327
  - then **attack**
- Rule 10 for **attack** (181; 0.989)
  - if flag = SF
  - and src_bytes <= 327
  - and dst_host_same_src_port_rate > 0.990
  - and dst_host_srv_diff_host_rate > 0.120
  - then **attack**
- Rule 11 for **attack** (35; 0.973)
  - if hot > 24
  - and hot <= 28
  - then **attack**

C5.0 Attack Profiles

We see that 98.7% of attacks are dos category. u2r accounts for only 0.01% of the attacks.

Value	Proportion	%	Count
dos		98.68	78075
probe		1.02	810
r2l		0.28	224
u2r		0.01	10

Attack Confusion Matrix:

Results for output field attack.connection

Comparing $C-attack.connection with attack.connection

'Partition'	1_Training		2_Testing	
Correct	55,370	99.98%	23,736	100%
Wrong	12	0.02%	1	0%
Total	55,382		23,737	

Coincidence Matrix for $C-attack.connection (rows show actuals)

'Partition' = 1_Training	dos	probe	r2l	u2r
dos	54,630	1	0	0
probe	5	578	1	0
r2l	1	0	157	1
u2r	1	0	2	5
'Partition' = 2_Testing	dos	probe	r2l	u2r
dos	23,444	0	0	0
probe	1	225	0	0
r2l	0	0	65	0
u2r	0	0	0	2

The matrix shows that C5.0 does an excellent job with 100% accuracy of classifying the 4 attack categories.

Conclusion:

All the models have comparable, high predictive accuracy for detecting intrusive connections. Each model chooses a different variable as important predictors.

KNN: The KNN model has 99.94% accuracy. This algorithm is memory intensive and took 23 min to complete, which is a drawback for implementing on real time large datasets bigger than those used in our study. In this model, hot- the number of "hot" indicators, service, logged_in and 2 sec time window traffic features like dst_host_srv_count,srv_rerror_rate etc are all equally important predictors.

TAN: The TAN model has a 97.89% predictive accuracy. The most important predictor in this model is cdst_host_diff_srv_rate (number of data bytes from destination to source).

CART: The CART algorithm has 98.98% predictive accuracy. This model has only one significant predictor, count - the number of connections to the same host as the current connection in the past 2 secs. The CART model has a simple decision tree and very simple decision rules and appears to be a very simple model for a complex problem.

C 5.0: From the four models, the C 5.0 model has the highest predictive accuracy at 99.97%. The most important predictor in the model is src_bytes - number of data bytes from source to destination. Since the C 5.0 classifier has been pruned to account for overfitting, it has robust decision rules. The algorithm also does an excellent job of sorting the four attacks types, DOS, PROBE, R2L, and U2R in the testing set with 100% accuracy. This can be important in order to identify the best approach to combat the attack.

In conclusion given the parsimony rule, we recommend a dual algorithm approach of CART to identify that an attack is occurring and C5.0 to identify the type of attack.