RESULTS ANALYSIS

Derrick Kempster

Weka

- Waikato Environment for Knowledge Analysis
- Preprocessing and classification capabilities
- Works with .arff (Attribute-Relation File Format) files

- A very accessible application for data identification and analysis tasks

Preprocessing

- Used the provided preprocessed dataset
- Data conversion programs created to change the format to .arff
  - *Part 1 misuse classifier*
    - Add the traffic type as a class attribute
  - *Part 2 misuse classifier*
    - Limit the types to the desired type and “Other”
  - *Anomaly classifiers*
    - Limit the types to “Anomaly” and “Normal”
- Used Weka to divide master sets into 70% training and 30% test sets
  - *Seed of 45 for part 1 and 378 for part 2*

Part 1: Random Forest

- 100 trees
- Depth of 100

IDS Type	Running Time (s)		Accuracy (%)
	Model Building	Model Testing	
Misuse	386.18	5.50	99.9773
Anomaly	368.32	3.75	99.9891

Part 1: Random Forest – Misuse

Actual Class	Determined Class																						Class Acc. (%)
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
A	317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0000
B	0	7	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	87.5000
C	0	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	20.0000
D	0	0	0	11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	91.6667
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	N/A *
F	0	0	0	0	0	1071	0	0	0	0	6	1	0	0	0	0	0	0	0	0	0	0	0	99.3506
G	0	0	0	0	0	0	6	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	75.0000
H	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	33.3333
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	N/A *
J	0	0	0	0	0	0	0	0	0	67989	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0000
K	0	0	0	0	0	9	0	0	0	0	482	0	0	0	0	0	0	0	0	0	0	0	0	98.1670
L	0	0	0	0	0	0	0	0	1	0	1	173173	0	0	0	0	1	0	0	0	0	4	0	99.9960
M	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	100.0000
N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	100.0000
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	67	0	0	0	0	0	0	0	0	100.0000
P	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	905	0	0	0	0	0	0	0	99.8896
Q	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1	0	0	0	0	0	0.0000
R	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	1	0	1480	0	0	0	0	0	99.3289
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	900	0	1	0	0	99.8890
T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.0000
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	281	0	0	100.0000
V	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	258	0	98.4733
W	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	4	66.6667
Determination Acc. (%) †	100.0000	100.0000	100.0000	100.0000	N/A ‡	99.1667	100.0000	100.0000	0.0000	99.9971	98.5685	99.9838	100.0000	100.0000	100.0000	99.8896	0.0000	99.9325	100.0000	N/A ‡	99.6454	98.4733	100.0000	

A: Back B: BufferOverflow C: FTPWrite D: GuessPassword E: Imap F: IPSweep G: Land
H: LoadModule I: MultiHop J: Neptune K: NMap L: Normal M: Perl N:PHF
O: Pod P: PortSweep Q: RootKit R: Satan S: Smurf T: Spy U: TearDrop
V: WarezClient W: WarezMaster *: None of this type were in the test set ‡: No determined instances to compare †: Horizontal

Part 1: Random Forest – Misuse

- False Negatives were 0.0162% of deemed-normal traffic
- Some effects of tracking everything on random split
  - *Imap and MultiHop each had no entries in the test set*
  - *Perl, PHF, and Spy each had 1 entry in the test set*
  - *No traffic was deemed to be Imap or Spy*
- Data split should not be entirely random from the set of all traffic
  - *Matters more when some types have a comparably small entry count*
- Could ensure that 70% of each traffic type gets in the training set
  - *Remainder 30% put in the test set*

Part 1: Random Forest – Anomaly

Actual Class	Determination Class		Class Acc. (%)
	Normal	Anomaly	
Normal	173174	6	99.9965
Anomaly	21	73810	99.9716
Deter-mination Acc. (%) †	99.9879	99.9919	

†: Horizontal

- False Negatives were 0.0121% of deemed-safe traffic
- False Positives were 0.0081% of deemed-unsafe traffic
- Each type is of sufficient size to not need the special split

Part 1: Random Forest – Overview

- The anomaly IDS classifier outperformed the misuse IDS classifier
  - *Higher overall accuracy*
  - *Lower false negative ratio*

Part 2: Multilayer Perceptron

- 5 chosen attacks have the most data (excluding Normal)
- Training time of 500 epochs
- 1 hidden layer
- Learning weight of 0.3
- Momentum of 0.2
- No validation set

IDS Type	Running Time (s)		Accuracy (%)
	Model Building	Model Testing	
IPSweep	645.37	1.29	99.8652
Neptune	641.42	1.31	99.9976
PortSweep	642.36	1.27	99.9899
Satan	656.95	1.26	99.9721
Smurf	631.50	1.28	99.9895
Anomaly	632.30	1.21	99.6956

Part 2: Multilayer Perceptron – IPSweep

Classified As	Actual Class		False (%)
	IPSweep	Other	
IPSweep	1084	322	22.9018
Other	11	245594	0.0045

- 99.8652% accuracy
  - *Second Worst*
- Highest False Positive ratio

Part 2: Multilayer Perceptron – Neptune

Classified As	Actual Class		False (%)
	Neptune	Other	
Neptune	68288	1	0.0015
Other	5	178717	0.0028

- 99.9976% accuracy
  - *Best*
- First classifier with a higher false negative ratio than false positive ratio
  - *Still acceptable for small size*

Part 2: Multilayer Perceptron – PortSweep

Classified As	Actual Class		False (%)
	PortSweep	Other	
PortSweep	852	3	0.3509
Other	22	246134	0.0089

- 99.9899% accuracy
  - *Second Best*

Part 2: Multilayer Perceptron – Satan

Classified As	Actual Class		False (%)
	Satan	Other	
Satan	1427	15	1.0402
Other	54	245515	0.0220

- 99.9721% accuracy
  - *Third Worst*

Part 2: Multilayer Perceptron – Smurf

Classified As	Actual Class		False (%)
	Smurf	Other	
Smurf	912	23	2.4599
Other	3	246073	0.0012

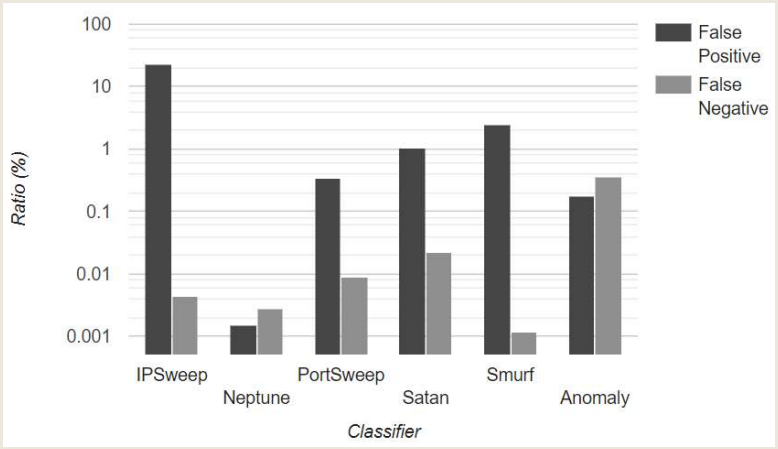
- 99.9895% accuracy
  - *Third Best*

Part 2: Multilayer Perceptron – Anomaly

Classified As	Actual Class		False (%)
	Anomaly	Normal	
Anomaly	73391	128	0.1741
Normal	624	172868	0.3597

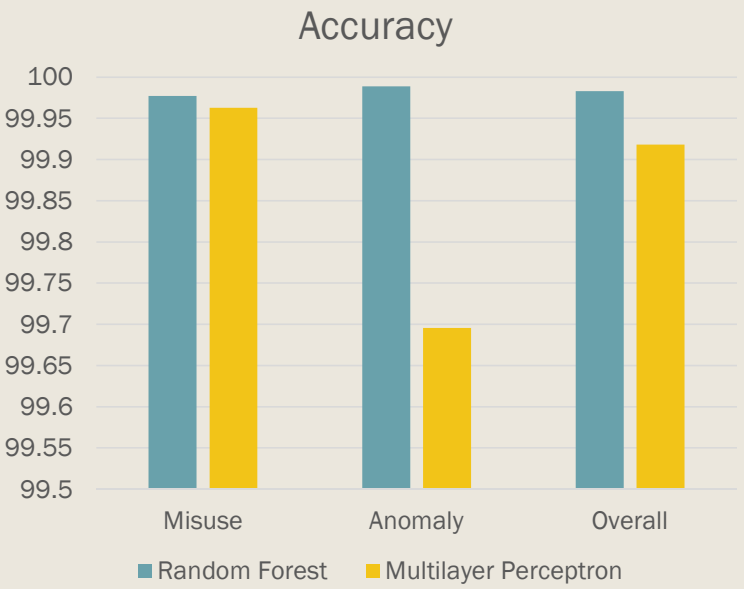
- 99.6956% accuracy
  - *Worst*
- Highest False Negative ratio
- Second classifier with a higher false negative ratio than false positive ratio
  - *Worse than Neptune, but still below 1%*

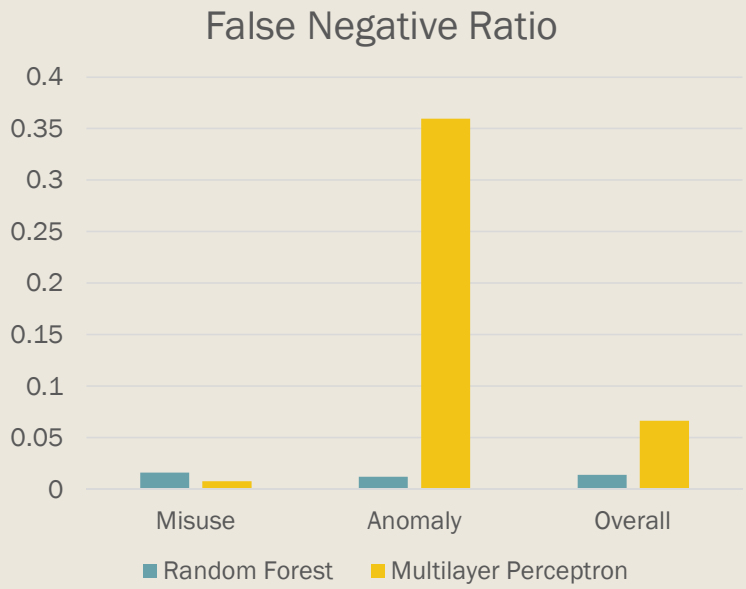
Part 2: Multilayer Perceptron – Overview



- The misuse classifiers outperformed the anomaly classifier
  - *Higher overall accuracy*
  - *Lower false negative ratio*

Part Comparison





Improvement

- Finding a way to save models / use a different classification tool
  - *Models are built right before a testing a data set, and only for that test*
  - *Cannot export the model to run on other data*
- Test additional, more complex classifiers
  - *Will cost additional time to construct*

Conclusions

- The random forest classifiers overall outperformed the multilayer perceptrons
  - *With the given settings applied*
- The multilayer perceptron could reasonably be used for misuse
  - *False negative ratio is lower*
  - *Accuracy is only slightly worse*