Multilayer Perceptron Neural Network for Detection of Encrypted VPN Network Traffic

Shane Miller, Kevin Curran, Tom Lunney

ulster.ac.uk

Structure

Background

- Virtual Private Networks
- Why Detect them?

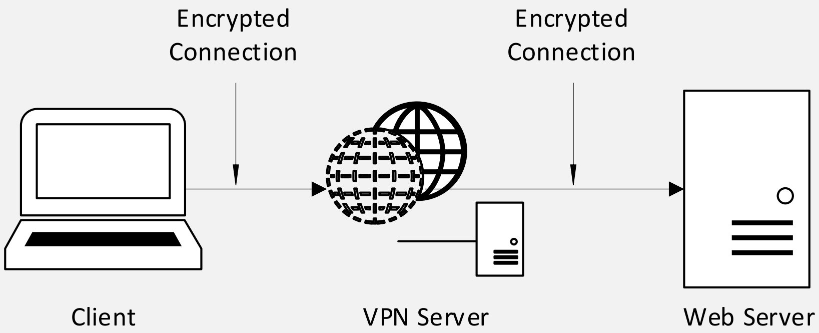
Methodology

- Dataset Capture
- Feature Selection
- Weka Experiment Setup
- Results

Summary and Future Work

Virtual Private Networks

What are they?



Virtual Private Networks (VPNs) provide private networks of resources and information over any public network.

They were commonly used by organisations to connect their resources over the Internet to remote workers.

There are multiple types of VPN available. PPTP, L2TP, L2TP + IPSec and SSL based are some examples.

The focus of this work is on OpenVPN, a well known and popular VPN invented in 2001 by James Yonan.

Popular due to it’s simple configuration, ease of use and a mix of enterprise-level security.

Why detect them?

Why are they a problem?

VPNs are gaining popularity amongst hackers and criminals around the world as they allow individuals to hide their online activities.

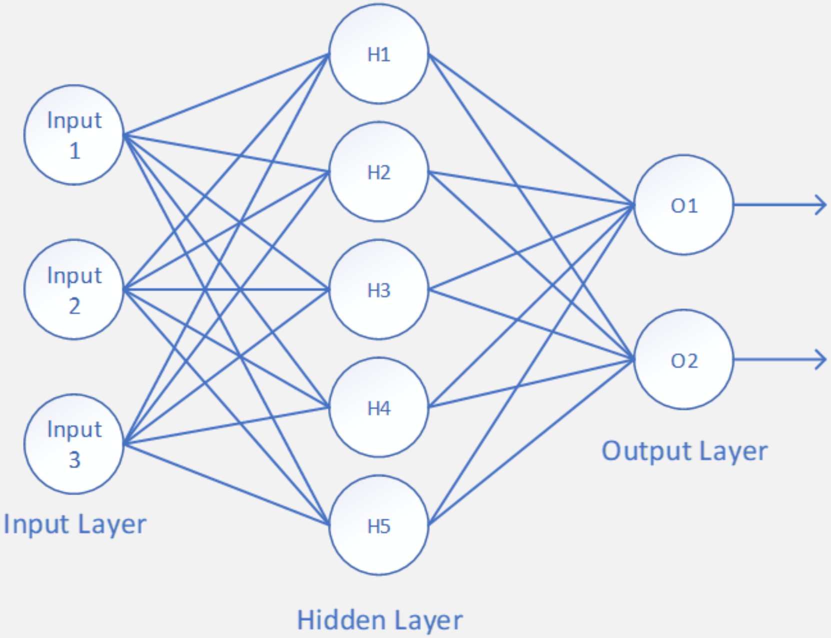
For example, if someone wishes to illegally access a companies internal network and the resources contained within, a VPN (or multiple VPNs) can be used to hide both their identity and their real location.

Tracking down criminals who do make use of VPNs to hide their identity can prove to be challenging if not completely impossible.

Attacks such as the Sony Pictures incident from 2014 or the LinkedIn breach from 2012 are examples of the kind of attack that could benefit from the use of VPNs.

Solution

Incorporate Machine Learning to detect VPN use



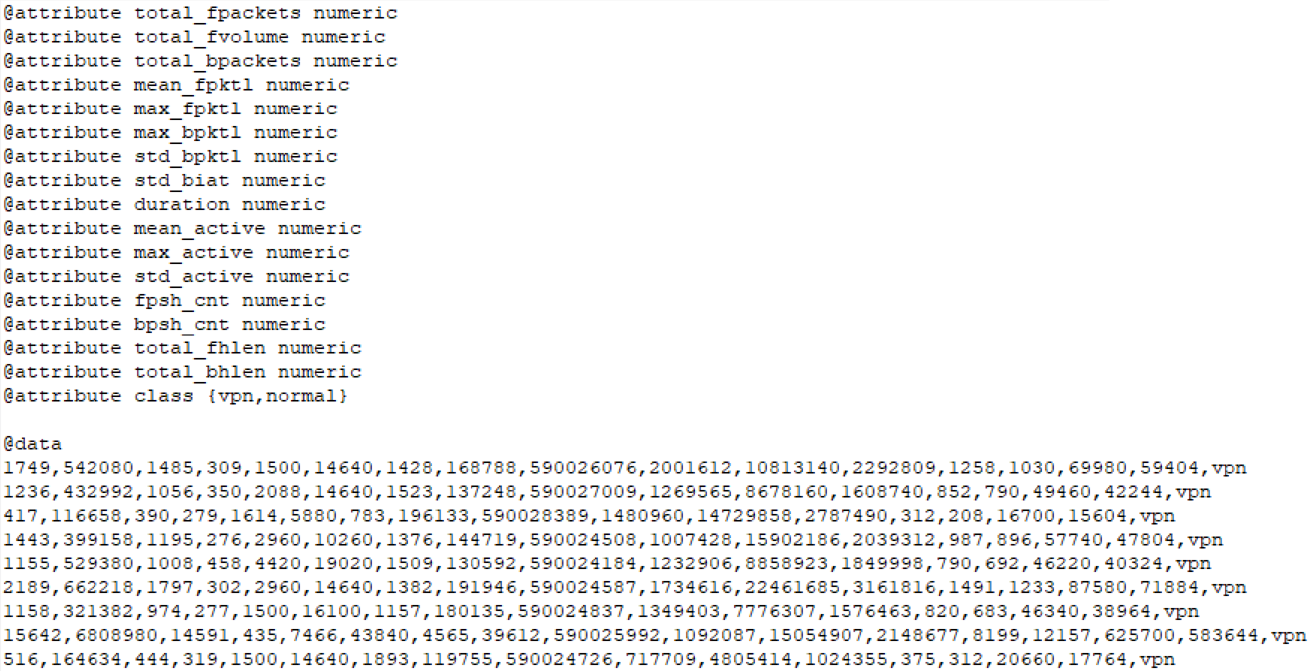
A possible solution is the use of machine learning to detect VPN network traffic.

The experiment here aims to test the ability of a multi-layered perceptron neural network in detecting VPN traffic amongst normal traffic.

The neural network will be trained using TCP flow statistics gathered from both VPN and non-VPN traffic.

Ulster University

Dataset Capture



```
@attribute total_fpackets numeric
@attribute total_fvolume numeric
@attribute total_bpackets numeric
@attribute mean_fpktl numeric
@attribute max_fpktl numeric
@attribute max_bpktl numeric
@attribute std_bpktl numeric
@attribute std_biat numeric
@attribute duration numeric
@attribute mean_active numeric
@attribute max_active numeric
@attribute std_active numeric
@attribute fpsh_cnt numeric
@attribute bpsh_cnt numeric
@attribute total_fhlen numeric
@attribute total_bhlen numeric
@attribute class {vpn,normal}

@data
1749,542080,1485,309,1500,14640,1428,168788,590026076,2001612,10813140,2292809,1258,1030,69980,59404,vpn
1236,432992,1056,350,2088,14640,1523,137248,590027009,1269565,8678160,1608740,852,790,49460,42244,vpn
417,116658,390,279,1614,5880,783,196133,590028389,1480960,14729858,2787490,312,208,16700,15604,vpn
1443,399158,1195,276,2960,10260,1376,144719,590024508,1007428,15902186,2039312,987,896,57740,47804,vpn
1155,529380,1008,458,4420,19020,1509,130592,590024184,1232906,8858923,1849998,790,692,46220,40324,vpn
2189,662218,1797,302,2960,14640,1382,191946,590024587,1734616,22461685,3161816,1491,1233,87580,71884,vpn
1158,321382,974,277,1500,16100,1157,180135,590024837,1349403,7776307,1576463,820,683,46340,38964,vpn
15642,6808980,14591,435,7466,43840,4565,39612,590025992,1092087,15054907,2148677,8199,12157,625700,583644,vpn
516,164634,444,319,1500,14640,1893,119755,590024726,717709,4805414,1024355,375,312,20660,17764,vpn
```

To generate OpenVPN network traffic, a VPN server was setup up using a mixture of AWS and DigitalOcean.

This VPN was connected to using an Ubuntu 16.04 VM which was running a browsing script coded in python.

Packets were captured by wireshark and then processed by NetMate in order to gather flow statistics.

Dataset (cont.)

To acquire non-VPN traffic, the same capture settings that were used to capture VPN traffic were used, except the VPN was not connected.

The non-VPN traffic was labelled as “normal” and passed through NetMate as well.

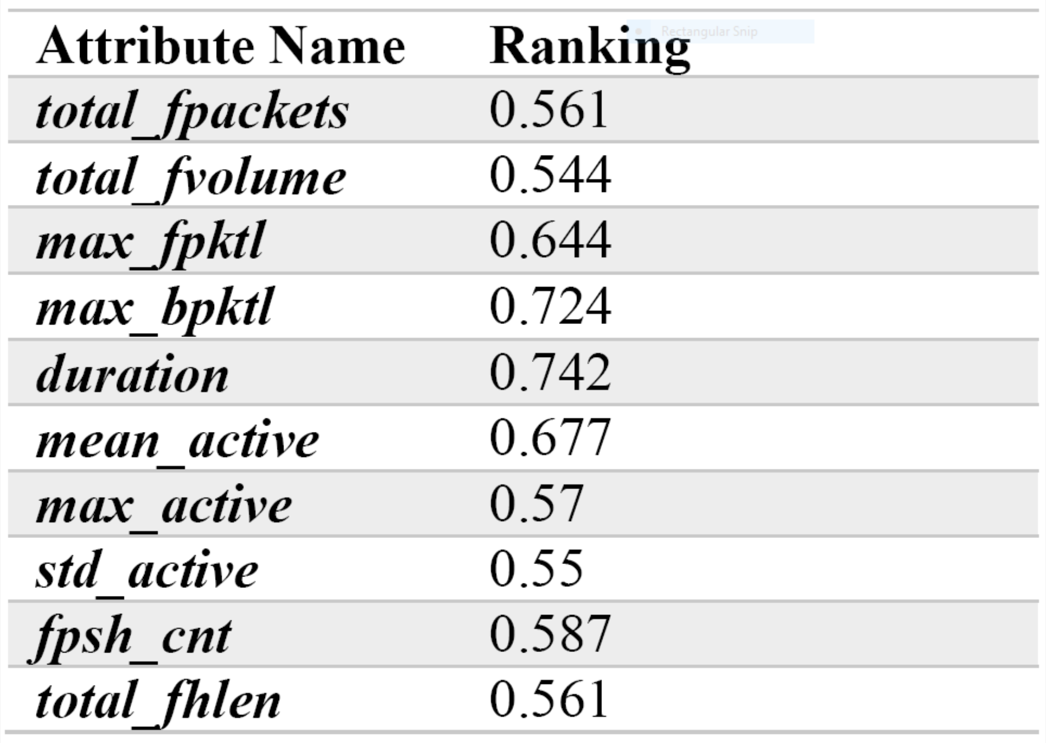
Once complete, the dataset was split into Training, Testing and Validation sets.

Training set = 7862

Test set = 1257

Validation set = 253

Feature Selection



Attribute Name	Ranking
total_fpackets	0.561
total_fvolume	0.544
max_fpktl	0.644
max_bpktl	0.724
duration	0.742
mean_active	0.677
max_active	0.57
std_active	0.55
fpsh_cnt	0.587
total_fhlen	0.561

Feature selection was applied to the original dataset to bring the number of features down from 44 features.

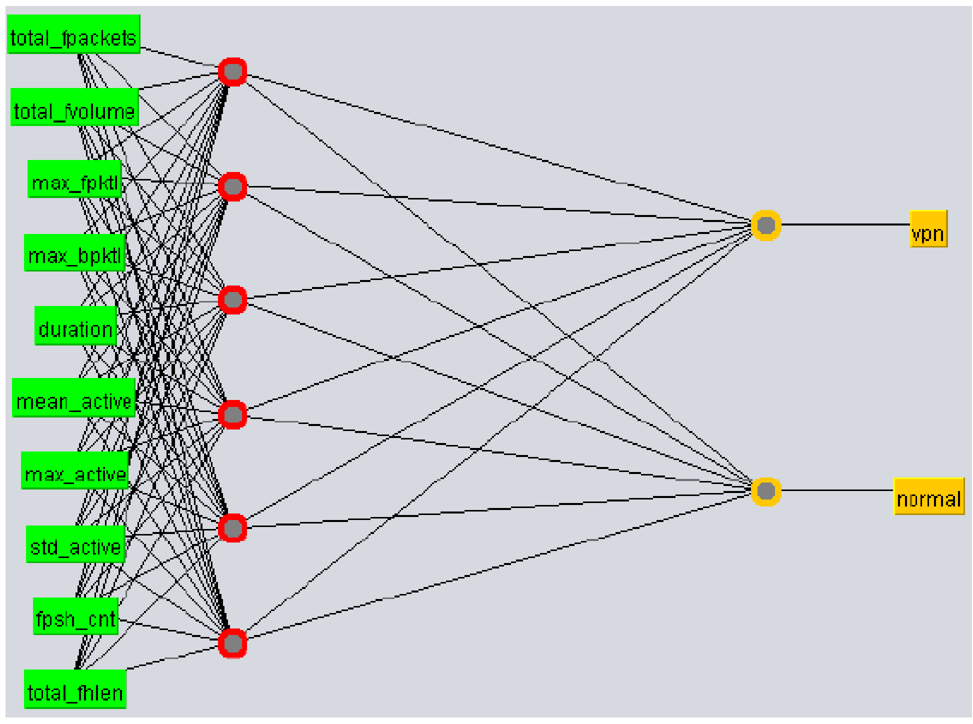
The Weka model *CorrelationAttributeEval* was used to find the correlation coefficient of each feature.

Any that had a coefficient at or above 0.5 were included in the dataset. This resulted in 10 total features for the dataset.

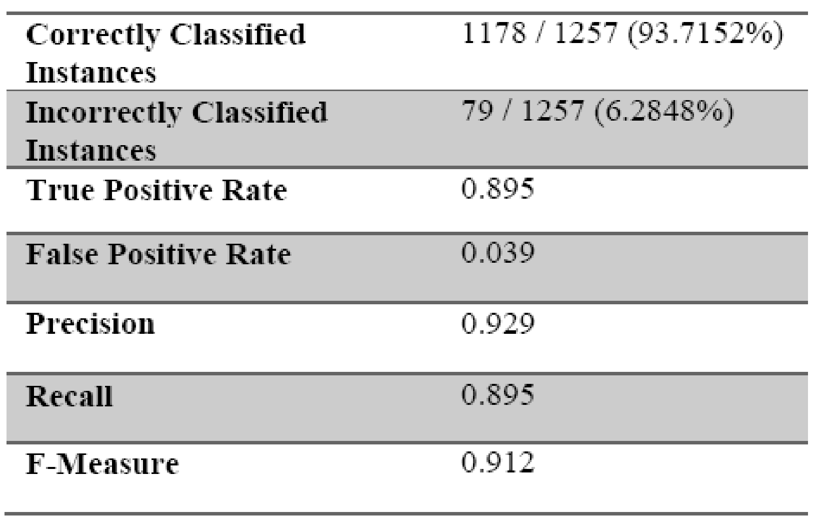
Weka Experiment

Weka was used for both processing the dataset and training of the neural network.

Neural network model was based on weka’s *MultilaterPerceptron* model*.* It is trained over 1000 instances on the aforementioned dataset

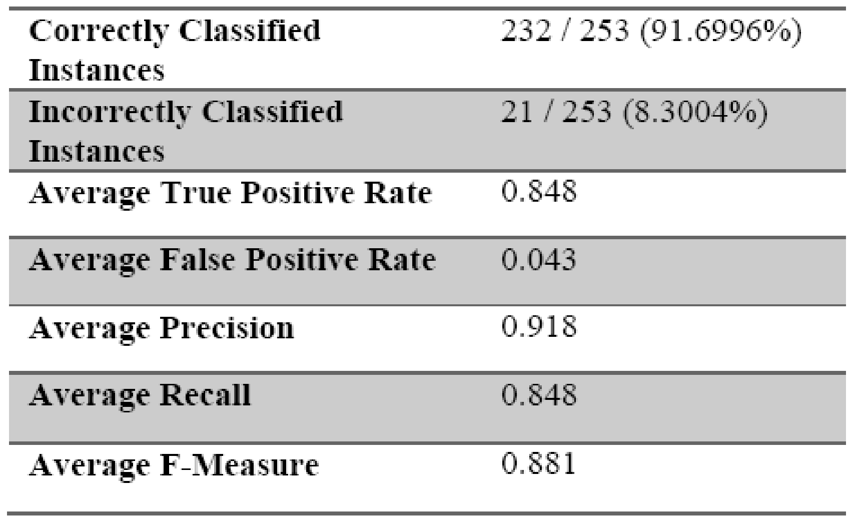


Results



Correctly Classified Instances	1178 / 1257 (93.7152%)
Incorrectly Classified Instances	79 / 1257 (6.2848%)
True Positive Rate	0.895
False Positive Rate	0.039
Precision	0.929
Recall	0.895
F-Measure	0.912

Validation Test



Correctly Classified Instances	232 / 253 (91.6996%)
Incorrectly Classified Instances	21 / 253 (8.3004%)
Average True Positive Rate	0.848
Average False Positive Rate	0.043
Average Precision	0.918
Average Recall	0.848
Average F-Measure	0.881

Training Test

Future Work

- Investigate the usefulness of this neural network for other VPN protocols.
- Investigate other machine learning techniques to compare against the machine learning results.
- Investigate different types of internet traffic other than just web traffic.

Ulster University

Question Time!

ulster.ac.uk