Project 2: Neural networks in intrusion detection design

Derrick Kempster
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
dk9977@rit.edu

***Abstract*— An IDS is an important piece of software in the security of many computer systems. This project is to use neural networks to create the classifiers for some misuse IDS and an anomaly IDS. To do this, Weka was employed for its ability to preprocess, classify, and analyze data. The provided network traffic data was preprocessed into several training and test sets to identify 5 attacks and normal activity individually. A multilayer perceptron was created for each of the training sets. Running the test sets through the classifiers showed a minimum accuracy of 99.6956%. Thus, Weka has shown to be quite a useful tool in classifier creation.**

***Keywords*— IDS, neural network, misuse, anomaly, Weka, classifier, multilayer perceptron**

I. Specification

An IDS – intrusion detection system – is a piece of security software that monitors the traffic on a computer system. It is intended to detect and alert about suspicious activity from the network. The goal of this project is to create classifiers for some misuse IDS and an anomaly IDS. A misuse IDS focuses on identifying the network activity the attack it represents or normal. An anomaly IDS doesn't identify by signature, but instead classifies activity as normal or anomalous. To achieve IDS creation, an artificial intelligence program called Weka – short for the Waikato Environment for Knowledge Analysis – was employed. The tool assists with data preprocessing, classification, and analysis. The data used by an IDS consists of all available information in connections to a system. The information from a single connection composes a single record.

II. Methods and Techniques

The data of choice was provided by Professor Leonid Reznik. It was already normalized and put in separate files by the known associated attack. In order to change the data to a format acceptable by Weka, two programs were written: one for the multiple misuse IDS and another for an anomaly IDS. In each, all of the provided files were combined into one Attribute-Relation File Format (.arff) file and each instance gained a "class" attribute. To discover misuse, this new attribute is the type of attack to identify or "Other" for normal behavior or other attack behavior. To discover anomalies, it is "Normal" for normal behavior or "Anomaly" for any attack. There is a total of 823367 instances in each of these files.

In order to appropriately test accuracy of the multilayer perceptron classifiers, there should be a significant number of positive examples in the test set. Among the data, aside from the normal behavior set, the data sets with the largest number of entries are for IPSweep, Neptune, PortSweep, Satan, and Smurf. Thus, these five attacks were chosen for the set of misuse classifiers. The preprocessing program was run for each of the attacks to generate full data sets in the appropriate Weka format.

Then, this data was divided into training and testing sets. In the "Preprocess" section of the Weka Explorer, a filter of

Resample –S 378 –Z 70.0 –no-replacement (1)

was applied to create and save a 70% training set (576356 instances). This filter means that the randomization has a seed of 378, select a number of entries equaling 70% of this set, and never reselect an entry to join the set again. By undoing that filter and applying a slightly different filter of

Resample –S 378 –Z 70.0 –no-replacement –V (2)

the inverse set (247011 instances) was created and then saved as the test set.

III. Implementation

In the "Preprocess" section of the Weka Explorer, the training set was loaded. Then in the "Classify" section, a classifier was trained and tested using a supplied test set – the one created before. Using the settings

MultilayerPerceptron –L 0.3 –M 0.2 –N 500
–V 0 –S 0 –E 20 –H 1 (3)

a multilayer perceptron was created. Specifically, it had a learning weight of 0.3, a momentum of 0.2, a training time of 500 epochs, no validation set, a seed of 0, and 1 hidden layer. Table I shows the building times of each multilayer perceptron.

TABLE I. Classification Time

IDS Type	Running Time (s)	
	Building the model	***Testing the model***
IPSweep	645.37	1.29
Neptune	641.42	1.31
PortSweep	642.36	1.27
Satan	656.95	1.26
Smurf	631.50	1.28
Anomaly	632.30	1.21

IV. Tests

After the model was constructed, the test set immediately ran. Each of the instances in the test set were run through the classifier to try to determine its identity. Table I also shows the testing times of each multilayer perceptron.

V. Results

Table II shows the overall accuracy of each classifier. Each classifier has shown to be quite good, with accuracy very near to 100%. The least accurate classifier – for identifying anomalies – was still at 99.6956%.

Tables III through VII show the results of the misuse IDS testing. Table VIII shows the same for anomaly IDS testing. In the Tables III through VIII, "False (%)" represents the portion of instances of that type which have been classified incorrectly. False positive ratios are shown on the rows with the class of the table's name; false negative ratios are shown on the "Other" or "Normal" line. These false positive and false negative ratios have also been compiled into Fig. 1, shown logarithmically.

It is often more important for the false negative ratio to be smaller than the false positive ratio for security purposes. It makes more sense to claim that there is a problem when there is none than to not find the problem. The only sets of results with a greater false positive ratio are in Tables IV and VIII, but in both cases, the ratios are still sufficiently low. Table III holds the largest false positive ratio of nearly 23%, but could still be considered reasonable.

TABLE II. OVERALL ACCURACY

Classifier	Accuracy (%)
IPSweep	99.8652
Neptune	99.9976
PortSweep	99.9899
Satan	99.9721
Smurf	99.9895
Anomaly	99.6956

TABLE III. IPSWEEP RESULTS

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

TABLE IV. NEPTUNE RESULTS

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

TABLE V. PORTSWEEP RESULTS

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

TABLE VI. SATAN RESULTS

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

TABLE VII. SMURF RESULTS

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

TABLE VIII. ANOMALY RESULTS

Determined Class	Actual Class		Class Acc. (%)
	Anomaly	Normal	
Anomaly	73391	128	0.1741
Normal	624	172868	0.3597

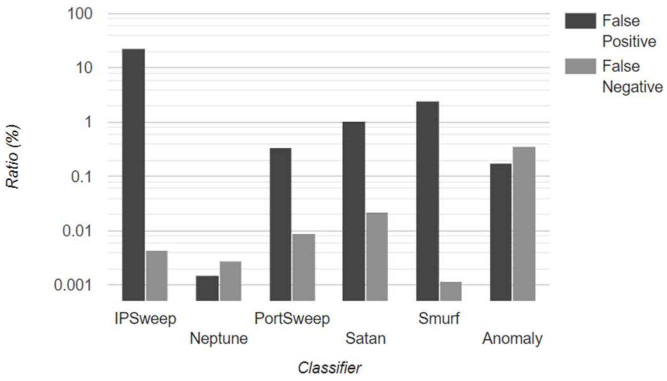

Fig. 1. Logarithmic graph of the false positive and false negative ratios of each classifier