Network Traffic Classification with Machine Learning for Network Intrusion Detection Systems

Marvin Newlin
Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH, USA
marvin.newlin@afit.edu

***Abstract*—TODO**

***Index Terms*—Machine Learning, Intrusion Detection Systems**

I. Introduction

Network Intrusion Detection Systems (NIDS) are an important aspect of cybersecurity in modern networks. One important task of many NIDS is being able to detect malicious traffic entering or exiting the network among the benign traffic. NIDS are a critical aspect to network security to prevent systems from being infected by malware or subjected to attack [1]. An important area of research for NIDS traffic classification is the utilization of Machine Learning (ML) methods for performing this traffic classification.

Intrusion detection systems analyze traffic

The dataset being utilized is the CICIDS2017 dataset presented by Sharafaldin et al. in [2]. This dataset is in the form of a network packet capture and contains both legitimate and malicious traffic. There are many features within the dataset like the standard Source/Destination IP address, Source/Destination port number, timestamp, and protocol. Each of the entries in this dataset is keyed as a network flow. The dataset also contains statistical features of each labelled flow. A subset of these statistical features will be the focus of the task of classifying network traffic as benign or malicious. The features being utilized are the following:

- Flow duration (seconds)
- Flow bytes per second
- Number of forward packets sent
- Number of backward packets sent
- Forward Packet Length Mean
- Backward Packet Length Mean
- Number of Forward Packets per Second
- Number of Backward Packets per Second

The problem of traffic classification presents two research questions (RQ).

1) *RQ1*: Can we successfully classify traffic as benign or malicious using the features from the list above?
2) *RQ2*: What features from the list above are most important for classifying traffic as benign or malicious?

In order to answer these research questions, the formal ML task being performed is the task of classification. The ML task of classification involves predicting a qualitative response for each observation, thereby separating or classifying observations into different categories [3]. In this case, the task is binary classification since there are only two categories.

Additionally, the CICIDS2017 dataset is labelled, meaning that each observation has an assigned value (i.e. benign or malicious). Thus, our ML task of classification is a supervised classification task [3].

[TODO] Overview of results

The structure of this paper is organized as follows. In section II, related works and areas of research are discussed. Section III provides more detail on the dataset, the models used, methods for selecting the models, and performance measures. Section IV details the results of the classification. Section V provides conclusions and recommendations for future work.

II. Related Work

[3] is an excellent text on methods and techniques for statistical learning. The textbook provides in depth coverage of introductory ML concepts such as supervised classification, regression, feature selection techniques, and unsupervised techniques as well.

Generally, supervised classification is a task that involves a qualitative prediction based on input features. To perform the task, we select a model, train the model using training data which is labelled in the supervised case, and then test the performance on previously unseen test data [3]. There are many different decisions to be made along the way with classification as with all other ML tasks. These steps and decisions are discussed in depth in section IV.

A. Supervised Traffic Classification

Along with generating the CICIDS2017 dataset, [2] performed traffic classification on the dataset along with feature selection. They utilized several techniques including K-Nearest Neighbors (KNN), Random Forest (RF), Quadratic Discriminant Analysis (QDA), and several others. In addition to classification they also analyzed the important features for detecting the different attacks within the dataset. With several of the algorithms they applied they achieved high accuracy, precision, and recall. However, with feature selection, they did not evaluate the significance of the weights of the selected features and in many cases the weights were almost zero which made it difficult to interpret the value of the selected features. Additionally, the CICIDS2017 dataset is severely imbalanced between benign traffic and the different types of malicious traffic that they classify on. Thus, the resulting accuracy values most likely reflect the underlying data distribution.

B. Unsupervised Traffic Classification

Mukkamala et al. [4] explored utilizing neural networks and Support Vector Machines (SVM) for NIDS. They utilized the KDD-CUP 1999 dataset and developed both a neural network based IDS and an SVM IDS. With both of these systems, they were able to achieve high classification accuracy, over 99% accuracy on the test set. However, they do not discuss any other metrics such as precision, recall, or F1 measure. Additionally, they do not discuss any of the pertinent features involved in the classification.

Chitrakar and Chuanhe [5] explored utilizing k-medioids clustering and naive Bayes classification for IDS traffic classification. This method uses the k-medoids clustering algorithm to cluster the traffic based on similarity metrics and then performs the classification with a naive Bayes classifier. Overall, they achieved high accuracy with this classification method and had a low false positive rate and a high true positive rate. However, they also do not discuss any features that are important to the classification.

C. Classification on Imbalanced Datasets

The CICIDS2017 dataset contains a severe class imbalance. On the subset we explore here, there are approximately 170,000 observations of which only 2,000 are malicious. This means that 99% of the data belongs to the benign class, qualifying it as severely imbalanced [6]. The problem with imbalanced data is that the traditional accuracy metric may reflect the underlying class distribution rather than correct classifications [7]. In this section, we explore some of the work related to classification on imbalanced datasets.

Haibo and Garcia in [6] explored the problem of learning from imbalanced data in the binary case. They discuss the ideas of random oversampling and undersampling. Random oversampling is done by sampling from the minority class of the dataset with replacement, thereby increasing the overall size of the minority class by the number of samples [6]. Conversely, undersampling randomly selects a smaller number of observations from the majority class without replacement and then combines that with the original minority class to increase the percentage of the minority class [6].

Some problems occur with these methods however. Oversampling leads to multiple copies of observations from the minority class in the data, thus tying the observations together and this can lead to overfitting [6]. Similarly, with undersampling, data that could be important is removed, leading to reduced performance [7].

Batista et al. [7] discuss some heuristic methods that can be used to improve classification on imbalanced datasets. One of the heuristic methods they discuss is the One Sided Selection (OSS) heuristic. This heuristic removes borderline examples from the the majority class while maintaining the entire minority class, thus creating better separation between the classes [7]. Additionally, they discuss the Neighborhood Cleaning Rule heuristic. The way it works is it finds the three nearest neighbors of a given observation using KNN with k=3. If the observation belongs to the majority class and the predicted class based on KNN is not the majority class, then that observation is removed. If the observation belongs to the minority class and the results of KNN misclassify it as the majority class, then the three nearest neighbors are removed [7].

Moving forward with the CICIDS2017 dataset, we will be utilizing methods to overcome the imbalance in the data to produce a classifier that performs well. As we have discussed, accuracy alone is not a good performance measure for classifiers, especially with imbalanced datasets and we will discuss the methods used in the following section.

III. Methodology

In this section, we discuss in detail the dataset, model fitting decisions, model evaluation decisions, and the analysis strategy for the classification task.

A. Dataset Details

For this work, we will be using a subset of the CICIDS2017 dataset. As described earlier, this dataset was synthetically generated via simulation in a sandbox and contains both benign and malicious traffic. The original dataset contains examples of Distributed Denial of Service (DDoS), Port Scanning, Web Attacks, Infiltration, Heartbleed, and other types of malicious traffic. Due to the large size of the total dataset, for this work, we selected the subset of the dataset that contained only benign traffic and the web attacks. This subset allows us to perform a binary classification task between benign traffic and the malicious web attack traffic. For the rest of this work, unless specifically mentioned, references to the dataset refer to this subset and not the entire CICIDS2017 dataset.

The dataset comes in a comma separated value (CSV) file and as such, is easily imported in code. The dataset itself contains about 80 features that are both packet level features (IP Addresses, port numbers, etc.) and statistical features like the ones listed in Table I. Table I includes the features that were selected from the original features. Since these selected features are all statistical features, they are real valued inputs and the range for each feature is displayed in Table II. The dataset contains approximately 170,000 observations, of which 2,000 are labelled as malicious, leaving about 168,000 observations belonging to the benign class. This means that the distribution of the classes is 99% benign and 1% malicious traffic. This is a very imbalanced dataset, so in a later section we will discuss our strategy for handling this imbalance.

Due to the wide separation of each range of values for each feature from Table II, we have performed standard scaling using the scikit-learn Standard Scaler normalizing all values so as not to negatively impact the classification. Additionally, we have changed the labels on the label column to be a 1 for

TABLE I
FEATURES USED FOR CLASSIFICATION AND THEIR DATA TYPE

Feature	Data Format
Flow duration (seconds)	float64
Flow bytes/second	float64
Number of forward packets sent	float64
Number of backward packets sent	float64
Forward Packet Length Mean (bytes)	float64
Backward Packet Length Mean (bytes)	float64
Number of Forward Packets per Second	float64
Number of Backward Packets per Second	float64

TABLE II
VALUE RANGES OF NON-TEST DATA FOR THE SELECTED FEATURES

Feature	Data Range
Flow duration (seconds)	$[0, 1.2 \times 10^8]$
Flow bytes/second	$[-2.61 \times 10^8, 2.07 \times 10^9]$
Number of forward packets sent	$[1, 200755]$
Number of backward packets sent	$[0, 270686]$
Forward Packet Length Mean (bytes)	$[0, 4183]$
Backward Packet Length Mean (bytes)	$[0, 3495]$
Number of Forward Packets per Second	$[0, 3 \times 10^6]$
Number of Backward Packets per Second	$[0, 2 \times 10^6]$

the benign class and a -1 for the malicious class so that the data frame contains all numerical values.

Exploring the dataset is made difficult due to the high imbalance between classes and large number of data points. However, based on the scatter matrix we have selected three scatterplots to attempt to show some of the class distributions. To generate these scatterplots, we selected 5% of the data points from each class at random to display in the plot. For better interpretation, the points in the plot are not scaled and represent the original units.

In Figure 1 we see a distinct separation between the majority of the benign traffic and a few of the malicious points. However, near the (0,0) point, there is a large cluster of malicious points there as well. In Figure 2 we see a larger distribution of the benign traffic with some malicious points scattered around. In Figure 3 we also see a larger distribution of the benign traffic and two main clusters of malicious traffic.

The main thing to note is that there do not appear to be any distinct boundaries between the malicious and benign traffic. This indicates that the more flexible models may perform better at this classification task.

B. Model Fitting

The ML task being performed is a supervised binary classification task. We have split the data into non-test and test sets, with $\frac{2}{3}$ of the data as the non-test set and $\frac{1}{3}$ of the data as the test set using the scikit-learn `train_test_split` function. For classification models we evaluate Logistic Regression (LR), Linear Discriminant Analysis, Quadratic Discriminant Analysis, and K-Nearest Neighbors. For all model tuning and hyper-parameter decisions we use 10-fold cross validation to select the best model performance on the non-test set which we then evaluate on the test set.

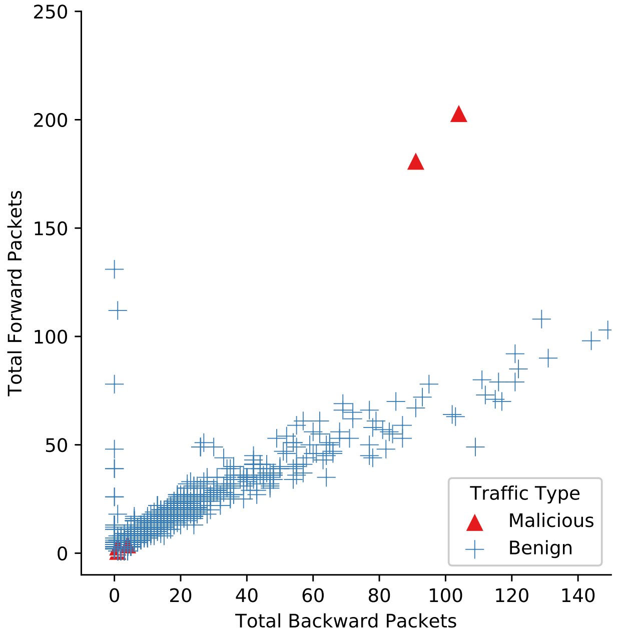

Fig. 1. Scatterplot of number of packets sent in each direction.

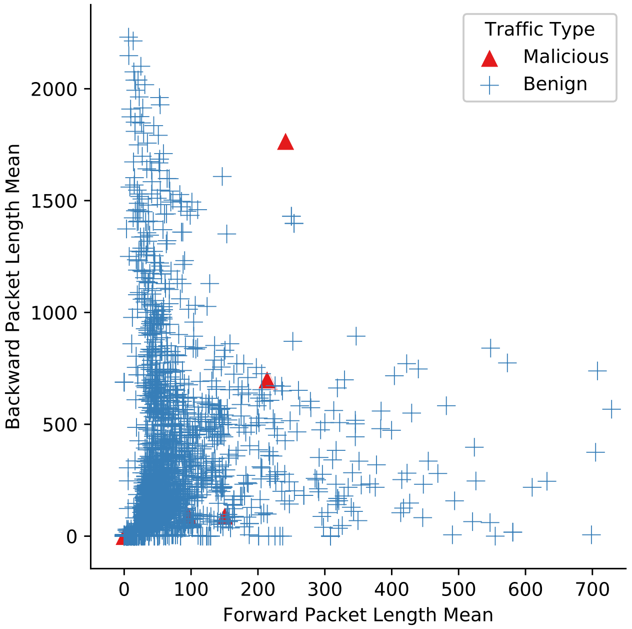

Fig. 2. Scatterplot of Packet Length Mean in each direction.

For performance evaluation, since we have an imbalanced dataset we will be utilizing the scikit-learn `balanced_accuracy_score` function [1]. This function reports the average of the recall for each class and is equivalent to normal accuracy with class balanced weights [8], [9].

[1] scikit-learn Balanced Accuracy Score - https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html

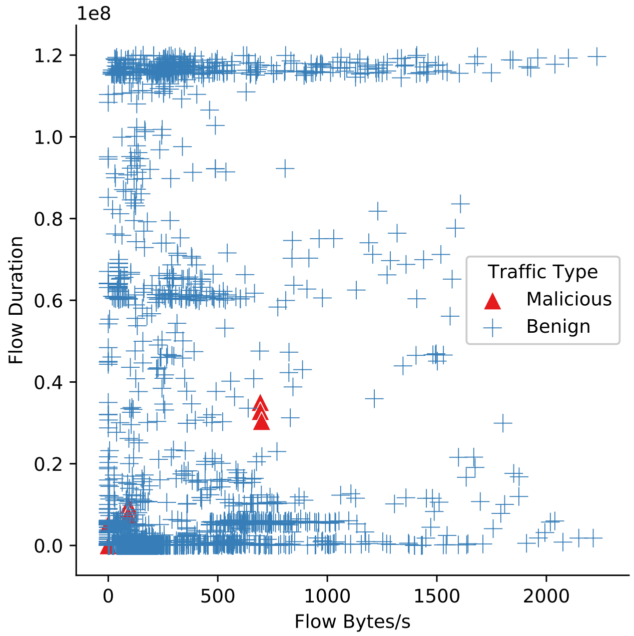

Fig. 3. Scatterplot of Flow Bytes/s vs. Flow Duration.

For model evaluation, the scikit-learn classification functions support class weights in order to balance the classification [2]. Additionally, to answer RQ2, we utilize Forward Stepwise Subset Selection to determine the features that are most important in the classification process.

C. Model Evaluation

Since we will are comparing the performance of several different models, we use cross validation to make the selection of the best performing model. For evaluating LR, LDA, and QDA, we compare the average cross validation balanced accuracy score. For KNN, we perform a hyper-parameter sweep using cross validation to determine the best value of k by selecting the k for which the average cross validation balanced accuracy score is minimized for the model. With the chosen value of k we then compare its average cross validation balanced accuracy score with the other models to select the best classifier with respect to the balanced accuracy score. After choosing this best classifier we retrain the selected model on the entire non-test set before evaluating test set performance. Additionally, for feature selection, we will run a hyper-parameter sweep to find the best value of alpha for the LASSO model to determine the features that affect the classification.

We also utilize typical classification evaluation metrics. We use the Receiver Operator Characteristic (ROC) curve and the Area Under the Curve (AUC) metric to determine the informedness of the classifiers. We also leverage confusion matrices and the precision and recall metrics to ensure that our classification is correctly classifying the data and not just reflecting the underlying data distribution. In addition, using the ROC curve we are able to discuss customization of sensitivity and specificity for detection purposes.

IV. Results

TODO: Results section

V. Conclusions and Future Work

TODO: Conclusions

References

[1] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7307098/

[2] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006639801080116

[3] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. New York, NY, USA: Springer, 2013.

[4] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *Proceedings of the 2002 International Joint Conference on Neural Networks*. IEEE, 2002.

[5] R. Chitrakar and H. Chuanhe, "Anomaly based Intrusion Detection using Hybrid Learning Approach of combining k-Medoids Clustering and Naïve Bayes Classification," in *8th International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, China, 2012. [Online]. Available: http://resolver.ebscohost.com.afit.idm.oclc.org/openurl?sid=EBSCO:edseee&genre=book&issn=21619646&ISBN=9781612846842&volume=&issue=&date=&spage=1&pages=1-5&title=2012%208th%20International%20Conference%20on%20Wireless%20Communications,%20Networking%20and%

[6] Haibo He and E. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 9 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5128907/

[7] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, 2004. [Online]. Available: https://dl-acm-org.afit.idm.oclc.org/citation.cfm?id=1007735

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/index.html

[9] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The Balanced Accuracy and Its Posterior Distribution," in *2010 20th International Conference on Pattern Recognition*. IEEE, 8 2010, pp. 3121–3124. [Online]. Available: http://ieeexplore.ieee.org/document/5597285/

[2] scikit-learn Class Weights - https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html