# Building Multiclass Classification Baselines for Anomaly-based Network Intrusion Detection Systems

****
*Department of Mathematics and Computer Science*
***** University*
****, US
****@****.edu

****
*Department of Mathematics and Computer Science*
***** University*
****, US
****@****.edu

****
*Department of Mathematics and Computer Science*
***** University*
****, US
****@****.edu

****
*Department of Mathematics and Computer Science*
***** University*
****, US
****@****.edu

***Abstract*—This paper showcases multiclass classification baselines using different machine learning algorithms and neural networks for distinguishing legitimate network traffic from direct and obfuscated network intrusions. This research derives its baselines from Advanced Security Network Metrics & Tunneling Obfuscations dataset. The dataset captured legitimate and obfuscated malicious TCP communications on selected vulnerable network services. The multiclass classification NIDS is able to distinguish obfuscated and direct network intrusion with up to 95% accuracy.**

***Keywords—Network Intrusion Detection System, Signature-based Intrusion Detection System, Anomaly-based intrusion detection system, multiclass classification.***

## I. Introduction

Network intrusion detection systems are deployed to help organizations find suspected network attacks and prevent them from being executed. One of the most common types of NIDS is signature-based intrusion detection systems (SIDS), which compares network traffic packets signatures to malicious network traffic packet signatures, from a database, and creates alerts on matches [9]. While constructing a signature-based intrusion detection system can be helpful for distinguishing predefined network attacks, it may not be able to alert the latest network attack[4]. This issue can be solved by updating the malicious traffic packet signature database on a frequent basis. However, this can be a very costly operation. An anomaly-based intrusion detection system (AIDS) compares network traffic to a baseline and alerts whenever the traffic is behaving unlike the establish baseline[5]. This research constructs an anomaly-based network intrusion detection system to not only distinguish but to also determine whether a network intrusion was obfuscated.

## II. Gathering the data

For the demonstration, this research used the Advanced Security Network Metrics & Tunneling Obfuscations dataset for the training data. The dataset gathered data by monitoring network traffic on devices which had existing vulnerable versions of services such as Apache Tomcat and Samba. The data collected was TCP communication over HTTP and HTTPS and different attributes related to the packets over virtual network conditions. In addition, they collected TCP communication of malicious attacks in four real world slightly varying network environments.

## III. Analyzing the data

The dataset had 895 attributes, 394 observations, 100 missing cells, and 2 duplicate rows. The target attribute was identified as “label_3”, denoting whether that network traffic was flagged as legitimate, direct, or obfuscated network intrusions.

## IV. Data Preprocessing

### A. Forward Feature Selection with Recursive Feature Elimination

The preprocessing phase began by replacing all the missing attributes with their respective mean value. After which, feature selection was performed. This used a Support Vector Classifier with a linear kernel as the estimator and forward feature selection with cross validation to rank the features and recursively eliminate the unnecessary attributes. The following features were identified: PolyInd10ordOut[3], OutPktLen64s10i[8], OutPkt4s10i[7], ConTcpFinCntIn, GaussProds4In[1], FourGonAngleAllN[2], MedTCPHdrLen, GaussProds8In[5], SumTTLOut, PolyTime10ordOut[2], InPkt64s20iTr2KB[14], OutPktLen64s10i[5], PolyInd13ordIn[7], InPkt1s10i[8], OutPkt32s20iTr4KB[11], PolyTime10ordOut[8], OutPktLen4s10i[3], PolyInd13ordOut[13], PolyInd13ordIn[12], and InPkt64s20iTr2KB[7]. The descriptions are available on GitHub [7].

### B. Normalizing Input Data and Encoding Output Layer

The extracted features from the ASNM TUN dataset were normalized between the range of 0.1 to 0.9. There are three target labels that indicate whether the network packet was legitimate, direct, or an obfuscated network intrusion. These were transformed with values of 0, 1 or 2 to indicate the respective class labels and were one-hot encoded.

## V. Building the anomaly-based neural networks

This research created two baseline models, each having different optimizer functions. The neural networks were developed with the same architecture; 6 layers, 4 of which were

This research is supported by Professor **** and **** under the **** University Mathematics and Computer Science Department. All the source code is hosted on GitHub [7].

hidden layers, 1 input layer and 1 output layer. The input and hidden layers used rectified linear unit as their activation function. The input layer had a density of 20 neurons, the first, second, third and fourth hidden layers have 40, 60, 30 and 10 neurons respectively. The output layer had three neurons, to match the target attribute’s shape, and used softmax as its activation function. The first neural network used the Adam optimizer with a learning rate of 0.09. The second neural network used stochastic gradient descent with a learning rate of 0.01 and momentum of .75. The models’ loss function was set to categorical cross entropy.

## VI. Building the non-neural network anomaly-based baselines

In addition to testing the neural networks’ accuracy, this research developed baselines using the following classification algorithms: Decision Tree, k-Nearest Neighbors, Random Forest, and Support Vector Machines.

## VII. Results and Evaluation

This research used 5-fold cross validation to evaluate the performance of the models. The baseline models were trained to each fold’s training data and target labels and was evaluated with their corresponding testing data and target labels. For both neural networks, the batch sizes ranged from 10 to 64 per iteration and the number of epochs ranged from 10 to 1200.

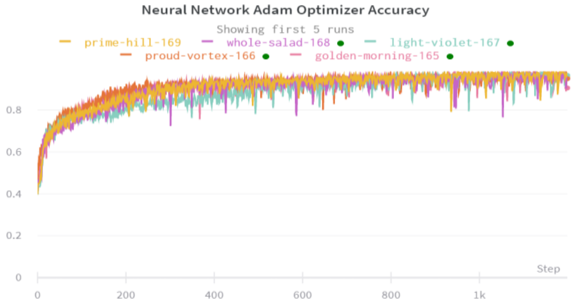

Fig. 1. Line plot graph of each model using Adam optimizer’s accuracy

During the training process, the initial accuracy for each model was less than 50% for the first 100 epochs. However, after approximately 400 epochs, the accuracy, for all the models, leveled off at around 85% with the best model having approximately 95% accuracy.

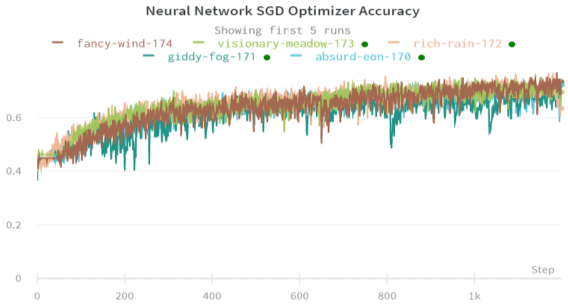

Fig. 2. Line plot graph of each model using SGD optimizer’s accuracy

The initial loss was very high but as the epochs progressed, the loss gradually declined to around 0-10% with quite several spikes for the model using Adam optimizer. This indicated the need for hyper-parameter optimization which will be the focus of future work. The Decision Tree model had up to an average of 93.67% accuracy having 100% of predicting obfuscated attacks, 76% predicting direct attacks, and 97% predicting legitimate traffic; however, these models are prone to overfitting [8]. The KNN model had up to an average of 91.14% accuracy with 91% precision predicting legitimate traffic, 92% predicting direct attacks, and 91% predicting obfuscated. The Random Forest model achieved up to 92.41% average accuracy with precision predicting 100% legitimate traffic, 92% predicting direct attacks, and 100% predicting obfuscated attacks. The SVC baseline achieved up to an average of 19.23% accuracy with its precision predictions being 0% legitimate, 100% direct attacks, and 82% obfuscated attacks.

## VIII. Conclusion

As cyber security threats become more advanced, where new exploits and network attack vectors are being discovered frequently, it is important to detect network intrusions before serious damage occurs to an organization. This research demonstrated the use of classification algorithms to detect obfuscated techniques that are used in network intrusion. The neural network approach achieved 81.73%±6.32% accuracy; with the highest being 95%. By showcasing these neural network baselines, this research hopes that NIDS will include proactive anomaly-based detection. Future work will perform hyperparameter optimization to improve the architectural design and prediction accuracy.

## References

[1] Ivan Homoliak, Petr Hanacek, "ASNM Datasets: A Collection of Network Traffic Data for Testing of Adversarial Classifiers and Network Intrusion Detectors", IEEE Dataport, 2019. [Online].

[2] HOMOLIAK Ivan, BARABAS Maros, CHMELAR Petr, DROZD Michal a HANACEK Petr.: ASNM: Advanced Security Network Metrics for Attack Vector Description. In: Proceedings of the 2013 International Conference on Security & Management. Las Vegas: Computer Science Research, Education, and Applications Press, 2013, s. 350-358. ISBN 1-60132-259-3

[3] Homoliak, Ivan & Barabas, Maroš & Chmelar, Petr & Drozd, Michal & Hanacek, Petr. (2013). ASNM: Advanced Security Network Metrics for Attack Vector Description.

[4] C. Sinclair, L. Pierce, and S. Matzner, “An application of machine learning to network intrusion detection,” Proceedings 15th Annual Computer Security Applications Conference (ACSAC99).

[5] K. Limthong, “Real-Time Computer Network Anomaly Detection Using Machine Learning Techniques,” Journal of Advances in Computer Networks, pp. 1–5, 2013.

[6] I. Homoliak. Intrusion detection in network traffic. PhD thesis, Dissertation, Faculty of Information Technology, University of

[7] Multiclass Classification Baselines for Anomaly-based Network Intrusion Detection Systems (2020), GitHub repository, https://github.com/SHU-ML-NIDS-Research/Multiclass-Classification-Baselines-NIDS

[8] N. Liberman, *Decision Trees and Random Forests*, 21-May-2020. [Online]. Available: https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991. [Accessed: 03-Jul-2020].

[9] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, 2019.