GEORGE WASHINGTON UNIVERSITY

ADA UNIVERSITY

COMPUTER SCIENCE AND DATA ANALYTICS

GUIDED RESEARCH I

Report 3

Asiman Mammadzada

Title: Network Intrusion Detection System using Machine Learning

Instructors: Prof. Dr. Stephen Kaisler, Assoc. Prof Dr. Jamaladdin Hasanov

Measurement and Scale

In this project, I research the application of Machine Learning in Network Intrusion Detection System. The main objective of the research is to investigate the rule based NIDS that can be replaced by ML algorithms, mainly tree-based. Beside, evaluation of the result as well as comparison of variety metrics in different intrusions. To accomplish the quantitative research, it is significant to measure the dependent variable, its occurrences and variation. In addition, the features, in other words, predictors are another standpoint in measurement of statistical analysis.

- **Dependent Variable**: As the problem is multi-class prediction, the dependent variable is the intrusion having specific types, such as: smurf, neptune, normal, back, satan, ipsweep, portsweep, warezclient, teardrop, pod, nmap.

- **Independent Features**: 'duration', 'protocol_type', 'service', 'flag', 'src_bytes','dst_bytes', 'land', 'wrong_fragment', 'urgent', 'hot', 'num_failed_logins', 'logged_in', 'lnum_compromised', 'lroot_shell', 'lsu_attempted', 'lnum_root', 'lnum_file_creations', 'lnum_shells', 'lnum_access_files', 'lnum_outbound_cmds', 'is_host_login', 'is_guest_login', 'count', 'srv_count', 'serror_rate', 'srv_serror_rate', 'rerror_rate', 'srv_rerror_rate', 'same_srv_rate', 'diff_srv_rate', 'srv_diff_host_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate', 'dst_host_rerror_rate', 'dst_host_srv_rerror_rate', 'label'.

- **Designing Experiments**: Since there is imbalanced dataset issue in terms of the labels, firstly, the dataset is splitted into several datasets where the balance is kept among different labels. For example, due to the fact that the occurrences of the following intrusions are close to each other, the separate model is experimented on the specific data containing intrusions such as back, satan, ipsweep, portsweep, warezclient, teardrop. Another model will be trained on neptune, smurf and normal intrusions whilst pod and nmap will be detected separately too.

- **Interpreting the results**: The results will be measured by different error metrics of ML, such as accuracy, precision, recall, f1-score and visualization of results in heatmap describing confusion matrix and classification report.
    - Accuracy: It measures the overall accuracy of the model's predictions and is calculated as the ratio of the number of correct predictions to the total number of predictions. It should be noted that it can be misleading if the dataset is imbalanced.

      Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)

    - Precision: It is the measure of the model's ability to correctly identify positive data out of the total instances predicted as positive. Precision worries on the quality of the positive predictions. It is calculated as true positives divided by the sum of true positives and false positives.

      Precision = True Positives / (True Positives + False Positives)

    - Recall: Being known as true positive rate, it measures the ability of the model to correctly detect positive instances out of the total actual positive ones. Recall focuses on the model's ability to detect all positive instances. It is calculated as true positives divided by the sum of true positives and false negatives.

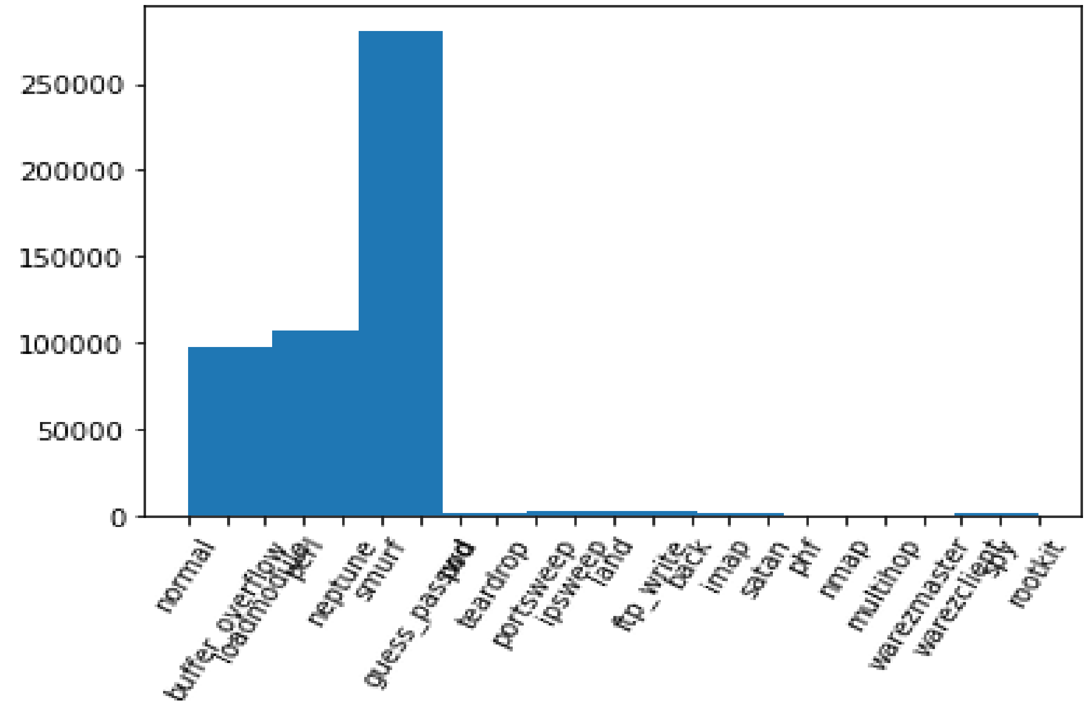
      Recall = True Positives / (True Positives + False Negatives)

    - F1 score: It is the described as combination of precision and recall and provides a balanced measure of the model's performance. F1 score considers both precision and recall, making it suitable for imbalanced datasets. It is calculated as 2 * (precision * recall) / (precision + recall).

      F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

- **Statistical Analysis:**
    - ***Feature Analysis***: I will analyze each features' importance in the dataset and assess their relevance and impact on the detection of intrusions. Beside, outliers will be detected and removed from the dataset. The z score will be 1.96 for two-tailed approach.
    - ***Measurement:*** The measurement of the model performance will be implemented as mentioned above techniques which applies precision, recall, accuracy scores of classifications.
    - ***Imbalance Distribution:*** The class imbalance distribution will be checked since majority of the NIDS datasets suffer from this. In case of found imbalance, I will apply techniques such as oversampling, undersampling, or synthetic minority oversampling technique (SMOTE) to address this issue.
    - ***Statistical Testing:*** I will do statistical tests to compare the performance of different models. For example, paired t-tests or Wilcoxon signed-rank tests will be used to compare the performance of two classifiers and find out that if there is a statistically significant difference.
- **Data Visualization:**

  I visualized the occurrences of labels to detect class imbalance:



- **Visual Storytelling**

  As the dataset does not contain close number of each label, the algorithm will be written to select labels that occurs closely. Then sub-set of datasets will be shuffled every time to train different models per intrusion.

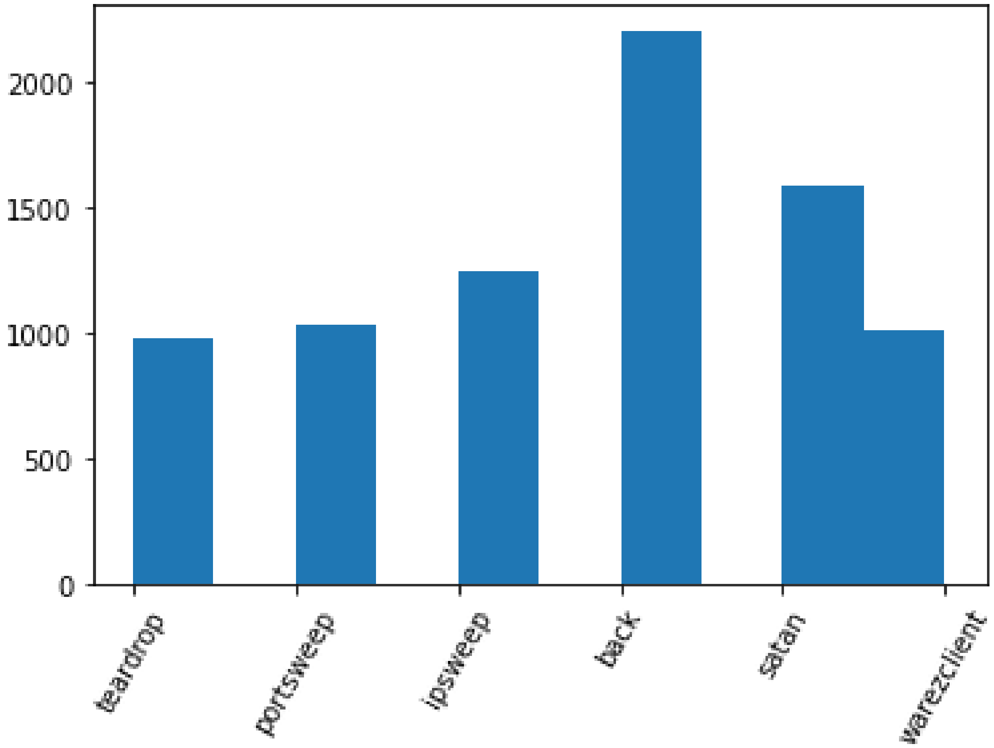


Fig 2. Closely appearing intrusions