An Experiment on Feature Selection using Logistic Regression

Raisa Islam[1], Subhasish Mazumdar[2], and Rakibul Islam[3]
Dept of Computer Science & Engineering
New Mexico Institute of Mining and Technology
Socorro, NM 87801 USA
[1]raisa.islam@student.nmt.edu, [2]subhasish.mazumdar@nmt.edu, [3]mdrakibul.islam@student.nmt.edu

Abstract

- Investigated a feature selection method using L1 and L2 regularization in logistic regression.
- Used the CIC-IDS2018 dataset to test the method, chosen for its size and challenging class separability.
- Ranked features using L1 and L2, then synthesized a feature set common to both rankings.
- Compared logistic regression models with L1 (LR+L1) and L2 (LR+L2) regularization.
- Found no significant accuracy difference between L1 and L2 once features were selected.
- Applied the synthesized feature set to complex models like Decision Trees and Random Forests.
- Despite the reduced feature set, accuracy remained high across models.
- Reported performance metrics: accuracy, precision, recall, and F1-score.

Experiment 1: Excluding one of the problematic classes.

- This experiment simplifies the classification task by removing a challenging class.
- The goal is to assess the model's performance on a less complex dataset.

Experiment 2: Including both problematic classes.

- This experiment introduces the full complexity of the dataset.
- The goal is to evaluate the model's ability to handle challenging classification scenarios.

Introduction

Method Overview: *Combines L1 and L2 regularization by ranking features separately* using *logistic regression* and *selecting features common to both sets.*
Dataset: A large, complex real-world dataset, with a particularly challenging class that is hard to distinguish from another.

Model Choice: Decision Tree for its explainability and Random Forest for higher accuracy, though less interpretable.
Results: The proposed method *reduced feature size by 72%, with only a 0.8% and 0.6% accuracy loss in Decision Trees and Random Forests*, respectively, even with the difficult class included.
Paper Structure: Includes background information (Section II), experiment details (Section III), results and analysis (Section IV), and conclusions with future research (Section V).

III. EXPERIMENT

A. Dataset and Pre-processing

1. **Dataset**:
    - **CIC-IDS2018**: A comprehensive dataset *representing network traffic observations over ten days.*
    - **Volume**: Contains *16,233,002 samples*, *distributed* across *10 CSV files.*
    - **Features**: Each sample includes *79 features* and one of *15 target classes.*
2. **Target Classes**:
    - The dataset comprises 15 distinct classes.
    - Sample sizes for classes vary significantly; for instance, the **DDOS Attack LOIC UDP** class has only **1,730 samples**.
    - Other classes with smaller sample sizes include **Brute Force XSS**, **Brute Force Web**, and **SQL Injection**.

The **CIC-IDS2018 dataset** consists of **15 classes**.

The **DDOS Attack LOIC UDP** class has **1,730 samples**, which is relatively small.

The classes for **Brute Force XSS**, **Brute Force Web**, and **SQL Injection** have significantly fewer samples: **230**, **611**, and **87** respectively.

Class imbalance is prevalent, as some classes have drastically fewer samples than others.

Most machine learning (ML) algorithms assume an **even distribution** of data across classes, making this imbalance a critical issue.

The dominance of majority classes can lead to **bias** in ML classifiers, often resulting in misclassification of minority classes.

B. Pre-processing

- In the **cleaning phase**, the following actions were taken:
    - Removed observations with feature values of **Infinity** and **NaN** (missing values).
    - Dropped **59 entries** classified as **unknown** (label was "Label").

    - Excluded the **timestamp feature** as it was deemed irrelevant, resulting in **78 usable features**.
- The goal was to extract **5,000 random samples** from each class that remained after cleaning.
    - To address the class imbalance, the classes with very small sample sizes (**Brute Force XSS**, **Brute Force Web**, and **SQL Injection**) were excluded, leaving **12 classes**.
    - Out of these, **11 classes** had **5,000 samples each**, while the **DDOS Attack LOIC UDP** class had **1,730 samples**.
- Ultimately, the dataset consisted of a total of **56,730 samples** after pre-processing.

Problematic Class Identification: The **DoS attacks-SlowHTTPTest** and **FTP-BruteForce** classes were identified as difficult to separate, with **DoS attacks-SlowHTTPTest** being specifically noted as problematic.

Experiment Design:

The experiment was divided into **two parts**:

- **Part One**: *Excluded the problematic class,* resulting in a dataset size of **51,730 samples** from **11 classes**.
- **Part Two**: *Included* the *problematic class* for *further analysis.*

target class	record count
Benign	13484708
DDOS attack-HOIC	686012
DDoS attacks-LOIC-HTTP	576191
DoS attacks-Hulk	461912
Bot	286191
FTP-BruteForce	193360
SSH-Bruteforce	187589
Infilteration	161934
DoS attacks-SlowHTTPTest	139890
DoS attacks-GoldenEye	41508
DoS attacks-Slowloris	10990
DDOS Attack LOIC UDP	1730
Brute Force Web	611
Brute Force-XSS	230
SQL Injection	87

TABLE I: target classes

The image we see is a table showing the distribution of different types of cyberattacks in a dataset. The table lists each attack type (e.g., DDoS, Brute Force) and the number of records associated with that attack type. The table also includes a "*Benign*" category, which likely *represents normal, non-malicious traffic*.

The data appears to be *heavily skewed towards benign traffic*, with *over 13 million benign* records compared to significantly smaller numbers for each attack type. This suggests that the *dataset* might be *imbalanced*, which could *pose challenges for machine learning models* trained on this data.

Conclusion

In this research, we *developed* a *logistic regression-based feature selection method* to *reduce* the feature set size for training supervised ML models. Using the CIC-IDS2018 dataset, we combined features from LR+L1 and LR+L2 methods, reducing the feature set by 72% (from 78 to 22 features) while maintaining strong model performance.

Our reduced feature set consistently outperformed subsets from individual L1 or L2 rankings and met baseline accuracy targets with fewer features. Even with complex models like Random Forest and Decision Trees, the accuracy loss was less than 1%, demonstrating the method's effectiveness. Future work will explore applying this approach to other datasets and evaluating its scalability and robustness.