AI Project Report

Network Intrusion Detection System

Andrea Mugnai
Jacopo Tucci

2024/2025

Goals

Our goal is to create a Network Intrusion Detection System (NIDS) capable of classifying raw network packets into the following categories:

- Normal
- Denial of Service (DoS)
- User to Root (U2R)
- Remote to Local (R2L)
- Probe

The classification models used for this task are based on **supervised learning**.

Dataset

We used a non cleaned dataset: UNSW-NB15. The raw packet was created by the `IXIA PerfectStorm tool`. This dataset is a labeled datset and in particular has nine types of attacks that we mapped in the categories we mentioned before as follows:

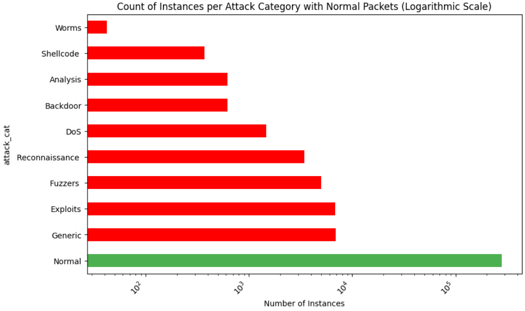
- DoS: DoS, Worms.
- U2R: Backdoor, Shellcode.
- R2L: Exploits, Analysis.
- Probe: Reconnaissance, Fuzzers, Generic.
- Normal: Benign packets.

We used the `label` column to map the attacks to categories. In particular all the `attack_cat` values were empty for the `Normal` class.

Category Distribution

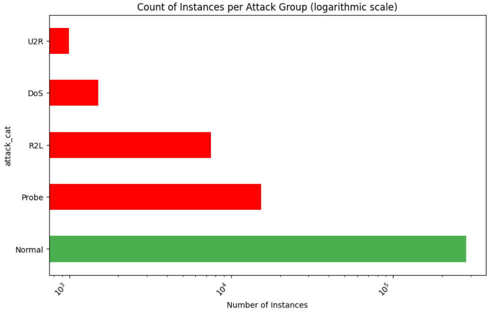
The dataset is highly unbalanced, with the majority of the samples belonging to the **Normal** class.

Normal	Generic	Exploits	Fuzzers	Reconnaissance	DoS	Backdoor	Analysis	Shellcode	Worms
281462	6894	6851	4970	3420	1465	623	621	371	42



Original Dataset attack category distribution

Normal	Probe	R2L	DoS	U2R
281462	15284	7472	1507	994



Our Dataset attack category distribution

Identify Missing and Erroneus Values

We identified 49 features in the dataset, 42 of which are numerical and 7 are categorical. The Dataset contains the following missing values:

	ct_flw_http_mthd	is_ftp_login	service	ct_ftp_cmd
Missing Values	273700	300350	167857	300350

Attribute`ct_flw_http_mthd`, that indicates how many *HTTP* method are present, is correlated with the value http in *attribute* `service`; but we found a discrepancy:

	ct_flw_http_mthd	service 'http'
Count Values	33019	32777

This means that the difference '242' that are signed as missing in `service` can be set to `http`.

Analysing the attributes `is_ftp_login`, `ct_ftp_cmd` we notice that they are equals in number, in values and in raws. Probably one of them is wrong, even if not anyway the attributes together are redundant.

So in the *pre-processing* phase we filled with *"missing"* the `service` attribute and we filled with *0* `ct_ftp_cmd` and `ct_flw_http_mthd`.

Feature Selection

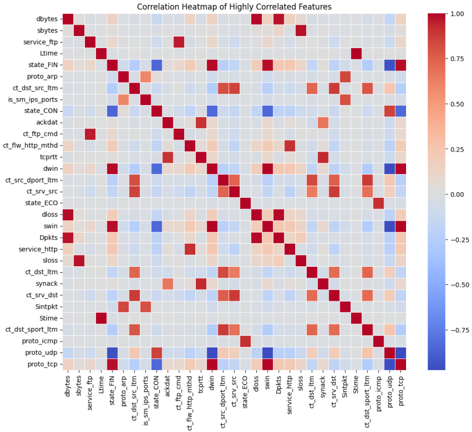
First we removed the following columns:

- Source and Destination IP addresses (`srcip, dstip`).
- Source and Destination Ports (`sport, dsport`).
- `is_ftp_login`.

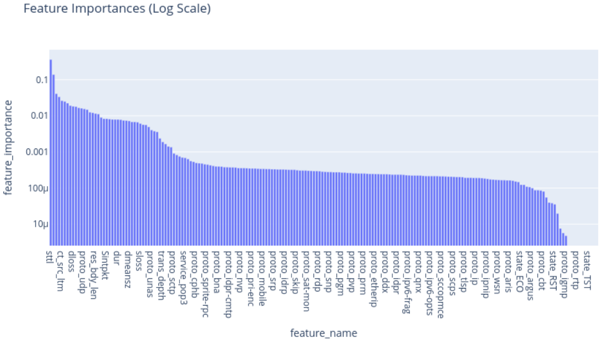
We dropped the duplicates. Then we adopted two different feature selection methods:

- **Logistic Regression**: We eliminated the features with a correlation higher than 0.8 among them.
- **Random Forest**: We used a **Decision Tree** model to estimate the coefficient of the most important features for constructing the tree. We dropped the others.

Feature Selection



Correlation matrix



Feature importance for Decision Tree

Data Preparation

As shown before the dataset is highly unbalanced with respect to the `Normal` class.

Normal	Probe	R2L	DoS	U2R
281462	15284	7472	1507	994

To minimize excessive transformations of the dataset, we first applied *undersampling* to the majority class, followed by *oversampling* of the others. Additionally, *Stratified Cross Validation* was performed. The following steps were taken:

- Data trasformation of nominal data with *OneHotEncoding* (using `get_dummies`).
- Performs Stratified Cross Validation with *StratifiedKFold* with $k = 10$.
- Split the dataset in training and test set to rebalancing only the first one, so
- Apply the `RandomUnderSampler` to reduce the *Normal* class to 100000 samples.
- Only then apply `SMOTE` to balance the training set.

Data Processing

Two types of models were utilized for the data processing task:

Logistic Regression

- Initially implemented for the *multi-class* classification problem.
- Performance was suboptimal, especially for `DoS` and `U2R` attacks, due to limitations in recognizing minority classes.
- We attempted to use the model for binary classification.

Random Forest

- Implemented only for multi-class classification problem.
- The result is quite better than Logistic Regression.
- We employed LIME (Local Interpretable Model-agnostic Explanations) to interpret model decisions.

Logistic Regression

Main Parameters: Max number of iterations `1000`; solver `lbfgs`.

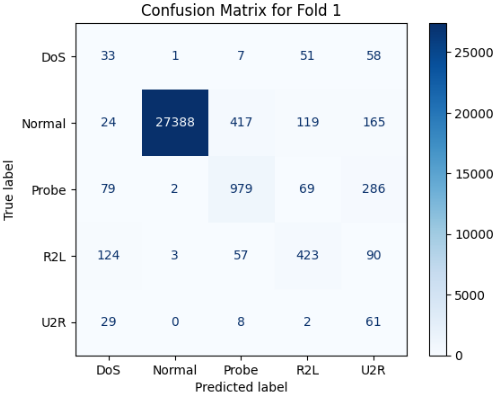
- Unbalanced dataset.
- Cross-validation with a balanced training dataset.
- Experimented with frequency encoding resulting in a faster model.

	Unbalance	Balance	
		OneHotEncoding	Frequency Encoding
Precision	0.491	0.495	0.492
Recall	0.493	0.614	0.645
F1-score	0.480	0.511	0.513

These approaches did yield poor improvement in `macro avg` performance.

Logistic Regression

A confusion matrix example for a cross-validation fold.



Fold one for `OHE` with balanced dataset.

Binomial Logistic Regression

- *'1'* represents an attack and *'0'* represents normal traffic.
- The results have improved significantly, as we expected.

Precision	Recall	F1-score
0.885	0.985	0.928

Random Forest

By using this model, we have observed improvements in performance.

Main Parameters:

– Number of Trees (n_estimators): 100

– Splitting Criterion (default): Gini Index

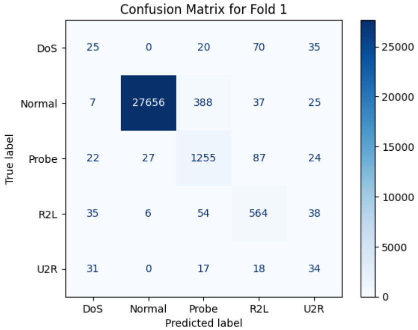
Macro avg Performance:

Precision	Recall	F1-score
0.565	0.628	0.590

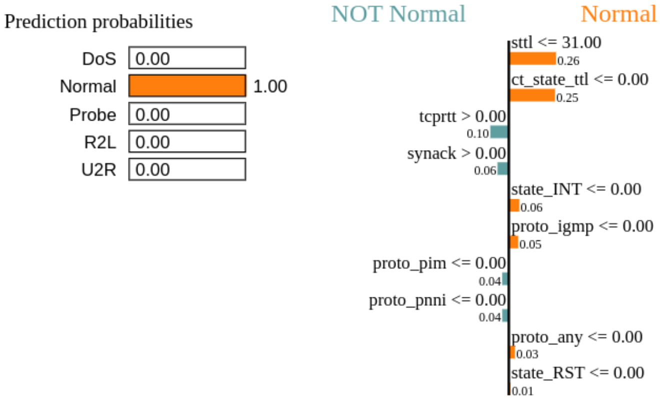
First Fold Performance:

	Precision	Recall	F1-score
Normal	0.998	0.984	0.991
DoS	0.208	0.167	0.185
Probe	0.724	0.887	0.797
R2L	0.727	0.809	0.766
U2R	0.218	0.340	0.266

Random Forest



First Fold Confusion Matrix



Lime Result

Conclusions

Comparing the two models, we can see that the **Random Forest** has `macro average` metrics better then **Logistic Regression** model.

	Precision	**Recall**	**F1-score**
Logistic Regression	0.495	0.614	0.511
Random Forest	0.565	0.628	0.590

- Since Logistic Regression requires a larger sample size to perform well. The limited occurrences of the `DoS` and `U2R` classes are insufficient to effectively train the model.
- Both models does not perform well on the `DoS` and `U2R` classes, that influence the `macro avg` metrics negatively.
- Dataset seems instead good for a binary classification problem.

Next Step Acknowledgements

- **Improve the dataset:** We could try to add to the dataset additional samples for *Minority class*. Extracting them from other datasets.
- **Hyperparameter tuning:** We could try to improve the model performance by tuning the hyperparameters.
- **Categorical feature handling:** We have applied One-Hot Encoding and Frequency Encoding for categorical features so far. Exploring alternative methods like Ordinal Encoding or Target Encoding might yield better results.

Since the results obtained were suboptimal, we believe the main issue lies with the dataset itself. Adding new samples for the *Minority classes* could help address the imbalance and resolve the challenges we encountered.

References

- Specification of `UNSW-NB15` dataset.
- "An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things".
- "UNSW-NB15: a comprehensive data set for network intrusion detection systems".