ML based Network Attack Analysis on NSL-KDD Dataset

A Multiclass Classification Problem

Artificial Intelligence For Cybersecurity Course

Author:
Luca Canuzzi

Agenda

- NSL-KDD Overview
- Preprocessing
- Multiclass Classification
- Conclusions
- Next Steps

NSL-KDD Overview

NSL-KDD [2] is an improvement to a classic network intrusion detection KDD'99 data set. The data was collected over nine weeks and consists of raw tcpdump traffic in a local area network (LAN) that simulates the environment of a typical United States Air Force LAN.

Some Improvements:

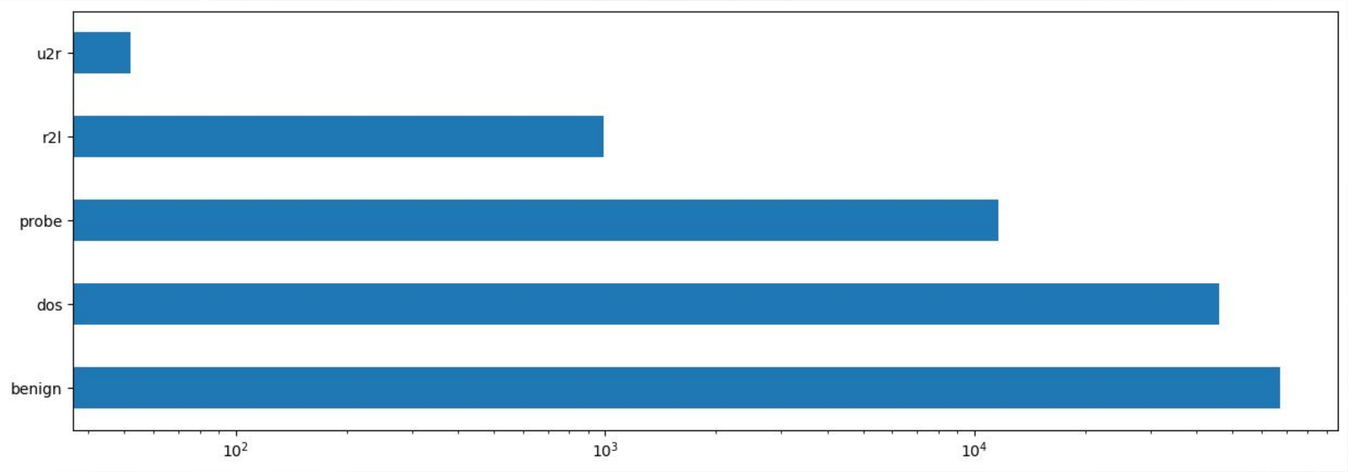
- It does not include redundant records in the train set
- There is no duplicate records in the proposed test sets.
- The number of records in the train and test sets are reasonable

The dataset includes 41 features and is divided into two sets: **KDDTrain+**, which contains **125,973** records, and **KDDTest+**, which contains **22,544** records, for training and testing purposes respectively.

NSL-KDD Overview

The dataset contains **38** different types of attacks, **24** available in the training set with an additional **14** in the test set only. These attacks belong to four general categories: **dos**: Denial of service, **r2l**: Unauthorized accesses from remote servers, **u2r**: Privilege escalation attempts, **probe**: Brute-force probing attacks.

Class distribution in the training set:



Preprocessing

- Differentiating between nominal, binary, and numeric features, by means of *kddcup.names* [3] file
  - *root_shell* marked as continuous but is supposed to be binary [3]
- Mapping from attack labels to attack categories specified in *training_attack_types.txt* [3]
- Dropping *success_pred* from dataframe [1]
- Cleaning binary features

	count	mean	std	min	25%	50%	75%	max
land	125973.0	0.000198	0.014086	0.0	0.0	0.0	0.0	1.0
logged_in	125973.0	0.395736	0.489010	0.0	0.0	0.0	1.0	1.0
root_shell	125973.0	0.001342	0.036603	0.0	0.0	0.0	0.0	1.0
su_attempted	125973.0	0.001103	0.045154	0.0	0.0	0.0	0.0	2.0
is_host_login	125973.0	0.000008	0.002817	0.0	0.0	0.0	0.0	1.0
is_guest_login	125973.0	0.009423	0.096612	0.0	0.0	0.0	0.0	1.0

Preprocessing

- Cleaning numeric features *num_ outbound_ cmds* has only one value “**0**”
- Splitting the test and training dataframes into data and labels
- Removing *attack_ cat* and *attack_ type* from data
- Transforming nominal features in binary features using one-hot encoding
  - *pandas.**get_ dummies***
- Transforming numeric features using z-score normalization
  - *sklearn.preprocessing.**StandardScaler***

    Example on the *duration* feature:

As is

count	125973.00000
mean	287.14465
std	2604.51531
min	0.00000
25%	0.00000
50%	0.00000
75%	0.00000
max	42908.00000

Name: duration, dtype: float64

StandardScaler

count	1.259730e+05
mean	3.916911e-16
std	1.000004e+00
min	-1.102492e-01
25%	-1.102492e-01
50%	-1.102492e-01
75%	-1.102492e-01
max	1.636428e+01

dtype: float64

Multiclass Classification

- Selected 5 Classifier
- A Stratified **10-fold** Cross Validation has been applied to determine the optimal **parameter** of each classifier and evaluate performance based on the **F1 score**
- Experiments were carried out with different balancing techniques
  - Unbalanced Training Set (As is)
  - Ensemble techniques: Random Undersampling to the majority classes **benign** and **dos**, using **probe** as a strategy, then 2 different oversampling strategy has been applied:
    1. Oversampling **u2r** using SMOTE and **r2l** as strategy
    2. Oversampling **r2l** and **u2r** using SMOTE and the default **auto** strategy

As is

benign	67343
dos	45927
probe	11656
r2l	995
u2r	52

Random Under.

benign	11656
dos	11656
probe	11656
r2l	995
u2r	52

i) R2L Strategy

benign	11656
dos	11656
probe	11656
r2l	995
u2r	995

ii) AUTO Strategy

benign	11656
dos	11656
probe	11656
r2l	11656
u2r	11656

Multiclass Classification

Classification Algorithms and Parameter Engineering:

- **Decision Tree (DT)**
  - Experimented with **max_depth*** = [10, 15, 20, 25]
    *The maximum depth of the tree, default=None
- **K Nearest Neighbors (KNN)**
  - Experimented with **n_neighbors*** = [3, 5, 10, 20]
    *Number of neighbors to use, default=5
- **LinearSVC (LSVC)**
  - Experimented with **C*** = [1, 5, 10, 20]
    *Regularization parameter. The regularization strength is inversely proportional to C, default=1
- **Random Forest (RF)**
  - Experimented with **n_estimators*** = [20, 40, 80, 100]
    *The number of trees in the forest, default=100
- **GaussianNB (GNB)**
  - Experimented with **var_smoothing*** = [1e-7, 1e-9, 1e-11, 1e-13]
    *Portion of the largest variance of all features added to stabilise the calculation, default=1e-9

Cross Validation

UNBALANCED DATASET

DT	KNN	LSVC	RF	GNB
max_depth = 10	n_neighbors = 3	C = 1	n_estimators = 20	var_smoothing = 1e-7
AVG FSCORE: 0.886	AVG FSCORE: 0.910	AVG FSCORE: 0.832	AVG FSCORE: 0.912	AVG FSCORE: 0.530
max_depth = 15	n_neighbors = 5	C = 5	n_estimators = 40	var_smoothing = 1e-9
AVG FSCORE: 0.891	AVG FSCORE: 0.860	AVG FSCORE: 0.845	AVG FSCORE: 0.925	AVG FSCORE: 0.423
max_depth = 20	n_neighbors = 10	C = 10	n_estimators = 80	var_smoothing = 1e-11
AVG FSCORE: 0.904	AVG FSCORE: 0.824	AVG FSCORE: 0.844	AVG FSCORE: 0.930	AVG FSCORE: 0.398
max_depth = 25	n_neighbors = 20	C = 20	n_estimators = 100	var_smoothing = 1e-13
AVG FSCORE: 0.915	AVG FSCORE: 0.805	AVG FSCORE: 0.842	AVG FSCORE: 0.927	AVG FSCORE: 0.359

Cross Validation

BALANCED DATASET R2L STRATEGY

DT	KNN	LSVC	RF	GNB
max_depth = 10	n_neighbors = 3	C = 1	n_estimators = 20	var_smoothing = 1e-7
AVG FSCORE: 0.902	AVG FSCORE: 0.842	AVG FSCORE: 0.774	AVG FSCORE: 0.928	AVG FSCORE: 0.542
max_depth = 15	n_neighbors = 5	C = 5	n_estimators = 40	var_smoothing = 1e-9
AVG FSCORE: 0.900	AVG FSCORE: 0.833	AVG FSCORE: 0.768	AVG FSCORE: 0.927	AVG FSCORE: 0.466
max_depth = 20	n_neighbors = 10	C = 10	n_estimators = 80	var_smoothing = 1e-11
AVG FSCORE: 0.898	AVG FSCORE: 0.818	AVG FSCORE: 0.766	AVG FSCORE: 0.926	AVG FSCORE: 0.415
max_depth = 25	n_neighbors = 20	C = 20	n_estimators = 100	var_smoothing = 1e-13
AVG FSCORE: 0.891	AVG FSCORE: 0.791	AVG FSCORE: 0.766	AVG FSCORE: 0.929	AVG FSCORE: 0.381

Cross Validation

BALANCED DATASET AUTO STRATEGY

DT	KNN	LSVC	RF	GNB
max_depth = 10	n_neighbors = 3	C = 1	n_estimators = 20	var_smoothing = 1e-7
AVG FSCORE: 0.724	AVG FSCORE: 0.807	AVG FSCORE: 0.675	AVG FSCORE: 0.927	AVG FSCORE: 0.571
max_depth = 15	n_neighbors = 5	C = 5	n_estimators = 40	var_smoothing = 1e-9
AVG FSCORE: 0.794	AVG FSCORE: 0.790	AVG FSCORE: 0.680	AVG FSCORE: 0.925	AVG FSCORE: 0.509
max_depth = 20	n_neighbors = 10	C = 10	n_estimators = 80	var_smoothing = 1e-11
AVG FSCORE: 0.829	AVG FSCORE: 0.769	AVG FSCORE: 0.680	AVG FSCORE: 0.932	AVG FSCORE: 0.477
max_depth = 25	n_neighbors = 20	C = 20	n_estimators = 100	var_smoothing = 1e-13
AVG FSCORE: 0.849	AVG FSCORE: 0.740	AVG FSCORE: 0.681	AVG FSCORE: 0.928	AVG FSCORE: 0.435

PAIRED WILCOXON TEST

UNBALANCED DATASET

	KNN Fscore: 0.910	LSVC Fscore: 0.845	RF Fscore: 0.930	GNB Fscore: 0.530
DT Fscore: 0.915	*p=0.6953125*	*p=0.005859375*	*p=0.375*	*p=0.001953125*
KNN Fscore: 0.910		*p=0.00390625*	*p=0.275390625*	*p=0.001953125*
LSVC Fscore: 0.845			*p=0.001953125*	*p=0.001953125*
RF Fscore: 0.930				*p=0.001953125*

With a confidence level α=0.05 and a p-value ≤ α the null hypothesis is rejected

PAIRED WILCOXON TEST

BALANCED DATASET R2L STRATEGY

	KNN Fscore: 0.842	LSVC Fscore: 0.774	RF Fscore: 0.929	GNB Fscore: 0.542
DT Fscore: 0.902	*p=0.001953125*	*p=0.001953125*	*p=0.001953125*	*p=0.001953125*
KNN Fscore: 0.842		*p=0.001953125*	*p=0.001953125*	*p=0.001953125*
LSVC Fscore: 0.774			*p=0.001953125*	*p=0.001953125*
RF Fscore: 0.929				*p=0.001953125*

With a confidence level α=0.05 and a p-value ≤ α the null hypothesis is rejected

PAIRED WILCOXON TEST

BALANCED DATASET AUTO STRATEGY

	KNN Fscore: 0.807	**LSVC** Fscore: 0.681	**RF** Fscore: 0.932	**GNB** Fscore: 0.571
DT Fscore: 0.849	*p=0.001953125*	*p=0.001953125*	*p=0.001953125*	*p=0.001953125*
KNN Fscore: 0.807		*p=0.001953125*	*p=0.001953125*	*p=0.001953125*
LSVC Fscore: 0.681			*p=0.001953125*	*p=0.001953125*
RF Fscore: 0.932				*p=0.001953125*

With a confidence level α=0.05 and a p-value ≤ α the null hypothesis is rejected

Evaluate the Models on the Test Set

UNBALANCED TRAINING SET

	DT	KNN	LSVC	RF	GNB	
	fscore	fscore	fscore	fscore	fscore	support
benign	0.792	0.784	0.753	0.781	0.805	**9711**
dos	0.872	0.859	0.834	0.860	0.758	**7636**
r2l	0.108	0.104	0.076	0.055	0.390	**2574**
probe	0.712	0.707	0.693	0.705	0.517	**2423**
u2r	0.177	0.085	0.129	0.048	0.104	**200**
AVG F-S	**0.532**	**0.508**	**0.497**	**0.490**	**0.515**	

BALANCED TRAINING SET R2L STRATEGY

	DT	KNN	LSVC	RF	GNB	
	fscore	fscore	fscore	fscore	fscore	support
benign	0.829	0.810	0.789	0.786	0.793	**9711**
dos	0.791	0.846	0.868	0.838	0.722	**7636**
r2l	0.148	0.207	0.226	0.122	0.437	**2574**
probe	0.702	0.747	0.713	0.767	0.527	**2423**
u2r	0.182	0.097	0.101	0.242	0.105	**200**
AVG F-S	**0.530**	**0.541**	**0.539**	**0.551**	**0.517**	

BALANCED TRAINING SET AUTO STRATEGY

	DT	KNN	LSVC	RF	GNB	
	fscore	fscore	fscore	fscore	fscore	support
benign	0.813	0.810	0.803	0.793	0.818	**9711**
dos	0.842	0.846	0.841	0.844	0.802	**7636**
r2l	0.156	0.207	0.507	0.207	0.519	**2574**
probe	0.739	0.748	0.705	0.794	0.583	**2423**
u2r	0.085	0.077	0.092	0.162	0.108	**200**
AVG F-S	**0.527**	**0.538**	**0.590**	**0.560**	**0.566**	

Next Steps

Possible future analysis:

- Experimenting with dimensionality reduction
- Experimenting with more sophisticated parameter engineering
- Experimenting with Neural Networks

References

1. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.
2. NSL-KDD dataset, *https://www.unb.ca/cic/datasets/nsl.html*
3. KDD Cup 1999 Data, *http://kdd.ics.uci.edu/databases/kddcup99/kddcup99*

THANKS