Deep Learning for Network Intrusion Detection System

CSC I1910
Spring 2023
Temidayo Akinyemi, Pushpen Bikash Goala, Ivan Miller
Professor: Michael Grossberg

Motivation

- Huge volume of network traffic to be analyzed
- Quick identification of malicious attack traffic is a challenge in cybersecurity
- Lots of detection systems are signature-based
- Security analysts are usually overwhelmed by the volume of data traffic they have to look through
- Lots of false positives - eventually causing alert fatigue
- Aim to create intrusion detection system to identify attack patterns with a high detection rate
- UNSW-NB15 Dataset - 9 attack classes, 49 features, 2mill records

Intended Experiments

Part 1: Data preprocessing

- Data cleaning
- Feature Selection
- Class Imbalance

Part 2: Baseline Methods

- Random Forest
- Logistic Regression
- KNN
- Simple Sequential Model

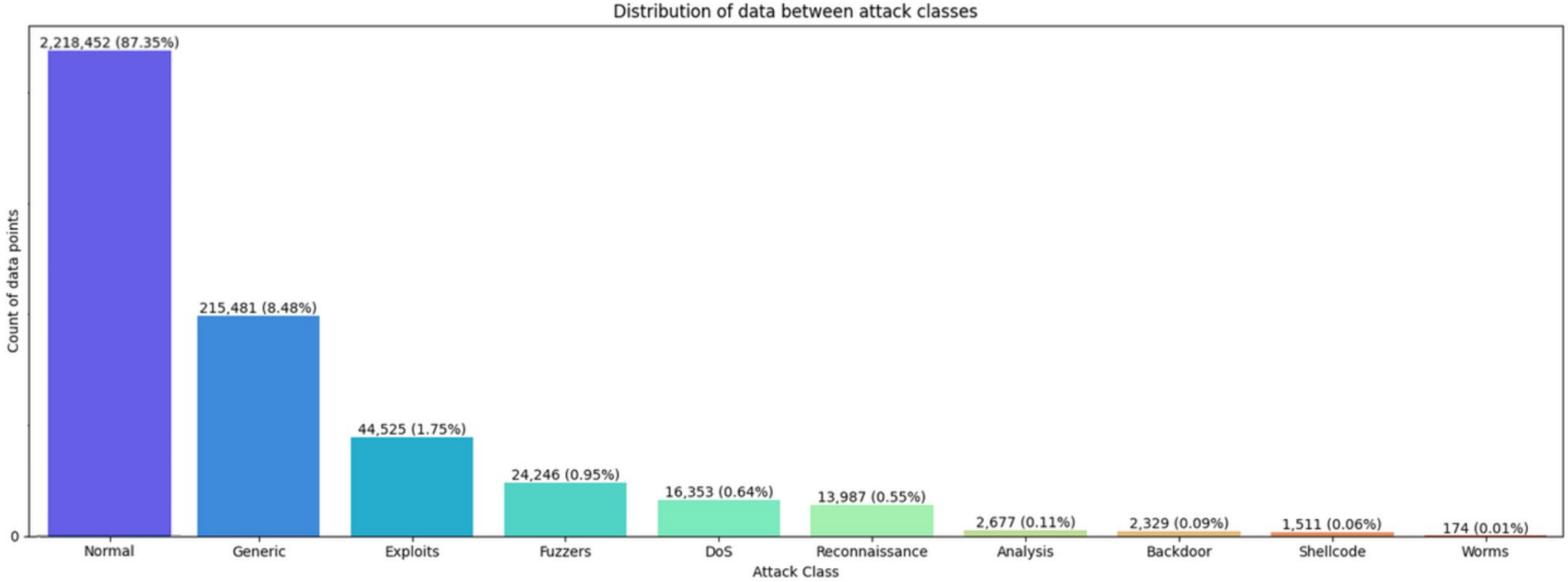
Part 3: Deep Learning Methods

- Larger Sequential Model
- Autoencoder & Variational Autoencoder

Part 1: Data preprocessing

NB-15 Dataset

- Data Cleaning
- Feature Importance (VIF, Pearson correlation, RF)
- Visualization of classes



Dataset: Pearson Correlation + VIF + Random Forest

- Pearson Correlation

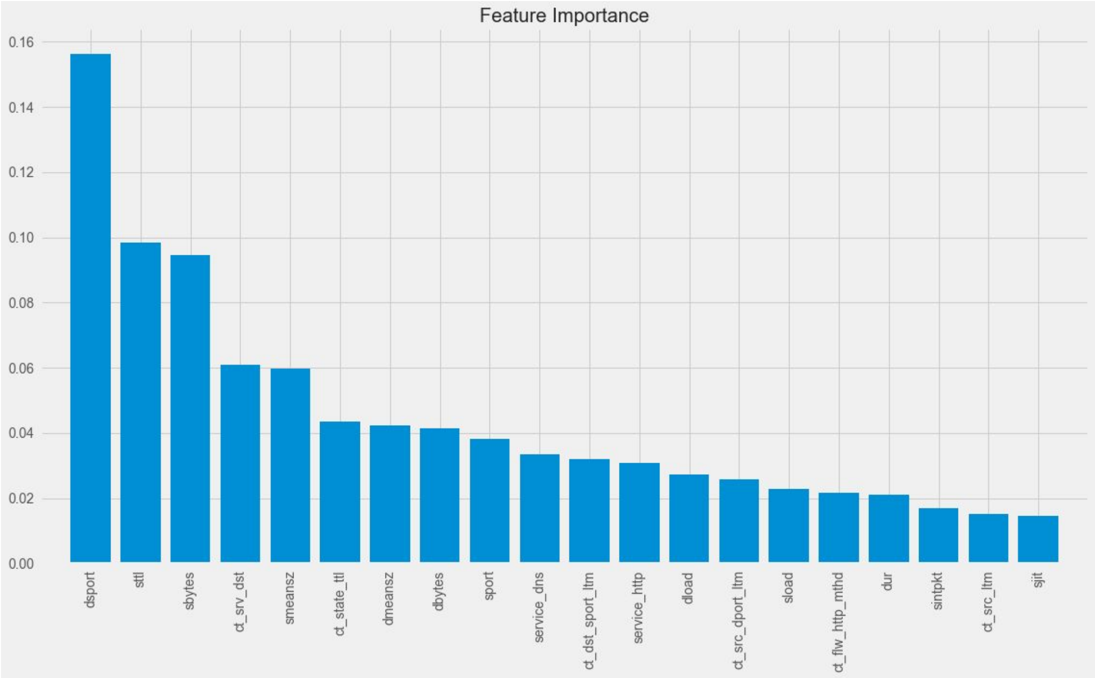
- Variance Inflation Factor (VIF) is given by

$$VIF_i = \frac{1}{1-R_i^2}$$

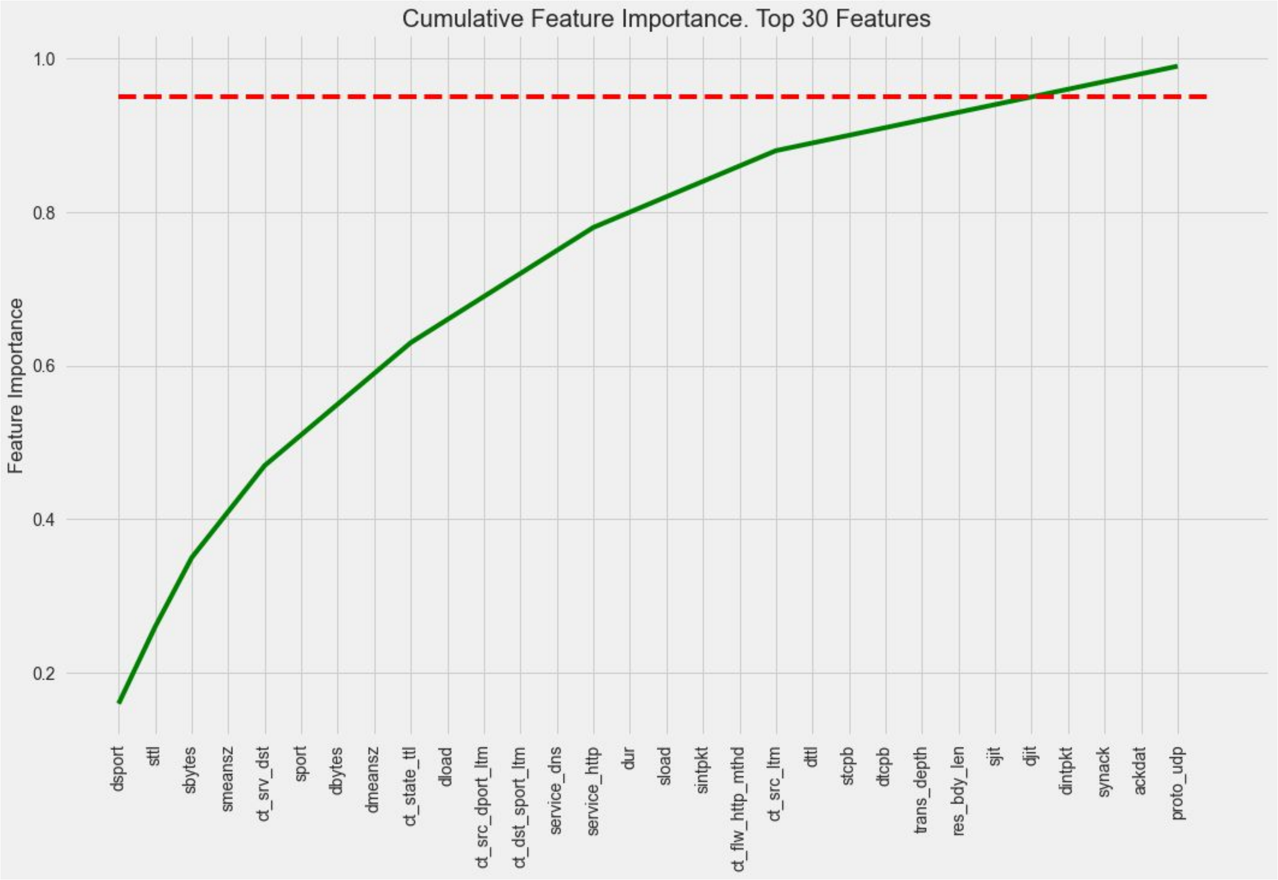
, where R^2 is the coefficient of determination

- Random Forest

Dataset: Feature Importance with Random Forest

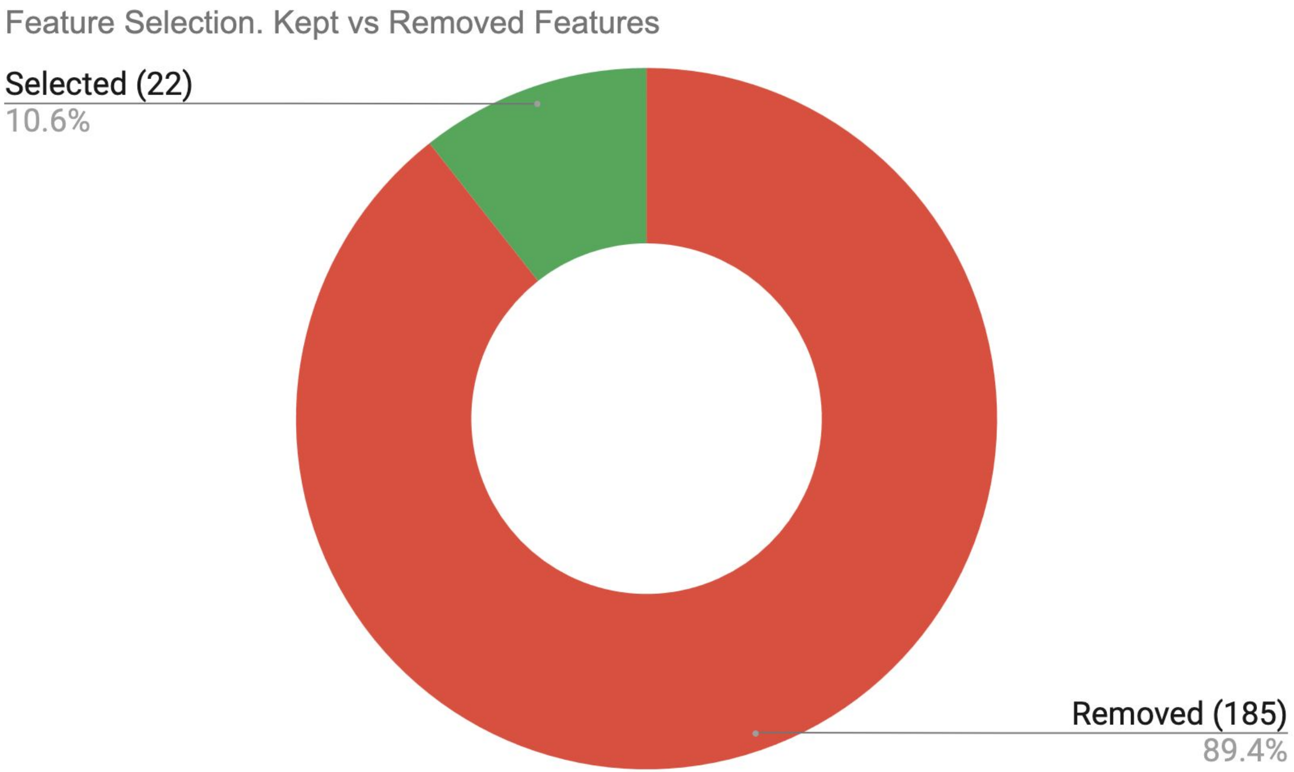


Dataset: Cumulative Feature Importance

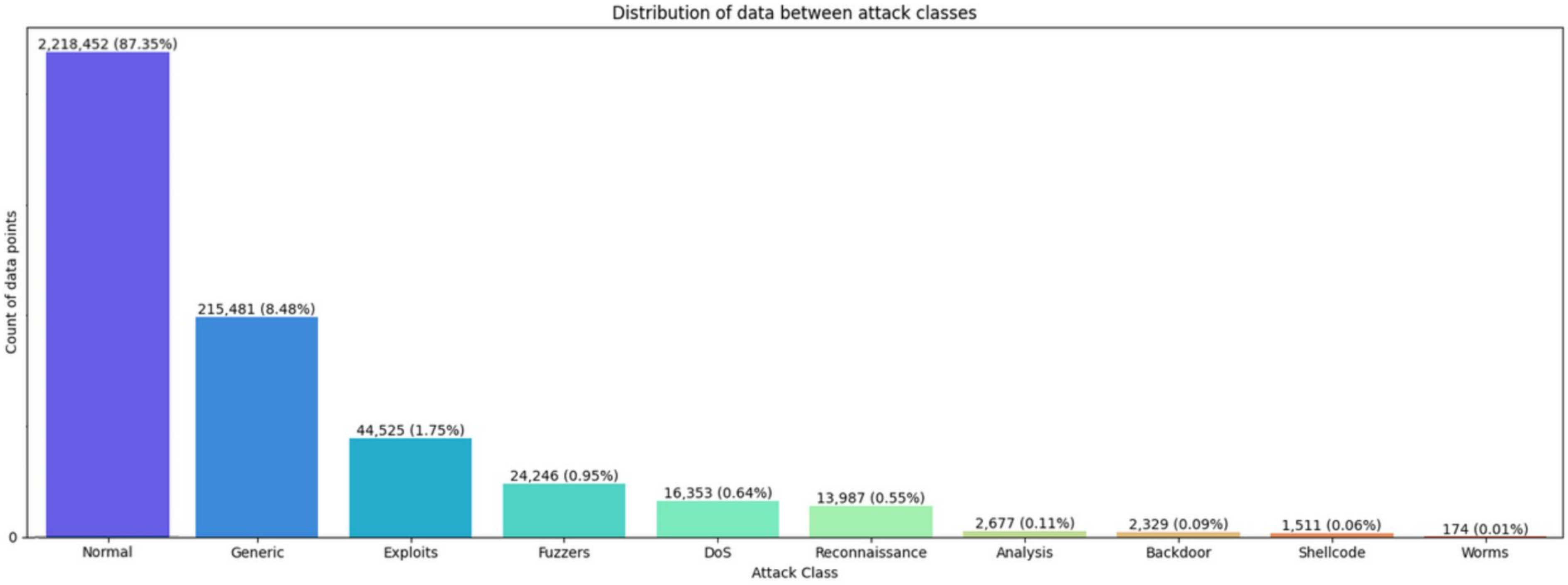


Dataset: Removed 89% of Features

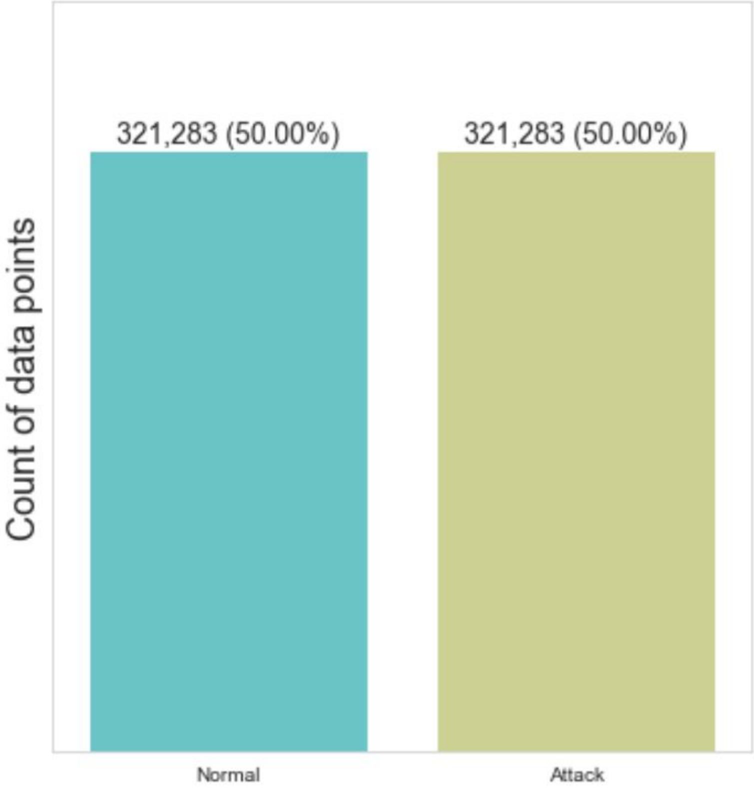
- sttl
- sbytes
- smeansz
- sload
- dmeansz
- dbytes
- dsport
- ct_srv_src
- dintpkt
- sport
- ct_dst_sport_ltm
- tcprtt
- dur
- dttl
- sintpkt
- ct_src_ltm
- spkts
- sjit
- djit
- stcpb
- res_bdy_len
- trans_depth



Class Imbalance



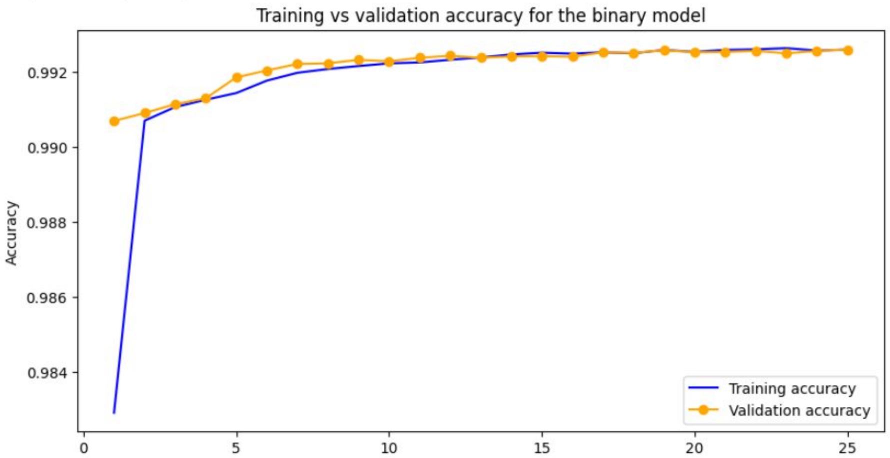
Binary Classification: Undersampling Majority Class

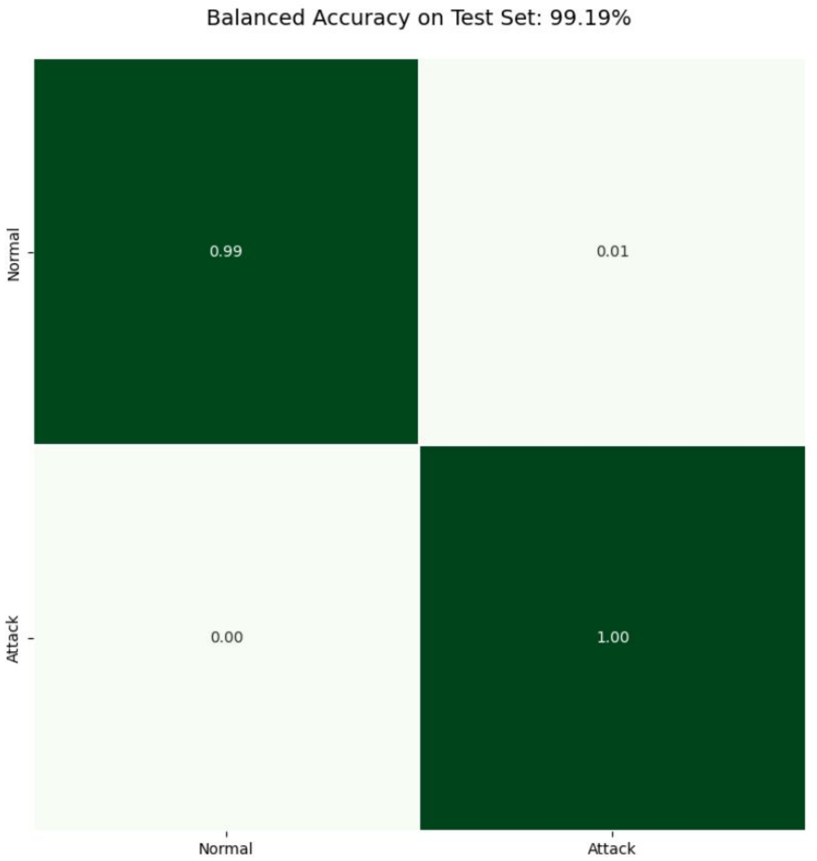


Part 2: Baseline Methods

Binary Classification: Results

- Random Forest: 99.32%
- Logistic Regression: 98.93%
- K-Nearest Neighbors (KNN): 99.19%
- Neural Network: 99.30%
    - 2 hidden layers (256 >> 128 units)
    - 40k params & 25 epochs

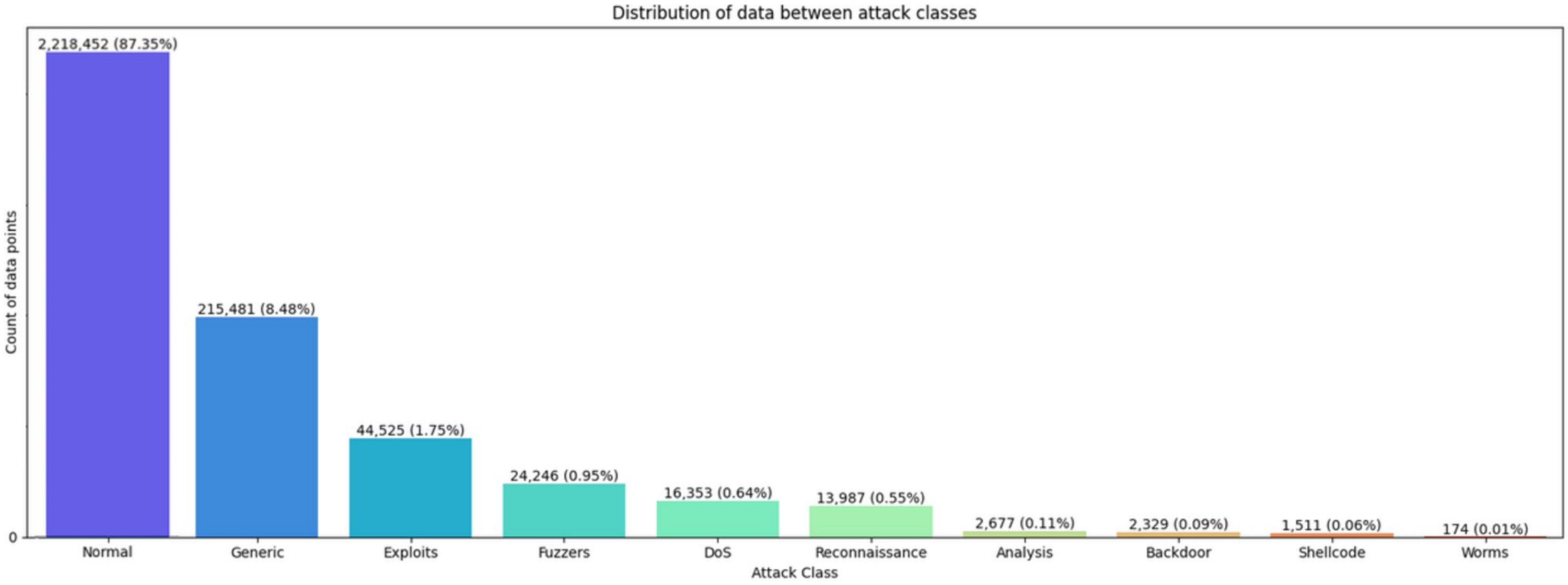




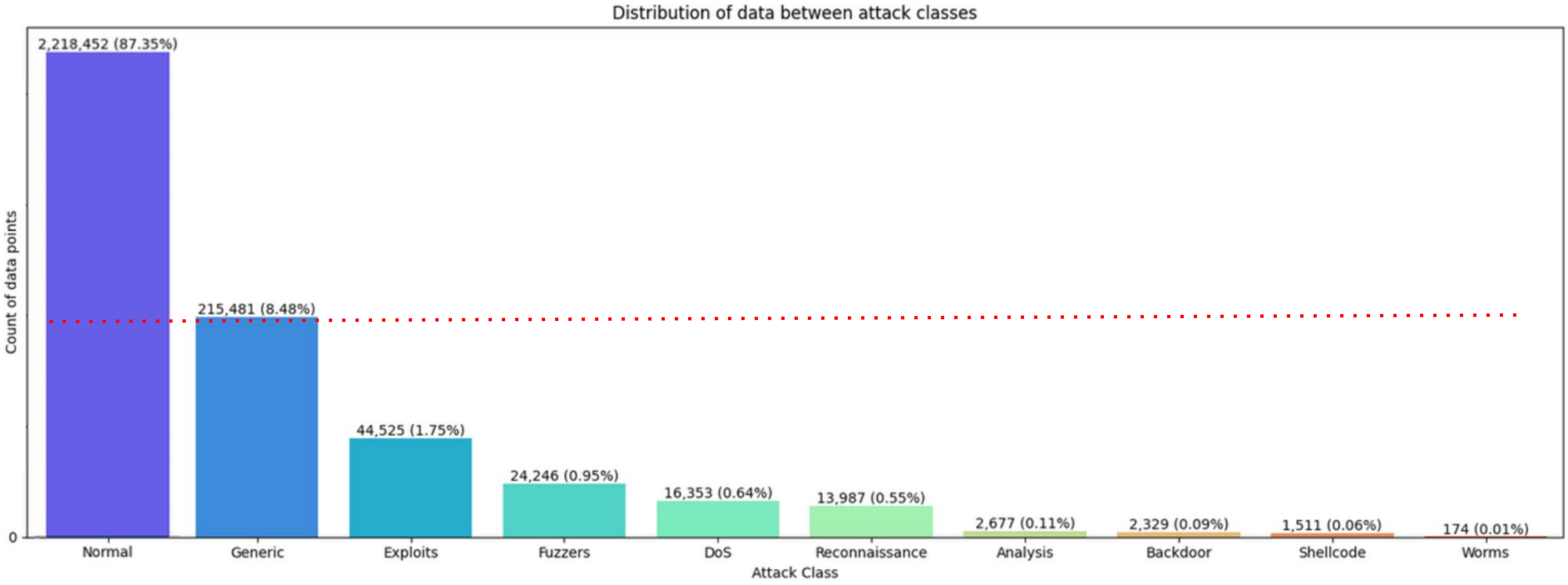
Classification Report:

	precision	recall	f1-score	support
Normal	1.00	0.99	0.99	48193
Attack	0.99	1.00	0.99	48193
accuracy			0.99	96386
macro avg	0.99	0.99	0.99	96386
weighted avg	0.99	0.99	0.99	96386

Multi-class. Addressing Imbalance with SMOTE

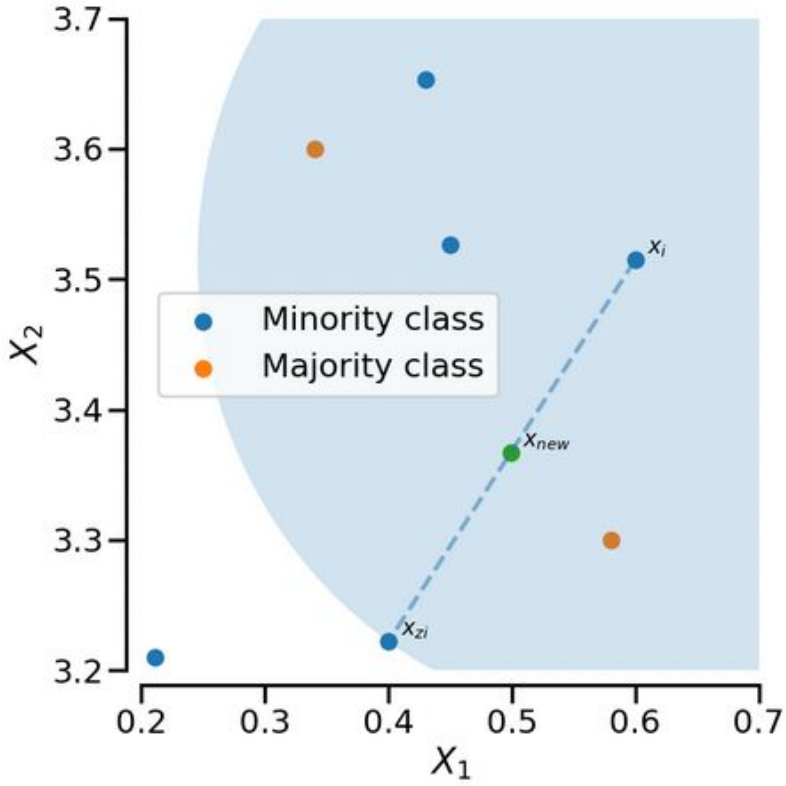


Class Imbalance

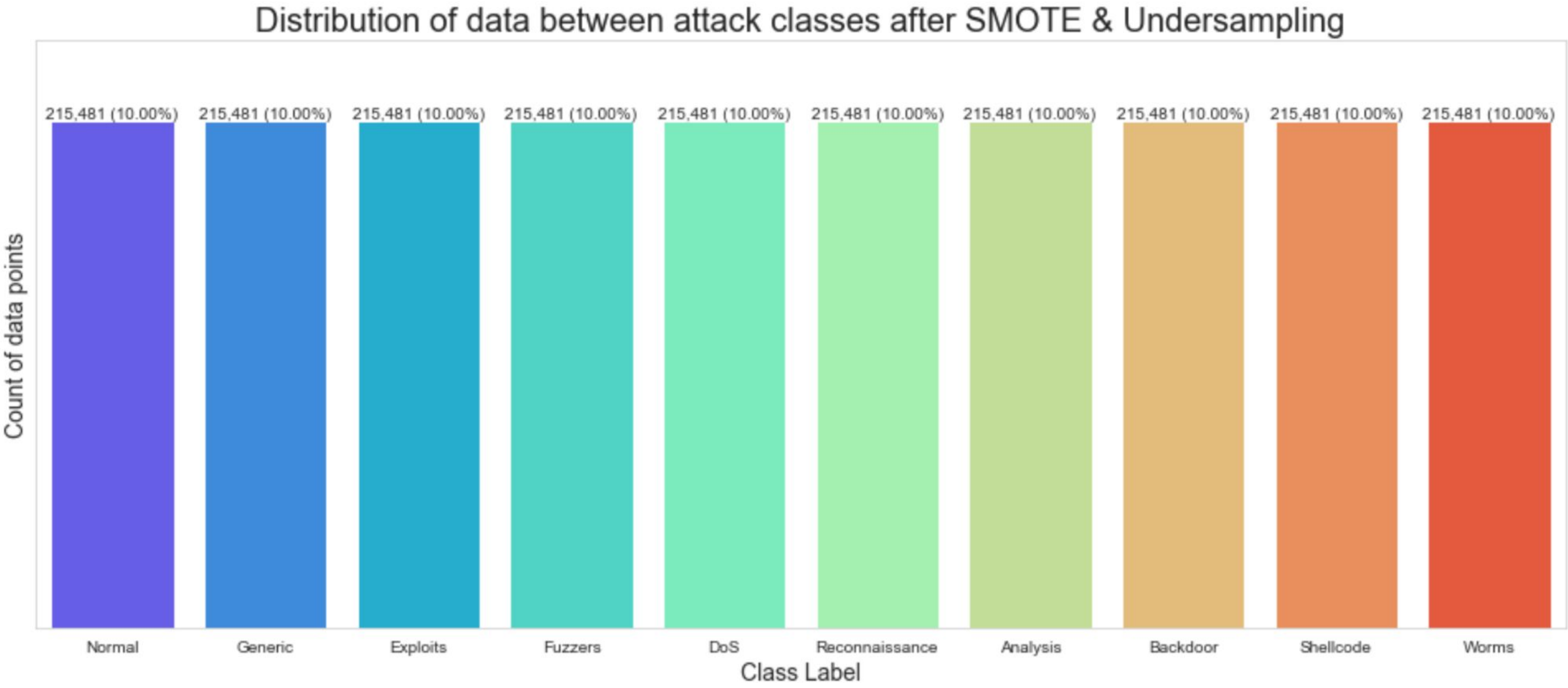


Class Imbalance: SMOTE & undersampling

Synthetic **M**inority **O**versampling **TE**chnique



Class Imbalance: SMOTE & Undersampling



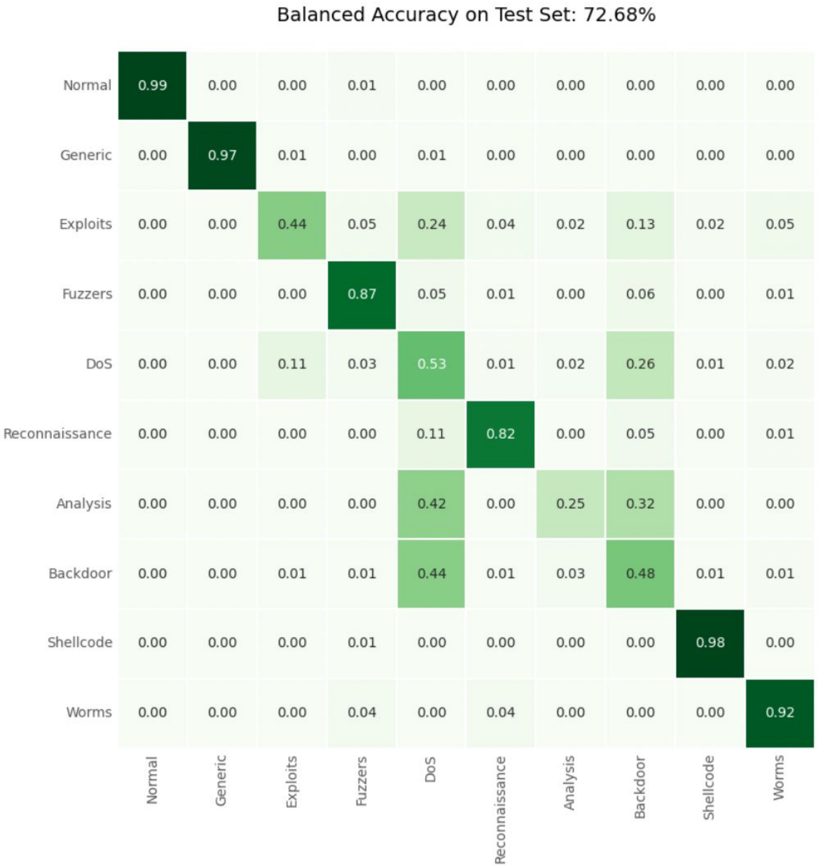
Cross-validation and Grid Search with SMOTE

Steps:

- Subsampled imbalanced data
- 5-Fold Cross-Validation
- Applied SMOTE & Undersampling on TRAIN folds
- Grid Search (80 combinations):
  - 300 trees
  - Min 3 samples to split a node
  - 10 nodes max
  - Min 2 samples per leaf

Trained Random Forest on the full balanced train set

Multi-class: Random Forest




```
inference time: 13.509408955000254
Accuracy:0.97
Balanced accuracy:0.73
Precision:0.98
Recall:0.97
F1-score:0.97
```

```
Classification Report:
                precision    recall  f1-score   support

        Normal       1.00      0.99      0.99    332768
       Generic       1.00      0.97      0.99     32322
      Exploits       0.82      0.44      0.58      6679
       Fuzzers       0.42      0.87      0.56      3637
           DoS       0.34      0.53      0.41      2453
Reconnaissance       0.81      0.82      0.81      2098
      Analysis       0.12      0.25      0.16       402
      Backdoor       0.08      0.48      0.13       349
     Shellcode       0.49      0.98      0.66       227
         Worms       0.04      0.92      0.07        26

      accuracy                           0.97    380961
     macro avg       0.51      0.73      0.54    380961
  weighted avg       0.98      0.97      0.97    380961
```

Multi-class: Random Forest

Balanced Accuracy on Test Set: 72.68%

	Normal	Generic	Exploits	Fuzzers	DoS	Reconnaissance	Analysis	Backdoor	Shellcode	Worms
Normal	0.99	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
Generic	0.00	0.97	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00
Exploits	0.00	0.00	0.44	0.05	0.24	0.04	0.02	0.13	0.02	0.05
Fuzzers	0.00	0.00	0.00	0.87	0.05	0.01	0.00	0.06	0.00	0.01
DoS	0.00	0.00	0.11	0.03	0.53	0.01	0.02	0.26	0.01	0.02
Reconnaissance	0.00	0.00	0.00	0.00	0.11	0.82	0.00	0.05	0.00	0.01
Analysis	0.00	0.00	0.00	0.00	0.42	0.00	0.25	0.32	0.00	0.00
Backdoor	0.00	0.00	0.01	0.01	0.44	0.01	0.03	0.48	0.01	0.01
Shellcode	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.98	0.00
Worms	0.00	0.00	0.00	0.04	0.00	0.04	0.00	0.00	0.00	0.92

```
inference time: 13.509408955000254
Accuracy:0.97
Balanced accuracy:0.73
Precision:0.98
Recall:0.97
F1-score:0.97
```

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	0.99	0.99	332768
Generic	1.00	0.97	0.99	32322
Exploits	0.82	0.44	0.58	6679
Fuzzers	0.42	0.87	0.56	3637
DoS	0.34	0.53	0.41	2453
Reconnaissance	0.81	0.82	0.81	2098
Analysis	0.12	0.25	0.16	402
Backdoor	0.08	0.48	0.13	349
Shellcode	0.49	0.98	0.66	227
Worms	0.04	0.92	0.07	26
accuracy			0.97	380961
macro avg	0.51	0.73	0.54	380961
weighted avg	0.98	0.97	0.97	380961

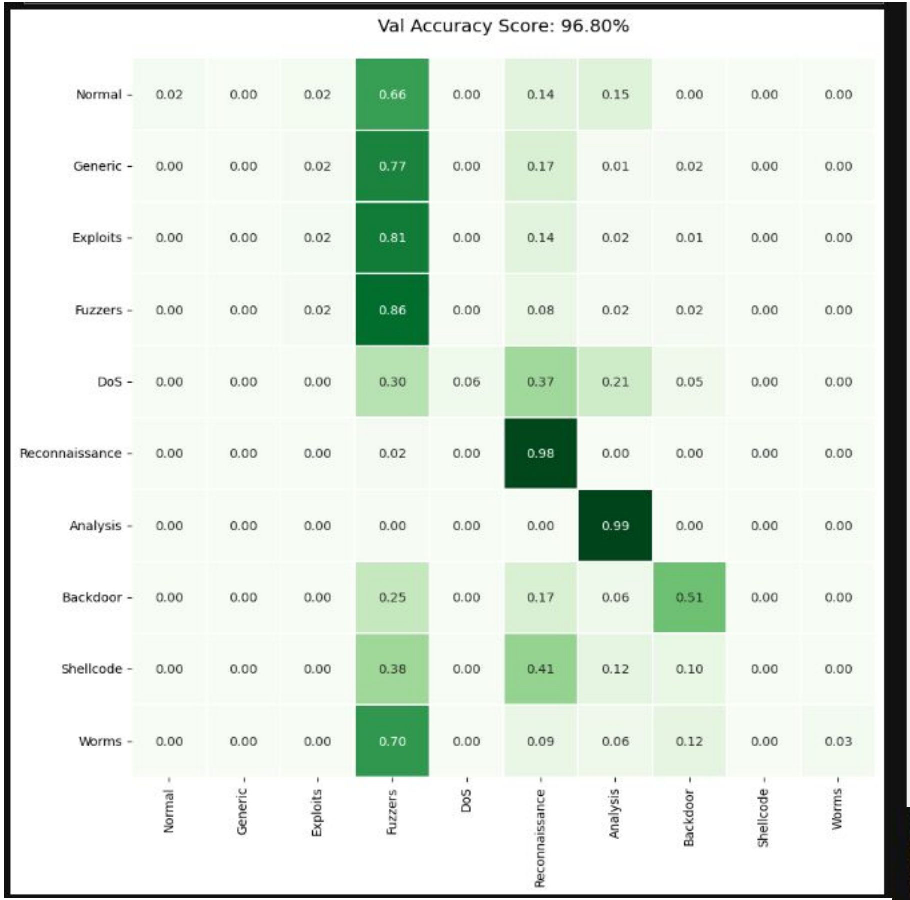
Logistic Regression (multiclass)(Test)

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	0.99	0.99	221846
Generic	1.00	0.98	0.99	21548
Exploits	0.77	0.43	0.55	4452
Fuzzers	0.43	0.80	0.56	2425
DoS	0.33	0.66	0.44	1635
Reconnaissance	0.65	0.68	0.66	1399
Analysis	0.10	0.28	0.15	268
Backdoor	0.07	0.18	0.10	233
Shellcode	0.54	0.95	0.69	151
Worms	0.02	0.82	0.03	17
accuracy			0.97	253974
macro avg	0.49	0.68	0.52	253974
weighted avg	0.98	0.97	0.97	253974

Dataset	Model	F1-score	Accuracy	Balanced Accuracy	Precision	Recall
Test	Logistic Regression	0.97%	0.97%	0.68%	0.98%	0.97%

Logistic Regression (after log transformation)

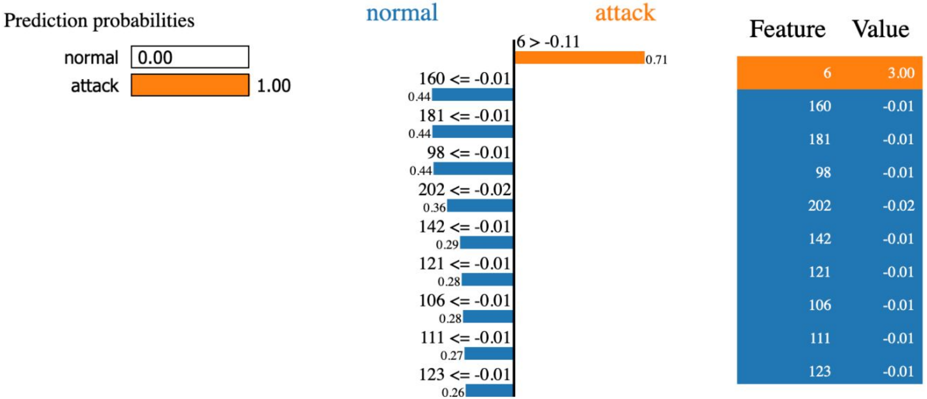


	column	range	std_dev
0	dur	8786.637695	12.790263
1	sbytes	13677393.000000	59568.237445
2	dbytes	14655417.000000	160579.679925
3	sttl	255.000000	74.598099
4	dttl	254.000000	42.903800
5	sload	5640000000.000000	118277920.527573
6	dload	128761904.000000	4228217.689794
7	spkts	10200.000000	77.736579
8	swin	255.000000	125.460743
9	stcpb	4294958913.000000	1421676319.890164
10	dtcpb	4294953724.000000	1422227817.896179
11	smeansz	1504.000000	152.161292
12	dmeansz	1500.000000	335.562090
13	trans_depth	172.000000	0.325749
14	res_bdy_len	6558056.000000	47617.334730
15	sjit	1483830.917000	17040.869328
16	djit	781221.118300	3383.518124
17	stime	2334691.000000	1134350.896264
18	sintpkt	60009.992000	2783.993022
19	dintpkt	57739.240000	1428.959277
20	tcprtt	10.037506	0.046891
21	synack	4.525272	0.026429
22	ackdat	5.512234	0.024193
23	is_sm_ips_ports	1.000000	0.040646
24	ct_state_ttl	6.000000	0.683346
25	ct_flw_http_mthd	36.000000	0.556569
26	is_ftp_login	1.000000	0.130179
27	ct_ftp_cmd	8.000000	0.184054
28	ct_srv_src	66.000000	10.830274
29	ct_dst_ltm	66.000000	8.160660
30	ct_src_ ltm	66.000000	8.204105
31	ct_dst_sport_ltm	59.000000	6.176511
32	total_bytes	14725649.000000	173750.649661

Dataset	Model	F1-score	Accuracy	Balanced Accuracy	Precision	Recall
Val	Logistic Regression	0.96%	0.97%	0.35%	0.97%	0.97%

Exploring LIME for binary classification

```
In [33]: import lime
         from lime import lime_tabular

In [36]: import numpy as np

In [37]: explainer = lime_tabular.LimeTabularExplainer(
                     training_data = np.array(x_train_ssc),
                     feature_names = pd.DataFrame(x_train_ssc).columns,
                     class_names=['normal', 'attack'],
                     mode='classification'
         )

In [49]: exp = explainer.explain_instance(
                 data_row = pd.DataFrame(x_test_ssc).iloc[44452],
                 predict_fn = model.predict_proba
         )

In [50]: exp.show_in_notebook(show_table=True)
```



Cross Validation & Grid Search

- Subsampled 20 percent of the imbalanced train data (all features) to use for cross-validation
- Reduced the feature number to 20 using Autoencoder
- Perform 5 fold cross-validation with upsampling using SMOTE on TRAIN folds ONLY
- Loop to find the best parameters considering balanced accuracy as the main metric

Cross Validation & Grid Search

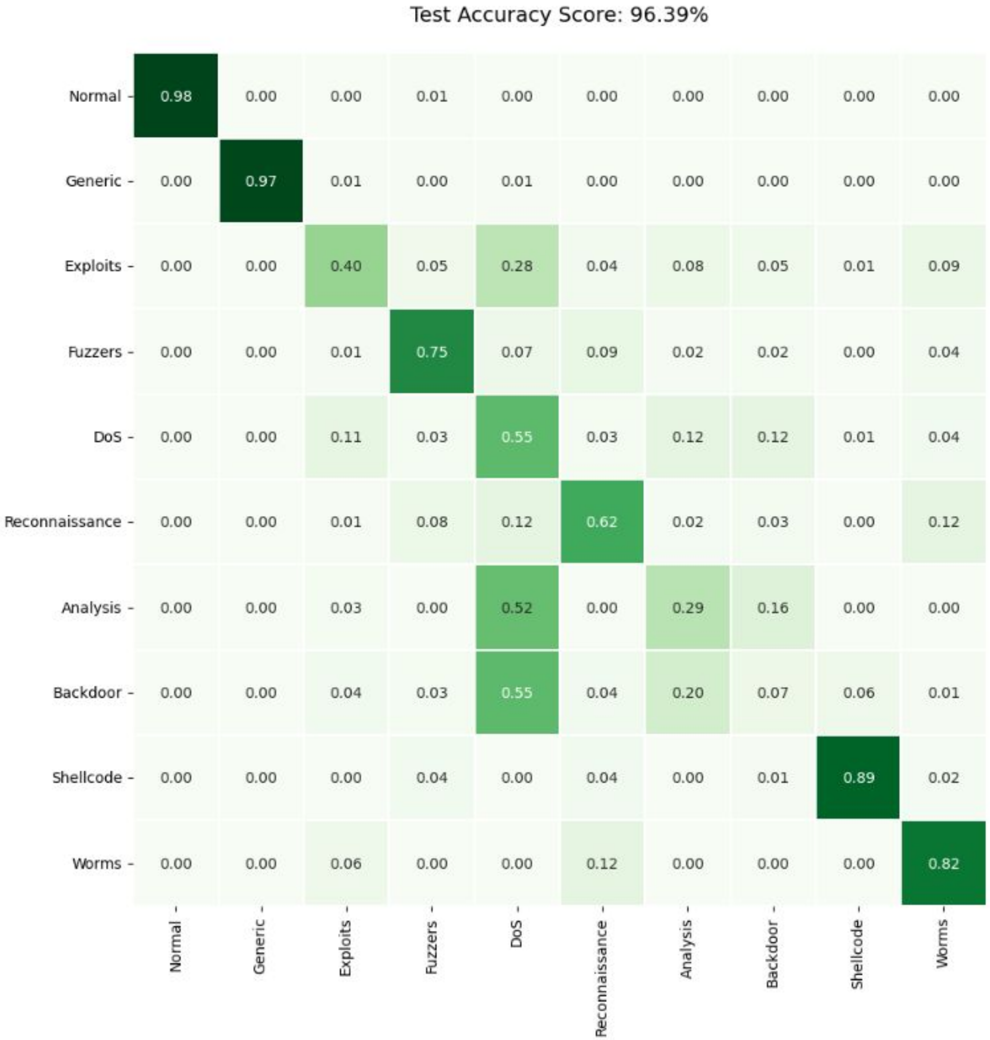
```
[{'n_neighbors': 49, 'balanced_accuracy': 0.550814805116250б},
 {'n_neighbors': 50, 'balanced_accuracy': 0.5511920811605482},
 {'n_neighbors': 51, 'balanced_accuracy': 0.5523819275900327},
 {'n_neighbors': 52, 'balanced_accuracy': 0.5557178600589406},
 {'n_neighbors': 53, 'balanced_accuracy': 0.5556471417887272},
 {'n_neighbors': 54, 'balanced_accuracy': 0.551750839937972},
 {'n_neighbors': 55, 'balanced_accuracy': 0.5599760683814748},
 {'n_neighbors': 56, 'balanced_accuracy': 0.5573045869512194},
 {'n_neighbors': 57, 'balanced_accuracy': 0.5569417807182738},
 {'n_neighbors': 58, 'balanced_accuracy': 0.5560335307225059},
 {'n_neighbors': 59, 'balanced_accuracy': 0.5630037411254806},
 {'n_neighbors': 60, 'balanced_accuracy': 0.56100506364247},
 {'n_neighbors': 61, 'balanced_accuracy': 0.5677061492385941},
 {'n_neighbors': 62, 'balanced_accuracy': 0.5653395355553826},
 {'n_neighbors': 63, 'balanced_accuracy': 0.566290560405714},
 {'n_neighbors': 64, 'balanced_accuracy': 0.5647718505852728},
 {'n_neighbors': 65, 'balanced_accuracy': 0.5643503723158758},
 {'n_neighbors': 66, 'balanced_accuracy': 0.5665008392798578},
 {'n_neighbors': 67, 'balanced_accuracy': 0.5669077215489262},
 {'n_neighbors': 68, 'balanced_accuracy': 0.5643401813505488},
 {'n_neighbors': 69, 'balanced_accuracy': 0.5649583175224001},
 {'n_neighbors': 70, 'balanced_accuracy': 0.5600643302747267},
 {'n_neighbors': 71, 'balanced_accuracy': 0.5617432271889845},
 {'n_neighbors': 72, 'balanced_accuracy': 0.5674144299502444},
 {'n_neighbors': 73, 'balanced_accuracy': 0.5630399576800995},
 {'n_neighbors': 74, 'balanced_accuracy': 0.5678219393440977},
 {'n_neighbors': 75, 'balanced_accuracy': 0.5655973678758706},
 {'n_neighbors': 76, 'balanced_accuracy': 0.5663929771522259},
 {'n_neighbors': 77, 'balanced_accuracy': 0.5683476918573426},
 {'n_neighbors': 78, 'balanced_accuracy': 0.5649599016050691},
 {'n_neighbors': 79, 'balanced_accuracy': 0.5703037967766694},
 {'n_neighbors': 80, 'balanced_accuracy': 0.5697340315202826},
 {'n_neighbors': 81, 'balanced_accuracy': 0.5702264107092452},
 {'n_neighbors': 82, 'balanced_accuracy': 0.564277279996679},
 {'n_neighbors': 83, 'balanced_accuracy': 0.5673289434214377},
 {'n_neighbors': 84, 'balanced_accuracy': 0.5689585643743597},
 {'n_neighbors': 85, 'balanced_accuracy': 0.5722217154031973},
 {'n_neighbors': 86, 'balanced_accuracy': 0.5649246007014843},
 {'n_neighbors': 87, 'balanced_accuracy': 0.5701193226449313},
 {'n_neighbors': 88, 'balanced_accuracy': 0.5702827576946465},
 {'n_neighbors': 89, 'balanced_accuracy': 0.5683590433240072},
 {'n_neighbors': 90, 'balanced_accuracy': 0.5688847766877598},
 {'n_neighbors': 91, 'balanced_accuracy': 0.5689720163302273},
 {'n_neighbors': 92, 'balanced_accuracy': 0.5696620910448992},
 {'n_neighbors': 93, 'balanced_accuracy': 0.5719220548252297},
 {'n_neighbors': 94, 'balanced_accuracy': 0.5721098872392114},
 {'n_neighbors': 95, 'balanced_accuracy': 0.5727270922342399},
 {'n_neighbors': 96, 'balanced_accuracy': 0.5708056959099765},
 {'n_neighbors': 97, 'balanced_accuracy': 0.5700645614486193},
 {'n_neighbors': 98, 'balanced_accuracy': 0.5730734125876973},
 {'n_neighbors': 99, 'balanced_accuracy': 0.5714835320094052},
 {'n_neighbors': 100, 'balanced_accuracy': 0.5710996858021758}]
```

n_neighbours=95

```
Model: "model_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_3 (InputLayer)        [(None, 207)]             0

 dense (Dense)               (None, 414)               86112

 batch_normalization_6 (Batc  (None, 414)              1656
 hNormalization)

 leaky_re_lu (LeakyReLU)     (None, 414)               0

 dense_1 (Dense)             (None, 207)               85905

 batch_normalization_7 (Batc  (None, 207)              828
 hNormalization)

 leaky_re_lu_1 (LeakyReLU)   (None, 207)               0

 dense_2 (Dense)             (None, 20)                4160

 dense_3 (Dense)             (None, 207)               4347

 batch_normalization_8 (Batc  (None, 207)              828
 hNormalization)

 leaky_re_lu_2 (LeakyReLU)   (None, 207)               0

 dense_4 (Dense)             (None, 414)               86112

 batch_normalization_9 (Batc  (None, 414)              1656
 hNormalization)

 leaky_re_lu_3 (LeakyReLU)   (None, 414)               0

 dense_5 (Dense)             (None, 207)               85905

=================================================================
Total params: 357,509
Trainable params: 355,025
Non-trainable params: 2,484
_________________________________________________________________
None
```

KNN (Multi-Class)

Accuracy:0.96
Balanced accuracy:0.63
Precision:0.98
Recall:0.96
F1-score:0.97

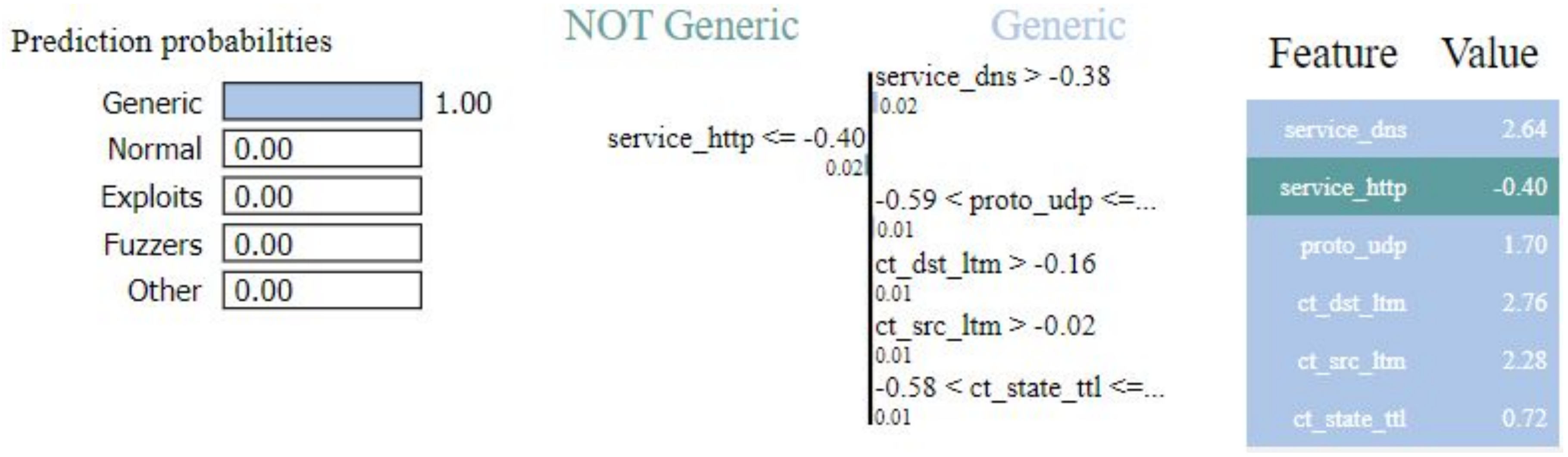


KNN (Multi-Class)

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	0.98	0.99	221846
Generic	1.00	0.97	0.99	21548
Exploits	0.70	0.40	0.51	4452
Fuzzers	0.39	0.75	0.51	2425
DoS	0.31	0.55	0.40	1635
Reconnaissance	0.54	0.62	0.58	1399
Analysis	0.07	0.29	0.11	268
Backdoor	0.03	0.07	0.04	233
Shellcode	0.42	0.89	0.57	151
Worms	0.02	0.82	0.03	17
accuracy			0.96	253974
macro avg	0.45	0.63	0.47	253974
weighted avg	0.98	0.96	0.97	253974

LIME For Multiclass

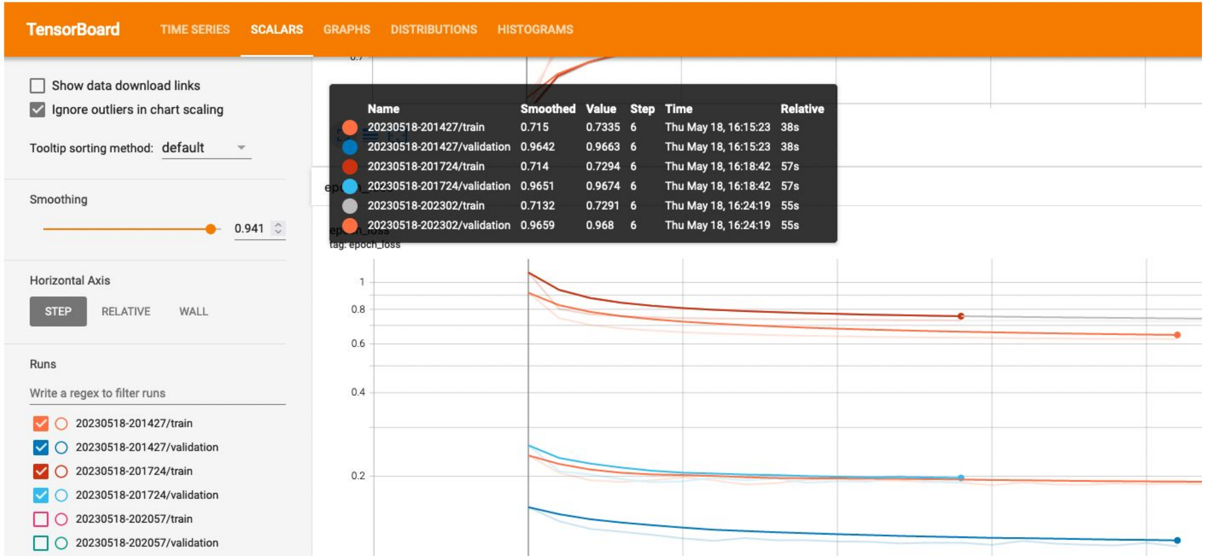


Part 3: Deep Learning

Baseline Neural Network for Multi-Class Case

Sequential Model: KerasTuner + TensorBoard

- Simple Architecture (6 hidden layers)
- Relu activation on hidden layers
- Train data upsampled w/SMOTE
- Adam optimizer
- 1024 batch size
- 0.001 learning rate



Sequential Model: KerasTuner + TensorBoard

- Simple Architecture (6 hidden layers)
- L2 regularization
- Tanh activation on hidden layers
- Train data upsampled w/SMOTE
- Adam optimizer

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 22)]              0

 dense (Dense)               (None, 512)               11776

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 512)               262656

 dropout_1 (Dropout)         (None, 512)               0

 dense_2 (Dense)             (None, 256)               131328

 dropout_2 (Dropout)         (None, 256)               0

 dense_3 (Dense)             (None, 256)               65792

 dropout_3 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 128)               32896

 dropout_4 (Dropout)         (None, 128)               0

 dense_5 (Dense)             (None, 128)               16512

 dropout_5 (Dropout)         (None, 128)               0

 dense_6 (Dense)             (None, 10)                1290

=================================================================
Total params: 522,250
Trainable params: 522,250
Non-trainable params: 0
_________________________________________________________________
```

Sequential Model: KerasTuner + TensorBoard

- Simple Architecture (6 hidden layers)
- L2 regularization
- Tanh activation on hidden layers
- Train data upsampled w/SMOTE
- Adam optimizer

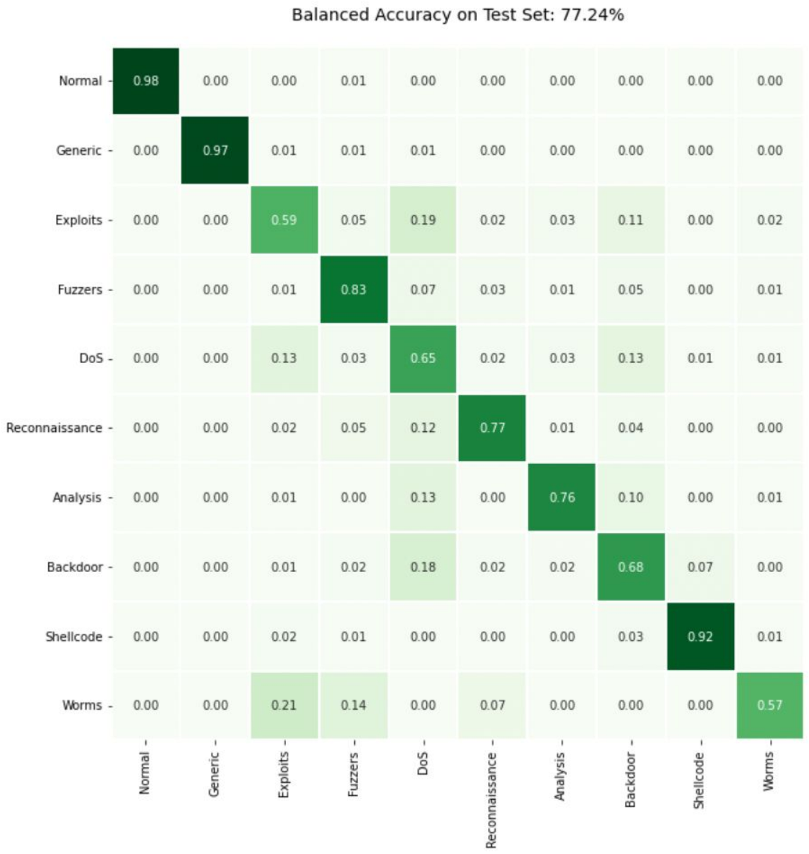
Ensemble of One vs All Classifiers

- 10 Small Neural Nets for each class
- Weighted Voting Classifier

```
Accuracy:0.97
Balanced accuracy:0.77
Precision:0.98
Recall:0.97
F1-score:0.98
```

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	0.98	0.99	135296
Generic	1.00	0.97	0.99	13285
Exploits	0.79	0.59	0.67	2719
Fuzzers	0.41	0.83	0.55	1466
DoS	0.42	0.65	0.51	981
Reconnaissance	0.78	0.77	0.77	857
Analysis	0.25	0.76	0.38	157
Backdoor	0.12	0.68	0.21	133
Shellcode	0.30	0.92	0.46	91
Worms	0.06	0.57	0.12	14
accuracy			0.97	154999
macro avg	0.51	0.77	0.56	154999
weighted avg	0.98	0.97	0.98	154999



Part 3: Deep Learning

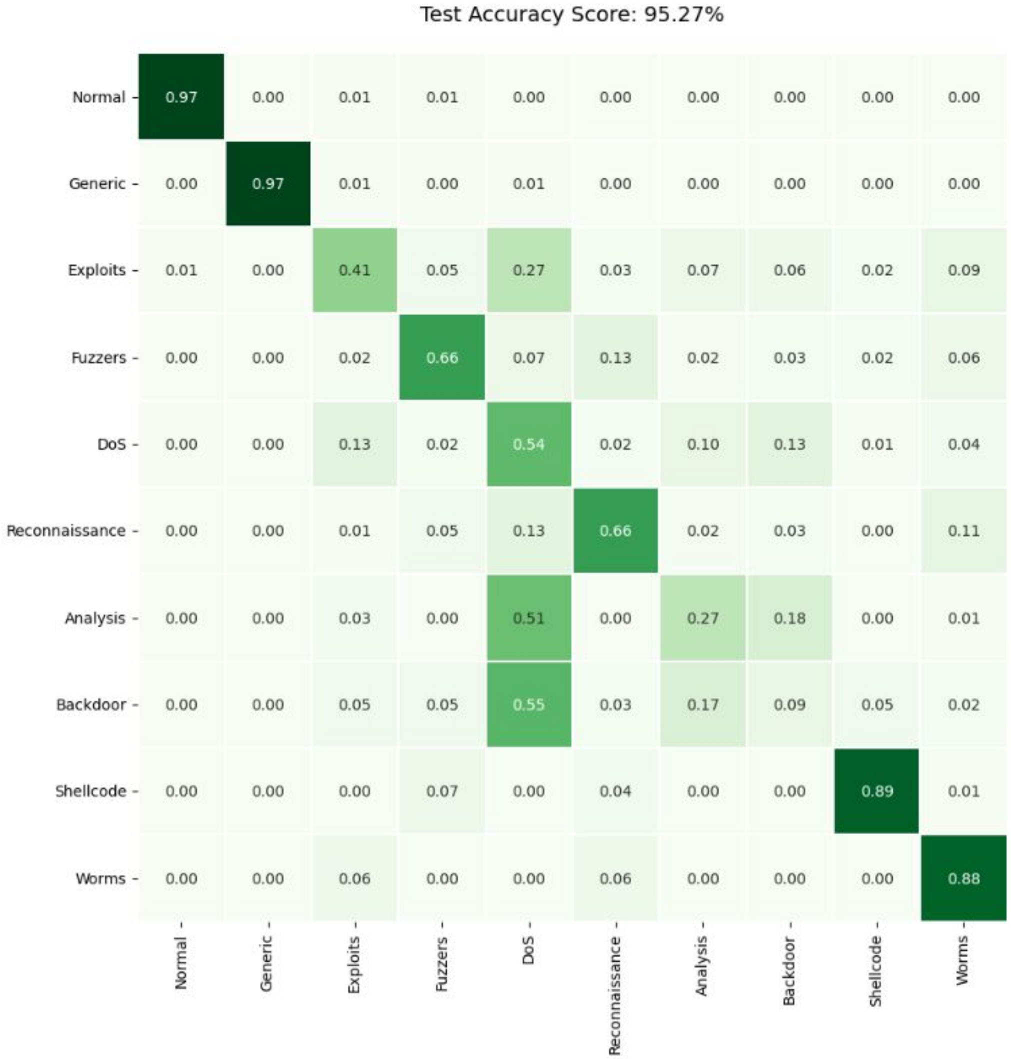
Variational Autoencoder

Variational Autoencoder

```
Model: "model_2"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_2 (InputLayer)           [(None, 207)]        0           []

 batch_normalization_4 (BatchNo  (None, 207)         828         ['input_2[0][0]']
 rmalization)

 encoder_hidden (Dense)         (None, 25)           5200        ['batch_normalization_4[0][0]']

 batch_normalization_5 (BatchNo  (None, 25)          100         ['encoder_hidden[0][0]']
 rmalization)

 z_mean (Dense)                 (None, 20)           520         ['batch_normalization_5[0][0]']

 z_log_var (Dense)              (None, 20)           520         ['batch_normalization_5[0][0]']

 z_sampled (Lambda)             (None, 20)           0           ['z_mean[0][0]',
                                                                  'z_log_var[0][0]']

 decoder_hidden (Dense)         (None, 25)           525         ['z_sampled[0][0]']

 decoded_mean (Dense)           (None, 207)          5382        ['decoder_hidden[0][0]']

==================================================================================================
Total params: 13,075
Trainable params: 12,611
Non-trainable params: 464
__________________________________________________________________________________________________
```

KNN With Variational Autoencoder

Accuracy:0.95
Balanced accuracy:0.634
Precision:0.97
Recall:0.95
F1-score:0.96



KNN With Variational Autoencoder

Classification Report:

	precision	recall	f1-score	support
Normal	1.00	0.97	0.99	221846
Generic	0.99	0.97	0.98	21548
Exploits	0.38	0.41	0.39	4452
Fuzzers	0.35	0.66	0.45	2425
DoS	0.30	0.54	0.38	1635
Reconnaissance	0.52	0.66	0.58	1399
Analysis	0.07	0.27	0.11	268
Backdoor	0.03	0.09	0.05	233
Shellcode	0.30	0.89	0.44	151
Worms	0.01	0.88	0.03	17
accuracy			0.95	253974
macro avg	0.40	0.63	0.44	253974
weighted avg	0.97	0.95	0.96	253974

Contributions. Repository on GitHub. Commits

April 18, 2023 – May 18, 2023

Period: 1 month

Overview

16 Active pull requests

6 Active issues

16 Merged pull requests	0 Open pull requests	6 Closed issues	0 New issues

Excluding merges, **4 authors** have pushed **76 commits** to main and **76 commits** to all branches. On main, **57 files** have changed and there have been **131,063 additions** and **19 deletions**.

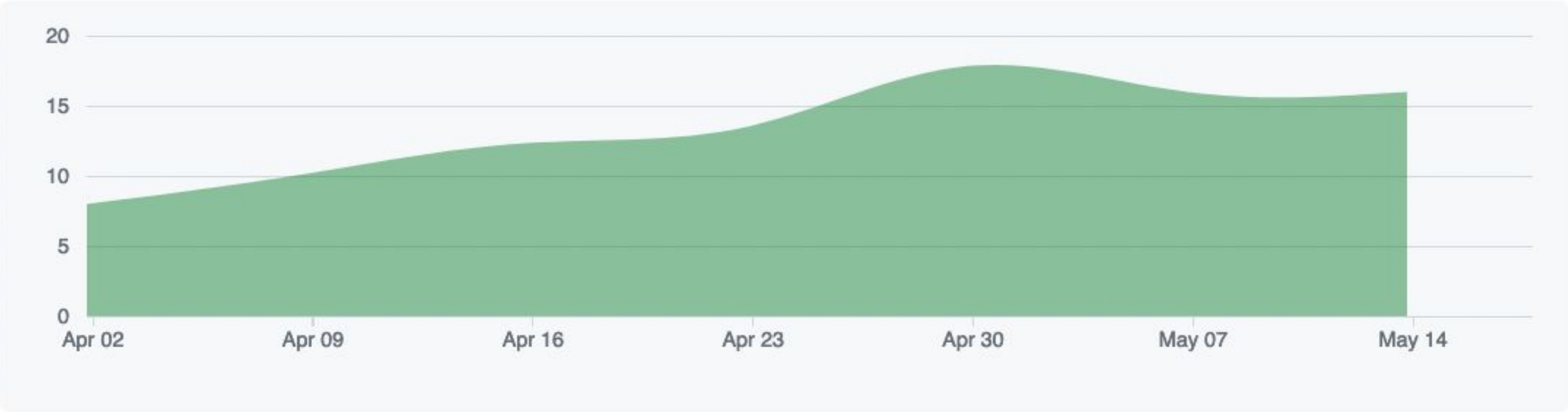
16 Pull requests merged by 3 people

Contributions. Repository on GitHub. Commits

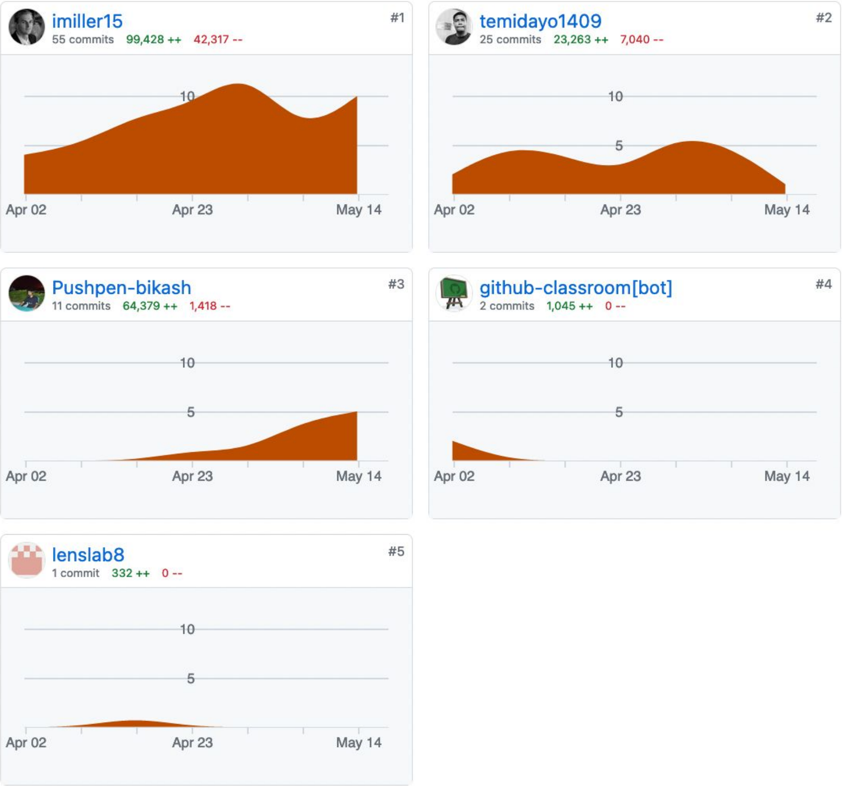
Apr 2, 2023 – May 18, 2023

Contributions: Commits

Contributions to main, excluding merge commits and bot accounts



Repository on GitHub. Individual Contributions



Repository on [GitHub](). New Code Over Time