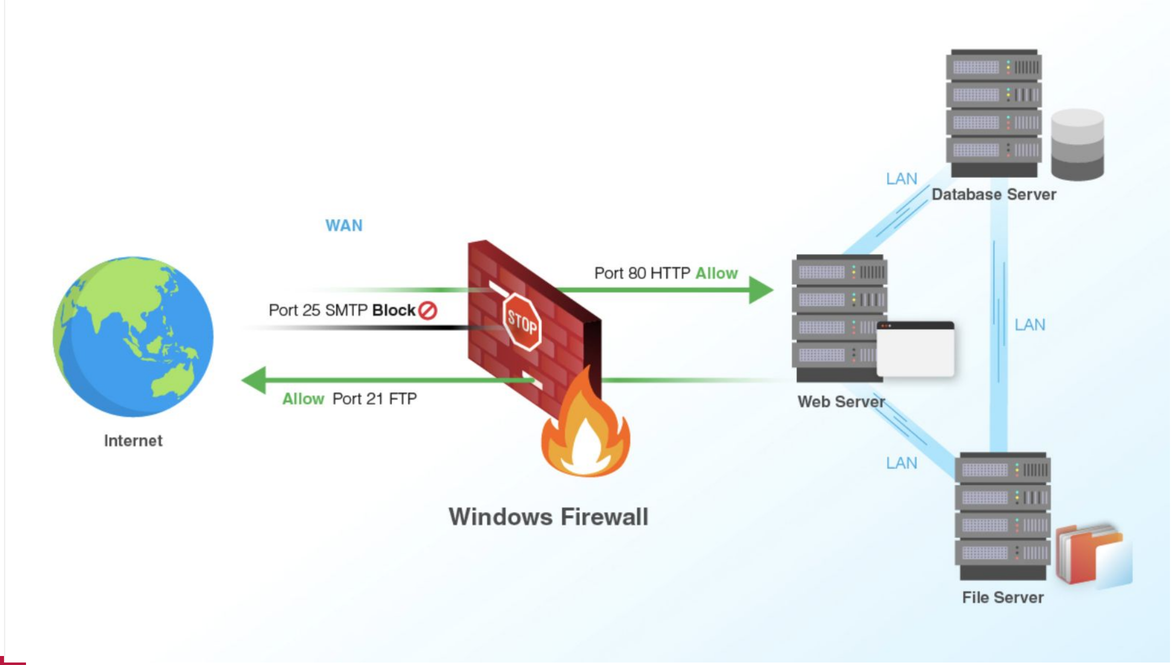
Classification of Firewall Logs

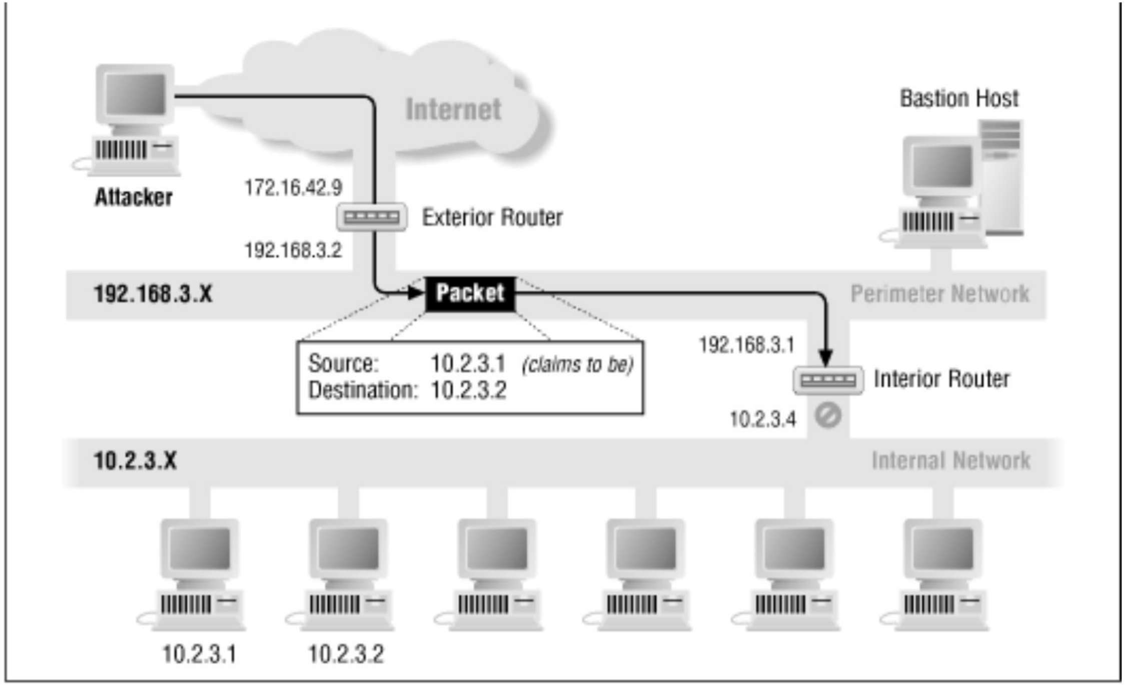
Dalya Manatova

What and How Firewalls work



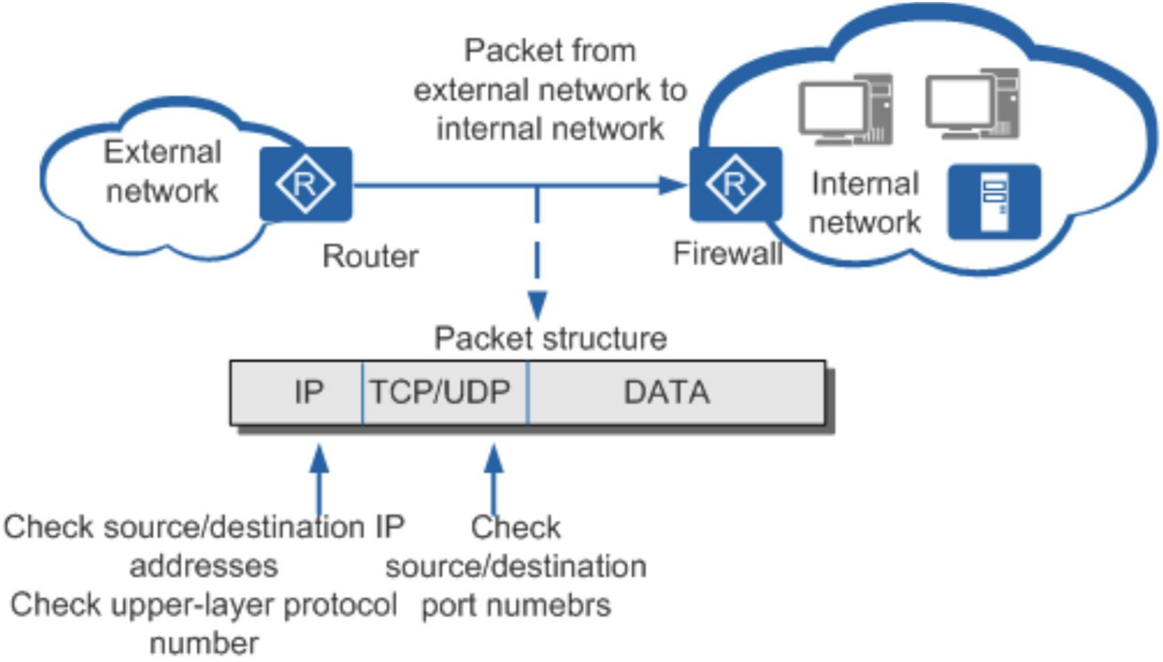
What and How Firewalls work

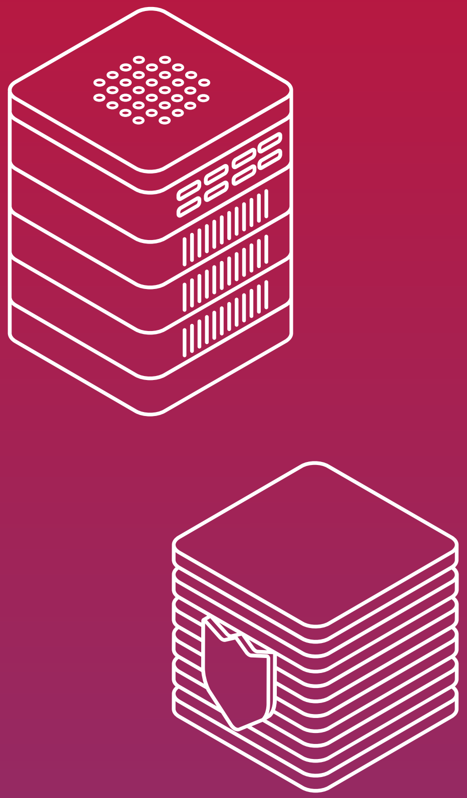
- **Packet filtering** - covering in this dataset



What and How Firewalls work

- **Packet filtering** - covering in this dataset





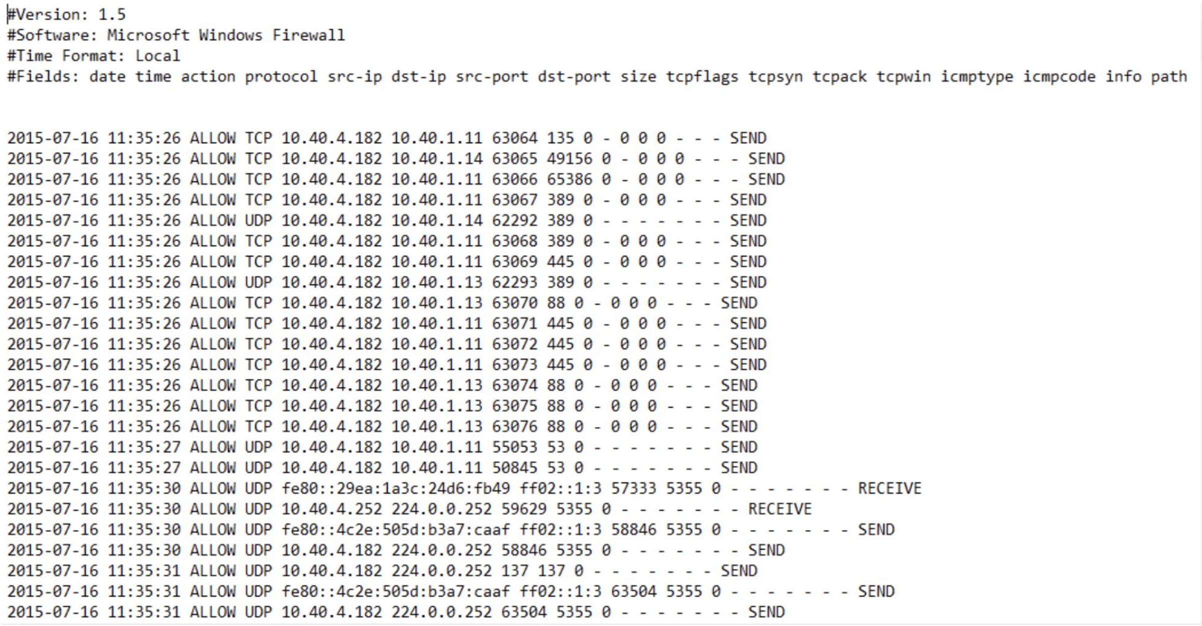
Why it’s so important?

Firewalls can help to prevent a number of different security risks, but usually monitoring and managing them takes a lot of human resources

- Backdoors in apps
- Denial of service - Flooding a server
- Macros running on background
- Remote unauthorized logins
- Spam
- Viruses

Data and related work

Data of Firewall logs



```
#Version: 1.5
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn tcpack tcpwin icmptype icmpcode info path


2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63064 135 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.14 63065 49156 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63066 65386 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63067 389 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW UDP 10.40.4.182 10.40.1.14 62292 389 0 - - - - - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63068 389 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63069 445 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW UDP 10.40.4.182 10.40.1.13 62293 389 0 - - - - - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.13 63070 88 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63071 445 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63072 445 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63073 445 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.13 63074 88 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.13 63075 88 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.13 63076 88 0 - 0 0 0 - - - SEND
2015-07-16 11:35:27 ALLOW UDP 10.40.4.182 10.40.1.11 55053 53 0 - - - - - - - SEND
2015-07-16 11:35:27 ALLOW UDP 10.40.4.182 10.40.1.11 50845 53 0 - - - - - - - SEND
2015-07-16 11:35:30 ALLOW UDP fe80::29ea:1a3c:24d6:fb49 ff02::1:3 57333 5355 0 - - - - - - - RECEIVE
2015-07-16 11:35:30 ALLOW UDP 10.40.4.252 224.0.0.252 59629 5355 0 - - - - - - - RECEIVE
2015-07-16 11:35:30 ALLOW UDP fe80::4c2e:505d:b3a7:caaf ff02::1:3 58846 5355 0 - - - - - - - SEND
2015-07-16 11:35:30 ALLOW UDP 10.40.4.182 224.0.0.252 58846 5355 0 - - - - - - - SEND
2015-07-16 11:35:31 ALLOW UDP 10.40.4.182 224.0.0.252 137 137 0 - - - - - - - SEND
2015-07-16 11:35:31 ALLOW UDP fe80::4c2e:505d:b3a7:caaf ff02::1:3 63504 5355 0 - - - - - - - SEND
2015-07-16 11:35:31 ALLOW UDP 10.40.4.182 224.0.0.252 63504 5355 0 - - - - - - - SEND
```

Data ~65K rows (~30s logs)

F1 Score: 0.76 in the source Challenge-beat the score

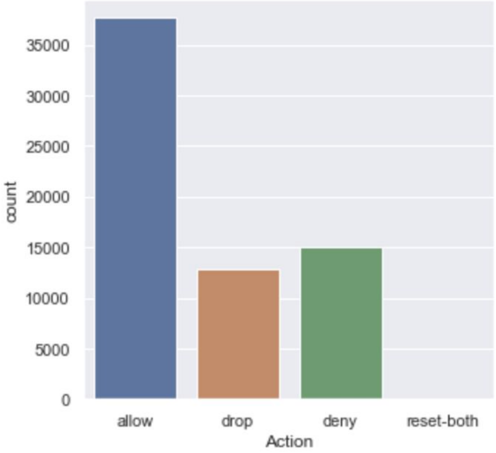
Field Name	Description
'Source Port'	Client Source Port
'Destination Port'	Client Destination Port
'NAT Source'	Network Address Translation Source (masked)
'NAT Destination’	Network Address Translation Destination (masked)
'Bytes Sent'	Bytes Sent
'Bytes Received',	Bytes Received
'Elapsed Time (sec)'	Elapsed Time for flow
'Pkts_sent'	Packets Sent
‘pkts_received’	Packets Received
‘Allow’ - label	Class (allow, deny, drop, reset-both)

Source: Ertam and M. Kaya,Classification of firewall log files with multiclass support vector machine, 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, 2018, pp. 1-4, doi: 10.1109/ISDFS.2018.8355382

Data imbalance problem

Labels:

- allow—session was allowed by policy
- deny—session was denied by policy
- drop—session was dropped silently
- reset —session was terminated and a TCP reset is sent to both the sides of the connection



Processing and model selection

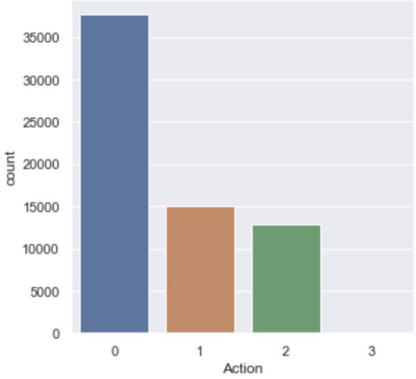
Ports and Addresses

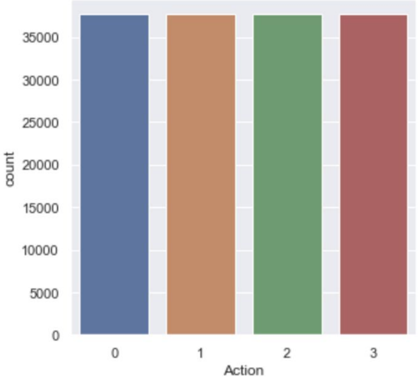
- Ports and Addresses are in fact categorical
- Theoretically ~65 000
- Should be converted to dummy variables

- Problem visualizing and memory usage with high dimensionality
- Sparse matrix is used
- OneHotCoding

Data imbalance

- SMOTE - oversampling underrepresented classes





Models

SVM (linear)

- NO Smote and NO OneHotEncoding
- Using OneHotEncoding
- Using Both (SMOTE and OneHotEncoding)

KNN

- NO Smote and NO OneHotEncoding
- Using OneHotEncoding
- Using Both (SMOTE and OneHotEncoding)

Random Forest (10 trees)

- NO Smote and NO OneHotEncoding
- Using OneHotEncoding
- Using Both (SMOTE and OneHotEncoding)

Metrics for evaluation: Recall (macro) and **Precision** (macro) and F1 score (macro)

Note: since time efficiency is important, models that are too time costly are avoided on purpose

Results

SMOTE and One-hot encoding

SVM ~6min

- Recall SVM: 0.918
- Precision SVM: 0.922
- F1 score SVM: 0.917

KNN ~ 5min

- Recall KNeighborsClassifier: 0.855
- Precision KNeighborsClassifier: 0.876
- F1 score KNeighborsClassifier: 0.848

Random Forest ~ 39.4s

- Recall: 0.989
- Precision: 0.989
- F1 score: 0.989

One-hot encoding only

SVM ~ 14.2s

- Recall SVM: 0.859
- Precision SVM: 0.934
- F1 score SVM: 0.887

KNN ~25.5s

- Recall KNeighborsClassifier: 0.821
- Precision KNeighborsClassifier: 0.931
- F1 score KNeighborsClassifier: 0.856

Random Forest ~3.26 s

- Recall: 0.950
- Precision: 0.999
- F1 score: 0.972

Basic

SVM

- Recall SVM: 0.738
- Precision SVM: 0.720
- F1 score SVM: 0.715

KNN ~1s

- Recall KNeighborsClassifier: 0.742
- Precision KNeighborsClassifier: 0.736
- F1 score KNeighborsClassifier: 0.739

Random Forest ~138s

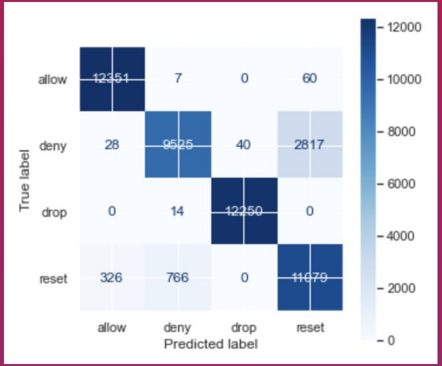
- Recall: 0.925
- Precision: 0.999
- F1 score: 0.955

SVM

SMOTE and One-hot encoding

SVM ~6min

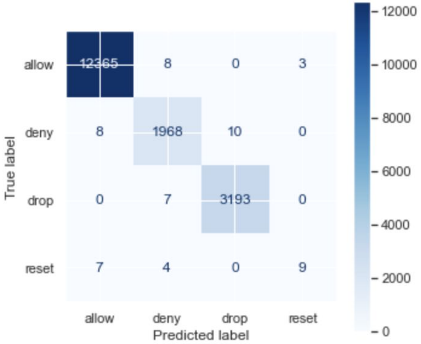
- Recall SVM: 0.918
- Precision SVM: 0.922
- F1 score SVM: 0.917



One-hot encoding only

SVM ~ 14.2s

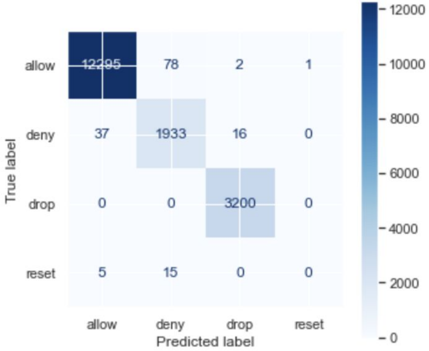
- Recall SVM: 0.859
- Precision SVM: 0.934
- F1 score SVM: 0.887



Basic

SVM

- Recall SVM: 0.738
- Precision SVM: 0.720
- F1 score SVM: 0.715

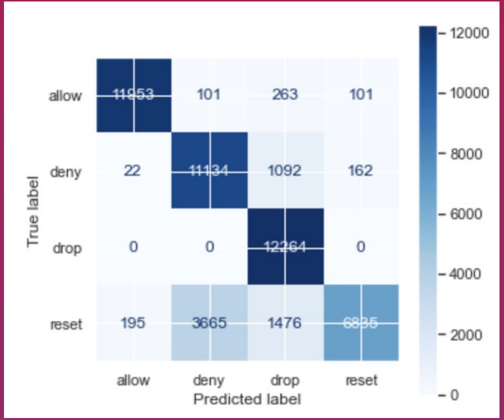


KNN

SMOTE and One-hot encoding

KNN ~ 5min

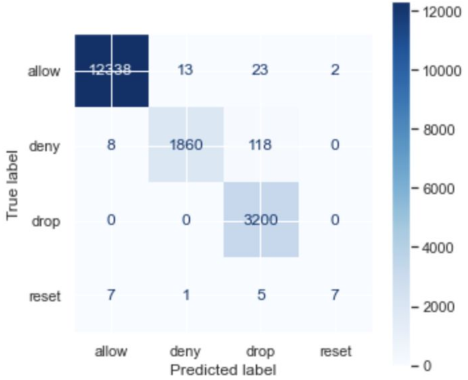
- Recall KNeighborsClassifier: 0.855
- Precision KNeighborsClassifier: 0.876
- F1 score KNeighborsClassifier: 0.848



One-hot encoding only

KNN ~25.5s

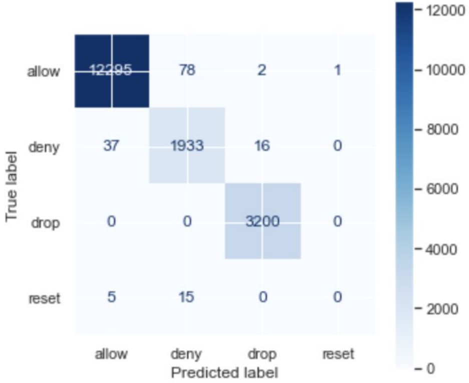
- Recall KNeighborsClassifier: 0.821
- Precision KNeighborsClassifier: 0.931
- F1 score KNeighborsClassifier: 0.856



Basic

KNN ~1s

- Recall KNeighborsClassifier: 0.742
- Precision KNeighborsClassifier: 0.736
- F1 score KNeighborsClassifier: 0.739

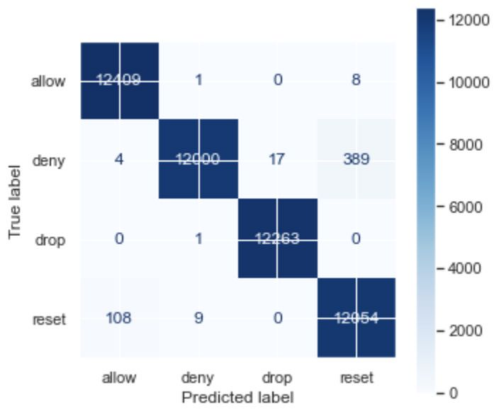


Random Forest

SMOTE and One-hot encoding

Random Forest ~ 39.4s

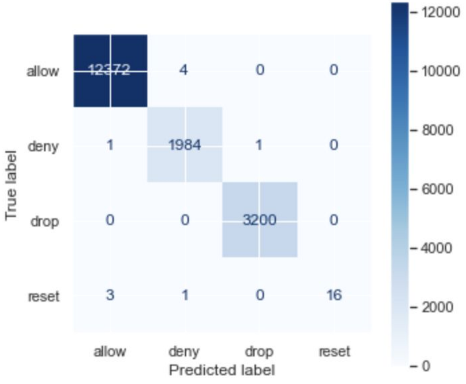
- Recall: 0.989
- Precision: 0.989
- F1 score: 0.989



One-hot encoding only

Random Forest ~3.26 s

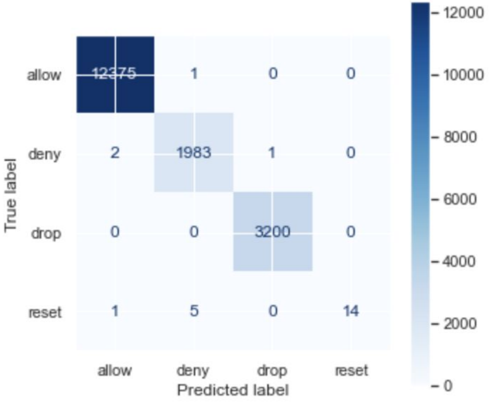
- Recall: 0.950
- Precision: 0.999
- F1 score: 0.972



Basic

Random Forest ~138s

- Recall: 0.925
- Precision: 0.999
- F1 score: 0.955



Future work

The research can be replicated at IU logs and with enough GPU resources and data - anomaly detection can be a further step for exploring

Thank you

Thank you