IoT attack classification

EMLET project

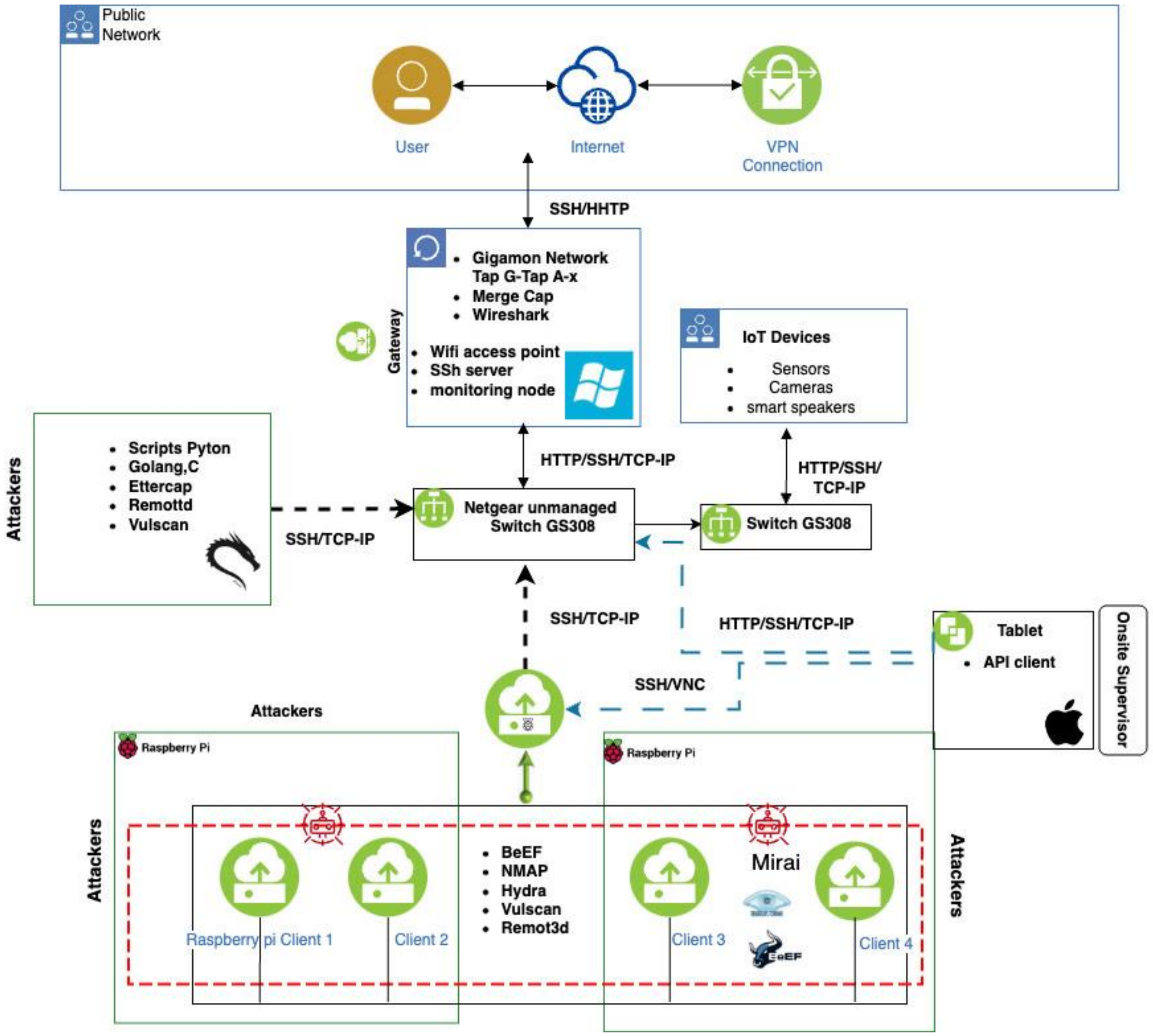
Sławomir Batruch

Introduction

- Aim – study feasibility of ML solutions usage in network security
- CICIoT2023 – big dataset of attack and benign IoT packet features
- Tech stack – Scikit-learn, Imbalanced-learn, XGBoost, Pytorch, Matplotlib, Seaborn, Numpy, Pandas

Dataset

- Testbed – a controlled environment of IoT devices with vulnerabilities, exposed to attackers
- Feature extraction – chosen packet and flow characteristics extracted from .pcap files into .csv



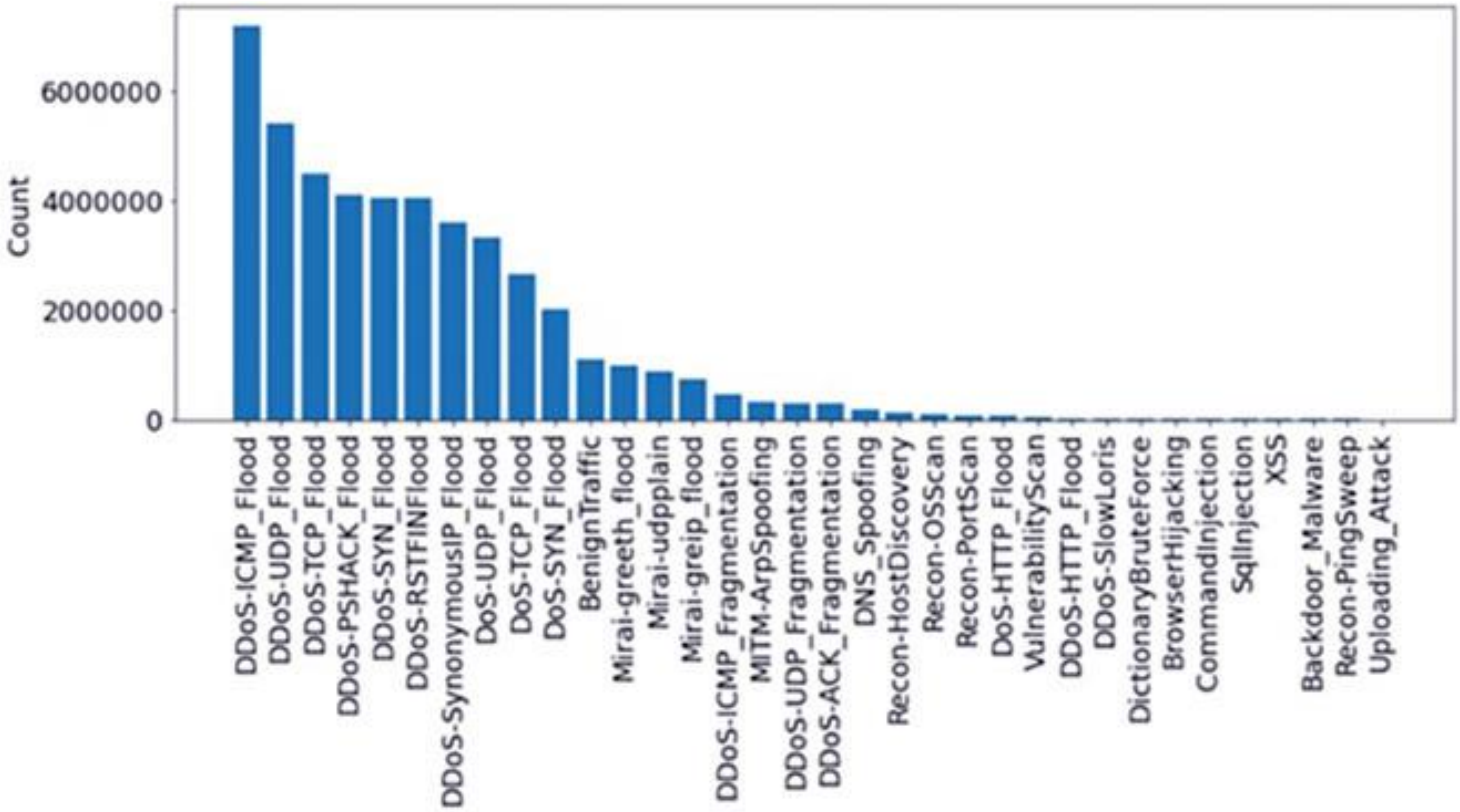
Packet features

- Chosen and extracted by the dataset provider
- Protocol type (HTTP, DNS, ICMP, ...)
- TCP flag information (SYN, RST, FIN, ...)
- Flow characteristics (duration, rate, lengths, time deltas, counts)
- Packet labels correspond to different attack types and subtypes

Labels

- (D)DoS - (Distributed) Denial of Service
  - ICMP, UDP, TCP, HTTP … flooding
  - Fragmentation
  - Slow Loris
- Mirai botnet - (GRE, UDP flooding)
- Spoofing (ARP, DNS)
- Reconnaissance
  - OS, Port, Vulnerability scan
  - Ping sweep
- Various web attacks (SQL injection, XSS, malware …)
- Benign (harmless) packets
- Total 34 classes, can be remapped to 8 or 2 classes (binary)
- **Considerable imbalance due to nature of (D)DoS attacks**

Label counts



ML algorithm evaluation

- 3 scenarios: 34 classes, 8 classes (remapped), 2 classes (binary)
- 4 chosen machine learning methods were evaluated:
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - eXtreme Gradient Boost
- Only 25% of the dataset was used due to the large size
- SMOTE + under-sampling vs. no preprocessing
- Other algorithms were tested, but are not analysed due to low scores (SGD, k-neighbors, ANN (torch))

ML algorithm evaluation – metrics

- Two modes of metrics were considered – macro and weighted
- Accuracy – rate of correct predictions
- Precision – score to determine false positives susceptibility
- Recall – score to determine false negatives susceptibility
- F1 – harmonic mean of precision and recall
- Main evaluation metric is f1 score (macro), not accuracy
- Confusion matrix is analysed to see algorithm prediction trends

Results – 34 classes

34 CLASSES					
Metric type	Model	Accuracy	Recall	Precision	F1
MACRO	LogisticRegresion	0.801	0.587	0.482	0.487
WEIGHTED	LogisticRegresion	0.801	0.801	0.912	0.836
MACRO	Decision Tree	0.993	0.819	0.832	0.824
WEIGHTED	Decision Tree	0.993	0.993	0.993	0.993
MACRO	Random Forest	0.992	0.851	0.718	0.729
WEIGHTED	Random Forest	0.992	0.992	0.994	0.993
MACRO	eXtreme Gradient Boosting	0.992	0.794	0.726	0.740
WEIGHTED	eXtreme Gradient Boosting	0.992	0.992	0.993	0.993

34 CLASSES; PREPROCESSED					
Metric type	Model	Accuracy	Recall	Precision	F1
MACRO	LogisticRegresion	0.788	0.505	0.559	0.496
WEIGHTED	LogisticRegresion	0.788	0.788	0.828	0.798
MACRO	Decision Tree	0.991	0.771	0.831	0.794
WEIGHTED	Decision Tree	0.991	0.991	0.990	0.990
MACRO	Random Forest	0.991	0.759	0.767	0.759
WEIGHTED	Random Forest	0.991	0.991	0.991	0.991
MACRO	eXtreme Gradient Boosting	0.994	0.835	0.827	0.829
WEIGHTED	eXtreme Gradient Boosting	0.994	0.994	0.994	0.994

Results – 8 classes

8 CLASSES					
Metric type	Model	Accuracy	Recall	Precision	F1
MACRO	LogisticRegresion	0.831	0.758	0.512	0.540
WEIGHTED	LogisticRegresion	0.831	0.831	0.952	0.878
MACRO	Decision Tree	0.994	0.822	0.837	0.828
WEIGHTED	Decision Tree	0.994	0.994	0.994	0.994
MACRO	Random Forest	0.995	0.934	0.716	0.735
WEIGHTED	Random Forest	0.995	0.995	0.996	0.995
MACRO	eXtreme Gradient Boosting	0.995	0.838	0.730	0.752
WEIGHTED	eXtreme Gradient Boosting	0.995	0.995	0.995	0.995

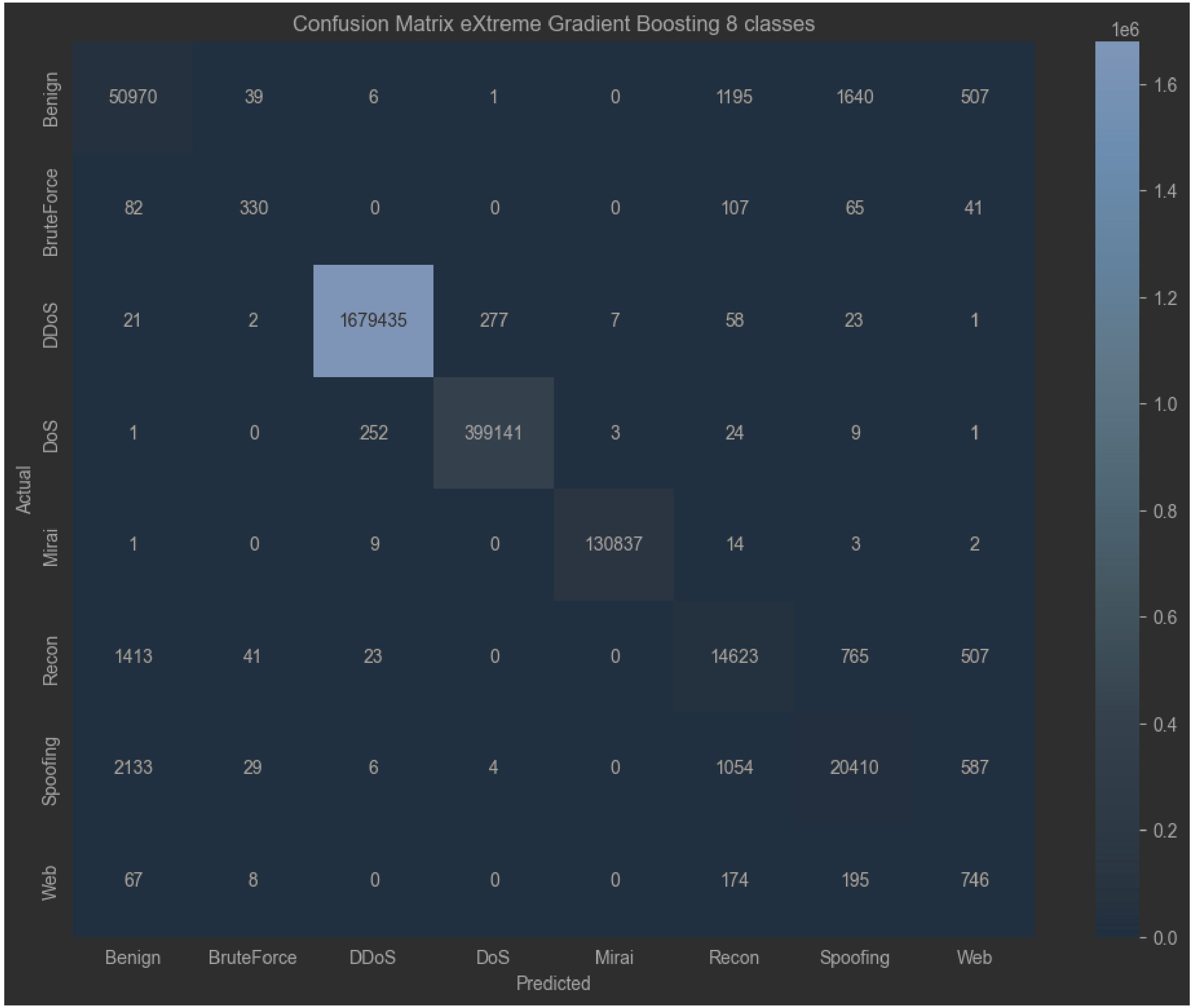
8 CLASSES; PREPROCESSED					
Metric type	Model	Accuracy	Recall	Precision	F1
MACRO	LogisticRegresion	0.809	0.561	0.545	0.490
WEIGHTED	LogisticRegresion	0.809	0.809	0.893	0.837
MACRO	Decision Tree	0.994	0.780	0.840	0.803
WEIGHTED	Decision Tree	0.994	0.994	0.993	0.993
MACRO	Random Forest	0.994	0.819	0.780	0.784
WEIGHTED	Random Forest	0.994	0.994	0.994	0.994
MACRO	eXtreme Gradient Boosting	0.995	0.839	0.847	0.834
WEIGHTED	eXtreme Gradient Boosting	0.995	0.995	0.995	0.995

Results – 2 classes

2 CLASSES					
Metric type	Model	Accuracy	Recall	Precision	F1
MACRO	LogisticRegresion	0.989	0.891	0.864	0.877
WEIGHTED	LogisticRegresion	0.989	0.989	0.990	0.989
MACRO	Decision Tree	0.996	0.956	0.957	0.956
WEIGHTED	Decision Tree	0.996	0.996	0.996	0.996
MACRO	Random Forest	0.997	0.965	0.971	0.968
WEIGHTED	Random Forest	0.997	0.997	0.997	0.997
MACRO	eXtreme Gradient Boosting	0.997	0.959	0.973	0.966
WEIGHTED	eXtreme Gradient Boosting	0.997	0.997	0.997	0.997

2 CLASSES; PREPROCESSED					
Metric type	Model	Accuracy	Recall	Precision	F1
MACRO	LogisticRegresion	0.982	0.952	0.637	0.708
WEIGHTED	LogisticRegresion	0.982	0.982	0.994	0.987
MACRO	Decision Tree	0.995	0.957	0.939	0.948
WEIGHTED	Decision Tree	0.995	0.995	0.995	0.995
MACRO	Random Forest	0.996	0.976	0.938	0.957
WEIGHTED	Random Forest	0.996	0.996	0.996	0.996
MACRO	eXtreme Gradient Boosting	0.997	0.969	0.963	0.966
WEIGHTED	eXtreme Gradient Boosting	0.997	0.997	0.997	0.997

Example confusion matrix



Conclusions

- Decision tree and random forest manage relatively well even on an unbalanced data set
- After preprocessing dataset (under-sampling + SMOTE), the results are slightly better
- Classical ML algorithms performed better than ANN.
  - More complicated ANN design, hyperparameter tuning and further data preprocessing might be needed for ANN to compete with other solutions
- Overall, application of ML for packet analysis can be a good enhancement to existing systems

Thank you for your attention