Gépi tanulási módszerek a behatolás felderítésében

IDS-modell építése gépi tanulással

Vágujhelyi Ferenc

Hírközlési és Informatikai Tudományos Egyesület

Az IDS legyen inkább saját? (Ne!)

- Kiváló IDS-termékek és szolgáltatások érhetők el a piacon, megfelelő szakértelemmel támogatva. Miért kell azzal foglalkoznunk, hogy mi hogyan fejlesszünk egyet, amikor valószínűleg nem fogunk?

- A piacon kapható termékek szinte bárki számára, így a potenciális támadók számára is elérhetők. Képesek elemezni működésüket, és képesek tervezett támadásaikat tesztelni. Amatőr eszközeink sokszor szuboptimális döntéseink miatt deviánsnak osztályozhatnak olyan hálózati műveleteket, amelyeket, alapos megfontolás után a profi fejlesztők jóindulatúnak osztályoztak.
  A meglévő rendszerek mellé bizonyos esetekben megfontolandó időnként saját modell feltanítása és futtatása.

Szaktudás nélküli osztályozás

- Amikor gépi tanulással behatolásdetektáló modellt építünk, a hálózati forgalomból vett tanítóadatokban rejlő statisztikai tulajdonságok alapján osztályozunk, és nem szándékunk annak felismerése, hogy valójában milyen műveleteket hajt végre a támadó vagy a támadást végrehajtó programkód. Kérdésünk az, hogy **"Jelen van?"**, nem pedig az, hogy **"Mi történik?"**.

- A vizsgált attribútumok nevét véletlenszámokkal helyettesíthetjük!

- Mindegy, hogy hálózati támadás vagy adócsalás detektálása a cél.

Szakmai cikkek

Cikkek a Cyber Intrusion Detection Using Machine Learning témában leírják, hogy

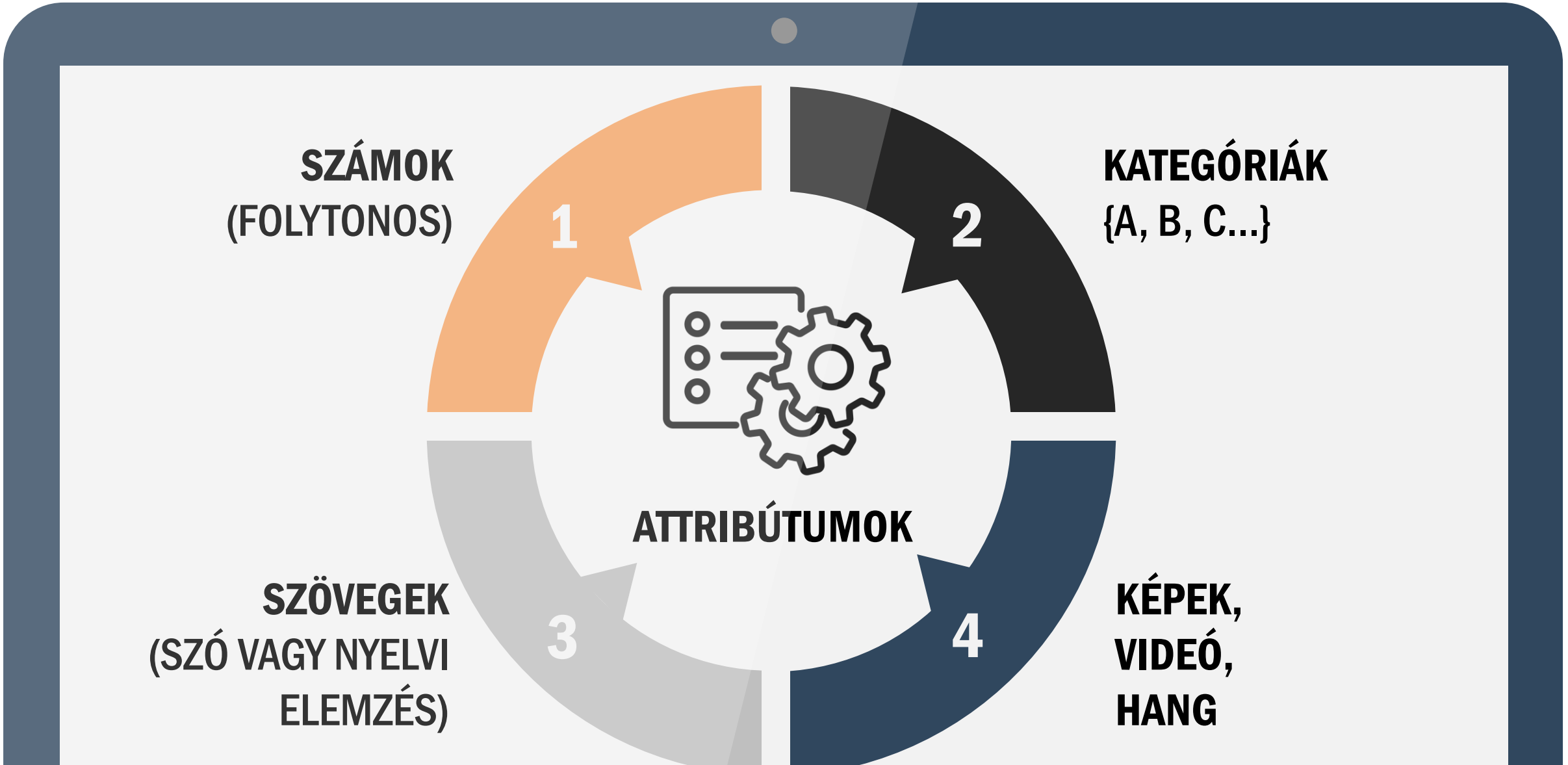
- mi az ID és, hogy a cél gépi tanulásra épülő Intrusion Detection System kifejlesztése
- ismertetik a főbb támadási osztályokat
- leírják a gépi tanulás fő modellező eljárásait (Bayesian Network, Naive Bayes classifier, Decision
- Tree, Random Decision Forest, Random Tree, Decision Table, Artificial Neural Network, k nearest neighbor...) **paraméterek nélkül!**
- definiálnak egy teljesítménymetrikát ( )
- felrajzolnak egy diagramot, hogy a különböző modellek hogyan teljesítenek (esetleg a különböző
- támadási osztályokon)

Mi kell a reprodukálhatósághoz?

- **adat-mintavételezés** (preferált!) vagy letölthető tanítóadatok, címkézettek vagy címkék kiválasztása
  (pl. adó regisztrációs e-mail-címhez hány k-nál több regisztráció tartozik?)
- támadási fajták praktikus kategóriákba csoportosítása (minden csoporthoz legyen sok, gazdagon feltöltött feature-mátrix)
- előfeldolgozás: duplikációk, hiányzó adatok kezelése, számmá alakítás, normalizálás, súlyozás,
- feature selection: az osztályoktól független feature-ök eldobása, csak a legjobb pl. 30-50 kiválasztása
- **szakmai jelentéstani leírásuk, mérnöki szempontok alapján az eddigiek felülvizsgálata**
- Tanító-, teszt- és értékelő adatok reprezentatív szétválasztása
- kiválasztott ML-eszközök (pl. scikit-learn, Pytorch, Keras-Tensorflow stb, esetleg saját ML fejlesztés!)
- a fentiek és a modellfeltanítás programozása
- modellek feltanítása, tesztelése, vizualizáció, ahol lehet
- **támadások generálása, osztályozás helyességének mérése, felismert zero-day támadásokból post-ot / cikket írni!**
- elemzés, képességek és korlátok felismerése, bemutatása
- a kifejlesztett eszköz produktív használatbavétele (tapasztalatok bemutatása)

Hogyan elemzünk?

{#define SPACE_METRICS}



Hogyan elemzünk?

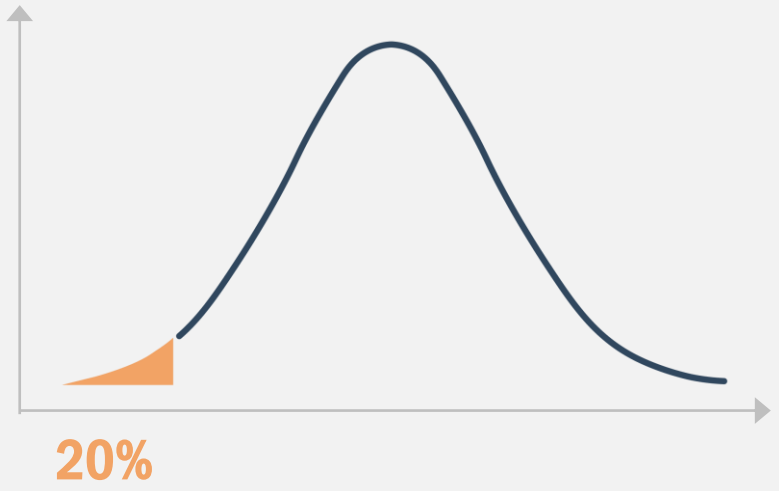
{#define SPACE_METRICS}

$$(1)\quad d_j(X_k, X_l) = \begin{cases} 0 & \text{if } X_k^j, X_l^j \text{belong to the same class.} \\ 1 & \text{else} \end{cases}$$

$$(2)\quad d_j(X_k, X_l) = \frac{|X_k^j - X_l^j|^{w_j}}{r_j^{w_j}}$$

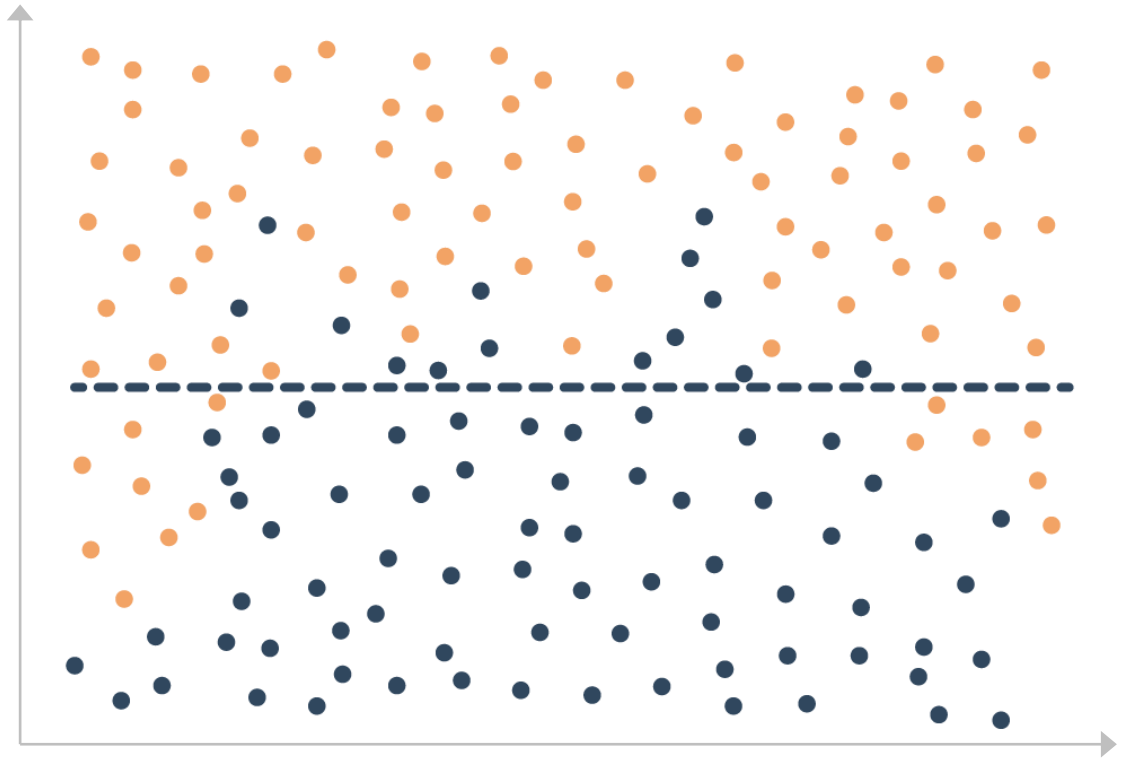
$$(3)\quad d(X_k, X_l) = \sum_j \alpha_j d_j(X_k, X_l)$$

$$(4)\quad A_{kl} = d(X_k, X_l)$$



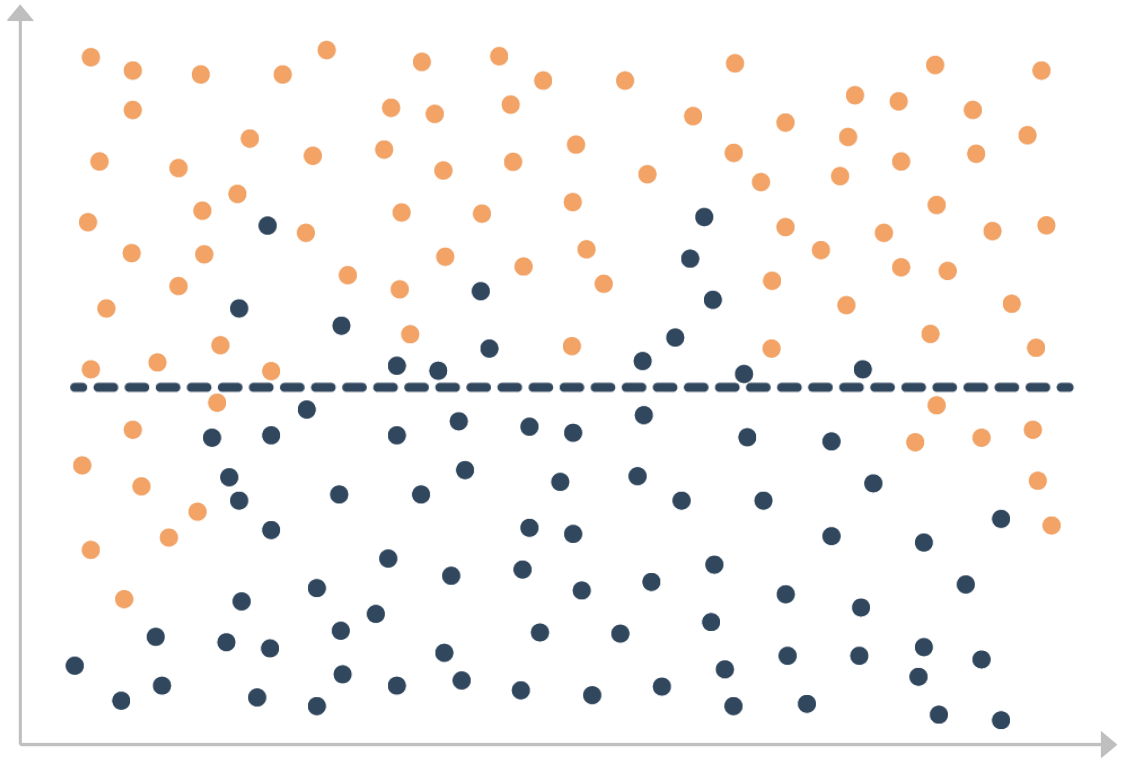
Kontrollált gépi tanulás

UNDERFITTING



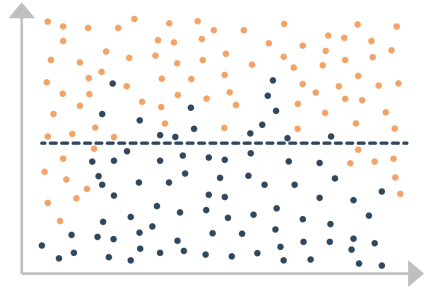
Kontrollált gépi tanulás

UNDERFITTING

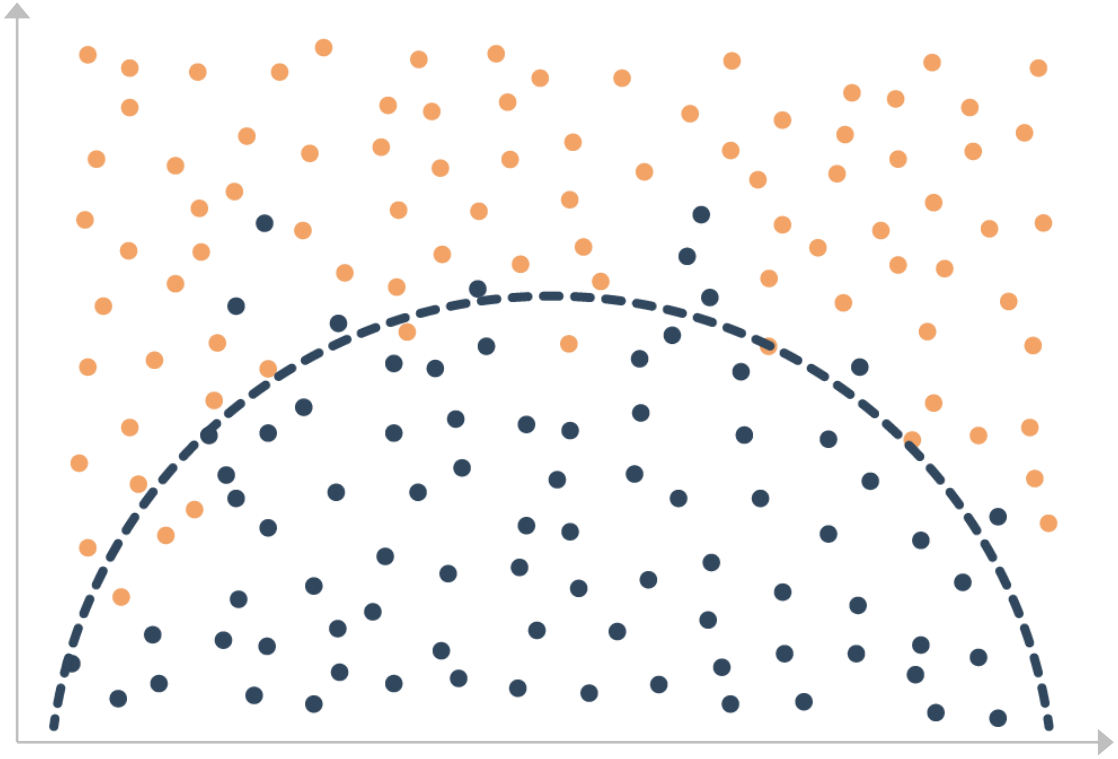


Kontrollált gépi tanulás

UNDERFITTING

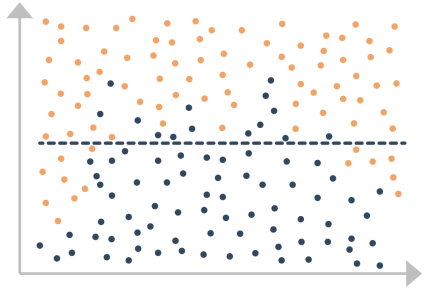


APPROPRIATE-FITTING

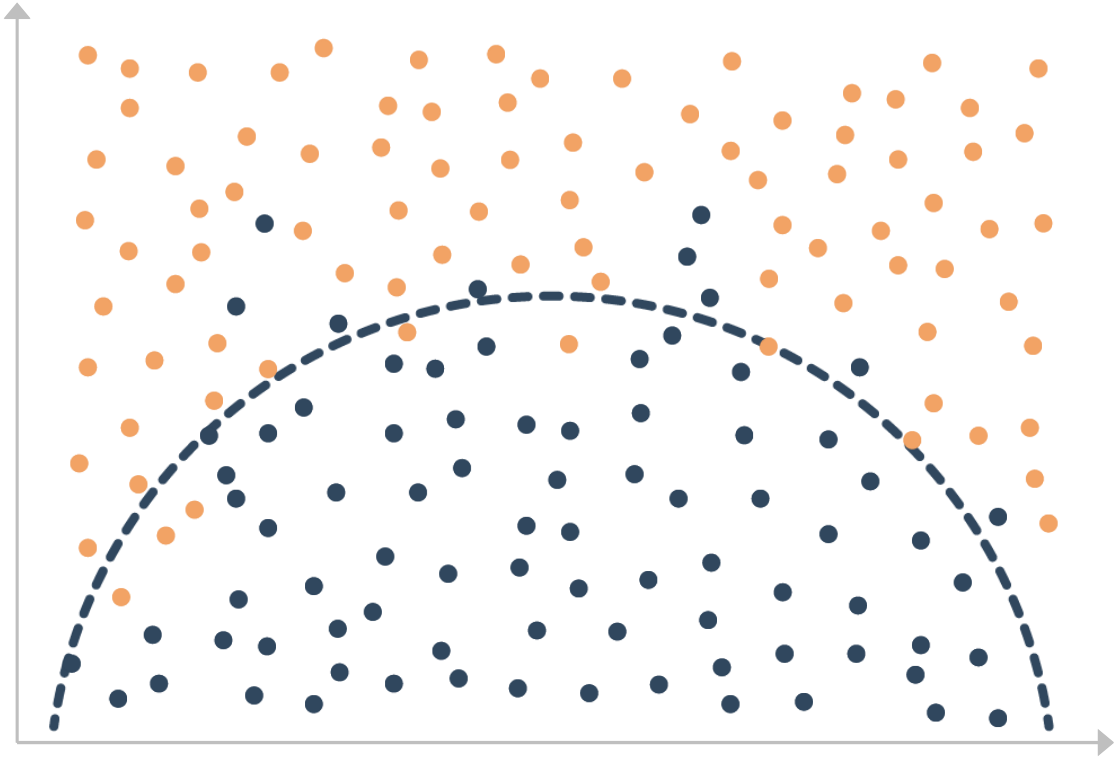


Kontrollált gépi tanulás

UNDERFITTING

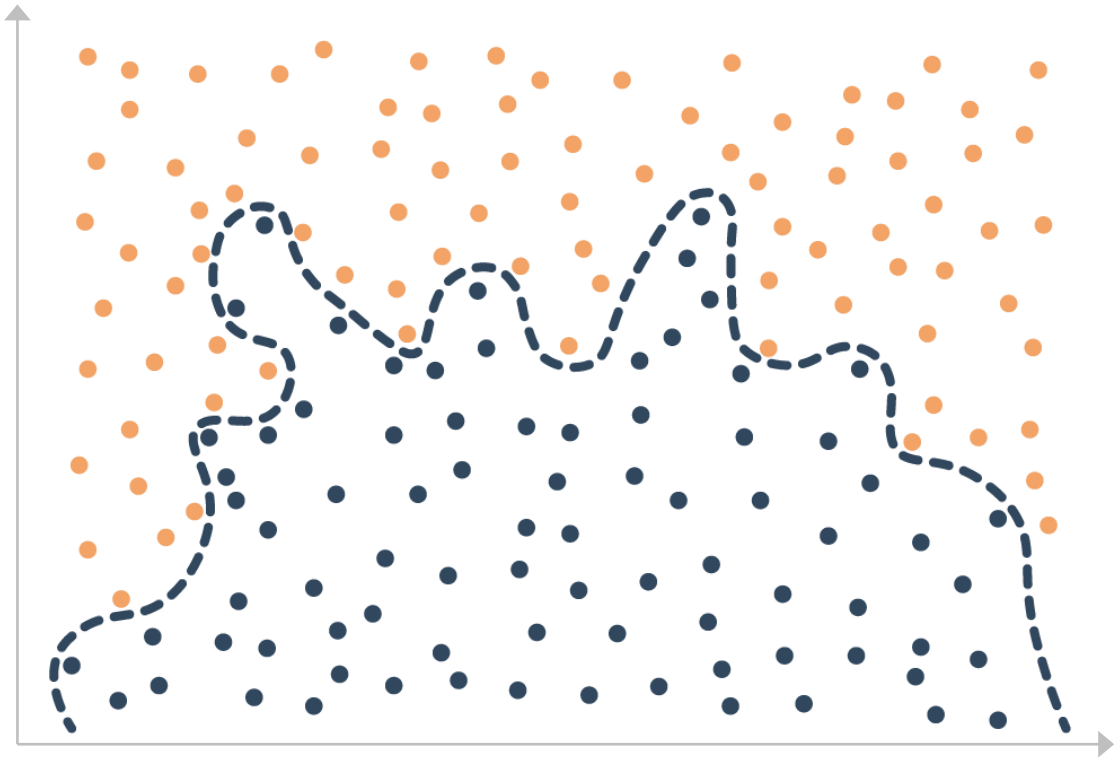


APPROPRIATE-FITTING

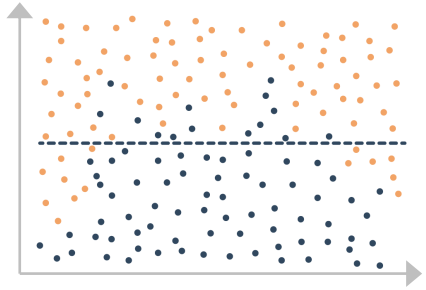


Kontrollált gépi tanulás

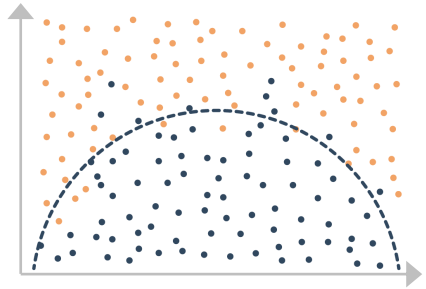
OVER-FITTING



UNDERFITTING

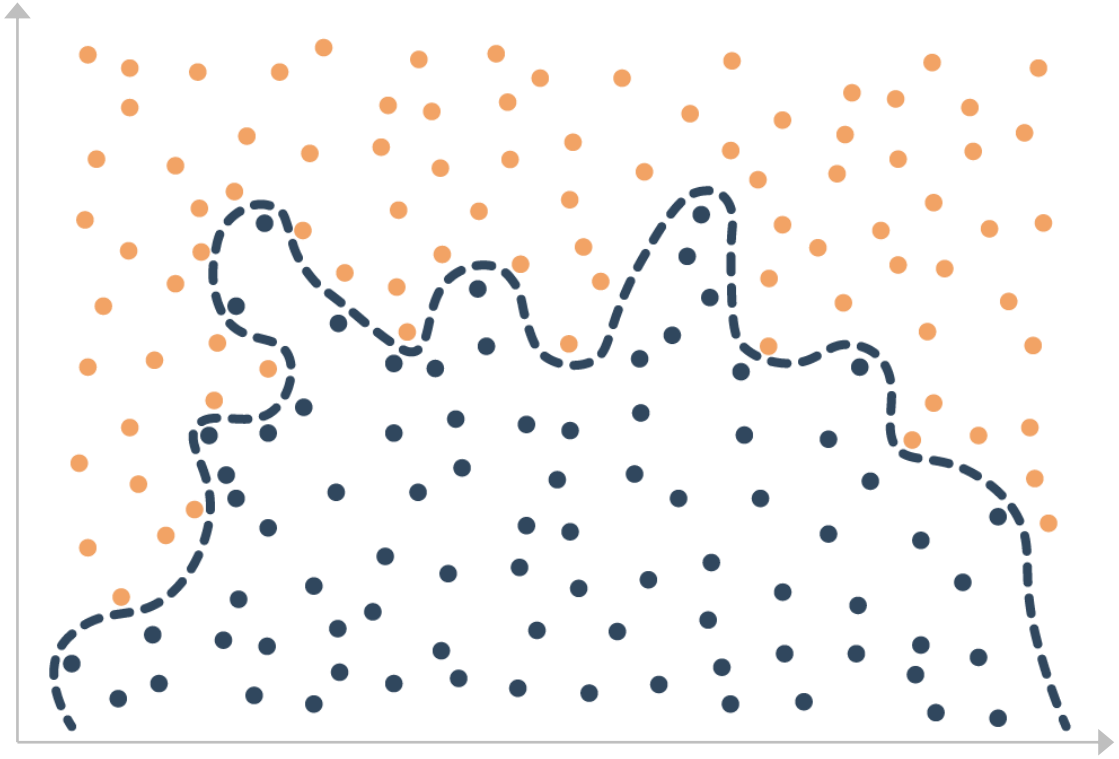


APPROPRIATE-FITTING

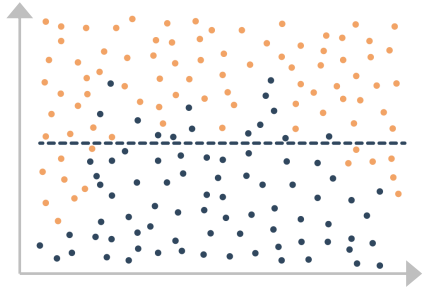


Kontrollált gépi tanulás

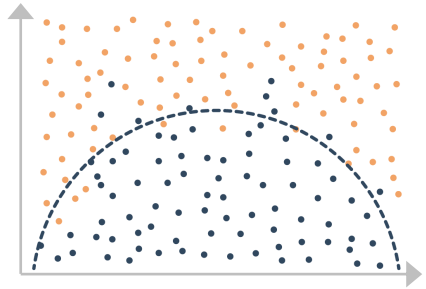
OVER-FITTING



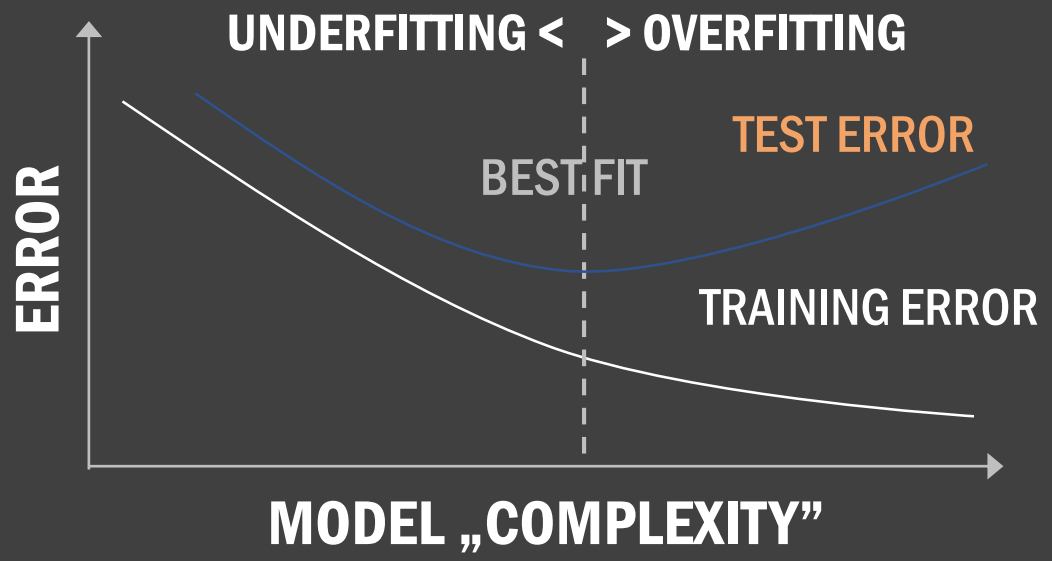
UNDERFITTING

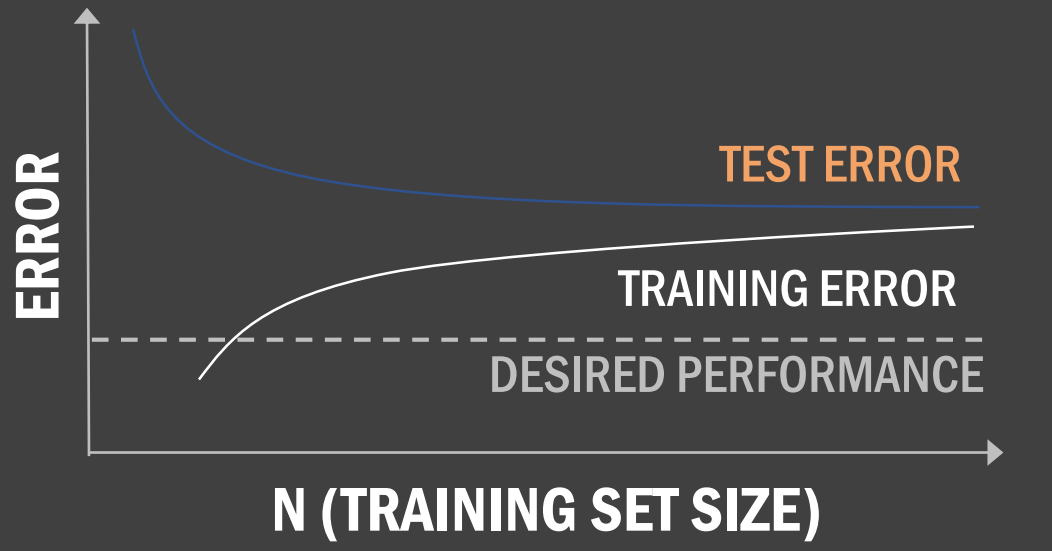


APPROPRIATE-FITTING

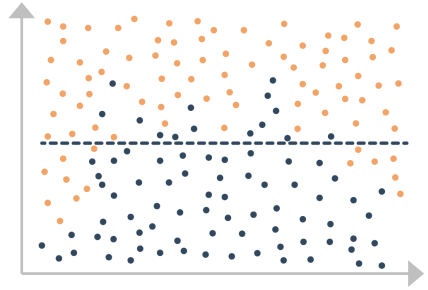


Kontrollált gépi tanulás

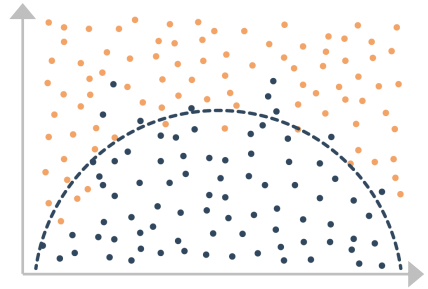




UNDERFITTING



APPROPRIATE-FITTING



OVER-FITTING

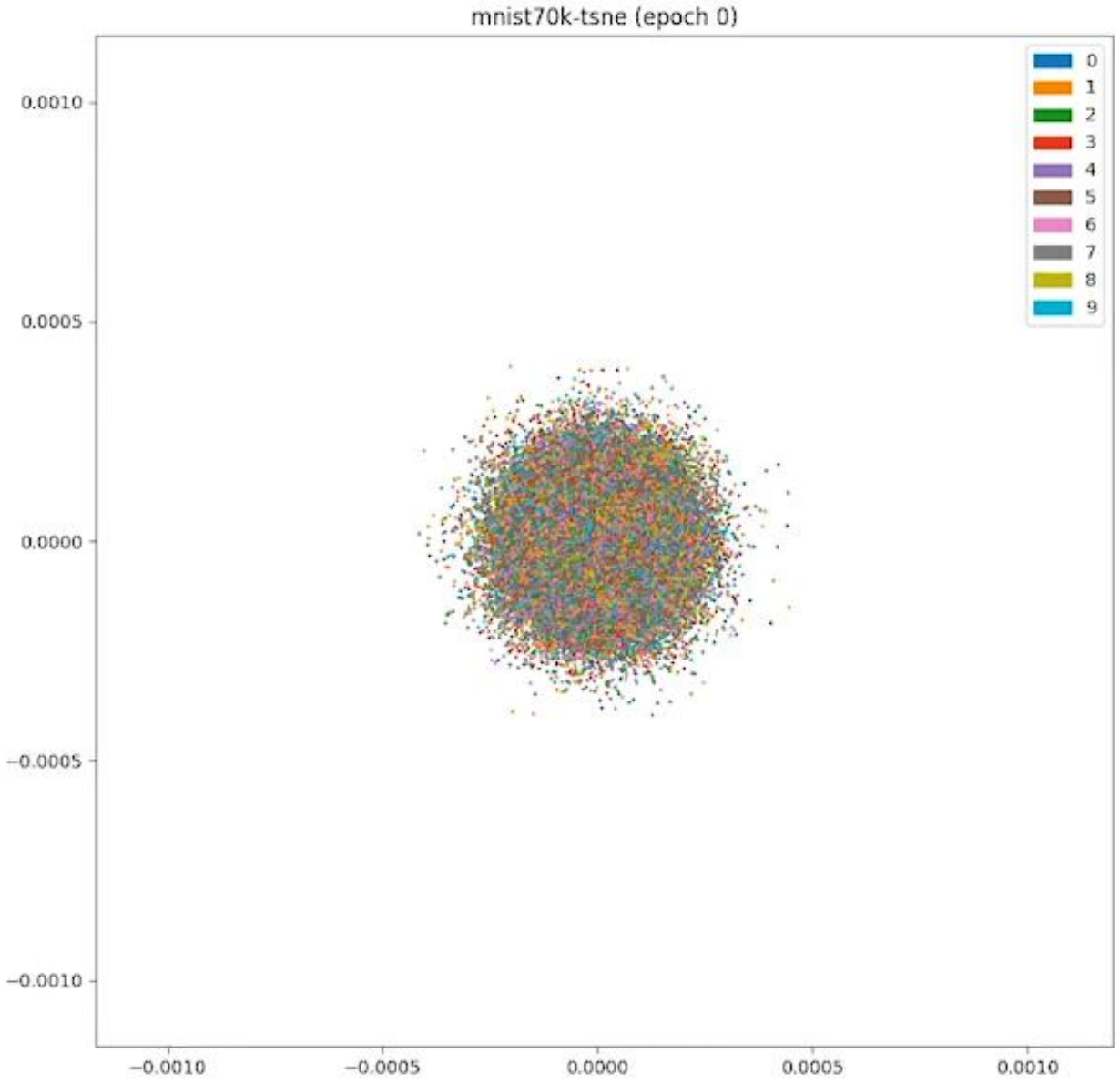
Tanító algoritmus

A bemeneti (tanító) adatok (*N* db) különböző tulajdonságát (mezőit) számokká alakítjuk, és elhelyezzük egy *N*-dimenziós térben. Így minden adatot egy pont reprezentál.

Kontrollált gépi tanulás (pl.): a pontok valamilyen kategóriajellemzővel bírnak (pl. csalárd / nem csalárd) és olyan (hiper) felületet tervezünk közéjük, amely az előzetes kategóriának megfelelően választja szét a pontokat.

Nem kontrollált gépi tanulás (pl.): a pontok az *N*-dimenziós térben definiált távolság (közelség) alapján többé-kevésbé elkülöníthető alakzatokat alkotnak. Ha ezekhez az alakzatokhoz tulajdonságot tudunk kötni, akkor kategóriákba soroltuk az adatok által leírt entitásokat.

A fenti modellek alapján a kategóriajellemzővel még nem rendelkező entitásokhoz kategóriát tudunk rendelni.
Magyarul: előítéletet tanítunk a géppel!



Tanító- és tesztadatok (+ kiértékelés)

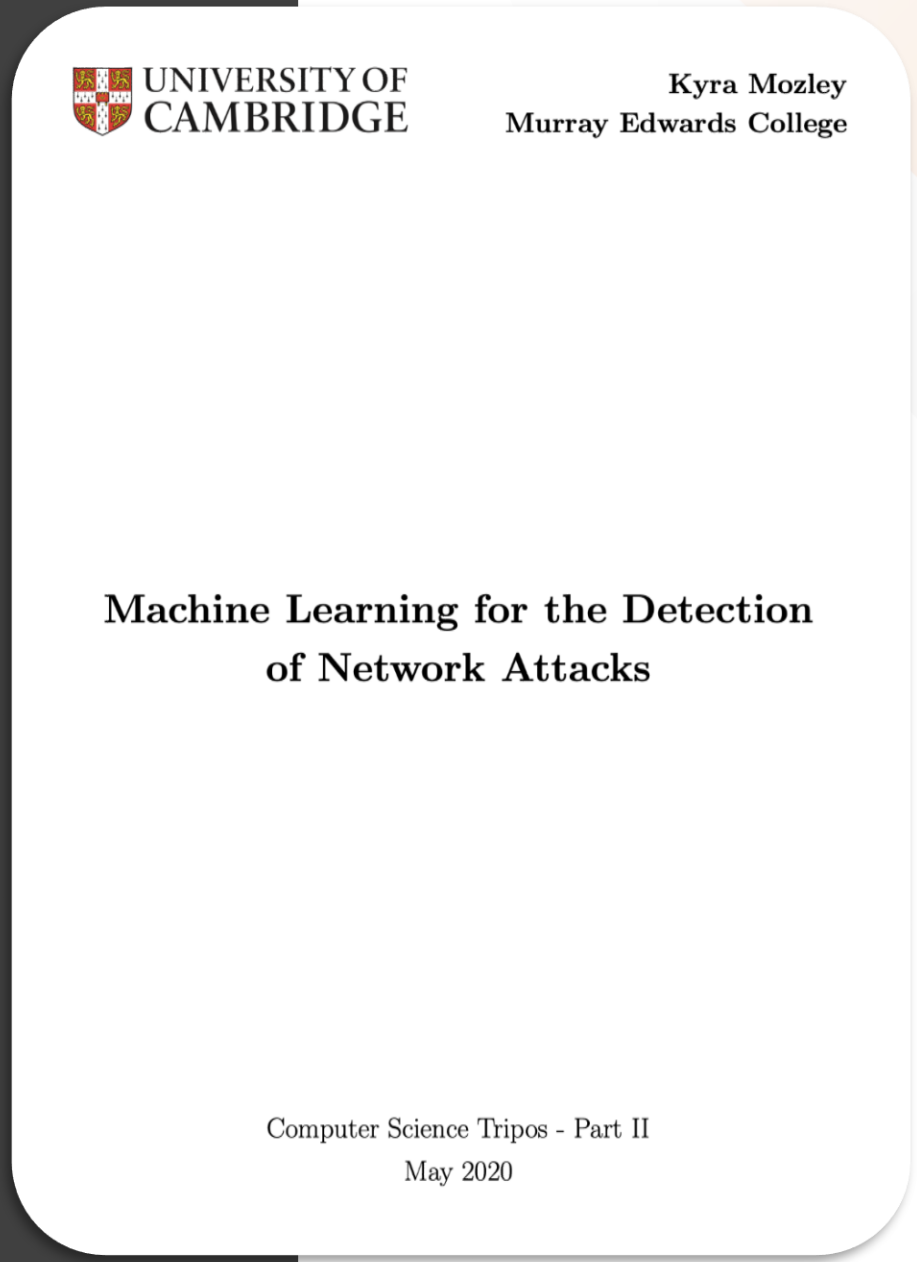
1. Intrusion detection evaluation dataset (CIC-IDS2017, Can. Inst. for Cybersec)
   - ”Generating realistic background traffic was our top priority in building this dataset.”

     „The data capturing period started at 9 a.m., Monday, July 3, 2017 and ended
   - at 5 p.m. on Friday July 7, 2017, for a total of 5 days. Monday is the normal day and only includes the benign traffic. The implemented attacks include Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS. They have been executed both morning and afternoon on Tuesday, Wednesday, Thursday and Friday.”

   University of New Brunswick Intrusion detection evaluation dataset

2. Saját adatgyűjtés szimulált támadásokkal vagy dobozos IDS-szoftver általi címkézéssel

Tisztán adatalapú megközelítés



UNIVERSITY OF CAMBRIDGE

Kyra Mozley
Murray Edwards College

Machine Learning for the Detection of Network Attacks

Computer Science Tripos - Part II
May 2020

Research program

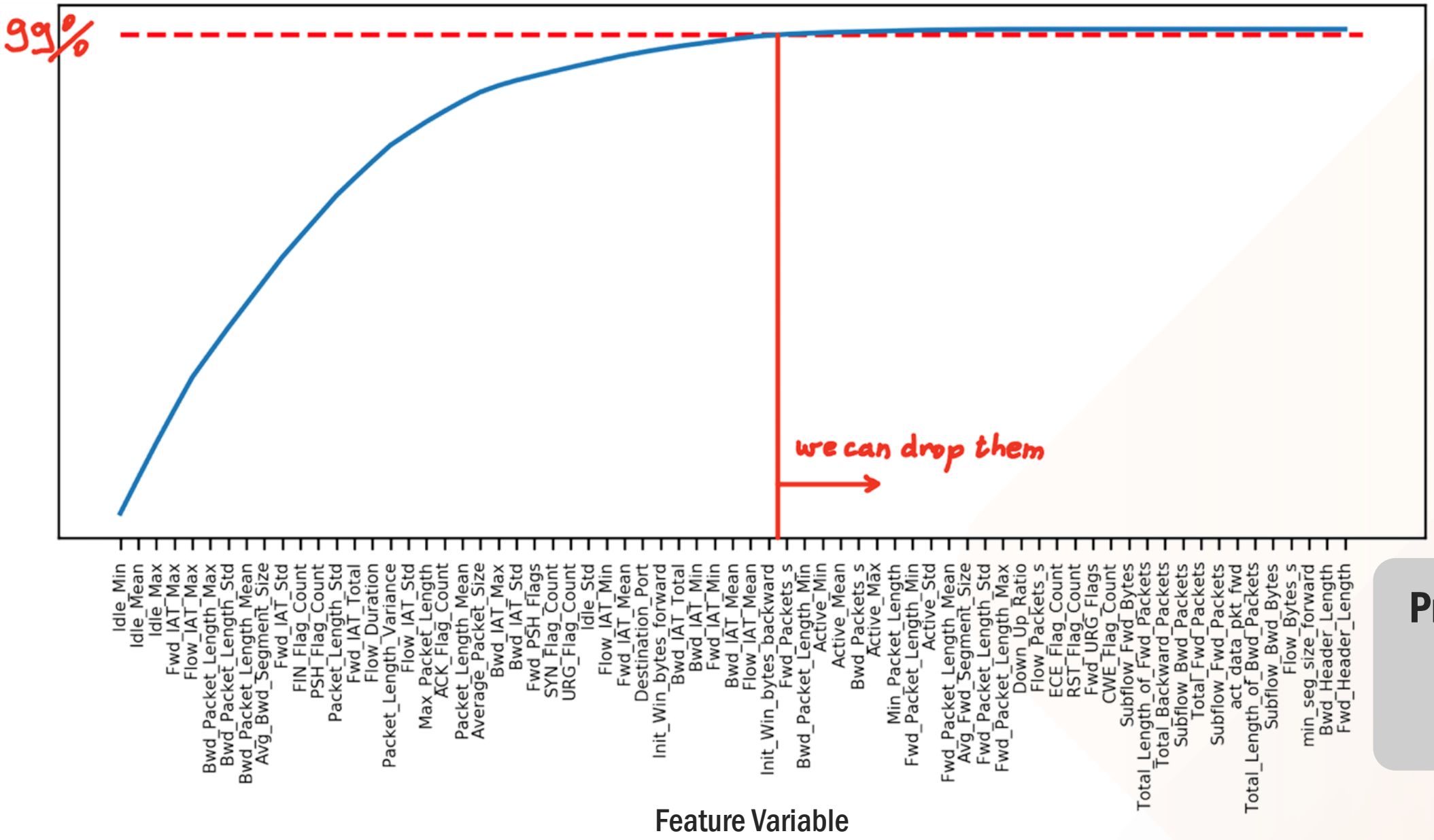
No.	Task	Priority	Difficulty	Risk
1	Find a suitable dataset	High	Medium	High
2	Preprocess the dataset	Medium	Low	Low
3	Perform feature selection on the dataset	Medium	Medium	Low
4	Perform the chosen five machine learning algorithms on the data, and evaluate using the validation set	High	High	Medium
5	Compare the different models to create the final model	Medium	Low	Low
6	Evaluate the final performance using the test set	Medium	Low	High
7	Build a system that can extract the chosen features from real-time traffic	High	High	High
8	Combine the machine learning model from task 5 and system from task 7 to create the intrusion detection system	Medium	High	Medium
9	Evaluate the performance of the IDS created in task 8 by simulating a variety of network attacks	Medium	High	High

Preprocessing – észszerű osztályozás

Label	Entries
Benign	2,273,097
DoS Hulk	231,073
Port Scan	158,930
DDoS	128,027
DoS GoldenEye	10,293
FTP-Patator	7,938
SSH-Patator	5,897
DoS Slowloris	5,796
DoS Slowhttptest	5,499
Bot	1,966
Web Attack: Brute Force	1,507
Web Attack: XSS	652
Infiltration	36
Web Attack: SQL Injection	21
Heartbleed	11

More general categories	Original categories
Botnet	Bot
Brute Force	FTP-Patator, SSH-Patator
DDoS	DDoS
DoS	DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS Slowloris
Probe	Port Scan
Web Attack	Web Attack: Brute Force, Web Attack: XSS

A Chart to Show Cumulative Feature Scores



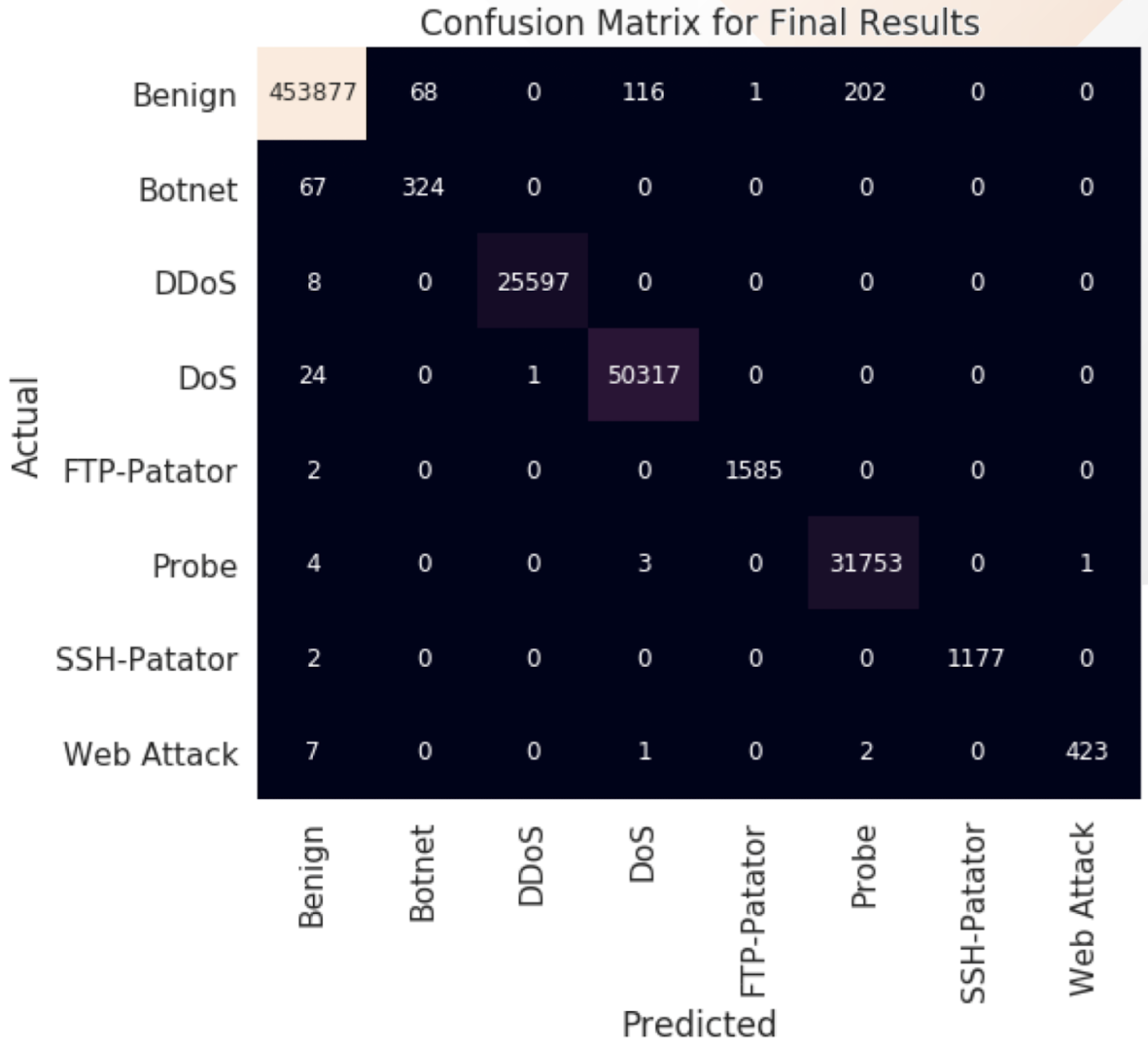
Preprocessing – megmaradó feature-ök

Preprocessing – megmaradó feature-ök

Feature	Description	Feature	Description
dest_port	Destination port number	fwd_PHS_flags	PSH (push) flag count (forward direction)
flow_duration	Duration of flow in microseconds	fwd_packets_s	Number of forward packets per second
bwd_packet_len_max	Maximum packet length (backward direction)	max_packet_len	Maximum packet length
bwd_packet_len_min	Minimum packet length (backward direction)	packet_len_mean	Mean packet length
bwd_packet_len_mean	Mean packet length (backward direction)	packet_len_std	Standard deviation of packet length
bwd_packet_len_std	Packet length standard deviation (backward direction)	packet_len_var	Packet length variance
flow_IAT_mean	Mean packet inter-arrival time	FIN_flag_count	FIN (finished) flag count
flow_IAT_std	Standard deviation of packet inter-arrival time	SYN_flag_count	SYN (synchronisation) flag count
flow_IAT_max	Maximum packet inter-arrival time	PSH_flag_count	PSH (push) flag count
flow_IAT_min	Minimum packet inter-arrival time	ACK_flag_count	ACK (acknowledgement) flag count
fwd_IAT_total	Total packet inter-arrival time (forward direction)	URG_flag_count	URG (urgent) flag count
fwd_IAT_mean	Mean packet inter-arrival time (forward direction)	avg_packet_size	Average size of a packet
fwd_IAT_std	Standard deviation of packet inter-arrival time (forward direction)	avg_bwd_segment_size	Average size (backward direction)
fwd_IAT_max	Maximum packet inter-arrival time (forward direction)	init_win_bytes_forward	Number of bytes sent in the initial window (forward direction)
fwd_IAT_min	Minimum packet inter-arrival time (forward direction)	init_win_bytes_backward	Number of bytes sent in the initial window (backward direction)
bwd_IAT_total	Total packet inter-arrival time (backward direction)	active_min	Minimum time a flow was active before becoming idle
bwd_IAT_mean	Mean packet inter-arrival time (backward direction)	idle_mean	Mean time a flow was idle before becoming active
bwd_IAT_std	Standard deviation of packet inter-arrival time (backward direction)	idle_std	Standard deviation of time a flow was idle before becoming active
bwd_IAT_max	Maximum packet inter-arrival time (backward direction)	idle_max	Maximum time flow idle before becoming active
bwd_IAT_min	Minimum packet inter-arrival time (backward direction)	idle_min	Minimum time flow idle before becoming active

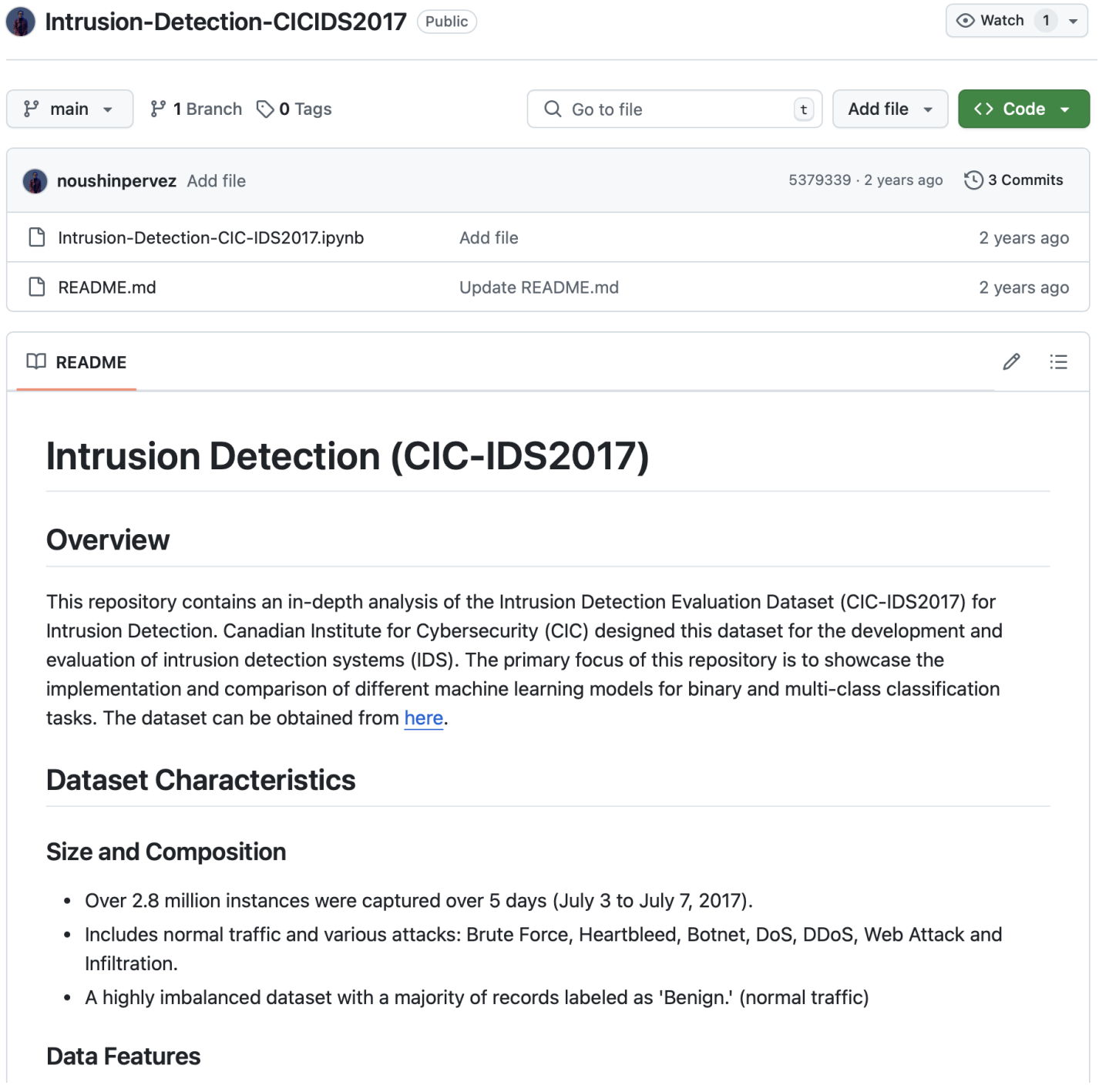
Eredmény scikit-learn használatával

Label	Precision (%)	Recall (%)	F1 Score (%)	% of Training Data
Benign	99.97	99.91	99.94	80.32
Botnet	82.65	82.86	82.76	0.07
DDos	100.00	99.97	99.98	4.53
DoS	99.76	99.95	99.86	8.90
FTP-Patator	99.94	99.87	99.90	0.28
Probe	99.36	99.97	99.67	5.62
SSH-Patator	100.00	99.83	99.91	0.21
Web Attack	99.76	97.69	98.71	0.08
Average	97.68	97.51	97.59	12.50



Adatalapú + kibervédelmi megközelítés

This repository contains an in-depth analysis of the Intrusion Detection Evaluation Dataset (CIC-IDS2017) for Intrusion Detection, showcasing the implementation and comparison of different machine learning models for binary and multi-class classification tasks.



Intrusion-Detection-CICIDS2017 Public

Watch 1

main 1 Branch 0 Tags Go to file t Add file Code

noushinpervez Add file 5379339 · 2 years ago 3 Commits

Intrusion-Detection-CIC-IDS2017.ipynb	Add file	2 years ago
README.md	Update README.md	2 years ago

README

Intrusion Detection (CIC-IDS2017)

Overview

This repository contains an in-depth analysis of the Intrusion Detection Evaluation Dataset (CIC-IDS2017) for Intrusion Detection. Canadian Institute for Cybersecurity (CIC) designed this dataset for the development and evaluation of intrusion detection systems (IDS). The primary focus of this repository is to showcase the implementation and comparison of different machine learning models for binary and multi-class classification tasks. The dataset can be obtained from here.

Dataset Characteristics

Size and Composition

- Over 2.8 million instances were captured over 5 days (July 3 to July 7, 2017).
- Includes normal traffic and various attacks: Brute Force, Heartbleed, Botnet, DoS, DDoS, Web Attack and Infiltration.
- A highly imbalanced dataset with a majority of records labeled as 'Benign.' (normal traffic)

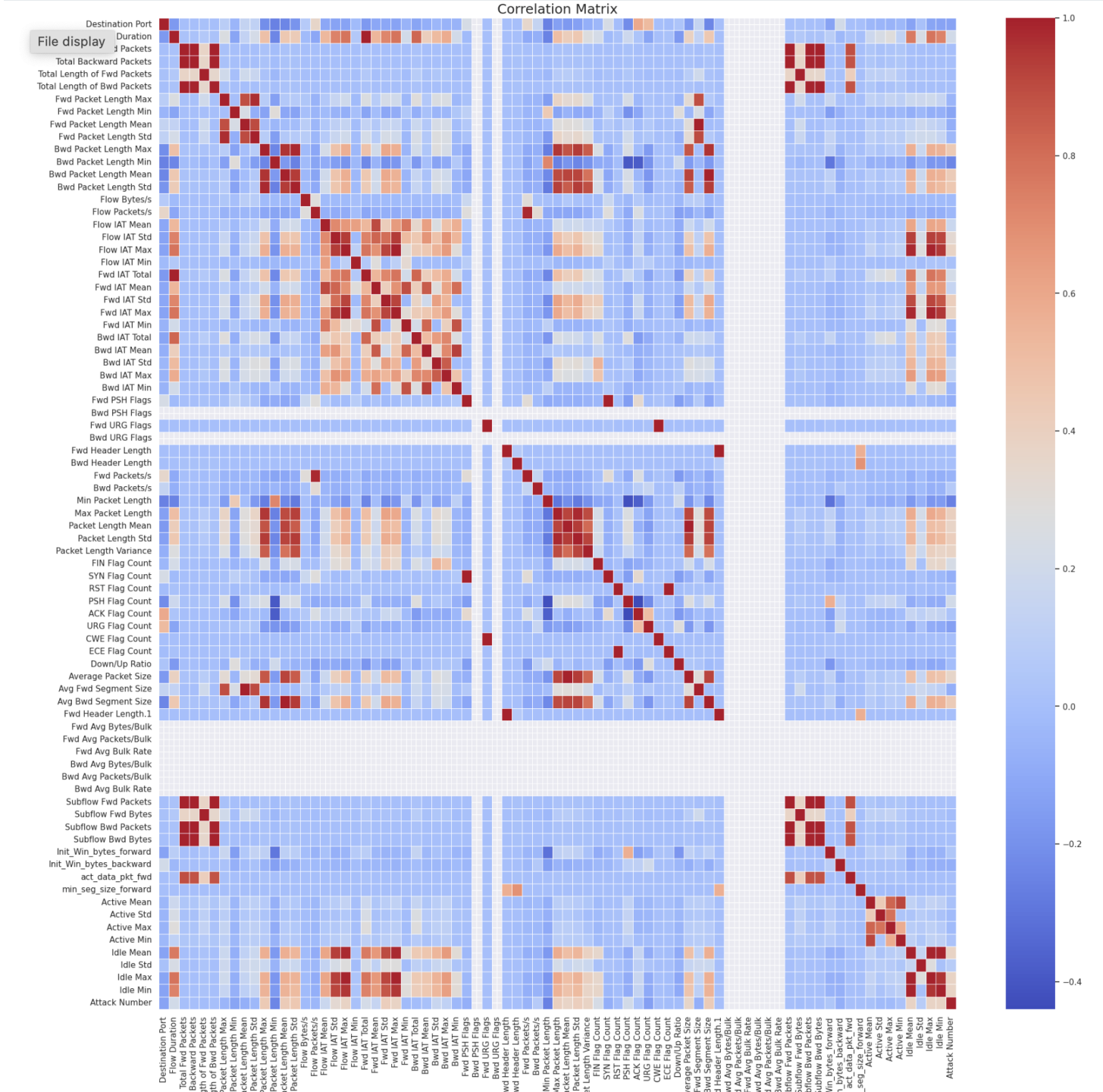
Data Features

Feature korreláció

Intrusion Detection
(CIC-IDS2017) GitHub repository

A „Jelen van?” mellett
a „Mi történik?”-re is választ keres a
szemantikus adatelemek egymás közti és
a címkékkel való összefüggések
elemzésével

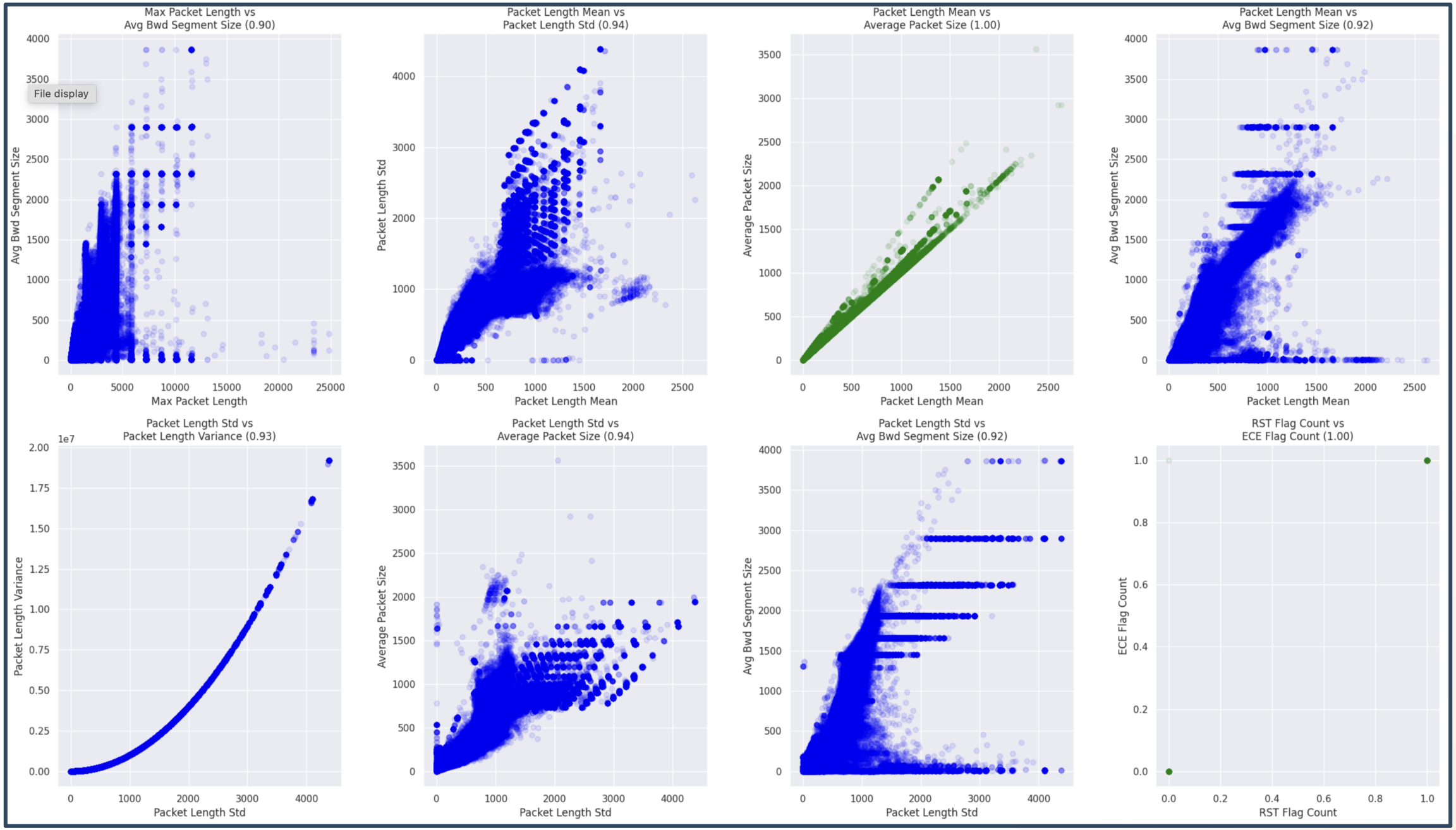
GitHub repository:
Intrusion-Detection-CICIDS2017



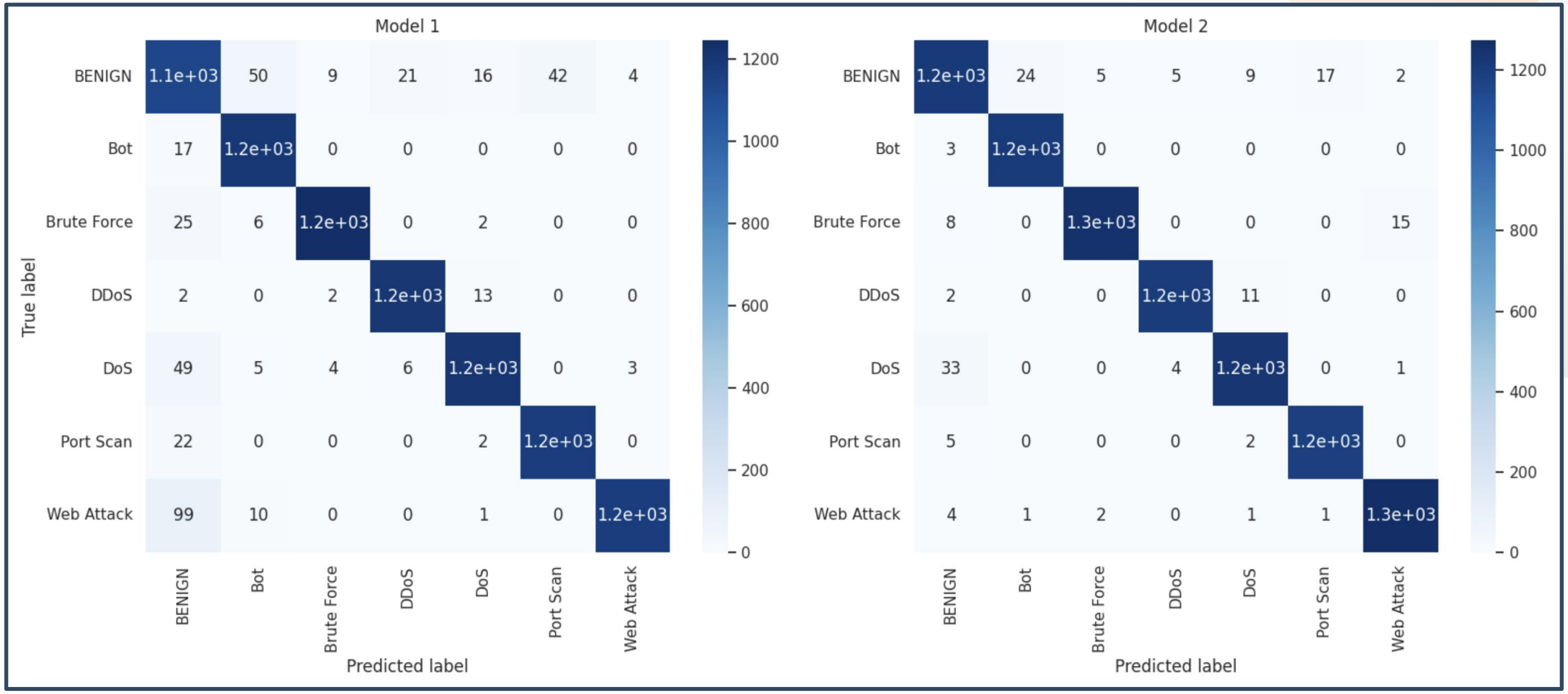
Korreláció címkével

```
# Positive correlation features for
'Attack Number'
pos_corr_features = corr['Attack
Number'][(corr['Attack Number'] > 0) &
 (corr['Attack Number'] < 1)].index.tolist()

print("Features with positive correlation with
'Attack Number':\n")

for i, feature in enumerate(pos_corr_features,
start = 1):
 corr_value = corr.loc[feature, 'Attack Number']
 print('{:<3} {:<24} :{}'.format(f'{i}.',
 feature, corr_value))
```

```
Features with positive correlation with 'Attack Number':
1.  Flow Duration            :0.21
2.  Bwd Packet Length Max    :0.44
3.  Bwd Packet Length Mean   :0.43
4.  Bwd Packet Length Std    :0.45
5.  Flow IAT Mean            :0.17
6.  Flow IAT Std             :0.33
7.  Flow IAT Max             :0.38
8.  Flow IAT Min             :0.01
9.  Fwd IAT Total            :0.22
10. Fwd IAT Mean             :0.15
11. Fwd IAT Std              :0.41
12. Fwd IAT Max              :0.38
13. Bwd IAT Mean             :0.01
14. Bwd IAT Std              :0.16
15. Bwd IAT Max              :0.12
16. Bwd Packets/s            :0.07
17. Max Packet Length        :0.4
18. Packet Length Mean       :0.37
19. Packet Length Std        :0.41
20. Packet Length Variance   :0.38
21. FIN Flag Count           :0.23
22. PSH Flag Count           :0.21
23. ACK Flag Count           :0.03
24. Average Packet Size      :0.36
25. Avg Bwd Segment Size     :0.43
26. Init_Win_bytes_forward   :0.04
27. Active Mean              :0.01
28. Active Min               :0.02
29. Idle Mean                :0.38
30. Idle Std                 :0.08
31. Idle Max                 :0.38
32. Idle Min                 :0.38
```


Max Packet Length vs
Avg Bwd Segment Size (0.90)
File display
Avg Bwd Segment Size
Max Packet Length
Packet Length Mean vs
Packet Length Std (0.94)
Packet Length Std
Packet Length Mean
Packet Length Mean vs
Average Packet Size (1.00)
Average Packet Size
Packet Length Mean
Packet Length Mean vs
Avg Bwd Segment Size (0.92)
Avg Bwd Segment Size
Packet Length Mean
Packet Length Std vs
Packet Length Variance (0.93)
1e7
Packet Length Variance
Packet Length Std
Packet Length Std vs
Average Packet Size (0.94)
Average Packet Size
Packet Length Std
Packet Length Std vs
Avg Bwd Segment Size (0.92)
Avg Bwd Segment Size
Packet Length Std
RST Flag Count vs
ECE Flag Count (1.00)
ECE Flag Count
RST Flag Count

Az eredmény


Model 1
BENIGN 1.1e+03 50 9 21 16 42 4
Bot 17 1.2e+03 0 0 0 0 0
Brute Force 25 6 1.2e+03 0 2 0 0
DDoS 2 0 2 1.2e+03 13 0 0
DoS 49 5 4 6 1.2e+03 0 3
Port Scan 22 0 0 0 2 1.2e+03 0
Web Attack 99 10 0 0 1 0 1.2e+03
True label
BENIGN Bot Brute Force DDoS DoS Port Scan Web Attack
Predicted label
1200 1000 800 600 400 200 0
Model 2
BENIGN 1.2e+03 24 5 5 9 17 2
Bot 3 1.2e+03 0 0 0 0 0
Brute Force 8 0 1.3e+03 0 0 0 15
DDoS 2 0 0 1.2e+03 11 0 0
DoS 33 0 0 4 1.2e+03 0 1
Port Scan 5 0 0 0 2 1.2e+03 0
Web Attack 4 1 2 0 1 1 1.3e+03
BENIGN Bot Brute Force DDoS DoS Port Scan Web Attack
Predicted label
1200 1000 800 600 400 200 0

HÍRKÖZLÉSI ÉS
INFORMATIKAI
TUDOMÁNYOS
EGYESÜLET

ALAPÍTVA 1949