Machine learning techniques for a flow-based intrusion detection system

Auteur: Axel Faes
Begeleider: ir. Bram Bonne
Begeleider: Pieter Robyns
Begeleider: Robin Marx
Promotor: Prof. Dr. Peter Quax
Co-promotor: Prof. Dr. Wim Lamotte

Traffic can contain attacks or anomalies:
- Malware
  Botnets, viruses, worms, etc.
- External malicious behaviour
  SSH scans, DDOS, etc.


[2]

- **Network** traffic


[1]

- **Aggregate** network packets into *IP Flows*

IP Flows:
- Source IP
- Destination IP
- IP Protocol
- Source port
- Destination port
- Starting time
- Duration
- Amount packets
- Amount bytes

Can detect:
- DDOS
- Scans
- Worms
- Botnets

Pre-processing:
Break IP Flows into a list of features

Feature = individual property

3 different sets were used:
- Standard feature set
  Uses continuous values from IP Flows
- TCP feature set
  Also uses TCP Flags
- Country feature set
  Uses country-of-origin instead of IP

- **Send** processed *IP Flows* to ML algorithms

Research questions:
- Can machine learning be used for intrusion detection?
- How can IP Flows be used with this approach?
- Can an intrusion detection system work out-of-the-box?
- Which anomalies can be detected?
- Are these techniques applicable in real-life scenarios?

[11]

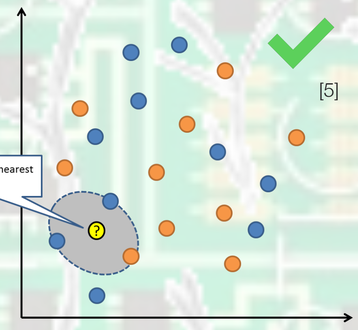
scikits learn machine learning in Python [3]

Naive Bayes Classifier

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

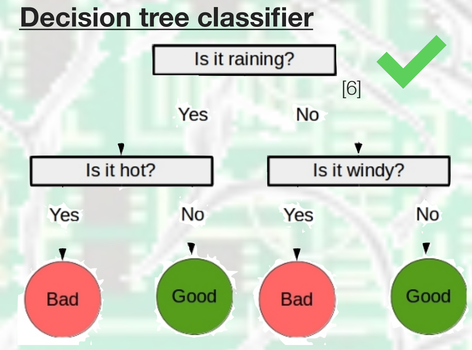
[4]

K-Nearest Neighbours (KNN)



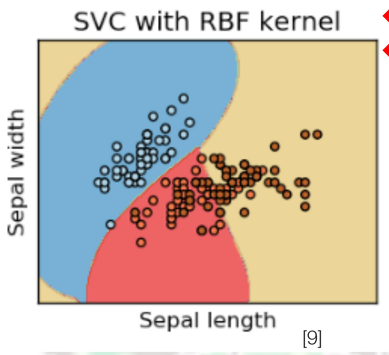
[5]

Decision tree classifier



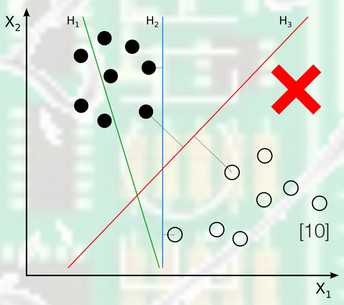
[6]

Non-Linear Support Vector machine



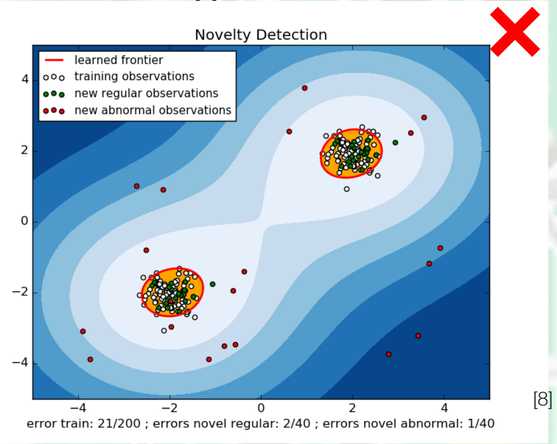
[9]

Linear Support Vector machine



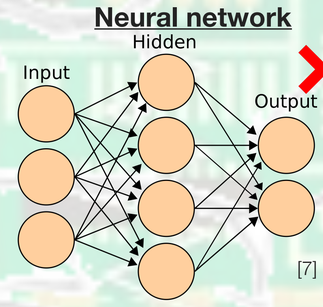
[10]

One-class Support Vector machine

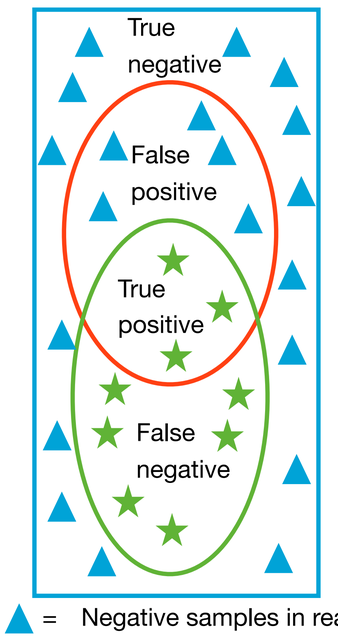


[8]

Neural network



[7]



▲ = Negative samples in reality

★ = Positive samples in reality

○ = Positive samples by system

F-score = mean recall/precision

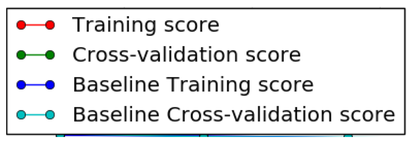
Recall = ability to detect anomalies

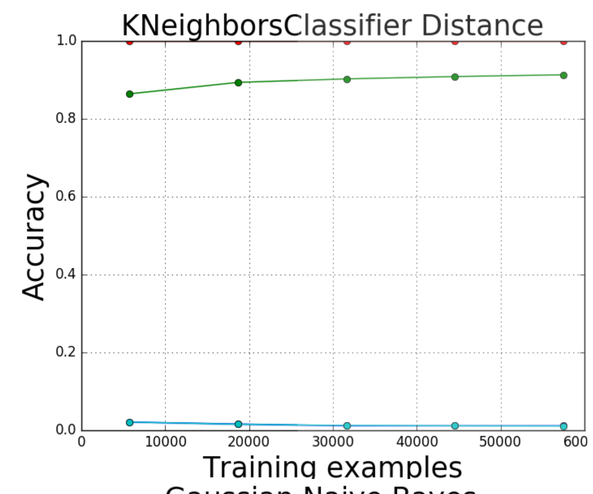
Precision = ability to only detect anomalies

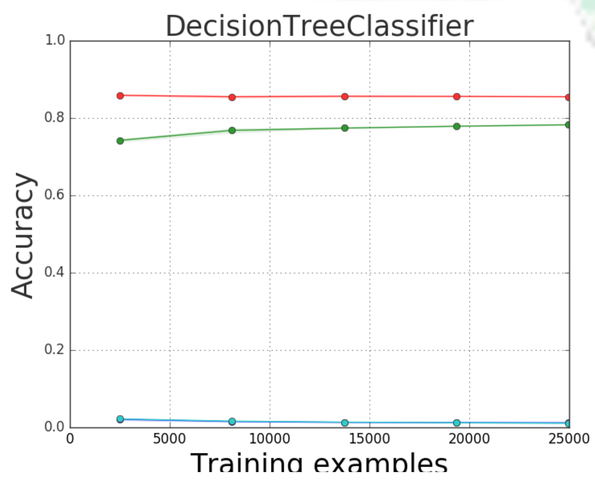
Evaluation:
4 steps:
- Learning curves
- Validation within same dataset
- Validation across datasets
- Validation with real-world data from EDM and Cegeka

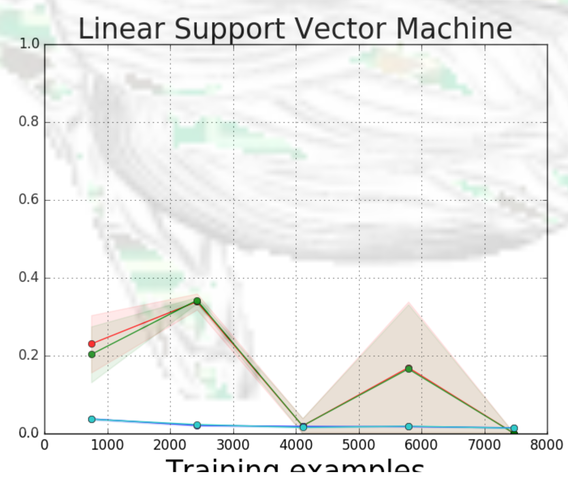
Step 1: Learning curves:
- Can show learning problems
- Shows effect of using more/less training samples

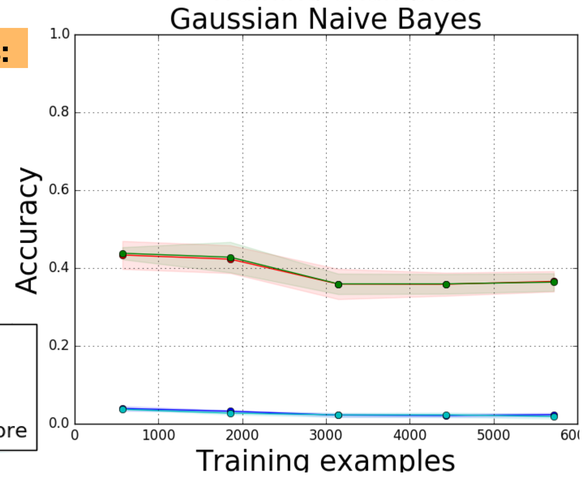
Legend

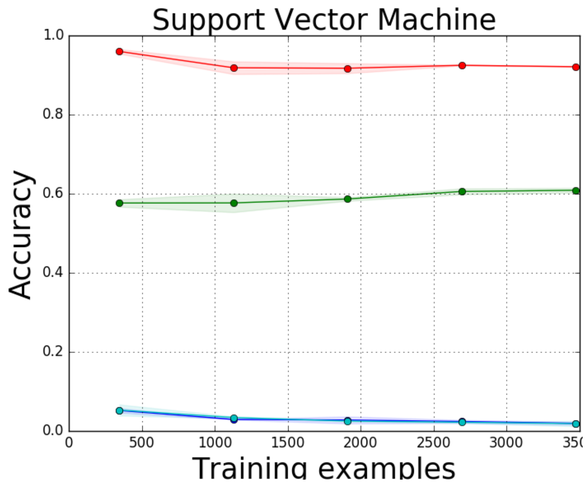


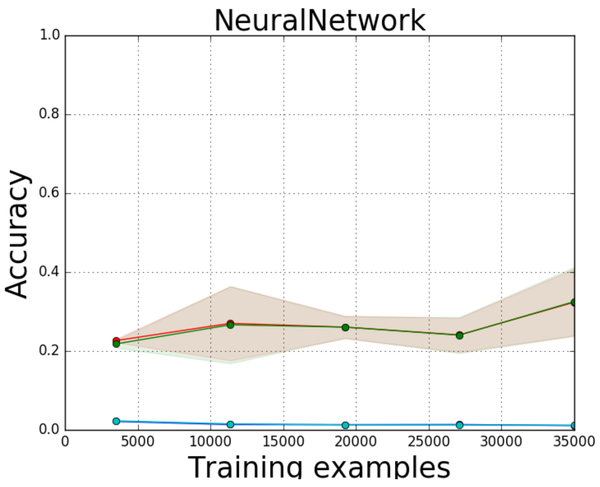


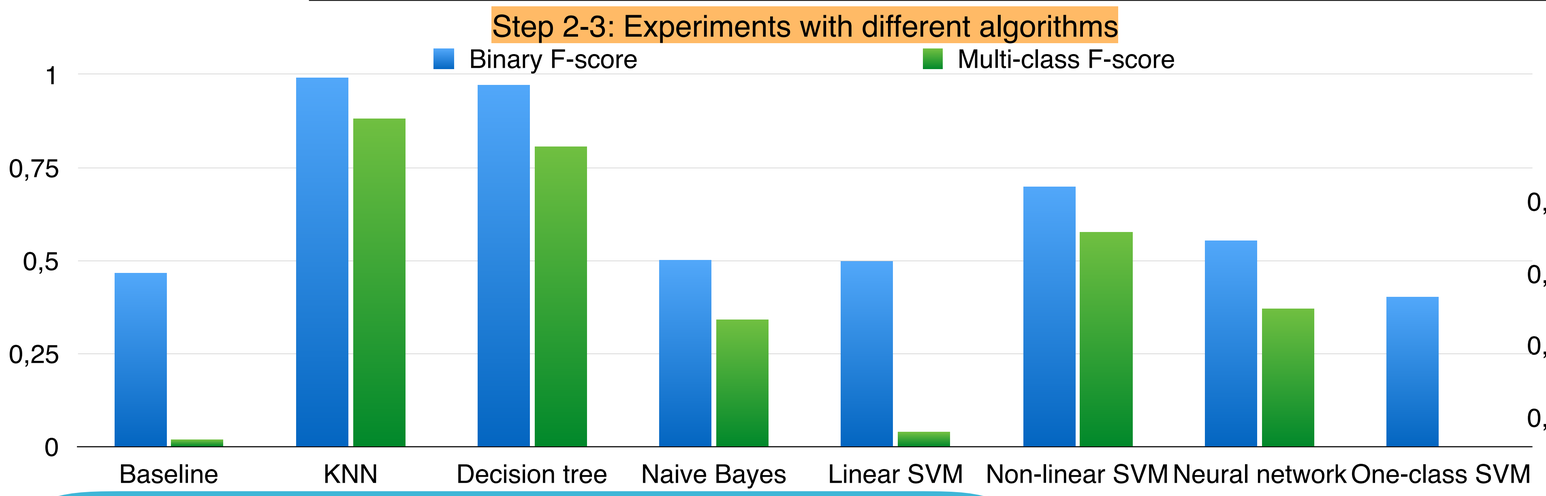


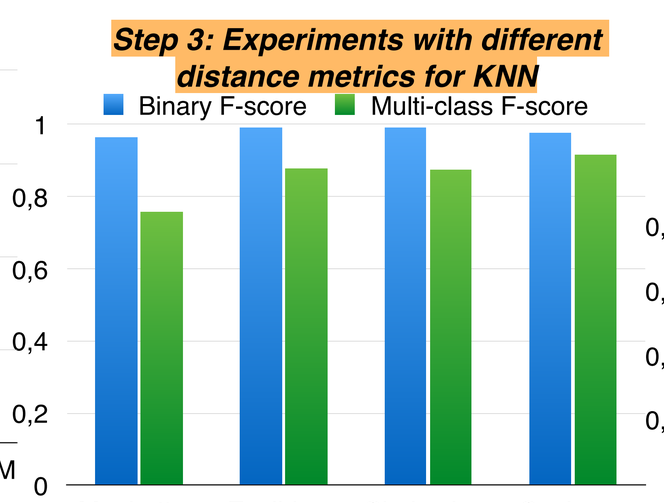


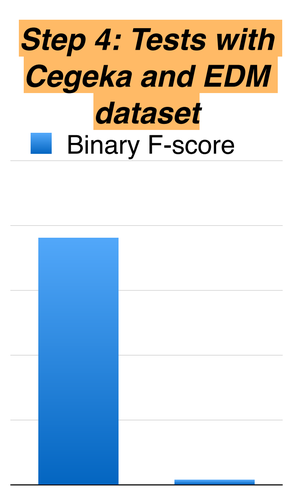












Conclusion:
- Machine learning can be effectively used for intrusion detection
- Can be applied in real-world
- More information (ie. TCP flags) is better

Cegeka/EDM experiments

	KNN	Decision tree
Binary F-score	0,7633	0,0155
Total samples	11072646	11072646
False negative	8905	14280
False positive	4164	3245349
True negative	11038482	7787297
True positive	21095	25720

Open-source
Source code from framework has been open-sourced on Github: (also contains image sources)
https://goo.gl/eayimk

Future work
- Intrusion prevention
- Combining algorithms
- Using packet data
- Binary classification

EDM universiteit hasselt
cegeka
KNOWLEDGE IN ACTION