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

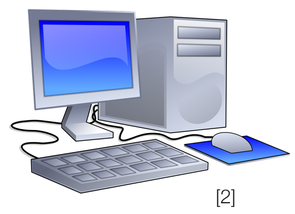
Research questions:

- 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?

Traffic can contain attacks or anomalies:

- Malware
  - Botnets, viruses, worms, etc.
- External malicious behaviour
  - SSH scans, DDOS, etc.

Network traffic



[2]

[1]

- Aggregate network packets into IP flows
- Send IP Flows to the Intrusion detection system

IP Flows:

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

1. Training phase

Train a machine learning algorithm

Uses a labeled dataset containing:

- Normal traffic
- Malware traffic
- External malicious behaviour

2. Prediction phase

Predict to which classification an IP Flow belongs

Afterwards checked whether the class is malicious or not

Which anomalies can be detected?

In normal conditions:

- DDOS
- Scans
- Worms
- Botnets

How to use IP Flows in algorithms?

Break IP Flows into a list of features

Feature = individual property
vb duration of the flow

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

How did the validation happen?

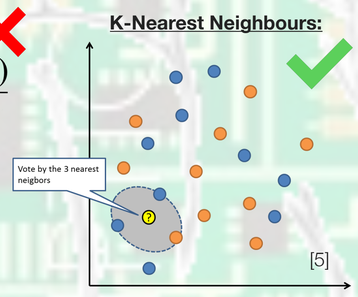
1. Labeled dataset
   Predictions can be checked with these labels
2. Cegeka dataset
   Firewall logs
3. EDM dataset
   Manual validation

Naive Bayes Classifier:

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

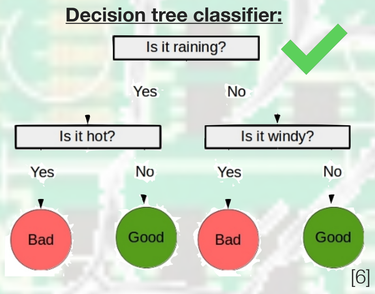
[4]

K-Nearest Neighbours:



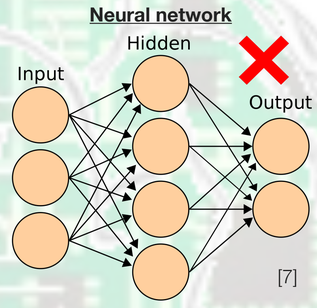
[5]

Decision tree classifier:



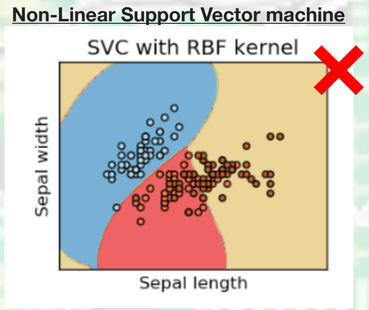
[6]

Neural network



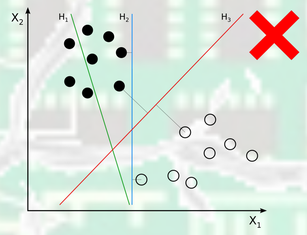
[7]

Non-Linear Support Vector machine



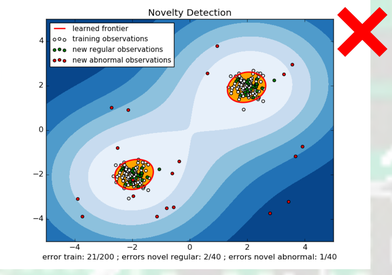
[9]

Linear Support Vector machine



[10]

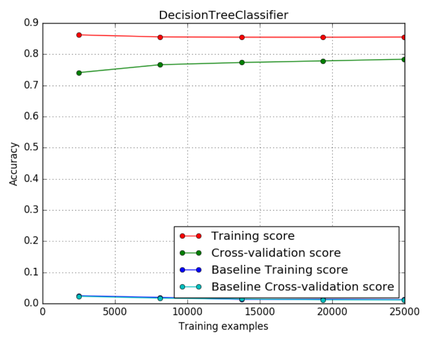
One-class Support Vector machine



[8]

How did the evaluation happen?

1. Baseline
   Random classifier
   All-non-malicious classifier
2. Learning curves

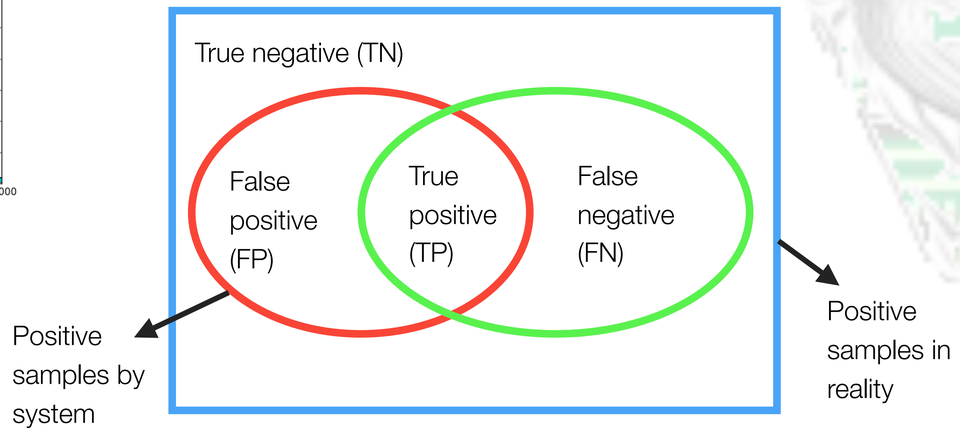


3. F-score

$$\text{Precision (P)} = \frac{truepositive}{truepositive + falsepositive}$$

$$\text{Recall (R)} = \frac{truepositive}{truepositive + falsenegative}$$

$$\text{F-score} = 2\frac{PR}{P+R}$$



What can happen next?

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

Source code from framework has been open-sourced on Github:
https://github.com/TheAxeC/Machine-learning-techniques-for-flow-based-network-intrusion-detection-systems

Image sources:

[1]: https://upload.wikimedia.org/wikipedia/commons/thumb/d/d7/Desktop_computer_clipart_-_Yellow_theme.svg/2000px-Desktop_computer_clipart_-_Yellow_theme.svg.png
[2]:https://pixabay.com/static/uploads/photo/2013/07/12/14/11/workstation-147953_960_720.png
[3]:https://kaggle2.blob.core.windows.net/competitions/kaggle/3428/media/scikit-learn-logo.png
[4]: http://media.charlesleifer.com/blog/photos/p1422977174.11.png
[5]: http://3.bp.blogspot.com/-ZslDMqm5M9o/T8ja_f_fALI/AAAAAAAAAt4/z7w55YAZXpw/s1600/p1.png
[6]: https://www.packtpub.com/sites/default/files/Article-Images/6053OS_03_05.png
[7]: https://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/Artificial_neural_network.svg/2000px-Artificial_neural_network.svg.png
[8]: http://scikit-learn.org/stable/modules/svm.html#svm-outlier-detection
[9]: http://scikit-learn.org/stable/modules/svm.html
[10]: https://upload.wikimedia.org/wikipedia/commons/thumb/b/b5/Svm_separating_hyperplanes_(SVG).svg/2000px-Svm_separating_hyperplanes_(SVG).svg.png
[11]: https://upload.wikimedia.org/wikipedia/commons/1/17/ArtificialFictionBrain.png

EDM
cegeka
universiteit hasselt
KNOWLEDGE IN ACTION