# Malicious traffic detection using Machine Learning

Nikolay Lyfenko [1]

[1] Positive Technologies

April 1, 2023

# Outline

1. Introduction

2. Proposed approach
    - Dataset collection and labelling
    - Online evaluation

3. Conclusions

# Introduction

# Task

**Task**

The task is to detect malware traffic

# What is malware?

## Avast definition

*Malware is an umbrella term for any type of "malicious software" that's designed to infiltrate your device without your knowledge, cause damage or disruption to your system, or steal data. Adware, spyware, viruses, botnets, trojans, worms, rootkits, and ransomware all fall under the definition of malware*[a]

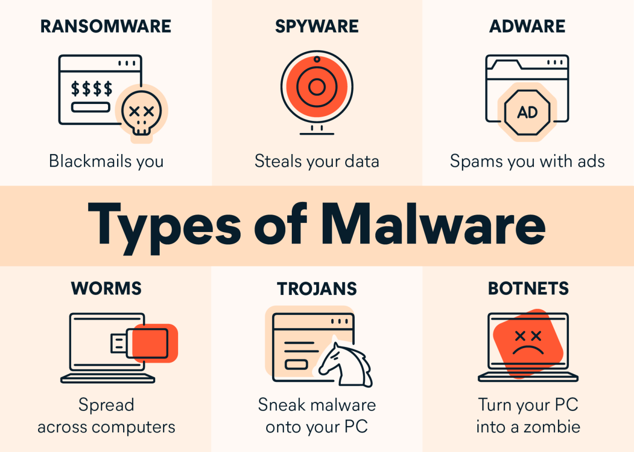
[a] https://www.avast.com/c-malware

## Kaspersky definition

*Malware is malicious software that is purposefully designed to cause harm to you or your device.*[a]

[a] https://www.kaspersky.com/resource-center/threats/malware-protection

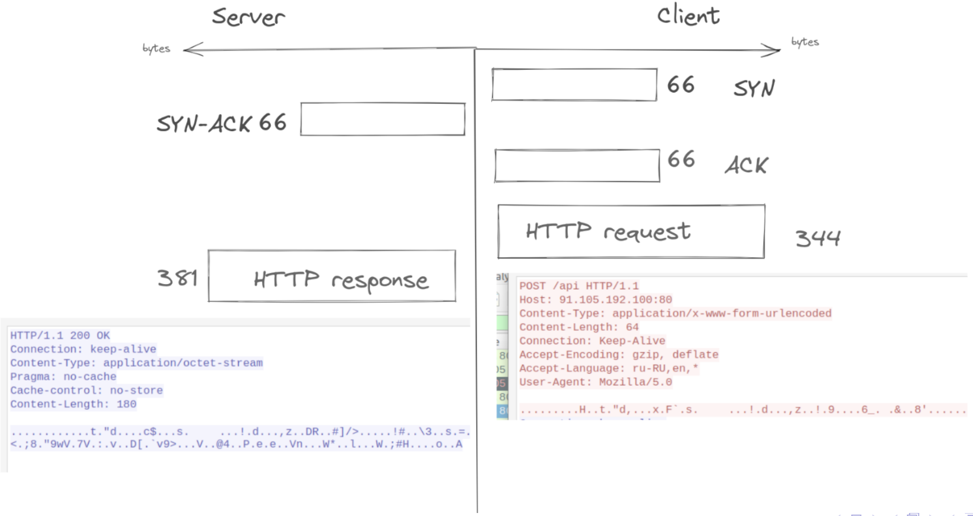
# Malware types



[1]

[1] https://www.avast.com/c-malware

# What is traffic?



```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/octet-stream
Pragma: no-cache
Cache-control: no-store
Content-Length: 180

...........t."d....c$...s.     ...!.d...,z..DR..#]/>.....!#..\3..s.=.
<.;8."9wV.7V.:.v..D[.`v9>...V..@4..P.e.e..Vn...W*..l...W.;#H....o..A
```

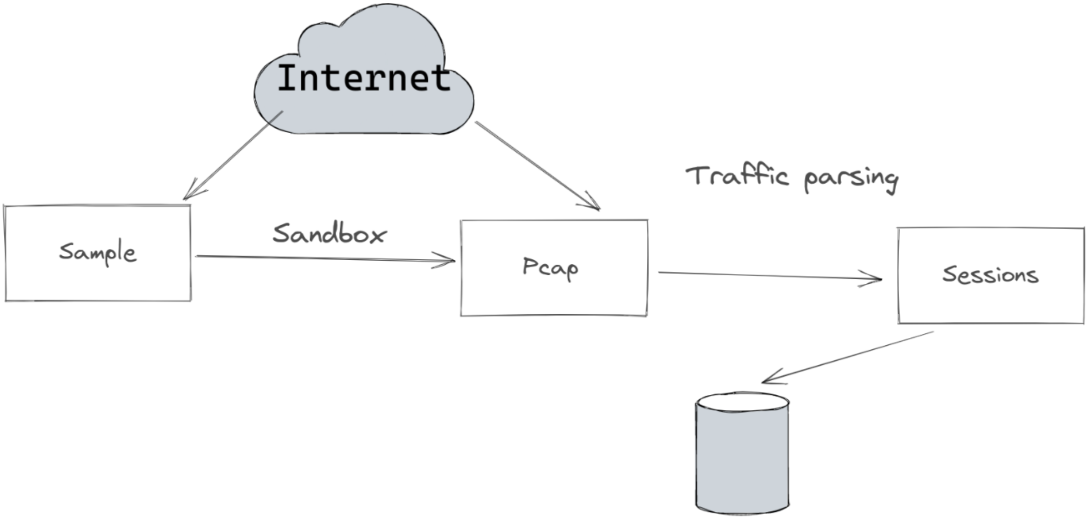
```
POST /api HTTP/1.1
Host: 91.105.192.100:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: ru-RU,en,*
User-Agent: Mozilla/5.0

........H..t."d,...x.F`.s.    ...!.d...,z..!.9....6_. .&..8'......
```

# Definitions

## Objects

- Sample is an executable file ($*.exe, *.so, *.dll$)
- Samples network activity can be saved in pcap file
- Session is a network activity between two IPs

# General data flow pipeline



# Rule based approaches

- Static analysis

# Rule based approaches

- Static analysis
- Dynamic analysis

# Rule based approaches

- Static analysis
- Dynamic analysis
- Network traffic analysis[1]

# Rule based approaches

- Static analysis
- Dynamic analysis
- Network traffic analysis[1]
    - Port based

# Rule based approaches

- Static analysis
- Dynamic analysis
- Network traffic analysis[1]
    - Port based
    - DPI based

# Rule based approaches

- Static analysis
- Dynamic analysis
- Network traffic analysis[1]
  - Port based
  - DPI based
  - Statistics based

# Rule based approaches

- Static analysis
- Dynamic analysis
- Network traffic analysis[1]
  - Port based
  - DPI based
  - Statistics based
  - Behaviour based, e.g. to identify application (web-server)

# ML based approaches

- Flow based
- Hybrd (aggregate static, dynamic, behaviour approaches)

# Pros and cons for rule based approach

## Pros

- Direct solving
- No ML magic, explainable
- High precision

## Cons

- Time to market is very slow
- Constantly needs an infosec expert for rule writing
- Poor recall

# Pros and cons for ML based approach

## Pros

- Time to market is rather fast
- Ideally needs a ML expert only once
- High recall and high precision
- Can detect zero-days*
- Process encrypted traffic*

## Cons

- Needs data
- Some ML magic, has issues with explainability
- Needs some feature engineering
- **We will always have FP**

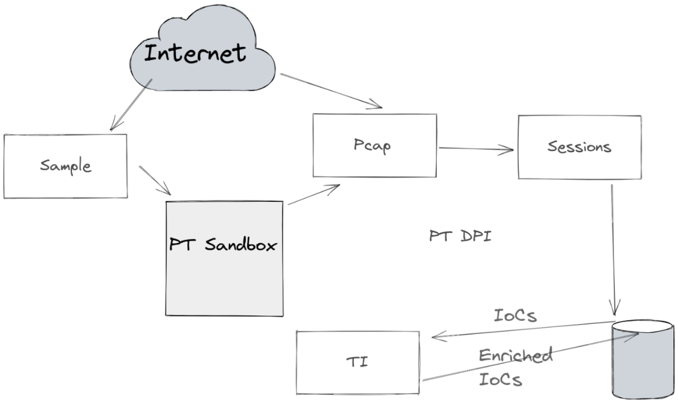
# Proposed approach

# ML task

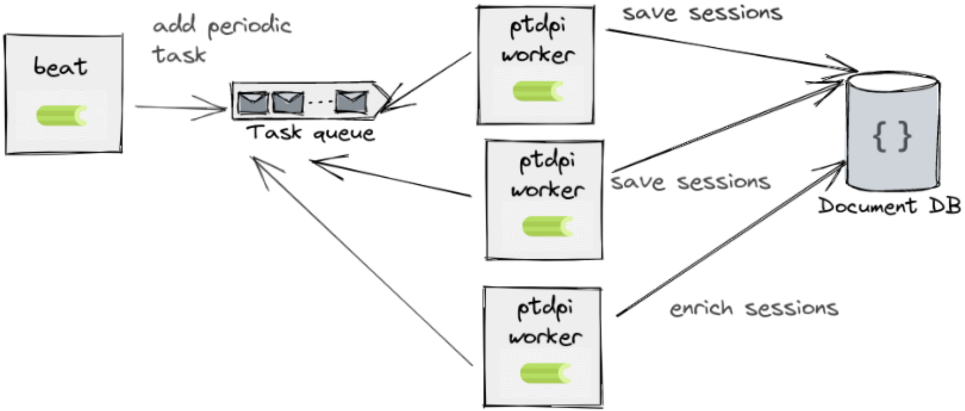
## Objects & labels

- Binary classification task: benign vs malicious
- Object is a tcp session
- Label can be taken from behaviour, static analysis for either sample or pcap

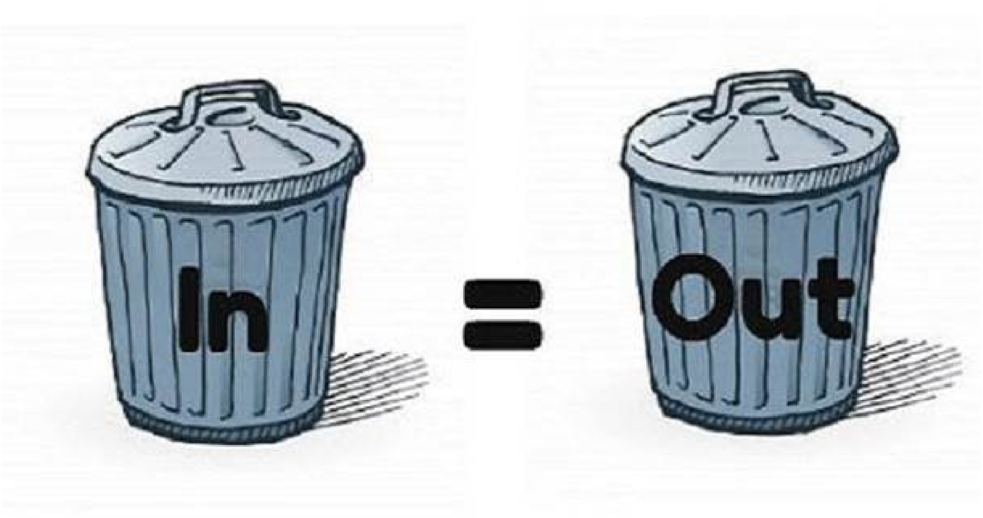
# Dataset collection



# Dataset collection. Detailed



# Dataset labelling. Garbage in, garbage out



# How we can label sessions?

## Each pcap can contain benign or malicious sessions

- **Based on whole pcap label**

# How we can label sessions?

### Each pcap can contain benign or malicious sessions

- **Based on whole pcap label**
- Based on Threat Intelligence (malware hosting problem)

# How we can label sessions?

## Each pcap can contain benign or malicious sessions

- **Based on whole pcap label**
- Based on Threat Intelligence (malware hosting problem)
- Based on triggered rules (infosec expert problem)

# How we can label sessions?

## Each pcap can contain benign or malicious sessions

- **Based on whole pcap label**
- Based on Threat Intelligence (malware hosting problem)
- Based on triggered rules (infosec expert problem)
- Based on IP statistics (tf-idf)

# How we can label sessions?

## Each pcap can contain benign or malicious sessions

- **Based on whole pcap label**
- Based on Threat Intelligence (malware hosting problem)
- Based on triggered rules (infosec expert problem)
- Based on IP statistics (tf-idf)
    - Pcap is a document. Session is a sentence. Destination ip is a token

# How we can label sessions?

## Each pcap can contain benign or malicious sessions

- **Based on whole pcap label**
- Based on Threat Intelligence (malware hosting problem)
- Based on triggered rules (infosec expert problem)
- Based on IP statistics (tf-idf)
  - Pcap is a document. Session is a sentence. Destination ip is a token
  - Some ips are present only for a particular family (CnC).

# How we can label sessions?

## Each pcap can contain benign or malicious sessions

- **Based on whole pcap label**
- Based on Threat Intelligence (malware hosting problem)
- Based on triggered rules (infosec expert problem)
- Based on IP statistics (tf-idf)
  - Pcap is a document. Session is a sentence. Destination ip is a token
  - Some ips are present only for a particular family (CnC).
  - Some ips are present for nearly every family (8.8.8.8). High idf

# Data

## Stats

- $\geq$ 700k pcaps. Daily $+ \sim 5k$ pcaps
- $\geq$ 5kk sessions
- $\geq$ 150 not normalized family names (*rat/redline vs trojan/rat/redline*)

# Features

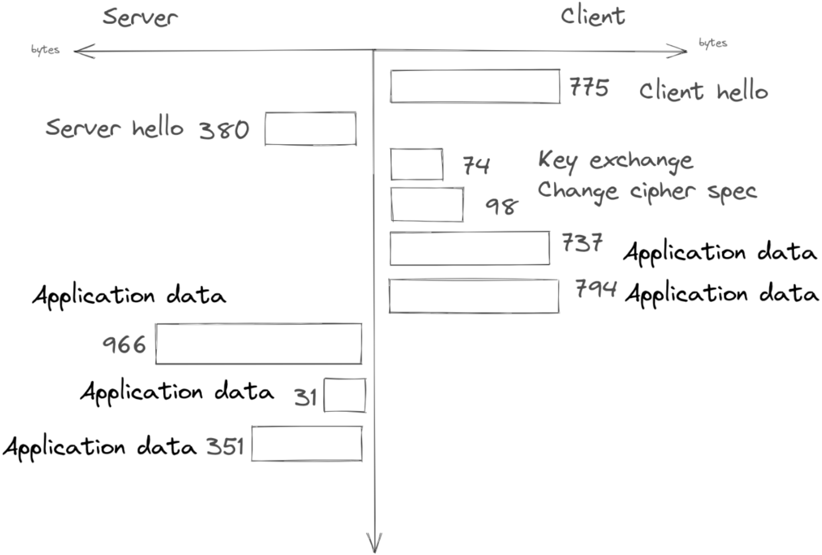
## Tcplen features

- Fixed size vector of tcp payload length in bytes. Max vector size is 30 (configurable)
- Contains packet direction: to server, to client

## Pros of simple features

- fast feature calculation (stream processing)
- fast training
- fast inference

# What is a tcplen vector?



# Features

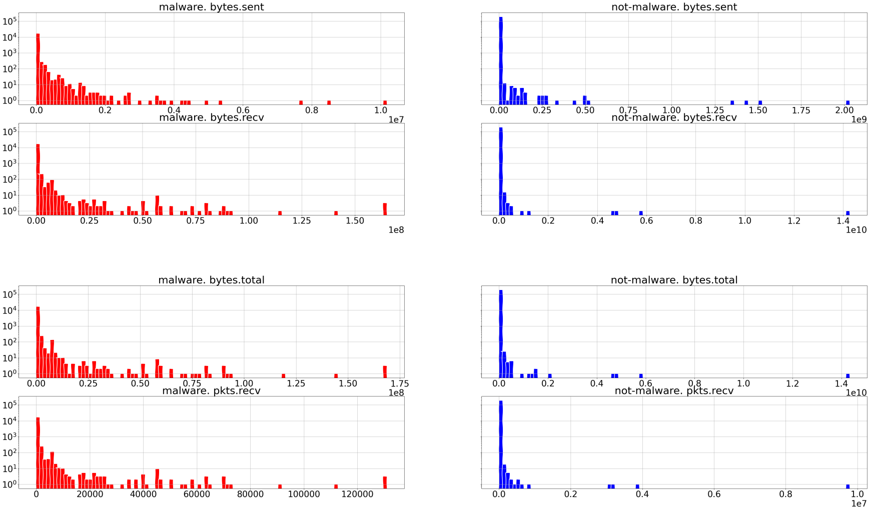
## Aggregated tcplen features

- Define: *tcplen* — padded tcplen_stat array with zeroes, *rcv* — array of bytes' lengths send to server,
  *snd* — array of bytes' lengths send from server.
- Calculate min, max, mean, std, mode for *snd*, *rcv*, *tcplen_raw*
- Join bytes in groups based on max MTU

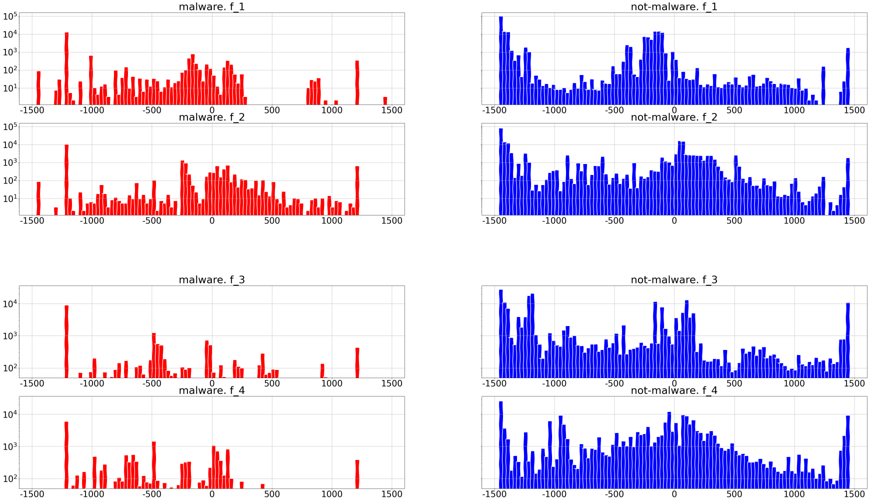
## General features

- Session duration in ms
- Bytes and packets send, recieved and total

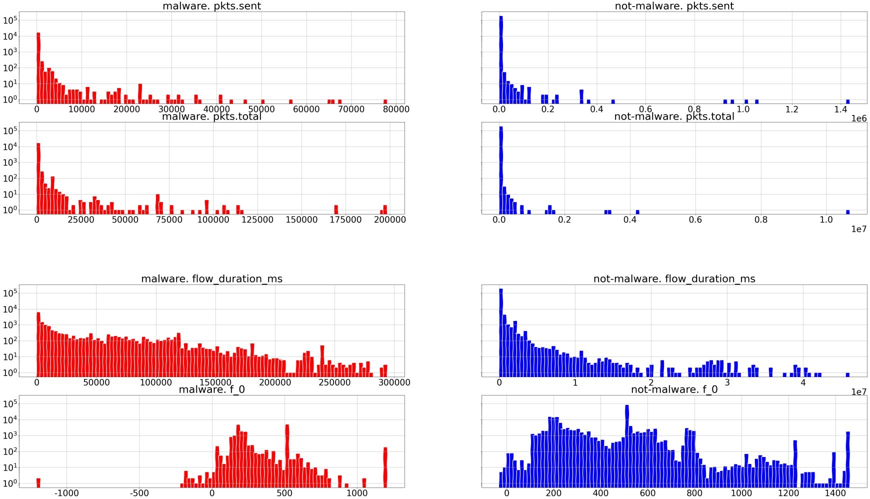
# Features distribution



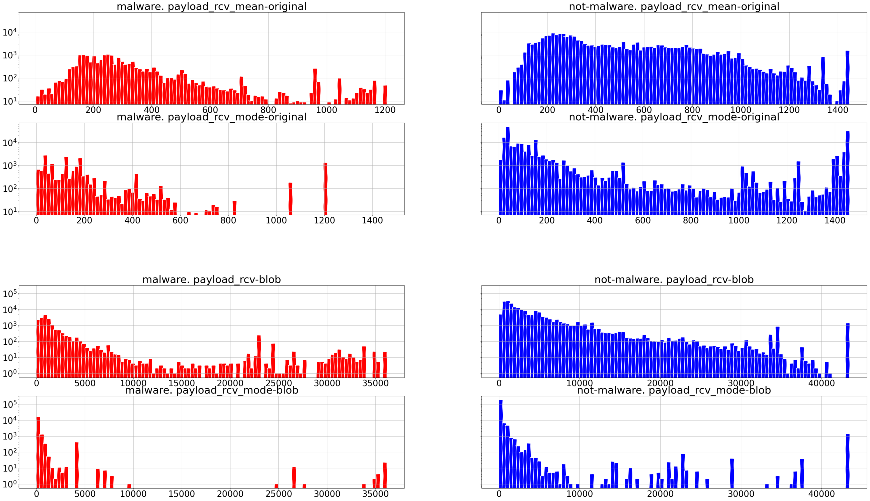
# Features distribution



# Features distribution



# Features distribution



malware. payload_rcv_mean-original

not-malware. payload_rcv_mean-original

malware. payload_rcv_mode-original

not-malware. payload_rcv_mode-original

malware. payload_rcv-blob

not-malware. payload_rcv-blob

malware. payload_rcv_mode-blob

not-malware. payload_rcv_mode-blob

# ML model

### Gradient-boosting

LightGBM is a gradient boosting framework that uses tree based learning algorithms

# Evaluation

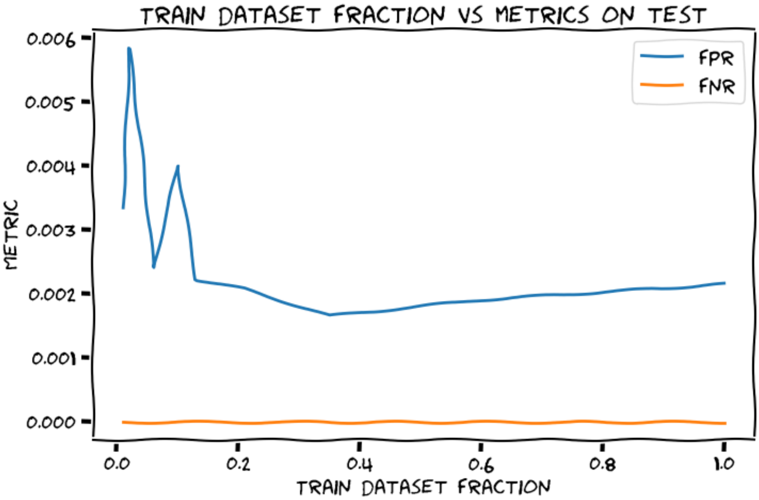
## Offline

- Perform stratified K-fold cross-validation on base dataset. Save offline metrics to mlflow
- Calculate permutation importance
- Select best features
- Tune model hyper-params
- Test best *n* models on future data (4–5 days). Save metrics to mlflow
- Get the best model

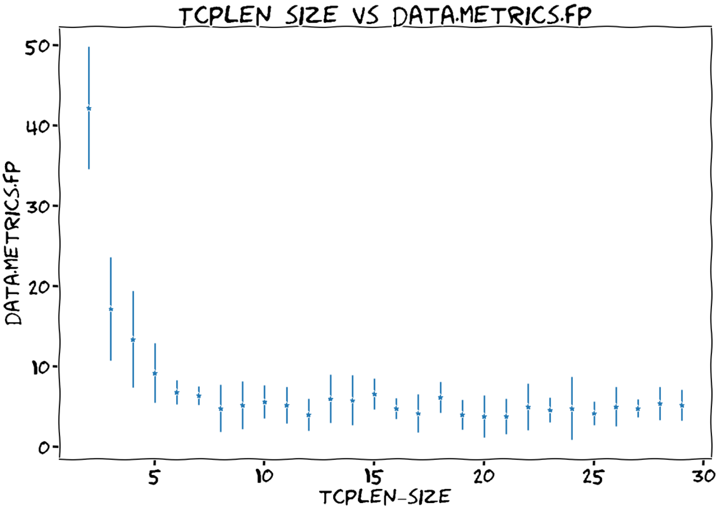
## Online

- Connect some best models to NAD's broker
- Analyze FP per day in grafana

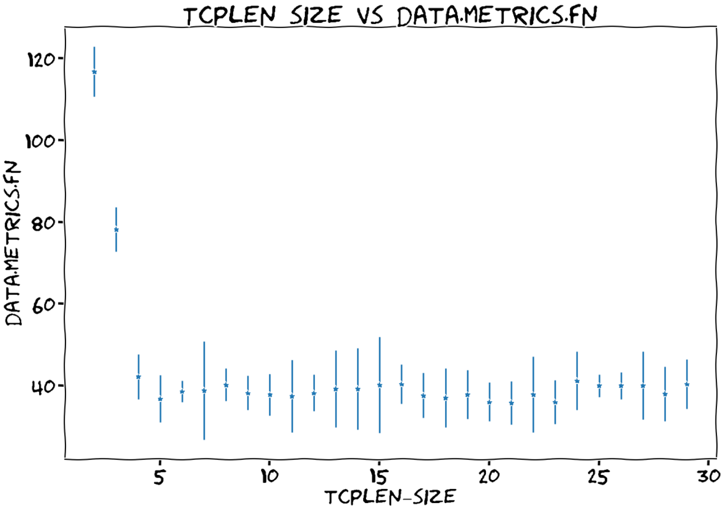
# How much data do we need?



# How many packets do we need?



# How many packets do we need?



# How many fp per day do we have?

## FPR vs FP

- Network consits of $\geq$ 1.5k hosts $+$ $\sim$15k servers, vm
- Daily $\sim$ 30kk sessions

| FP | FPR |
|---|---|
| 10 | 0.9999666 |

# Conclusions

# Main achievements

- Can detect some malware families ($\sim 5$) with low fpr and high precision
- The use of simple features $\rightarrow$ better perfomance
- Proposed approach is general. Can be applied not only for malware detection

# Perspective

## To do

- perform more experiments on recall analysis. New malware family detection
- integrate into NAD
- creatre a fully automated pipeline (no need for infosec expert)

# References I

E. W. Biersack, C. Callegari, and M. Matijasevic.
Data traffic monitoring and analysis.
In *Lecture Notes in Computer Science*, 2013.