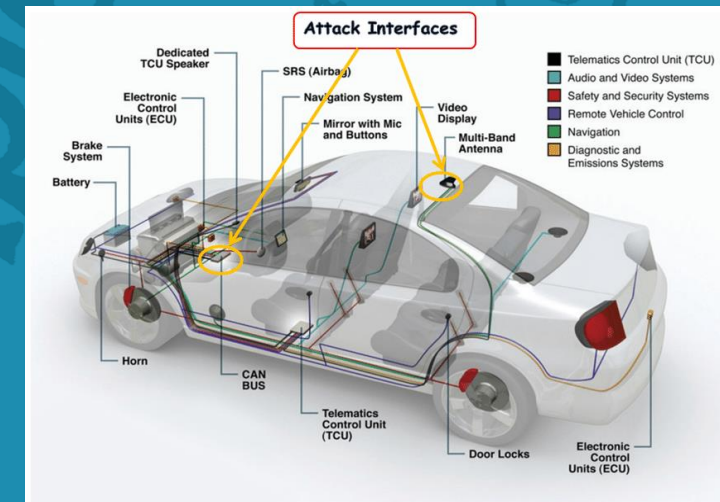


https://github.com/KULeuven-Diepenbeek/res_mol_canids

Development of a statistics-based IDS for automotive security

Wouter Hellemans, Jo Vliegen, Nele Mentens



Reviewing the questions from the preparation



Reviewing the questions from the preparation

Q1: What do messages on the CAN-bus look like, and how can these be parsed with pandas in Python?

Reviewing the questions from the preparation

Q2: What kind of attacks can be performed on CAN networks?

Reviewing the questions from the preparation

Q3: Which statistical parameters can be derived from certain fields in network frames, to detect attacks on CAN?

Reviewing the questions from the preparation

Q4: How do you extract statistical parameters from a dataset?

Short overview of CAN-bus

... and the frames that use it

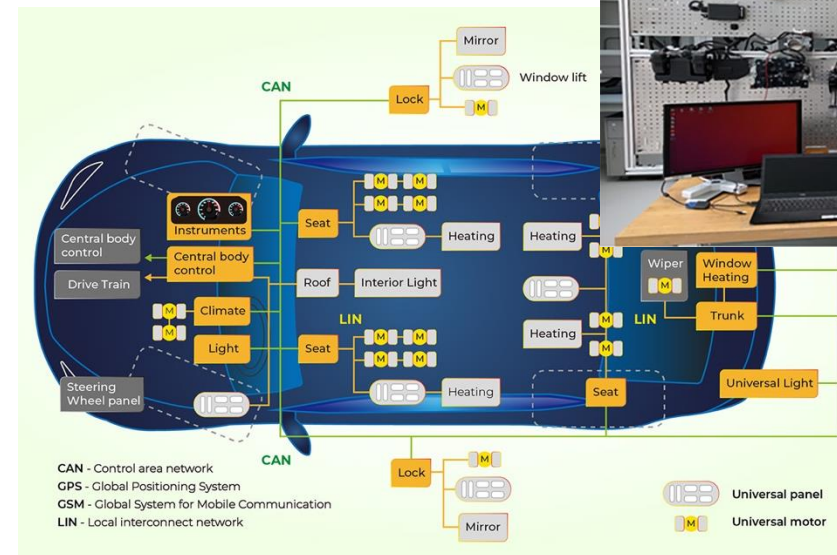
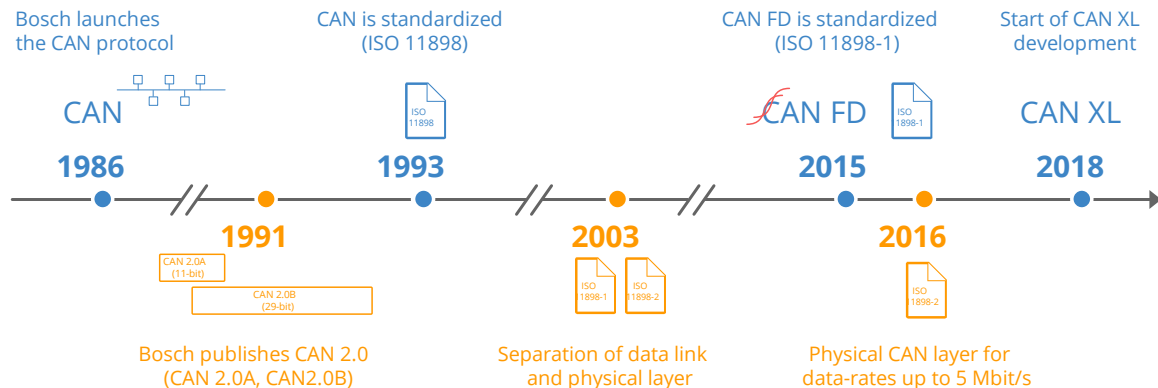


Short overview of CAN-bus

Modern cars can contain up to 150 **Electronic Control Units (ECUs)**

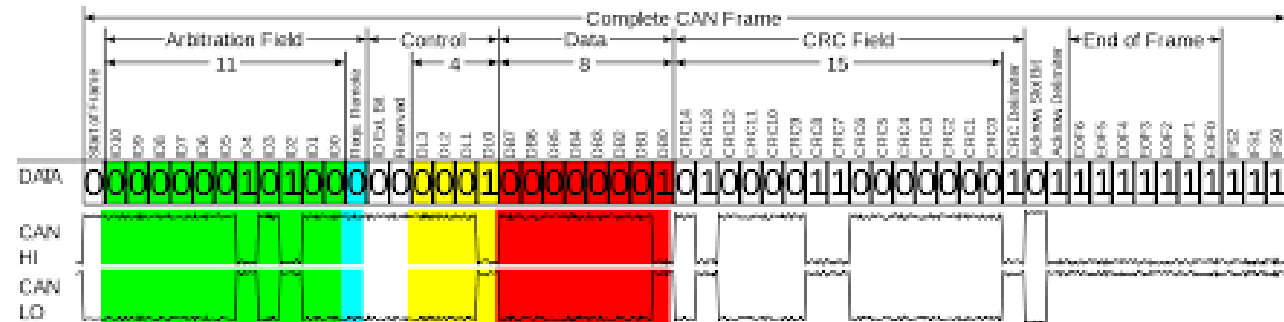
Controller Area Network (CAN): protocol for ECU-to-ECU communication

- 3 generations: CAN CC, CAN FD, and CAN XL
- IP of Robert Bosch GmbH



... and the frames that use it

A message on the CAN bus is called a **frame**



A frame consists (mainly) of 3 fields

- The identifier (**Identifier**)
- Metadata concerning the length of the message (**DLC**)
- The actual data (**Data**)

Attacks on CAN



Hackers took control of a Tesla Model S and turned it off: cnb.cx/1Uqx6UV

Tweet vertalen



11:11 p.m. · 6 aug. 2015



Sam Curry
@samwcyo · Follow

Super excited to release our car hacking research discussing vulnerabilities affecting hundreds of millions of vehicles, dozens of different car companies:

samcurry.net/web-hackers-vs...

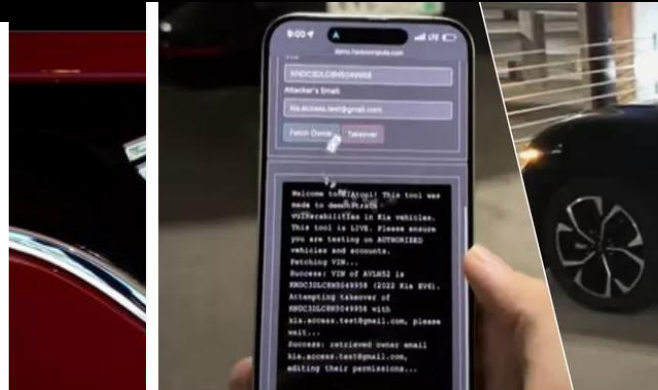
Contributors:

@_specters_ @bbuerhaus @xEHLE_ @iangcarroll, @sshell_ @infosec_au @NahamSec @rez0_



Hackers compromised Tesla vehicle systems twice during three-day Tokyo hacking spree. The same hackers walked away with \$450,000 cash at the Pwn2Own Automotive event.

Post vertalen



De onderzoekers tonen hoe ze een speciale app maakten om miljoenen voertuigen te hacken. © HLN

Miljoenen auto's waren te hacken met eenvoudige truc: "Traceren, ontgrendelen en starten"



CNN Business
@CNNBusiness · Follow

Recall Alert: Fiat Chrysler is recalling 1.4 million hackable vehicles. Check affected cars: cnnmon.ie/1OrrqGv



2:59 AM · Jul 25, 2015



CAN is vulnerable to cyberattacks such as: DoS, Fuzzing, Spoofing, Replay...

11:07 AM · Jan 3, 2023



wielen", zegt HLN-techexpert Kenneth Dée.

Kenneth Dée 27-09-24, 15:00 Laatste update: 28-09-24, 08:26

Datasets & analysis



Datasets

We'll be doing experiments on a **dataset** from HCRL (Hacking and Countermeasure Research Lab):

A "normal_data_set" is available in .txt format.

```
≡ normal_run_data.txt ×  
hcrl > ≡ normal_run_data.txt  
1   Timestamp: 1479121434.850202      ID: 0350    000    DLC: 8    05 28 84 66 6d 00 00 a2  
2   Timestamp: 1479121434.850423      ID: 02c0    000    DLC: 8    14 00 00 00 00 00 00 00  
3   Timestamp: 1479121434.850977      ID: 0430    000    DLC: 8    00 00 00 00 00 00 00 00  
4   Timestamp: 1479121434.851215      ID: 04b1    000    DLC: 8    00 00 00 00 00 00 00 00
```

Also "attacked" datasets can be found:

1. DoS Attack : Injecting messages of '0000' CAN ID every 0.3 milliseconds. '0000' is the most dominant.
2. Fuzzy Attack : Injecting messages of totally random CAN ID and DATA values every 0.5 milliseconds.
3. Spoofing Attack (RPM/gear) : Injecting messages of certain CAN ID related to RPM/gear information every 1 millisecond.

Datasets

We'll be doing experiments on a **dataset** from HCRL (Hacking and Countermeasure Research Lab).

The dataset can be downloaded from this URL:

<https://drive.google.com/drive/folders/1ed2PlvcSu9ONt-8KK3sgG4Qw1Bp0ccOr?usp=sharing>

Example – How to detect malicious frames?

Maybe the Hamming distance between 2 subsequent frames might be an indicator?

Hamming distance = number of positions in which two (equally sized) inputs differ. For example:

- `Hello Jim`
`Hello Tim` Hamming distance: 1
- `31 = 0b01 1111`
`32 = 0b10 0000` Hamming distance: 6

Example – How to detect malicious frames?

Maybe the Hamming distance between 2 subsequent frames might be an indicator?

The two messages: 0350_8_052884666d0000a2
02c0_8_1400000000000000

The two messages, in binary:

```
0000001101010000_1000_0000010100101000100001000110011001101101000000000000000010100010
0000001011000000_1000_000101000000000000000000000000000000000000000000000000000000000
```

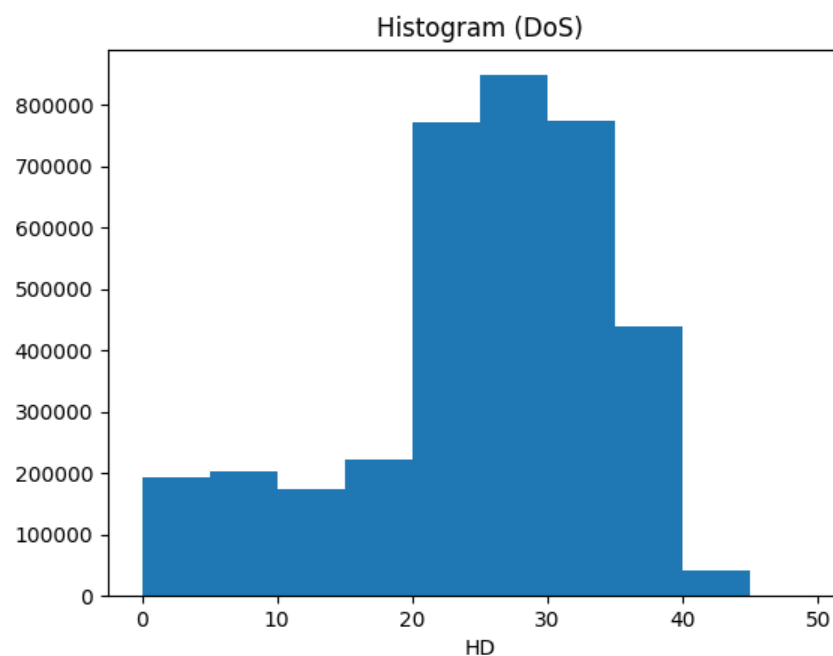
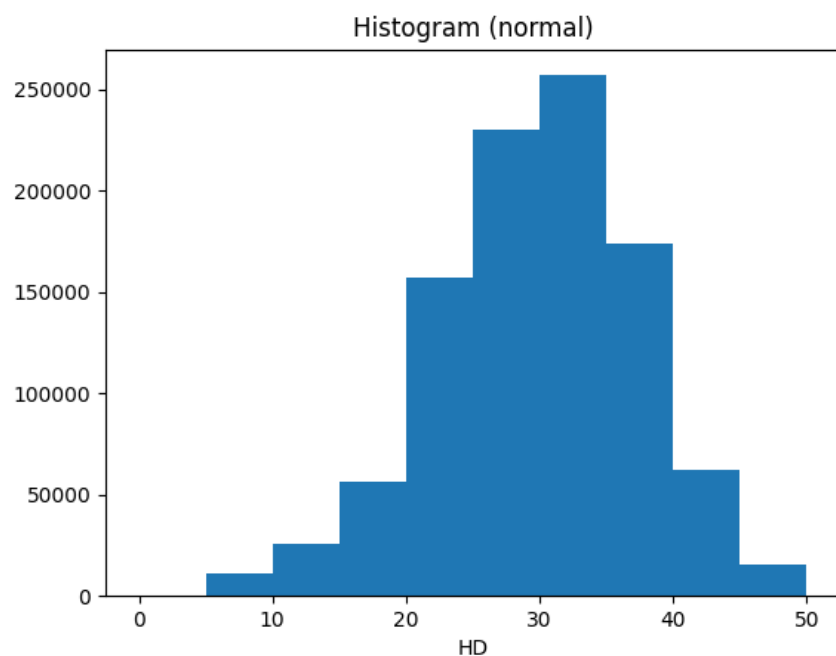
... and exored ...

```
0000000110010000_0000_0001000100101000100001000110011001101101000000000000000010100010
```

... shows 21 bits of value '1', so the Hamming distance is **21**

Example – How to detect malicious frames?

Maybe the Hamming distance between 2 subsequent frames might be an indicator?



... now you try

Can you find a statistical feature that MIGHT be an indication?



Example

Our "*hypothesis*" is that if the HD with the previous frame is < 10 , the message is classified as "malicious"

When applied to the first 100 frames of the DoS dataset, two frames are marked as "malicious".

How to evaluate the performance of our IDS?

Evaluation

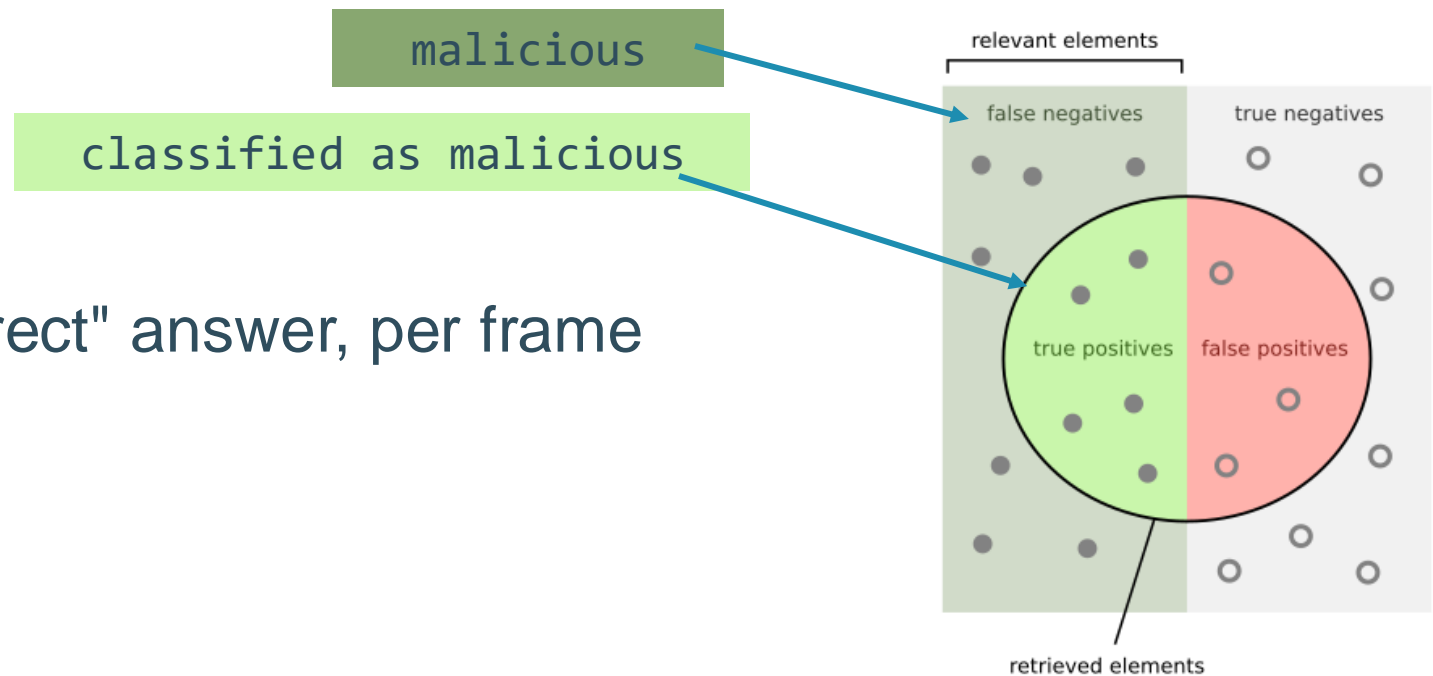
The dataset comes with the "correct" answer, per frame

T: injected message

R: normal message

For every frame, the datasets also provides: ok / not ok

	T	R
Ok	False negative	True negative
Not ok	True positive	False positive



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

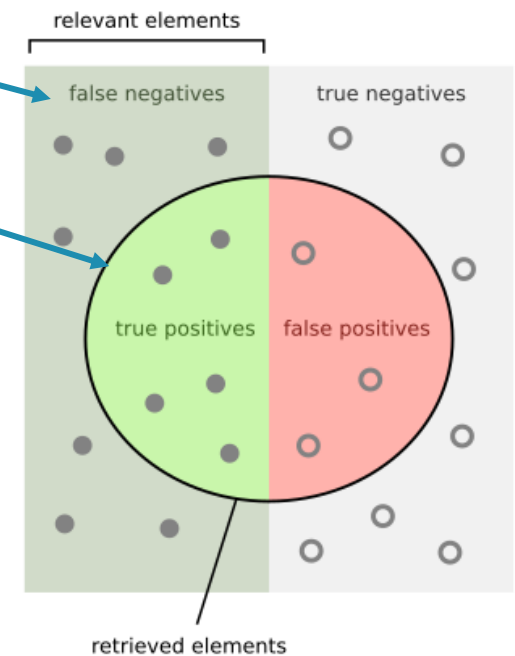
$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Evaluation

malicious
classified as malicious

The dataset comes with the "correct" answer, per frame

	T	R
Ok	False negative	True negative
Not ok	True positive	False positive



Precision: how good are your positives ?

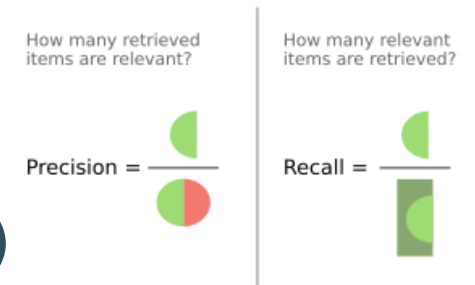
Accuracy: how good are your decisions, overall?

Recall: how good did you cover all positives?

$$Tp / (Tp + Fp)$$

$$(Tp + Tn) / (Tp + Tf + Fp + Fn)$$

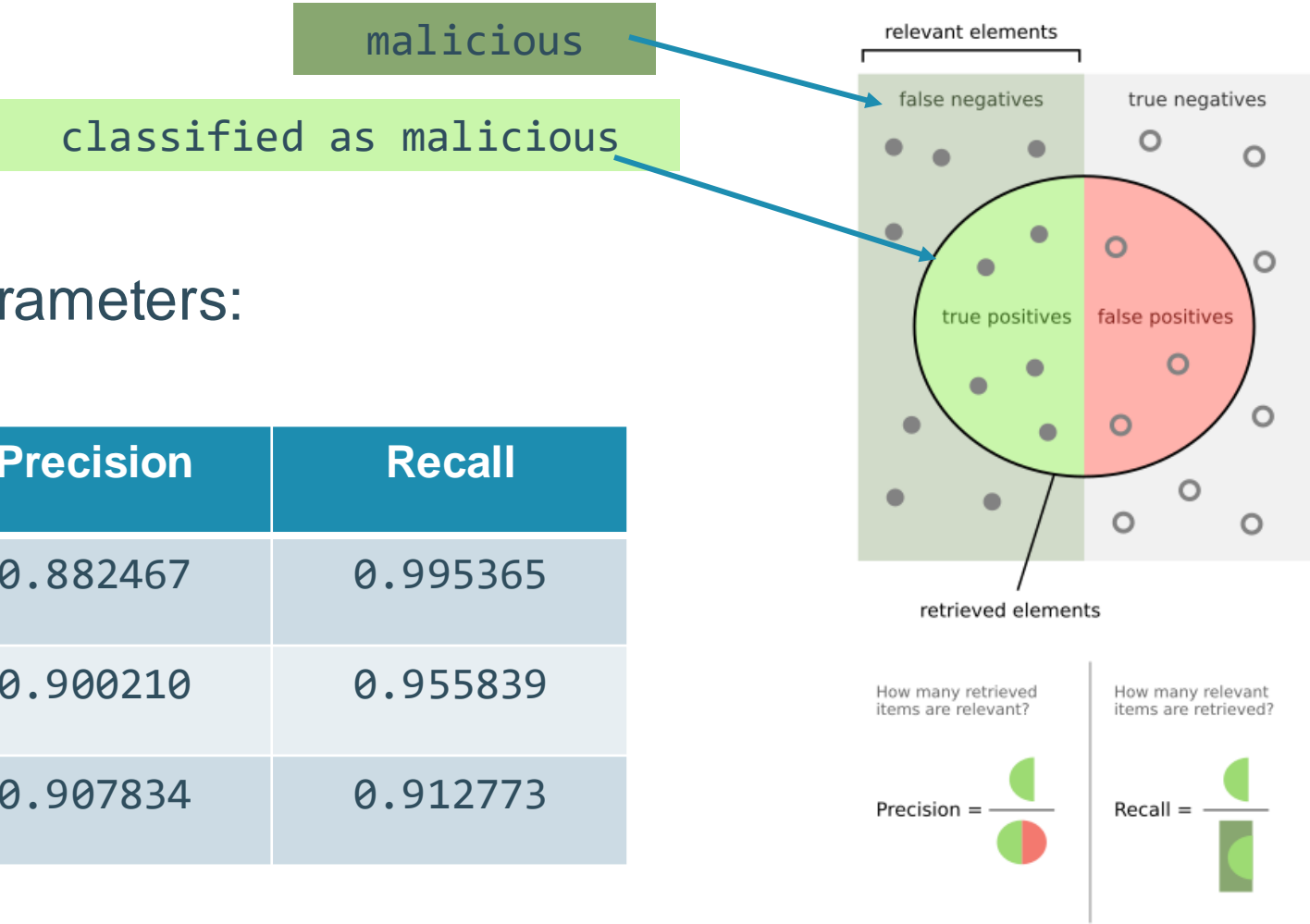
$$Tp / (Tp + Fn)$$



Evaluation

If we experiment with different parameters:

HD	Accuracy	Precision	Recall
5	0.884785	0.882467	0.995365
10	0.873942	0.900210	0.955839
15	0.848938	0.907834	0.912773



... now you try

How do your hypotheses score?



Combined decision making

How would make a decision when multiple models are at work ?

- Classify as malicious when 1 out of n models classifies as malicious
- Classify as malicious when all n models classify as malicious
- Classify as malicious when $\geq n/2$ out of n models classify as malicious

Q&A



That's it

We hope you had fun (and have learned something :))

