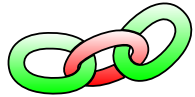
Machine Learning Methods for In-Vehicle Intrusion Detection

Roland Rieke

ITMO, 2018

Overview



- Security Challenges for Connected Vehicles
- Security Measuring
- Anomaly Detection
- Machine Learning Methods
- Data Sets
- Machine Learning Evaluation

“If I had asked people what they wanted, they would have said faster horses.”

Henry Ford

Challenge 1898

Faster Horses?

- 1st international urban-planning conference in NYC 1898
  Topic: growing crisis posed by urban horses and their output
- London 1900: 11.000 cabs + X.000 buses (each 12 horses/day) > 50.000 horses

- London Times 1894: in 50 years streets buried under 9 feet of manure
- No solution – the conference was abandoned after 3 days (scheduled 10)
- Unexpected solution – transition from horses to motor vehicles

Source: Stephen Davies, “The Great Horse-Manure Crisis of 1894”

Transition from relatively isolated autonomous driver-vehicle systems to massively (inter)connected driverless vehicles & global ecosystem.

Challenges for Connected Vehicles

- more efficient (reduce pollution)
- aware of the situation (but keep privacy)
- secure (despite of increased attack surface)
- robust against new threats (faking AI or sensors)
- autonomous (e.g. handle ecosystem failures)

2018 Problem: Air Pollution

Connected Cars

Vehicular Ad Hoc Network (VANET) / Inter-Vehicle Communication (IVC)

VANET:	Mobile ad-hoc network whose nodes are vehicles.
Modes:	Car-2-Car and Car-to-Infrastructure,e.g. Road Side Units
Characteristics:	self-organising, decentral
Applications:	Platooning, electronic brake lights, traffic info systems, safety warning
Technology:	WAVE (Wireless Access in Vehicular Environments);VVLN (Vehicular Visible Light Network)

Internet of Vehicles (IoV)

IoV:	Highly integrated IoT manifestation with respect to vehicular Ecosystem
Extend VANET to:	Humans (V2H), Sensors (V2S), Clouds (V2C), Internet (V2I)
Technology:	Mobile Internet connection (GPRS, . . .), GPS

“Once you add a Web browser to a car, it’s over, ”

Charlie Miller, Black Hat USA 2014

Connectivity Enables Attacks

Attacks on safety

- Unauthorized brake
- Attack emergency call
- Inflate airbags

Attacks on privacy

- Trace vehicle movement
- Compromise driver privacy

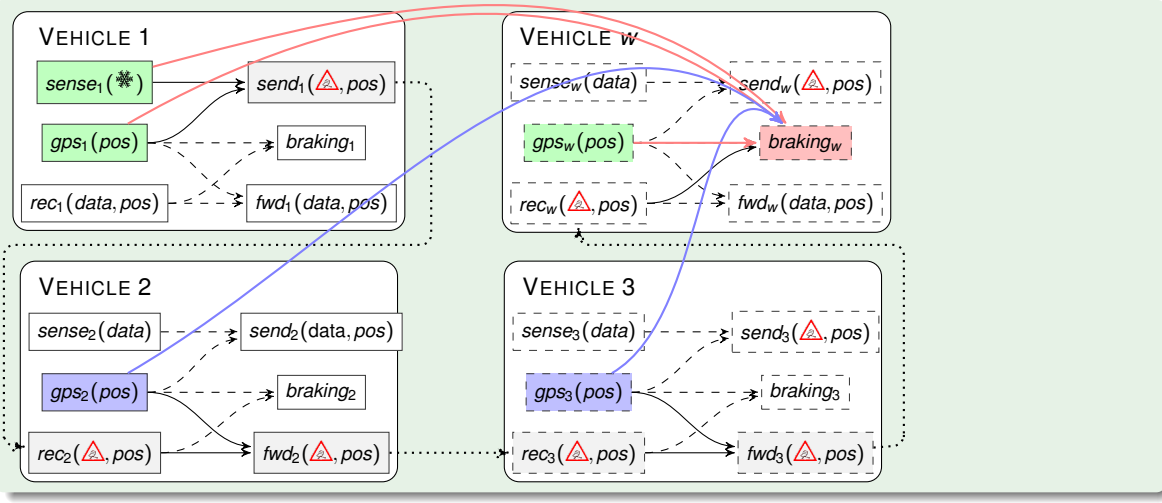
Manipulate traffic flow

- Simulate traffic jam
- Force green lights ahead
- Manipulate speed limits

Economic Advantage

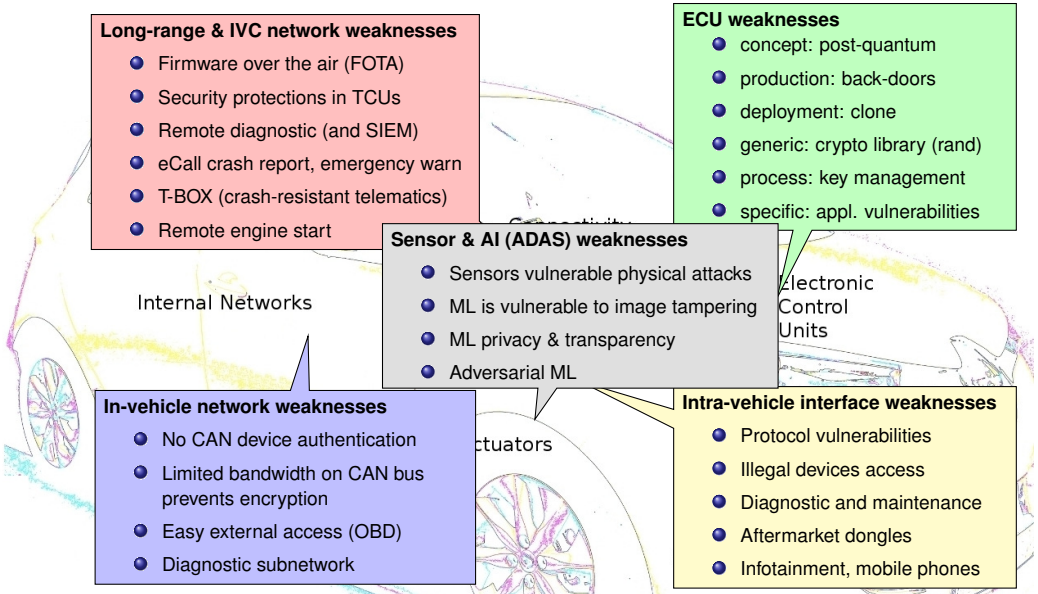
- Steal car
- Change driver's toll bill
- Manipulate e-charging

Example: Security Dependencies in Systems of Systems



$$Authenticity_i = Authenticity_{i-1} \cup \{auth(gps_i(pos), braking_w, Driver_w)\}$$

Security Risks - Connected Vehicles



Long-range & IVC network weaknesses

- Firmware over the air (FOTA)
- Security protections in TCUs
- Remote diagnostic (and SIEM)
- eCall crash report, emergency warn
- T-BOX (crash-resistant telematics)
- Remote engine start

ECU weaknesses

- concept: post-quantum
- production: back-doors
- deployment: clone
- generic: crypto library (rand)
- process: key management
- specific: appl. vulnerabilities

Sensor & AI (ADAS) weaknesses

- Sensors vulnerable physical attacks
- ML is vulnerable to image tampering
- ML privacy & transparency
- Adversarial ML

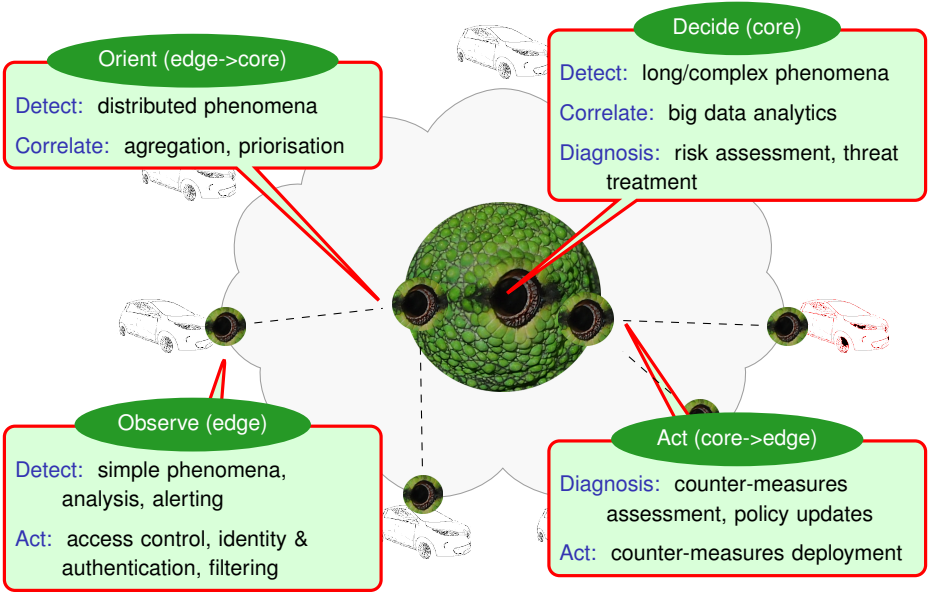
In-vehicle network weaknesses

- No CAN device authentication
- Limited bandwidth on CAN bus prevents encryption
- Easy external access (OBD)
- Diagnostic subnetwork

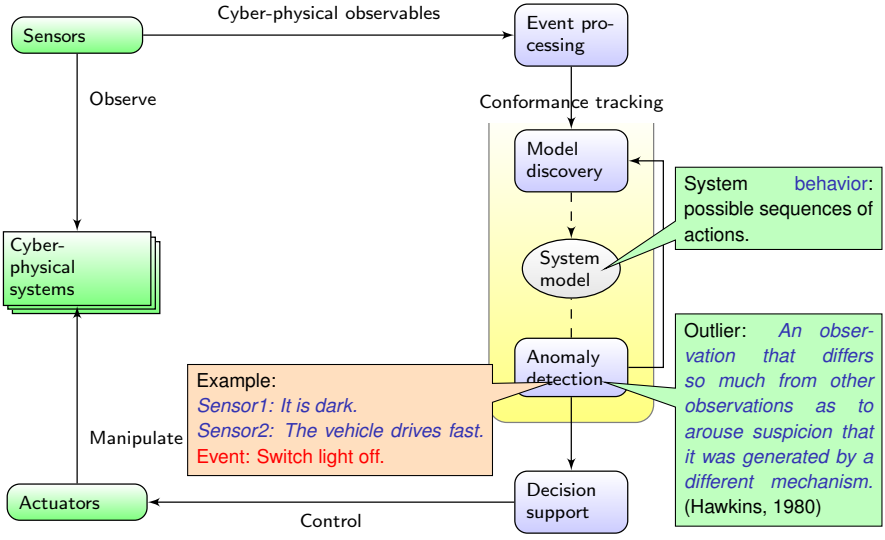
Intra-vehicle interface weaknesses

- Protocol vulnerabilities
- Illegal devices access
- Diagnostic and maintenance
- Aftermarket dongles
- Infotainment, mobile phones

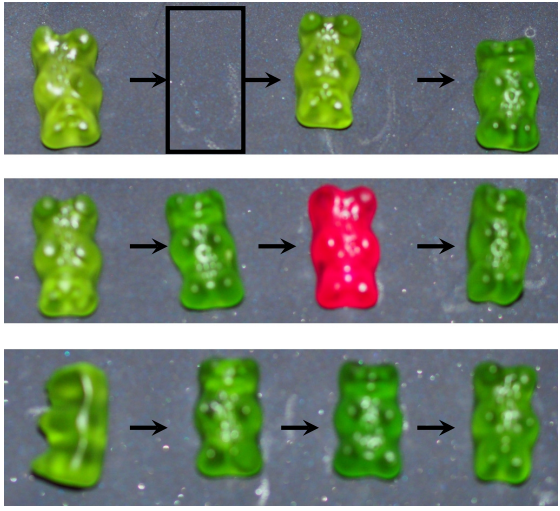
Automotive Threat Intelligence Framework



Conformance Tracking: Expected vs. observed behavior



Anomaly Detection (Behavior-based)



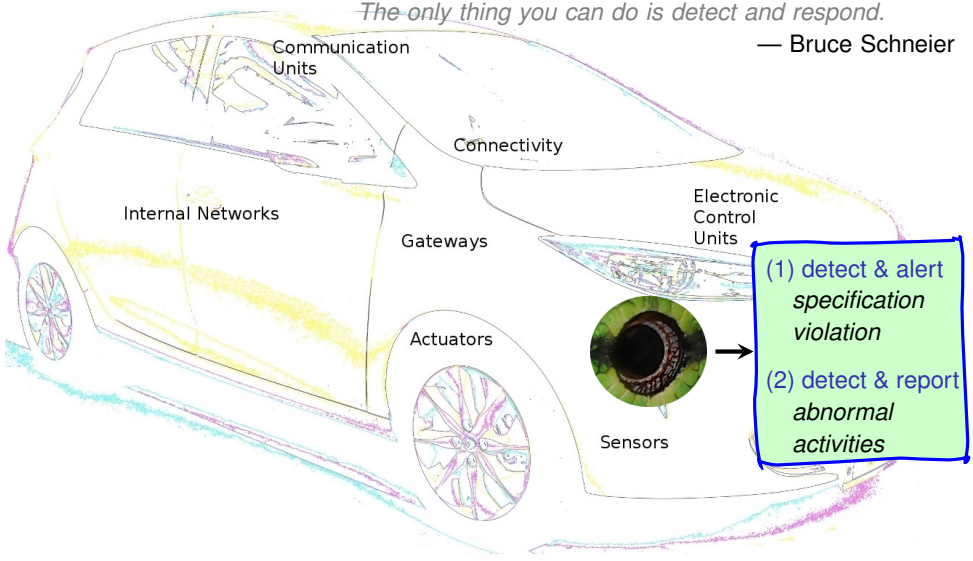
Behavior requirements:

- cyclic messages,
- protocol flow,
- process behavior,
- subsequent payload dependencies.

The behavior of a discrete system can be formally described by the set of its possible sequences of actions.

On-board Security Analysis (Observe at the Edge)

You can't defend. You can't prevent.
The only thing you can do is detect and respond.
— Bruce Schneier



CAN intrusion detection methods

Detect specification violations

- Formality, Location, Range
- Sequence (Frequency, Correlation, Protocol)
- Semantic (Plausibility, Consistency)

Detect ECU impersonation

- ECU voltage fingerprinting
- ECU clock skew fingerprinting
- ECUs check messages with own ID (parrot defense)
- remote frame (response time)

Detect packet insertions

- entropy + state
- time interval
- OCSVM (DoS insert / delete packets)
- LSTM

Detect behavior anomalies

- deep learning (e.g. LSTM)
- OCSVM
- hidden Markov
- entropy
- process mining

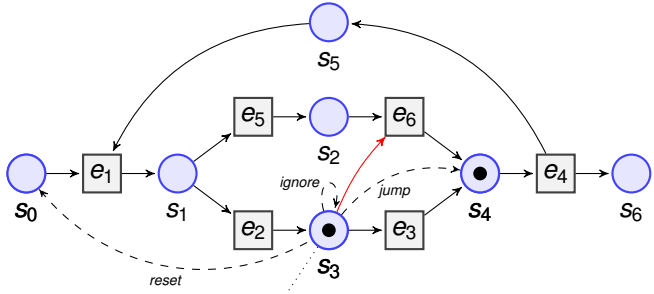
Behavior-based Models

- Construction of Models
  - from specifications
  - from logged behavior without attacks (process mining, OCSVM)
  - from logged behavior with marked attacks (SVM, neural networks)
- Monitoring
  - At operation time, the event stream is compared to the expected behavior (represented by model).
  - Anomalies indicate possible attacks
  - Unknown types of attacks can be detected
- Problems:
  - Overfitting/Underfitting
  - False positives
  - state space explosion (model construction)
  - insufficient throughput (classification of event stream)

Process Mining & Synchronization

Training set: e_1 , e_2 , e_3 , e_4 , e_1 , e_5 , e_6 , e_4

Resulting model (Petri net generated by process mining with alpha algorithm):

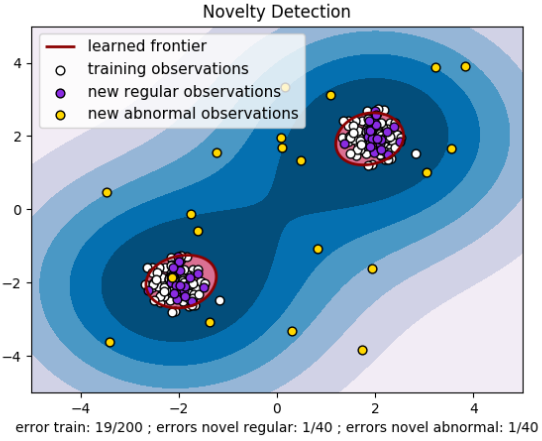


Conformance checking: e_1, e_2 , e_6

ignore e_6 and continue from s_3
reset after e_6 and continue from s_0
jump to some place reachable by transition e_6, e.g. s_4

(One-Class) Support Vector Machine

- Classic
  - Linear classifier
  - “Max-margin”
  - Resource efficient
- One-Class
  - Novelty/Outlier Detection
  - Boundary of seen data

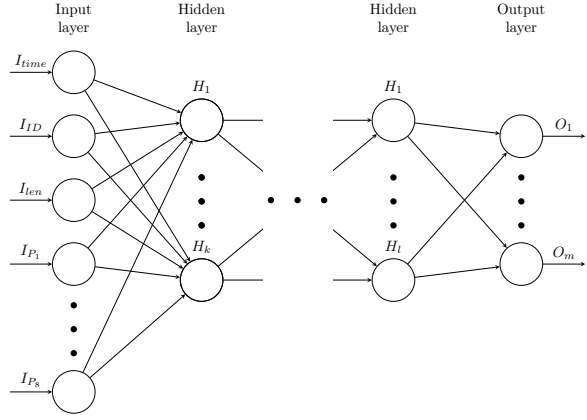


Computed by: `http:`

`//scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html`

Neural Network

- Layers of Neurons
- Non-linear classifier
- Many parameters
- Very flexible



Long Short Term Memory (LSTM) Neural Network

- Very complex
- Computationally intensive
- Models temporal relationships

Methodology

1. Data Preprocessing
2. Creating Train/Test Split
3. Fit Model using Training-Set
4. Validate Model using Test-Set
5. Visualization
6. Real-time classification of data-stream

Data Sets

ZOE Data Set

- Collected from Renault Zoe electric car
- about 10 Minutes; 1.000.000 messages

HCRL Data Sets

- Made available by Hacking and Countermeasure Research Lab
- 4 Data sets, 3.5 to 4.5 Million Messages
- DoS, Spoofing/Impersonation (Fuzzy, Gear), and RPM Attacks

HCRL: `http://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset`

Attacks in Data Sets

ZOE

time	ID	len	p1	p2	p3	p4	p5	p6	p7	p8	type
0.0	530	6	254	61	192	108	0	0	117	118	1
0.000206	394	6	255	240	0	6	64	0	117	118	1

HCRL DoS

0.852103	0	8	0	0	0	0	0	0	0	0	-1
0.852353	1349	8	216	0	0	138	0	0	0	0	1
0.852599	0	8	0	0	0	0	0	0	0	0	-1

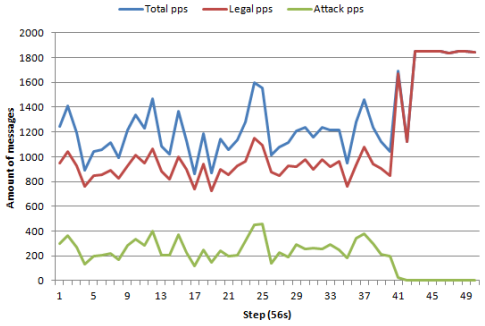
HCRL Gear (RPM is similar)

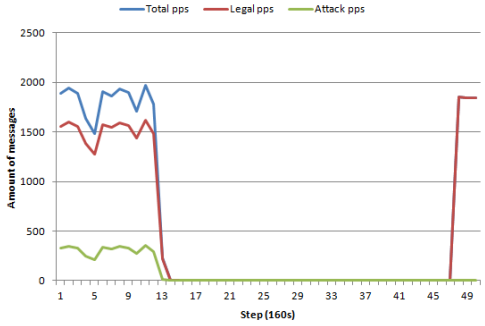
1.348859	1087	8	1	69	96	255	107	0	0	0	-1
1.349731	848	8	5	32	180	104	119	0	0	142	1
1.349963	1087	8	1	69	96	255	107	0	0	0	-1

HCRL Fuzzy

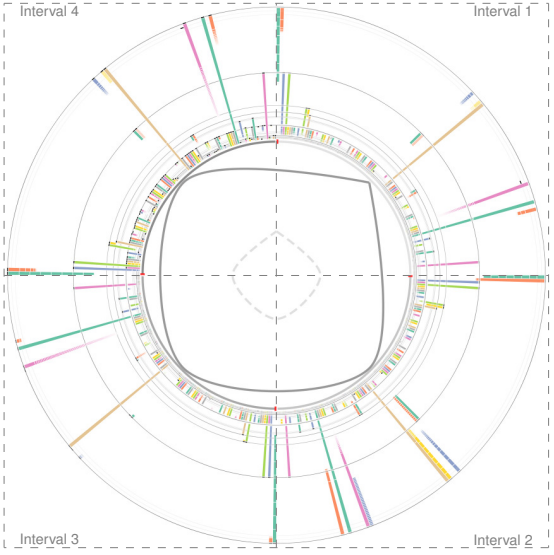
0.972222	1869	8	68	51	82	16	80	85	48	212	-1
0.977422	1087	8	16	64	96	255	125	146	9	0	1
0.982961	1139	8	148	217	62	32	201	26	23	44	-1

Data Sets: DoS vs. Gear Attack Distribution in HCRL Data

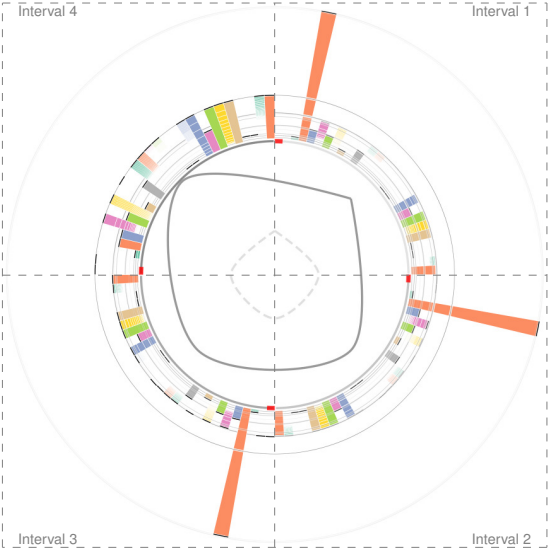




Data Sets: Attack visualization by radial time intervals



(a) *ZOE* data without attacks



(b) $HCLR_{DoS}$ attacks (orange bars)

Process Mining Problem (Alpha Algorithm): Construction time and size of models

Petri net model discovery (alpha algorithm)						
Model	Start	Events	Time	Tran.	Places	Edges
M500	0	500	1.176	83	112	455
M1000	200.000	1.000	10.611	97	232	1.423
M1000'	490.000	1.000	1.717	95	170	879
M1000"	700.000	1.000	2.647	95	170	849
M2000	200.000	2.000	11.333	104	317	2.056
M2000'	490.000	2.000	3.233	102	271	1.337
M2000"	700.000	2.000	29.213	101	313	1.819
M3000	200.000	3.000	65.473	104	566	4.603
M3000'	490.000	3.000	235.250	104	623	4.899
M4000	200.000	4.000	75.018	105	537	3.816
M4000'	490.000	4.000	1.671.779	105	900	7.994

Start: Position in the logfile where the first event for the model is taken.
Events: Number of consecutive events used for the model discovery.
Time: Maximum time in milliseconds for the generation of the model.
Trans., Places, Edges: Complexity of the generated Petri net.

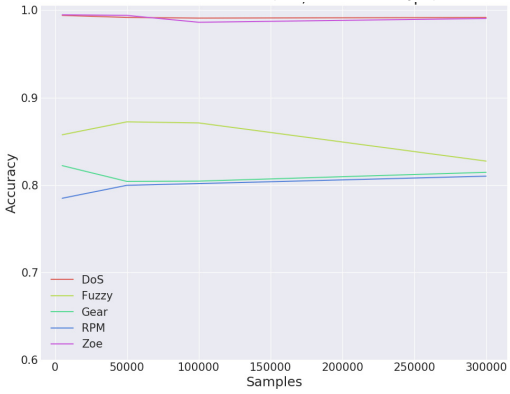
Model quality: Process Mining (Anomaly rate for strategy Ignore)

	Model	Unknown	Unanticipated	Ignored (sum)
Testdata: 1.000.000 events	M500	96.938	5.210	102.148 (10,21%)
	M1000	3.102	9.066	12.168 (1,22%)
	M1000'	4.158	880	5.038 (0,50%)
	M1000"	4.170	645	4.815 (0,48%)
	M2000	457	9.045	9.502 (0,95%)
	M2000'	809	490	1.299 (0,13%)
	M2000"	985	1.130	2.115 (0,21%)
	M3000	457	8.844	9.301 (0,93%)
	M3000'	457	451	908 (0,09%)
	M4000	280	8.623	8.903 (0,89%)
	M4000'	280	455	735 (0,07%)

- Very low anomaly rates when the model is carefully adjusted.

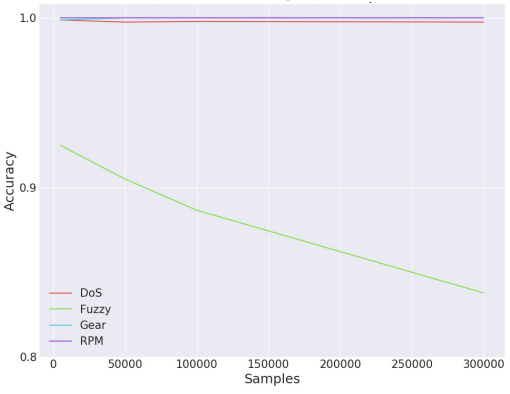
Model quality: (One-Class) Support Vector Machines

One-Class SVM Results (no timestamps)



- OCSVM detects clear outliers

SVM Results (no timestamps)

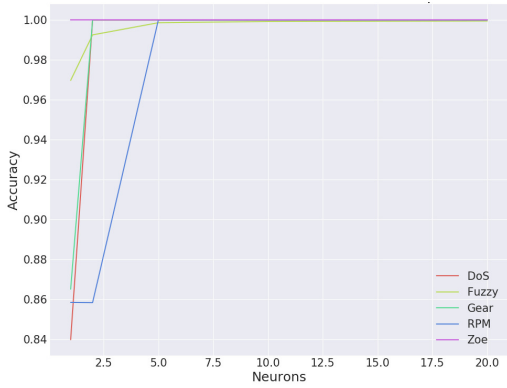


- SVM can classify simple attacks

Both struggle with randomized attacks

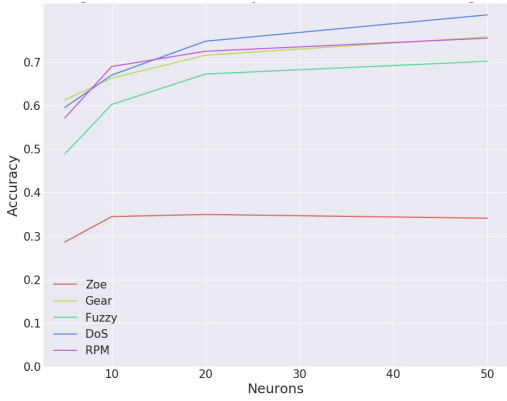
Model quality: Fully-Connected Neural Networks

- Good results with small networks
- Diminishing return for more coplex networks
- Can distinguish between random and regular data
- “Learns” ECU behaviour



Model quality: LSTM Neural Networks

- Can learn temporal behaviour partly
- Mostly periodic CAN IDs
- No external triggered Events
- Needs more analysis
- Computationally complex (memory error on 200GB machine)



Findings: Process Mining (simple alpha algorithm)

- Model preparation: state space explosion problems
- Model execution: resource efficient
- Model synchronization: further research needed
- Model quality:
  - In unsupervised learned model normal behaviour wrongly classified as anomaly is highly dependent on synchronization strategy
  - Detection rate (false positives/negatives with respect to attacks) not yet evaluated

Findings: (One-Class) Support Vector Machine

- Potential to detect simple intrusions/faults
- Resource efficient
- Better results with improved versions
  - Andreas Theissler: Anomaly detection in recordings from in-vehicle networks
- Pure classification of very limited use

Findings: Neural Networks

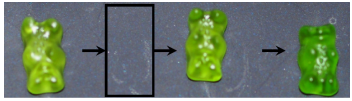
- Very good classifier at low complexity
- Tool for specification extraction
- Detection known and random attacks
- Not suited for general anomaly detection

Findings: Long Short Term Memory (LSTM) Neural Network

- Can get very complex
- Detection of simple temporal behaviour
- Did not detect missing events
- Successful application of more complex networks
  - Chockalingam et al. 2016: Detecting Attacks on the CAN Protocol With Machine Learning
  - Taylor, Leblanc and Japkowicz 2018: Probing the Limits of Anomaly Detectors for Automobiles with a Cyber Attack Framework

Lessons Learned

- Machine learning is a viable option
- Applicable on existing architecture design
- More analysis with more driving situations and sophisticated attacks needed
- Actual deployment on embedded systems may face performance problems
- Problems with long term phenomena:



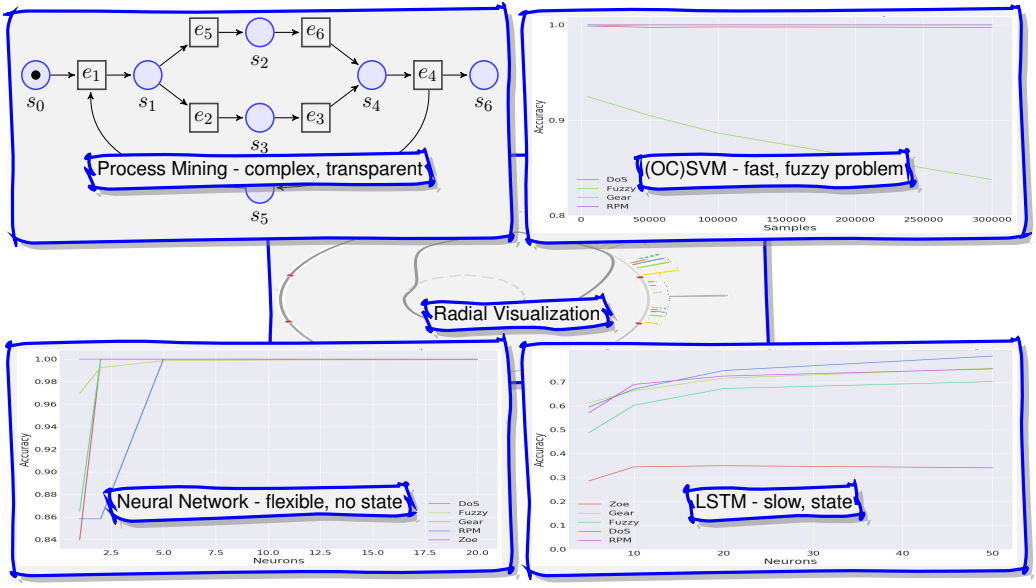
- ML robustness, transparency, explainability
- Other Results from Literature
  - Taylor et al. 2017: one-step Markov model is not much better than guessing, two-step model is worse
  - Choi et al. 2018, Cho et al. 2016: Detection of ECU impersonating attacks by physical (voltage) fingerprinting

Publications

Ivo Berger, Roland Rieke, Maxim Kolomeets, Andrey Chechulin, Igor Kotenko.
Comparative Study of Machine Learning Methods for In-Vehicle Intrusion Detection.
4th Workshop on the Security of Industrial Control Systems & of Cyber-Physical Systems (CyberICPS 2018) in conjunction with ESORICS 2018
Submitted to Springer LNCS

Roland Rieke, Marc Seidemann, Elise Kengni Talla, Daniel Zelle, Bernhard Seeger.
Behavior Analysis for Safety and Security in Automotive Systems.
25th Euromicro International Conference on Parallel, Distributed and Network-based Computing (PDP 2017), IEEE, 2017

Conclusions – ML for In-vehicle Intrusion Detection



Intrusion Detection in Context