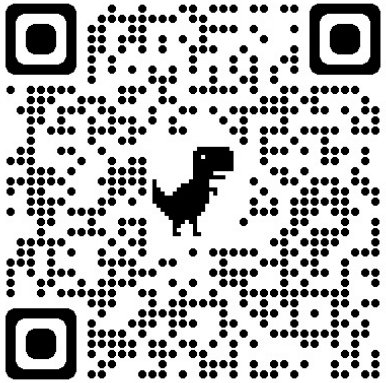


Cyber Resilience for Vehicles

V2C: A Trust-Based Vehicle to Cloud Anomaly Detection Framework for Automotive Systems

Thomas Rosenstatter, Tomas Olovsson, Magnus Almgren

Published @ *ACM ARES 2021* | *here*

V2C:A Trust-Based Vehicle to Cloud Anomaly Detection Framework for Automotive Systems

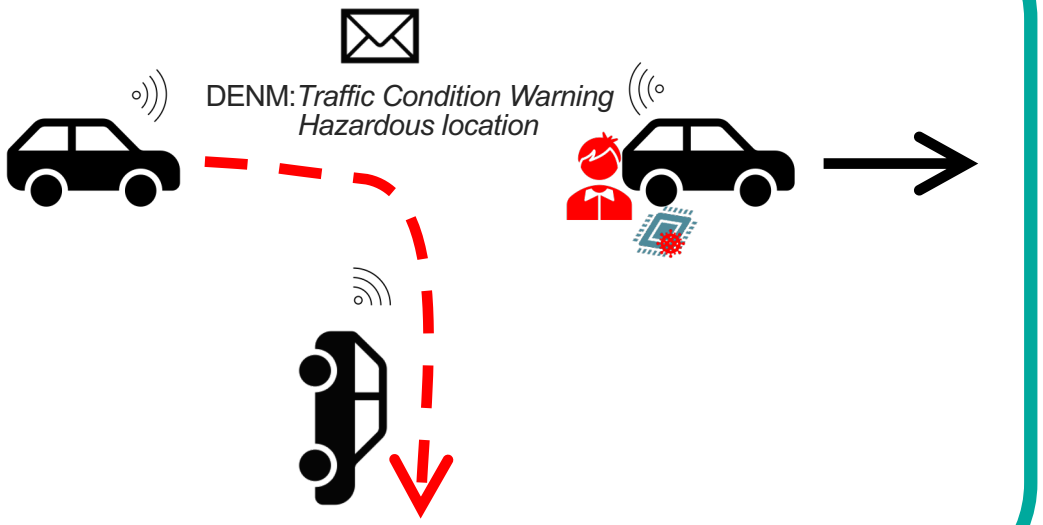
A framework for anomaly detection that combines individual peer evaluations of V2V interactions with analysis in the cloud.

1) Identify attack scenarios
2) Propose a framework consisting of four modules
3) Define the requirements for each module
4) Identify relevant techniques and approaches for each module
5) Perform individual assessments of each module in regards to required functionality or ability to detect the specified attack scenarios.
6) Discuss the framework based on a use case

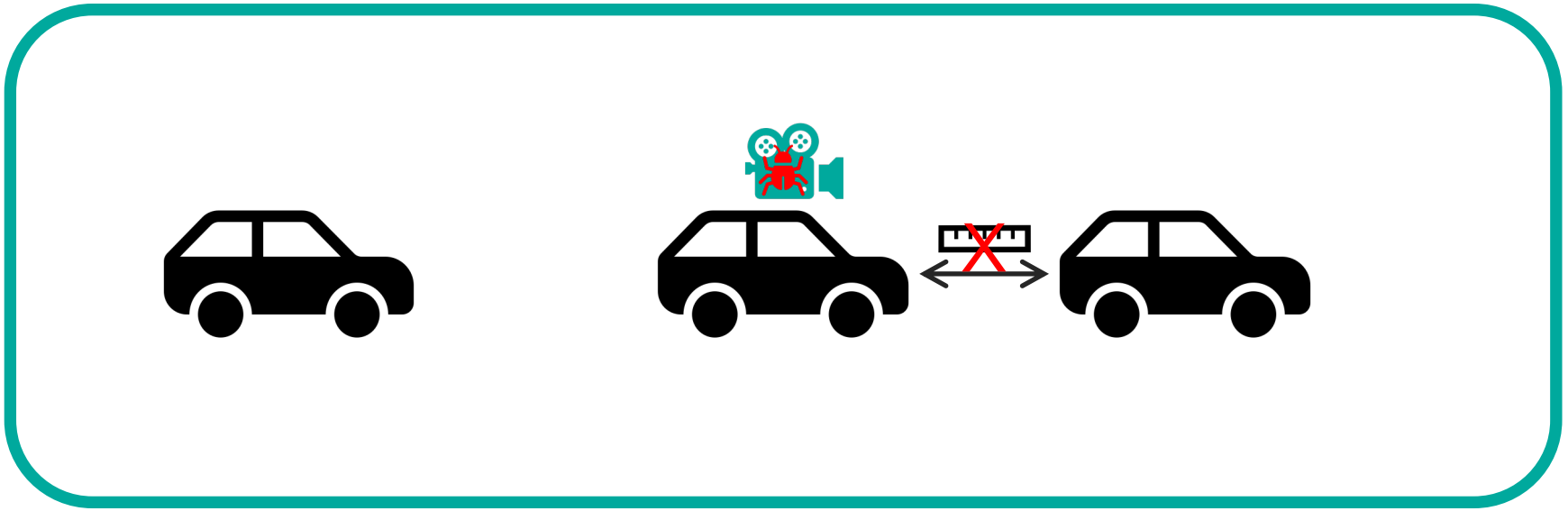
Attacks and Anomalies

We assume that attackers gained control of parts or the entire vehicle and therefore are **able to change the vehicle behavior** to achieve their goal.

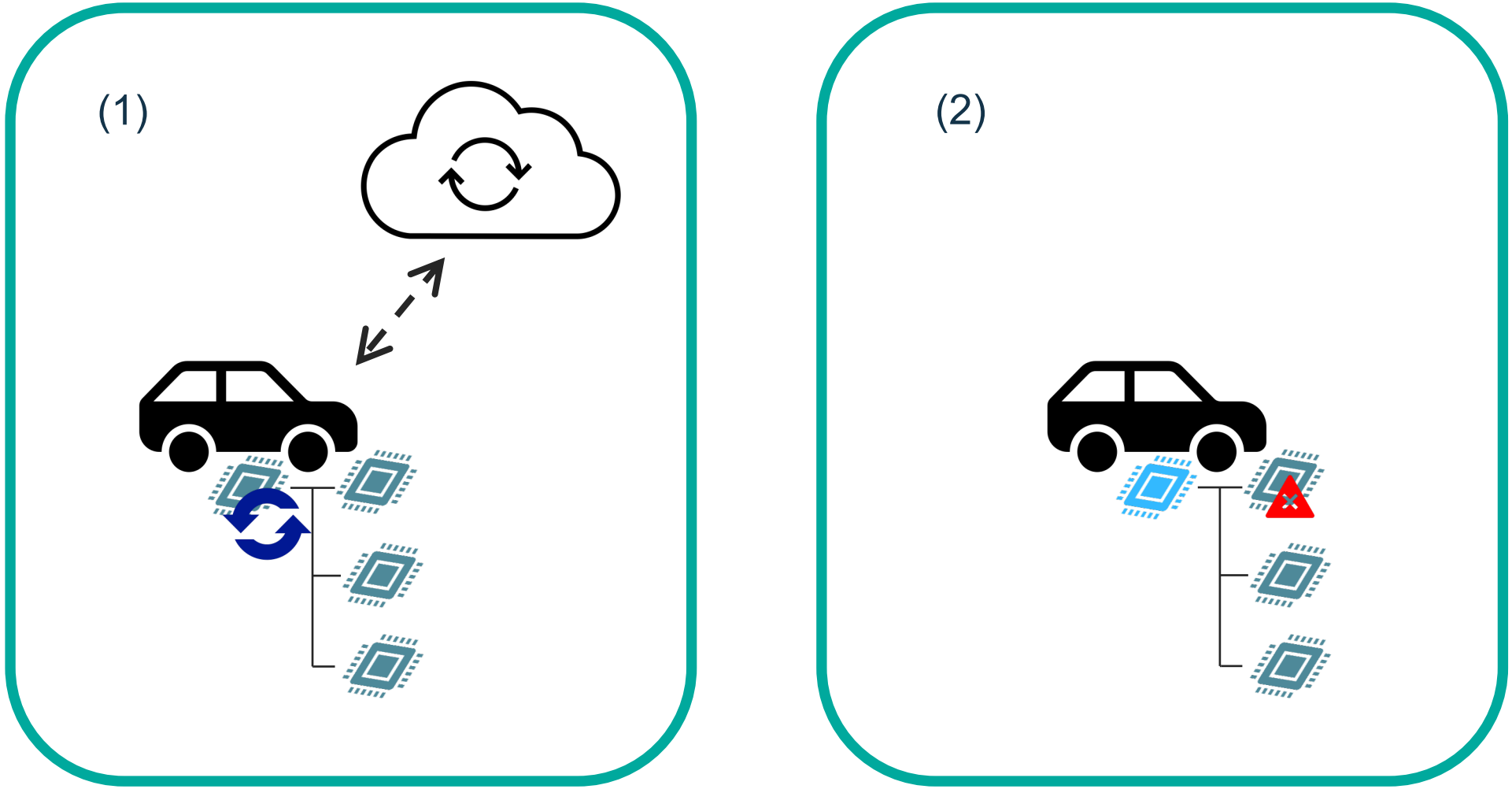
Unauthorized Firmware Manipulation
Attack Scenario 1



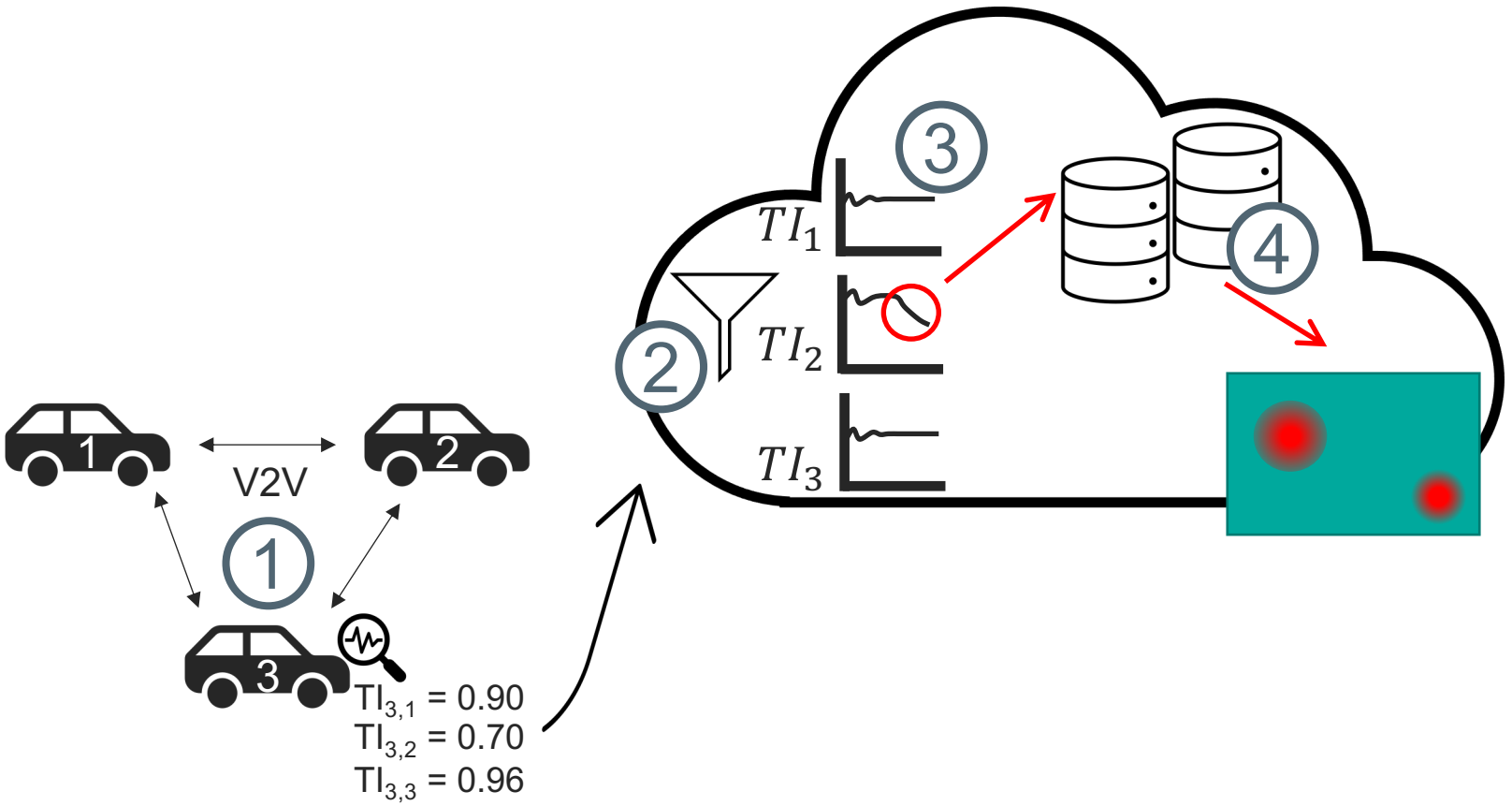
HW/SW failures
Attack Scenario 2



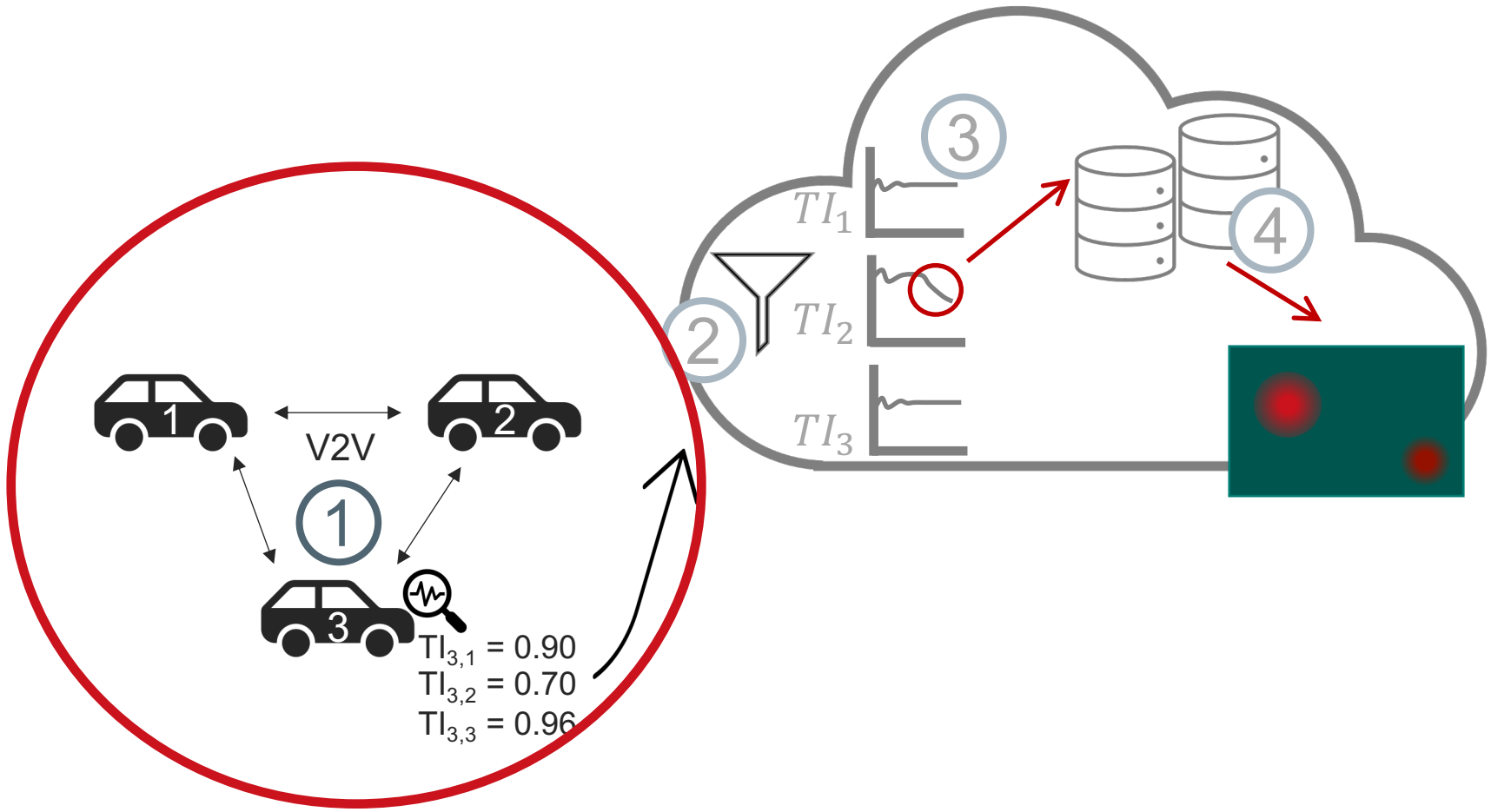
Legitimate SW/HW update
Attack Scenario 3



Framework Structure

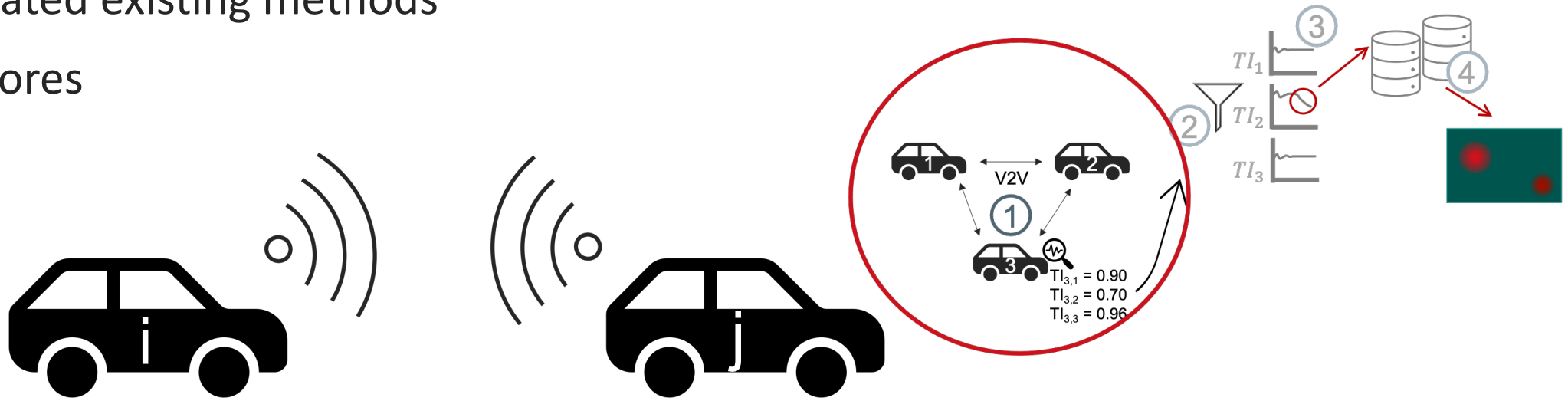


(1) Trust and Reputation Models



(1) Trust and Reputation Models

- Investigated existing methods
- Trust scores

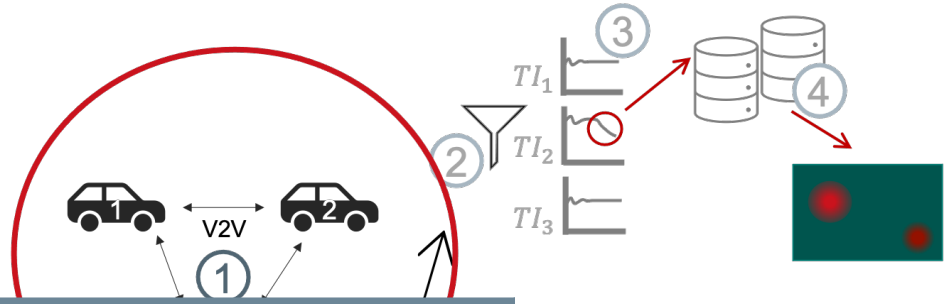


$TI_{i,i}$ … Trust score own vehicle

$TI_{i,j}$ … Trust score of vehicle j perceived by vehicle i

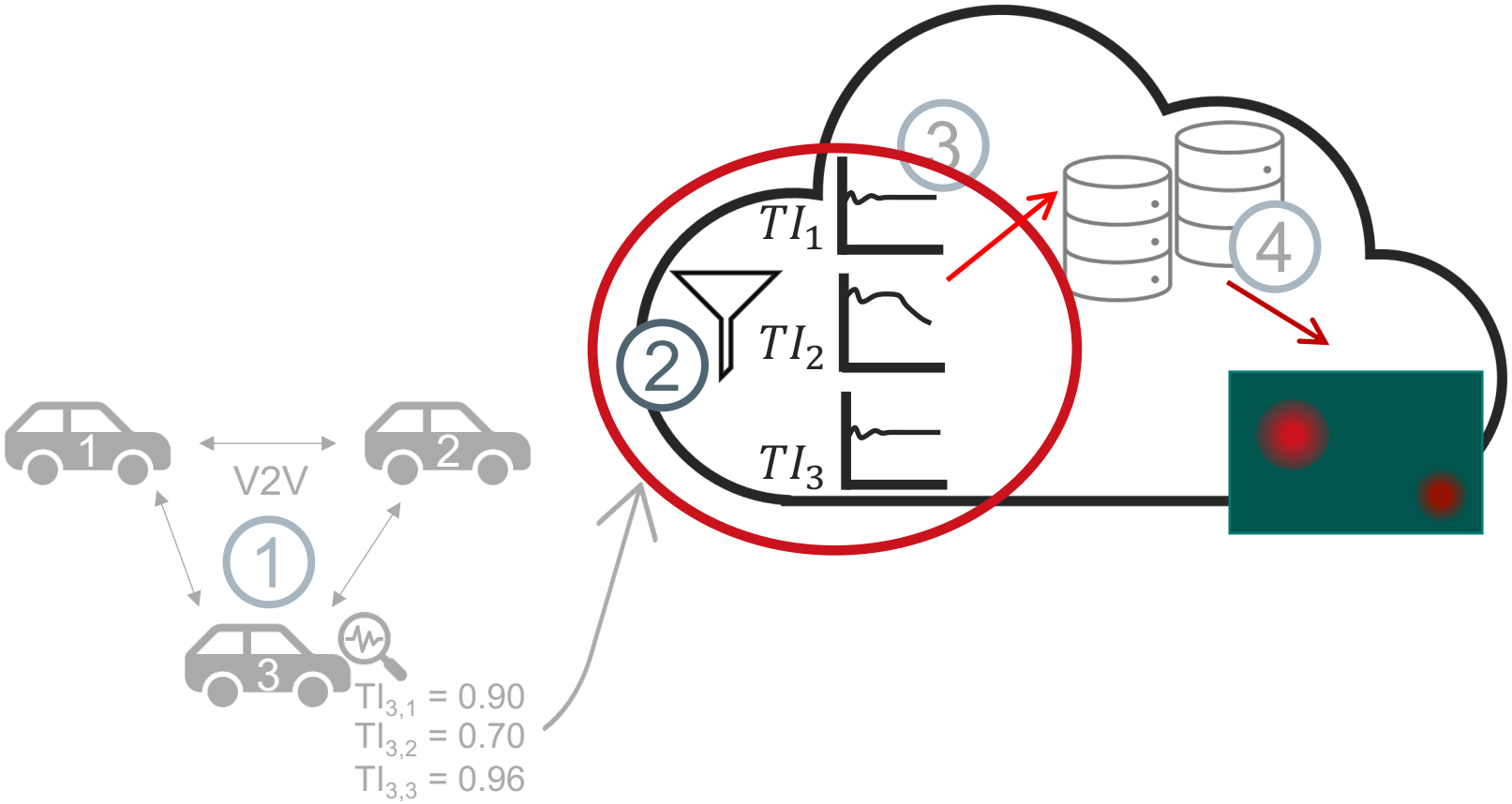
(1) Trust and Reputation Models

- Investigated existing methods
- Trust scores $TI_{i,i}$, $TI_{i,j}$
- Confidence $C_{i,j}$
- Reporting the the scores



Timestamp	TI	Confidence Value
2021-08-17	$TI_{1,1}$ = 0.9	0.9
2021-08-17	$TI_{2,1}$ = 0.8	0.8
2021-08-17	$TI_{3,1}$ = 0.75	0.8

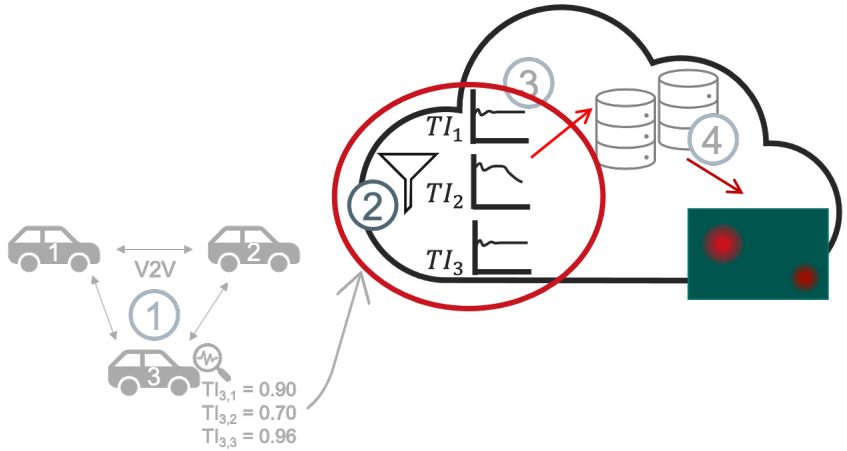
(2) Combining Trust Scores



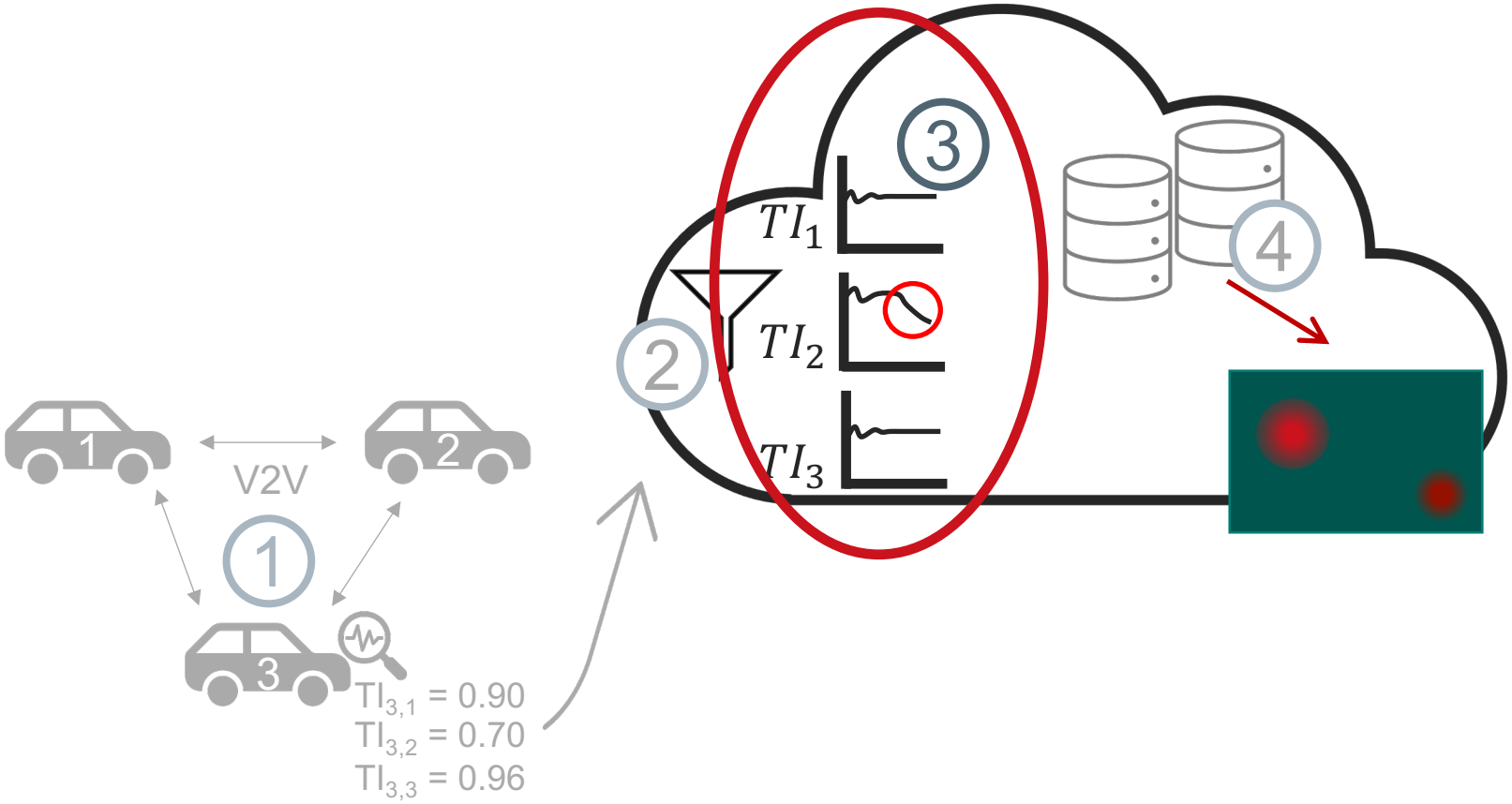
(2) Combining Trust Scores

- Means (arithmetic and geometric)
- Dempster-Shafer Theory (Rule of combination)

Timestamp	TI
2022-03-08	TI_1 = 0.78
2022-03-08	TI_2 = 0.9
2022-03-08	TI_3 = 0.9

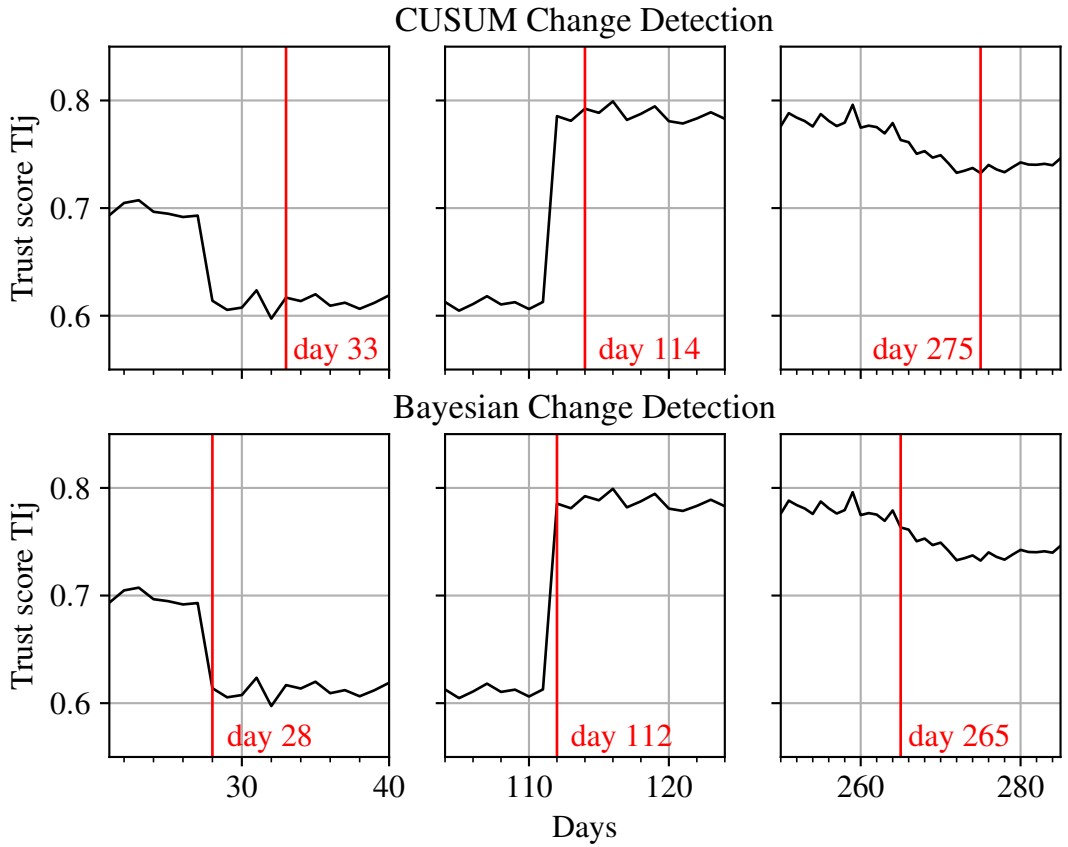


(3) Detecting Change in Behavior

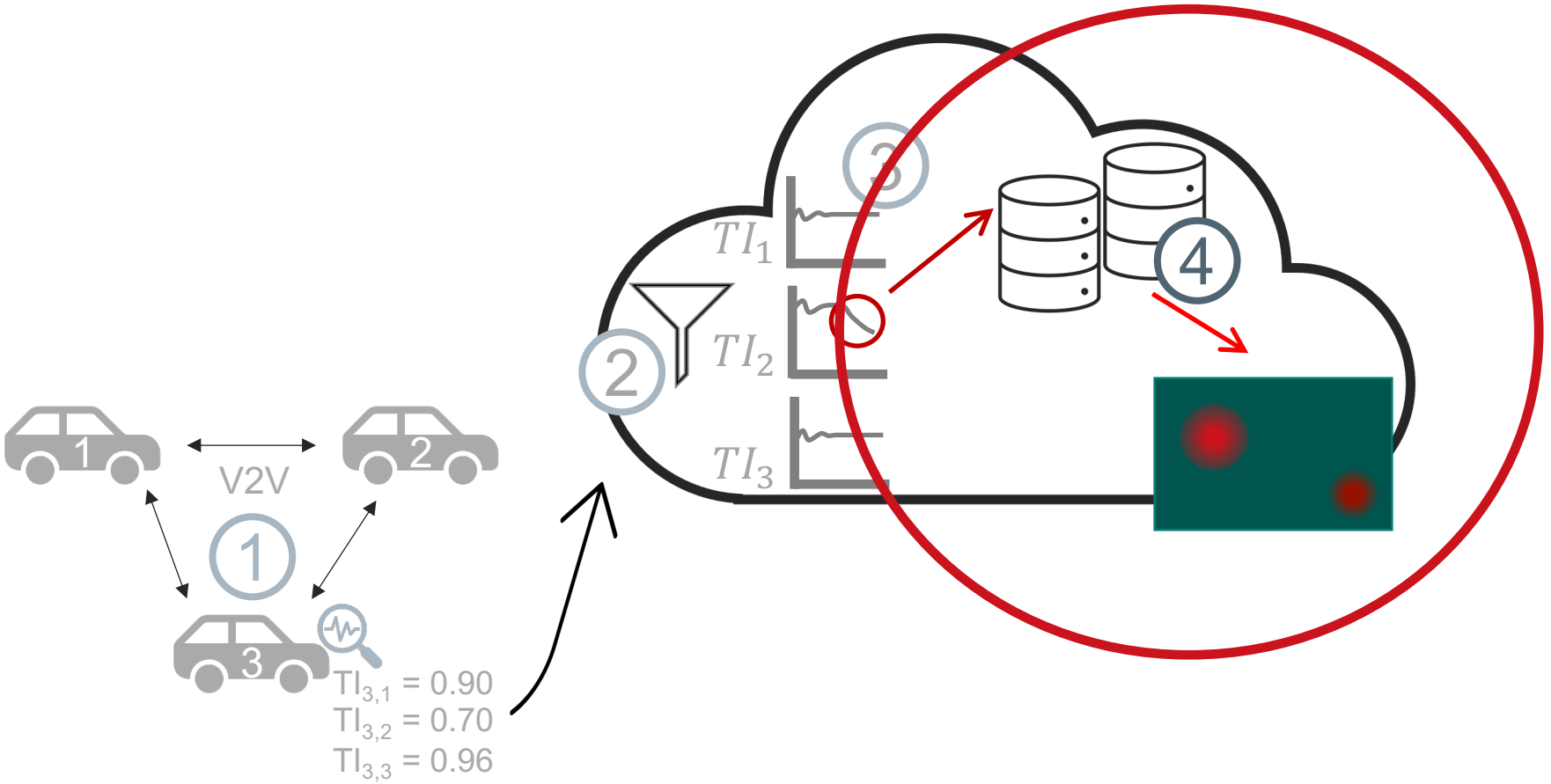


(3) Detecting Change in Behavior

- Detection of change in a one-dimensional time-series data
- Candidates (chosen based on review [1])
  - Cummulative Sum (CUSUM) [2, p.40],[3]
  - Bayesian Change Detection [4],[5]

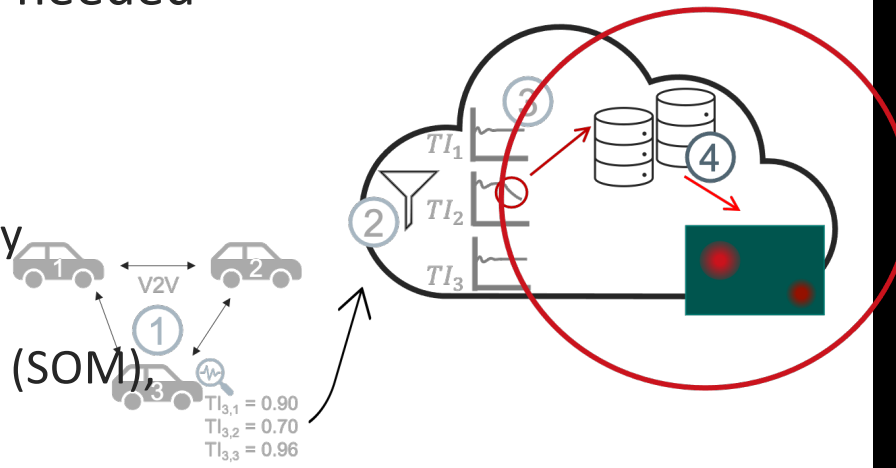


(4) Data Analysis in the Cloud



(4) Data Analysis in the Cloud

- Manual investigations based on available data in the cloud needed
- Anomaly-based detection
  - (1) Defined by specifications
  - (2) Following established processes e.g., KDD process to identify suitable techniques
  - Automation of detection techniques, e.g., self-organizing maps (SOM), isolation forest



V2C:A Trust-Based Vehicle to Cloud Anomaly Detection Framework for Automotive Systems

→ A framework for anomaly detection that combines individual peer evaluations of V2V interactions with analysis in the cloud.

1) A framework consisting of four modules
2) Define the requirements for each module
3) Identify relevant techniques and approaches for each module
4) Perform individual assessments of each module in regards to required functionality or ability to detect the specified attack scenarios.
5) Discuss the framework based on a use case

RI. SE
Thomas Rosenstatter, Researcher, RISE
thomas.rosenstatter@ri.se

References

- [1] Samaneh Aminikhanghahi and Diane J. Cook. 2017. A survey of methods for time series change point detection. Knowledge and Information Systems 51, 2 (2017), 339–367. https://doi.org/10.1007/s10115- 016- 0987- z
- [2] Michèle Basseville and Igor V. Nikiforov. 1993. Detection of abrupt changes: theory and application. Vol. 104. Prentice Hall, Englewood Cliffs, NJ.
- [3] Marcos Duarte. 2020. detecta: A Python module to detect events in data. https://github.com/demotu/detecta. visited on 2020-11-12.
- [4] Ryan P. Adams and David. J. C. MacKay. 2007. Bayesian Online Changepoint Detection. arXiv:0710.3742 [stat.ML]
- [5] Johannes Kulick. 2020. Bayesian Changepoint Detection – Python Implementation. https://github.com/hildensia/bayesian_changepoint_detection visited on 2020-11-12.