Technische Universität München
TUM School of Computation, Information and Technology

# A Holistic Methodology for Quality Assessment of Intrusion Detection Systems

Thomas Fabian Hutzelmann

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Florian Matthes

Prüfende der Dissertation:

1. Prof. Dr. Alexander Pretschner
2. Prof. Dr.-Ing. Felix Freiling

Die Dissertation wurde am 24.06.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 02.09.2024 angenommen.

*In the beginning of my PhD,*
*I was told that a thesis is*
*"documented learning".*
*Back in time I haven't believed it,*
*but now, here it is:*
*The documentation of*
*the craziest growth in my life.*

# Acknowledgments

Composing this thesis would not have been possible without the support of my colleagues, family and friends. I want to express my great gratitude to Professor Alexander Pretschner for the support and supervision of this thesis. Thank you for providing me with the space I needed and tolerating my stubbornness about crazy ideas. Moreover, my gratitude goes to Professor Felix Freiling, my second supervisor. Thank you for dedicating time to comprehend my research and supporting me during the final steps.

Besides, I am thankful to all humans who formed the chair for Software and Systems Engineering which hosted me during my research. Thank you to Severin, Mojdeh, Amjad, Saahil, Alei, Mohsen, Florian, Ehsan, Ana, Patrick, Hafeez, Tabea, Markus, Valentin, Daniel, Stephan, Jan, Claudius, Julia, Ilias, Raphael, Lena, Simon, Fabian, Nicola, Julia, Roland and David. My PhD experience would lack a critical part without you and I proudly call you friends. In particular, thank you Stefan whose door was always open for me. I entered it frequently and always left with calm relief. Also, thank you a lot Traudl, in my view you are the soul of the chair.

In addition, I want to state my gratitude for the opportunity to join the industry and apply my research in a real environment. Thank you to Sebastian, Bernhard, Peter, Daniel, Felix and Sirko. Your feedback and differing perspectives were a unique contribution to my work. Especially, I want to point out my reinforcement from Nadine. Thank you for investing the time to enable very lengthy and detailed meetings, listening carefully, and ultimately helping to bring structure into my chaos. Furthermore, I want to thank all the students who worked under my supervision: Thank you to Valentin, Tabea, Lukas, Malte, Dominik, Florian, Ayisha, Hardik, Jens, Vera, Peter and Feri. My thesis would not have been the same without the many hours we discussed and refined my and your research. Your work was a vital support to realize my ideas in code and real-world applications.

Aside from that, I want to express my gratefulness to my mom. Thank you for laying the foundation leading to this thesis and nurturing my curiosity from very early on. Furthermore, thank you Mathias. Your support for so many years on my way to and through this thesis means a lot and guided me toward the better. Last but maybe most importantly: Thank you Ana. I don't think I could exhaustively enumerate the countless reasons I am thankful to you. You have been part of so many important things, but I want to highlight this: Thank you for not giving up on me and making me not give up. I would not have endured all the lows and downs of the time of this thesis without you.

**Zusammenfassung**

Ein Intrusion Detection System (IDS) ist die letzte Verteidigungslinie gegen potenzielle Angreifer, die die Sicherheitseigenschaften des geschützten Systems verletzen. Nach dem derzeitigen Stand der Technik wird die Entscheidung für ein bestimmtes IDS anhand einer Bewertung getroffen, bei der ein Vergleich der Erkennungsrate und die Falsch-Positiv-Rate verschiedener IDS-Kandidaten im Mittelpunkt stehen. Um diese Raten zu ermitteln, werden alle IDS-Kandidaten vollständig implementiert, integriert und auf demselben großen Satz beschrifteter System-Traces ausgeführt, die sich aus dem Systemverhalten bei regulärem Betrieb und exemplarischem Angreiferverhalten zusammensetzen. Dieses Standardbewertungsschema ist jedoch mit hohen Kosten verbunden und birgt das Risiko schwerwiegender Flüchtigkeitsfehler. In dieser Dissertation erweitern wir dieses Schema um zwei zusätzliche Schritte: (1) einen frühen Vorfilter, der die Eignung einer breiteren Menge von IDS-Kandidaten ohne jegliche Implementierungen bewertet, und (2) eine abschließende, automatische Generierung von Verhaltensweisen, die spezifisch und besonders anspruchsvoll für die Klassifizierungen durch ein bestimmtes IDS sind. Für die frühe Eignungsanalyse vor der Anwendung des Standardbewertungsschemas schlagen wir vier verschiedene Vorfilter vor, die das Einhalten der Anforderungen von Domänenexperten im konkreten Anwendungsfall fördern. Wir haben die Vorfilter aus einem Vergleich der abstrakten Eigenschaften von 21 IDS-Kandidaten für denselben Anwendungsfall in einer einheitlichen Darstellung abgeleitet. Diese Darstellung basiert auf einer Taxonomie abstrakter Eigenschaften von IDSen, die wir durch eine Kombination aus 1) Interviews mit Industrieexperten und 2) einer umfassenden Studie der akademischen Literatur, die bestehende Übersichtsarbeiten über den beispielhaften Anwendungsfall zusammenführt, herausgearbeitet haben. Darüber hinaus schlagen wir eine abschließende Phase vor, um die anspruchsvollsten Traces für einen IDS-Kandidaten zu ermitteln, der nach dem Standardbewertungsschema ausgewählt wurde. Unsere Methodik generiert repräsentative Beispiele für sechs verschiedene Klassen von System- und IDS-Verhalten für eine abschließende Beurteilung durch Domänenexperten. Zu diesem Zweck verwenden wir die szenariobasierte Optimierung, um einen breiten Raum aller potenziellen Datensätze zur Bewertung von IDS für sicherheitskritische cyber-physische Systeme zu erkunden und zu aufzubereiten. Wir validieren unsere vorgeschlagene ganzheitliche Methodik in beiden Schritten an einem kontinuierlichen Anwendungsfall aus dem Automobilbereich. Genauer gesagt, führen wir 1) eine Fallstudie mit Industrieexperten durch, um unsere Vorfilter in der Praxis zu evaluieren, und 2) entdecken und dokumentieren sicherheitskritische Angriffe auf ein Open-Source-Fahrerassistenzsystem auf Stufe Zwei. Insgesamt finden wir Belege dafür, dass jede Bewertung der Qualität eines IDS relativ zum geschützten System und den betrachteten Angriffen ist. Mit anderen Worten: Die Eignung eines IDS hängt von verschiedenen Faktoren ab, die für den jeweiligen Anwendungsfall spezifisch sind, und eine umfassende Analyse erfordert eine sorgfältige Berücksichtigung der Systemeigenschaften und des Verhaltens der Angreifer vor und nach dem Standardbewertungsschema.

**Abstract**

An Intrusion Detection System (IDS) is the last line of defense against potential attackers violating the security properties of the protected system. In the state-of-the-art, the decision to deploy a specific IDS centers around an evaluation that compares the detection rate and false positive rate of various IDS candidates. To determine these rates, all IDS candidates are fully implemented, integrated and executed on the same large set of labeled system traces composed of system behavior at regular operation and exemplary attacker behavior. However, this standard evaluation schema comes at high costs and has a risk of critical oversights. With this thesis, we extend this schema with two additional steps: (1) An early prefilter to assess the suitability of a broader set of candidate IDSs without any implementations, and (2) a final, automatic generation of behavior that is specific and particularly challenging for the classifications of a given IDS. For the early suitability analysis before following the standard evaluation schema, we propose four distinct prefilters fostering compliance with the requirements of the domain experts in their concrete use case. We deduced the prefilters from a comparison of the abstract properties of 21 candidate IDSs for the same use case in a unified representation. This representation is based on a taxonomy of abstract properties of IDSs that we elicited through a combination of 1) industry expert interviews and 2) an extensive study of academic literature subsuming existing survey papers about the exemplary use case. Furthermore, we propose a concluding phase to pinpoint the most challenging traces for a candidate IDS that has been selected with the standard evaluation schema. Our methodology generates representative examples for six different classes of system and IDS behavior for a final assessment by domain experts. For this purpose, we use scenario-based optimization to explore and augment a broad space of all potential datasets to evaluate IDSs for safety-critical cyber-physical systems. We validate our proposed holistic methodology with both steps on a continuous use case from the automotive domain. Namely, we 1) conduct a case study with industry experts to evaluate our prefilters in practice, and 2) elicit and document safety-critical attacks on an open-source advanced driver assistance system on level two. Overall, we find evidence that any assessment of the quality of an IDS is relative to the protected system and the considered attacks. In other words, the suitability of an IDS depends on various factors unique to the use case, and a sophisticated analysis requires careful consideration of system properties and attacker behavior before and after the standard evaluation schema.

# Contents

# Part I

# Prelude

# 1 Introduction

*This chapter depicts and defines the scope of this thesis. It outlines gaps in the state-of-the-art, formulates research questions, and enumerates contributions.*

## 1.1 General Context

How can we ensure that the systems we engineer and operate are secure? After decades of research, all suggestions and approaches towards addressing this challenge remain fractional and cross-disciplinary [11]. Security engineering comprises actions during the design and the system's operation aiming to prevent, detect, and respond to potential attacks. Nevertheless, there are always multiple approaches that mitigate the same threat, and none of them alone provides perfect security. Hence, security remains a continuous trade-off between the overhead of the utilized security measures and the mitigated risks. Particularly to anticipate threats unknown at design time, intrusion detection is vital to ensure security properties during operation.

An Intrusion Detection System (IDS) is an additional component added to an operational system. The IDS monitors the system's operation during runtime and yields an alarm once it observes "suspicious behavior". The definition of suspicious behavior is a design decision of the engineer of the IDS. Possible definitions are signatures of the attacker's behavior, models of the system's normal behavior, or both. In addition, every alarm requires an adequate reaction as a response and mitigation to the ongoing attack. This reaction may vary from logging all suspicious events—to enable a human expert to understand better and fix the vulnerability used by the attacker—up to an automatic attack mitigation—which extends the IDS to an intrusion detection and prevention system. The above definitions of core terminology align with the recommendations of the National Institute of Standards and Technology [117].

The IDS's utility and potential financial benefit mainly depend on its actual detection capabilities and deterrence of attackers [27]. On the one side, an ongoing attack not triggering an alarm (a false negative) puts the security assets in the system at risk. Conversely, the IDS raising an alarm during regular system operation (a false positive) causes an unnecessary reaction. Therefore, the most crucial attribute of a "good" intrusion detection system is a high detection rate and a low false positive rate [130].

Nevertheless, the suitability and actual benefit of an IDS always remain domain- and use-case-specific. First, the IDS, as an additional component, requires more resources for development and maintenance. Furthermore, the constant monitoring of the protected system introduces extra overhead in runtime, computational power, and resource usage. Finally, the consequences of false negatives and positives strongly depend on the use case. For example, shutting down the system as a reaction to an alarm is costly. Hence, a few false alarms might exceed the actual damage from a successful attack.

### 1.1.1 Characteristics of a "Good" Intrusion Detection System

In more detail, the characteristics of a good IDS group into three aspects:

The first and most important aspect is the detection capabilities of the IDS. An ideal IDS detects all ongoing attacks while it does not raise a single false positive. However, on realistic system workloads, such behavior is improbable. In general, due to an effect called Base-Rate-Fallacy [14], an extremely low false positive rate is considered more important than detecting all attacks. Another characteristic of the detection capabilities is the response time, i.e., the delay from the first modification of an attacker inside the system until the IDS raises the alarm. To enable the prevention of the attack, a good IDS detects an attack immediately after the first modification and before the attacker can cause any damage inside the system.

The second aspect considers the additional overhead required to develop and operate the IDS within the system. Adding the IDS into the system increases the overall complexity and the development and maintenance costs. Furthermore, every event monitored from the system must be processed, which induces additional workload and higher resource consumption. Logging suspicious events implies more storage occupation and network capacities. Finally, domain experts might need to manually investigate and label the collected data before any impact on the system's security. Based on this newly gathered knowledge during operation, the IDS might require adjustments, i.e., retraining machine learning-based approaches or continuous optimization of the detector during the lifetime in general. In short, a good IDS minimizes the need for configuration or training and consumes negligible computational resources for the monitoring process.

The third and final aspect is soft constraints on the IDS inherited from the protected system. After adding the IDS into the system, the composition still must fulfill the same requirements. For example, this might become problematic from a legal perspective since the collected data for detection might contain confidential or private information. Further restrictions may apply in safety-critical scenarios, such as maintaining specific runtime guarantees. A good IDS complies with all the constraints given by the use case.

All these aspects and requirements contradict and cannot be fulfilled equally by one universal IDS. Moreover, the optimal trade-off incorporated in the best IDS depends on the domain and use case. For example, the deployment in cloud environments has loose constraints on the system overhead, whereas the deployment in an embedded system is constrained to strict limitations. Especially, safety-critical environments require the IDS to fully comply with the system specification while preventing potential attackers from violating security and safety properties [69].

Considering these limitations and the fact that security is a constant arms race between attackers and defenders, any IDS cannot be the optimal solution invariant of time or usage. Instead of building and deploying an IDS once, repeatedly identifying the best IDS among a set of candidates is essential [79]. This set of candidates consists of different detection approaches as well as different adjustments or configurations of the same approach. For a given use case, the set changes with the emergence of new detection approaches or advanced knowledge about the use case. Only the deployment of the best candidates from this diverse and varying set elicited regularly in a profound evaluation promises the

best detection behavior in the long run and, thereby, the highest benefit and protection. Consequently, a sound and affordable methodology for comparing and differentiating various IDS candidates to identify the best among them is vital.

### 1.1.2 The Intrusion Detection Evaluation Problem

Concerning this constant need for this evaluation, the current research focuses on refining methods for comparing two or more IDSs to identify the more suitable approach for a given use case. This challenge is known as the so-called "Intrusion Detection Evaluation Problem" [26]. Given a use case (with a particular security risk and specific expectation for "good" detection as introduced before) and a set of potential candidate IDSs, how can we choose the best detection approach for this use case among them and identify its optimal configuration?

As discussed before, a suitable detection algorithm is highly use-case-specific. Yet, software engineering aims for generic processes and approaches invariant to use cases that reliably result in economic software. Hence, researchers and software engineers aspire to establish standardized and well-investigated approaches to ensure a high quality of the process and the final product. Therefore, this problem demands a general methodology to assess the quality of all candidate IDSs and to ultimately build sufficient support for a sound decision if an IDS and which of the candidates is worthwhile for the security and adequacy of the entire system.

Despite the increasing number of suggested IDSs, there is no commonly agreed procedure or explicit guideline on evaluating and comparing a set of approaches. Over the years, scientific literature suggesting new detection approaches for different workloads and forms of inputs [83] has converged roughly to the same schema for their evaluation. Nevertheless, the refinement of this high-level schema in each publication is individual and often varies drastically. Hence, subjective choices and individual preferences often guide the evaluation, or researchers follow methodologies designed for other domains like statistics or machine learning [126].

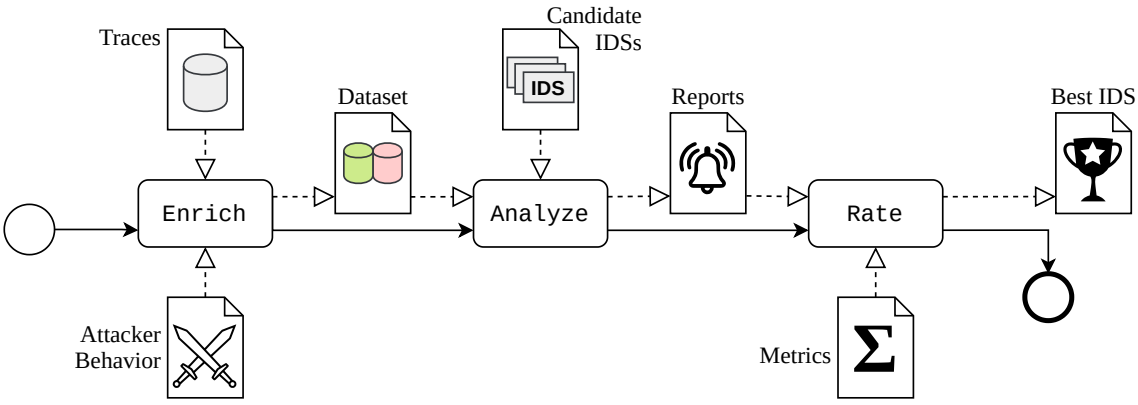


**Figure 1.1:** *Overview of the key steps of the standard process to evaluate IDSs.*

This standard evaluation schema for the evaluation of IDSs depicted in Figure 1.1 contains three major steps: (1) An initial *enrich* step creating a labeled dataset for further evaluation, (2) an *analyze* step where the implemented IDS prototypes investigate this dataset, and (3) a *rate* step that compares the performance shown by the IDSs in the analyses with respect to some chosen metrics. Thus, this process elicits the IDS performing best on the given dataset from the initial set of candidates.

The foundation for assessing the performance is a diverse and realistic dataset. The dataset must contain labeled attacks to measure true positives and false negatives during the evaluation. Unfortunately, such datasets are barely available, and their generation with sufficient quality to enable the reproduction of the results for different detection approaches requires high effort and careful preparations [39, 91, 92]. A common strategy to reduce this workload, especially when no exploit databases exist, is to start with benign recordings of the normal system operation and inject attack samples [83]. These samples of attacker behavior are either obtained through attack scripts or from a simulation of the attack manifestation in the features of the dataset. The later evaluation considers the resulting dataset enriched with labels of attacks and benign behavior as a ground truth.

In the next step, each IDS in the evaluation set processes the prepared dataset. Most prominently, the methodology requires records of all potential alarms raised through the IDS. Depending on the specific requirements of the use case, other observations may complement these reports, e.g., the detection delay or the computational resources required for detection. Overall, each IDS analyzes the same dataset and yields comparable reports on the same performance attributes.

In the final step, an analysis of these reports accesses the differences in the detection quality of the different IDS approaches. As a core, several metrics can be calculated based on the true and false positives and negatives [126]. These metrics enable ranking all candidate approaches and a deeper understanding of their configuration. The best-performing IDS in the evaluation that complies with an adequate trade-off between all requirements—if such an IDS exists among the candidates—is the recommended implementation for deployment into the system.

A better understanding of the evaluation methodology and the actual confidence in the obtained assessment of the candidates is the motivation of this thesis.

## 1.2 Thesis Statement

Through the issues we faced and investigated while following the standard evaluation schema, we elicited the following hypothesis guiding all research efforts in this thesis:

*The quality of an IDS cannot be determined absolutely and in isolation but only measured relative to three inextricable factors of a specific use case: (1) the requirements of the system and peculiarities of the system's behavior, (2) the precise definition of the attacker model, including attacker behavior and (3) the chosen model of the intrusion encoded in the detection algorithm.*

To the best of our knowledge, this view has not been raised in academic research before the work in this thesis. Moreover, it contradicts the current state-of-the-art practice for evaluating new intrusion detection systems.

## 1.3 Goal

The goal of this thesis is to propose a holistic methodology that solves instances of the intrusion detection evaluation problem. Our methodology extends the standard evaluation schema to equally consider the properties of the IDS of the system to be protected and of the attacker. We aim for an iterative approach that increases the number of considered candidate IDSs, reduces the overall costs for evaluation, and provides options to tailor the evaluation to use-case-specificity where necessary.

We base this ambition for improvement of the state-of-the-art on a different view of the evaluation process: While the standard evaluation schema considers all candidate IDSs at a singular point in the development process on an equal maturity level, we differentiate between different maturity levels. In the beginning of the search for a potential IDS protecting a system, the requirements might be fuzzy and the understanding of the available detection approaches might be vague. The evaluation of these candidates is shallow and without implementation mostly based on abstract, qualitative traits of the candidates. On the other extreme, there might only be a single candidate IDS left that was selected from various previous evaluations but still the system owners lack the final confidence needed for a deployment decision. The evaluation of such a final candidate requires a fine-grained approach to spot shortcomings in the design and exceed previous evaluation approaches. Respecting and exploiting these different degrees of maturity is our intent driving each step of this research and reflects in each problem we address.

Our overarching objective is to contribute to the evaluation of IDSs in general. However, the topic of intrusion detection spans countless domains and unites various differing approaches. Hence, it is not feasible to tackle this problem universally in a single thesis. Therefore, we decided to focus our main research on a single use case (presented in 3) and analyze the specific IDS evaluation in depth from the candidate elicitation to the final deployment. In the discussions of the approaches, we highlight aspects that will most likely generalize and guide future work toward validating other aspects that are more speculative based on the current research alone.

## 1.4 Problem Statements, Gaps, and Research Questions

The scope of this thesis aligns with specific aspects of the Intrusion Detection Evaluation Problem. In general, all research questions focus on understanding what is a good process for the quality assessment of IDSs. Each sub-part of this thesis contributes towards refining and extending the standard evaluation schema towards specific levels of precision and invested effort within the analyses. In the following, we discuss problems of the standard evaluation schema hampering this progression, spot corresponding gaps, and formulate research questions. Chapter 6 gives a detailed elicitation of these gaps.

**Problem 1:** *The standard evaluation schema requires an initial investment in working IDS implementations*. The evaluation process for the best IDS of a use case resides within a trade-off: On the one hand, it should cover a maximal number of diverse, potential candidate IDSs, but on the other hand, minimize the necessary resources for the evaluation. The standard evaluation schema requires complete and executable implementations of every candidate IDS, including adequate detection models. Even if the candidate is incompatible with the use case, the system developers can only notice this during or after fitting it into the system it should protect. Especially research prototypes from scientific publications might be unavailable or unusable before complete self-reimplementation. Investing in functional implementations prevents a quick and cheap assessment, resulting in a significant cost factor limiting the number of candidates under investigation.

**Gap 1a: No holistic catalog of abstract properties.** There is no holistic catalog of properties that classify and differentiate various candidates for the same use case. While such properties are listed and structured in academic literature, these works only describe the research landscape but do not use the properties to rate IDSs. The view of experts from the industry about relevant properties is neglected in these academic surveys.

**Gap 1b: No early assessment solely with properties known before implementation.** The standard evaluation schema neglects properties of the IDS beyond the detection. No systematic methodology compares approaches solely based on properties that can be determined without implementation and execution of the IDS. In particular, the standard evaluation schema does not verify the suitability of an approach before implementation and the complete analysis of the dataset.

**Research questions:**

RQ 1.1: What are key attributes to describe and contrast intrusion detection approaches abstractly independent of any implementation of the system or IDS?

RQ 1.2: How can these key attributes be utilized to assess the suitability of a candidate IDS for a given use case?

**Problem 2:** *The standard evaluation schema does not ensure suitable and challenging datasets*. The standard evaluation schema uses universal and standardized datasets to evaluate competing detection approaches. Domain experts of the protected system have collected these datasets by recording sample traces of the system's operation or its prototypes. In this manual process, it is not feasible to systematically consider all characteristics of the system and their interplay with the IDS. Furthermore, the edge cases in the system's behavior might not be the edge cases for the detection decision of each specific IDS under analysis, especially when not anticipating all IDSs in advance. Hence, the dataset only depicts critical behavior specific to the IDS by chance. Furthermore, the requirement of attack samples for the evaluation further complicates the composition of a good evaluation dataset. On the one hand, attacks provided by the developers of the IDS might be biased towards easy detectability and, thereby, good performance of their approaches but not stressing an active avoidance of the detection by realistic attackers. On the other

hand, domain experts might not know about the specialties of an IDS and miss critical attacks while composing the dataset. Even if the dataset contains all edge cases for the classification by all IDSs, these critical traces might get lost in the vast dataset size and scores that only accumulate the overall performance. Such datasets, therefore, inherit the risk of not discovering problematic edge cases before the deployment of the IDS, which complicates addressing them appropriately and potentially violates security properties.

**Gap 2a: No quality metric for individual traces in benchmark datasets.** Previous quality assessments of benchmark datasets for IDS evaluation have only investigated the properties of the dataset as a whole. There is no measure of quality for a singular trace within a dataset nor in isolation.

**Gap 2b: No tailoring of the evaluation to the evaluated IDS.** The standard evaluation schema only reflects general domain knowledge about the system and attacker in the dataset and accumulates the performance of the IDSs in numeric scores. In particular, datasets do not intentionally contain system behavior that particularly matches or confuses the specific detection mechanism under evaluation. Inspecting specific system or attacker behavior that is peculiar solely for the classification by the IDS under analysis is vital for a valid evaluation, however, such behavior remains hidden in previous evaluations.

**Research questions:**

RQ 2.1: How can the suitability of individual traces within a dataset be quantified?

RQ 2.2: How can particularly suitable traces be included reliably and efficiently while composing datasets?

## 1.5 Solution

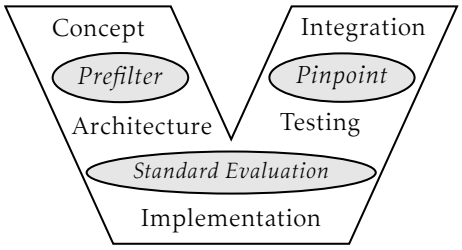


**Figure 1.2:** *Location of the three phases in the V-model*

This thesis aims to extend the standard evaluation schema towards an iterative workflow, optimizing the efficiency and costs to solve an instance of the Intrusion Detection Evaluation Problem. Our evaluation workflow ideally occurs in parallel to the development of the system to be protected and concludes in a deployment-ready IDS. Aligned to the software engineering process, this workflow consists of three phases, embedding the standard evaluation schema as the second step in the middle of the two new phases we propose. Figure 1.2 depicts an exemplary integration of these three phases of our proposed methodology into the V-model.

After the initial system design, the first phase of our quality assessment, the prefilter, focuses on spotting promising ideas for intrusion detection. This phase covers a broad range of concepts and ideas and does not require implementing the system or the IDS. In short, it identifies detection approaches that align with the system's conceptual design and cover attacks prioritized during a prior threat analysis. IDS candidates passing this filter are promising enough to invest further resources in a deeper analysis. Thus, this phase solves problem 1.

Then, the second evaluation phase follows the standard evaluation schema. The approaches are implemented and configured, analyze an invariant dataset, and the evaluation selects the best candidates according to selected performance measures like false positive rate or detection rate. Overall, this phase results in selected IDS prototypes suitable for being ported and integrated into the deployment-ready system.

Finally, the third phase, the pinpoint of datasets, focuses on validating the utility and raised security level of the previously elicited IDS. The core of this phase is structured deduction and automatic optimization of traces extending behavior from the system and penetration testing. The outcome of this phase is a small set of new traces specifically showcasing the strengths and weaknesses of the IDS in the current system. This, this phase solves problem 2. Domain experts can investigate this set with two possible outcomes: higher confidence in the IDS or directions to revisit previous workflow phases.

All three phases combined constitute a sound and systematic quality assessment and a holistic workflow that step-wise refines a broad set of intrusion detection approaches into—if any candidate qualifies as such—one deployed and valuable IDS.

## 1.6 Contributions

In summary, this thesis makes the following contributions towards the previously identified gaps in addressing the intrusion detection evaluation problem and with the current state-of-the-art in the quality assessment of IDSs.

1. **A holistic taxonomy for abstract properties of IDSs, the protected systems, and potential attackers.** ***(Addresses gap 1a)*** Abstract properties can describe the properties and suitability of an IDS for a given use case and specific attackers. Academic publications use such descriptions to advocate for newly proposed IDSs, and industry experts use them to match products to their requirements. We elicited and structured their utilized properties into a holistic and extensible taxonomy, relying on two sources: (1) opinions collected from interviews with industry experts from four different departments of an automotive manufacturer, and (2) a literature study on surveys summarizing and opposing existing work on the same use case. This taxonomy is a requirement and foundation for the following, second contribution.
2. **Four strategies to identify promising IDS candidates solely by abstract properties** ***(Addresses gap 1b)*** On the foundation of our taxonomy of abstract properties, we composed a unified and formalized description of 21 candidate IDSs for the same use case. From the analysis of the resulting dataset, we deduced four prefilters that match this structured information with the requirements of the domain experts, namely the design of their protected system, identified risks and anticipated attackers, and desired IDS properties. We demonstrate in a case study with the industry experts these prefilters spot promising IDSs with relatively low effort.

3. **Six fitness functions describing the suitability of traces in a dataset for evaluating a given IDS** ***(Addresses gap 2a)*** The scoring in the standard evaluation schema relies on counting and accumulating true positives, false positives, true negatives, and false negatives. We propose fitness functions that quantify the suitability of a trace concerning these four classifications. Each of our four fitness functions follows one of these classification types, and two fitness functions provide a baseline without considering the IDS. To quantify and compare the quality of traces with these functions is necessary for the following, fourth contribution.
4. **A methodology to systematically deduce traces showing edge case behavior in the IDS classification** ***(Addresses gap 2b)*** We embed our fitness functions as the core elements into a methodology known in scenario-based testing that we transfer to this new purpose. Our methodology systematically spans the space of all possible system traces and deduces in three steps a set of six optimized traces that particularly showcase different types of edge case behavior of the IDS. Domain experts can access this set for a final decision about the deployment of the IDS.
5. **Various insights about the state-of-the-art of intrusion detection in our exemplary use case.** All analyses and experiments conducted in this thesis, including the penetration testing revealing actual vulnerabilities, were conducted on the same use case: Securing an open-source [33] level 2 advanced driver assistance system [113] against attacks via the Controller Area Network (CAN) bus [29, 63, 85]. Hence, this thesis composes an in-detail analysis of several IDSs under real-world conditions substantiated with experiments in a real car. We spot several shortcomings in the current implementation of the IDSs and challenges for future research aiming to secure autonomous driving through intrusion detection.

During the research resulting in this thesis, we have released parts of the above contributions in peer-reviewed publications. We mention such publications in the short descriptions at the beginning of each chapter. Due to the obvious content overlapping, we do not explicitly mark quotes from such publications within the text.

[53] **Hutzelmann, Thomas**; Mauksch, Dominik; Petrovska, Ana; Pretschner, Alexander: *Generation of Tailored and Confined Datasets for IDS Evaluation in Cyber-Physical Systems*. IEEE Transactions on Dependable and Secure Computing, 2023

[54] **Hutzelmann, Thomas**; Mauksch, Dominik; Pretschner, Alexander: *How to Conduct Experiments with a Real Car? Experiences and Practical Guidelines*. Communications in Computer and Information Science. Springer International Publishing, 2020

[52] **Hutzelmann, Thomas**; Banescu, Sebastian; Pretschner, Alexander: *A Comprehensive Attack and Defense Model for the Automotive Domain*. SAE International Journal of Transportation Cybersecurity and Privacy 2 (1), 2019,

## 1.7 Summary of the Results

The results achieved through the research in this thesis span two significant aspects: (1) various findings related to our core hypothesis and (2) insights about the state-of-the-art about intrusion detection in our selected use case.

### 1.7.1 Support for our Thesis Statement

Through utilizing our methodology for eliciting a suitable IDS for our selected use case, we found regarding our hypothesis that:

- Industry experts and academic researchers cover aspects of all three factors (IDS, system properties, and attacker behavior) extensively when they discuss IDSs for our use case. However, the focus and granularity of details in each aspect differ in each publication and do not follow a unified structure. In addition, we could not identify a singular dominant structure or order of these aspects. Nevertheless, broad coverage of all these details is critical to match the use case and the IDS. Furthermore, the best differentiation for our four prefilters is a combination of properties from all three factors that are specific to the concrete use case. This observation suggests equal importance of all three factors to assess the suitability of an IDS. See Section 4.7.1 for more details and support for this observation.
- The whole space of traces for benchmark datasets is too broad and diverse to be covered in static datasets. The traces deduced through optimization were not part of the static dataset used for the initial evaluations. This lack indicates that the original evaluation datasets consider and check critical behavior only by chance. Furthermore, changes in the protected system, the attacker's behavior, or the configuration of the IDS result in differing edge cases. This divergence hints that universal datasets are unsuitable for showcasing all strengths and weaknesses, and a sound evaluation requires tailoring each evaluation to the analyzed IDS, namely to the interferences of the system specificities, attacker capabilities, and the concrete configuration of the detection approach. See Section 5.7.1 for more details and support for this observation.

All these results from our research support our hypothesis and emphasize the importance of equally considering the properties of the protected system, the IDS, and the attacker for a sound evaluation. Similar to other researchers before [123], we observed reluctance by the industry to adopt academic intrusion detection prototypes despite their high performance documented in previous publications. We hold the missing alignment with the system development process and the specificities of use cases in the standard evaluation schema documented in this thesis and our results as a reasonable explanation. Hence, we are hopeful that our contributions and the methodology proposed in this thesis oppose these long-term struggles in fully utilizing the achievements of IDS research against real-world security threats.

### 1.7.2 Observations about our Exemplary Use Case

During the application of our methodology and the elicitation process of the best IDS among various state-of-the-art approaches for our exemplary use case—the Controller Area Network (CAN) [110] bus used in the automotive domain—we found that:

- Academic literature provides over 248 publications presenting approaches for intrusion detection on the CAN bus and over 22 survey papers about them. It is not feasible in an industry context to implement all these approaches for evaluation. Our prefilters efficiently reduce this landscape to a set of one to five promising candidates for further evaluation. See Section 4.3.3 and Section 4.6 for details.
- The implementation of IDSs presented in academic literature requires more information than provided within the publication. Especially, when the characteristics of the protected system slightly varied from the system they used for evaluation, we struggled to achieve similar performance as denoted in the publications. These variations seem unavoidable to us, as different car manufacturers use individual architectures and implementations. Furthermore, the hardware we needed to collect the required data for training and conduct the evaluation inhibts further individual properties. This divergence is a continuing threat to validity in our experiments documented in this thesis and within other related research projects. See Section 5.5 and Section 5.7.4 for details.
- We applied automatic optimization of a spanned dataset space for evaluating IDS protecting an open-source driving assistance system on level 2 [33]. With the deduced traces, we successfully circumvented a state-of-the-art IDS [131], causing a severe safety violation without detection. This particular IDS has passed the standard evaluation schema without raising suspicion of the high risk through the attacks within the considered attacker model, which we have documented now. See Section 5.6.2 for details and the full experimental analysis.

All these findings on the exemplary use case showcase the utility of our methodology and contribute towards developing more secure autonomous cars.

## 1.8 Structure and Outline

The rest of this thesis is structured as follows: Part I gives an introduction to the scope and contributions of this thesis, and provides background information about the standard evaluation schema and the continuous use case of this thesis. Part II is split into two chapters that align with the two gaps our work addresses: Chapter 4 proposes four prefilters based on abstract properties to assess the suitability of many candidate IDSs before their implementation. Chapter 5 proposes a methodology to tailor and confine benchmark datasets to the evaluation of a specific IDS. Part III discusses related work to our contributions and summarizes our findings and conclusions. Finally, Part IV lists the tables and figures throughout the thesis and contains the bibliography and the appendix.

# 2 Background

*This chapter contains foundational knowledge related to the intrusion detection evaluation problem. Namely, the subsections select different aspects of the standard evaluation schema and elaborate them in a broader context. This background information forms the framing on which we have built our contribution.*

## 2.1 Knowledge about Attacks and Attack Samples

The main idea of IDS evaluation is feeding the investigated candidate IDSs with the input data from the protected system. To achieve full coverage of the IDS functionality, namely ignoring regular system operation and raising alarms during attacks, this data needs to contain both benign and malicious data. The first step of the standard evaluation schema (see Section 1.1.2 for details) focuses on the collection of adequate traces for the evaluation. While data from benign system operations is easily available, the data about the attacker's behavior needs special considerations and preparations for a sound evaluation.

Please note, that although it is possible to develop an IDS without attack samples, is not possible to skip the malicious data in the evaluation. For example, an anomaly-based IDS builds its internal model through learning patterns from the regular system operation and yields an alarm when it observes behavior that does not comply with this model. Hence, the development does not depend on malicious samples and these IDS aim to detect unknown and even unanticipated attacks. However, the verification of these detection properties in evaluation requires samples from these unanticipated attacks. While it is not feasible to anticipate or cover all potential attacks, the benchmark dataset requires representatives of the main threats to the protected system. With an insufficient diversity and realism of the malicious data, the resulting evaluation will be partial and its validity is threatened.

Malicious data (similar to benign data) can be collected in various ways [83]: In the best case for evaluation, the attack is available in executable form. This executable form can be specifically built for the evaluation with large manual efforts or reused from an existing exploit database. If such exploits are not available, it is an alternative to intentionally inject vulnerabilities into the system and benchmark the IDS with corresponding exploits. Finally, the necessary traces of attacks can also explicitly be generated, for example from testbed environments or honeypot systems. Although such traces are more easily available, they might lack the realism needed for a valid evaluation.

When merging benign and malicious traces in the evaluation, there arises another issue from the diverging nature of malicious and benign data: the implications of the Base-Rate-Fallacy [14]. Oversimplified, a real-world system is most likely attacked very rarely, especially if seen relative to the long operation time with a large user base without any malicious intent. In other words, when picking any trace from recorded real-world system behavior, there is a high imbalance between the tiny likelihood of an ongoing attack

and the dominant likelihood of regular system behavior. This has severe implications for the requirements for an IDS and consequently for the evaluation methodology. We want to illustrate this effect with an example from our use case, intrusion detection for an autonomous car. Assuming that a car is operated for 100000 kilometers in its lifetime and a repair shop takes care of 100 cars. Such a fleet of cars accumulates 10 million kilometers in their lifetime. Now, assume the deployment of a very good IDS that successfully detects all attacks and a likelihood of 0.01% percent to raise a false alarm in a hundred kilometers of regular driving. This would result in the repair shop seeing 10 false alarms for this fleet in their lifetime. However, until today, there is no publically documented case of a single vehicle that had crashed because of a cyber attack. Hence, not a single repair shop similar to our example would most likely ever see any actual attack among a constantly growing pool of falsely investigated cars. In case the hypothetical IDS does not successfully detect all attacks, this ratio is even worse. None such IDS would be suitable for a car manufacturer as they need even lower likelihoods of false alarms. Accordingly, there is a strong necessity for a very low rate of false alarms in any IDS used in relatively benign environments. For the evaluation of an IDS, this requirement caused by the Base-Rate-Fallacy has two implications: 1) To accurately measure such a low false-positive rate, the dataset requires to contain an enormous amount of benign traces with the IDS only wrongly classifying very few as an attack. Moreover, such a dataset cannot realistically contain as many attack samples as benign traces. While classical metrics like the F-Measure can comprehend imbalanced datasets, they assume that both types of misclassification are equally important. As we elaborated with the example above, this assumption is wrong for IDSs when the sum of all false alarms is more costly than missed attacks. Thus, 2) the IDS evaluation requires specific metrics that consider the effects of an imbalanced dataset in combination with imbalanced costs for false positives and negatives (see Section 2.4).

## 2.2 IDS Configuration and Customization

Although the standard evaluation schema considers all candidate IDSs as indifferent black boxes, it is important to differentiate several degrees of similarities between the candidates and reflect these in the evaluation. While it is in theory possible to build an IDS without any configuration option as one single, universal executable, modern IDSs provide various options and means to configure and thereby tailor them to the protected systems and individual security needs. These options have an impact on the development process and consequently on the application of the standard evaluation schema.

Rule-based IDSs provide the simplest means for configuration and customization. Their detection is based on a set of rules that describe concrete conditions when an alarm is raised. Such IDSs can be customized by enabling or disabling rules and shaping the ruleset according to specific needs and performance observed in a specific system. Often single rules can contain parameters and thresholds that permit further configuration. Different rulesets result into different configurations of the same detection approach.

Complementary to these rules forming a domain-specific model, other detection approaches rely on larger and more complex models with a generic structure. These univer-

sal models are transferred into the security domain from the field of artificial intelligence and the study of machine learning algorithms. While these models are more powerful and require minor to no interaction with domain experts, they require training with a large amount of data to encode the full spectrum of behavior into the thousands of weights and values of their internal model. Such a model is barely comprehensible for human domain experts which impedes a direct manual configuration and tuning. A different and better performance of the IDS requires a different and better training dataset. This need for large amounts of good training data led to several modifications and extensions of the standard evaluation schema (see Section 2.5 for details). Each change in the training dataset potentially results in a different learned model and consequently in another configuration of the same detection approach.

For the most sophisticated detection, a single IDS can internally combine multiple detectors to assess the observed system behavior for different attack types or wide ranges of normal behavior. In addition to the configurations of each detection individually, the concrete way of combination provides additional parameters for customization. For example, this combination can utilize a simple decision tree [46] or more complex evidence theory [132]. A sound combination of detections provides a lower false positive rate and a higher detection rate than each individual classifier in isolation. Consequently, each variation of individual models and their combination forms other configurations of the same generic detection approaches.

Finally, independent of the concrete model, the decision of when to raise an alarm depends on a threshold. The assessment of the observed behavior by the IDS's detection model is in most of the cases internally represented as a suspiciousness score. This score is a numeric value, however, the decision about a raised alarm is binary. Hence, the threshold is an upper boundary for this value denoting the minimal suspiciousness score after which an alarm is raised. Consequently, finetuning the threshold provides direct control of the balance and trade-off between the detection rate and the false positive rate.

## 2.3 Benchmark Datasets

Aside from properly configured IDSs, the most important part of the evaluation is the benchmark dataset. In the second step of the standard evaluation schema (see Section 1.1.2 for details), all candidate IDSs analyze all traces in the dataset while all yielded alarms are recorded. These recordings are the main input for the metrics and ranking calculated in the third evaluation phase. Therefore, the quality of the dataset has a critical impact on the validity of the evaluation results.

On the one hand, the benign traces in the benchmark dataset need to be an accurate representation of all possible system behavior. If they are too simplified, an IDS deployed in the system after a successful evaluation will result in too many false alarms. If they are unrealistically complex and diverse, an in-theory fully suitable IDS might fail in the evaluation. On the other hand, the attack samples in the benchmark dataset need to be an accurate representation of the full spectrum of potential attacks. If they are too monotonic, the deployment of an IDS after a successful evaluation inhibits the risk

of false trust in a successful defense that actually can be circumvented easily. If they only represent very sophisticated attacks, an IDS, that is actually defeating many attacks, might fail in the evaluation. The expectation of a candidate IDS to successfully detect all attacks is unrealistic. In short, the validity of the measured metrics and all corresponding deployment decisions have a critical dependency on the quality of the used benchmark dataset. The composition of a good benchmark dataset complying with all the above requirements is a challenging and expensive task. The strategies towards benchmark datasets differ in industry and academia. Hence, we want to contrast both situations with each other in the following.

### 2.3.1 Benchmark Datasets in Industry

Industry experts have large access to recordings of benign behavior during system development and maintenance. For example, these can be recordings during (automatic) test execution or samples recorded from development prototypes. Nevertheless, this large availability also has negative effects, as it is expensive and therefore not feasible to include all available traces in the benchmark dataset. The selection of diverse and representative traces for a minimal set of regular system behavior is the biggest problem here.

As elaborated in Section 2.1 the addition of attacker behavior into the dataset requires more effort and consideration. For industry experts, this task aligns with the penetration testing of their systems without an activated IDS. The penetration provides a structured frame for the data recordings and full documentation of the attacks and their criticality. While this offers the deepest understanding of attacks and a large diversity of samples when executed carefully, traditionally penetration testing happens at the end of the system development. Consequently, the IDS built and evaluated with this data is becoming one of the last components added to the system.

Industrial datasets are rarely published or shared with other researchers. While this is understandable due to the fact of the high specificity of the protected system and confidentiality of the known attacks, this also influences the quality of the benchmark dataset. The dataset is not investigated independently on biases and flaws. Furthermore, maintaining and adjusting the dataset on updates in the system or newly discovered attacks requires continuous resources and investments during the operation of the system. These changes on the dataset, especially if they are conducted by different experts, might further endanger the balance in the dataset.

### 2.3.2 Academic Benchmark Datasets

Academic researchers are confronted with the opposite situation and problems. It is feasible to conduct penetration testing on selected components for a small team. This process produces the necessary attack data for the benchmark dataset, especially if their proposed or investigated IDS is directed toward these specific attacks. Known attacks are documented in various publications and provide enough details to repeat the attack. However, the collection of benign system behavior is more problematic during the academic composition of good benchmark datasets. When they are not the developer of the system or

collaborate with the industry, academic researchers do not have access to test case specifications or automatic test suites. Hence, researchers tend to use record data based on their intuition and only a single or very few instances of the protected system.

As it is part of good scientific practice to share the used data and implementation, the publication of a good benchmark dataset with the methodology and considerations for its creation is appreciated as a value contribution worth a distinct publication. Consequently, there have been many domains and fields of applications with a stream of publications focusing on the creation of good datasets. While this enables a discussion about the datasets and independent investigations, it also documents the lack of these investigations in less active fields of research. This results in differing maturity levels of the published benchmark dataset and questions the usage of some of them.

In the following, we want to highlight two separate efforts to create benchmark datasets for the evaluation of IDSs: A general use case of TCP-network traffic among servers and a specific use case of traffic on the CAN bus within a car. The TCP datasets are interesting because they reassemble the largest effort to create a universal reference dataset for the development and benchmark of IDSs. The CAN bus datasets are interesting because they show an earlier stage of a less mature research area of IDS and show differences. Furthermore, CAN bus IDSs are the subject of the continuous case study of this thesis introduced in Chapter 3.

There are obvious factors that require a regular update of a benchmark dataset, e.g. the emergence of new attacks or major changes in the protected system. However, as the case of TCP-network data sets showcases, these factors are not the main reason for the creation of new datasets. Instead, we observe this repeating pattern: Once a dataset is published, other researchers focus on a deeper analysis of this data and spot shortcomings. This results in the collection of a new dataset avoiding these flaws and this loop repeats a few years later. In the case of TCP-network traffic, this has happened in three iterations with the original DARPA dataset [122] as discussed by Ring et al. [109]. Namely, the datasets CICIDS-2017 [120], CIDDS-001 [108], UGR'16 [75], and UNSW-NB15 [87] are all successors of the original DARPA dataset. The existence of so many variations and improvements of the same initial dataset underlines the need for deeper investigation and high attention in the creation of benchmark datasets.

This outline of a single type of benchmark dataset for IDS evaluation only shows a trend. However, the risk of wrong decisions potentially introduced with inappropriate datasets for evaluation is more alarming in smaller, domain-specific research like the use case of this thesis. For this specific type of system, we could only find a few publications providing access to their dataset and among other publications, the datasets are rarely reused. In total, we are aware of two different occasions for publishing datasets for the evaluation of IDSs in this domain: (1) data traces directly published aside a security analysis as a documentation of the conducted attack (e.g. [84]). However, these datasets are rather small and contain merely a few variants of the same attack. Most critically, these datasets barely cover benign system behavior and are insufficient benchmark datasets without complementation of these samples. (2) Papers proposing new detection approaches and sharing the therefore collected dataset (e.g. OTIDS [67] or SYNCAN [47]) However, these published datasets are not reused for the development and evaluation of other IDSs. To

the best of our knowledge, only OTIDS was reused in further research within the same research institute and in its 2017 Information Security R&D dataset challenge. Instead, other researchers do not choose these datasets for their evaluations but favor generating synthetic data (e.g. ICSim [34] used in [28]) without publishing the concrete dataset. While all this enables a reproduction of the documented experiments, there is no critical reflection on these datasets.

Overall in both example use cases, there is no single, commonly agreed benchmark dataset, and no guidance or general methodology to collect data for the evaluation while ensuring a high quality of the benchmark dataset. Consequently, the quality and soundness of the results from evaluations with these benchmark datasets could be questionable without further validation of the IDS.

## 2.4 Metrics and Ranking

In the core, intrusion detection is a binary classification problem: The samples from reality have two different states, benign system operation and ongoing attacks, and the IDS aims to distinguish two classifications for each sample, staying silent and raising an alarm. Ideally, the IDS yields an alarm for all attacks while it stays silent for all benign system behavior. The third and final phase of the standard evaluation schema (see Section 1.1.2 for details) centers around metrics that quantify and rank the classifications of all candidate IDSs. The most basic way to quantify the classification behavior with a benchmark dataset is a confusion matrix as depicted in Table 2.1 denoting four values: the true positives (TP) counting alarms during ongoing attacks, true negatives (TN) counting full silence in benign system traces, false positives (FP) counting false alarms in a benign system trace, and false negatives (FN) counting missed attacks.

| | | Reality | |
|---|---|---|---|
| | | **Attack** | **Peace** |
| IDS | **Alarm** | True Positives (TP) | False Positives (FP) |
| | **Silence** | False Negatives (FN) | True Negatives (TN) |

**Table 2.1:** *Different classifications of system and attacker behavior*

There are various ways to interpret and summarize these four basic counters. As absolute numbers limit the comparison among datasets, the first step is to transfer them to relative rates between 0 and 1. The two commonly used rates are the Detection Rate (DR) and False Positive Rate (FPR).

$$DR = \frac{TP}{TP + FN} \tag{2.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.2}$$

They are estimators for the likelihood of an ongoing attack being detected and the rise of an alarm during regular system operation. Both estimators provide the y- and x-coordinates for the plot of the receiver operating characteristic (ROC) curve. To obtain a curve the performance of various configurations of the same detection approach (e.g. different thresholds, see Section 2.2 for details) are depicted in the same plot. Ideally, an IDS has a detection rate of 1 and a false positive rate of 0, hence its point in this plot is in the very top left. The closer the curve of all configurations of an approach is to this ideal point, the better. A curve of a candidate above the curve of another candidate is a strong indicator of a better-performing IDS. This notion be quantified with the ROC area under curve (ROC AUC) metric measuring the area below the curve seen from this direction.

Both estimators describe complementary properties of an IDS and the decision for deployment of IDS needs a suitable trade-off between the both. Consequently, more sophisticated metrics aim to objectify this tradeoff and unite both values into a singular score. The F-Measure or F1-score (the harmonic mean of precision and recall), although it is a common metric for evaluating binary classifiers, is the only used by the minority of publications [145] for evaluating intrusion detection systems. It does not account for the large imbalance in the IDS benchmark datasets and the base-rate fallacy (see Section 2.1 for details). An alternative, more suitable way to unite both estimators is a calculation of the expected costs [138] caused by not detecting an attack and the reaction to a false alarm. While it can be problematic to estimate the likelihood of an attack, it is a convincing metric to decide on the deployment of an IDS or a combination of multiple detectors. Other approaches use concepts from the information theory, for example, entropy reduction in the observations of an IDS [43] for a combination of both individual scores without further input. If needed, the entropy formulas can also include weights to model the notion of differing costs of missed attacks and false alarms.

## 2.5 Beyond the Standard Evaluation Schema

The standard evaluation schema (see Section 1.1.2 for details) provides the basic skeleton for an evaluation according to state-of-the-art knowledge. However, there are several existing directions of extension of this schema fostering different purposes than the contributions of this thesis. In the following, we finally want to highlight and motivate the main directions.

Resource usage and scalability are properties of particular interest for the assessment of candidate IDSs. Even the best detection is useless if the IDS is not capable of processing all information at the speed the system yields it during operation. While the required resources can simply be measured aside from all the other metrics in the standard evaluation schema, it makes sense to use a distinct dataset in a separate iteration to simulate a realistic workload for the IDS under evaluation. This dataset could be multiplications of the unaltered input from the original dataset only scaling the throughput into the IDS. An interesting metric is when the candidate IDSs cannot hold up with the amount of input data anymore or run out of the assigned resources. This knowledge allows extrapolation of the overall required resources and the costs of the deployment aside from the costs for processing and reacting to alarms.

Another direction to extend the standard evaluation schema stems from the need for data for training a more complex internal model of an IDS (see Section 2.2 for details). As it inhibits the risk of overfitting, it is important to strictly split training data from the dataset used for evaluation. Such a split between the first and the second step of the standard evaluation schema uses selected parts of the benchmark dataset for training and only uses the remaining data for the evaluation. However, this split can be implemented in multiple ways. Most prominently, k-fold cross-validation takes k different subsets of the benchmark dataset and conducts individual pairs of training and evaluation of the resulting slightly differing configuration of the same detection approach. If all different configurations show similar performance, this indicates the ability of the internal model to generalize well. Furthermore, an evaluation tailored for validating generalization can compose multiple benchmark datasets from multiple systems and evaluate configurations obtained from different combinations of these datasets. Ayoubi et al. [16] give a deeper elaboration on a potential evaluation schema for machine-learning based IDSs.

The existence of various configurations of the same IDS in the analysis also opens another direction to extend the standard evaluation schema. Before comparing different detection approaches with each other, first, the optimal configuration for each approach needs to be identified. For example, the ROC curve (see Section 2.4) is a metric that guides the search for these optimal configurations. However, concerning the increasing complexity and potency of the internal models a two-dimensional curve might not be sufficient to depict the full space of suitable configurations. Hence, it can make sense to investigate all configurations of the same IDS with a distinct setup of the standard evaluation schema to identify and tune the optimal configuration. If more than one detection approach is considered for deployment, another setup of the standard evaluation schema can be used to compare the best configurations of each candidate.

Finally, we want to point out that applying the standard evaluation schema also brings informal insights about the candidate IDSs. Domain experts get familiar with the candidates during configuration and training, although the first experiments with each detection approach might show very suboptimal performance. If the protected system changes or receives updates during the composition of the dataset or execution of the IDSs, these changes also provide insights about the generalization of the candidates from these changes. During the first configuration and all later tunings, the domain experts obtain the first impressions about how the later maintenance of the IDS during operation might look. Furthermore, the candidate IDSs, especially for updated or improved implementations, run through the standard evaluation schema multiple times. This and other development artifacts result in the extension, refinement and optimization of the datasets. Penetration tests that are conducted after the first version of the IDS has been deployed, also need to aim to circumvent the IDS and might result in new ideas for detection approaches or refining the existing setup. All analyses and considerations of the IDSs spin around the standard evaluation that mainly serves as a fixpoint in the search for quality improvement of the existing defense systems.

# 3 Use Case

*This chapter presents the use case we continuously used to shape and evaluate our methodology. It gives general information about the system and the relevant attackers and defenses. It highlights how the intrusion detection evaluation problem emerges in this use case. We have published parts of this chapter in [52–54].*

## 3.1 System: Automated Driving in Modern Cars

Security is a property of an entire system and does not exist in isolation without including a use case. In modern systems, it involves an analysis of an entire landscape spanning beyond the system and component boundaries. We decided to study the automotive domain and its current security problems very early in the process of this thesis. However, we found the enormous dimensions of all aspects and components relevant for the security of a modern vehicle too complex to make concrete contributions to the field [52]. Hence, we focused on very particular components and specific functionalities we believe to have high relevance and the most significant benefit from using intrusion detection as a defense measure. The following will introduce the relevant context and background information about our use case to understand their relation to our contributions and the foundation for our experiments.

### 3.1.1 The Controller Area Network Bus

The Controller Area Network (CAN) bus [110] is a network standard developed in the early 90s. It specifies very cheap, still reliable bus connectivity, which connects electronic control units, enabling safe communication within the vehicle. Hence, it became the de facto standard for internal network communication inside automotive vehicles in the following decades. Back in time, vehicles were isolated systems, and security concerns were not relevant during the design of the systems. Hence, the protocol does not contain authentication or tamper protection, which opens an attack vector.

In a simple view, the CAN bus connects the electronic control units (ECUs) in the car, allowing them to share information via broadcasts among all participants. A message consists of two parts: an arbitration ID defining the topic of the message and eight bytes of binary payload. A unified timing on the bus and scheduling mechanism based on the arbitration IDs ensure reliable communication. Lower arbitration IDs have a higher priority and are favored when multiple ECUs want to transmit a message simultaneously. The other ECUs (transmitting messages with higher arbitration IDs and lower priority) will again aim to send their message in the next slot. The CAN bus standard does not define the meaning of the arbitration ID beyond their priority nor the interpretation of the transmitted payload. These details are individual and specific to each car manufacturer and concrete car model. The manufacturers consider their definition, the so-called message-matrix, as confidential.

Car manufacturers are currently actively engaged in the development of new communication systems to replace the CAN bus. However, currently produced vehicles rely on the CAN bus for at least some parts of their critical communication. Consequently, the security vulnerabilities associated with the CAN bus will remain pertinent for the foreseeable future, as the existing fleet of vehicles on the streets incorporates this technology. Therefore, efforts to protect critical communication must consider both the legacy and future automotive communication systems, to ensure the safety and security of all vehicles in the coming years.

In our view, the CAN bus is a well-understood system with broad documentation from professional suppliers to simple hardware designed for hobby enthusiasts. Various tools and hacking instructions [34] are available, easing our investigations and research. Furthermore, we could buy essential network parts and chips with a low budget and build custom setups by combining many off-the-shelf components.

### 3.1.2 An ADAS Level 2: Open Pilot

The most revolutionary modern application is automated and autonomous driving. Car manufacturers equip modern cars with more and more sensors to constantly perceive the environment. These sensors provide sufficient information to comprehend the driving situation and predict future trajectories. Consequently, more and more systems provide assistance functionalities to the driver for increased safety and comfort. Ultimately, the final goal of these efforts is the fully autonomous driving of the car.

Although big commercial companies are the main drivers in this development, there are also smaller contributors following different business models and development approaches. Most prominently, comma.ai[1] provide an upgrade kit for various car models using the information from existing sensors beyond the original functionality in the car. Their software openpilot, among others, provides 1) adaptive cruise control (ACC), keeping the car at a specific speed while slowing down in front of obstacles or vehicles in front, and 2) lane-keeping assistance (LKAS) holding the vehicle in the middle of the lane. The combination of their hardware and all assistances resembles automated on Level 2 [113]. They publish their source openly, have released several public datasets, and propose open research challenges.

From the perspective of automotive security research, openpilot provides another significant contribution: A published and up-to-date message matrix for various modern car models. Their information does not cover all messages on the buses, but it deceivers all relevant information related to the automated driving functionality. We know approaches that aim to automate the manual reverse engineering [24, 25] of network recordings, but they still require substantial effort and are error-prone. Hence, this catalog was an essential foundation for the experiments in this thesis and several other experiments in its preparation. We could use it inside the car without further modifications, and it covers various signals.

For this thesis, the continuous use case resides within the automotive domain: Namely, using information and commands transmitted on the CAN bus to provide automated

[1] See `https://comma.ai/` and `https://github.com/commaai/`

driving functionalities on at least level 2. When necessary, we relied on the implementation and information provided by openpilot. However, we also have investigated several other cars and their technologies in the preparation of the research in this thesis and are confident we could adjust our implementation to the particularities of different systems, models, and manufacturers.

## 3.2 Attacker: Network Manipulations

As indicated in the introduction of the automotive bus system, the lack of cryptographic signatures in the CAN buses is a severe weakness within the internal network. However, this lack alone is no critical problem for the security properties of the car. As long as the bus is isolated from external access or other networks, the possibilities for exploits are limited to physical access to the car. With the emergence of digital technologies and higher demand for comfort features, cars became gradually connected to the outer world: first to the manufacturer and repair shops, then to the mobile network and the Internet, and finally, to the smartphones of their drivers. Thereby, the weaknesses of the internal network have become accessible from the outside of a car, posing a risk to the vehicle's safety. Miller and Valasek [85] were the first researchers to showcase viable remote attacks on an unaltered vehicle, which enabled them to crash the car under attack at will. This achievement follows previous research from Checkoway et al. [29] and Koscher et al. [63] exploring the attack surface on modern automotive vehicles.

Most importantly for follow-up research, all these documented attacks eventually manifest on the CAN bus and the attacker tampering with transmitted messages to manipulate the vehicle functionality. Therefore, this bus is the common segment in all these attacks that chain multiple exploits on various components. If some defense could prevent these manipulations on the bus, it would mitigate the potential risk of these attacks in a singular effort. Hence, we and other researchers focused on the security of this singular car component in isolation. Consequently, these works ignore all previous steps for an attacker and assume the attacker has gained access to the bus network, e.g., by infecting an ECU.

From the perspective of the CAN bus, an attacker can take four different locations in the network that provide the ability to inject new messages and tamper in different degrees with other transmitted messages. Figure 3.1 schematically depicts each of these location.

**Attached to the network:** The attacker attaches a new node he controls to the existing network. Aside from forcing errors in the communications, for example with message flooding, this attacker cannot tamper with the sent messages. For example, the attacker can use the on-board diagnostics (OBD) port located near the steering wheel of modern cars to gain this position on the network.

**Infection of an existing node:** The attacker infects an existing node of the network and uses its abilities to send messages. In addition to the attached to the network location, this also enables the attacker to modify the other messages that are originally sent by the infected node. For example, an attacker remotely infects the infotainment system or the tire pressure monitoring systems (TPMS) of the car.

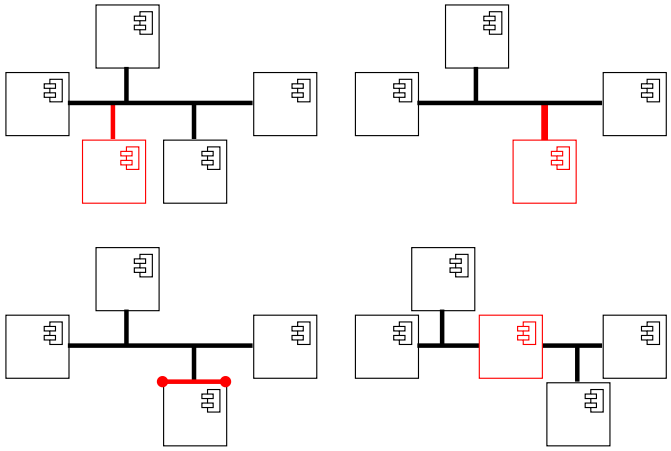

**Figure 3.1:** *Different locations of an attacker on a CAN bus. In lines from left to right, the figure depicts an attaching, infecting, isolating, and man-in-the-middle attacker. The original bus is drawn in black, while the modifications of the attacker are highlighted in red.*

**Isolation of a single node:** The attacker manages to infect parts of an existing node, e.g. the internal message dispatcher, but not be able to take full control yet. This location is very similar to the full infection of an existing node, but the concrete abilities of the attacker to manipulate might be limited to only a subset of the originally sent messages of the ECU. For example, an attacker might gain this location when the ECU is more powerful and complex and uses some form of virtualization or process isolation.
**Man-in-the-Middle:** The attacker splits the network into two sub-networks and acts as a transparent gateway between both halves. This location gives the attacker full control of the messages transmitted from one subnetwork to the other and vice-versa. For example, an attacker can gain this location through physical manipulations of the network (often used by car owners who aim to tune their cars) or via infecting a router between two networks (as a result of spreading the infection of another network node).

Concerning our use case and the experiments in the thesis, we want to point out two specificities of our attacker that go beyond the common attacker model. 1) While there are various potential goals for a CAN bus manipulation, e.g., tuning the car or unlocking premium features without payment [52], we focused the scope of this thesis on a single main goal: The violation of safety properties of the car to physically harm the driver and the passengers of the car. This focus mainly restricts the payloads the attacker transmits on the network (e.g. no system exploration or reverse engineering) and excludes deeper analyses and infestations of the ECUs. 2) As the majority of works focus on intrusion detection, we skipped the first steps of the attack gaining access to the network of the car. To conduct attacks in a real car, we obtained access to the internal CAN bus through physical manipulations within the car. Concretely, we extended an internal CAN bus with additional cables and connectors (see Section 5.5.1 for the technical details in the context of our attacks). This approach is unrealistic for a real-world attacker and the manipulations we document require further exploits to be weaponizable in the field. Nevertheless, this reduced the scope of the manipulation, lowered the necessary effort, and enabled us to ensure our safety during the manipulations in a driving car.

## 3.3 Defense: Intrusion Detection

The CAN bus is a minimal system not fully compatible with classical security approaches. Its old hardware, with minimal free resources and the limited capacities of the bus medium, does not permit an upgrade to classical security measures like cryptographic signatures. More powerful hardware enables strong cryptography and new protocols like Automotive Ethernet [135] that prevent the above security problems. Nevertheless, this transition requires a complete network redesign and is a significant effort for the car industry. It will require several generations of cars until the manufacturer has entirely replaced the CAN bus in all critical communication.

From a security point of view, this transition is the right approach as it will drastically increase the complexity of all attacks discussed and conducted in this thesis. However, the required transition period still leaves the cars already on the streets and currently manufactured at risk. Therefore, academia and industry have focused on directly addressing the weakness of the CAN bus. Especially Intrusion Detection has become the preferred approach to mitigate this risk [23]. Assuming a reliable detection capability and a suitable reaction to detected attacks, manufacturers could comparably easily add a new component monitoring the existing networks to increase security tolerably in the transition.

After the first documentation of these critical attacks, academia had high competition in the past decade for developing and optimizing an IDS for the CAN bus. Several literature reviews [10, 55] aim to structure the research contributions and various publications. Despite the long research period, we are unaware of a sophisticated CAN bus IDS deployed in a regular car model. However, IDSs are currently becoming standard components for vehicles to fulfill the requirements of security and intrusion detection stated in new standards like ISO 21434 and will surely mature and advance further in the future. For the research in this thesis, we had the chance to cooperate directly with a car manufacturer facing this plethora of approaches, which continuously provided valuable insights and guidance.

Car manufacturers produce standardized models in high numbers. Hence, a few cents more or less per component already make a significant impact on their profit. Furthermore, each car must fulfill a growing safety requirement catalog to obtain street homologation. These requirements transfer to all components inside a vehicle and apply to any potential IDS. Building autonomous cars is, therefore, subject to high cost-pressure and rigorous safety requirements. Due to the shape of automotive supply chains, car manufacturers are privileged to choose among components and IDSs offered to them. However, they lack a suitable workflow that helps them differentiate and choose between similar detection approaches. In short, this use case is an excellent example of the intrusion detection evaluation problem.

# Part II

# New Phases for IDS Evaluations

# 4 Prefilter by Abstract Properties

*This chapter presents four different ways to use abstract properties to prefilter and select promising IDS candidates early in the development process and without any execution or implementation. For this purpose, we elicit a broad taxonomy of abstract properties that describe detection approaches from expert interviews and a systematic literature study. We analyze this taxonomy on an exemplary use case to identify effective means to differentiate between IDS candidates presented in academic literature. Our findings support the suitability of our prefilters and we showcase their utility based on an exemplary case study on the use case.*

## 4.1 Introduction

The emergence of new technologies continuously changes the need for protection and enables new detection methods. For example, the progress in machine learning has been reflected in the creation of systems with new capabilities and architectures, and also a new generation of powerful anomaly-based detectors. Hence, there is no fixed "universal detector" that can be developed once for all use cases or critical systems and after deployment remains invariant during the lifetime of the system. The decision for the most suitable IDS with given constraints from the use case involves several trade-offs and all the related factors change over time. The ongoing change in technologies implies continuous validation and reconsideration of potentially better IDSs. Therefore, precisely identifying and describing trade-offs in the approaches for detection as well as quickly comprehending the difference between competing IDSs is vital to keep up with these developments. It is the only way to identify the best currently available detection approach and, thereby, assure optimal protection.

Academic publications about new detection approaches follow the standard evaluation schema (see Section 1.1.2). A set of implemented IDSs analyses the same invariant benchmark dataset and selected metrics rank their performance. On the foundation of these publications, industry experts take a different view on this growing number of IDSs. In larger teams, the experts responsible for security might have little impact on the overall system they need to protect. In our exemplary use case introduced in Chapter 3, the industry experts even need to find a protection for a legacy technology. Furthermore, system owners are only interested in and aim to only invest in the single best IDS out of the set of all potential detection approaches. The spectrum of requirements in this setup and the corresponding suitability of an IDS for a given use case reaches beyond the detection abilities measured in the standard evaluation schema. Even worse, the external and exemplary benchmark datasets used for evaluation might not be accurate representations of the industry expert's concrete system. Consequently, the final decision requires the domain experts to conduct subjective analyses with their own systems and customization of any investigated IDS. However, this also implies costs proportional to the number of

considered detection approaches. In our study presented later in this chapter, we found 248 publications about new protection mechanisms for the CAN bus based on intrusion detection. It is not feasible for system owners to analyze and compare all these approaches based on full system integrations with respective benchmarks and therefore, many approaches are currently neglected before any investigation.

This chapter elicits and utilizes a taxonomy of the abstract properties of IDSs and their interconnections with the system and attacker independent of their concrete implementations. The methodology for its creation considers on the one hand the perspective of industry experts gathered in interviews and on the other hand the academic view extracted from peer-reviewed literature. Although we propose a universal methodology for their elicitation, following this methodology results in the majority of these properties describing fine-grained and thereby use-case-specific differences. Most importantly, such taxonomy classifies and describes various aspects and characteristics of an IDS without requiring any system, execution or realistic data collection. Furthermore, the elicited taxonomy tailored to the exemplary use case covers broader information than all previously published taxonomies of intrusion detection in this use case (Section 4.5.1 presents an in-depth comparison).

This taxonomy structures the properties of a candidate IDS and it explicitly lists a diverse spectrum of characteristics. Mapping the competing candidates to this same structure enables a quick and comprehensible overview. Furthermore, it lays the foundation for systematically identifying promising candidates for further investigations and investments. In our study, we analyze a mapping of the properties of 21 IDSs to our taxonomy and propose four distinct prefilters based on our findings. Our suggested prefilters rely on enforcing or denying particular properties for a use case, or requiring a specific diversity among the candidates. Thereby, the combination of the taxonomy and systematic prefilters provides an early preselection before further investigations through the standard evaluation schema. This saves resources for implementing and tailoring the excluded candidates and increases the chances for beneficial IDSs within the set of candidates, before conducting further analyses.

In parallel to the diverse nature of systems and use cases for the IDS, it is problematic to create a single, universal taxonomy that comprehensibly covers all important aspects of all IDSs. We try to accommodate this challenge through different levels in our taxonomy. Namely, the metamodel and generic core properties (see Section 4.3.1) are invariant of the use case and form a solid skeleton for representing a use case with additional information. As confirmed with the analysis in Section 4.5, more specific details are required for a suitable description and discrimination of the specificities of the various candidates for the same use case. In total, our taxonomy unites three different views on core aspects of intrusion detection (protected system, attacker behavior and detection mechanism) and integrates different levels (from generic to use case-specific aspects) in a common structure suitable to describe an IDS before and without an implementation.

In other words, taxonomies of IDS properties reside within a trade-off in their generalization level, namely between a universal, but abstract view describing high-level properties on the one side and a very detailed, but also use-case-specific view describing fine details on the other side. As a first step to gaining an understanding of the spectrum

of granularities and their suitability as a filter, we focused on our exemplary use case and aimed to cover all its relevant aspects. We base our taxonomy on (1) an initial structure assembled through interviews with industry experts (in Section 4.3.2) and (2) refinement and validation through a systematic literature study considering all 22 existing survey papers (in Section 4.3.3) as well as (3) extracting and mapping the properties of 19 selected, relevant new-idea publications about IDSs for our exemplary use case (presented in Section 4.5.2) to the taxonomy. Although our taxonomy is designed for the automotive domain and the CAN bus, our structure and methodology can be extended and transferred to use cases of other domains and lays a valuable foundation for this future work.

In an analysis of the differing perspectives in our research, we found that the industry experts' perspective focuses on properties related to processes and the integration of the IDS. In contrast, the academic view focuses on the detection mechanism and specific features used for detection. In addition, we found a varying density of references to characteristics in our taxonomy that we utilize to design four potential prefilters. Finally, we demonstrate in a case study (see Section 4.6) that with these prefilters only very few, but selected properties are sufficient to identify suitable approaches for further extensive analyses with the standard evaluation schema.

Overall, this work makes the following contributions: 1) The proposal of four systematic prefilters identifying suitable candidates for a use case without any execution of code based on eliciting requirements on abstract properties of the use case. 2) A taxonomy about abstract intrusion detection with an invariant metamodel and general properties in its core and additional refinement for automotive CAN bus intrusion detection. We build this taxonomy by uniting two smaller contributions: 3) A collection of views on abstract properties of automotive IDSs from industry experts obtained through multiple interviews, and 4) a systematic literature study on academic literature about intrusion detection on the automotive CAN bus establishing links of the abstract properties of IDSs presented in these works to the unified structure of the taxonomy.

The remainder of this chapter is structured as follows: Section 4.2 relates our contributions to the standard evaluation schema. Section 4.3 elaborates on the methodology we used in this research to create the taxonomy. Section 4.4 presents the resulting taxonomy in full detail. Section 4.5 analyses in depth the links of the survey publications and the selected IDS candidates to the taxonomy. Section 4.6 introduces the prefilters and documents their utilization in a case study with industry experts. Section 4.7 summarizes the findings of each previous building block and discusses their implications.

## 4.2 Relation to the Intrusion Detection Evaluation Problem

The standard evaluation schema is shaped around a central step: The candidate IDSs analyze the same benchmark dataset representative of the system and attacker behavior. More precisely, each IDS in the candidate set has to analyze this large dataset completely, while its performance during operation and especially raised alarms are recorded. Afterward, various metrics summarize the alarms and compare the performance documented in these recordings. However, all these metrics require the full, recorded data from the analysis in the central step. Hence, it is mandatory to execute all candidate IDS for the evaluation like they were used in the real system.

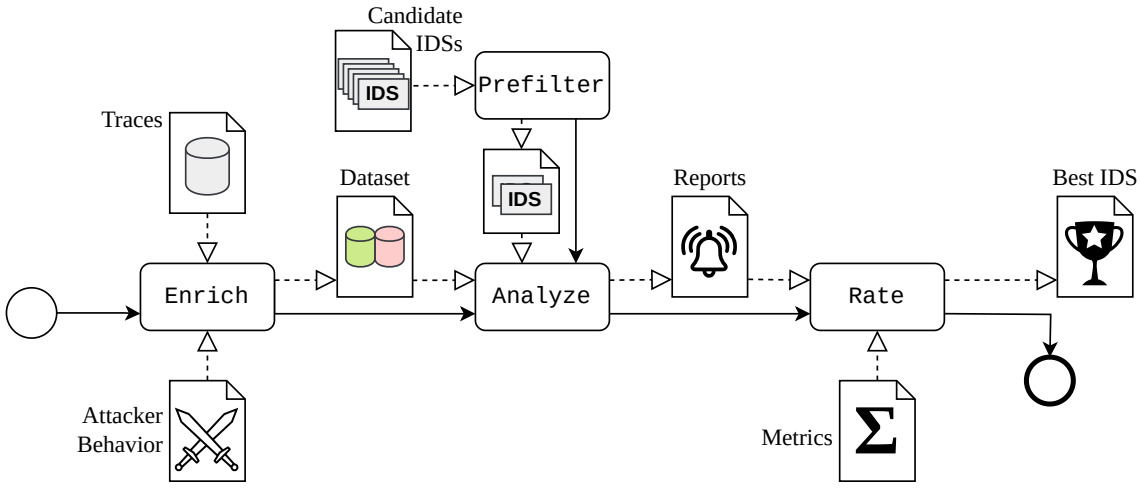


**Figure 4.1:** *The standard evaluation schema extended by our prefilter.*

An evaluation around such a central step suffers from a fundamental flaw from the perspective of a user searching for the best IDS for their use case: To obtain meaningful results the execution requires implementations and proper configurations of the system and each analyzed IDS. First, without an implemented and operational system, the traces for the benchmark dataset cannot be collected. This delays the first evaluations of potential IDS in the overall development process to the testing phase and separates the system design from the decisions about a potential IDS. Hence, the implementation of the system might unknowingly neglect properties of the system that would be beneficial for the detection of attacks. After the system implementation, it might be expensive or impossible to change these properties. In addition, each candidate IDS needs to be executable within the system before its performance can be recorded. This implies costs and overhead for implementing and integrating every IDS before any analysis is conducted. In the worst case, the IDS might turn out fully inappropriate for the use case and is without any value after these investments. Ideally, an evaluation would consider a large number of candidate IDSs to obtain the best detection capabilities among all available options. However, the inclusion of a large number of candidates also linearly increases the costs for their initial implementations and configurations. Therefore, domain experts naturally tend to cut the number of considered candidates short. Even worse, they are forced to select candidates for implementation before the standard evaluation schema is applicable and without sufficient information or guidance.

Our work addresses these limitations by extending the standard evaluation schema as follows (cf. Figure 4.1): We propose the usage of a prefilter that systematically selects the most promising IDS based on qualitative properties and without depending on any implementations or configurations. This analysis and comparison can take place very early in the system development and parallel to the system design and the early security engineering. Thereby, this prefilter raises awareness for relevant system properties that can be considered before system implementation. The extraction of abstract properties and evaluation of the filters we propose requires much less time and resources than an

implementation. Hence, a broad set of candidate IDSs can be considered without major investments. While investigating new approaches, our taxonomy structures and guides the analysis to cover all relevant characteristics, especially when the original presentation lacks this information. Furthermore, the prefilter identifies candidates that are not suitable for the use case early in the process, and, therefore, prevents further wasted investments into these approaches.

While continuing with the development process, domain experts analyze only the candidate IDS passing this prefilter by following the unaltered standard evaluation schema. They invest in functional prototypes of the selected candidate IDSs and extract the traces necessary for a benchmark dataset from the system once it is implemented and operational. The standard evaluation schema guides further decisions about the suitability of the candidates and provides quantitive analyses complementing the qualitative considerations of the prefilter. In short, the prefilter has the purpose of increasing the number of considered candidates at the same costs and increasing the likelihood of selecting a well-performing IDS for full evaluation.

## 4.3 Methodology and Taxonomy Elicitiation

The core of a prefilter for suitable IDSs is a collection of abstract properties that describe all possible candidate IDSs and differentiate them in the necessary level of detail. These properties describe qualitative aspects related to the IDS and can be determined without executing the IDS or the protected system. The properties are natural language formulations and are informally used whenever IDSs are introduced or compared. Their required level of detail for differentiation between approaches depends on the traits of all compared candidate IDSs. We contribute by naming these properties explicitly and structuring them within a taxonomy complying with a generic metamodel. We aim for a broad collection of diverse abstract properties that we investigate further in the following analyses to deduce the prefilters. This section depicts our considerations and steps for eliciting the taxonomy of properties.

The design of a taxonomy resides within a tradeoff of two contradictory goals: the adequate and universal description of every IDS and the fine-grained differentiation between similar approaches of the same use case. As we are unaware of a means to determine the suitability for differentiation of a property upfront, we decided to favor the deep and detailed analysis. The following methodology aims for a complete collection of characteristics on various abstraction levels and views, but only concerning an individual use case. In a follow-up step, we dedicate an explicit analysis of an exemplary mapping of candidate IDSs to the taxonomy to elicit appropriate and universal filters and investigate the suitability of the different abstraction levels for this purpose. We want to tailor the prefilters to this fine-grained differentiation as domain experts in practice face this scenario and the decision between slightly different IDSs mitigating the same security risk. Consequently, in this study we only investigate one specific use case and application for IDSs: Intrusion detection for the controller area network bus (CAN bus) in the automotive domain. The original equipment manufacturer gathers candidate IDSs as black boxes

from various possible vendors to identify suitable and, foremost, the cheapest mitigations. This setup is an optimal scenario to investigate the novel idea of our prefilter.

The identification and enumeration of various aspects and characteristics is a challenging task. From our experience, creating such a taxonomy all at once in a single step is impossible. Therefore, we executed multiple iterations of extension and refinements that concluded in the taxonomy we present later. Namely, our methodology to create the taxonomy consists of three main building steps: 1) The introduction of a metamodel to structure all relevant properties (see Section 4.3.1), 2) interviews with industry experts from an automotive manufacturer (see Section 4.3.2) to obtain a skeleton of properties and foundation for 3) the completion with properties gathered from academic literature about the use case consisting of survey papers and publications proposing new detection mechanisms (see Section 4.3.3). Furthermore, we used the pool of newly proposed detection mechanisms for the following case study and evaluation in industry (see Section 4.6). The metamodel is the invariant structure we used to consistently describe and discuss abstract properties. It enabled us to unite different views and abstraction levels in a uniform representation. Furthermore, the metamodel is the data format used by the prefilters and the corresponding toolchain. In interviews with domain experts from the industry, we gathered views and experiences of accessing IDSs in practice. We used them to validate our metamodel and compose an initial collection of abstract characteristics. This skeleton served as a foundation for the literature study and prevents bias from and overfitting to the publications we analyzed first. In the literature study, we linked properties discussed in the publication to the taxonomy and each contained properties. If necessary, we added new properties or refined existing properties to represent the new property adequately. If this step was necessary, we revisited previously linked publications and ensured correctness and consistency in all links. The following presents each of the building steps in more detail.

### 4.3.1 Taxonomy Metamodel

We use a metamodel as a unified way to formalize the properties of the candidate IDSs as well as the underlying data model for the prefilters. In short, the metamodel defines how the properties are represented and how they are used to describe an IDS. This design with a metamodel enables easy extensibility and customization during the elicitation of properties. Furthermore, it provides possibilities for adjustments to future development in detection mechanisms and transfers to other protected systems. Where needed, the taxonomy can be extended or refined by additional information in the same way at any time. Furthermore, properties that do not apply to the use case or do not add to the differentiation in the concrete usage can also be removed. With the invariant metamodel, the implementations of the following filters and analyses remain functional and descriptive.

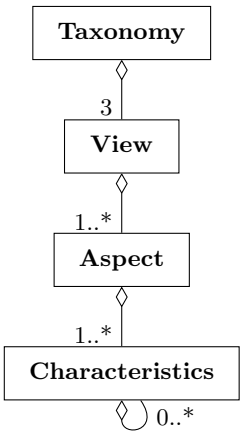


**Figure 4.2:** *Metamodel of the Taxonomy of abstract properties*

Overall, the metamodel follows the idea of hierarchical clustering with three levels. Each IDS is described via a complete mapping to the broad set of characteristics in the taxonomy. The characteristics are grouped into aspects, that belong to one of three views. Figure 4.2 provides a formal definition of metamodel. In the following, we will describe the structure from the bottom upwards:

The smallest element in our taxonomy is a *characteristic*. A characteristic is a human-understandable text describing a specific abstract property. An abstract property is a qualitative trait expressive without referring to the concrete implementation in code, especially without executing any system. For describing a particular IDS or candidate, we link all characteristics with one of the following labels: If the candidate complies with the characteristic, we encode this with a boolean *"`true`"* linking the characteristic and IDS. If the characteristic does not describe the properties of the candidate, we encode this with a boolean *"`false`"* linking characteristic and IDS. While this mapping in theory is sufficient to map all candidates, we introduced two more labels extending this binary logic to enable a convenient application in industrial workflows. As characteristics abstract from the concrete system or execution, the currently available knowledge may lack the information required. Thereby, it is impossible to confirm or disprove the characteristic of the candidate, and we encoded this with a *"`maybe`"*. Finally, we rarely have been in situations when one of the candidates was more extraordinary and a more specific characteristic did not make sense in this content. We encoded this as not-applicable, *"`n/a`"*.

Our characteristics aim to differentiate fine differences that might not always be known with the currently available information about the IDS and can only be assumed with some uncertainty. This results in a situation, where details in a characteristic are not known, while the remaining higher-level information of the characteristic can easily be classified. Therefore, we grouped, where appropriate, more concrete formulations under a more generic aspect. Consequently, in the meta-model, any characteristic can be composed of other characteristics. Furthermore, we noticed that this structure helped the domain experts understand the minor differences between the aspects more precisely.

The following example showcases how an IDS can be described with the characteristics and corresponding mappings: The most often referenced aspect to differentiate a detection mechanism is the "source of data". Related to this aspect, two general characteristics are: 1) monitoring information about the system and the program execution on it; 2) monitoring information about the communication sent and received on the network. A mapping to these characteristics may describe an IDS as the following. A classical host-based IDS monitors the system and not the network traffic, hence, its mapping is encoded with the tuple `(true, false)`; a classical network-based IDS monitors the network and neglects all system information, hence, its mapping is encoded with the tuple `(false, true)`; A modern, hybrid approach might combine information from both sources, hence, its mapping is encoded with the tuple `(true, true)`. Overall, this aspect with these two distinct characteristics classifies all these types of IDS comprehensibly.

As depicted by the example above, characteristics have a strong relation to other characteristics. Also, the textual explanations might be redundant in large parts of their formulations. Therefore, we decided to group closely related and similar characteristics in an *Aspect*. Every aspect is represented by a natural text formulating an explanatory question

asking for a property of the candidate IDS. Grouping within aspects removes redundancy in the formulations of the subordinate characteristics. Choosing or dismissing the grouped characteristics can be interpreted as possible answers to the question stated in aspect. In the example of the source of data, the question denoting this aspect is: Where does data monitored by the IDS originate? The aspects from the network or the host system provide the reply to this question.

Finally, in alignment with the core hypothesis of the thesis, we related each aspect to a *view* on the use case, namely, the protected system, the attacker and his behavior, and defense and detection in the IDS. These views aim to ease the orientation and navigation within the taxonomy. We will also relate to the views later in the experiments and evaluation. Generally, we fostered a high number of characteristics while undesirable or non-disciminatory characteristics are removed later once a concrete set of candidates is provided. Consequently, our final taxonomy consisted of 179 individual characteristics. We grouped these in a total of 34 aspects, and three views forming our taxonomy.

At first sight, the simplistic taxonomy might look insufficient to depict reality and its complexity. There are interrelations between the aspects, for example, specific characteristics that exclude or imply another. In the example of host- and network-based IDSs, one type of data source excludes the other for non-hybrid approaches. Furthermore, a mapping of `(false, false)`, or a detection without using a data source is not reasonable. However, we intentionally ignored such interrelations in the model to keep the taxonomy flat and each characteristic distinct and comprehensible. Nevertheless, all such correlations are provided "by examples" as each mapped IDS is a real IDS that exists concretely within all these trade-offs and design decisions and complies with all such logical deductions. While accumulating the mapping of many or all candidate IDSs, the combination of various realizations implicitly encodes these interrelations. In particular, all interrelations prominent in the investigated candidates are dominant via this accumulation while others, non-present interrelations are neglected. Therefore, our prefilters presented in Section 4.6 still profit and use these interrelations without any explicit modeling. Noteworthy, in the investigation of the exemplary mapping presented in Section 4.5 we discovered more of these correlations than we had anticipated and were named during the industry interviews. However, we could not assess their relevance objectively or prioritize them universally. Therefore, we decided against efforts to model any selection or subset explicitly in our models and kept the structure strictly hierarchical.

This flat and simplistic structure unfortunately emphasizes an effect that we observed while eliciting all aspects: redundancy. In our discussions about the aspects and the industry interviews, the most frequent question was about how two similar aspects differ precisely and why they are not merged. However, we noticed that merging aspects makes the formulation longer and more complex to be comprehended. On the other hand, having similar aspects independently listed keeps slight differences and details visible. In addition, it inhibits minimal overhead in the processes of mapping and filtering. Therefore, we decided to consider redundancy as a feature and did not aim to reduce it in the final catalog of aspects.

### 4.3.2 Industry Expert Interviews

With the metamodel structuring the candidate descriptions, the next step is to start the collection of suitable properties. For this purpose, we decided to conduct interviews with domain experts from the industry to obtain a snapshot of the properties relevant for them to describe, distinguish and rate IDS candidates. We collaborated with an automotive manufacturer, or more precisely, with individual employees of three different departments. All three experts were from the security field and had worked with intrusion detection systems for the CAN bus but on different areas. They all had experience with approaching the intrusion detection evaluation problem during their work. Various original equipment manufacturers and suppliers propose security solutions, and our industry experts are responsible for making the selection and deployment decisions.

To create a first draft of our taxonomy, we had an initial round of interviews in each department independently. We asked a set of open questions: What are the properties that describe an IDS in general? What properties differentiate the candidate approaches? We collected and noted their answers and clarified terminology and domain specificities during the interviews. Often, the domain experts had concrete candidate IDSs in their minds and enumerated properties they would need to describe these approaches and their strengths and weaknesses. Finally, they also linked characteristics that are important for their domain and use case. After the interviews, we consolidated and merged the noted properties. Characteristics that were named multiple times, we only listed once. Wherever suitable, we grouped related characteristics in the same aspect. After the initial interview, we conducted a meeting with the participants where we presented a draft of the taxonomy and collected another round of feedback. Namely, we validated our summary and identified misunderstandings in the formulations. Furthermore, we collected more suggestions about refinements and new characteristics. This process resulted in 120 characteristic covering 26 distinct aspects. It shows the perception and understanding of abstract IDS descriptions from our industry experts. The analysis in Section 4.5.1 opposes this view of the industry experts in the framing of the final taxonomy to the view in academic publications that we will collect in the following.

We want to allude that we have not and cannot determine how representative this rather small group of industry experts from three departments in a single company is. However, the main role of the industry experts in this step was to provide an initial seed for the following literature study. Missed or unrelated aspects and characteristics will therefore be spotted in the later analysis. More details about the impacts of the small group size are discussed in threats to external validity in Section 4.7.2.

### 4.3.3 Systematic Literature Study

The industry expert interview provided a consolidated first version of the taxonomy. However, to complete and refine our taxonomy we continued our work by examining academic literature. We conducted the following study during the year 2021. In our literature study, we differentiated the literature into two different types of publications that we distinctly used for different purposes:

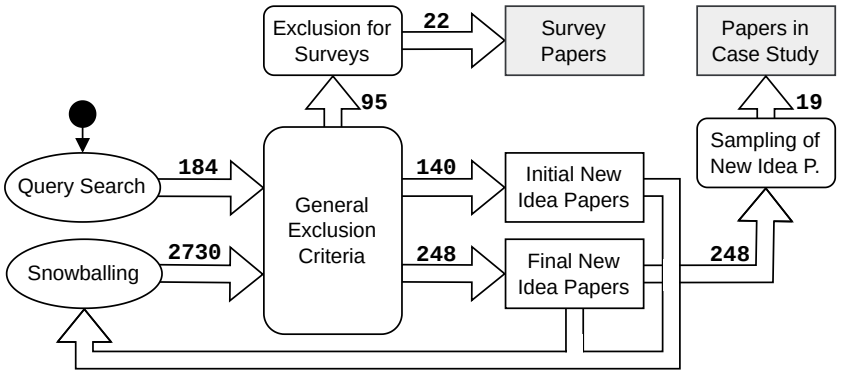


**Figure 4.3:** *The process of our literature study showing all important steps. The numbers next to the arrows denote the number of publications as the input and the outcome of each corresponding step.*

**A) survey papers**: The main focus of survey papers is to review diverse, existing works about intrusion detection and provide a structured overview of the state-of-the-art as the main contribution. Intuitively, their structure and classification are related and similar to the properties denoted in our taxonomy. Hence, we used survey publications in the next step (see paragraph References from the taxonomy to survey publications) to validate, complement and diversify the properties in our taxonomy previously collected in the expert interviews.

**B) new idea papers**: The main focus of new idea papers is to present a new detection mechanism and showcase its benefits over other IDSs. Intuitively, these descriptions and argumentations provide similar information and the same details available for domain experts during their decisions for promising candidate IDSs before implementation and customized evaluation in the standard evaluation schema. Hence, we used a mapping of the detection mechanisms introduced as the main contribution by these peer-reviewed publications to the taxonomy (see paragraph Mapping of new idea publications to the taxonomy) as the candidate IDSs in the later case study in Section 4.6.

#### Identification of Relevant Literature

We followed the guidelines from Kitchenham and Charters [60] and Wohlin [141] while working with the literature. In particular, we used the concepts of a mapping study to link the properties of our taxonomy to the published texts. These descriptions and Figure 4.3 elaborate in detail on the process of how we have identified sets of publications from both types. In the figure, the boxes with rounded corners correspond to the subsections in the following explanations with the same title. We want to point out that both utilized literature sets, survey papers and new idea papers, have no intersections. In other words, although we selected them within the same process, no publication is considered twice for both purposes.

For building our literature pool we used two different methods: The search by query and snowballing. We started with a query search with the identical query (see the paragraph Query search: for details) on the search interface of four different databases for academic

literature: Scopus, WebOfScience, dblp, IEEE Explore and ACM digital library. Then, we removed all duplicates that resulted from the same publication found multiple times in different databases. The resulting set of 184 publications contained several papers that were not relevant to our research. Hence, we filtered the publications by several exclusion criteria based on automatic checks and manual expert voting (see the paragraph General exclusion criteria: for details). Most importantly we separated all survey papers from the discovery process and continued only with new idea papers. The 140 publications passing all filters are relevant papers proposing new detection approaches. We used them as a seed for forward- and backward snowballing in the literature databases of Scopus and WebOfScience. Then, we applied the same processing steps (duplicate removal, filtering and survey separation) to all newly found publications resulting in a larger set of final new idea publications. Finally, we repeated the snowballing process two more times with only three new publications discovered in the second iteration and no new publications in the third iteration.

Overall, this process resulted in 248 publications presenting new ideas and 22 survey papers. As investigating all new idea papers in detail is beyond the scope of this thesis, we sampled these publications to 19 publications for our case study (see the paragraph Sampling of new idea papers: for details). For orientation, these final sets are denoted as gray boxes in Figure 4.3. The remainder of this subsection focuses on specific steps in our methodology and provides all the details necessary to reproduce our literature discovery.

**Query search:** Our query is the conjunction of two parts that both list phrases that need to be contained in the title, abstract, and keywords (if supported by the search engine) of the publication: The first part yields publications about intrusion detection and the second part restricts the broad landscape to the concrete use case. In detail, the query was the following:

("Intrusion Detection" `OR` "Anomaly Detection")
`AND`
("Controller Area Network" `OR` "CAN bus")

If our methodology were applied to another use case, only the second part of the query would be adjusted.

**General exclusion criteria:** Then, we used a set of exclusion criteria to remove nonrelevant papers from the previously discovered literature. In particular, the automatic filters helped keep the number of papers that required manual investigation and voting sufficiently low. Some exclusion criteria check for the words of the search query and related terms. While publications that are found via query search directly pass these checks, these are particularly relevant for papers that are discovered via snowballing and lack these exact terms.

1. Exclude papers that have been published before 1991, the release year of the CAN 2.0 specification.

2. Exclude papers that do not contain at least one security-related string in their title, abstract or keywords. Namely, these strings are "security", "attack", "defen", "intrusion", "anomal", "detect", or "vulner". Strings that are fragmentary words ensure that all words with this radical do match, e.g. "defen" matches the words defense, defender, defending, etc.
3. Exclude posters. We identify this type of publication by checking for the string "poster" in the title or keywords.
4. Exclude papers that do not contain at least one string related to our use case in their title, abstract or keywords. Namely, these strings are "automotive", "vehic", "driver", "embedded system", "electronic control unit", "ECUs", "autonomous car", "ADAS", "controller area network", "CAN bus", or "arbitration ID". Fragmentary words again match all words with this radical.
5. Separate potential survey papers that contain at least one string denoting their research nature in their title or keywords. Namely, these strings are "survey", "taxonomy", "mapping study", "comparative study", "ontology", "review", "overview", "research results", "research directions", "research challenges", "future challenges", or "future directions". The papers excluded in this step were considered in a separate analysis continued in paragraph Exclusion for survey literature: resulting in a lost of Identified survey papers:.
6. Exclude papers that do not present an IDS for our use case via human expert voting. In the voting, each researcher read the title and abstract of the paper and gave a boolean answer to the question "Does the paper propose at least one IDS for protecting the CAN Bus?". Each paper was investigated independently by at least two researchers who discussed until agreement on the classification if their initial judgments differed.

All publications passing this exclusion process form the set of new idea papers. We used this literature set for our case study where we extracted a set of candidate IDSs from it that our prefilter should assess. Furthermore, these papers were the initial seed for the snowballing process. All citing and cited publications from these papers were added to the same pipeline as the results from the query search was before. Namely, we removed duplicates, excluded nonrelevant publications and separated survey publications. Ultimately, this resulted in 248 unique new idea papers after two rounds of snowballing without new literature in a potential third round.

**Sampling of new idea papers:** While sketching this methodology, we did not expect this large amount of scientific publications of new ideas for detection algorithms. We do not have the resources to conduct the full-text analyses of all these publications that are necessary in the next step towards our case study and the later prefilters. Hence, we needed to sample this set of papers down to a smaller number of papers that we could analyze in depth. As all publications are relevant and suitable for the following analyses, it would have been acceptable to just do a random selection. However, we decided to focus on impactful and highly discussed approaches, as we believe this property relates to a high quality of presentation and high adequateness of their approach. With this decision, we

wanted to ensure the existence of promising approaches in our sample. Furthermore, it is more realistic that domain experts use the prefilters presented in Section 4.6 to differentiate between prominent and more mature solutions instead of rather unknown newcomers. Therefore, we used other criteria based on citation counts for the sampling process.

Using high citation counts as the only selection criterion results in an inherent bias: Older publications are naturally more often cited than new publications. Hence, we used two different citation counts: total citations and citations from the year 2020 (the year before this study was conducted). Furthermore, to prevent biases from varying results in the literature databases, we used both Scopus and WebOfScience for the lookup and merged the resulting publications. Finally, we had a look at the venues where all new idea papers were published and decided to include all papers from the highest-ranked conference and most impactful journal. Namely, to determine these rankings we used the ERA and Qualis conference ranks as well as CiteScore 2020 on Scopus. Overall, these considerations resulted in the six sets denoting prominent publications that we will list below.

- `Os` the top five publications with the highest total citations listed on Scopus
- `Ow` the top five publications with the highest total citations listed on WebOfScience
- `Ys` the top five publications with the highest citations in 2020 listed on Scopus
- `Yw` the top five publications with the highest citations in 2020 listed on WebOfScience
- `TJ` publications in the IEEE Transactions on Information Forensics and Security (TIFS)
- `TC` presented at the ACM Conference on Computer and Communications Security (CCS)

We later use the denoted two-character codes to reference the source of each publication in the final selection. As these definitions are rather similar, a publication can be contained in multiple of these sets. Thus, this process overall resulted in 19 unique publications after removing the duplicates for the considered 26 publications.

**Selected new idea papers:** This discovery process and sampling yielded the following literature as new idea papers for the case study.

[30] (`TC`) K.-T. Cho and Shin (2016): *"Error handling of in-vehicle networks makes them vulnerable"*

[31] (`TC`) K.-T. Cho and Shin (2017): *"Viden: Attacker identification on in-vehicle networks"*

[32] (`Ow`, `TJ`) Choi et al. (2018): *"VoltageIDS: Low-level communication characteristics for automotive intrusion detection system"*

[42] (`TJ`) Groza and Murvay (2019): *"Efficient Intrusion Detection with Bloom Filtering in Controller Area Networks"*

[47] (`Ys`, `Yw`) Hanselmann et al. (2020): *"CANet: An Unsupervised Intrusion Detection System for High Dimensional CAN Bus Data"*

[50] (`Os`, `Ow`) Hoppe et al. (2011): *"Security threats to automotive CAN networksPractical examples and selected short-term countermeasures"*

[56] (`Os`, `Ow`) M.-J. Kang and J.-W. Kang (2016): *"Intrusion detection system using deep neural network for in-vehicle network security"*

[61] (TC) Kneib and Huth (2018): *"Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks"*
[74] (Ow) Loukas et al. (2017): *"Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning"*
[90] (Os) Müter and Asaj (2011): *"Entropy-Based Anomaly Detection for In-Vehicle Networks"*
[89] (Ow) Murvay and Groza (2014): *"Source identification using signal characteristics in controller area networks"*
[94] (Ys, Yw) Olufowobi et al. (2020): *"SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing"*
[104] (Ys) Rehman et al. (2021): *"CANintelliIDS: Detecting In-Vehicle Intrusion Attacks on a Controller Area Network using CNN and Attention-based GRU"*
[116] (Ys, Yw) Al-Saud et al. (2020): *"An Intelligent Data-Driven Model to Secure Intravehicle Communications Based on Machine Learning"*
[125] (Os) H. M. Song et al. (2016): *"Intrusion Detection System Based on the Analysis of Time Intervals of CAN Messages for In-Vehicle Network"*
[124] (Ys, Yw) H. Song et al. (2020): *"In-vehicle network intrusion detection using deep convolutional neural network"*
[129] (Yw) Tariq et al. (2020): *"CAN-ADF: The controller area network attack detection framework"*
[131] (Os) Taylor et al. (2016): *"Anomaly detection in automobile control network data with long short-term memory networks"*
[143] (TJ) Xiao et al. (2021): *"Reinforcement Learning-Based Physical-Layer Authentication for Controller Area Networks"*

During the study, we assumed that one publication about a new idea for intrusion detection presents and focuses on a single new approach. Hence, we expected the list of 19 publications to result in 19 IDSs for the case study. After we read the full texts of all selected papers, we confirmed that most of the publications presented only a single approach. However, Hoppe et al. [50] present three distinct detection approaches in one publication that we analyzed separately in our study. Consequently, our case study contains of 21 IDSs.

**Exclusion for survey literature:** Side products of the above literature discovery are the publications providing a survey on existing work which we separated in step 5. With the previous exclusion process, a large share of these publications focused on different topics and perspectives that are not relevant to the description and comparison of IDSs. For example, publications analyzed the automotive landscape as a whole or gave an outlook on potential future developments. Therefore, we filtered the publications separated in step 5 further with the following exclusion criteria.

1. Exclude papers that do not provide an abstraction or summary of multiple existing detection approaches via human expert voting. In the voting, each researcher read the title and abstract of the paper and tried to give a boolean answer to the question "Does this paper provide a detailed overview about multiple IDSs, e.g. in more than

one subsection?". Each paper was investigated independently by two researchers. The paper was only excluded if they both with certainty could reply with "no" to the question.

2. Clarify the voting on uncertain candidates from the previous step and exclude papers that do not provide an abstraction or summary of multiple approaches. Both researchers had a look at the full text of the paper and answered the same question as in Step 1 again. In cases of disagreement, they discussed until agreement on the classification.

**Identified survey papers:** This above process yielded the following 22 survey papers passing all exclusion criteria. We used them later for the validation and refinement of our taxonomy.


[10] Aliwa et al. (2021): *"Cyberattacks and Countermeasures for In-Vehicle Networks"*
[18] Berger et al. (2019): *"Comparative study of machine learning methods for in-vehicle intrusion detection"*
[23] Bozdal et al. (2020): *"Evaluation of CAN bus security challenges"*
[36] Dibaei et al. (2020): *"Attacks and defences on intelligent connected vehicles: a survey"*
[37] Dupont et al. (2019): *"A Survey of Network Intrusion Detection Systems for Controller Area Network"*
[40] Gmiden et al. (2019): *"Cryptographic and Intrusion Detection System for automotive CAN bus: Survey and contributions"*
[44] Hafeez et al. (2020): *"State of the Art Survey on Comparison of Physical Fingerprinting-Based Intrusion Detection Techniques for In-Vehicle Security"*
[51] Hu and Luo (2018): *"Review of Secure Communication Approaches for In-Vehicle Network"*
[55] Al-Jarrah et al. (2019): *"Intrusion Detection Systems for Intra-Vehicle Networks: A Review"*
[58] Khatri et al. (2021): *"Security issues with in-vehicle networks, and enhanced countermeasures based on blockchain"*
[59] Kim et al. (2021): *"Cybersecurity for autonomous vehicles: Review of attacks and defense"*
[65] Lamssaggad et al. (2021): *"A Survey on the Current Security Landscape of Intelligent Transportation Systems"*
[72] Lokman et al. (2019): *"Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review"*
[73] Loukas et al. (2019): *"A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles"*
[103] Rajbahadur et al. (2018): *"A Survey of Anomaly Detection for Connected Vehicle Cybersecurity and Safety"*
[111] Rumez et al. (2020): *"An Overview of Automotive Service-Oriented Architectures and Implications for Security Countermeasures"*
[114] Sakiz and Sen (2017): *"A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV"*

[121] Sharma and Kaul (2018): *"A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET Cloud"*
[133] Tomlinson et al. (2018): *"Towards viable intrusion detection methods for the automotive controller area network"*
[142] Wu et al. (2020): *"A survey of intrusion detection for in-vehicle networks"*
[144] Xie et al. (2021): *"Cybersecurity protection on in-vehicle networks for distributed automotive cyber-physical systems: State-of-the-art and future challenges"*
[146] Young et al. (2019): *"Survey of automotive controller area network intrusion detection systems"*

**Linking Literature and the Taxonomy**

The goal of the above literature elicitation is to identify scientific publications with a high relation to the properties enumerated in our taxonomy as a foundation for the analyses (see Section 4.5) and the case study (see Section 4.6). Namely, we use the survey publications to validate and complement the properties in the taxonomy, and we use the new idea publications for a case study and the deduction of suitable prefilters. Both purposes require links between sections within the publications and aspects in the taxonomy. Hence, all the following steps work on the full text of all selected publications. We were inspired to use this type of analysis by Petersen et al. [99] that proposed a similar analysis about the types of publications and their methods.

These links from the taxonomy to the text were especially helpful while clarifying formulations and further refining the characteristics. Whenever we changed a characteristic we could always directly list the parts of all previously annotated publications that cover this characteristic and, if needed, adjust their links according to our changes. Each full text of a publication was investigated by a single researcher. Unclarities and questions were discussed among two researchers and a second researcher investigated samples of the links of the other researcher.

**References from the taxonomy to survey publications** With the survey publications, we aimed to extend and validate the initial taxonomy from the industry experts. Concretely, we created links from the taxonomy to the full text including images and tables of the publication to highlight usages of the corresponding characteristic to differentiate and classify existing IDSs. This spans abstract classification trees, the split of sections in the text as well as summaries given in the full text. We used simple links without further information to enable quick navigation and quantitative analyses. As anticipated, we adjusted and extended our initial taxonomy elicited from the industry experts during the survey analyses. This resulted in various revisits of all affected links to previously investigated publications, whenever we modified the taxonomy.

In detail, we observed two effects: 1) The granularity and grouping of our aspects were not sufficient to depict all information from the survey publications. Hence, we refined their formulations and, when beneficial, split them into multiple aspects covering related, but complementary properties. Overall, this increased the number of 26 aspects from industry by 8 aspects to a total of 34 aspects. 2) While our taxonomy contained an aspect

describing the higher concept mentioned in the publication, it lacked a characteristic that fully matched the elaborated property. Hence, we added new characteristics to the existing aspects and, whenever needed, adjusted the formulations of the other characteristics in the same aspect. Overall, this increased the number of 120 characteristics from industry by 59 to a total of 179 characteristics. Section 4.5.1 analyses the frequency of survey papers mentioning each aspect and characteristic in more depth and highlights properties that were only named by our industry experts. Although we slightly adjusted and extended the wording in the aspects and characteristics after the survey analyses in the mapping process of new idea publications, we did not change their semantics, structure, number or grouping.

**Mapping of new idea publications to the taxonomy** With the new idea publications, we aimed to create a dataset of properties structured by our taxonomy from all their presented IDSs. Concretely, we created links from the taxonomy to the full text of the publication to support each mapping (with `true`, `false`, `maybe` and `n/a`) to each characteristic of the taxonomy with citations. However, the mapping of direct citations from the text is hindered by a discrepancy in granularity between any written text and the detail level of our taxonomy. Academic publications have strict limitations in space, whereas our taxonomy enumerates hundreds of characteristics explicitly. We found that papers do not explicitly elaborate on many of our characteristics in detail but instead present their ideas on a more abstract level and skip obvious properties, especially if they do not apply to the presented approach. This is not surprising as our taxonomy intentionally included various properties from diverging detection approaches. Nevertheless, our prefilters only work optimally with a complete mapping of all candidate approaches to all aspects of the taxonomy and only small unknown characteristics. In theory, we could have quickly clarified all these uncertainties and potential open questions by talking with the IDS developers. However, contacting each of them was not feasible for our work as we used IDSs and literature from numerous and diverse authors.

Therefore, we distinguished two different types of mappings: explicit and implicit. We refer to mappings based on citing a specific part of a paper as explicit mapping because explicit explanations in the text directly provide the classification of the taxonomy. In cases where we did not find the required information elaborated directly, we often found the overall text to be sufficiently descriptive. We could still classify the approach based on our domain knowledge and understanding of the publication. Such mappings we call implicit because they lack explicit support by the paper. Both mappings combined allowed an almost complete mapping of all approaches to all characteristics. In total, our study accumulated 3759 annotations in total, containing 886 explicit annotations (24%) and 2873 implicit annotations (76%). The large share of implicit annotations underlines their necessity.

## 4.4 Our Taxonomy

This section presents our final taxonomy for describing a candidate IDS. It is the result of executing the methodology that we presented in the previous section. The structure of the paragraphs and text follows the meta-model of views, aspects and characteristics introduced in Section 4.3.1.

For references in the following analysis and the ease of discussion, we assigned IDs to each aspect, consisting of a letter denoting the corresponding view (S for system, A for attacker, D for defense) and an increasing number. The numbers align with the order we present the taxonomy in this subsection. Both give the impression that this is the order in which the taxonomy shall be flattened and sorted. We want to point out, that we don't believe there is a single desirable order within the taxonomy. Sorting is only a technical requirement of this linear representation. All aspects as a whole describe a candidate and depict a complex, multi-dimensional space of properties and dependencies. We will investigate different potential sortings of the aspects in Section 4.6 without identifying a preferable order. The order in the following is rather arbitrary and only loosely corresponds to the growth of the taxonomy within the discussions and iterations.

### 4.4.1 System View

**Aspect S1: Layer for attachment of IDS**

**Explanatory question** What is the latest stage of message parsing and interpretation the IDS is attached to?

**Characteristics**

1. Physical, electronic Symbols: Low and high voltages profiles + the time of regular patterns
2. Separated Packages: Information about the distinct packages forming a datagram including transmission errors, partial messages and network overhead + time of start and end of each package
3. Full but raw datagram: Header-Fields (e.g. receiver or sender) and raw payload (i.e. all 0 and 1) with Size + time of full receipt
4. Structured and interpreted datagram: Header-Fields and Payload value i.e. data types (e.g. enums) and values of each field + time of full receipt
5. Content of communication: Semantic Interpretation and time of changes from the payload and without network frame information

**Aspect S2: Occurrence of Messages to Be Protected on the Network in Relation to Changing Information**

**Explanatory question** In what pattern are the messages, that are protected by the IDS, sent to share an information on the monitored network?

**Characteristics**

1. Aperiodic traffic (no fixed schedule): Event-based messages, e.g. at the moment of an information change

1.1. Fixed number of cyclic messages after a specific event
2. Periodic/Cyclic traffic (fixed schedule)
   2.1. Cyclic messages with a fixed schedule, independent of the information carried
   2.2. Cyclic messages with various schedules for different modes (e.g. driving/parking), independent of the information carried

**Aspect S3: Partitioning of Payload**

**Explanatory question** Is a payload to be sent on the network partitioned into multiple messages?

**Characteristics**

1. Partitioned datagram, i.e. oversized payload is sent in multiple messages
2. No partitioning, i.e. payload is sent in a single message

**Aspect S4: Transmitted Payload Type**

**Explanatory question** What types of payload might be used in messages that are protected by the IDS?

**Characteristics**

1. Continuous, physical signals, e.g. values measured by sensors
2. Multi-value fields
3. Unordered values from fixed interval (with jumps), e.g. concrete inputs given by a user of the system
4. Counters, i.e. values starting at an initial value and increasing their value linearly with potential overflows
5. Flags, i.e. boolean values
6. Constants

**Aspect S5: Confirmation of Message Receipt**

**Explanatory question** What message exchange patterns might be used on the application level for the protected communication on the network?

**Characteristics**

1. Fire and Forget: Sender does not check any receipt
2. Fire and Forget with repetition: Senders sends the same message multiple times, but does not check the receipt
3. SYN+ACK: Sender asks for confirmation and the correct receiver acknowledges the receipt
4. Receiver reports an error if the message is not received within the time frame from a schedule

### 4.4.2 Attacker View

**Aspect A1: Location of the Attacker inside the Network**

**Explanatory question** Where is the attacker located inside the attacked network?

**Characteristics**

1. The attacker is attached to the network as an additional, new node
2. The attacker has control over an existing network node
   2.1. The attacker has limited control over an existing network node (e.g. the attacker can only send extra messages, or he controls only a sub-service of the node)
   2.2. The attacker has full control over an existing network node
3. The attacker has control over a gateway node bridging two sub-networks, which is routing messages between the subnetworks
4. Man in the middle: The network is forcefully split by the attacker and the attacker acts as a hidden gateway between both parts

**Aspect A2: Time of the Infection**

**Explanatory question** When is the system infected by the attacker to prepare for the manipulation?

**Characteristics**

1. The monitored system is infected while it is not operating
2. The monitored system is infected as soon as it starts operating
3. The monitored system is infected during the operation of the system

**Aspect A3: Level of Infection**

**Explanatory question** What capabilities and access has an attacker gained before the manipulation during the operation of the protected system?

**Characteristics**

1. Read data: Observing events, values, data sets or the state of the monitored system
2. Read control flow: Read the processes or source code of the monitored system
3. Store data: Create new events or data sets that are processed by the monitored system
4. Start new processes: Create and execute new system logic previously not in the system
5. Modify stored data: Manipulation of existing events or values inside the system
6. Modify control flow: Rewriting or changing parts of the system logic
7. Invalidate data: Prevent events or data sets from full processing but (invalid) data fragments remain in the monitored system
8. Delete data: Remove existing events or data sets completely from the monitored system without any indicator of the suppression

**Aspect A4: Goal of the Attacker**

**Explanatory question** What is the goal of the attacker?

**Characteristics**

1. Attack on Confidentiality
   1.1. Eavesdropping/Sniffing/Spying: Silently read information

1.2. System exploration: Get information about the system
2. Attack on Integrity: Take control by influencing the system
3. Attack on Availability
   3.1. Suspension: Silence a single component
   3.2. Denial of Service: Stop entire system from communicating

**Aspect A5: Strategy of the Attacker**

**Explanatory question** What is the attacker actively doing in the protected network?
**Characteristics**

1. Passive in general
   1.1. Eavesdropping/Sniffing: the attacker is passively listening to the network communication
   1.2. Side-channel attack: Extract information indirectly from the implementation or realization of a system without using a weakness in the design or implementation itself (e.g. through timings, power consumption, electromagnetic effects)
2. Active in general: Perform attacks against the vehicle
   2.1. Frame injection: Insert arbitrary frames
      2.1.1. Send additional diagnoses-/debug-/status-messages, e.g. to obtain information
      2.1.2. Fabrication/spoofing: Insert frames of another ECU
      2.1.2.1. Impersonation/Masquerade: Insert frames of another ECU that has been disabled
      2.1.2.2. Frame falsifying/Malfunction attack: Send wrong payloads
      2.1.2.3. Replay attack
      2.1.2.4. Fuzzing: Enumerate many different input combinations and message patterns, aiming for maximal search space exploration
      2.1.3. Message Flood: Send many arbitrary messages: the same message or pattern of messages might be sent repetitively multiple times
   2.2. Message Drop: Remove valid messages from parts of the network (suspension)
      2.2.1. Error Framing/Bus-off: Suppress communication by invalidation of transmitted messages
   2.3. Bit flip/Modification: Modify the payload in the message that is currently in transmission over the network

**Aspect A6: Stealthiness of Manipulation**

**Explanatory question** Which efforts for stealthiness of the network manipulations can the IDS still detect from a technical point of view?
**Characteristics**

1. Manipulation of the attacker does not violate the physical network properties, e.g. electronic manipulations or short-circuits

2. Fields in the header of the manipulated message match the concrete network, e.g. only the existent sender or receiver
3. Payload of the manipulated messages is syntactically right, e.g. correct byte-sizes and all values in their valid ranges
4. Manipulated message is sent by a legitimate sender of that message type
5. Proper timing of messages, e.g. through the manipulation no message is sent too early, too late, too rarely or too frequently
6. Message does match the context known in the protected system, e.g. the system is in the right state to process the message correctly

**Aspect A7: Sophistication of Manipulation**

**Explanatory question** Which sophistication in the data manipulation can the IDS still detect from a view of the application?

**Characteristics**

1. Through the attack a value reaches an extreme corner case value that only happens very rarely, e.g. value is set to zero or the maximum negative value
2. The manipulated value only slightly differs from the real value provided to or stored in the system
3. The manipulation of a continuous value is built up gradually but increases over multiple consecutive states of the systems
4. The manipulation of a discrete value constitutes valid variable changes and progression
5. Through the manipulation a continuous value in the system jumps suddenly
6. Multiple inputs to or values inside of the system are manipulated in parallel to mimic a plausible system state

### 4.4.3 Defense View

**Aspect D1: Location of the IDS**

**Explanatory question** Where is the IDS physically located?

**Characteristics**

1. Offloaded to remote service (e.g. to a cloud service)
2. Stand-Alone component attached to the system (e.g. OBD-II port)
3. Inside a single component of the system (e.g. single node)
4. Single, but central component (e.g. router, gateway)
5. Spread over multiple components of the system
6. Distributed over all components of the system

**Aspect D2: Data Source**

**Explanatory question** Where does data monitored by the IDS originate?

**Characteristics**

1. Host-based: IDS examines nodes/hosts/operating systems

2. Network-based: IDS examines the traffic on the network

**Aspect D3: Time of Configuration**

**Explanatory question** Does the IDS need to be adjusted after the first deployment?
**Characteristics**

1. Offline preparation: Fixed before deployment by abstract network architecture
2. After deployment: Specific, active adjustments once inside the actual network without real network traffic
3. First usage adaption: Requires a short period of real network workload once to be automatically fitted to the network
4. On the fly: Continuous refinements and adjustments during the usage

**Aspect D4: Number of Additional Packages Involved in Detection**

**Explanatory question** How many additional (previous) packages does the IDS need to determine whether a certain package/field/byte is malicious?
**Characteristics**

1. No additional packages needed (the single package is enough)
2. Sliding window of earlier packages
   2.1. Sliding window of a fixed number of earlier packages
   2.2. Sliding window of packages in a specific earlier timespan

**Aspect D5: Used Type of Information**

**Explanatory question** What type of information does the IDS take as an indicator for the intrusion?
**Characteristics**

1. Details from the concrete realization in software, e.g. stack layout, configuration, log files
2. Physical or electronic properties of the system's hardware (e.g. for fingerprinting)
   2.1. Properties of the electrical signal (e.g. sharpness of electronic signal spikes, voltage distribution)
   2.2. Clock skew/offset
3. Specific parts of the message
   3.1. Whether the message is a diagnostic message
   3.2. Whether the message is a remote frame
   3.3. Whether the message contains an error flag
   3.4. Topic (arbitration) ID
   3.5. Payload
      3.5.1. Whole payload (bits)
      3.5.2. Specific signals encoded inside the payload
4. The whole message+header (bits)
5. Network flow/Cyber features in general

5.1. Order of messages
5.2. Meta-Data not directly represented inside the events of the system, e.g. workload of the system
5.3. Time-/Frequency-related, e.g. Message rate/timing, network latency
5.4. Offset/Interval/Response time of remote frame
6. Information about the business process in the system, e.g. repetitions, message order, X happens before Y
7. Current (real world) state of the vehicle / Physical features of the environment in general (as monitored by system e.g. speed, location, signal courses)

**Aspect D6: Influence on Network Traffic**

**Explanatory question** How does the IDS influence the network traffic?
**Characteristics**
1. No influence (i.e. doesn't send messages)
2. Increases network traffic (e.g. injects remote frames)

**Aspect D7: Used Specification of Communication**

**Explanatory question** What information from a network specification except the network protocol is reused by the IDS?
**Characteristics**
1. Requires some parts of the specification
1.1. Properties of message headers
1.1.1. Valid topic (arbitration) IDs
1.1.2. Valid topic (arbitration) IDs per subnetwork
1.1.3. Valid topic (arbitration) IDs per network node, i.e. a mapping of the topic (arbitration) ID and sender/receiver
1.2. Properties of message payloads
1.2.1. Separation of Payload Fields: the start and end of a signal, if the payload of the transmitted message combines multiple pieces of information
1.2.2. Message length: Explicit limitation to specific numbers of bytes in a valid payload that further restricts the valid lengths in the network protocol
1.2.3. Valid variable range: which range can a valid variable inside the massage take
1.3. Communication pattern
1.3.1. Timing information, e.g. cycles, repetitions
1.3.2. Model (e.g. state machine or physical formula) of signal changes describing information transitions over multiple consecutive messages of the same signal

**Aspect D8: Preprocessing of Audit Features**

**Explanatory question** How are the features used as indicators for the intrusion preprocessed before being used in/with the detection model?

**Characteristics**

1. No preprocessing
2. Normalization (e.g. clamp to lower/upper bound)
3. Aggregation (e.g. average)
4. Distance measure (e.g. Hamming distance)
5. Entropy

**Aspect D9: Type of Learning**

**Explanatory question** What type of learning is used to build the model for the intrusion?

**Characteristics**

1. Supervised learning: Labeled training data is required
2. Unsupervised learning: Training data does not have to be labeled

**Aspect D10: Complexity of Learned Information Represented in the Model**

**Explanatory question** How much training does the model need before successful detection?

**Characteristics**

1. Rule-based: Fixed rule set without learning or adjustments, i.e. no dynamic changes
2. Learning: Learns from system data and adapts behavior
   - 2.1. Statistical-based: Learning adjusts parameters in a fixed system-specific rule set, i.e. adjusts the conditional statements, but the control flow graph remains unchanged
   - 2.2. Addition/Removal of rules and filters to an initial rule set, i.e. changes in the control flow graph, but only with predefined components
   - 2.3. Machine Learning: Complete model build from system traces through general ML-Algorithms, i.e. the definition of normal/intrusion is deduced from the training data

**Aspect D11: Correlation of Intrusion Model to System Specification**

**Explanatory question** To what degree does the model of the intrusion rely on information that can be explicitly specified except the used syntax (e.g. statements or protocols) inside the monitored system?

**Characteristics**

1. IDS uses information that cannot be specified (e.g. specificities of the used hardware)
2. IDS learns information independently that is or could be also part of the specification

3. IDS uses information and variables from the system specification as additional input to its learning phase for further refinements
4. IDS transforms a knowledge base from the specification directly into detection mechanisms

**Aspect D12: Collaboration for the Learned Information**

**Explanatory question** How unaffiliated does a single instance of the IDS build its internal model?

**Characteristics**

1. The model for detection is created externally by an expert or expert system
2. A deployed instance of the IDS creates the model alone and does not share it
3. Several deployed instances of the IDS share information to refine their individual detection model
4. Several deployed instances of the IDS collaborate in the creation of a common detection model
5. A deployed instance of the IDS reports to a central expert system (e.g. a server collecting reports and human experts accessing it) and adjusts its model with the feedback

**Aspect D13: Type of Detection**

**Explanatory question** What audit type is used to detect the intrusion?

**Characteristics**

1. Anomaly-/Behaviour-based: Unknown anomalies by aberration of known regular behavior
2. Signature-/Knowledge-based: Match with a known set of concrete attack signatures

**Aspect D14: Structure of Detector**

**Explanatory question** How is the detector structured internally?

**Characteristics**

1. One single detector for all messages
2. Ensemble of detectors for distinct parts of communication, e.g. one detector per topic (arbitration) ID

**Aspect D15: Abstraction by the Detection Mechanism from Specificities of Individual Systems**

**Explanatory question** How far does the detection mechanism abstract from specificities of instances of the monitored system and the concrete field of usage?

**Characteristics**

1. No system specificities incorporated or relevant, same IDS for each instance of the monitored system and each field of usage

2. Parameters (e.g. weights, intervals, ranges, factors) used for detection are adjusted for each concrete system; the decision logic remains constant per field of usage.
3. Different model (e.g. different paths for decision) of detection used for each concrete instance of the monitored system

**Aspect D16: Granularity of Reported Intrusion**

**Explanatory question** What is the smallest unit transmitted over the network that is spotted as malicious?

**Characteristics**

1. A single bit transmitted over the network, i.e. the information that is interpreted as 0 and 1
2. Multiple bits (no correlation to signals)
3. A single signal inside the transmitted package
4. A single package, i.e. the smallest full message that can be transmitted over the network
5. A complete datagram, i.e. messages split into different packages once they are combined
6. A stream of multiple, consecutive datagrams

**Aspect D17: Detection Latency**

**Explanatory question** How long does it take in the best case from the start of the manipulation to detection?

**Characteristics**

1. Real-time Detection
   1.1. At the first moment: Manipulation attempt is detected before any effect on the monitored system
   1.2. With the first effects: Detection during the attack, but after a first manipulation of the system state
2. Post Mortem Analysis: Intrusion is detected at a later point in time after the attack has taken place

**Aspect D18: Certainty Information Included in Alarm**

**Explanatory question** What information about the certainty of the detection is included in an alarm?

**Characteristics**

1. Boolean: Continuous information about "Alarm" or "No Alarm"
2. Probability between 0 and 1 (inclusive): Continuous stream of certainty scores about the suspicious of the observed behavior
3. Human Readable Report: Background Info and annotations of a time stream helpful for incidence management

**Aspect D19: Attack Details Included in Alarm**

**Explanatory question** What information about the attack is included in an alarm?
**Characteristics**

1. Classification of alarm, e.g. a denial of service is ongoing
2. Spot malicious information: Identify which data is tampered
3. Localization of attacker: Which nodes/parts of the network are infected
4. Root cause: Insights about the attack vector used for the infection

**Aspect D20: Response Mechanism**

**Explanatory question** What is intended to happen after an alarm has been raised to mitigate the attack?
**Characteristics**

1. Exemplary sketch of potential reactions (e.g. "sending a warning to the user" without further details)
2. Reference to a suitable and evaluated reaction (e.g. a link to academic work, documentation of experiments)
3. Indicator for an appropriate reaction implied by the detection mechanism (e.g. "ignoring all suspicious events before processing")

**Aspect D21: Required Updates on System Changes**

**Explanatory question** In case the monitored network changes (e.g. Software Updates, Replacement of a node), when does the internal model in the IDS need to be adjusted?
**Characteristics**

1. In case some hardware in the network is replaced (e.g. replacing a defective network node with new hardware)
2. If the payload of the protected messages changes (e.g. adding new information to the package or adjusting variable sizes/ranges)
3. If the concrete deployment of the abstract network plan is changed (e.g. relocation of a sender process to another network node)
4. If the location the system is deployed in changes (e.g. migration of the whole network to a new environment)

**Aspect D22: Robustness of the IDS**

**Explanatory question** How is the robustness of the IDS against adjusted attacks analyzed?
**Characteristics**

1. Discussion of specific cases or system behavior that is hard to distinguish from an intrusion
2. Discussion of weaknesses and potentials of abuse of the reaction triggered by an alarm of the IDS

3. Discussion of potential strategies to hide an attack from the IDS
4. Distinct Security and Risk Assessment of the IDS implementation

## 4.5 Analysis of the Taxonomy and Mapping

The previously listed taxonomy forms the backbone for our prefilters by providing their data model. However, as elaborated along with our methodology in Section 4.3, it is only the first step of the research presented in this chapter. This section focuses on a deep analysis of the data we extracted during the literature study. Namely, we linked two types of publications to our taxonomy: 1) survey papers presenting abstract views on the landscape of intrusion detection approaches to validate our taxonomy, and 2) new idea publications presenting novel detection approaches in detail. The new idea publications are of higher importance in this chapter, as with their information we created the dataset that we used for deducing and evaluating the prefilters (presented in Section 4.6). This dataset consists of the full mappings of each presented IDS in the new idea publication to all characteristics of the taxonomy. In other words, it is the formalized description of which abstract properties listed in the taxonomy apply to each candidate IDS. Such a feature matrix is the main input for each prefilter.

To foster the deduction of the prefilters, this section investigates different perspectives on the collected data. Section 4.5.1 opposes the industry perspective with the perspective of the academic survey publications. Section 4.5.2 discusses insights from the unaggregated mapping of the candidate IDSs to the characteristics in the taxonomy. Section 4.5.3 quantifies the discriminatory power of each characteristic on the candidate IDSs via its entropy. Finally, Section 4.5.4 investigates the similarity between all candidate IDS with data mining techniques. Please note, that each subsection in this section focuses on analyses related to its individual view, whereas the discussion in Section 4.7 aims to unite multiple views and the experiences from the case study for more holistic insights.

### 4.5.1 Common Properties in our Taxonomy and Other Surveys

The full and final taxonomy is the result of following a multi-step methodology combining the views of industry experts and various academic surveys. Of course, these views overlap, especially within the large corpus of publications we analyzed. However, there are different frequencies of overlaps and some aspects or specific characteristics were only named once. Table 4.1 lists each aspect and measures the coverage of the contained characteristic by each investigated source.

In detail, the first two columns denote the number of characteristics of this aspect in the taxonomy after the industry expert interviews (#industry) and in the final version (#final). Refinements and complementation of the industry perspective through the academic literature have increased the number of characteristics and aspects. These changes are highlighted in the #final column, whereas unchanged aspects remain without highlighting. Aspects that were newly added during the literature study have no value for #industry. Please note, that these numbers are only an approximation for all the changes and refinements through the literature study. For example, they are oblivious to changes

in the formulation and order or hierarchy of the characteristics. Most prominently, aspect S3 originates from a refactoring of the industry version and was not covered in the literature. Hence, such aspects denote a value in the #final column but not in the #industry column. The remaining columns each denote the number of characteristics that have been discussed in the survey papers from the literature pool. Their color coding denotes the ratio of covered characteristics, namely a strong color indicates full coverage, while no highlighting indicates that this aspect was not considered by the publication. In total, this overview showcases a varying coverage of our taxonomy by the survey publications. Most prominently, there are survey publications we identified in the literature selection that do not structure intrusion detection approaches for the CAN bus further in their full texts. For example Rajbahadur et al. [103] investigate the security of vehicles while communicating with their environment that is only slightly related to the vehicle internal CAN bus, or Sakiz and Sen [114] that do not mention CAN bus at all.

Regarding the coverage of aspects from our taxonomy in academic publications, we found that the majority of aspects are common in many publications and the industry perspective, especially in surveys that are more focused on comparing concrete IDSs of our use case. However, this visualization showcases remarkable patterns. Some aspects are very frequently used and well-covered in the majority of publications. They consider more generic concepts, e.g. D13 (Type of Detection), A4 (Goal of the attacker), D2 (Type of Data) or D9 (Type of Learning). These aspects provide a general orientation in the field of security, but they highlight a few high-level concepts (most often encoded by two characteristics) denoting families of detection mechanisms but without detailed differentiation within them. Some other aspects are commonly used in literature but differ drastically in the provided level of detail between individual publications. They enumerate concrete technical details, e.g. A5 (Strategy of the Attacker), D5 (Used Type of Information), or D7 (Used Specification of Communication), which can be discussed on various abstraction levels. In our taxonomy, these aspects consequently contain over a dozen individual characteristics. These aspects provide fine differentiation among candidates but often exceed the level of detail provided in academic publications.

Most importantly, our taxonomy contains aspects that are not used by survey publications to structure detection approaches. They relate to the perspectives of our industry experts but have not been considered yet in the academic discourse. In short, our industry experts consider the entire process of deployment and maintenance in their decision process. They know about potentially problematic specificities of their system, reflected in S3 (Partitioning of Payload) and S5 (Confirmation of Message Receipt). They analyze the full, stepwise intrusion of the attacker, reflected in A2 (Time of Infection) and A3 (Level of Infection), to consider IDSs beyond their responsibility and interrelation with other means of defense. Although researchers consider it a different field, industry experts require a response to an ongoing attack (reflected in D20 (Response Mechanism)). They consider all artifacts as continuous, eventually, the system receives updates, reflected in D21 (Required Updates on System Change), or attackers might learn about the existence of the IDS and adapt their attacks, reflected in A6 (Stealthiness of Manipulation), A7 (Sophistication of Manipulation), and D22 (Robustness of the IDS). Overall, these are

| | #industry | #final | [55] | [10] | [59] | [142] | [73] | [72] | [133] | [51] | [146] | [37] | [144] | [58] | [23] | [111] | [18] | [36] | [40] | [121] | [44] | [65] | [103] | [114] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S1 | 5 | 5 | | 2 | | 1 | 2 | | | 1 | | | 1 | | | | | | | | | | | |
| S2 | 5 | 5 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | | | 1 | | | |
| S3 | | 2 | | | | | | | | | | | | | | | | | | | | | | |
| S4 | 4 | 6 | 4 | | | | | | 1 | | | | | | | | | | | | | | | |
| S5 | 4 | 4 | | | | | | | | | | | | | | | | | | | | | | |
| A1 | 5 | 6 | | 2 | 1 | 2 | | 2 | 1 | 1 | | | | 3 | 3 | | 1 | 1 | | | | | | |
| A2 | 3 | 3 | | | | | | | | | | | | | | | | | | | | | | |
| A3 | 8 | 8 | | | | | | | | | | | 2 | | | | | | | | | | | |
| A4 | | 7 | 4 | 1 | 5 | 3 | 6 | 2 | 2 | 3 | 2 | 3 | 1 | 4 | 3 | | 2 | 1 | | | | 4 | | |
| A5 | 10 | 15 | 12 | 11 | 12 | 9 | 11 | 11 | 8 | 4 | 7 | 7 | 6 | 7 | 7 | | 3 | 3 | | 2 | 3 | 2 | | |
| A6 | 6 | 6 | | | | | | | | | | | | | | | | | | | | | | |
| A7 | 6 | 6 | | | | | | | | | | | | | | | | | | | | | | |
| D1 | 5 | 6 | 1 | 3 | 2 | 1 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | | 1 | 1 | | | 1 | 2 | | | | |
| D2 | | 2 | | 2 | | 2 | | 2 | | | 2 | | | 2 | | 2 | | | 2 | | | | | |
| D3 | 4 | 4 | 2 | 2 | 1 | 1 | | 1 | | 1 | | | 1 | | | | | | | | | | | |
| D4 | | 4 | 3 | 3 | 2 | 1 | 1 | 1 | 2 | | | 3 | 1 | 2 | | | | | | | | | | |
| D5 | 5 | 20 | 12 | 12 | 10 | 8 | 13 | 8 | 7 | 8 | 4 | 9 | 4 | 6 | 9 | 7 | 6 | 2 | 2 | 2 | 3 | | | |
| D6 | | 2 | | | | | | | | | | 1 | | | 1 | | | | | | | | | |
| D7 | 6 | 12 | 2 | 4 | | | 3 | 1 | 1 | 1 | 1 | 5 | 4 | | 1 | 1 | 1 | | | | | | | |
| D8 | | 5 | 3 | 3 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | | 1 | | 1 | 1 | | | | | |
| D9 | | 2 | 1 | 2 | | 2 | | 2 | | | 1 | | 2 | 2 | | | | | | 1 | 1 | | | |
| D10 | 4 | 5 | 4 | 3 | 3 | 2 | 4 | 3 | 3 | 3 | 1 | 2 | 2 | 3 | 1 | 2 | 2 | | | 2 | 1 | 1 | | |
| D11 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | | | 1 | | | 1 | | | | | | | | | | |
| D12 | 5 | 5 | | | | | 1 | | | 1 | | | | | | | | | | | | | | |
| D13 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 2 | 2 | 1 | 2 | 2 | 2 | | | 1 | |
| D14 | | 2 | 1 | 1 | 1 | | | | 1 | | 1 | 1 | | | | | | | | | | | | |
| D15 | 3 | 3 | 1 | | | | | | | | | | | | | | | | | | | | | |
| D16 | 5 | 6 | 1 | 1 | 1 | 1 | 2 | | 1 | | 1 | 1 | | | | | | | | | | | | |
| D17 | 3 | 4 | 2 | 1 | 1 | 1 | | | 1 | 2 | 1 | 2 | | | | | | | 1 | 1 | | | | |
| D18 | 3 | 3 | 1 | | 1 | | 1 | | | | | | | | | | | | | | | | | |
| D19 | 4 | 4 | 2 | | 1 | | 2 | 1 | 2 | 1 | | | 1 | | | | | | | | 1 | | | |
| D20 | 3 | 3 | | | | | | | | | | | | | | | | | | | | | | |
| D21 | 4 | 4 | | | | | | | | | | | | | | | | | | | | | | |
| D22 | 4 | 4 | | | | | | | | | | | | | | | | | | | | | | |

**Table 4.1:** *Amount of unique characteristics annotated per survey paper (columns), for each aspect (row). Entries with the value 0 are omitted. The values are highlighted in shades of green, from white (zero characteristics annotated) to green (all characteristics of that aspect annotated). The total number of characteristics per aspect of the initial/refined taxonomy is listed in the second/third column, respectively. A changed amount is highlighted in light blue. Sorted from left to right by accending numbers of covered characteristics.*

indications that the detection capabilities of an IDS are not the only relevant property of an IDS, but their entire lifecycle matters.

Finally, we also found aspects that were subject to big refinements during the literature study. Academic publications denote characteristics related to these aspects in more detail and diversity than our industry experts did, e.g. A5 (Strategy of Attacker), D5 (Used Type of Information), or D7 (Used Specification of Communication). We assume, that this originates from the fact that our industry experts do not develop or configure an IDS themselves on this low, technical level and are indifferent to these finer differentiations. We used nested characteristics to comply with these differing granularities.

Although these analyses of diverging perspectives in industry and academia are merely a side product of our methodology, they relate to a broader problem in this research domain. Newer ideas for intrusion detection are only very slowly adopted by industry and used in productive systems (e.g. the usage of machine-learning algorithms [123]). The above denoted divergences and in particular the missing information might explain the lack of adoption of academic prototypes for productive systems in the industry. Fostering an investigation of the broader properties relevant for the entire lifetime of systems beyond the technical details of the detection could increase the relevance of these systems for industry usage.

### 4.5.2 Mapping of the Selected Candidate IDSs

The key dataset for the prefilters is the mapping of all candidate IDSs to the properties enlisted in our taxonomy. Due to the flat structure of our taxonomy, this dataset is a two-dimensional matrix. The first dimension is a list of all characteristics in the taxonomy. The information from the related aspects and views in the taxonomy only serves as a support for navigation and is omitted in the dataset. The second dimension is a list of all considered IDS candidates. We name and identify the candidate IDSs after their original publication and use their reference as the index, e.g. [50] for the publication of Hoppe et al. In case one publication proposes multiple candidates, we used an incremental counter for them. Each entry in the matrix denotes the mapping of the candidate to the characteristic, namely the label of the corresponding like with a value of `true`, `false`, `maybe` or `n/a`. Other values are not possible and there is no missing value in the matrix.

For all following analyses and the prefilters, this dataset consists of the 21 IDSs selected in the literature study. Combined with the 179 characteristics in the taxonomy, this results in 3759 entries in the matrix. Table 4.2 shows an excerpt from the overall dataset. As the full dataset presented in the same way spans multiple pages, we have put it in Appendix A for the interested reader. The tables there list each IDS we extracted from the new idea papers and provide the mapping to all aspects in full detail. Despite its length and complexity, the full, raw matrix gives a good, first intuition of our mapping and some particular entries already provide valuable insights. However, we see it as not necessary for the understanding to elaborate on this mapping in detail. In the following, we will only highlight some prominent correlations of the full mapping.

The meta-model (see Section 4.3.1) only structures the aspects into three views: the system, attacker and defense views. The distribution of aspects and characteristics among

| | | [30] | [31] | [32] | [42] | [47] | [50](1) | [50](2) | [50](3) | [56] | [61] | [74] | [90] | [89] | [94] | [104] | [116] | [124] | [125] | [129] | [131] | [143] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S1 | 1 | X | ✓ | ✓ | X | X | X | X | ✓ | X | ✓ | X | X | ✓ | X | X | X | X | X | X | X | ✓ |
| | 2 | ✓ | X | X | ✓ | X | ✓ | ✓ | X | ✓ | ? | X | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? |
| | 3 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 4 | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 5 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

**Table 4.2:** *Excerpt of the mapping-dataset for aspect S1. The columns denote all IDS candidates and the rows denote each characteristic. The entries represent the mapping, namely ✓ in green for* `true`*, X in red for* `false`*, ? in orange for* `maybe`*, and* $^{n}/_{a}$ *in white for* `n/a`*.*

them grew unguided during the expert interviews and literature study. From the 34 aspects, the final taxonomy links 5 aspects to the system view (14.7%), 9 aspects to the attacker view (26.5%), and 22 aspects to the defense view (64.7%). From the 179 characteristics, the final taxonomy contains 22 characteristics in the system view (12.3%), 51 characteristics in the attacker view (28.5%), and 106 characteristics in the defense view (59.2%). The share of each view is roughly equal when considering aspects or characteristics. Overall, the defense view contains by far the largest diversity of properties of an IDS. The system view is the least represented, but still relevant for a holistic view.

The meta-model allows four different classifications in the mapping of candidates: `true`, `false`, `maybe` and `n/a`. While `true` and `false` are the expected, dominant mapping types, `maybe` and `n/a` are meant to be exceptions. In our mapping, we only needed the `n/a` for aspect A3 (Level of Infection)—notably for all characteristics and candidate IDSs. As already elaborated in Section 4.5.1 for the data from the survey publications, this aspect is a property solely stated by our industry experts. None of the candidate IDSs provided any further information about the infection and just assumed the attacker to exist in the system with the denoted capabilities. In this situation, the `n/a` was the best choice to disable this aspect completely for the prefilters. Consequently, we also decided to not consider this aspect in the following analysis.

The distribution of `maybe` mappings in our dataset is remarkable too. There is no candidate IDS where the publication provided and investigated all properties of the taxonomy with full certainty. This might have multiple origins. For example, the more limited space in some publication venues might have required skipping properties that are not vital for the approach used for detection and not prominent in the academic discourse. The frequency of `maybe` correlates with the less frequently covered aspects in survey publications (see Section 4.5.1, e.g. A2 time of infection or D15: Abstraction by the detection mechanism from specificities of individual systems). As a decision for our study design, we also mapped characteristics conservatively if they were not explicitly mentioned in the publication. In particular, we stayed hesitated with mapping a `true` or `false`, if properties were not explicit which are critical for the security of the IDS, e.g A1 (The Location of Attacker), A5 (Strategy of the Attacker) or A7 (Sophistication of manipulations). The

IDS of K.-T. Cho and Shin [31] has the highest number of *maybe*-mapping in our data. Although we identified this paper in our literature selection, their publication is not meant to present an IDS in the classical sense. They present an approach for hardware fingerprinting to distinguish malicious and benign messages and identify their senders. While we believe this is a valid approach for intrusion detection according to our definition of an IDS (see Section 1.1), this might not have been the intention of the original authors. Consequently, the publication was missing the most information in our dataset as it understandably was the most immature IDS.

Interestingly, there are 23 characteristics where all candidate IDSs except one share the same mapping. Such characteristics are unique selling points of the corresponding IDS in the investigated set of candidates. An example of such a characteristic is the usage of the communication pattern from the manufacturer's specification by K.-T. Cho and Shin [31]. These characteristics spread around various aspects in the taxonomy and different IDSs from the candidate set. However, they did not result in IDSs that are outstanding compared to the other (as we will see later in Section 4.5.4) and are not considered important differences by our industry experts (see Section 4.6.3). Most likely, the broad coverage of our taxonomy and its relatively large number of characteristics increased the likelihood for these particular attributes to exist.

With looking at the mapping of all candidate IDSs next to each other, the mapping differs hugely from a random distribution and assignment. We got the impression of frequently repeating patterns in the individual properties in this visualization and more prominently during the mapping. These patterns and similarities between candidates occur naturally and originate from identical design decisions or ideas for detection. Nevertheless, without any aggregation, we could not elicit or reason about these patterns further. Therefore, the following subsections use different techniques for accumulation to shed more light on these similarities in the candidates.

### 4.5.3 Entropy of the Aspects and Characteristics

Our methodology fosters a taxonomy that contains all properties that describe any candidate IDS for the use case. We aimed to cover a broad spectrum of different views from industry and academia on the detection algorithm. However, this collection does not consider the suitability of the listed properties to distinguish one IDS from the others in the set. We intentionally made this decision, as we could not define or agree on the notion of suitable or unsuitable characteristics upfront. The taxonomy in isolation is, hence, mostly a structured enumeration of equally relevant properties.

However, there is an intuitive understanding of suitability characteristics in combination with a full mapping to all candidate IDSs as presented in Section 4.5.2: Common properties that are identical for all candidates are ineffectual in their differentiation. Similarly, very specific properties that are unique for a single candidate do not contribute much, as all other approaches are still indifferent regarding this property. Ideally, a quick differentiation for the entire set is driven by properties that are common for half of the approaches and differ from the other half. In this constellation, deciding in favor or against such properties equally filters the candidates the most efficiently, i.e., it reliably excludes half of the candidates at once with a single decision. Please note, that this notion is use-

case specific and only relative to the considered candidates. It is no indication of the universal suitability of a characteristic in other use cases.

In information theory, the above-described intuition is called entropy, first formalized by Shannon [118]. His formulas can be tailored to a characteristic within our taxonomy for a given mapping as follows:

$$H(c) = -p(c = \texttt{true}) \cdot log_2(p(c = \texttt{true})) - p(c = \texttt{false}) \cdot log_2(p(c = \texttt{false}))$$

$$p(c = \texttt{true}) = \frac{|\text{candidates with } c = \texttt{true} \text{ or } c = \texttt{maybe}|}{|\text{all candidates}|}$$

$$p(c = \texttt{false}) = \frac{|\text{candidates with } c = \texttt{false} \text{ or } c = \texttt{maybe}|}{|\text{all candidates}|}$$

The biggest difference to the classical definition of the Shannon-Entropy is that it does not only consider and quantifies the distinct symbols `true` and `false`. The mapping also contains `maybe`, which is either one of the two symbols, but with no information about which. Hence, this third symbol adds further uncertainty observed in the sequence of symbols. The adjusted probabilities considering `maybe` for the probability of both outcomes encode this increased uncertainty. This entropy (same as the classical Shannon-Entropy) links a numeric value between 0 and 1 (inclusive) to each characteristic. An entropy value close to 1 denotes a high discriminatory power, i.e. a choice in favor or against this characteristic splits the set of candidates in half by a single decision. An entropy value close to 0 denotes common and thereby non-discriminatory characteristics, i.e. any choice either results in zero candidates or an almost unchanged set of candidates.

While this formula defines the entropy of a characteristic, we see multiple suitable ways to generalize this concept to an aspect. Intuitively, the entropy of an aspect should be determined via all the characteristics it contains. However, the varying number of characteristics per aspect and drastic differences in their values make such generalizations more complex. Hence, we decided against a singular metric and investigated the raw distribution of characteristic entropies within each aspect. Table 4.3 depicts this entropy distribution for each aspect of our taxonomy on the candidate set of our case study. The black horizontal lines show the range between the minimal and maximal observed entropy of the characteristics in this aspect. The vertical marks denote concrete observed entropy values with the numbers above counting the characteristics with this entropy value. For example, of the three characteristics in aspect A2 (time of infection) two characteristics have an observed entropy of 0.38 and one characteristic has an entropy of 0.35. An interesting attribute of an aspect is the maximum or minimum of all its characteristics' entropy. An aspect with a low maximum is in general less discriminatory as even its best characteristic only describes a common principle among the candidates. An aspect with a high minimum is in general highly discriminatory as even its worst characteristics still exclude a large share of candidates. Both values can be almost identical in the case of an aspect with two opposing characteristics (e.g. D13 (type of detection)) has mostly identical values for both. This effect can also occur only within a subset of characteristics in an aspect when the characteristics refer to very similar concepts or opposing properties. In the following, we will have a closer look at these values in our dataset.

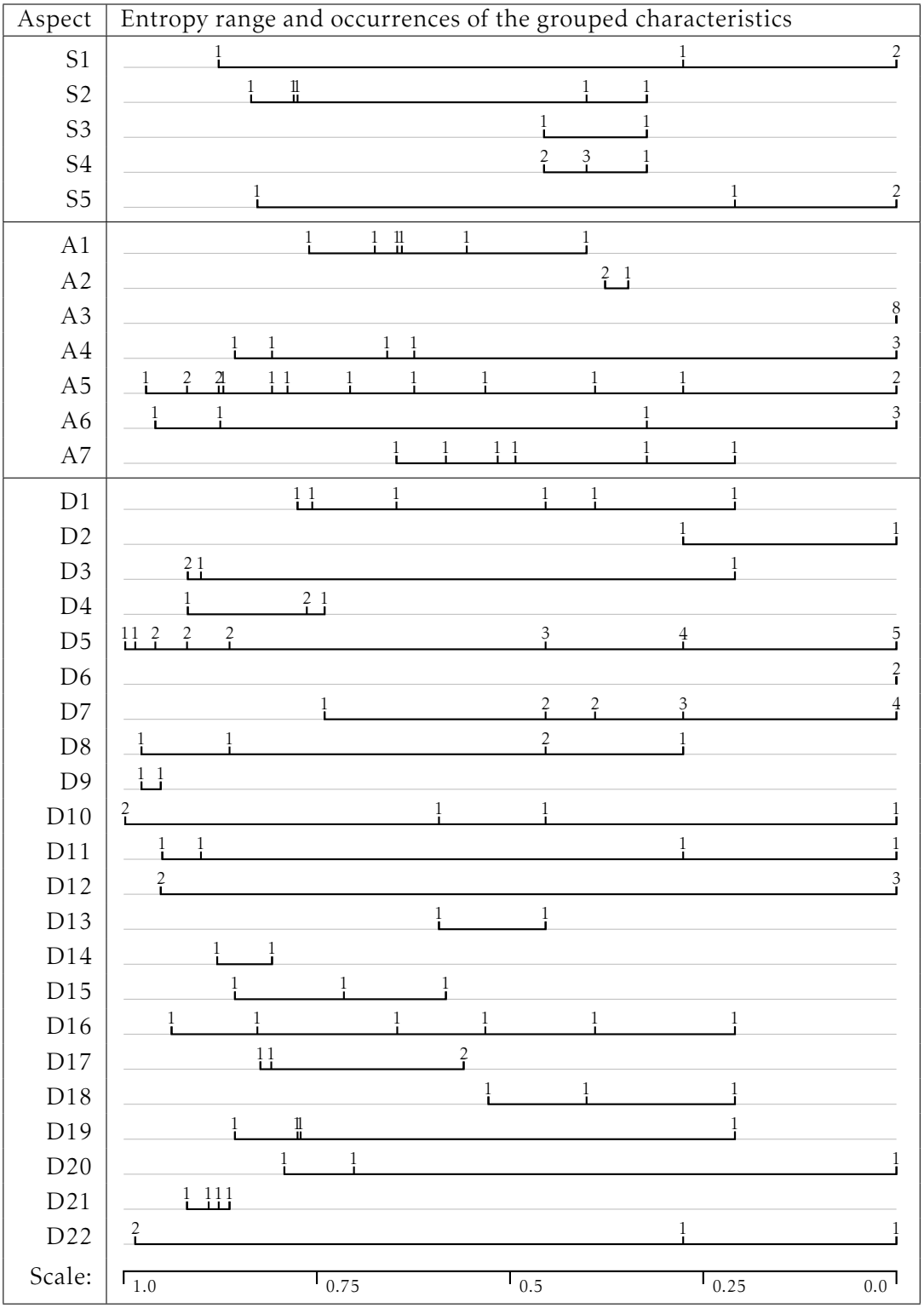


**Table 4.3:** *The entropy of all characteristics grouped by aspect. The lines of all aspects are aligned and show an entropy of 1 on the left to an entropy of 0 on the right. Each dash denotes an observed characteristic entropy value and the digit on the line counts the characteristics with this entropy.*

The entropy within the majority of the aspects spans a very broad spectrum. In detail, 24 aspects (71% of the taxonomy) contain at least one characteristic with an entropy above 0.75; 15 aspects (44% of the taxonomy) contain at least one characteristic with entropy 0; 11 aspects(32% of the taxonomy) have both. Only 8 aspects (23% of the taxonomy) have a difference of maximum and minimum lower than 0.25. This indicates the grouping via aspects does not correlate with the entropy of the contained properties. In other words, the higher abstraction of the aspects puts logical groups but barely partitions characteristics with similar ranges of entropy together.

However, this visualization highlights some aspects that stand out from the others on both ends of the spectrum. The aspects with the lower entropy were already outstanding in Section 4.5.1 and Section 4.5.2. For example, A3 (Level of Infection) and D6 (Influence on the Network Traffic) have an entropy of 0. A3 is fully mapped with `n/a` as it is the only aspect where required information was not present in any publication. D6 is the only other property that was always mapped identically: All candidate IDS do not send any messages. The entropy also reveals aspects with a rather low mapping diversity among our characteristics. The low entropy of some aspects, e.g. S3 (Partitioning of Payload) or A2 (Time of Infection), originates from their general absence of explicit discussion in academic literature, leaving their mapping at rather generic default values and assumptions. Also, aspects that describe common concepts and general families, e.g. S2 (Occurrence of Messages), D2 (Data Source), or D13 (Type of Detection), show a relatively low entropy. On the higher end, our taxonomy also contains aspects that reliably split our candidate set into equal parts. For example, D9 (Type of Learning), D21 (Required updates on System Updates), D4 (Number of additional packages involved), or D14 (Structure of the Detector). We did not anticipate these aspects to be the most discriminatory, especially since they are also neglected in the majority of surveys (see also Table 4.1). One possible explanation could be, that these aspects encode multiple complementary, binary decisions in the algorithm design that align well with our IDS selection.

### 4.5.4 Similarity Between Candidate IDSs

All previous analyses investigated the mapping of IDS candidates to the taxonomy structured and grouped by individual characteristics. The following analysis constrasts that structure and groups the mapping dataset by individual IDSs. We aim to identify IDSs that share an identical mapping to the majority of the taxonomy and only differ in a small number of characteristics. Ideally, such similar IDSs form larger groups that are comprehensible. This would enable us to describe common families of detection mechanisms based on their common properties. Describing these families provides a higher understanding of all available candidates than any discussion of individual characteristics.

Abstractly formulated, the mappings describe a feature matrix linking all the candidate IDSs as individuals to all the characteristics as features through categorical values. Grouping similar candidates based on such data format is a well-known problem in the field of data mining [8]. As such matrix (for example, the dataset introduced in Section 4.5.2) is in general rather small in comparison to other data mining datasets, we used a simple approach: We transferred the categorical data to a numerical scale (0 = `false`, 1 = `maybe`,

2 = `true`). Characteristics mapped with `n/a` are excluded from the mapping distance calculation. Using this encoding, we statistically compared all candidate IDSs with all others in pairs of two. Table 4.4 lists each candidate IDS as rows and columns and depicts the Pearson Correlation Coefficient [68] of each pair as color-coded cells. With this correlation score, a pair of similar candidates shows a positive value close to 1 while a pair of diverging approaches shows a high negative close to −1. The table is symmetric while the diagonal opposes each candidate with itself and consequently denotes perfect matches. We used Morpheus[1] to calculate the coefficient and generate this table.

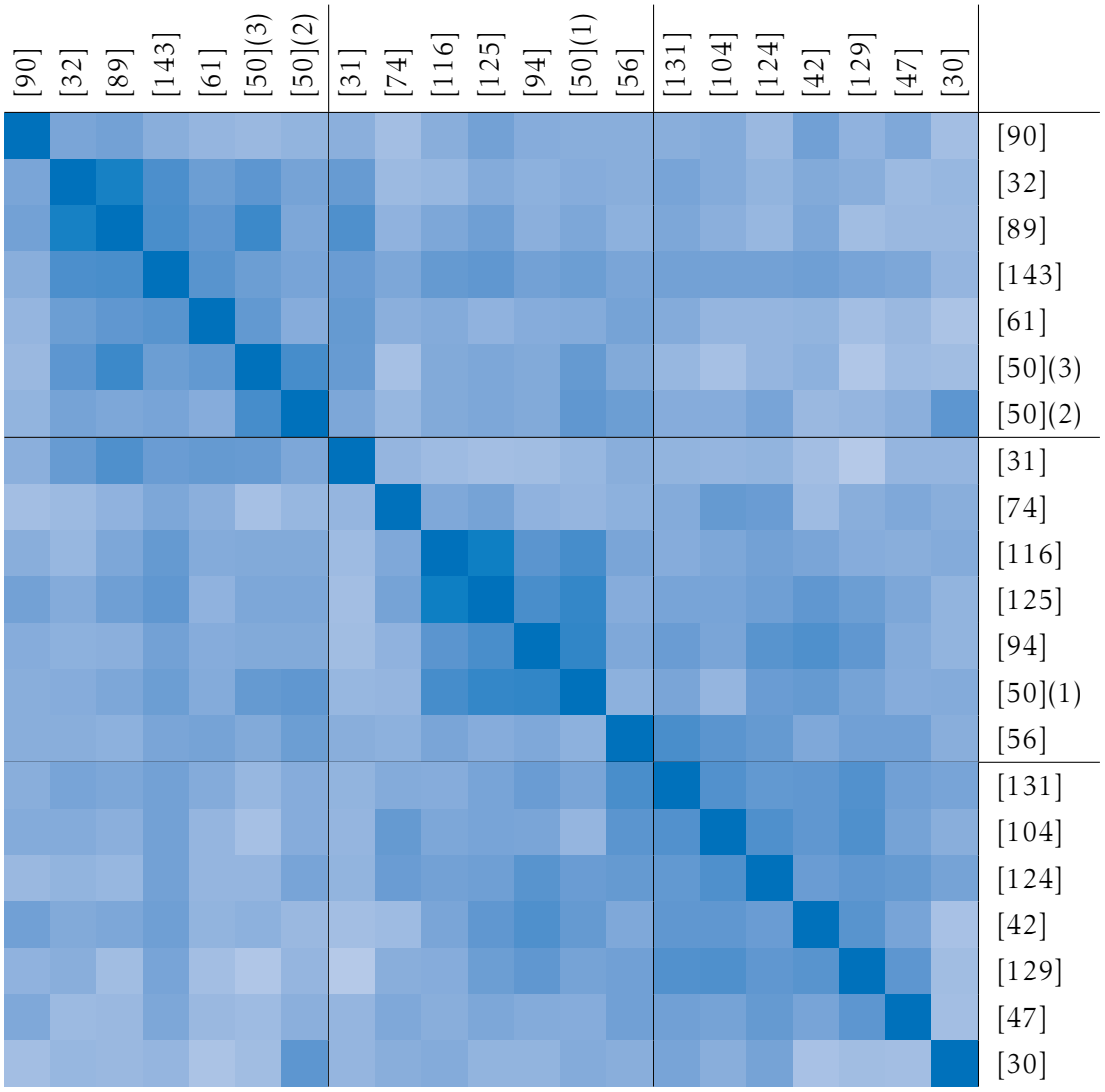


**Table 4.4:** *The similarity matrix showing the Pearson Correlation Coefficient for all pairs of candidate IDSs. The color ranges from white (no similarity) to blue (identical). Manually ordered to put more similar candidates next to each other. The lines split the table into three parts and frame clusters of stronger related IDSs.*

[1] `https://software.broadinstitute.org/morpheus`

In our mapping, the clustering revealed three rough clusters, that we could label manually after looking at the candidates: 1) Low-level approaches focusing on physical signal analysis and hardware fingerprinting (located in the first third of the diagonal), 2) approaches that recognize patterns in the arrival times and frequencies (located in the middle third of the diagonal), and 3) approaches that use computational-intensive machine-learning-based models on the network payload (located in the last third of the diagonal). The analyzation also indicates outliers like, for example, the IDS from Loukas et al. [74] that has the lowest maximal correlation score to all candidates but itself (i.e. in the table the darkest cell in its row/column has the highest brightness compared to the other darkest cells per row/column). This IDS is the only approach in our candidate set that uses a cloud-based architecture.

## 4.6 Prefilters and Case Study

In the following, we present our ideas for four different prefilters and evaluate them in a case study with industry experts. Each prefilter provides the same functionality: It compares all candidates via the abstract properties of the taxonomy and selects "better" candidates relative to the requirements of the domain experts. It operates with a complete mapping of all candidate IDSs to all properties of the taxonomy as we presented and discussed in Section 4.5.2. As a result, a prefilter yields a smaller set of candidates which are promising for further analysis with the standard evaluation schema. The elicitation of a suitable notion of "better" candidates requires active interaction with the domain experts who are responsible for deciding which IDS to investigate and deploy. A good prefilter quickly and with minimal effort converges to a set of candidates smaller than the maximum permitted number. Furthermore, it seamlessly supports the domain experts according to their intuitions and processes. The following case study assesses these properties during real-world usage based on the opinions and experience of our industry experts. For the internal comparison and interaction with the domain expert, each prefilter utilizes a different approach inspired by the previous analysis of the mapping in Section 4.5. We want to remark, that this validation with only such a small group of industry experts is not representative. Nevertheless, it provides at least an anecdotal validation and still contains valuable insides about the utility and problems of each prefilter. More details about the impacts of the small group size are discussed in threats to external validity in Section 4.7.2.

The presentation of each prefilter in the following subsections follows the same structure for discussion and assessment: *Procedure* describes the implementation and usage of the prefilter to identify promising IDS candidates. *Case Study* elaborates on our experiences from suggesting the prefilter to our industry partners, namely four domain experts from different departments collaborating to secure the CAN bus with intrusion detection. Due to a necessary non-disclosure agreement for this research project, we cannot provide full details but will sketch the experiences during the application. Finally, *Advantages* and *Disadvantages* summarize the positive and negative traits of this prefilter. Our experience shows that no prefilter is superior or inferior to the others. The concrete choice for a

prefilter depends on the requirements of the use case and progression in the evaluation and selection process. As we discuss in Section 4.6.5 combinations of prefilters are possible and can be beneficial. In the following, we present the ideas in the order we have introduced and evaluated them with our industry experts.

### 4.6.1 Prefilter 1: Assign Priorities and Penalties

Similar to the mapping linking candidate IDSs to the characteristics of the taxonomy, the requirements and wishes of the system owners for their ideal IDS can be expressed in such linking to characteristics with numeric values. As there most likely is no candidate IDS fulfilling all wishes, we propose a simple scoring system to rank the candidates according to their coverage.

#### Procedure

The domain experts discuss each characteristic in the taxonomy and assign a numeric value to each according to their requirements. With the value, they denote their wish about how desirable (positive values), disadvantageous (negative values) or irrelevant (weight zero) the corresponding characteristic for their ideal IDS is. Higher absolute values represent higher priorities, and values closer to zero represent less importance. There is no logical restriction in the values, although the domain experts might define a scale helping with relative prioritization between characteristics.

After this priority assignment, the prefilter calculates a ranking of all candidate IDSs based on a simple algorithm. The idea of the algorithm is to combine the priority value of the domain experts with the mapping of a candidate IDS on the same characteristics and sum up the result to a final score for the candidate IDS. In detail, the score calculation is rewarding a candidate IDS having desired properties (adding their *positive priority value* to the score) or not having undesired properties (subtracting their *negative priority value* to the score) while punishing a candidate IDS having undesired properties (adding their *negative priority value* to the score) or not having desired properties (subtracting their *positive priority value* to the score). More precisely, the score of a candidate IDS is calculated following this pseudo-code, where $mapping : Characteristic \times IDS \rightarrow Label$ yields the mapping of a candidate IDS and $prio : Characteristic \rightarrow numeric$ yields the priority value of the domain experts.

Positive values for the score $score_{IDS}$ denote a fitting approach, whereas negative values denote an unsuitable approach. The higher the absolute value of the number is, the bigger the match (or discrepancy) from the rated candidate IDS with the ideal IDS of the domain experts. Sorting the scores of all candidate IDS in ascending order provides a ranking by their degree of fulfillment of the domain experts from the best-fitting candidate on the top down to the candidates with the least desirable properties at the bottom. To select the final set of candidates, the prefilter chooses the top n approaches of this ranking of scores.

```
score_IDS ← 0;
foreach Characteristic c in the taxonomy do
    switch mapping(c, IDS) do
        case true do
            score_IDS ← score_IDS + prio(c);
        case false do
            score_IDS ← score_IDS − prio(c);
        case maybe do
            skip;
        case n/a do
            skip;
    end
end
return score_IDS
```

**Case-Study**

This prefilter was the first idea we presented to our industry experts very early in the development of this research. We asked the domain experts for a mapping of an early taxonomy before completing all interviews or the extension and validation through the literature. The domain experts followed the procedure and assessed each aspect and characteristic, while we noted the final values they assigned to each score. The discussion turned out to be more vivid and extensive than we expected, as the different departments differed in their priorities and needed time to agree on a common rating. Before this meeting, the domain experts were not aware of these divergences and rated the discussions as valuable. However, it was mostly the time pressure that enforced an agreement on potentially preliminary weights to enable us to continue our study instead of clearing all different opinions. We presented the ranking of the candidates yielded by their priority scores and received positive initial feedback. In particular, they liked the fact that the ranking showed a large gap in the score values. Three candidates in a preliminary set of candidates obtained a much higher score than the other candidates which separated these candidates with an easy and clear cut. They confirmed the identified candidates as promising based on their current knowledge. With the dimensions of our final taxonomy, the industry experts refused to apply this prefilter as they did not want to invest the time to prioritize the full set of characteristics.

**Advantages**

**Clear ranking of candidates.** As the only of the four we propose, this prefilter provides a complete ranking and numeric rating of all candidates as a final result. Based on the numerical ranking this filter even indicates which IDS of all selected candidates should be investigated first. The numeric scores highlight closely competing candidates that should also be included in further investigations. In addition, the scores reveal relatively big gaps to the next candidates in the ranking, if they do not match the priorities as well as the previous entry in the ranking. Both provide a distinct criterion for which candidates to consider and which to neglect before further investments.

**Enforces fixed and explicit trade-offs.** The task of rating all characteristics of the taxonomy via assigning a numeric value denoting its priority in favor or against this property is challenging. Each domain expert had some intuition about their priorities, but expressing them in numbers is another level of complexity. Especially seeing various values next to each other enforces a reflection on trade-offs and decisions about their relative importance. Unifying the priorities among multiple domain experts in the same meeting yielded vivid discussions about the reasons for individual opinions from different departments and eventually resulted in reasonable compromises in the trade-offs. In isolation from the rating, the domain experts considered these discussions as valuable for awareness in the team.

**Disadvantages**

**High time effort.** Walking through the full taxonomy and assigning priorities to all characteristics requires much more time than all other prefilters. A detailed discussion and rating of the full taxonomy was not feasible for our industry experts. Even with a relatively small taxonomy, they ended up only focusing on rating a few characteristics with the subjectively highest importance and all non-assigned weights remain zero. Thereby the non-prioritized properties did not take time for discussion but also had no impact on the ranking. However, such preliminary and nonreflected scores endanger the validity of the encoded priorities, affect the final ranking and inhibit the risk of wrong selections. Nevertheless, a focus on a few characteristics with higher importance is seemingly a big timesaver but requires more care in the selection process.

**Suitable weights are unknown.** The procedure of this prefilter does technically not restrict the values that can be assigned to a priority and leaves the choice to the domain experts. However, while the algorithm and ranking handled them flawlessly, our industry experts struggled with this freedom and invested even more time in their discussion. In practice, they only assigned very high numbers (e.g. 1000) to their must-have properties, used increments of 1 to relatively sort the characteristics within the same aspect, and aimed to group characteristics into classes of equal priority. Minor differences in the values caused larger discussions and resistance between some domain experts. This is particularly problematic as these relatively small value differences compared to the high numbers assigned earlier are almost without impact on the final ranking. They did not make any difference between suitable and not suitable candidates in the final selection. Hence, it seems to be more beneficial to limit the assessment of the domain experts to only a few, simple choices, but the question of which choices remains open. There are multiple potential ways possible: Our domain experts limit themselves between ±1000. In theory, ±100 or ±10 would have provided them with enough distinct values for their rating, but it would also have impacted the score calculation as the big gap was caused by a few characteristics with very large values. Maybe sparse intervals could have mitigated this impact on the rating, but would most likely also impact the flow of discussion.

### 4.6.2 Prefilter 2: Interactive Exclusion on Selected Characteristics

The domain experts know their highest properties for an IDS without a look at the taxonomy. The taxonomy should follow and support their initial discussion of these priorities without forcing it in any direction. After the set of candidates is reduced according to these priorities first, their mapping reveals previously not-considered properties.

#### Procedure

We asked the domain experts for the properties that, out of her head, are the most important for a candidate IDS to have. According to their answer, we searched the taxonomy (simple full-text search) for the corresponding element in the taxonomy. We marked the characteristics we found either as mandatory (requiring the candidates to have a mapping with `true` or `maybe` to this characteristic) or undesired (requiring a mapping with `maybe` or `false`). Based on this selection, the prefilter removed inappropriate candidates from the set and we announced the progress in reducing the number of candidates to the domain experts. We continued iterations of marking required properties and exclusions until either: 1) No candidates fulfilled the current set of requirements. In this case, we asked the domain experts to reconsider and change their choice of requirements. After a short discussion, the new requirements yielded another set of candidates and we started the next iteration. 2) Only a set smaller than the desired number of candidates by the domain experts is left. In this case, this set is the final outcome of this prefilter and all candidates are suitable for further investments in their evaluation. 3) The domain experts couldn't name more priorities. In this case, we paused the prefilter procedure and the domain experts had a closer look at the mapping of the remaining candidates. This resulted in them either spotting a new requirement (with a new iteration of the procedure) or considering all remaining candidates as interesting with only minor differences although their number was above the original maximum number of candidates. Please note, that while keeping the set of requirements invariant, the order in which these are marked has no impact on the resulting set of candidates after all requirements from the set have been marked. However, different orders result in differing intermediate sets of candidate IDSs and different numbers of removed candidates in each intermediate step. The final set of candidates is only impacted when the set of requirements is changed by the domain experts.

#### Case-Study

We have built a small tool to support this decision process interactively. After explaining the procedure, we left the structure of the discussion to the domain experts and only asked for short pauses to note down their conclusions in our tool and report their impact. Our domain experts appreciated the lean integration of this prefilter into their natural flow of discussions. They found the feedback on the reduced candidate number after every requirement particularly valuable. It provided them with a new understanding of the challenge within each requirement and highlighted problematic combinations of requirements permitting them to balance the underlying tradeoffs. While with prefilter 1, the

domain experts aimed to describe a single IDS for all their needs, this prefilter without our intention guided them in a different direction. When ending up with zero candidates after a requirement, the domain expert took a step back and reflected on the previous requirements. Finally, they ended up defining three different IDSs for specific purposes/attackers each with a distinct set of requirements that shall be used as an ensemble for full protection of the system. Neither we nor the domain experts anticipated this change in selection strategy, but consider it as a valuable realization.

In addition to this case study with the domain experts, we want to point out that we also decided on the IDS of Taylor et al. [131] that we used in Chapter 5 based on this prefilter. Namely, we searched for an IDS that: 1) Complies with the network characteristics of the car we investigated (periodic messages (Characteristic S2.2) containing various payloads from sensors and counters and flags (Characteristics S4.1-6)). This did not exclude a single IDS from the set of candidates. 2) Contrasts the IDS of Marchetti et al. [77], the other IDS we investigated in Chapter 5 that uses simple rules on the package arrival time; such IDS should use a machine-learning-based model (Characteristic D10.2.3) from training with an unlabelled dataset (Characteristic D9.2) utilizing information from the payload (Characteristic D5.3.5). These criteria resulted in two candidates and we selected the more cited approach. We want to point out, that this choice of IDSs also correlates to the clusters we found with Section 4.5.4. Namely, we searched for candidate IDSs opposing an existing IDS from cluster 2 with an IDS complying with the key properties of cluster 3, while still being suitable for the protected system.

#### Advantages

**The domain experts define the flow of discussion.** This prefilter leaves the structure and order of the discussion mostly to the domain experts. Our domain experts from different departments have already formed mechanisms for collaboration and hierarchies and responsibilities in decisions. Instead of aiming to hardcode these individual structures into a procedure or enforcing a different workflow, this prefilter naturally and dynamically supports already established processes and structures with numbers and additional information. The taxonomy is mostly used to keep track of mutual conclusions by unifying and formalizing the stated requirements and immediately reporting the impact of these decisions.

**Feedback on self-evident and impossible requirements.** After stating a requirement, this prefilter provides direct feedback to the domain experts on how many candidates are excluded based on this requirement. On the one hand, this immediately spots impossible requirements as zero candidates comply with them. On the other hand, this also reveals requirements that do not restrict the remaining set of candidates any further. Both were valuable feedback for the prioritization and negotiation of requirements within the different departments represented by our industry experts. Often, it was not a single requirement that was problematic, but the combination of two or more. The explorative reconsideration of requirements was spinning around avoiding these problematic combinations through an ensemble of multiple IDSs.

#### Disadvantages

**Explorative approach without clear end.** A while-loop without a guaranteed end criterion forms the core of the procedure of this prefilter. In the theoretical worst case on problematic mappings, this while-loop forces the domain experts to change the decision about each aspect and countless combinations of them. At any iteration, there is no assurance that all relevant requirements have been covered until now nor is there any mechanism to determine if the current set only consists of almost identical approaches. Hence, this prefilter needs to rely on the expertise of the domain experts to stop iterating neither prematurely nor before full exhaustion.

**Problems to distinguish similar candidates.** While this prefilter works efficiently to reduce the full, initial set of candidate IDSs, it struggles to distinguish IDSs that only differ in a few, specific aspects. Although these candidates are not identical, domain experts only name the aspects necessary to decide between similar options by chance and in the worst case might not even be aware of their fine differences. What is the main difference in the remaining candidates was an often stated question without a clear answer by this prefilter. Albeit it is debatable if not just any choice among very similar approaches is, in the end, acceptable, this prefilter in our experiments often resulted in sets with a too high number of candidates. These larger sets could not be reduced further only with the requirements stated by the domain experts without further input from the mapping.

### 4.6.3 Prefilter 3: Characteristics with the Highest Entropy First

As we analyzed in Section 4.5.3, many characteristics of our taxonomy are not useful to effectively differentiate the set of candidate IDSs. The entropy of a characteristic as we defined in that analysis quantifies the discriminatory power of each characteristic. Stating requirements concerning characteristics with high discriminatory power is the key to a quick prefilter and should be discussed first.

#### Procedure

The core of this prefilter is identical to prefilter 2, but uses a slightly modified iterative loop: The domain experts do not state the requirements in their own order but have to follow the order given by the prefilter. Namely, in each iteration, this prefilter automatically brings up a single characteristic and only asks the domain experts for their assessment of this characteristic. The domain experts discuss their concerning requirements and, similarly as in prefilter 2, decide in favor or against the desired IDSs having this characteristic. The prefilter then drops all candidates whose properties contradict this requirement and reports the progress in the selection. If the characteristics are not relevant to their requirements, the domain experts can also skip its assessment and continue with the next automatically suggested characteristic. This loop continues until only a single candidate or identical candidates considering the non-skipped characteristics are left.

The prefilter picks the characteristic with the highest entropy in the mapping in the set of remaining candidates as the next characteristic raised for domain expert assessment.

This design and calculation is analogous to the analysis in Section 4.5.3. To enable the iterations, we used a slightly adjusted formula to calculate the Shannon entropy that only considers the data of the currently remaining candidates instead of considering all candidates. Despite their entropy, characteristics that have been skipped by the domain experts once will not be considered or stated again.

**Case-Study**

We extended our tool from prefilter 2 to calculate the entropy of the remaining candidates and state the characteristic with the highest entropy in real-time. We used this tool in the same way as earlier to enforce the properties, reduce the set of candidates, and report progress in the selection. At the beginning of each iteration, we stated the next characteristic to be discussed with the domain experts guided by the entropy values. This different order reduced the set of candidates very efficiently in comparison to prefilter 2. However, the selected order of characteristics felt arbitrary for the domain experts. Consequently, they needed information from the higher aspect and the corresponding other characteristics for their assessment. Furthermore, the selected characteristics jumped unpredictably between the main responsible departments for the stated property further scattering the flow of discussion. This delayed the start of the discussion significantly compared to prefilter 2. We have to admit that analyzing the previous two prefilters with the same domain experts has introduced biases. Especially, the idea of multiple, specialized IDSs occurred again once the entropy had selected a characteristic that was chosen by the domain experts in prefilter 2 before. For other, previously unprioritized characteristics the domain experts decided in favor of the common-sense "better" they expressed in the minor numerical difference in the total classification in prefilter 1. Although theoretically, this prefilter spots even minor differences and selects at the end a single candidate, in practice the domain experts did not care much about these differences and skipped their assessment. Consequently, the analysis ended early with an exhaustive list of non-relevant properties and consequently multiple very similar candidates. The presence of `maybe` classifications in the mapping has artificially increased these similarities.

**Advantages**

**Fast convergence to small sets.** Ideally, the highest entropy denotes the characteristic that is only present in half of the candidates. In our case study, depending on the choice of reply, this applied to the first two to three characteristics in our candidate mapping. This reliably reduced the number of candidates by 70% to 80% within a very short time. As the entropy ensures this property, we expect this effect to scale to datasets much larger than in our case study.

**Successful discrimination of almost identical candidates.** Without guidance, it is complex for humans to spot differences between similar candidates in the long matrix of mapped characteristics. The entropy-based procedure guides the focus immediately toward these differences. Although frequently the assessment by our industry experts of these last characteristics was without conclusion, such insight about irrelevant differences was not yielded explicitly with the other prefilters.

#### Disadvantages

**Unintuitive order and characteristics without context.** During the design of the taxonomy, we used the coupling between an aspect and its grouped characteristics to ease the understanding and remove redundancy in the formulations. This prefilter selects individual characteristics from varying aspects in an unexpected and unanticipated order for the domain experts. Hence, at first sight, they lack the context that is necessary for the domain experts to assess the desired realization for promising candidates. While we provided this contextual information with our tool support, this property still disturbed and hampered the natural flow of discussion and thoughts among our industry experts.

**Non-relevant characteristics delay the termination.** After deciding on the first characteristics, the selection progress noticeably reduced its speed. We often faced situations where the few remaining candidates mostly differed in `true` and `maybe` (or `false` and `maybe` respectively). The domain experts often had no requirements for the other remaining characteristics presented by the prefilter and discussing them was pointless. It would be better to end the prefilter procedure before this exhaustion, e.g. when the size of the set is within the maximum desired size. However, the procedure lacks a fixed mechanic to end the assessment earlier and just consider all remaining candidates as worthy of further investigation. Therefore, this critical step is unfortunately left to the experience of the domain experts noticing this in time by the stated characteristics.

### 4.6.4 Prefilter 4: Pick from Clusters of Similar IDSs

In prefilter 2 and 3 we have seen that the prefilter concludes in a set of rather similar candidate IDSs. As hinted by the analysis in Section 4.5.4, these groups of similar approaches can be elicited from the mapping without further input. These groups help humans to compare their designs and either 1) ensure different designs among the final selection or 2) solely focus on the most promising design.

#### Procedure

In Section 4.5.4 we encoded our categorical mapping to a numerical feature matrix and used the Pearson Correlation Coefficient to highlight clusters. However, there are further ways to identify clusters and especially, explore them interactively. We chose Morpheus to implement this prefilter as it unites various alternatives in an intuitive GUI that can process our feature matrix. In detail, it implements k-means++ [13] and Hierarchical Clustering [93] based on several distance measures. In this prefilter, the domain experts explore and assess the common properties of such clusters instead of asking about their opinion on individual characteristics or IDSs as in the previous prefilters. We asked them to note and mark clusters that they considered as suitable for their requirements. To use the marked clusters to select candidates we see two possible approaches: 1) We pick an equal amount of candidate IDSs out of each marked cluster, totaling the final desired number. To pick representative IDSs from each cluster, we could select the candidates closest to the centroid of the cluster, the most cited publication of the cluster, or simply

random members. This ensures maximal diversity among the considered candidates focusing an evaluation between differing ideas. 2) We pick all or most members of (smaller) marked clusters until we exceed the final desired number. Once again, candidates closer to the centroid, most citations or other rankings could be preferred if needed. This ensures maximal similarity among the considered candidates focusing an evaluation on the best available implementation.

**Case-Study**

We provided a labeled feature matrix of our mapping and access to Morpheus to our industry experts. They experimented with different clusterings and explored potential hierarchies of the aspects to structure all candidate IDSs. The structure and result were dependent on the concrete choice of clustering algorithm and differed widely.

The domain experts confirmed our interpretation and labeling of the three clusters from the analysis Section 4.5.4. The diversity of these clusters exceeded their expectations and reached beyond their initial scope of considered IDSs. However, it was only possible for them to consider one cluster in their search for an IDS. Namely, the third cluster did violate their requirements for low computational complexity. Furthermore, implementing an IDS from the first cluster would require changes from their suppliers that could not be implemented in the current system generation already behind the design phase. Nevertheless, they considered the ideas of the first cluster as reasonable for future generations, where this design could still be changed. In total, they focused on and continued their analysis with all candidates of the second cluster. Although the clustering algorithms proposed various hierarchies inside this cluster, a further differentiation with this prefilter alone was not humanly comprehensible.

**Advantages**

**Minimal input necessary.** This prefilter does not require any input from domain experts but interpretations of the yielded clusters. The only information needed for the clustering is the mapping of all potential candidates. Most notably, the clustering algorithms were able to form groups that we could label manually without confusion, although our taxonomy is flat and does not explicitly model any interrelations or dependencies. Hence, this prefilter might provide a better comprehension for domain experts at the beginning of the requirement elicitation.

**Broaden knowledge and break biases.** This prefilter is the only among our procedures aiming for a diverse set of candidates. Although the domain experts disliked most of the suggested IDSs, the suggested approaches were beyond the candidates they originally were aware of. They had not considered any similar approaches at the beginning of their IDS elicitation happening in parallel to this research project. The neglected clusters did not contribute to the further system development, but assessing this broader space gave them the confirmation to not have missed out on vital approaches that are of major importance in academic publications. Hence, we consider this prefilter as useful for teams that are not familiar in depth with intrusion detection for the given domain and want to gain an overview of the landscape of all potential candidates.

#### Disadvantages

**Small clusters are hard to comprehend.** While it was intuitive to label the three big clusters in our mapping, our industry experts were not able to put clear, comprehensible labels on the smaller clusters or sub-clusters. The data mining approach had no human-understandable information to justify their grouping and clustering. Our design for the final selection of candidates in the procedure was meant to cover corner cases but turned out to be mandatory in all selected clusters by the domain experts. Hence, this prefilter might not be suitable for making the final decision among a few, rather similar candidates.

**Outliers are neglected.** This prefilter focuses on selecting individual, rather central candidates out of clusters of many similar IDSs. Only with an unreasonably high number of selected candidates, outliers from the main clusters are selected by this prefilter. This inhibits the risk of missing revolutionary approaches, that drastically differ from previous approaches. If the initial pool of candidate IDSs is constituted of academic publications, this especially is the case for recently published approaches that have not been refined or varied in follow-up research. While it is possible to explicitly search for outliers too, they outnumber the given, desired number of candidates.

### 4.6.5 Combinations of Prefilters

Our industry experts were not able to determine a clear preferable best prefilter among our suggestions. This originates from the diverging strengths and weaknesses of each prefilter we elaborated above. Ultimately, we believe that a sound methodology for candidate reduction should apply multiple of these prefilters in a row. The general idea should be to reduce the set of candidates first and then sample from the remaining options. For example, (prefilter 2+3) by first excluding with the selected, desired properties and then agreeing on more requirements by the entropy of the characteristics in the remaining set. Or (prefilter 4+1) partitioning the candidates via the clustering approach first and then selecting clusters and candidates based on their priority score within each cluster. In fact, after we presented all the ideas for the prefilter, the industry experts combined them in their exploration and informally transferred the insights from one prefilter to the others. Instead of following one of our proposed methodologies strictly, they fostered their knowledge gain and increased confidence in a decision about the selection. They iteratively selected approaches, read their publications, and experimented with different views on our dataset and the corresponding differing or related recommendations.

In theory, our four basic prefilter ideas can be combined into twelve different pairs of two. Investigating each of them in more detail goes beyond the scope of this thesis and the research project with our industry experts. Especially, since our later experiments already revealed biases from previous analyses. Furthermore, our initial pool of candidate IDSs is most likely too small, as a single prefilter in our case study was already sufficient to identify the desired, small number of candidates. However, we expect this to change with a larger pool of candidates, especially as we became aware of 248 publication of candidate IDSs in our literature study. Therefore, we leave these efforts to future work.

## 4.7 Discussion

### 4.7.1 Observations about Qualitative Properties and IDS Selection

This research about prefilters combines multiple artifacts and research methods: Expert interviews, a literature study and data analysis. While the previous sections investigate the properties of each artifact individually, this subsection aims to provide higher-level observations that unite the individual views. Namely, we want to highlight four observations and support them with evidence from our study.

#### Observation 1: The Taxonomy Is Too Large, But Irreducible

While designing this research, we made some design decisions in the hope of benefiting the usage as a prefilter. Our intent was a precise description of all IDS candidates with a fine-grained differentiation of all of them. However, we could not determine upfront which of the properties are relevant and most discriminatory for potential candidates. Hence, we designed and followed a methodology towards the maximum coverage of all potential aspects and characteristics of an IDS in the hope of identifying the relevant parts later.

Our process of expert interviews and literature study resulted in a large taxonomy with 34 aspects and 179 characteristics. Indeed, our final taxonomy unites more properties than any previous taxonomy on IDSs about the use case as we investigated in Section 4.5.1. Notwithstanding, the time to map an IDS increases with the number of characteristics in the taxonomy, especially when they require detailed and precise knowledge. In addition, the industry experts while prioritizing all aspects for prefilter one (see Section 4.6.1) yielded that this taxonomy is too large for efficient comprehension and application. This raised the question of how the size of the taxonomy could be reduced to only the most relevant properties.

In alignment with our two-fold methodology, it might be reasonable to either only depict the broader view of the industry experts or the deeper, detailed view of academic publications. Aside from the question of which of the both is preferable, this also inhibits very limited potential to reduce the taxonomy size. As investigated in Section 4.5.1, skipping concrete, technical details would only skip three aspects and skipping the broader perspectives would skip eight aspects. Both constitute not even a third of the taxonomy. Overall, the benefits of such reductions are limited, since both views mainly provide complementary characteristics within the same universal aspects instead of adding new characteristics to the taxonomy.

Another idea might be to reduce the taxonomy size by skipping properties that are not suitable to efficiently discriminate the potential candidates. The best approximation for this suitability is the entropy gain as utilized for prefilter 3 (see Section 4.6.3). While such reduction might hamper the differentiation of similar approaches, it also would not reduce the size of the taxonomy. As discussed in Section 4.5.3 only two aspects have an entropy of zero and could be skipped without loss. Only five other aspects have a relatively low minimal and maximal characteristic entropy. Furthermore, most characteristics contain

at least one aspect with a high entropy and should not be skipped using this idea. Hence, once again, the majority of the taxonomy is relevant and would not be reduced.

Most critically, there is one problem in the design of all these notions of important taxonomy parts: These assessments utilize the mapping of existing publications and concrete candidates of a specific use case. They do not relate to the actual suitability or any general reasoning. Even if we were able to identify a clear indicator for size reduction, this choice would be subjective and only valid for the use case that is the focus of this thesis. For other use cases or other perspectives and prefilters or future developments, in the worst case, it might even be harmful to skip properties prematurely.

In sum, based on our analyses, we do not see a way to reasonably reduce the size of our taxonomy. Therefore, we do see the need for structures, methodologies and tools that enable a comprehension and efficient utilization of such a large taxonomy of properties. We propose such tooling with the prefilters 2, 3 and 4. Especially, data mining approaches as indicated by prefilter 4 (see Section 4.6.4) profit from a larger taxonomy without any manual reduction of their feature space. All this supports the conclusion, that a reduction of the taxonomy might not be possible or desirable.

**Observation 2: Explicit System Assumptions and Attacker Model**

An evaluation with a benchmark dataset only quantifies the classification behavior of the analyzed IDS on the behavior contained in the dataset. Consequently, the results of the standard evaluation schema are only meaningful regarding the behavior depicted in the benchmark dataset. Complementary, the behavior of the IDS on samples not included or behavior represented in the dataset remains hidden and untested. Hence, the central property of a benchmark dataset is to contain representations of all the types of expected system and attacker behavior.

Thus, it is a critical oversight in the evaluation to not consider relevant behavior in the benchmark dataset. As we discuss in Section 6.2.2 composing a benchmark dataset is a challenging task that regularly has resulted in the publication of flawed datasets. This issue gets amplified in situations when not all requirements on relevant traces are known and enumerated exhaustively. Especially early in the development phase due to the low maturity, it is likely that the domain experts to not be able to state all expectations explicitly and completely. On this background, it is of major importance to establish guidance in eliciting these requirements.

Our taxonomy and prefilter help to consider all relevant aspects of the attacker and the system in the evaluation in two ways: 1) the prefilters ask the domain experts directly and detailedly for their expectations on the system and attacker behavior in both corresponding views. The prefilters use these expectations to exclude unsuitable approaches before applying the standard evaluation schema. Namely, this drops IDSs upfront when it is known to not be capable of correctly classifying a specific type of expected behavior. The holistic taxonomy eliminates the risk that the domain experts might have missed this property in an unstructured analysis. 2) Although our prefilters are designed to filter IDSs, they also support the composition of a benchmark dataset for the following more detailed evaluation. The selected and prioritized properties of an IDS in the views of the

attacker and system can serve as a checklist for the composition of the benchmark dataset. A suitable benchmark dataset to evaluate IDSs with these desired properties should contain at least a few samples for each selected aspect. Consequently, the information selected for applying the prefilters also contributes to a higher quality of the utilized benchmark dataset. Both aspects reduce the risk of oversights in the evaluation process.

In the case study with the industry experts, we could observe both aspects immediately in the evaluation of prefilter 1 (see Section 4.6.1 for details). The industry experts did not anticipate the fine details in the taxonomy and stated multiple questions and clarifications about the differentiation between the characteristics. Furthermore, they have not been aware before of the impact of these seemingly minor differences in the outcome of all the prefilter. They indicated that these experiences made them reconsider their criteria during the benchmarking of IDSs with the standard evaluation schema.

In sum, the taxonomy and prefilter clearly foster explicit discussions and specifications of all assumptions on the system and attacker model. This early commitment during all steps before the standard evaluation schema yields positive effects that reach beyond the preparation and also improves the validity of the obtained evaluation results.

**Observation 3: Structured Knowledge Base Instead of a Fixed Prefilter**

The main criterion to judge the quality of a prefilter is its efficiency while reducing the set of candidates. Nevertheless, we are not able to identify a concrete, smaller set of candidates as the outcome of one prefilter (except the IDS from Taylor et al. [131] we will use later research in this thesis). We believe that this originates from multiple reasons that we want to list in the following.

First, there simply might not be a fixed number of candidates that is the ideal, desired outcome of the prefilter procedures. We asked our industry experts very early in this research what number of candidates they could afford to investigate with the standard evaluation schema, so we aim to reduce the set of all candidates down to this number. Without further justification or explanation, they wanted us to aim for three candidates as this number seemed reasonable to them. If we were enforcing a decision after applying the prefilters, all would result in a different number of candidates roughly around this goal. Prefilter 2 suggested six similar approaches, prefilter 4 five similar or three differing approaches, prefilter 1 suggested three candidates and prefilter 3 suggested only one, or three candidates for three distinct purposes. The set of candidates differed largely, depending on the focus on similarity or diversity. Often, the boundaries for including or excluding a specific candidate in the final set were fuzzy. Having seen this outcome, our industry experts were no longer confident in their original limit of three candidates. No set of candidates was clearly preferable over another and they all seemed reasonably sized for the spectrum of options they represent. Instead of a clear decision, the industry experts decided in favor of considering all these approaches over different stages in the development and also in potential future projects.

Most notably, with all four prefilters, our industry experts did not follow the methodology to select candidates to the end. Instead, they used our tools to explore different presentations and the interrelations that now became visible. Prefilter 1 made its biggest

impact by enforcing an explicit ranking of all the requirements in a single document while the ranking was actually of little interest to our industry experts. The most valuable outcome of prefilter 2 is not the reduced set of candidates, but the feedback to the domain experts on which combinations of their requirements are problematic. Prefilter 3 yielded the quickest reduction of candidates, but profited and was dependent on the understanding that our industry experts have gained with applying the previous prefilters. Prefilter 4 made the industry experts curiously explore the full, broad spectrum of candidates and identify their key properties, although most of these candidates might be unsuitable for the concrete requirements of the industry experts. They used all prefilters to dig deeper into the information about particular groups of candidates many times, while they were not able to spot individual concrete information or representations that made them decide in favor of specific candidates.

However, we noticed an internal structure in the industry experts' explorations with the prefilters. The prefilters have different strengths and weaknesses that complement each other. Prefilter 1 focuses on an overview of all properties of IDSs while prefilter 4 provides a comprehension of all potential candidates. Both perspectives are particularly valuable in the early development stages and fragmentary knowledge about potential candidates and requirements. Prefilter 2 links requirements on IDSs with the amount of candidates providing these desired properties. It helps with the transition from unrelated requirements into more concrete technical designs by spotting required trade-offs. Finally, prefilter 3 provides the means to efficiently select candidates based on a clear understanding and structure of all requirements. In other words, they follow and align with the refining phases in the requirements engineering process. Successful requirements engineering needs flexibility and customization to handle early imprecisions. Our proposed set of prefilters provides support for domain experts during the full requirement engineering as long as each filter is utilized at the most beneficial time.

In sum, we do not believe it is desirable to only propose and use a single prefilter. The elicitation of potential candidates is no one-time effort and does not happen linearly. It is an iterative process fostering a better overview for the domain experts leading to a better understanding of the design decisions. Ultimately, the goal is to increase the confidence of the domain experts in their decision process. The four prefilters we proposed in their combination contributed towards this goal as our industry experts confirmed. Similarly, the standard evaluation schema does not enforce the usage of one particular metric for the assessment but requires the domain experts to choose suitable metrics from a set of complementary alternatives.

**Observation 4: Complementary Views Instead of one Universal Representation**

The backbone of our research is the creation of a taxonomy of qualitative IDS properties. By definition, a taxonomy categorizes and classifies the study subjects. Ideally, it should aim for a hierarchical organization and index the knowledge in a way, that the users can more easily find the information they need. Our prefilter design and the case study benchmark our degree of achievement towards these goals.

Before we analyzed the survey papers, we thought of utilizing some existing taxonomy instead of proposing our own. However, after extracting and linking properties from the first publications, we realized that each existing taxonomy had its own categorization and hierarchies (see Table 4.1). Being confronted with taxonomies utilizing contradicting structures, we saw no alternative to keep the meta-model (see Section 4.3.1) of our taxonomy flat and minimalist. This enabled us to continue with the collection of properties but postponed the decision about a suitable hierarchy to the prefilter design. Our experiences with a neglected, easy navigation in both prefilter 1 and 3 underline the need for such orientation for a quick and comprehensible candidate comparison and rating.

As we showcased in Section 4.5.4 our flat taxonomy in combination with the mapped candidates forms a two-dimensional data matrix. A hierarchical presentation would at least require a choice for one dimension out of these two. However, our prefilters successfully use both views to reduce the number of candidates. Namely, prefilter 2 and 3 choose a view on the aspect and even individual characteristic level, whereas prefilter 4 presents a view on groups of IDSs sharing a large set of common properties. Hence, already this minimal decision has no objectively preferable choice.

Breaking the properties down into sets of aspects only increases this effect. Similarly as already elaborated in observation 1 it is not feasible to reduce the size of the taxonomy, it is not beneficial to structure and order the aspects by their relevance or impartance indicated by the same principles. The priorities of the domain experts (see prefilter 2) do not align with the properties with the highest discrimination (see prefilter 3). Properties to group the IDSs in bigger clusters (see prefilter 4) or complementary to properties that differentiate almost identical approaches (see prefilter 3). Ultimately, each new idea publication faces the same challenge to order the properties of their new approach and how much space of the publication is dedicated to each of them. As the irregular patterns of mappings with `maybe` (see Section 4.5.2) indicate, there is no common agreement among the authors about these priorities.

In sum, our research showcases many complementary views that all serve different purposes and follow different justifications for this means of presentation. Consequently, we believe that it is not possible to elicit one, universally best taxonomy of qualitative IDS properties. However, we want to point out that our taxonomy provides a unified data model for storing the core properties, but because of its simplicity enables the various ways for analyzing and presenting them as we did in this research. A change in the properties complying with the meta-model would seamlessly integrate into all our views. Hence, we believe that this split is vital for future research as it enables maximal reuse of once-identified properties.

### 4.7.2 Threats to Validity

#### Internal Validity

To conduct the systematic literature survey and to select the most cited papers, we have collected and relied on a pool of all papers on our use case and their references to each other. We cannot assure that we included all literature and that our selection of the most

cited publications is accurate, as we copied them from literature indexes [106]. A paper not listed there has silently been not included in our study. Furthermore, missing publications in the index impact the total count of citations per paper. Hence, our selection of the publications for mapping that we based on these scores might have been misguided. We reduced this risk by using six different online libraries for literature discovery and two different sources for the citation count. Indeed, the results obtained from the libraries differed, but we expect the union of these sources to mostly represent the actual state-of-the-art. Hence, our taxonomy should depict an accurate and holistic view of the state-of-the-art in our use case.

Most severely for the internal validity, we needed to reduce the required effort of our methodology to match the resources available to us for this study. For example, we used automatic exclusion criteria that have not yet been analyzed in academic literature. We investigated their effect and the filtered papers with manual random samples, but still, the criteria might have excluded publications that a manual expert filtering would have included in the study. In addition, most steps that required human expert judgment were only conducted by two researchers who needed to conclude the same opinion in their classifications. Last, we decided against mapping all 248 IDSs but only mapped 19 IDSs to our taxonomy for creating the dataset we used for our evaluation. This also introduced a threat to the external validity that we will discuss later.

Second, our taxonomy aims to provide a unified terminology to precisely describe the characteristics of an IDS and prevent misunderstandings. Although in the majority of the literature, we found a common terminology, we also came across publications that used drastically differing wordings. The most prominent example is the work of Xiao et al. [143] proposing an "authentication framework that exploits physical layer features of the messages". According to our definitions (see Section 1.1) this work presents an IDS that indeed can be flawlessly mapped and classified with our taxonomy. Nevertheless, this is our interpretation and might potentially not reflect the view of the authors. To reduce the risk of such misinterpretations every non-clear paper has been analyzed in full text by at least two researchers. This survey and our mapping depict their common opinion. Therefore, we are confident that our taxonomy uses a common terminology used in other publications.

Finally, the decision about whether a characteristic should be added to our taxonomy is surely subject to subjective views and biases and does not follow a strict and formal rule. While it is, in theory, possible to add an infinite amount of characteristics and aspects complying with the meta-model, trying to add more characteristics than we did often resulted in unproportional amounts of work and discussion. From an abstract perspective, we see two strategies to identify more characteristics. 1) Differentiating the existing characteristics further by considering minor differences that only apply to few IDSs. Based on the experience from the discussions with the industry experts, such details can cause confusion and lengthy discussions about imaginary examples and how these potentially would be mapped. Explaining these minor differences and clearing uncertainties was of little relevance in the mapping and IDS selection but very time-consuming and exhausting in the discussions. 2) Adding new aspects by spreading our taxonomy to other domains or fields of research, e.g. more details about potential reactions or processes used for devel-

opment and evaluation. In our study, the best example of such additional aspects is the internal model learned by machine-learning-based IDSs. Initially, we aimed to differentiate between the exact type of internal model, e.g. decision tree or convolutional neural network. Scientific publications about new detection ideas present information regarding the internal model in high detail as it forms the core of their detection approaches. Consequently, there are taxonomies solely focusing on classifying such model types used for intrusion detection. For example, Liu and Lang [71] differentiate 20 distinct types of machine-learning algorithms. However, we did not find two IDSs in our dataset that rely on the same internal model and accordingly, such an aspect would not further structure the candidates. In the end, there was not even a common understanding of the terms among all our industry experts as they are no experts in machine learning. Such decisions about characteristic exclusion might be debatable and not fully reproducible by other researchers in other setups. However, we did not encounter a situation where the aspects and characteristics we collected in the taxonomy were insufficient to depict any candidate or the requirements of our industry experts appropriately. Therefore, we see no issues with other researchers or practitioners extending and refining our taxonomy toward their needs while our overall methodology remains valid.

**External Validity**

The biggest concern in the construction and deduction of the prefilter is the lack of ground truth about the actual performance of all analyzed IDSs. As the IDS prototypes are subject to ongoing research and development, up till today nobody knows which are the best defenses or the most suitable in a given use case. To the best of our knowledge, there is in general domain, even beyond our use case, where the best IDS among a broad set of prototypes. The assessment with the industry experts confirmed the plausibility of our selections, but only implementing and evaluating all candidates within the real system is proof of this opinion. If a "good" IDS among our candidates were known by certainty, we could evaluate our prefilters about whether they reliably select this IDS in various constellations. Similarly, if we knew about "bad" IDSs among our candidates with certainty, we could confirm their exclusion by the prefilters, and ideally with such labeling for all potential candidates conduct a fully sound evaluation. Furthermore, we could use the performance in reducing the candidate sets to optimize the prefilters or debug them if necessary. However, as long as such ground truth does not exist, these optimizations are not possible and an evaluation of the prefilters is limited to analyzing the systematic process but not the yielded set of candidates.

A sound evaluation of the processes around the prefilters requires a larger setup and more resources than what was available to us. Our collaboration with industry experts provided us with intense discussions and feedback on the prefilter design but was limited to a relatively small group. While the literature study mitigated the negative effects of data obtained from a small, non-representative expert group in the construction of the taxonomy, such post-validation was not feasible during the evaluation of the prefilters. Still, we decided to summarize the comments on the prefilters of the industry experts as a case study for a preliminary evaluation and valuable hint for future research. Never-

theless, the analysis of the divergence between industry and academia and the validation in the case study can only be seen as anecdotal evidence. We expect a representative study to confirm our findings but also reveal further insights and possibly biases in the data we analyzed.

Our taxonomy uses an intentionally flat and simple meta-model of aspects and characteristics. According to the meta-model, all aspects are equally important and on the same level of detail. However, during the selection process, we had various discussions about their granularity and discrimination. Overall, the aspect's level of detail varies drastically and only a few are on a universal and abstract level. The majority of aspects on a more detailed level encode the specificities of the use case and do probably not apply in general to all IDS. However, excluding all specific attributes would severely reduce the size of the taxonomy and thereby the utility and effectiveness of our prefilters. Consequently, the generalizability of our taxonomy in its current state is limited. Without further research, it is impossible to estimate the degree of this limitation and how drastically the taxonomy would differ if our methodology were repeated on another use case. Nevertheless, we want to point out that due to the universal meta-model, our tools and prefilters could process different taxonomies providing the same functionality.

Finally, we are uncertain about the desirable and reasonable number of candidates considered in the case study. We considered 19 publications which is significantly larger than the number of candidates our domain experts originally considered before the development of our prefilter. However, compared to the 248 candidates discovered in the literature search this case study is relatively small. As the idea of the prefilters before the standard evaluation schema is novel, we only have an estimation which number of potential candidates is realistic. A full investigation of all candidates, that we became aware of, would have increased the size of the dataset we used to develop and showcase our prefilters by a magnitude. In the analyses, no two IDSs had an identical mapping and we were also able to identify clusters of similar IDSs. Hence, we expect our dataset to be sufficiently sized to expose the relevant statistical properties forming the foundation of our prefilters. Nevertheless, the prefilters might show differing results and performance with more remaining candidates in each step. Most likely, these bigger candidate sets can simply be further reduced with more iterations within the same prefilter. We expect the factor of reduction from a large set of candidates into a smaller dataset to stay invariant.

### 4.7.3 Future Work

We believe that our work is an important step towards enabling an early and cheap assessment of a maximal number of intrusion detection systems before any implementation and execution. Nevertheless, this initial step lays the path to further research and this chapter gives an outline of the most important future work.

The biggest problem of our work is that there is no commonly agreed and proofed "good" IDS for our use case. Hence, we couldn't properly validate our proposed methods for suggesting IDSs as the actual quality of the considered candidates elicited before is unknown to us, our industry experts and in research. However, in the future, such IDS might be known. As elaborated in Section 3.3, automotive manufacturers currently

work on deploying basic IDSs in the internal car networks and invest in their refinement. Therefore, in a few years, a successfully operating IDS most likely will exist and could be classified with our taxonomy. This new information and data could enable new research in two directions: 1) Our filters ideally should select this "good" IDS for further analysis. In case they do not, further investigations are needed about the reasons and potential extensions and adjustments of our filters. This would be validation research addressing a major threat to the validity of our research and would open new performance metrics for the prefilters. 2) As explained in Section 4.6.5, we see the potential for combining our prefilters into a more sophisticated, multi-step prefiltering methodology. However, investigating a dozen different methodologies with industry experts considering our experience with only four prefilters is most likely not feasible. Assuming such "good" IDSs in the set of candidates are known, this would provide alternatives to this form of evaluation. Each combination could be validated objectively about whether they successfully identify the "good" IDSs and how efficiently they do this selection in the current combination and configuration. This would extend our research and most likely result in more useful and powerful prefilters.

Through our choice of a concrete use case, as also elaborated as a threat to validity, we scoped our study to our use case and thereby tailored our taxonomy to specific traits. Although we provided a meta-model and identified general aspects, the potential and consequences of generalizing our taxonomy should be investigated further. Therefore, we see the need to repeat our approach on a completely different use case shedding light on more aspects. In the automotive domain, network-based detection is the dominant approach and the IDS operates in a cyber-physical system. Therefore, the most differing results are obtainable with a use case from a different domain, ideally where host-based detection is favored. The resulting taxonomy will overlap with our taxonomy but also contain new aspects and refine existing aspects with new information that did not apply to the automotive use case. The number of not applicable aspects characteristics from one use case to the other might also yield a new problem worth investigating. We assume that there is a common and generic core that is universally applicable and use-case-specific extensions covering a broad spectrum of aspects. However, our research does not mark both explicitly and leaves their ratio unknown. Any taxonomy can never be complete, but we hold such research to be a desirable extension deepening the understanding of the use-case as well as the nature of taxonomies like ours. Nevertheless, this project might require a similar, additional effort than we invested into this work and is therefore out of the scope of this thesis.

Finally, we see possibilities to transfer our methodology of a systematic prefilter to other use cases rather than solely identifying promising IDSs. Based on our experience, we see the biggest potential in scenarios, where various machine-learning algorithms are suitable to solve the problem. Many aspects of our taxonomy classify the internal models and the utilized features of IDSs using machine learning for anomaly detection. The steps we used for eliciting our taxonomy and our filter methods can serve as a blueprint for future initiatives that aim for the prefiltering of various candidates early in the system development phase. It would be particularly interesting to try if our prefilter and candidate selection methods still work for other usages.

## 4.8 Conclusion

This chapter presented four prefilters that identify promising candidates from a large set of candidates before investing in their analysis with the standard evaluation schema. These prefilters are solely based on abstract properties and do not require any execution of the candidate IDSs. Hence, this suitability analysis can happen in the early development stages and without operational systems or configured IDSs. Thereby, they reduce the costs of the evaluation process and increase the chance to consider promising candidates. We confirmed the suitability of the prefilters for a selected use case during a case study with industry experts.

The core of the prefilters is a taxonomy of properties that describe abstract traits of the candidate IDSs. We propose a simplistic meta-model and have built a corresponding taxonomy through multiple steps: Initial industry interviews, extension through other existing, peer-reviewed surveys and an extension and validation through the mapping of the most important new idea papers for intrusion detection. Thereby, this taxonomy covers a broad spectrum of properties and unites multiple perspectives on the decision process for potential candidates. This taxonomy can describe candidate IDSs as well as requirements for a suitable IDS through simple mappings to each property in the taxonomy.

To deduce the prefilters, we mapped 21 IDSs to the taxonomy and analyzed their so structured properties in detail. Our analyses highlight the diversity of covered properties as well as the issues with ordering them or removing properties from the taxonomy. Furthermore, we documented various complementary views on the flat collection of IDS properties. For example, the initial perspective on the properties of intrusion detection from our industry experts differs from the view presented in academic publications. A quick comprehension of all possible candidates requires another view than the effective differentiation of similar approaches. The prefilter utilizes a link to selected perspectives with the requirements stated by domain experts to reduce the of potential candidates to the IDSs optimally fulfilling the needs of the use case. Thereby, the prefilters complement each other and match different stages in the requirement engineering.

The taxonomy's detail level as well as the interviews and case study with the domain experts would not have been possible without a clear focus and restriction on a specific use case. Consequently, parts of the taxonomy are use-case-specific and do not apply to other domains or use cases. However, the taxonomy also covers generic aspects of intrusion detection. Additionally, the meta-model as well as the prefilters handle future extensions and properties in the taxonomy seamlessly that cannot be applied to all concrete use cases. Therefore, we consider this research as an important building block toward establishing a prefilter as an early solution to the intrusion detection evaluation problem, that goes beyond the concrete use case investigated in this thesis.

# 5 Tailored and Confined Datasets

*The state-of-the-art approach for the Intrusion Detection Evaluation Problem relies on evaluation with benchmark datasets composed of the regular system's and potential attackers' behavior. These datasets are collected once and independently of the IDS under analysis. This chapter questions this practice by introducing a methodology to elicit particularly challenging samples to benchmark a given IDS. These new datapoints differ drastically from the points inside static benchmark datasets. We have published parts of this chapter in [53].*

## 5.1 Introduction

An essential requirement in the standard evaluation is having a dataset representative of the use case, which is also challenging for the analyzed IDSs. Especially for comparing various approaches in science, reusing such a dataset as a benchmark and baseline for future research makes creating sound datasets mandatory. Following the standard evaluation schema, domain experts collect the dataset first without considering the IDS they will evaluate later. This separation fosters a realistic evaluation and prevents biases. In science, researchers publish their dataset in a final and invariant form once they consider the collected traces sufficient. Other researchers then choose one or multiple of these datasets to showcase the performance of their newly proposed IDS relative to other competitors. However, this setup does not assess how suitable the chosen dataset is for evaluating the proposed IDS. Using an inadequate dataset endangers the validity of the obtained results.

In this thesis, we work with the hypothesis that the quality of a benchmark dataset needs to be measured relative to the IDS under evaluation. In other words, the IDS under analysis determines the properties and samples constituting a "good" dataset for its evaluation. This hypothesis questions whether the standard evaluation schema to using a constant dataset is a sensible approach—or if a fair assessment of the IDS requires tailored and confined datasets that incorporate characteristics of the IDS and detection model itself. This new view requires a link between the IDS's classifications and the suitability of dataset samples for evaluation. If suitable data points are absent in current datasets, a sound evaluation requires a new methodology that fosters the inclusion of the most suitable samples for a given IDS.

To establish this missing link between the IDS and the dataset, we propose fitness functions that quantify the suitability of individual dataset samples regarding six different qualities for evaluating a given IDS. Furthermore, we propose a methodology based on scenario-based optimization to systematically deduce data points for the given IDS with the best rating according to our fitness function. This methodology extends the state-of-the-art evaluation with a final step, generating new sample points for the last assessment before deploying an IDS. In a case study, we investigated two state-of-the-art IDSs defending the same attack and analyzed the datasets used for their evaluation. With our

methodology, we found, in one case, valid attacker behavior that circumvents the IDS and causes critical damage to the system. The finding that both original evaluations do not consider behavior similar to the points we deduced supports our hypothesis. We suppose that the transition from signature-based detection to machine learning-based approaches and the increasing complexity in the detection models further impede the manual deduction of challenging data points. The small size of our investigated datasets might have contributed to their lack of critical points. But even massive datasets might only include such data points by chance and currently do not guarantee their inclusion. In the worst case, the dataset includes these critical points, but they get lost in the enormous number of other classifications of the IDS summarized in percentage scores. Hence, domain experts might not become aware of this critical behavior during the evaluation of the IDS.

Although we believe our hypothesis generalizes to the evaluation of all IDSs in general, this thesis focuses on a deep investigation of our use case: Intrusion detection on the Controller Area Network (CAN) bus in the automotive domain. Our deductions use concepts from the network domain and modify an attack by manipulations of the communication on the network. This choice of an attacker makes our methodology more intuitive for network-based-IDS. However, the evaluation of a host-based IDS that, for example, validates and monitors the internal models during computation does look identical as long as it mitigates the same type of attacker. Our fitness functions require a direct measure of the success of an attack or its attempt. While we also refer to approaches to generalize this concept, we tailored our methodology, for now, to safety-critical cyber-physical systems, as safety distances and violations provide this measure with high precision.

We use the following definitions and terminology to talk about datasets. A *trace* is a finite observation of the system's behavior during operation, potentially under the influence of an attacker, i.e., a time series of events recorded from the system. Together with a label about the nature of the recorded behavior, *benign* or *malicious*, a trace serves as a *data point* to benchmark the correct classification by a given IDS. A *(benchmark) dataset* is a fixed collection of various and diverse data points proposed by researchers or domain experts. Please note that other literature uses terms like sample, record, observation, item, or instance to describe elements of a dataset. Other terms in the literature reflect the concrete representation of a data point, e.g., log files, NetFlows, event streams, or raw package dumps. However, the standard evaluation schema and our methodology are agnostic to the nature of a data point as long as they provide the features for the IDS and sufficient description for the domain experts. We introduce the term *dataset space* to refer to the set containing all possible datasets.

To summarize, in this chapter we make the following *contributions*: (1) We propose an ensemble of six distinct fitness functions quantifying the suitability of data points in a dataset for evaluating a given IDS. We intentionally limit ourselves to safety-critical cyber-physical systems but sketch possible extension points for future work. (2) We propose a scenario-based optimization to systematically elicit edge case behavior of the system and network attacker to highlight strengths and weaknesses relative to the particular IDS under analysis. (3) We exemplify this methodology for a detailed security analysis of an open-source advanced driver assistance systems (ADAS) on automation level two (Adaptive Cruise Control and Lane Keeping Assist) [113] and two state-of-the-art IDSs.

We provide a deduction of various realistic attack samples on an automated car usable as a benchmark for any CAN bus IDS. Overall, the divergence between the optimized data points and the previous benchmark dataset and between both IDSs leads us to conclude that optimal benchmark datasets must reflect the IDSs under evaluation. Therefore, (4) we elaborate on possibilities to establish our methodology within the development and evaluation process for IDSs in general and propose an adaptation in future benchmark datasets to ensure such critical data points.

The rest of this chapter is structured as follows: Section 5.2 relates our contributions to the standard evaluation schema. Section 5.3 elaborates our proposed methodology to generate optimized samples forming minimal benchmark datasets. Section 5.4 applies this methodology to our use case and elicits potential experiments. Section 5.5 presents the setups we use for these experiments. Section 5.6 investigates the selected scenarios and our experimental results in depth. Section 5.7 discusses the generated datasets and reflects on the impact of our work on future approaches for IDS evaluation.

## 5.2 Relation to the Intrusion Detection Evaluation Problem

Our new methodology discussed in this chapter aims to extend the standard evaluation schema introduced in Section 1.1.2 with an additional final assessment phase. This phase uses the candidate IDSs successfully passing a basic evaluation of the standard schema and elicites further, tailored challenges to ultimately benchmark the candidates before a deployment. The new methodology also consists of the same steps (Enrich, Analyze, and Rate) but combines them in an extended workflow. The third step differs the most, as it uses a different way to compare the yielded alarms of all IDS with the dataset labels. To handle the numerous data points in the standard evaluation schema, domain experts choose one or multiple metrics [126] and accumulate the counts of correctly and wrongly classified data points to obtain a final ranking of the best candidates. Our methodology uses optimization to reduce the number of datapoints to a small set that domain experts can fully investigate.

The classical metrics, e.g., Detection Rate or False Positive Rate, combine four different counters during evaluation: false positives (FP), true negatives (TN), false negatives (FN), and true positives (TP) depicted in Table 2.1 in Section 2.4. These counters are combinations of two properties: (1) the data point showing peaceful system behavior or system behavior under the manipulation of the attacker and (2) the IDS yielding an alarm or staying silent while analyzing this data point. We followed the same categories in defining our optimization criteria for deducing our fitness functions. In a classical evaluation, however, these metrics approximate the actual detection capabilities of the IDS that highly depend on the quality, quantity, and diversity of the analyzed data points. Our methodology aims to showcase the IDS's capabilities by focusing on particular data points for each category instead of an unprioritized accumulation of large datasets.

Noteworthy, the standard evaluation schema focuses only on the evaluation phase and assumes the existence of fully configured and executable candidate IDSs. Engineers generally differentiate between two datasets, one used for development (e.g., to train the

detection algorithm) and one for evaluation to avoid overfitting and biases [64]. Although anomaly-based IDSs, unlike signature-based IDSs, do not require knowledge about the attacker and are trained on purely peaceful system behavior, the evaluation always requires realistic and diverse attack samples. Without attack samples in the dataset, it is impossible to determine the number of true positives (attacks spotted by the IDS) and false negatives (attacks remaining unnoticed by the IDS), and the evaluation remains partial. Since the first step in the standard evaluation schema relies on manual effort to collect and label the dataset, it might unnoticeably invalidate the evaluation. Wrong-labeled, oversimplified, incomplete, monotone, biased, or undersized datasets may result in good evaluation results in step three, even when the IDSs, in reality, do not provide sufficient protection against actual attackers. Therefore, collecting more advanced and subtle samples of attacker behavior from diverse attacks is vital for a sound evaluation and a core part of our methodology.

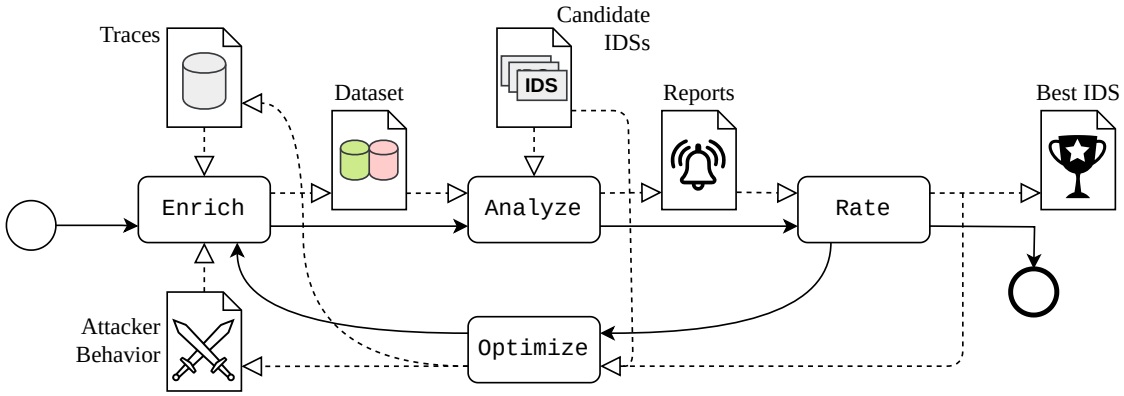


**Figure 5.1:** *The standard evaluation schema extended by our dataset generation.*

Our work addresses these limitations by extending the evaluation schema as follows (cf. Figure 5.1): We introduce a feedback loop from the observed detection abilities on an initial dataset of the candidate IDS in the Rating step to the Enrich step to generate new, labeled data points. The essential artifact of this loop is the systematically spanned space of all potential datasets. To automate this process, we propose optimization using fitness functions to guide the generation of data points toward system behavior that showcases behavior specific to the IDS under analysis. These data points represent edge case behavior and are complementary to existing metrics for the detection rate and false positive rate of a candidate IDS and, in theory, make an evaluation without these metrics feasible. The methodology proposed in our work reduces the manual effort needed to create a challenging dataset. It supports the quality assessment with the worst edge case data point for each candidate IDSs.

## 5.3 Methodology

This section presents our methodology at a high level and discusses the concepts without implementation details. To make these abstract ideas more tangible, Section 5.4 fol-

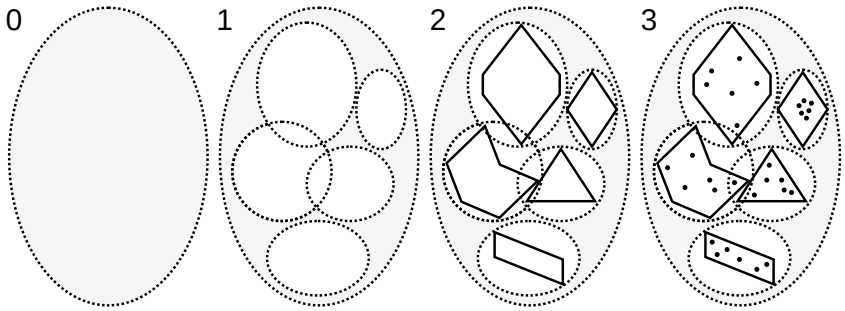


**Figure 5.2:** *Schematic model of the dataset deduction from the space of all datasets (from left to right): (0) the space of all datasets, (1) partitioned with functional scenarios, (2) further refined with logical scenarios, and (3) the concrete data points elicited through optimization.*

lows the same three steps to present an in-depth case study. Please note that the fitness functions are an independent contribution usable without the rest of our methodology. Nevertheless, for the text flow, we decided to introduce them in Section 5.3.5 within the scoping of the methodology.

### 5.3.1 General Overview

Our methodology provides a final validation before deciding whether to deploy an IDS implementation within the system. In theory, any concrete system implementation and an attacker model define the set containing all possible system behavior and all possible interference of the attacker. We refer to this set as *dataset space* as it is a superset of all potential benchmark datasets. In other words, any benchmark dataset is a diverse and broad selection of data points from this dataset space. Due to the nature of the system environment and partial knowledge during engineering, this space is only a theoretical concept. Vague textual formulations without details describe this space, e.g., "an attacker inserts new messages into the system's network communication".

As depicted in Figure 5.2, the first two steps systematically span and refine the dataset space. In the final third step, an optimizer elicits critical data points that particularly challenge the IDS in this system. In **step 1**, the deduced functional scenarios partition the dataset space into subspaces, each spanning a particular functionality of the system and a particular type of manipulation by the attacker, e.g., the attacker manipulating specific information on the network while the system is in a specific state. Depending on the use case, these subspaces might unavoidably overlap, but our methodology aims for broad coverage by systematically creating numerous partitions. In **step 2**, introducing parameters and their domains concretizes each subspace to form logical scenarios, e.g., the attack begins between n and m seconds after a specific event. With broad domains, these parameters may include the entire unrestricted subspace. Still, for more efficient evaluation, we recommend reducing the spaces to reasonable system operations, e.g., to conditions with guaranteed safety or expected system operation. Finally, in **step 3**, optimizations with specific fitness functions identify concrete scenarios from each spanned subspace. Each concrete scenario specifies a data point for the evaluation, e.g., the system's operation

in a specific setup and situation with a concrete step-wise manipulation of the attacker. They show particularly relevant behavior within the subspace as they are optimized to stress the classification of the IDS towards edge cases. This results in a benchmark dataset containing only a few but particularly critical points. If needed, the fitness functions can further compare and rank points of different subspaces to a single data point for each fitness function. This dataset is too small to replace a traditional benchmark dataset but small enough to be analyzed in detail by domain experts. Their investigation might result in new insights and provide final confidence in deploying the IDS under analysis.

Please note that none of these steps depends on explicit modeling or knowledge of the IDS under analysis. The dataset space is spanned relative to the system and considered attackers but is invariant for any IDS mitigating these threats. In other words, domain experts never explicitly name edge cases in the IDSs' classifications. Only the optimization in step 3 investigates the IDS and its peculiarities. However, the fitness functions as objective measures ensure the optimizer converges to critical behavior without biases. Furthermore, the optimizer automatically identifies irrelevant parameters and focuses on the remaining relevant for the fitness of the concrete scenarios. Therefore, the generated data points are specific to one IDS, but the methodology presents equal challenges for analyzing competing IDSs.

We intend the steps of this methodology not as a one-time effort but as an iterative process that increases the quality and understanding of the IDS under analysis. The identified weaknesses and highlighted behavior of the IDS to the specific system and attacker behavior require investigation by domain experts who decide on the respective actions. If the identified weaknesses are critical for the use case, the detection mechanism within the IDS needs to be adjusted (e.g., retraining the model or adding a filter). This results in a new IDS prototype that must pass another evaluation, including our methodology. If the spotted points show irrelevant samples of behavior, i.e., should not be part of the dataset space, adjusting the parameter space in step 2 or fitness functions in step 3 can exclude or avoid this behavior in the analysis. Both outcomes and their corresponding changes imply a reiteration of the methodology, yielding different data points and a follow-up investigation.

### 5.3.2 Preparation: Find System and Attacker Models

Our methodology spans the space of all datasets using models of the system and the potential attacker. There is no requirement for a specific model notation or level of formalization in these models. A sound engineering process of the system or IDS under analysis should already provide this information and models. Every model is suitable as long as it answers these questions: • What states of operation can the system go through? • What information does the system process in what value ranges? • Which properties of the system need protection against an attacker? • What manipulations can the attacker conduct within the system and its data? • When can an attacker's influence on the system be tolerated, and when can it no longer be ignored?

### 5.3.3 Step 1: Partition the Dataset Space

Functional scenarios are the first fixation and partition within the dataset space. Each functional scenario defines the scope for potential system and attacker behaviors and focuses on one particular action within the system and by the attacker. They differ in the system's context, the set of signals processed by the system, or the different strategies of the attacker. We use the term signal to refer to the smallest logical unit of information packed into a message transmitted on the network, e.g., a Boolean flag, a counter, or value, but not individual bits representing them in parts.

A set of functional scenarios ideally covers the system and attacker behavior as extensively as possible and in all relevant aspects. Broad coverage is desirable, as it increases the chances of discovering unanticipated system and attacker behavior later during optimization. We recommend using many smaller subspaces, as they are easier to deduce, analyze, and optimize than a single abstract description of complex behaviors. Combining scenarios and optimizing for the best fitness values over multiple scenarios is possible. For example, our later experiments merge two logical scenarios that only differ in the sign of a single parameter. However, in general, we consider merging scenarios problematic as it may unintentionally add meaningless data points and complicates the exploration task for the optimizer. Only domain experts should explicitly decide after careful consideration about any reduction of the scenario space.

For our methodology, functional scenarios need to compose system and attacker behavior. We propose to independently define the behavior of the system and the attacker and combine them as a cross-product. The deduction of scenarios for the regular system behavior does not differ from the process of scenario-based system testing. Any requirements specification or existing test suites can be used as a foundation to deduce these scenarios.

We propose starting with a broad framing refined later to deduce the attacker behavior as functional scenarios. We recommend analyzing different sets of capabilities, locations, or goals of the attackers and combining them in different functional scenarios as reasonable in the use case under analysis. To refine and translate the attacker behavior into complete descriptions of actual attacks, we suggest using the notion of data or signal changes that cause reactions by the system or change the system state. Any signal, interpreted as binary or scalar, at any interface within the system can change in two directions: (1) increase or (2) decrease in its value. An attacker can use both of these signal changes in two ways: (A) suppressing the propagation of a legitimate change. Hence, the system's state remains and does not adjust to the changing environment. (B) faking a change that factually has not happened. Hence, the system transitions to a new state and does not behave according to the unchanged environment. Therefore, it is sufficient to enumerate all relevant signals in the system and analyze them for all four manipulations of an attacker.

Following these steps results in many functional scenarios, while not all are equally important. Methodologically, it is the best choice to further investigate all functional scenarios in step 2 and 3. Nevertheless, this might not be feasible within the available resources. In that case, we recommend conducting a risk analysis of all these functional scenarios and only continuing our methodology with the most critical scenarios. However, this reduces the size of the resulting benchmark data and threatens the completeness of

the analysis. This reduction inhibits the risk of the generated data points barely identifying critical behavior. Hence, adjusting this risk analysis to be more inclusive within the following iteration might be necessary.

### 5.3.4 Step 2: Introduce Parameters

Logical scenarios formalize relevant parameters within each functional scenario and define domains for them. These parameters can parameterize the physical properties of the environment or the system, high-level goals, environment behavior, or timings of events. A standard optimizer supports integer and floating-point parameters. For our methodology, these parameters must cover the relevant system parts and variations of the attacker's behavior to enable a diverse dataset. We recommend modeling the attacker behavior as an addition to the system behavior and describing both in isolation. The best method for deducing system parameters depends on the concrete domain. In our view, the first three steps of the well-known category partition method [95] for test suite generation are the most generic approach for ensuring the quality of the coverage. Describing the system's behavior is a well-studied problem, and methodologies and templates exist in various domains, for example, for our use case from the automotive domain [107].

As a starting point for attacker parametrization, we suggest minimal proof-of-concept implementations of the attacks on the system and extending them with parameters on suitable positions. On a high level for attacks, we propose three different categories of parameters: Timing, Payload, and Volume. *The timing of events*, e.g., the time of the first modification of an attacker or the duration of the manipulation. This category aims to identify the most critical point in time for the attack relative to a specific manipulation. Also, the faking and suppression of signals happen related to different events and hence different parameters. Complex interferences require a global timer and might need to encode a temporal order for introducing multiple time parameters depending on each other. Overall, we found it sufficient to specify the absolute time since the beginning of the concrete scenario as a parameter. *The used payload for the manipulation*, e.g., the aberration from the manipulated signal to its original value. This category balances the trade-off between attacks with big and small aberrations. The latter have a smaller but potentially still critical impact and are more challenging to detect by an IDS. *The attack volume*, e.g., the number of manipulated messages per time or the pattern of the manipulation. This category encodes modifications that indirectly make the attack less intrusive in potentially monitored information. These parameters are the most individual to the used attacks and subsume everything that changes the shape of the attack within the observable features. In our case study, we inject or overwrite messages and use parameters to select patterns for the frequency of the manipulations.

Like a broad scenario space, broad parameters potentially include more challenges for the analyzed IDS. Higher confidence in the evaluation requires such wide space, especially when it does not yield a critical attack or circumvent the IDS. Nevertheless, the investigated space grows exponentially with the number of parameters, so we again recommend prioritization. In our experiments (see **??**), we have only used one timing and one payload parameter after a first validation as these only showed neglectable impact.

### 5.3.5 Step 3: Optimize with Fitness Functions

Any choice of concrete values for the parameters in a logical scenario forms a concrete scenario. It describes a data point in the dataset space that can assess the performance of an IDS. An optimizer with fitness functions enables automatic identification of the most relevant data points in the following way: The fitness function rates the suitability of a data point for a specific purpose and compares it with other data points. The optimizer iteratively samples the parameter space and compares the fitness values of the selected data points for choosing the following candidates. This process repeats until it converges toward the example with the highest rating.

The core of such optimizations is the fitness function, i.e., a measure of the suitability of a concrete scenario for the intended purpose. To create benchmarks for IDS evaluation, we propose six different fitness functions covering six distinct traits of data points: Two fitness functions, $f_R$ (regular) and $f_A$ (attack), measure the functionality of the overall setup. They establish a baseline for reference in the later investigation by domain experts. Four fitness functions, $f_{FP}$, $f_{TN}$, $f_{FN}$, and $f_{TP}$, reassemble the four classifications of data points in the confusion matrix during evaluation. Instead of approximating each class with percentage scores or accumulating metrics, the optimization yields concrete examples of extreme representatives in each classification. These samples are condensed descriptions of the worst and best performance of the analyzed IDS. Their small number makes a manual assessment by domain experts feasible.

**The Optimization Problems:** Formally, the optimizer's task is to solve the following set of optimization problems:

$$\begin{aligned} \forall f \in F = \{f_R, f_A, f_{FP}, f_{TN}, f_{FN}, f_{TP}\} : \\ X = P \times A, \;\; sim : X \rightarrow S \\ \min f(sim(x)) \\ s.t. \;\; x \in X, \end{aligned}$$

where X is the previously deduced parameters space composed of parameters $P$, denoting the peaceful system behavior, and parameters $A$, denoting the attacker behavior. $sim$ describes the execution of the system or a simulation with the given parameters, returning a trace of a concrete scenario $s$ from the corresponding space $S$. F is the set of our six fitness functions. Please note that this optimization problem includes no explicit constraints except for each parameter being an element of the corresponding domain. Properties observable at runtime (e.g., if the attack was successful) do not exist before the simulation $sim$. Hence, the fitness functions model these constraints on the observed concrete scenario, and the optimizer considers the constraints implicitly through the fitness value before choosing the subsequent parameters.

**Instantiation:** Our methodology relies on two functions to be defined individually for each domain and IDS under evaluation: the success of the attack $sam(t)$ and the detailed assessment of the IDS($t$).

For the attack's success, we use impacted safety distances as continuous quantification after the attack has started. Observed time spans or distances are suitable candidates for

such direct measures. Outside the safety context, as we discuss in detail in Section 5.7.3, alternatives like damage or costs of the attack are promising. $sam(t) = 0$ denotes a successful attack—the beginning of an enforced safety violation. Positive values describe the proximity to a successful attack—a higher safety margin. In other words, the smaller the number, the more significant the impact by the attacker.

The IDS analyses all data within a trace and continuously calculates a suspiciousness score IDS($t$) based on the current observations. If desired, IDS($t$) can consider a delay for including computational performance and overhead in the analysis. For the analysis, we norm this score such that a value ≥1 indicates a yield alarm and 0 denotes behavior that fully complies with the internal model. The range between 0 and 1 denotes more anomalous behavior that is still below the threshold of raising an alarm. If the internal model of the IDS provides higher certainty in an attack beyond the threshold, the score should also exceed 1 correspondingly. Although alarms are binary, it is critical to map the internal model of the IDS to this continuous anomaly score for guiding the optimization.

**Building Blocks:** Both custom functions describe properties at a given point in time. To describe a data point, we aggregate the entire timespan within the trace and quantify attack success, assessment of the IDS, and timings. Namely, to describe the system behavior, we measure the minimal observed safety margin *sam* in a concrete scenario $s$ with $saf(s)$. Thus, $saf(s) \leq 0$ denotes a documented safety property violation in the scenario $s$.

$$saf(s) = \min_t (sam(t))$$

To compare successful attacks, $att(s)$ describes the time passing from the start of an attack to a safety violation in a concrete scenario $s$, where $t_{attack}$ denotes the point in time of the first manipulation by the attacker.

$$att(s) = \operatorname{argmin}_t (sam(t) = 0) - t_{attack}$$

To describe the classification of the IDS in more detail, we generalize the IDS assessment IDS($t$) such that $sus(s)$ describes the maximum suspiciousness score observed within a concrete scenario $s$. Complementary to enable minimization by the optimizer, $san(s)$ denotes the sanguineness of the behavior observed in the concrete scenario.

$$sus(s) = \max_t (\text{IDS}(t))$$
$$san(s) = 1 - sus(s)$$

Completely not suspicious behavior ($sus(s) = 0$) results in a sanguineness of 1, whereas alarmingly suspicious behavior ($sus(s) \geq 1$) results in a potentially large negative sanguineness. As the optimizer minimizes the chosen fitness function, $san(s)$ guides the optimizer continuously towards more suspicious behavior.

Finally, $rea(s)$ quantifies the reaction time after the IDS raises an alarm and before the attack results in a safety violation. We logically define a successful attack without an alarm as a reaction time 0.

$$rea(s) = \underset{t}{\operatorname{argmin}}\,(\mathrm{IDS}(t) \geq 1) - \underset{t}{\operatorname{argmin}}\,(sam(t) = 0)$$

**Fitness Functions:** As proposed by Hauer et al. [49], we build fitness functions following a template of 1) starting with form criteria that assure a desired behavior is part of the concrete scenario and 2) quality criteria that quantify the suitability for the desired purpose of this fitness function. The template manifests in one nested case statement for each criterion in our functions. We intentionally skip domain- and scenario-dependent outer criteria ensuring compliance to form constraints on the scenario first. All $\sigma_n$ are suitably big constant integers (e.g., $\sigma_1 = 10^2$, $\sigma_2 = 10^4$ in our experiments) to nest the fitness criteria without overlaps from the value ranges of each basic block. The fitness functions $f_R$ and $f_A$ for establishing the reference baseline are defined as follows:

$f_R$ optimizes for attack impact in the running system without active attack or considering the IDS. This function is identical to classic scenario-based testing and ensures that the system, without interference from the attacker, does not violate the security properties. In our case, $f_R$ determines the minimal safety margin in the logical scenario.

$$f_R(s) = saf(s)$$

$f_A$ optimizes for security violations with an active attacker but without IDS monitoring. It confirms that the attacker's behavior successfully manipulates the system, in our case, violates the safety properties. A successful attacker should significantly reduce the minimal margin measured by optimizing $f_R$. When the optimized attack does not violate the security properties or is not considered relevant by the domain experts, the attack needs manual refinements in the previous steps of our methodology before continuing.

$$f_A(s) = \begin{cases} saf(s) + \sigma_1 & \text{if } saf(s) > 0 \\ att(s) & \text{else} \end{cases}$$

The fitness functions $f_{FP}$, $f_{TN}$, $f_{FN}$ and $f_{TP}$ describe different classifications by the IDS and are defined as follows: $f_{FP}$ and $f_{TN}$ optimize the sanguineness and suspiciousness of the observed system behavior. Most critically, the attack is disabled and not part of the concrete scenario.

$$f_{FP}(s) = san(s)$$
$$f_{TN}(s) = sus(s)$$

The requirement of a successful attack for $f_{FN}$ and $f_{TP}$ is accommodated in the nesting of multiple optimization criteria. First, both functions optimize and ensure a successful attack via $saf(s)$, then their structure differs. To model a critical attack that the IDS does not notice, $f_{FN}$ in the second step optimizes for less suspicious attacker behavior until no alarm is triggered. Finally, it optimizes for the attack with the most immediate success.

$$f_{FN}(s) = \begin{cases} saf(s) + \sigma_2 & \text{if } saf(s) > 0 \\ \begin{cases} sus(s) + \sigma_1 & \text{if } sus(s) \geq 1 \\ att(s) & \text{else} \end{cases} \end{cases}$$

To model a critical attack that showcases the detection abilities of the IDS, $f_{TP}$ in the second step optimizes for a more suspicious attack. Finally, it optimizes for an attack that is ideally immediately detected.

$$f_{TP}(s) = \begin{cases} saf(s) + \sigma_2 & \text{if } saf(s) > 0 \\ \begin{cases} san(s) + \sigma_1 & \text{if } san(s) > 0 \\ att(s) - rea(s) & \text{else} \end{cases} \end{cases}$$

With these fitness functions, an optimizer automatically navigates the generation of concrete scenarios toward particularly relevant data points. However, each elicited point requires a manual investigation in detail, as there are multiple possible outcomes: A) The concrete scenario reveals an undesired behavior of the IDS. The domain experts should report this wrong classification to the IDS developer to improve IDS. B) The observed behavior represents the desired purpose but is neglectable during operation. In this case, each step of our methodology requires adjustments to exclude such scenarios, e.g., by spanning the parameter space differently or adding a form criterion to the fitness function. C) The optimizer has not identified relevant behavior despite exhaustive iterations. This observation does not provide imminent conclusions as the optimizer only samples the space of all potential benchmark data. Nevertheless, when observing a broad coverage of the dataset space by the sampled and investigated scenarios, this observation is an indicator of the IDS operating correctly in the modeled context. The first two outcomes result in another iteration of evaluation with a refined setup, and the third outcome supports the deployment of the IDS in the analyzed configuration.

## 5.4 Exemplary Application

This section presents a case study analyzing two different IDSs for the CAN bus. It follows the same structure and enumerations as Section 5.3 introducing our methodology.

### 5.4.1 Preparation: Manipulation of Bus Communication

Our methodology's foundation is eliciting models for the defended system and the attacker. For our exemplary analysis, we focus on aspects that describe manipulations of the CAN bus. Thus, the communication on this bus is the only information utilized by the IDSs within our analysis. We also assume that a vector for the attacker to access the bus exists without further discussion. As this network is not directly accessible from outside a vehicle, this assumption skips other required exploits for a multistep attack to finally gain access to the internal car network, e.g., through the car's infotainment system.

Our case study analyzes the security of openpilot, an open source level two [113] advanced driver-assistance system (ADAS), which is our concrete system under considera-

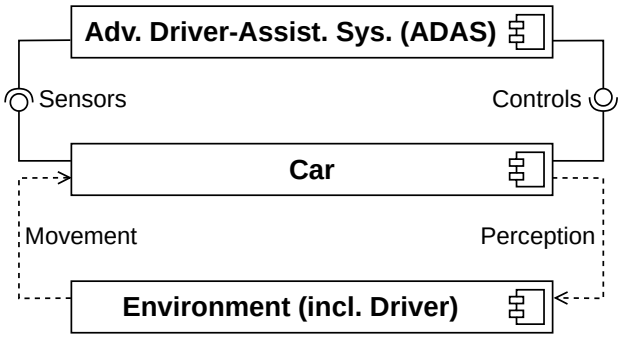


**Figure 5.3:** *Control Loop of Advanced Driver-Assistance Systems*

tion. Openpilot provides 1) adaptive cruise control (ACC), keeping the car at a specific speed while slowing down in front of obstacles or vehicles in front, and 2) lane-keeping assistance (LKAS) holding the vehicle in the middle of the lane. All assistance systems comply with the same high-level architecture of a control loop depicted in Figure 5.3. The ADAS receives information about the environment from sensors inside the car and calculates desired maneuvers accordingly. The ADAS transmits these maneuvers as control commands into the network of the car, which realizes the desired steering movements. The new movements result in differing sensor readings, which closes the control loop.

Our attacks on the ADAS tamper with this control loop in the following way: The attacker has gained access to the CAN bus and manipulates the channel transmitting information from the sensors to the ADAS. Thereby, the ADAS receives false signals that deviate from the actual sensing of reality. This wrong information results in a wrong internal model aberrating from reality. Based on this wrong model, the ADAS deduces steering commands and sends them to various actuators via the network. However, due to the enforced aberrations of model and reality, these steering commands only constitute safe driving maneuvers relative to the internal model but no longer to reality. Ultimately, this enables the attacker to craft a manipulated model that causes the ADAS to actively violate its safety properties.

To mitigate such attacks, the IDS is attached to the network and monitors the communication. The IDS should raise an alarm for each manipulation by an attacker. As we will discuss in Section 5.6, the time frame to react to these alarms is tight and only within a few seconds. We see two potential reactions: 1) immediately warn the driver to take over the steering while the ADAS turns off gracefully or 2) try to suppress manipulated signals with the car falling back to a fail-safe state. Nevertheless, the elicitation and evaluation of an appropriate reaction is out of the scope of this work.

### 5.4.2 Step 1: Deduction of Functional Scenarios

For the scenarios without attack in our use case, we use existing work about the test-suite generation for testing the regular behavior of autonomous vehicles[107]. According to the functionality of the ADAS under analysis, the functional scenarios are driving on a track without traffic, a vehicle appearing in front of the ego vehicle, and the user changing

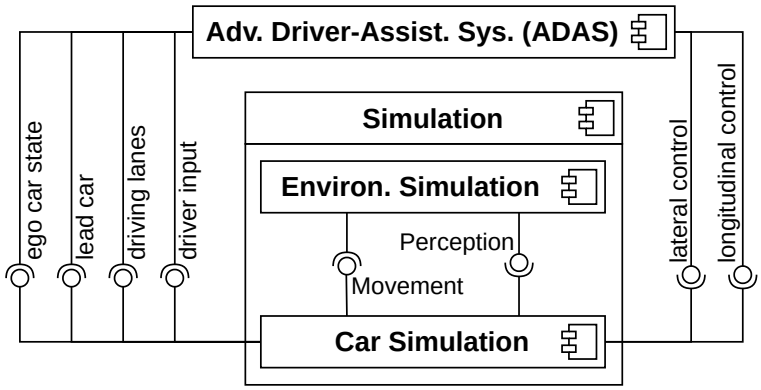


**Figure 5.4:** *Detailed control loop with the data flow of the ADAS plus complementary inputs from the closed loop simulation.*

the operation mode of the ADAS. In each scenario, the ADAS is engaged. Thus, the ACC keeps a specific speed, and the LKAS keeps the car on the lane. This selection of scenarios is simplified for our case study but could be further refined and extended. For the attacks, we use the attacker model also used to develop the IDS in our case study. An attacker has gained access to the internal CAN bus through an additional device attached to the bus (e.g., via the OBD connection) or hijacking an ECU (e.g., via a bug in the infotainment system). Manipulation is possible by injecting new messages or as a man-in-the-middle overriding existing information. In the system model, the attacker manipulates the signals sent from the sensors to the ADAS. In our methodology, we consider each signal twice: 1) with the suppression of legitimate changes and 2) with the injection of fake changes. Figure 5.4 depicts the entire data flow of our exemplary ADAS, and we selected the three sensor inputs realized on the CAN bus for our further analyses. The lane detection on our ADAS uses a separate camera.

Following our deduction of attack scenarios, an attacker can manipulate each signal in four different ways. These ports transmitting six signals results in 24 different attacker scenarios depicted in Section 5.4.2. Each field of the table names the direct impact of the manipulation and classifies its overall consequence in isolation. For example, the ADAS driving at a higher speed than desired might confuse the driver. However, the operation at this higher speed is still within the operational safety of the ADAS. An ADAS not engaging or turning off might irritate and annoy the driver. However, it is not a safety problem on autonomous driving level two, as in this level, the driver still needs to keep his hand at the steering wheel and monitor the driving all the time. Hence, these attacks have lower priority than others, which might lead to collisions or the car leaving the lane.

Finally, our methodology forms a catalog of all scenarios as the cross-product of (1) all scenarios of regular system operation and (2) all manipulations of an attacker. To showcase regular system behavior, we considered a simplistic set of variations of a) straight or single curve tracks and b) a car in front or behind—resulting in four distinct functional scenarios without attack. Overall this yields 24 times 4 equals 96 functional scenarios to analyze further in the following step. Comprehensively analyzing all these scenarios requires years of total system runtime in the optimizer and is, therefore, not feasible for

| | | suppress legitimate changes | | inject faked changes | |
|---|---|---|---|---|---|
| | | **increase** | **decrease** | **increase** | **decrease** |
| ego car state | **current speed** | car drives too fast (confusion) | car stops (rear collision) | car stops (rear collision) | car drives too fast (confusion) |
| | **steering angle** | increased steering angle (leave lane)$^{\alpha}$ | counter-steering (leave lane)$^{\alpha}$ | counter-steering (leave lane)$^{\alpha'}$ | increased steering angle (leave lane)$^{\alpha'}$ |
| lead car information | **relative velocity** | car stops (rear collision) | crash into slower car (frontal collision)$^{\beta}$ | crash into slower car (frontal collision)$^{\beta}$ | car stops (rear collision) |
| | **distance to lead car** | car stops or drives too slow (rear collision) | crash into slower car (frontal collision)$^{\beta}$ | crash into slower car (frontal collision)$^{\beta}$ | car stops or drives too slow (rear collision) |
| user input | **engaged** | ADAS stays inactive (irritation) | ADAS stays active (confusion) | ADAS activates unexpectedly (confusion) | ADAS deactivates unno- ticed (confus., leave lan.) |
| | **target speed** | car will not accelerate (irritation) | car will not decelerate (confusion) | car suddenly accelerates (confusion) | car suddenly decelerates (rear collision) |

**Table 5.1:** *Full deduction of the functional attack scenarios. Each cell denotes a functional attack scenario that is the result of a manipulation of the signal (row) in the given way (column) and holding all other factors fixed. The Information in brackets refers to the potential consequences of this attack.*

automotive manufacturers and is out of our scope. We needed detailed domain knowledge about each scenario to prioritize them by the relevance for the IDS usage. Therefore, we conducted experiments about the functional scenarios marked with $\alpha$ and $\beta$ in Section 5.4.2. We used a real car and documented our setup in Chapter 3.

### 5.4.3 Step 2: Logical Scenarios and Parameterizing

In the following steps, we only focus on and discuss the manipulation of the steering angle (the scenarios marked with $\alpha$ in Section 5.4.2). We found this manipulation to be gradual and the least noticeable by a human driver, i.e., to have the highest need for an alarm by an IDS. The driver or ADAS moving the steering wheel to the left or right results in an increased or decreased angle broadcasted by the steering wheel sensors. The ADAS uses this signal in a feedback loop: For keeping the lane or driving a curve, it calculates a suitable angle and triggers motors to apply slight torque on the wheel for turning it. If the sensor messages report the approximation of the desired angle, the ADAS lowers this torque; if the divergence does not shrink, the torque is increased based on an internal model. The corresponding attack is the following: An attacker fakes the sensor signal, reporting that the steering wheel is off to the left of its actual angle. This results in the ADAS requesting a movement of the steering wheel to the opposite, right side, supposedly to keep the car on the track. In reality, this movement pushes the steering wheel to the left and out of the angle that matches the desired trajectory. Consequently, this manipulation causes the car to unintentionally leave the lane on the left. An identical manipulation to the right causes mirrored behavior.

We elicited reasonable variables and domains based on our experiments within the real car. For this attack, we describe the logical scenario with the parameter space $X = P \times A$ as follows:

$$
\begin{aligned}
P = &[32, 33, \ldots, 96]\, km/h \times \{\text{straight}\} \cup \\
&[32, 33, \ldots, 72]\, km/h \times \{\text{curve}\} \\
A = &[0, 30]\, s \times [0, 1, \ldots, 2095] \cdot 0.0573° \times \\
&[0, 1, \ldots, 5]\,\text{messages} \times (\{0\} \times \{\text{replace}\} \cup \\
&[1, 2, \ldots, 19] \cdot 0.05\ \text{T} \times \{\text{inject}\})
\end{aligned}
$$

Set P describes the peaceful operation of the car. The parameters are used the same as for testing of the ADAS: *the target driving speed* as an integer of km/h (transmitted over the bus) and *the curvature of the street* (set in the simulation software). For simplicity, we chose two setups: driving autonomously, with speeds from 32 km/h to 96 km/h on a straight street and speeds from 32 km/h to 72 km/h in a slight curve. Set A is the parametrization of the attacker and follows our methodology on timing, payload, and volume. The time is a rational number in seconds after the concrete scenario starts denoting *the point when the attacker manipulates the first message*. This single time generically specifies the attack's timing without considering any system behavior within the scenario. However, during space exploration, the optimizer will adjust this parameter without manual modeling to

the most critical point, e.g., when the car reaches the curve for all driving speeds and road setups. The payload for this attack is *the aberration from the actual steering signal* from 0° to 120° in steps of 0.0573°. This step size is the equivalent of changing one bit in the fixed-point representation of the steering angle signal. The attack volume is represented by one parameter for replacement attacks and an additional parameter for injection attacks. The first and common parameter is an integer that denotes *the manipulation of only every n-th original message*. The additional parameter is the time as a floating-point number denoting *the delay of the injected message from the original message* transmitting the signal. Such manipulation results in an alternating sequence of values, mixing actual and manipulated signals. For injections, any fake message received after an actual message overrides the internal state until the next actual message is received. We split the regular time interval $T$ between two messages into 20 spans, and the parameter selects a multiple. A short delay causes the fake information to be dominant most of the time for decisions of the ADAS. In particular, this dominance unfolds if the attacker injects directly after consecutive messages. Nevertheless, this type of manipulation is more obvious to detect by potential IDSs.

### 5.4.4 Step 3: Instantiating the Fitness Functions

In the final step, we need concrete implementations for $sam(t)$ for each logical scenario and IDS($t$) for each IDS under analysis. Please note that this is the only time our methodology considers the IDSs under analysis. In our use case, $sam(t)$ is defined via the physical properties of the car and corresponding safety margins. In scenarios with the risk of front and rear collisions, we define $sam(t)$ as the distance to the safety margin where a human driver can still avoid a collision. In scenarios with the risk of leaving the lane, we define $sam(t)$ either as (1) the distance of the lane marking to the wheels to include weaker attacks or (2) the distance of the lane marking to the center of the car to narrow the analysis to critical attacks. Scenarios involving the attacker's impact on the driver are out of our scope, but, for example, approximations of human reaction times and awareness levels are promising definitions of $sam(t)$.

For our experiments we analyzed two state-of-the-art IDSs, $\text{IDS}_T$ proposed by Taylor et al. [131] and an $\text{IDS}_M$ by Marchetti et al. [77]. $\text{IDS}_T$ uses Long Short-Term Memory (LSTM) networks processing blocks of 20 consecutive messages on the network of the same signal to predict the following message to be transmitted. It calculates a loss and anomaly signal based on the difference to the actual message received next. An anomaly signal equal to 0 denotes a total match with the prediction and, thereby, normal behavior. In contrast, higher numbers denote more unanticipated, anomalous behavior and, thereby, a higher likelihood of an ongoing attack. For $\text{IDS}_T$, we normalized the anomaly signal's value at the threshold for an alarm. $\text{IDS}_M$ monitors the header entropy yielded from all arbitration IDs within non-overlapping time windows. The model describes an attack with the observed entropy being outside an interval of plus/minus a fixed number of standard deviations around the mean observed entropy in the training set. For $\text{IDS}_M$, we normalized this interval through the distance from the middle of the interval and the closest side.

## 5.5 Experimental Setups

The most challenging aspect of experiments in the chosen use case is the safe conduction of the attacks while keeping the experiments with potentially fatal outcomes realistic. We decided to conduct the central experiments for evaluation with HIL simulation as described in Section 5.6 in full detail. In addition to using them with the simulator, we used the same attack setup in the real network of a real car. Most importantly, we kept the hardware components of the simulation including the driving assistant system and the manipulations of the attacker identical inside the HIL simulation and the real car. Without further modifications, all attacks still successfully manipulate the real driving behavior almost identically. In other words, our attack setup and scripts work interchangeably in simulation and reality. With this design, it is of the highest criticality that the simulated and the real car show the same physical behaviors and timings. Hence, we conducted a supplementary set of experiments to ensure this and underline the realism of values for impact on the safety margins that we finally obtained in the simulated experiments. Section 5.5.2 describes this validation inside the car after the next subsection has introduced the utilized hardware.

### 5.5.1 One Set of Attack Hardware for Reality and Simulation

While modifying a car it is of vital importance to ensure the full operation of the car and to exclude unwanted side effects. Therefore, we favored a setup that minimizes the modifications inside the car and on the EON. We designed an adapter and cables compatible with the internal CAN cables inside the car. Our adapter has two configurations: 1) splitting the bus physically in two parts, giving a Man-in-the-Middle attacker complete control over the communication, and 2) adding a new node to the bus, enabling injection-based attacks. The three cables within the vehicle transmitting the data relevant to the EON are accessible behind the rear mirror. Figure 5.5 depicts our adapter: The ODB-II connectors link the EON via three CAN buses to the car or the car simulation. For normal operation with this adapter, we connected each D-sub 9 connector with a gender switcher to close each bus. For Man-in-the-Middle attacks, we used up to six D-sub 9 to USB adapters and corresponding Python scripts forwarding and modifying selected messages. For injection-based attacks, we combined a pair of D-Sub 9 connectors with a three-way bus segment adding another D-Sub 9 connector utilized for one D-sub 9 to USB adapter. Due to the direct connection in the bus segments, this setup for injections doesn't depend on forwarding messages and induces only minimal delay.

In the HIL, we combined the unaltered EON and the attacker connection (identically used for the validation in the real car) with a versatile car simulation. We built a physical CAN bus mocking the real car connecting for the simulation to expose the same attack surface. Our simulation combines the simulation software IPG CarMaker with a custom bridge to the CAN bus for realistic network communication. The bridge uses the same message matrix as the real car and translates information between the simulation and the bus communication. In addition to the signals transmitted over the CAN bus, openpilot uses a camera to identify the lane on the street in front of the car. To simulate this, we

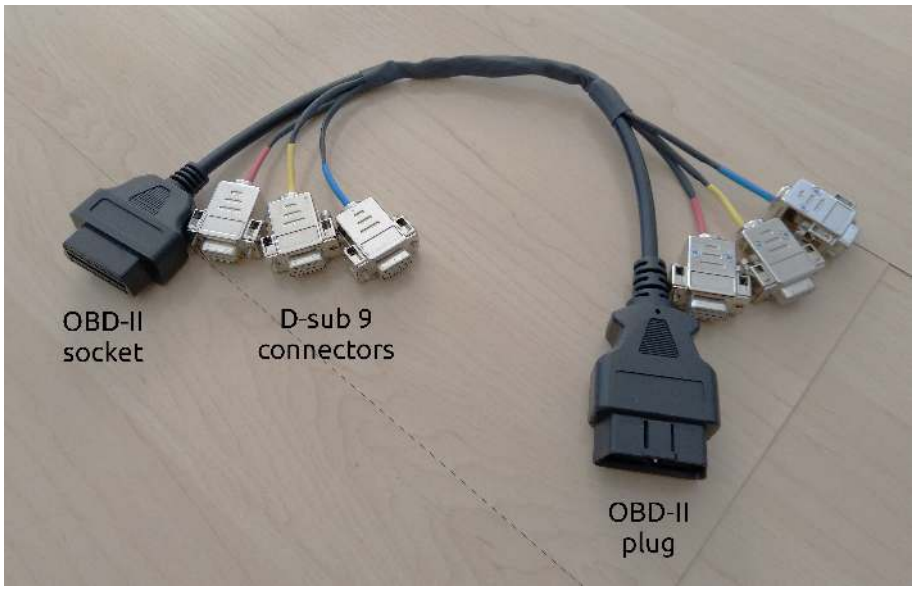


**Figure 5.5:** *Our custom adapter for conducting attacks on the CAN bus. Three distinct can buses are connected via the two OBD-II connectors for compatibility with the EON and the car. The colors mark three pairs of interception points to each bus with the corresponding D-sub 9 connectors.*

positioned the EON in front of a monitor (see Figure 5.6) that shows a real-time view from the ego vehicle in the simulation. While constructing the HIL, we focused on keeping the attacker's behavior realistic and the system boundary between the EON and the car untouched. This ensured that our attack setup and scripts worked interchangeably in simulation and reality.

### 5.5.2 Validation and Calibration with a Real Car

For maximal realism and relevance of the spanned dataset space, we studied the behavior of openpilot in a real car with street homologation[1]. We considered three groups of functional scenarios: $\nu$) regular driving behavior of the ADAS at various speeds, tracks, and without attack. Furthermore, we investigated all functional scenarios leading to $\alpha$) leaving the lane or $\beta$) a frontal collision. For the scenarios in group $\alpha$, we manipulated the steering angle while driving on a free track. Namely, for the scenarios in group $\beta$, we manipulated the collision detection sensor on a free track and with a solid, non-moving obstacle on the track. Figure 5.7 shows our setup for testing the functional scenario $\beta$ with the EON ADAS. See the entries in Section 5.4.2 marked with $\alpha$ and $\beta$ for details on the attacks and the violated safety guarantees.

Our selection is a drastic filter of the entire functional scenario space. Nevertheless, while we found successful attacks with very few messages and manipulations, we also found indications that car manufacturers cannot omit the analyses of the entire space

[1]We intentionally do not disclose the manufacturer or car model. The investigated vulnerability applies to any CAN bus and not particularly to openpilot or the car. Our attacks alone are not weaponizable and are already known to the car manufacturers we have talked to.


**Figure 5.6:** *A view on our Hardware-in-the-loop setup, showing the EON (the smartphone in the front) and the carmaker simulation (the monitor in the back) running a concrete scenario.*

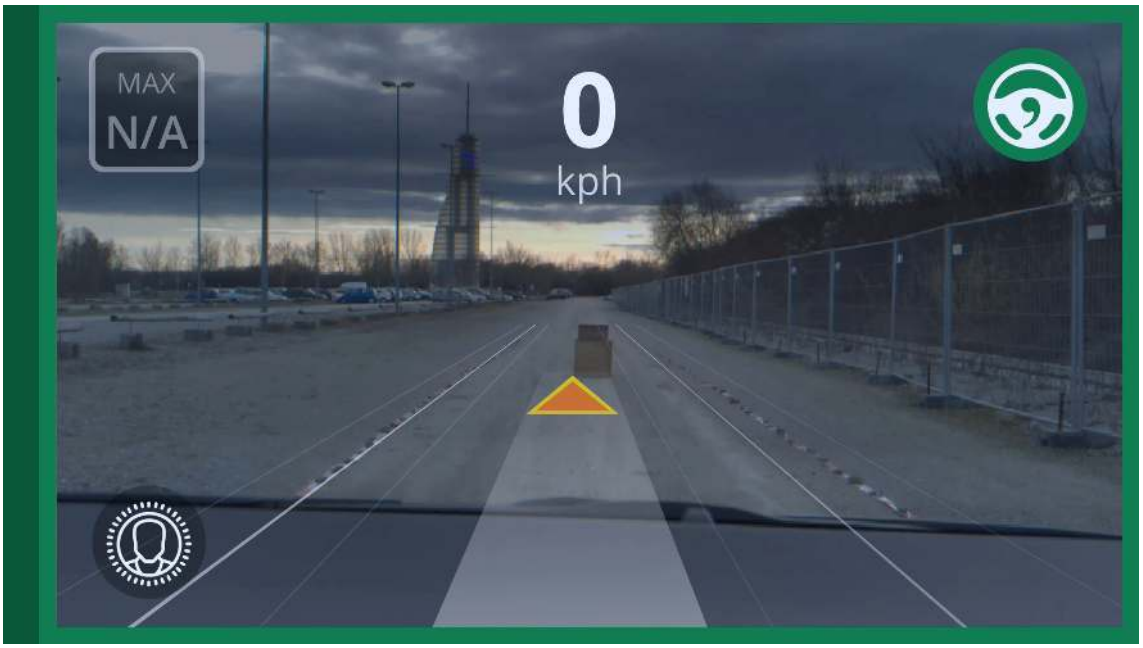


**Figure 5.7:** *The EON's view on our testing track with correct detection: a soft foam box in the middle of the lane and front of the vehicle.*

in detail for a complete security analysis indicating a flawless IDS classification. Several manipulations look identical on a higher abstraction level but differ fundamentally from a network perspective. For example, to extend the fixed 8-byte size of a standard CAN message, the ADAS combines consecutive messages through matching included counters, e.g., for information about multiple obstacles infront of the car. After receiving a specific counter variable once, the system drops all packages with a lower counter. Hence, an injected package with the increased counter invalidates all original information the sensors transmit with a lower counter. This design escalates the capabilities of a supposedly more harmless attacker just injecting messages to fully replace all signals.

In our investigation, we followed several safety considerations [54] and restricted the scenario space. The drivers of the car (i.e., we) were aware of the moment when an attack would happen and intentionally did not interfere. This behavior differs from a real-world driver, who might react unpredictably. While attacking the car, we drove not faster than 10 km/h, used soft obstacles, and had a flat surface outside the lane. These experiments resulted in a set of recorded network traces and videos of 1) normal driving (group $\nu$) and 2) ongoing attacks during automated driving (group $\alpha$ and $\beta$). The ADAS steered the car flawlessly in group $\nu$. In group $\alpha$, the manipulated steering signal forced the car aside from its planned trajectory. However, due to the low speed, leaving the trajectory took more than 10 seconds, and a human driver could interrupt the ADAS in time easily. In group $\beta$, especially the hiding of existing obstacles resulted in a crash three seconds after the start of the attack. This gradual severity increase in group $\alpha$ was the most interesting for further investigations. We used all scenarios to debug and simplify our attack scripts. Furthermore, we validated our simulation with the driving behavior in reality.

## 5.6 Experiments and Analysis

### 5.6.1 Setup and Implementation

**Enumeration of Concrete Scenarios:** For our experiments, we modeled the two similar logical scenarios (see Section 5.4.2 marked with $\alpha'$) about fake changes in the steering angle with parameters elaborated in Section 5.4.3. For a more comprehensive discussion, we optimized both tracks (straight and curve) and attackers (replace and inject) individually. Combining both tracks and attackers in a single optimization would be possible and would quarter the execution time and manual analyses. However, a combined analysis would conclude only in the single data point with the highest fitness value that is challenging to comprehend in isolation and without considering other scenarios.

In total, we ran the following optimizations: $F_R$ only depends on the system and is optimized twice - once for both tracks. $F_A$ depends on the system and attacker but not on the IDS. Therefore, we optimized it four times - once for both attacks on both tracks. The other four fitness functions depend on the IDS. Consequently, we optimized once on both tracks and with the attack corresponding to the IDS. This setup totals 22 optimizations. To conduct the experiments, we built a Hardware-In-The-Loop (HIL) simulator (see Section 5.5.1). While aiming for approximately a day of execution time per optimization, we iterated over 1000 concrete scenarios in each optimization. Each concrete scenario, on

average, constitutes 30 seconds of driving. This schedule roughly accumulates to eight days of specific driving behavior and requires a month of execution time.

**Prepartion of the IDSs:** The IDS configurations and internal models are constant in all optimizations. We trained both candidate IDSs with the same traces of regular behavior recorded from our HIL. We collected them manually and separately from all optimization. The training dataset contains traces from various speeds and six hours of driving on both tracks in our setup. We followed the instructions for parameter-tuning and optimization listed in the publication introducing these IDSs. Please note that our implementations and models are not fully optimized and just depict approximations of the IDSs, and conclusions about the quality of the original algorithms are, therefore, limited. Our focus is on the methodology for evaluating any IDS, good or bad, and not a critique of the specific IDSs we chose for our experiments.

**Optimizer:** The optimizer chooses and schedules the next concrete scenario automatically based on ratings of the fitness function of previous populations and an initial population obtained with randomly selected parameters. Our implementation is based on the Optuna optimizer framework [5] and used the Tree-structured Parzen Estimator algorithm in combination with a median pruning algorithm [19, 20]. The optimizer grows an independent population for each logical scenario, each IDS, and each fitness function without considering the other optimizations. The optimizer internally calculates the correlation of each parameter with the obtained fitness values. After each optimization, we manually investigate the traces with the highest fitness value and elaborate our findings in the following.

### 5.6.2 Experimental Results

Since the system's behavior and the IDS's classifications are highly multidimensional, we could not find any beneficial way to visualize them to comprehend the manual interpretation of individual data points. Concretely, each parameter in each scenario forms one dimension, and multiple parameters might correlate with specific classifications of the IDS and particular events within the scenarios. Furthermore, any modifications in the internal model of the IDS shift the boundaries of true and false positives and negatives within this entire multidimensional space. In our experience, this shift is unpredictable and does not follow simple correlations. We directly tailored our methodology for deducing scenarios and optimization with different fitness functions to a small set of data points that can be manually investigated and interpreted at low costs to approach this complexity. Therefore, in the following, we analyze the recordings of the concrete scenarios with domain knowledge.

**Baseline:** $f_R$: All trials confirm that the operation of the ADAS on our tracks within our chosen parameters is safe. The car does not leave the lane, and the car's center stays at least 1.6 meters away from the lane marking. $f_A$: The optimization yielded numerous traces with safety violations forced by the attacker. Every angle above 46.0° malicious aberration leads to the fastest discovered attacks, needing around 4.0 seconds and 79 manipulated messages before forcing leaving the lane. The most important hyperparameter is the manipulated angle, and the parameters driving speed and timing of the attack are only of

minor importance. Injection attacks are the more effective, the shorter after the original the attacker injects the manipulated message. With minimal delays, the manipulated information stays dominant most of the time.

**IDS$_T$:** Regular system behavior is classified correctly. However, the value of $f_{FP}$ drops by 0.2 points in the classification certainty when the car drives below 44 km/h. We see a link to some minor swinging within the lane in our simulation at these lower speeds. Optimizing $f_{TN}$ confirms this since an operation with higher speeds leads to more certain classification, at best at the top speed of 96 km/h. The analysis with $f_{TP}$ shows that the classification of IDS$_T$ depends on the steering aberration, and an angle between 97.0° and 106.8° has the fastest reaction time of 0.36 seconds. Nevertheless, $f_{FN}$ reveals undetected attacks with smaller aberrations, the worst attack with an angle of 69.0°.

**IDS$_M$:** Regular system behavior is classified correctly. $f_{FP}$ and $f_{TN}$ show no impact between any of the available parameters changing the system behavior and the detection ability of IDS$_M$. All generated scenarios were equally not considered suspicious, with an almost identical rating. An optimization of $f_{FN}$ does not identify an undetected attack, although the optimizer tried various injection timings and patterns. The optimized $f_{TP}$ converges closely to 0.5 seconds and documents that the window size (configured to 0.5 seconds) is a lower boundary for the detection.

### 5.6.3 Into the Next Iteration

After the domain experts have analyzed the optimized traces, our methodology supports the deployment decision or guides the next iteration in the development process. In general and as discussed in Section 5.3.5, there are three possible outcomes (we identify them as A, B, and C):

IDS$_T$ is an example of outcome A, an IDS suffering from a critical and relevant vulnerability revealed by the optimization. Depending on the internal detection model and the determined threshold, manipulations with a particular steering angle exist that do not yield an alarm but cause a safety violation. The optimization with $f_{FN}$ extends towards that edge point and converges at this particular angle. However, a different configuration of IDS$_T$ classifies this particular generated data point correctly. Yet, a new optimization of $f_{FN}$ for this IDS configuration results in a different data point that documents this weakness again with a different steering angle. In addition, the optimization with $f_{FP}$ guided the optimization to a set of parameters that correlate with stronger steering actions by the ADAS, most likely caused by a small aberration from the internal car model and the actual steering behavior. Although none of the generated samples results in a false positive classification, we see them as an indicator for another problem with an adjusted IDS model or in further experiments in different scenarios. Our simple track is insufficient to challenge correct IDS classifications during more complex steering maneuvers potentially described by refinements of the parametrization. These findings indicate that IDS$_T$ needs adjustments or extensions to fully protect an autonomous car.

IDS$_M$ is an example of outcome C, an IDS passing all tests generated by the optimization in our parameter space. However, this finding alone is no proof for the security of the IDS and the absence of any false classification, but only relative to the spanned and exam-

ined parameter space. The comparison of the four IDS-specific fitness functions with the baseline reveals little impact of all parameters provided to the optimizer despite a broad probing of various combinations. Such evaluation yields two questions for the domain experts: (1) Is this entire space parameter space realistic and representative, and (2) is the gained safety through the IDS sufficient? A real-world setup would require extending the parameter space and analyzing more functional scenarios. Different parametrizations describe more complex patterns of attacker timing, and the regular system behavior in our scenarios always has a perfectly deterministic communication pattern with a constant busload. Nevertheless, our optimizations already provide an approximation for the remaining safety margin. Especially, $f_{TP}$ showed that the window size within the detection model of $\text{IDS}_M$ is the minimal delay of a potential alarm. Hence, small window sizes are preferable if the detection abilities remain high. Suppose domain experts consider both properties in question as sufficiently fulfilled. In that case, our evaluation provides a witness in support of the deployment decision of an IDS configuration.

The deductions only give the idea of further steps, but we see investigating them more closely out of the scope of this work. Refining or extending $\text{IDS}_T$, suggesting a new idea with an evaluation showing the improved security, is a new distinct research effort. In the case of $\text{IDS}_M$, we expect such deeper analysis to overall support the functionality of the IDS. But more importantly, such investigation would produce valuable insights about handling the dataset space in our evaluation framework. We elaborate on possible aspects to investigate the dataset space more deeply in Section 5.7.3.

### 5.6.4 Comparison with Benchmark Datasets

The datasets generated using our methodology drastically differ from those used for the original IDS validation. Our insights are not elaborated nor mentioned in the original papers. Similar data points were not present or not noticed in the dataset. All our optimized data points are not edge cases within our parameter space that could be determined or anticipated upfront, especially if the traits and quality of the evaluated IDS are not known upfront. All this combined showcases the need for an additional evaluation considering the concrete IDSs under analysis.

Another important difference in the generated traces to the original dataset is the explicit models used for the generation. Any evaluation with a benchmark dataset only permits conclusions relative to behavior depicted by the samples in the dataset but does not permit general statements as it does not assess behavior not present in the investigated samples. Hence, as we have documented with observation 2 in Section 4.7.1, a precise and versatile attacker model is of vital importance in the benchmark dataset creation. Traditional benchmark datasets describe the sampled attacker models implicitly in verbal descriptions but do not provide sufficient concrete details to exactly replicate the generation process. In contrast, our methodology deduces the space of benchmark datasets with stepwise refinements of explicit and formal models of the attacker's behavior. These models are more descriptive and comprehend large datasets in a small set of parameters. Thus, it is easier to cover versatile behavior and oversights become more apparent. This property is underlined by the specific parameters our methodology provides in addition

to the corresponding recorded trace. These parameters comprehensively describe the behavior inside the trace and thereby provide direct support to the domain experts while analyzing the highlighted traces.

Moreover, our generated datasets document that a dataset optimized for challenging an IDS cannot evaluate other IDSs than that for which it has been generated. In particular, on the one hand, the dataset generated for $IDS_M$ does contain a broad set, probing all parameter combinations. Nevertheless, the critical points for the classification behavior for $IDS_T$ were omitted in all explored concrete scenarios. Thus, an evaluation of $IDS_T$ with the dataset of $IDS_M$ results in a wrong evaluation that misses the false negatives completely. On the other hand, the dataset optimized for $IDS_T$ only depicts focused behavior, mostly varying the payload of the attack. Furthermore, the data points with the strongest misclassification depend on the concrete internal detection model. Moreover, various other parameter combinations documenting an accurate functionality of a different detection feature, e.g., different timings as for $IDS_M$, are not provided by a dataset generated with a specific set of attacks in mind. Thus, an evaluation of $IDS_M$ with the dataset optimized for $IDS_T$ results in an evaluation with neither false negatives nor false positives but with much less actual confidence as desirable in this assessment.

### 5.6.5 Threats on Implementation

We require executable and functional implementations of the IDS under analysis for our optimizations. As the original authors did not publish their code or models, we reimplemented their algorithms and trained them on data collected from our setup. We cannot verify if and how much our IDSs differ from the original publications. As our research focuses on a methodology for the IDS evaluation in general, using slightly diverging implementations of the IDS is not problematic. However, we want to point out that a different implementation or training might not result in the identical weaknesses of the IDSs we documented.

In addition, using a HIL for our simulation reduces the realism of the observable traces. However, due to safety and monetary restrictions, repeatedly attacking a real car during operations at high speed is not feasible. Therefore, we conducted selected concrete scenarios at low speed in a real car, validated our attacks and simulation, and compared the collected traces to ensure high similarity (see Section 5.5.2). Another factor is the imperfect determinism of our simulation. Although the car operates in an identical environment among different runs, the network communication differs in timing and, thereby, in minorly different payloads. We believe hitting identical timings is impossible without a full simulation of all hardware, and we intentionally decided against that. Nevertheless, in multiple executions with the same parameters, we observed only differences within milliseconds of the otherwise identical behavior. We use an optimizer that can work with noisy fitness values and that repeats identical parameters if they seem promising to show the optimal behavior.

## 5.7 Discussion

### 5.7.1 Implications on IDS Evaluation

The evaluation of IDSs requires the analysis of an exemplary dataset as a benchmark for their performance. However, our results indicate that upfront generated datasets do not contain the data points that especially showcase all strengths and weaknesses of all IDSs they will evaluate. This shortcoming emerges from two aspects inherently missing in these datasets: (1) normal behavior of the system that is not edge case behavior for the system itself but might still be misleading for the classification by the specific detection model in the IDS, resulting in undocumented false positives, and (2) attacks with modifications within the attacker's abilities actively trying to avoid and hide from the concrete detection model in IDS while an unaltered attack is detected, resulting in undocumented false negatives. Both aspects are relative to the IDSs under analysis and, in our view, must be reflected in the evaluation. Massively increasing the size and diversity of a fixed dataset reduces the risk of missing these critical points, as they might be part of the dataset by chance. But most severely, even if the meaningful traces would all exist within a large benchmark dataset, domain experts might not notice them in the multitude of investigated classifications and the finally summed scores. Considering the exponential growth of the dataset space with the number of parameters, universal, fixed datasets are not feasible and waste resources by evaluating data points without particular meaning for the IDS under analysis.

Therefore, we argue for a cultural change in creating, publishing, and sharing benchmark datasets. A good benchmark dataset must provide means to adjust directly to the IDS under evaluation and structured information to further extend them with data points after publication. For our methodology, this information is the deduction that spans the dataset space corresponding to the dataset, including the parameter domains and a simulation environment. Ideally, a standardized and reusable format or universal catalog of guided generation methods scope these confinements. However, a standardized and reusable format or universal catalog of guided generation methods might also consider further possibilities. For example, it might include further models tailored for generating new points (e.g., Generative Adversarial Networks [82]). In the extreme, even dataset generators (e.g., [97]) are possible without any concrete dataset provided. In situations where such additional data generation is not possible, models for the synthetic modification of the datasets might be an alternative.

Finally, we highlight that domain experts are central to the evaluation process. In our experiments, we spotted attacks not detected by both IDSs under evaluation and still cause safety-relevant effects on the overall system. An example of such an attack is a MitM-attacker transmitting the messages regularly but freezing the current steering angle. In sophisticated evaluations covering large dataset spaces, ideally fulfilling some completeness criteria, undesired behavior might always exist. Nevertheless, not the entire space or shown edge case behavior is critical for the system's daily operation and requires a correct classification by an IDS. After a deeper analysis of the generated system data points, domain experts might conclude that all these remaining flaws in the IDS are ac-

ceptable and form a reasonable trade-off between security and efforts for mitigation. Our optimization-based methodology, or more specifically, the fitness functions, can be adjusted to exclude data points that are not considered relevant and explore the remaining parameter space for other critical behavior. In our view, it is impossible to fully investigate the whole dataset space and demonstrate an IDS is secure. Hence, the goal should be to support the trade-off for a reasonably secure system with sufficient evidence. With our methodology, we aim to focus the attention of the domain experts on the most relevant data points.

### 5.7.2 Limitations

The fitness functions play a central role in our methodology. They assess the suitability of the sampled traces and guide the optimization towards better samples. For this purpose, the optimization shows the best results if the fitness function describes a fine-grained, continuous descent towards the optimal samples. The utilization of safety margins as a building block in our six fitness functions provides such curves. However, this imposes a limitation on the attacks that can be analyzed.

Another example of an attack on a car is to maliciously trigger or suppress the alarm for an unclosed seatbelt. The suppression increases the risk of accidents without a closed seatbelt if the passangers are negligent. False triggers can annoy the passengers (and thus harm the reputation of the manufacturer) and might have an impact on the safe operation of a car if triggered in a critical moment. However, the impact on safety cannot be directly quantified and the nature of the attack is binary—either the alarm is triggered or not. Constructions with simulation of the worst driver reactions or counting the time of the alarm manipulation are not as expressive as the safety margins we used for $\mathrm{sam}(t)$. Although these attacks are less critical than the attacks we analyzed, our methodology cannot be applied without further modifications.

In detail, a binary implementation of $\mathrm{sam}(t)$ can still be used in the fitness functions. However, it is much less suitable to guide the optimization and the performance might be as bad as drawing random samples from the dataset space. This most likely implies infeasible minimal runtimes and invalidates all optimizations that utilize the building block $\mathrm{sam}(t)$. Namely, this directly affects the other building blocks $\mathrm{saf}(s)$, $\mathrm{att}(s)$ and $\mathrm{rea}(s)$. Consequently, the fitness functions $f_R$, $f_A$, $f_{FN}$, and $f_{TP}$ cannot be utilized anymore. Regardless, the fitness functions $f_{FP}$ and $f_{TN}$ can still be applied to assess any candidate IDS that mitigates such attacks. These limitations are the main restriction of our methodology.

### 5.7.3 Future Work

We see a direct potential to extend our methodology by including the reaction to alarms. Our evaluation approach analyses the functionality of the system and IDS as a whole and could cover automatic reactions to all alarms seamlessly. Suitable reactions, especially in safety-critical environments, are a different field of research and, therefore, out of the scope of this work. Nevertheless, our methodology could evaluate the overall quality of the IDS and the reaction using adjusted fitness functions differentiating between safety margins before and after an alarm.

As an important next step, we consider extending our methodology for space exploration of potential datasets beyond network IDSs and the domain of safety-critical cyber-physical systems. Namely, two core points need to be adjusted to generalize our methodology: (1) The schema we used for scenario deduction needs to be transferred to other domains, and (2) The building block $sam(t)$ needs to be generalized. About (1): For deducing our methodology, we followed the steps of scenario-based testing, which has been developed for the automotive domain and adopted for other use cases of cyber-physical systems. However, scenario-based testing is a specific functional testing strategy and a refinement of the equivalence class testing concept. The choice of the optimal test strategy depends on the system under analysis. Especially, concrete attack implementations are domain-specific and thus require further analysis and other technical implementations that we see out of scope from this thesis. The scope of penetration testing approaches and tools is broad [21], and we see potential instrumentation in all of them. About (2): We use $sam(t)$ to quantify the impact and success of an attack. We rely on quantifying safety violations as they are measurable directly and provide a fine-grained differentiation between each manipulation. Although this, for the moment, is restricting the number of use cases for IDSs (see Section 5.7.2 for details), we are confident that an adaption of the fitness functions for general application is possible in the future. Estimating costs caused by a successful attack or a false positive has been proposed for evaluating IDS for over two decades [38]. Replacing the safety margin with costs in our fitness function seems the most promising. Still, it requires an additional step of indirection to estimate the costs and should be evaluated further on another use case.

Furthermore, we see the need to broaden the knowledge of the space of potential benchmark datasets. With the setup of our experiments, we focused on maximal realism and validity using the realistic simulation setup and the confirmations from inside the car. Future research can implement our methodology with less realistic but faster simulation tools. Such faster tools enable more iterations and a deeper dataset space analysis than we have done. These analyses could prove the existence of convergence points for the dataset generation and IDS evaluation. Furthermore, they might identify reasonable visualization of the optimization process and the analysis of the elicited points.

Faster execution of a concrete scenario would enable another extension of the parameter space of attacker behavior. Until now, our parameters of the attacker customize the behavior of a single attack, i.e., describe how an attacker performs a single interaction in the infected system. In our experiments, this space was enough to circumvent one IDS under analysis and for both resulting datasets to diverge. Nevertheless, a realistic attacker is not bound to a single attack step but could use arbitrary combinations of different manipulations to achieve the goal. Several small manipulations might be similarly effective than a single big one but much harder to detect by the benchmarked IDS. More complex parametrization can encode such attacker behavior within the optimized parameter space. Future research should focus on identifying the most effective strategies to parameterize attacks.

We focused solely on the generation and augmentation of datasets to evaluate IDSs and separated these efforts strictly from IDS development and tuning. However, these two activities cannot be separated, as the wrong-classified samples yielded by our methodology

are valuable inputs for refining IDSs. Future work can explore these research challenges. For example, the IDS developers could simply add these data points to their training dataset for a machine learning-based IDS. The new training encodes the problematic behavior within a new detection model. However, we do not expect this process to result in a robust IDS withstanding a thorough evaluation. In the worst case, the new sample gets lost among the other samples in the training set. Such a new training procedure depends on the concrete detection mechanism. A proper encoding with the IDS might require more sophisticated feedback strategies.

### 5.7.4 Threats to Validity

The main threat to the validity of our methodology and, overall, the generated data points lies within the dependency on the appropriate deduction and modeling of the dataset space. Describing and parameterizing the regular system operation in isolation is as challenging as deducing complete test suites for the system. In addition, identifying and describing potential attacks on this system behavior is the core of penetration testing. After documenting such an attack, additional parametrization of the attacker's behavior is necessary. Too complex attacks or attacks without functional variations will result in much longer optimization times and the optimizer only spotting exactly the prepared attack. Conversely, too simplified scenario spaces will not contain exploits to an IDS and result in wrong confidence. Despite the weaknesses we spotted, we know little about the relation between our data points and the overall dataset space.

In our analysis, we chose a scenario space that, based on our experience, is around the ideal area between simple and complex. Our methodology can encode more complex attacks and combinations than we did, but our scenarios already discovered successful attacks. Furthermore, the sophisticated generation of attacker behavior is not the focal point of this work. Choosing a more straightforward attacker behavior simplifies the optimization and reduces the required runtime. Hence, future research should investigate more sophisticated attack generation approaches while spanning the dataset space and repeat our dataset comparison on the new attack samples. However, we expect these data points to show more extreme behavior and support our findings even more.

A threat to validity lies within the trade-off between the realism of the explored dataset space and the time needed to record traces from it. The runtime in our simulation and analysis is slower than the real time, and our setup does not allow beneficial parallelization through more computational resources. This overhead limits the total space we can explore but assures realism of the generated traces and is already sufficient to demonstrate the divergence in exploration. Therefore, we consider this design decision as reasonable as it is the most important first step to ensure the validity of our findings. We consider another set of experiments based on simplified simulation and variations in the dataset space as future work elaborated in Section 5.7.3.

Finally, we only conducted the experiments within a single use case related to network intrusions. However, there are generally various use cases and approaches for IDSs. We expect the evaluation of host-based IDSs to show the most extensive divergence from our findings. Although we aim to better understand the ground nature of evaluating all

IDSs, all experiments can only consider specific IDSs for specific use cases. Therefore, our findings yield the necessity of further investigations. Nonetheless, it will remain impossible to generally prove the properties of all the potential spaces for datasets for all IDSs universally.

## 5.8 Conclusion

This chapter critically reflects the process of creating benchmark datasets independent of the IDSs in the evaluation. We chose a previously well-studied use case for intrusion detection on automotive networks for our investigation. Based on successful attacks against an open-source ADAS, we used scenario-based optimization to systematically explore the space of all possible data points in benchmark datasets. Using six fitness functions that quantify the suitability of a data point for the evaluation, we generated system and attacker behavior that especially showcase edge cases for the classifications of the IDS under analysis.

Analyzing the so-generated confined dataset, we found that only a tiny part of the edge case classifications are covered when the system and attacker are considered in isolation of the IDS while collecting a benchmark dataset. Utilizing our methodology, we identified scenarios demonstrating classification behavior for each IDS that previous datasets have not shown and that the original evaluations of the two IDSs have not touched. Most importantly, these data samples are specific for each IDS and show almost no intersection. In other words, a dataset suitable for a sound benchmark of one of the IDSs is not beneficial for analyzing another.

Our finding questions the practice of using static, universal benchmark datasets for evaluating an IDS suggested afterward. Even in identical systems and attacker models, all IDSs need to be evaluated with data points specifically reflecting the properties of the IDS' detection model. Only these specific data points can emphasize edge case behavior and unique strengths of each candidate IDS under analysis missing in static datasets generated upfront. Consequently, invariant benchmark datasets are not sufficient to compare IDSs. Instead, the evaluation should follow a fixed methodology, considering each IDS individually to create tailored dataset points. Our blueprint for scenario-based dataset generation is a step toward a new culture for benchmark datasets and a well-founded evaluation methodology providing final confidence in deploying an IDS.

# Part III

# Closing Considerations

# 6 Related Work

*This chapter relates our work and contributions to other research efforts. Two distinct parts focus in higher detail on the two proposed new phases and the gaps closed in the state-of-the-art as well as in the standard evaluation schema.*

## 6.1 Regarding Taxonomies and Qualitative Attributes

The standard evaluation schema aims to create a realistic representation of benign system operations and potential malicious manipulations. This representation simulates the deployment of an IDS in the actual system and exemplifies the detection behavior. To compare the candidate IDSs, the standard evaluation schema uses metrics to quantify the observed behavior into numbers. Sorting the achieved performance of all candidates according to these numbers guides the deployment decision. In other words, only quality traits that can be measured and quantified in such metrics contribute to a decision with the standard evaluation schema.

To the best of our knowledge, Stakhanova and Cárdenas [126] provide the most complete overview of metrics used for intrusion detection system evaluation. They conducted a systematic literature study reviewing 212 publications about intrusion detection systems from seven different conferences[1] and gathered information about the means these publications used for evaluation. They found six traditional metrics (that are commonly used for the evaluation across all the publications they analyzed), seven alternative metrics (accompanied by reasons why these are not commonly used yet), and several metrics they only encountered in a single of the analyzed publications. All these metrics quantify the accuracy of the IDS's behavior classification. This study provides a strong indicator of the importance and central role of the detection capabilities of the IDSs in their evaluation. However, other research indicates that this focus might miss important details in the evaluation.

### 6.1.1 The Role of Abstract Properties in the IDS Evaluation

Mell et al. [80] consider a broader scope and list various quantitatively measurable IDS characteristics. In addition to the probability of successful detection and a false alarm, they suggest quantifying the coverage of attacks from a wide collection of vulnerabilities, the resistance of attacks that are directed against the IDS, and the ability to handle high bandwidth traffic. Furthermore, they suggest quantifying the ability to link and correlate all events from an attack, the ability to detect never-before-seen attacks, and the ability

[1] These conferences are: 1) the International Symposium on Recent Advances in Intrusion Detection (RAID), 2) the European Symposium on Research in Computer Security (ESORICS), 3) the Annual Network and Distributed System Security Symposium (NDSS), 4) the USENIX Security Symposium, 5) the IEEE Symposium on Security and Privacy (S&P), 6) the ACM Conference on Computer and Communications Security (CCS) and 7) the Annual Computer Security Applications Conference (ACSAC).

to label the attack with a common vulnerability name or assign the attack category. For network intrusion detection systems they recommend further differentiation of the detection capabilities on various densities of network traffic. They wrap up their work by hinting against other measurements without further details, e.g. quantifying the ease of use and maintenance or the resource requirements.

McHugh et al. [79] enumerate even more properties beyond quantifiable metrics and discuss the role of an intrusion detection system within an organization. Beyond the detection capabilities, they name deployment costs and ongoing operation and maintenance costs during the full IDS lifecycle. In detail, they state that support by the management in the organization is mandatory to ensure a sustained long-lived deployment. The evidence to gain this support requires efforts to identify the assets that need protection and determine actual threats against these. In addition, it needs a reflection on the organization's tolerance for loss, damage or disclosure of these protected assets. They elaborate further, that rapid changes in information technology require continuous evaluation and refinement. Moreover, any deployed IDS itself needs some monitoring to avoid attacks and distractions by attackers. Depending on the complexity of the IDS and its alarms, personnel need to be hired to administer the IDS. The Interrelations of the IDS with other means of defense need to be understood. This is in particular relevant for the placement of the required sensors inside the system to provide the IDS with an adequate input. Finally, all IDSs require the establishment of a forensic procedure to preserve evidence and how to correlate alerts with other logged information. Although all these properties follow the notion of costs related to the IDS, the majority of these costs are problematic to estimate in advance and, therefore, cannot be properly considered in the standard evaluation schema.

Tucker et al. [134] provide a concrete example of the importance of these properties in the evaluation: Two network IDSs A and B operate in the same system and aim to mitigate the same types of attacks. However, IDS A is signature-based and often indicates the type of attack and exploit, while IDS B is anomaly-based and lacks this information. In a traditional evaluation, IDS B may show better detection performance, but also considering the details in the output domain experts most likely might favor IDS A. IDS A will identify the suspicious network nodes and the nature of the attack, which enables the support staff to take specific actions quickly. IDS B, however, requires the staff to undertake further investigations before they are capable of stopping the intruder. Hence, IDS A yields lower overall costs and provides a faster reaction and is therefore the preferable IDS. A comparison with a metric on detection capabilities alone lacks this vital difference.

Over the years, various structured reviews and surveys have summarized the research field of intrusion detection systems has been summarised (e.g. [15, 35, 66, 100, 112]). The proposed taxonomies contain and link various properties to classify and contrast the broad spectrum of detection approaches. These properties range from basic discriminations between signature-based and anomaly-based approaches to their field of usage, e.g. host-based approaches vs. network-based approaches, and implementation details, e.g. the design of the internal models and the usage of different machine-learning techniques. While these review structure and summarize the research landscape, they do not analyze the potential of these properties to differentiate suitable and inappropriate IDSs for a specific use case. Furthermore, they do not investigate and differentiate the resulting

grouping of candidates to shed light on smaller differences enabling a rating of the IDSs within each cluster. In short, these taxonomies and the standard evaluation schema have not been linked together.

### 6.1.2 Previous General and Use-Case-Specific Taxonomies

The backbone of the systematic prefilter is a taxonomy of abstract properties that precisely differentiate a broad landscape of IDSs for the same use case and almost identical implementations of the same detection mechanism. Our efforts in composing a taxonomy stand in a long line of numerous previous publications that structure IDSs from a use-case-specific perspective (e.g. [10, 55, 59, 73, 142]). Section 4.5.1 provides a deep investigation of and comparison to all use-case-specific survey publications that we identified during the systematic literature study. As documented there, our taxonomy covers more properties than any of the previous taxonomies. This applies to the number of covered aspects as well as the provided details in the listed characteristics. To the best of our knowledge, our taxonomy constitutes the first work combining and opposing the industry perspective with a systematic study of academic literature.

Furthermore, to the best of our knowledge, our work is the first taxonomy with an explicit meta-model. This meta-model enables a further extension of our work in future works as well as further customization before the application of our prefilter methodology. Existing taxonomies follow an implicit tree-like structure: Various abstract aspects are positioned as child nodes from a central root node. Each of these child nodes is then refined into various, more concrete characteristics that describe a candidate IDS. Finally, we want to point out that none of these taxonomies use their structure to compare or select candidate IDSs for a specific use case. They do not provide any notion of "better" or "more suitable" among their characteristics or the taxonomy. Their research ends with a structured overview of the existing research landscape and a general investigation of potential gaps and future directions.

This broad range of IDS quality attributes is challenging to quantify in universal and realistic metrics. While there are metrics that aim to estimate the overall costs of an IDS subsuming these various factors, they require detailed knowledge and understanding of several input values that are challenging to estimate [26, 126]. An accurate measure requires experienced domain experts and relies on their ability to consider and correctly estimate all relevant factors from the entire lifecycle of the protected product and IDS before the deployment of the IDS. Although several taxonomies are structuring and classifying the properties of IDSs, their main purpose is to give an overview of the field. The presented structures show hierarchies that follow the individual decisions of the authors and do not provide an objective means to order the attributes.

**Gap 1a: No holistic catalog of abstract properties.** *There is no holistic catalog of properties that classify and differentiate various candidates for the same use case. While such properties are listed and structured in academic literature, these works only describe the research landscape but do not use the properties to rate IDSs. The view of experts from the industry about relevant properties is neglected in these academic surveys.*

### 6.1.3 Selection of IDSs by Qualitative Properties

Describing the traits of the compared candidates with abstract and qualitative descriptions contains the potential to exclude candidate IDSs very early in the decision process. As we have sketched in Section 4.2, our proposed methodology uses these properties to prefilter the considered candidates before investments in their implementation. In the following, we investigate other works about such abstract properties and highlight the relation of such properties to the standard evaluation schema as seen in other research.

Tucker et al. [134] propose an initial differentiation of the candidate IDSs before a detailed evaluation and comparison. They suggest a two-dimensional taxonomy classifying intrusion detection systems by their output (differentiating five different levels) and the system they operate on (differentiating four different scales). Any detection approach covers a differing area in this two-dimensional space that they refer to as footprints. They suggest comparing candidates on a higher level based on their footprint first and only conducting further evaluations in the overlapping areas. For this purpose, they suggest linking particular metrics and setups to each combination of these two dimensions. Such evaluation follows the standard evaluation schema and efficiently focuses on comparing directly competing approaches. In comparison to our prefilter, their taxonomy is much more simplistic and minimalistic. Furthermore, Tucker et al. do not completely exclude any candidate as unsuitable based on their properties but only aim to reduce and guide their evaluation to the meaningful experiments following the standard evaluation schema.

Biermann et al. [22] deduce a more sophisticated taxonomy of comparison criteria. They differentiate different approaches for anomaly-based and signature-based (called misuse detection in their publication) detection. The addition of a second dimension they call comparison criteria enables them to characterize abstract types of detection approaches and deduce their strengths and weaknesses. This second dimension covers the used data, the types of detected attacks, details about the assumed network and system architecture, and the different granularities of alarm. However, their work only compares abstract types of IDSs and does not work with concrete implementations or approaches. They do not provide direct means to compare candidates nor guidance for the decision process of domain experts. Their work focuses on a presentation and discussion of generic trade-offs among representatives of each abstract type of IDS. Our taxonomy comprises the properties of their taxonomy and our talks with the industry experts resemble large similarities to the trade-offs described by Biermann et al. Nevertheless, being able to investigate exemplary IDS and the continuous link to concrete realizations during the selection process was a vital benefit for the industry experts in our case study.

### 6.1.4 Candidate Comparison by Abstract Attributes

When seen on a higher, more abstract level, any prefilter in our proposed methodology is a solution to a multiple attribute decision-making (short: MADM) [136] problem. The mapping of all candidates to our taxonomy denotes a large criteria catalog, the decision matrix, and the domain experts face the task of accessing and rating the trade-offs and priorities to decide on their preferred candidate. The relatively large amount of criteria and detection candidates inhibiting various trade-offs requires a sophisticated analysis with sophisticated methodologies from this field of research. Alamleh et al. [6] provide a

systematic review of MADM applications in the field of intrusion detection and all related decisions. In the following, we want to highlight and relate to work that particularly focuses on the evaluation of IDSs.

In comparison to previous work about systematic decision-making, our work follows an important differentiation. Previous work has followed the standard evaluation schema while applying several metrics [126] and quantifications of the observed performance of the candidate IDSs. While these results are challenging to interpret and MADM provides the means for a sound decision, these criteria only cover a fraction of what domain experts should consider. To the best of our knowledge, our approach is the first to strictly separate the rating of the candidate IDSs from every observation made during the standard evaluation schema. This separation is critical to provide a prefilter saving costs before the implementation and execution of the IDS. In the following, we will distinguish between approaches that purely use attributes obtained from a metric after executing the standard evaluation schema and approaches that also consider further attributes.

Approaches that solely consider metrics obtained from the standard evaluation schema follow the same setup. After the analysis of the same datasets with the considered candidates, they form a decision matrix based on selected metrics from the performance calculation. Ultimately, an MCDM technique is used to rank the considered candidates to identify the best-performing IDS among them. For example, [98] use the MCDM technique TOPSIS, VIKOR and PROMETHEE to compare the classification of the three different attacks in the KDD99 dataset (among three other classification use-cases). Ultimately, they use all three techniques to calculate a ranking of the performance of seven different classification algorithms. Rejimol Robinson and Thomas [105] use multicriteria decision aid to minimize type 1 errors (false negatives) and type 2 errors (false positives) observed with the standard evaluation schema. Ultimately, they decide on the best approach among ten machine learning-based algorithms detecting DDoS attacks in three different datasets. Panigrahi and Borah [96] use the MCDM technique TOPSIS to accumulate thirteen different performance evaluation metrics. Ultimately, they decide on the best approach among three classifiers of the same family analyzed with two different datasets. Finally, Alamleh et al. [7] use the MCDM technique fuzzy Delphi method to compose a framework for evaluating IDSs. They compose a decision matrix with 20 evaluation criteria covering detection and performance metrics from the standard evaluation schema. Ultimately, they apply three different ranking approaches to decide on the best detection approach among 12 machine-learning-based candidates in their case study. While all these publications document a successful usage of the MCDM techniques in the decision process, none of their used criteria is available before the application of the standard evaluation schema.

For a more accurate and expressive ranking, other work has in addition to the performance-oriented metrics also included other criteria in their decision catalog. As these works follow a similar schema as the pure performance-oriented works discussed before, we want to put a particular highlight on the other considered criteria. Ahmad et al. [4] and Ahmad et al. [3] center their decision matrix around a hierarchy tree for evaluating IDSs and Neural networks respectively. In addition to performance metrics (e.g. the detection rate), they further consider the adaptability of the IDS measured in time and cost, the minimum training and updates needed for the IDS, the maturity of the approach, and

the flexibility in handling varied and coordinated attacks. Their work provides weights for the assessment and concludes with an exemplary comparison and ranking of five concepts for intrusion detection and five neural-network-based IDSs respectively. Alharbi et al. [9] and Abushark et al. [1] both investigate the same case study comparing ten and respectively eight different IDSs installed in hospitals of Uttar Pradesh, India using the MCDM technique called fuzzy-based AHP-TOPSIS. For this purpose, both publications use the same catalog of eight criteria to describe an ideal assessment of machine-learning-based IDSs. Aside from the accuracy (their only performance-based metric) their criteria concern the complexity of the implementation and the ability to detect six different types of intrusion: Spam, Phishing, Malware, Denial of Service, Misuse and Anomalies. Their rating with the provided weights resulted in a ranking of all candidates and selection of a clear favorite in the set of candidates. All utilized decision criteria from all four publications considering criteria available before the application of the standard evaluation schema have been included, as far as applicable to our use case, in our taxonomy and correspondingly into the prefilters.

An alternative approach to handle concrete performance metrics is abstracting them to more generic categorical descriptors. An example of this approach is presented by [147], who use an MCDM-technique called MacBeth [17] to compare and rank eight different architectures for intrusion detection in cloud environments. In total, they elicit 17 criteria for describing the different quality-related aspects of an architecture spanning from risk analysis to location within the cloud application and detection capabilities. Following MacBeth they conducted a pairwise comparison of all criteria to create the foundation of a numerical scale. By describing the realization of each criterion in each architecture with categorical descriptors they calculated scores for each architecture and concluded with two preferable IDS architectures with almost identical ratings. Nevertheless, their calculation contains subjective weights that decision-makers have to adjust to their individual needs and does not constitute a universal recommendation for any architecture.

In particular, the problem formulation of MCDM aligns with the direct rating of all properties without aggregation or filtering of prefilter 1 (see Section 4.6.1 for details). Especially MacBeth [17] could provide the industry experts with more guidance and a clear methodology for the comparison of the aspects and characteristics. However, we see one major issue with this technique: MacBeth requires a pairwise comparison of all criteria. If this technique were applied to the 179 characteristics in our taxonomy, this would require roughly 16k comparisons. This amount of effort would surely not have been feasible for our industry experts. Nevertheless, a customized MacBeth that assesses the characteristics in each aspect individually first and then rates the characteristics could reduce this effort drastically and seems promising to us for future research.

**Gap 1b: No early assessment solely with properties known before implementation.** *The standard evaluation schema neglects properties of the IDS beyond the detection. No systematic methodology compares approaches solely based on properties that can be determined without implementation and execution of the IDS. In particular, the standard evaluation schema does not verify the suitability of an approach before implementation and the complete analysis of the dataset.*

## 6.2 Regarding Tailored and Confined Datasets

In the standard evaluation schema, the benchmark dataset is the most critical and sensitive artifact. The first phase in particular focuses on the collection of a broad set of benign and malicious traces representative of the system and the later usage of the IDS. Whatever performance of the candidate IDSs is observed in the later analyses is only a quantification relative to the traces contained in the benchmark dataset. Cárdenas et al. [26] point out a flaw in the traditional evaluation metrics in the standard evaluation schema: They assume that an intruder behaves similarly before and after the implementation of the IDS. Such behavior of an attacker is highly unlikely as attackers aim to circumvent all defenses including potential IDSs. In fact, the attacker has an impact on the main quantities in the evaluation, the base rate, false alarm rate and detection rate. While Cárdenas et al. propose means to quantify the robustness of a metric against such impact, they found the traditional metrics to not consider a changing attacker behavior. In our view, the resilience of an IDS against the circumvention of an attacker is a complex property that requires distinct considerations beyond a robust metric. Most importantly, the dataset should contain more sophisticated attacks that also stress stealthiness and circumvention against potential detections. Due to its central role and the large benefits of reusing datasets for comparison in academic publications, their creation and analysis have become a distinct field of research. Several aspects contribute to the quality and there exist a few approaches to ensure compliance during dataset generation.

### 6.2.1 Quality Analysis of Benchmark Datasets

Kenyon et al. [57] provide a holistic overview of the benchmark datasets for IDSs available in academic research. They list in total 28 datasets and analyze their strengths and weaknesses. Their investigations elicit several core properties of good benchmark datasets. Concretely these properties are: 1) full disclosure of the creation process with clear statements regarding all synthetic data modifications or generations and de-identification techniques; 2) Accurate and appropriate scoping and recording of all recorded events within the dataset; and 3) clear documentation of all constraints and limitations in the design or data. The most important property in their view is 4) the insurance of a broad and accurate representation of normal and threat events. About the lack of quality metrics, they namely demand future work and efforts towards "standardized metrics for measuring how realistic such data is with regard to live network and system performance, and appropriate presence and distributions of anomalies and threat patterns across various domains." [57]

Haider et al. [45] propose to the best of our knowledge the only numeric metric to quantify the realism of a dataset. They use a Fuzzy Logic System [127] to encode six factors they acquired from previous works. Namely, they describe a complete capture of audit logs, the maximum of included attacks, consideration of current attack behavior, real-world normal traffic dynamics, the maintenance of cyberinfrastructure performance during capture, and the inclusion of ground truth in the labeling process. By classifying the analyzed dataset with linguistic terms regarding these categories, their fuzzy logic

system yields the probability (between 0 and 1) of maximum realism in the dataset. Although their metric provides a quantification of the realism of a dataset, this realism is only one of the important properties of a benchmark dataset.

Aside from metric-based approaches, researchers have also systematically analyzed existing datasets to validate or improve their quality. Verma and Ranga [139] conducted a statstical analysis of the CIDDS-001 dataset. Namely, they used the k-nearest neighbor and k-means algorithms to partition the points of the dataset into clusters. They found that these clusters are almost identical to the labels about benign, malicious and suspicious behavior. Hence, they conclude that this specific dataset with the provided features is sufficient to benchmark IDSs and that suitable IDSs should be able to classify it completely correctly. Nevertheless, they do not provide lower boundaries when a dataset a dataset is not suitable for the evaluation anymore.

Sharafaldin et al. [119] conducted a similar analysis of the CICIDS2017 dataset. They analyzed the dataset with different data-mining and machine-learning approaches and confirmed the suitability for analysis. A particular focus in their work is the suitability of the features listed in the dataset. They introduced the notion of superfeatures which they obtained via dimension reduction of the original features. In their analysis, these new features enabled an improved classification compared to the original features. Both work underlines that a choice of descriptive features is of high importance for the quality of a dataset and can be verified by applying clustering with data-mining algorithms.

Previous work agrees that the quality dataset for evaluating intrusion detection systems is a challenging and multi-dimensional problem. Several publications list various properties and confirm their relevance and impact on the classification abilities of an IDS. While researchers have proposed a few metrics that measure the quality of the benchmark dataset, all these metrics measure the quality of the dataset as a whole and cannot be applied to a singular data point in isolation. The statistical analysis and clustering techniques similarly utilize the full dataset for the analysis. To the best of our knowledge, there yet exists no metric to assess the suitability of a singular data point for the evaluation of an IDS. Without such metrics, a guided systematic composition of a benchmark dataset targeting the desired properties is not possible. Furthermore, this gap prevents the guided reduction of existing datasets to its key essence.

**Gap 2a: No quality metric for individual traces in benchmark datasets.** *Previous quality assessments of benchmark datasets for IDS evaluation have only investigated the properties of the dataset as a whole. There is no measure of quality for a singular trace within a dataset nor in isolation.*

The lack of metrics blocks a direct improvement of the evaluation with the standard evaluation schema. As we discussed in Section 5.2, such metrics can directly be used to improve and refine any benchmark dataset used during the evaluation. For example, data points that are not suitable for the evaluation could be dropped from the dataset. Furthermore, while collecting new data points for the benchmark dataset such a metric could be used as a guidance towards particularly good data points that would directly increase the quality of the dataset and the evaluation.

### 6.2.2 On Benchmark Datasets

Our efforts to identify challenging data points align with the general creation of benchmark datasets for IDSs. Some intuitive factors imply an update of a dataset, e.g., the emergence of new attacks or major changes in the protected system. However, these factors are not the main reason for creating new datasets. We found two repeating patterns: On the one hand, high-impact datasets as the most utilized and investigated dataset for IDS evaluation originated from the DARPA Intrusion Detection Evaluation [70]. After closer investigation, researchers spotted various flaws [78] and proposed improved variants [76] and re-recordings of the dataset in similar ways [75, 87, 108, 120, 122]. On the other hand, in domain-specific networks, like the CAN bus in our case study, the datasets are either 1) data traces directly published aside a security analysis as a documentation of the conducted attack (e.g., [84]), or 2) are collected from papers proposing new detection approaches (e.g., [47, 67]).

Despite all efforts to fix the spotted weaknesses, the successor dataset might still contain other or newly created flaws and biases. Without an objective metric or generation methodology, only further deep analysis will identify them and propose a better future dataset. This problem applies more to domain-specific datasets since they are relatively small and, in the beginning, contain merely a few variants of the same attack. Most importantly, no follow-up analysis has investigated their suitability for evaluation as profoundly as shown to be required by the experience from previous datasets. In our work, we investigate a methodology that actively tailors the term of a "good" dataset relative to the concrete protected system, the attacker behavior, and—crucially—to the IDSs under analysis. Thus, we aim to automate the dataset generation, focusing on objectivity and excluding implicit biases.

A general issue in the datasets used for IDS evaluation is a high imbalance among the types of data points represented in the dataset [41]. Benign behavior is more dominant than malicious data points, and datasets do not represent each type of attack with the same amount of data points. These imbalances impact the obtainable scores in the evaluation. Large datasets especially underrepresent critical attacks that remain unnoticed in the scores. Machine-learning algorithms, therefore, might tend to ignore these attacks completely [115]. Our methodology focuses the evaluation on a few data points that showcase the edge cases in classification. Each logical scenario (see Section 5.3.4) always results only in three benign and three malicious data points. The attacks always focus on the most impactful attack among the modeled attacks. Therefore, balance is no problem in our methodology.

In this context, Apruzzese et al. [12] promote the cross-evaluation of an IDS with data points from multiple datasets. While such evaluation provides additional insights compared to the individual dataset alone, our methodology fosters the challenge in the evaluation even more. First, only data points in the original datasets are in any combination. The combination compensates for oversights in a few datasets but does not improve on behavior that none of the datasets included. Considering our evaluation, this particularly applies to the edge case behavior we pinpoint with our methodology that was not present in the original dataset. Furthermore, the unfiltered combination of datasets fur-

ther increases the number of classifications in the evaluation, increasing the problems of imbalance and oversights our methodology aims to mitigate. The systematic deduction of datasets offers a different perspective to the conceptual model for cross-evaluation. Datasets are samples from a spanned dataset space. Hence, instead of merging the sample sets, we hold combining the spanned spaces and selecting samples anew to be more beneficial. Our stepwise process can combine two sets of artifacts as a direct union before using the optimizer. Despite the larger space, the optimizer still converges towards the most relevant data points for domain experts and neglects irrelevant areas in the dataset space without further adjustments.

**Gap 2b: No tailoring of the evaluation to the evaluated IDS.** *The standard evaluation schema only reflects general domain knowledge about the system and attacker in the dataset and accumulates the performance of the IDSs in numeric scores. In particular, datasets do not intentionally contain system behavior that particularly matches or confuses the specific detection mechanism under evaluation. Inspecting specific system or attacker behavior that is peculiar solely for the classification by the IDS under analysis is vital for a valid evaluation, however, such behavior remains hidden in previous evaluations.*

### 6.2.3 Attack Generation

On an abstract level, we leverage automatic attack generation to improve the evaluation methodology. Szegedy et al. [128] first introduced the idea of automatically generating adversarial examples resulting in wrong classifications on a given deep neural network classifier. This idea transfers to IDS that internally use neural networks [140] or other machine learning classifiers [101, 102]. This technique can also analyze the classifiers in generic IDS without restrictions on their internal implementations [48]. These approaches follow a general pattern: The analyzed classifier processes a feature matrix of given or random points. In multiple iterations, the adversarial sample generator computes modifications of a selected entry in the feature matrix. The classifier repeats the analysis of this new point, hopefully resulting in a change in the certainty of the classification or the assigned class. Guided by this change, the generator mutates the modifications until a generated sample results in a wrong classification.

Compared to our work, these approaches focus only on the internal classifier and the feature representation but ignore the surrounding system and potential attacker behaviors. A generated modification of the feature matrix is artificial data and not actually behavior observable in the system or by an actually performed attack. This indirection questions the practical significance of the generated samples as they might constitute impossible behavior. Our optimization of data points focuses on the entire space of potential system behavior with gradual impacts of the attacks on the system's operation. In comparison, adversarial learning aims to tamper with an individual classification or flip a single decision. Overall, we investigate a broader system scope with comprehensible parametrizations ensuring realistic samples. Furthermore, this broader scope includes diverse sources of side effects for a wrong classification. Our fitness functions also provide a means to rank and prioritize all generated attacks to focus on the most critical oversights.

Finally, our evasions equally analyze machine learning models, anomaly detection, or any black box intrusion detection method.

There is significant work about evasion techniques against IDSs [86]. The parametrization of the attacker behavior we used in our case study to generate new attacks in comparison is relatively simple. Nevertheless, the evasion of one IDS we analyzed does not require more advanced techniques. In the future, the analysis of other IDS might require more complex attacker behavior beyond the initial benchmark we implemented. Other evasion approaches suggest the usage of systematic variations of existing attacks and several automatic schemata for their generation. The optimization in our methodology can extend its guidance to general evasion techniques, for example, through optimizing the parameters in machine learning-based evasions.

### 6.2.4 Scenario-Based Testing

Scenario-based testing is a methodology to systematically choose test cases that stress and verify the behavior of a system under test. On a high level, a structured set of scenarios describes all potential behavior of the system as a whole. As defined by Ulbrich et al. [137], a scenario is a temporal development between several scenes that describe a snapshot of the environment, including scenery, dynamic objects, and all actors' and observers' behavior. Menzel et al. [81] propose three different levels of abstraction for scenarios during the development and verification of a system: functional, logical, and concrete scenarios. Functional scenarios describe the relevant entities of a domain on a semantic level and their relationship using a consistent and domain-specific vocabulary. Logical scenarios refine entities and relations of the functional scenarios into parameter ranges within a state space. Finally, concrete scenarios depict the behavior through concrete values for each parameter within the state space. Each scenario type can be consistently refined or abstracted into the other types, thereby grouping a theoretically infinite amount of concrete scenarios.

Each concrete scenario can form test cases if the expected behavior is encoded and checked during the execution. In our work, we utilize this idea to describe undesired behavior with the notion of a safety envelope [62] that enables the overall system to still react in time to prevent safety violations in the future. We used scenario-based optimization [88] to choose meaningful parameters for refining logical scenarios into concrete scenarios. This approach mutates the parameter set and automatically optimizes towards the Pareto Front according to a given fitness function. In the automotive domain, the fitness functions mainly encode safety envelopes and safety-critical edge case behavior, among other desired properties of the scenario, as a single objective. Our methodology follows the general concepts and the three abstraction levels known and approved for scenario-based testing. We put our focus on extending them to include attacker and system behavior in parallel and propose fitness functions similar to existing fitness function templates [49] that stress the advantages and disadvantages of a specific IDS under analysis. To the best of our knowledge, we are the first to combine scenario-based optimization with IDS evaluation.

# 7 Conclusions

*This chapter concludes this thesis. It summarizes this thesis, lists the key lessons learned and highlights our main results.*

## 7.1 Synthesis of Results

At the time of this thesis, the evaluation of intrusion detection systems follows the standard evaluation schema. After the implementation of all candidate IDSs, these IDSs each analyze a benchmark dataset and the quality of their classifications is rated with selected metrics. While scientific publications foster this methodology, the schema inhibits problems for practitioners in the process of choosing an IDS to defend their systems. For practitioners, this methodology comes with high costs multiplied by the number of investigated candidates and remaining uncertainty about the actual suitability and utility of the selected IDS. This thesis addresses these gaps with 1) a systematic prefilter before and 2) a tailored assessment for the best candidates after the elicitation with the standard evaluation schema. These two new steps form a holistic evaluation methodology that is more cost-efficient and more scrutinizing than the standard evaluation schema in isolation.

Before implementing any candidate IDS, we suggest a prefilter based on abstract properties to assess the suitability of a broader set of candidates (see Chapter 4). This prefilter does not require any implementation of IDS and the protected system and therefore comes at a low cost and a relatively fast analysis. Hence, it enables domain experts to consider a broader set of candidates. Our experiments (see Section 4.6) document that the qualities even of a small set of similar IDSs differ sufficiently in their attributes. Overall, we discuss various ways to reduce a broad set of candidates to a small selection that promises the best outcome after implementation and evaluation.

Before the final decision and the deployment of the best candidate, we suggest a final evaluation with a tailored and confined dataset (see Chapter 5). Our methodology for dataset generation tailors a dataset to 1) the peculiarities of the protected system, 2) modeled attacker behavior and 3) the IDS under analysis. We optimize the generated traces to equally showcase the strengths and weaknesses of the IDS under analysis. In particular, the combination and interrelation of these three aspects in the generated traces is particularly challenging and descriptive for the evaluated IDS. In our experiments (see Section 5.6), the optimization yielded datasets showcasing the key trade-offs in the evaluated detection models and successfully circumvented one of the analyzed IDSs. Such analyses provide final confidence before a deployment or hint directly toward the remaining weaknesses of the analyzed IDS.

We applied all steps of our holistic methodology for IDS evaluation on several candidates from the same continuous use case introduced in Chapter 3, namely the protection of a modern car's internal network from manipulations. With our proposed additional phases we were able to prefilter a large scope of 21 candidate IDSs to a set of promising

candidates for implementation and further investigation. Finally, we tailored benchmark datasets for two of these candidates, resulting in a detailed documentation of the detection behavior including an unknown circumvention of one candidate.

Based on all our evaluations, the IDS proposed by Marchetti et al. [77] showed the based performance during the holistic evaluation. However, the tailored traces document a lower boundary for the detection time encoded in the window size of the internal detection mode. Based on our results and findings we would recommend further finetuning between the configured timing of the internal detection model proposed by Marchetti et al. and the intended reaction on an alarm in the concrete vehicle. Only if these modifications yield sufficient protection against the attacks generated by our methodology, we would consider the deployment of this refined IDS as desirable.

## 7.2 Insights and Lessons Learned

### 7.2.1 About the Definition of a "Good" IDS

The goal of this thesis is to propose an extended version of the standard evaluation schema to identify the best IDS among a set of candidate IDSs for a specific use case. The notion of a "good" IDS is central in this evaluation. However, while everyone we interacted with during this research had an intuitive understanding of "good", we could not find a common and holistic definition of a "good" IDS. Hence, we want to state the main observations we made in this work about the contributing factors in a definition of a "good" IDS.

**Qualitative and quantitative description:** The first question about IDS quality is the suitable unit to measure and describe it. The main unit of the standard evaluation schema is a numerical value used for the ranking of candidates that is obtained via accumulating the observations from the benchmark dataset. Such metrics are the detection rate, false positive rate or other metrics combining them (see Section 2.4 for details). However, our extension puts two other units to describe quality in the focus. The prefilter defines quality with the binary classification on the possessed properties of the IDS denoted in natural language descriptions. This notion provides a qualitative analysis and relative ranking of the candidates, e.g. property A is more desirable than property B, hence an IDS with property A is more promising than an IDS with property B. However, these descriptions cannot be transferred into a numerical ranking easily (see prefilter 1 in comparison to the others in Section 4.6). The generation of confined datasets defines quality with exemplary traces for the best- and worst-case behavior. While these individual traces can be quantified in various ways (see the fitness functions in Section 5.3.5 for detail), we strongly argue against their accumulation or sampling. These traces require a full manual investigation by the domain experts. Both indicate that a definition of a "good" IDS needs to unite various notions and levels of details considering qualitative descriptions, accumulated quantifications as well as exemplary showcase behavior.

**Unification of academic and industry views:** The second question is about the aspects and traits that should be mandatory for a "good" IDS. Especially in with the prefilter, we shed light on a broad range of potential quality attributes beyond the detection behavior. In particular, we collected broad input and feedback from industry experts. Although we aimed to include all inputs from our discussions, the scope of academic publications and the views of industry experts in some parts differed widely (see Section 4.5.1 for details). Academic publications put their focus on detection capabilities and their implementation as the main property for the quality of proposed IDS. For a deployment in production by industry, the relevant properties are much broader. Many of these properties result from organizational constraints, for example, the availability of a component among multiple suppliers or existing contracts and trade restrictions. The aspects named by industry experts consider maintenance costs, flexibility for future updates and changes and comprehension of the full infection path of the attacker. All these aspects contribute to the final costs per unit as quantification of IDS quality, which we consider as the ultimate decision factor in the industry. However, we are not aware of an academic publication comparing the maintenance costs of an IDS or studying the entire lifecycle of an IDS after real-world deployment. Nevertheless, the academic and industrial ultimately should agree on the same definition of a "good" IDS.

**Handling different degrees of maturity and information:** The final question considers the correlation between the maturity of a candidate IDS and the quality. Intuitively, the data about a refined and well-tried IDS that has been used in the field is complete and a strong indicator of the high quality of that IDS. However, such an IDS has only yielded proof of its quality after a large investment and a long time. In particular, such a notion of quality is unfair to research prototypes that have not yet reached the same level of maturity and lack the information and optimizations from the application. Nevertheless, the diverse approaches and prototypes require a quality description to differentiate them in the academic discourse. Furthermore, in the very early development stages of the system or the IDS, even less information about the candidate IDSs is available. The known requirements might still be imprecise and the understanding of the system fuzzy, however, domain experts still should be able to foster security and the utilization of an IDS as early as possible. Any definition of a "good" IDS should span all these levels of maturity and available information. Only with such a definition it is possible to guide the development process towards the deployment of a "good" IDS.

### 7.2.2 About the Actual Quality of Investigated Candidate IDSs

Although the focus of this thesis is the methodology for quality assessment of IDSs, for its evaluation we also applied the methodology to various candidates IDSs for the exemplary use case. Thereby, we gained also insights into the quality of state-of-the-art IDSs. This subsection summarizes these findings.

**Requirements on automotive IDSs are exceptionally high:** At the beginning of this research, we chose our use case in the knowledge of its high requirements on potential

IDSs. Nevertheless, documenting and eliciting all requirements to the full extent depicted a level that seems almost impossible to realize. Our prefilter enumerates the requirements, but filtering the candidates reveals trade-offs and asks domain experts to prioritize. No single candidate was fulfilling all their requirements and they converged towards various complementary approaches they would combine for full protection (see Section 4.6 for details). Ultimately, our practical implementation of an attack against the safety of the car (see Section 5.5.2 for details) posed the highest requirement. With an optimized attack (see Section 5.6.2 for details), a timeframe of two seconds is sufficient to cause a safety violation endangering the passengers. The IDSs we analyzed required in the best case half a second to detect the ongoing attack leaving one and a half seconds for a reaction preventing the crash. But even with an immediate alarm, this timeframe is too tight for non-sophisticated reactions. In our personal experience, warning the driver is no solution, as realizing and comprehending the situation already exceeds this small time window.

**Non-reproducable benchmarks:** For the experiments about the tailored dataset creation, we needed a fully functional and configurated IDS. This requirement is similar to the input of the standard evaluation schema. Ideally, we would have wished for the best candidate out of such evaluation. Unfortunately, we could not find any public implementation or reference IDS for the CAN bus. Therefore, we reimplemented in total eight different IDS proposed in academic publications. Although we followed the details in the publications closely, we were not able to reproduce the performance and result documented in any of the original publications. Only the two IDSs we investigated in Section 5.6 showed reasonable results for our protected system during assessment with the standard evaluation schema. Differing peculiarities in our system and the transmitted messages to the originally used system (details that are omitted in the publications) can explain this aberration. Nevertheless, overall, we observed worse performance and some IDSs were not usable at all. This documents poor replicability of the results as anticipated by Ahadi et al. [2]. We believe that this underlines the necessity to state and differentiate minor system properties to a wider extent than the original publications did, especially when the original benchmark dataset and IDS implementation are not also published. Our detailed taxonomy (see Section 4.4) can serve as a reference to spot these details. Furthermore, this problem poses a threat to the validity (see Section 5.6.5) of our findings about the candidate IDSs.

**Inappropriate assumptions about the system and attacker:** Throughout this thesis, we often refer to candidate IDSs with "showing worse properties" or as "insufficient defense". However, we do not aim to state that any IDS is bad or useless. As stated in our core hypothesis (see Section 1.2), the quality of an IDS can only be determined relative to a specific system and attacker. As we saw during the design of the systematic prefilter, the new idea publications introducing the candidate IDS evaluate their implementations within specific test systems and only consider selected attackers (see Section 4.5.2). However, these might diverge from the concrete requirements of other industry experts and real-world usage in other contexts. Classical benchmark datasets mostly encode these

specificites and fulfill explicit and implicit assumptions by only providing corresponding examples. As we showed with the tailored generation of benchmark datasets from an abstract dataset space (see Section 5.7.1), critical data points from the full spectrum of all potential traces are only considered by chance with fixed datasets. Therefore, we see the bigger problem in assuring that valid and realistic requirements are established, documented and verified during the evaluation. Our talks with domain experts indicate that our prefilter is an important foundation to raise awareness about the diversity of these requirements. Furthermore, the tailored dataset generation is a viable means to assure that these requirements are covered within the utilized benchmark dataset.

## 7.3 Recapitulation of Significant Findings

### 7.3.1 About the Research Questions

To guide our research towards addressing the identified gaps, we stated four research questions (see Section 1.4 for details). To conclude this work, we want to briefly revisit them and summarize our answers.

**Research questions in the first phase:**

RQ 1.1: What are key attributes to describe and contrast intrusion detection approaches abstractly independent of any implementation of the system or IDS?

RQ 1.2: How can these key attributes be utilized to assess the suitability of a candidate IDS for a given use case?

**RQ 1.1:** The foundation of publications and discussions about IDSs in industry and academia are abstract descriptions of the detection algorithm and potential attacks. We conducted interviews with industry experts and a literature study scoped to a selected use case. To describe the protected system, attacker behavior and detection mechanism we elicited a taxonomy of 179 abstract characteristics describing 34 distinct aspects (see Section 4.4). However, we could not find a unified notion of their importance, a definition of key attributes or order among them. Each publication, the domain experts and objective properties like uniqueness and information gain prioritize the characteristic differently (see Section 4.5 for details).

**RQ 1.2:** These different views prevent a distinct structure of the characteristics that universally assesses the suitability of an IDS. Nevertheless, deduced from these different views, we propose four prefilters to reduce the set of candidate IDSs to a small subset (see Section 4.6 for details). All prefilters utilize the mapping of the use case and all potential candidates onto the taxonomy. However, each prefilter selects different characteristics and results in a different set of candidates. The suitability of an IDS is no general concept but depends on the concrete requirements and selected view and respective prefilter. Consequently, the importance of a characteristic in this process depends on the selected prefilter and on the mapped properties of all considered IDS candidates.

**Research questions in the second phase:**

RQ 2.1: How can the suitability of individual traces within a dataset be quantified?

RQ 2.2: How can particularly suitable traces be included reliably and efficiently while composing datasets?

**RQ 2.1:** An individual trace can serve to showcase one of several properties of the detection capabilities of an IDS. These properties align with the fields of the confusion matrix used for metrics in the standard evaluation schema. We defined fitness functions that quantify this suitability for safety-critical use cases (see Section 5.3.5 for details). Their calculation relies on measuring values about the system behavior, the attack's success and the IDS classifications over the timespan recorded within the trace.

**RQ 2.2:** Based on these fitness functions, we documented two ways to foster the inclusion of particular challenging traces within a dataset. 1) The fitness functions can rate and prioritize all traces in an existing dataset. This provides feedback for domain experts on which traces to manually investigate first and provides insights about the utility of the bundled traces. However, the inclusion of highly challenging traces depends on the understanding of the domain experts and might only be considered by chance with an unaltered collection process (see Section 5.6.4 for details). 2) The fitness scores guide an optimizer to automatically suggest new traces potentially relevant for the IDS under analysis. Our experiments document the fitness function structure and describe the dataset space sufficiently such that an automatic generation of tailored and confined datasets for the final evaluation of IDSs becomes possible (see Section 5.6.2 for details).

### 7.3.2 About the Core Hypothesis

Finally, we want to recap our findings concerning the core hypothesis (Section 1.2) of this thesis: *The quality of an IDS cannot be determined absolutely and in isolation but only measured relative to three inextricable factors of a specific use case: (1) the requirements of the system and peculiarities of the system's behavior, (2) the precise definition of the attacker model, including attacker behavior and (3) the chosen model of the intrusion encoded in the detection algorithm.*

Our prefilter accumulates various qualitative properties of an IDS. We structured the underlying taxonomy according to these three factors inside our hypothesis. By linking the taxonomy with all mentioned properties from industry interviews as well as from all academic survey publications about the use case, we found that the properties of the IDS (or defense mechanism in general) account for roughly more than half of the provided details (see Section 4.5.2 for details). Roughly one-third of the properties elaborates on the attacker and one-sixth specifies the protected system. However, we found that aspects from all three factors were vital for all prefilters to identify the most promising IDSs. In other words, none of the factors turned out to be underrepresented or irrelevant to the filter process. Thereby, all three factors are linked to the suitability and quality of an IDS in real-world usage (see Section 4.7.1 for details).

The standard evaluation schema uses the IDS performance on benchmark datasets as the main indicator for quality. Our experiments and the methodology to tailor the dataset to optimally challenging classifications. The spanned dataset space combines parametrizations of the system and the attacker substituting the first two factors in our hypothesis. The experiments showed that the parameters of both domains are related to the degree of challenge to the IDS under analysis (see Section 5.6.2 for details). Finally, our results indicate that a "good" benchmark dataset requires traces that are specific to the IDS under assessment (see Section 5.7.1 for details). In other words, benchmark datasets ideally combine and consider all three factors inside our hypothesis.

We see both facts as proof of the importance of all three aspects and as strong support for our core hypothesis.

# Part IV

# Back Matter

# List of Figures



# List of Tables

# Bibliography


[1] Y. B. Abushark, A. Irshad Khan, F. Alsolami, A. Almalawi, M. Mottahir Alam, et al. "Cyber Security Analysis and Evaluation for Intrusion Detection Systems". In: *Computers, Materials & Continua* 1 (2022). ISSN: 1546-2226. DOI: 10.32604/cmc.2022.025604.

[2] A. Ahadi, A. Hellas, P. Ihantola, A. Korhonen, and A. Petersen. "Replication in computing education research: researcher attitudes and experiences". In: *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*. New York, NY, USA: Association for Computing Machinery, 2016. ISBN: 9781450347709. DOI: 10.1145/2999541.2999554.

[3] I. Ahmad, A. Abdullah, and A. Alghamdi. "Towards the selection of best neural network system for intrusion detection". In: *International Journal of the Physical Sciences* (Oct. 2010).

[4] I. Ahmad, A. B. Abdullah, and A. S. Alghamdi. "Comparative Analysis of Intrusion Detection Approaches". In: *2010 12th International Conference on Computer Modelling and Simulation*. Mar. 2010. DOI: 10.1109/UKSIM.2010.112.

[5] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A Next-Generation Hyperparameter Optimization Framework". In: *International Conference on Knowledge Discovery and Data Mining*. Anchorage, AK, USA, 2019. ISBN: 9781450362016. DOI: 10.1145/3292500.3330701.

[6] A. Alamleh, O. S. Albahri, A. A. Zaidan, A. H. Alamoodi, A. S. Albahri, et al. "Multi-Attribute Decision-Making for Intrusion Detection Systems: A Systematic Review". In: *International Journal of Information Technology & Decision Making* 01 (2023). DOI: 10.1142/S021962202230004X.

[7] A. Alamleh, O. S. Albahri, A. A. Zaidan, A. S. Albahri, A. H. Alamoodi, et al. "Federated Learning for IoMT Applications: A Standardization and Benchmarking Framework of Intrusion Detection Systems". In: *IEEE Journal of Biomedical and Health Informatics* 2 (Feb. 2023). ISSN: 2168-2208. DOI: 10.1109/JBHI.2022.3167256.

[8] M. Alamuri, B. R. Surampudi, and A. Negi. "A survey of distance/similarity measures for categorical data". In: 2014. DOI: 10.1109/IJCNN.2014.6889941.

[9] A. Alharbi, A. H. Seh, W. Alosaimi, H. Alyami, A. Agrawal, et al. "Analyzing the Impact of Cyber Security Related Attributes for Intrusion Detection Systems". In: *Sustainability* 22 (2021). ISSN: 2071-1050. DOI: 10.3390/su132212337.

[10] E. Aliwa, O. Rana, C. Perera, and P. Burnap. "Cyberattacks and Countermeasures for In-Vehicle Networks". In: *ACM Comput. Surv.* (2021). ISSN: 0360-0300. DOI: 10.1145/3431233.

[11] R. Anderson. *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons, 2020.

[12] G. Apruzzese, L. Pajola, and M. Conti. "The Cross-Evaluation of Machine Learning-Based Network Intrusion Detection Systems". In: *IEEE Transactions on Network and Service Management* (2022). ISSN: 1932-4537. DOI: 10.1109/TNSM.2022.3157344.

[13] D. Arthur and S. Vassilvitskii. "k-means++: the advantages of careful seeding". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007. ISBN: 9780898716245.

[14] S. Axelsson. "Base-rate fallacy and its implications for the difficulty of intrusion detection". In: *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS)*. 1999. DOI: 10.1145/319709.319710.

[15] S. Axelsson. "Intrusion detection systems: A survey and taxonomy". In: (2000).

[16] S. Ayoubi, G. Blanc, H. Jmila, T. Silverston, and S. Tixeuil. "Data-Driven Evaluation of Intrusion Detectors: A Methodological Framework". In: *Foundations and Practice of Security*. Ed. by G.-V. Jourdan, L. Mounier, C. Adams, F. Sèdes, and J. Garcia-Alfaro. Cham: Springer Nature Switzerland, 2023. ISBN: 978-3-031-30122-3.

[17] C. A. Bana e Costa, J.-M. de Corte, and J.-C. Vansnick. "MACBETH (Measuring Attractiveness by a Categorical Based Evaluation Technique)". In: *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Ltd, 2011. ISBN: 9780470400531. DOI: 10.1002/9780470400531.eorms0970.

[18] I. Berger, R. Rieke, M. Kolomeets, A. Chechulin, and I. Kotenko. "Comparative study of machine learning methods for in-vehicle intrusion detection". In: *International Workshop on Security and Privacy Requirements Engineering* (2019). DOI: 10.1007/978-3-030-12786-2_6.

[19] J. Bergstra, D. Yamins, and D. Cox. "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures". In: *International Conference on International Conference on Machine Learning*. 2013.

[20] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. "Algorithms for Hyper-Parameter Optimization". In: *International Conference on Neural Information Processing Systems*. Granada, Spain, 2011. ISBN: 9781618395993.

[21] D. D. Bertoglio and A. F. Zorzo. "Overview and open issues on penetration test". In: *Journal of the Brazilian Computer Society* (2017). DOI: 10.1186/s13173-017-0051-1.

[22] E. Biermann, E. Cloete, and L. Venter. "A comparison of Intrusion Detection systems". In: *Computers & Security* 8 (2001). ISSN: 0167-4048. DOI: 10.1016/S0167-4048(01)00806-9.

[23] M. Bozdal, M. Samie, S. Aslam, and I. Jennions. "Evaluation of CAN bus security challenges". In: *Sensors (Switzerland)* 8 (2020). DOI: 10.3390/s20082364.

[24] A. Buscemi, G. Castignani, T. Engel, and I. Turcanu. "A Data-Driven Minimal Approach for CAN Bus Reverse Engineering". In: *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*. Nov. 2020. DOI: 10.1109/CAVS51000.2020.9334650.

[25] A. Buscemi, I. Turcanu, G. Castignani, R. Crunelle, and T. Engel. "CANMatch: A Fully Automated Tool for CAN Bus Reverse Engineering Based on Frame Matching". In: *IEEE Transactions on Vehicular Technology* 12 (Dec. 2021). ISSN: 1939-9359. DOI: 10.1109/TVT.2021.3124550.

[26] A. A. Cárdenas, J. Baras, and K. Seamon. "A framework for the evaluation of intrusion detection systems". In: *IEEE Symposium on Security and Privacy*. 2006. DOI: 10.1109/SP.2006.2.

[27] H. Cavusoglu, B. Mishra, and S. Raghunathan. "The value of intrusion detection systems in information technology security architecture". In: *Information Systems Research* 1 (2005). DOI: 10.1287/isre.1050.0041.

[28] B. Chaithanya, N. Dayanand Lal, K. Geetha, and G. Nida Kousar. "Pervading and exploiting can bus protocol using icsim and cansniffer". In: *International Journal of Advanced Research in Engineering and Technology* 6 (2020). DOI: 10.34218/IJARET.11.6.2020.051.

[29] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al. "Comprehensive experimental analyses of automotive attack surfaces". In: *Proceedings of the 20th USENIX Conference on Security (SEC)*. 2011. DOI: 10.5555/2028067.2028073.

[30] K.-T. Cho and K. Shin. "Error handling of in-vehicle networks makes them vulnerable". In: 2016. DOI: 10.1145/2976749.2978302.

[31] K.-T. Cho and K. Shin. "Viden: Attacker identification on in-vehicle networks". In: 2017. DOI: 10.1145/3133956.3134001.

[32] W. Choi, K. Joo, H. Jo, M. Park, and D. Lee. "VoltageIDS: Low-level communication characteristics for automotive intrusion detection system". In: *IEEE Transactions on Information Forensics and Security* 8 (2018). DOI: 10.1109/TIFS.2018.2812149.

[33] comma.ai. *openpilot*. 2023. URL: https://github.com/commaai/openpilot (visited on 11/2023).

[34] S. Craig. *Car Hacker's Handbook – A Guide for the Penetration Tester*. Ed. by C. Smith. No Starch Press, Mar. 2016.

[35] H. Debar, M. Dacier, and A. Wespi. "Revised taxonomy for intrusion-detection systems". In: *Annales des Telecommunications/Annals of Telecommunications* 7 (2000). DOI: 10.1007/BF02994844.

[36] M. Dibaei, X. Zheng, K. Jiang, R. Abbas, S. Liu, et al. "Attacks and defences on intelligent connected vehicles: a survey". In: *Digital Communications and Networks* 4 (2020). DOI: 10.1016/j.dcan.2020.04.007.

[37] G. Dupont, J. den Hartog, S. Etalle, and A. Lekidis. "A Survey of Network Intrusion Detection Systems for Controller Area Network". In: *IEEE International Conference of Vehicular Electronics and Safety, ICVES 2019, Cairo, Egypt, September 4-6, 2019*. IEEE, 2019. ISBN: 978-1-7281-3473-4. DOI: 10.1109/ICVES.2019.8906465.

[38] J. Gaffney and J. Ulvila. "Evaluation of intrusion detectors: a decision theory approach". In: *IEEE Symposium on Security and Privacy*. 2001. DOI: 10.1109/SECPRI.2001.924287.

[39] A. Gharib, I. Sharafaldin, A. Lashkari, and A. Ghorbani. "An Evaluation Framework for Intrusion Detection Dataset". In: *International Conference on Information Science and Security (ICISS)*. 2017. DOI: 10.1109/ICISSEC.2016.7885840.

[40] M. Gmiden, M. H. Gmiden, and H. Trabelsi. "Cryptographic and Intrusion Detection System for automotive CAN bus: Survey and contributions". In: *16th International Multi-Conference on Systems, Signals and Devices, SSD 2019*. International Multi-Conference on Systems Signals and Devices. 2019. ISBN: 978-1-7281-1820-8. DOI: 10.1109/SSD.2019.8893165.

[41] S. S. Gopalan, D. Ravikumar, D. Linekar, A. Raza, and M. Hasib. "Balancing Approaches towards ML for IDS: A Survey for the CSE-CIC IDS Dataset". In: *International Conference on Communications, Signal Processing, and their Applications*. 2021. DOI: 10.1109/ICCSPA49915.2021.9385742.

[42] B. Groza and P.-S. Murvay. "Efficient Intrusion Detection with Bloom Filtering in Controller Area Networks". In: *IEEE Transactions on Information Forensics and Security* 4 (2019). DOI: 10.1109/TIFS.2018.2869351.

[43] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skorić. "Measuring intrusion detection capability: an information-theoretic approach". In: *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*. ASIACCS '06. Taipei, Taiwan: Association for Computing Machinery, 2006. ISBN: 1595932720. DOI: 10.1145/1128817.1128834.

[44] A. Hafeez, K. Rehman, and H. Malik. "State of the Art Survey on Comparison of Physical Fingerprinting-Based Intrusion Detection Techniques for In-Vehicle Security". In: *SAE Technical Papers* April (2020). DOI: 10.4271/2020-01-0721.

[45] W. Haider, J. Hu, J. Slay, B. Turnbull, and Y. Xie. "Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling". In: *Journal of Network and Computer Applications* (2017). ISSN: 1084-8045. DOI: 10.1016/j.jnca.2017.03.018.

[46] S.-J. Han and S.-B. Cho. "Combining Multiple Host-Based Detectors Using Decision Tree". In: *AI 2003: Advances in Artificial Intelligence*. Ed. by T. ( D. Gedeon and L. C. C. Fung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. ISBN: 978-3-540-24581-0.

[47] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer. "CANet: An Unsupervised Intrusion Detection System for High Dimensional CAN Bus Data". In: *IEEE Access* (2020).

[48] M. J. Hashemi, G. Cusack, and E. Keller. "Towards Evaluation of NIDSs in Adversarial Setting". In: *CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*. 2019. DOI: 10.1145/3359992.3366642.

[49] F. Hauer, A. Pretschner, and B. Holzmüller. "Fitness Functions for Testing Automated and Autonomous Driving Systems". In: *Computer Safety, Reliability, and Security*. 2019. ISBN: 978-3-030-26601-1. DOI: 10.1007/978-3-030-26601-1_5.

[50] T. Hoppe, S. Kiltz, and J. Dittmann. "Security threats to automotive CAN networksPractical examples and selected short-term countermeasures". In: *Reliability Engineering and System Safety* 1 (2011). DOI: 10.1016/j.ress.2010.06.026.

[51] Q. Hu and F. Luo. "Review of Secure Communication Approaches for In-Vehicle Network". In: *International Journal of Automotive Technology* 5 (2018). DOI: 10.1007/s12239-018-0085-1.

[52] T. Hutzelmann, S. Banescu, and A. Pretschner. "A Comprehensive Attack and Defense Model for the Automotive Domain". In: *SAE International Journal of Transportation Cybersecurity and Privacy* (2019). DOI: 10.4271/11-02-01-0001.

[53] T. Hutzelmann, D. Mauksch, A. Petrovska, and A. Pretschner. "Generation of Tailored and Confined Datasets for IDS Evaluation in Cyber-Physical Systems". In: *IEEE Transactions on Dependable and Secure Computing* (2023). DOI: 10.1109/TDSC.2023.3341211.

[54] T. Hutzelmann, D. Mauksch, and A. Pretschner. "How to conduct experiments with a real car? experiences and practical guidelines". In: *Communications in Computer and Information Science* (2020). DOI: 10.1007/978-3-030-59155-7_37.

[55] O. Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby, and A. Mouzakitis. "Intrusion Detection Systems for Intra-Vehicle Networks: A Review". In: *IEEE Access* (2019). ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2894183.

[56] M.-J. Kang and J.-W. Kang. "Intrusion detection system using deep neural network for in-vehicle network security". In: *PLoS ONE* 6 (2016). DOI: 10.1371/journal.pone.0155781.

[57] A. Kenyon, L. Deka, and D. Elizondo. "Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets". In: *Computers & Security* (2020). ISSN: 0167-4048. DOI: 10.1016/j.cose.2020.102022.

[58] N. Khatri, R. Shrestha, and S. Nam. "Security issues with in-vehicle networks, and enhanced countermeasures based on blockchain". In: *Electronics (Switzerland)* 8 (2021). DOI: 10.3390/electronics10080893.

[59] K. Kim, J. Kim, S. Jeong, J.-H. Park, and H. Kim. "Cybersecurity for autonomous vehicles: Review of attacks and defense". In: *Computers and Security* (2021). DOI: 10.1016/j.cose.2020.102150.

[60] B. Kitchenham and S. Charters. "Guidelines for performing Systematic Literature Reviews in Software Engineering". In: (Jan. 2007).

[61] M. Kneib and C. Huth. "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks". In: 2018. DOI: 10.1145/3243734.3243751.

[62] P. Koopman and M. Wagner. "Challenges in Autonomous Vehicle Testing and Validation". In: *SAE International Journal of Transportation Safety* (2016). ISSN: 23275626, 23275634.

[63] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, et al. "Experimental security analysis of a modern automobile". In: *IEEE Symposium on Security and Privacy*. 2010. DOI: 10.1109/SP.2010.34.

[64] M. Kuhn and K. Johnson. *Applied predictive modeling*. Springer New York, 2013. DOI: 10.1007/978-1-4614-6849-3.

[65] A. Lamssaggad, N. Benamar, A. Hafid, and M. Msahli. "A Survey on the Current Security Landscape of Intelligent Transportation Systems". In: *IEEE Access* (2021). DOI: 10.1109/ACCESS.2021.3050038.

[66] A. Lazarevic, V. Kumar, and J. Srivastava. "Intrusion Detection: A Survey". In: *Managing Cyber Threats: Issues, Approaches, and Challenges*. Ed. by V. Kumar, J. Srivastava, and A. Lazarevic. Boston, MA: Springer US, 2005. ISBN: 978-0-387-24230-9. DOI: 10.1007/0-387-24230-9_2.

[67] H. Lee, S. H. Jeong, and H. K. Kim. "OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame". In: *Privacy, Security and Trust*. 2017. DOI: 10.1109/PST.2017.00017.

[68] J. Lee Rodgers and W. Alan Nice Wander. "Thirteen ways to look at the correlation coefficient". In: *American Statistician* 1 (1988). DOI: 10.1080/00031305.1988.10475524.

[69] H. Lin, H. Alemzadeh, Z. Kalbarczyk, and R. Iyer. "Challenges and Opportunities in the Detection of Safety-Critical Cyberphysical Attacks". In: *Computer* 3 (2020). DOI: 10.1109/MC.2019.2915045.

[70] R. Lippmann, R. K. Cunningham, D. J. Fried, I. Graf, K. R. Kendall, et al. "Results of the DARPA 1998 Offline Intrusion Detection Evaluation". In: *Recent Advances in Intrusion Detection*. 1999.

[71] H. Liu and B. Lang. "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey". In: *Applied Sciences* 20 (2019). ISSN: 2076-3417. DOI: 10.3390/app9204396.

[72] S.-F. Lokman, A. Othman, and M.-H. Abu-Bakar. "Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review". In: *Eurasip Journal on Wireless Communications and Networking* 1 (2019). DOI: 10.1186/s13638-019-1484-3.

[73] G. Loukas, E. Karapistoli, E. Panaousis, P. Sarigiannidis, A. Bezemskij, et al. "A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles". In: *Ad Hoc Networks* (2019). DOI: 10.1016/j.adhoc.2018.10.002.

[74] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, et al. "Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning". In: *IEEE Access* (2017). DOI: 10.1109/ACCESS.2017.2782159.

[75] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón. "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs". In: *Computers & Security* (2018). ISSN: 0167-4048. DOI: 10.1016/j.cose.2017.11.004.

[76] M. V. Mahoney and P. K. Chan. "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection". In: *Recent Advances in Intrusion Detection*. 2003. ISBN: 978-3-540-45248-5.

[77] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni. "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms". In: *IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. 2016. DOI: 10.1109/RTSI.2016.7740627.

[78] J. McHugh. "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory". In: *ACM Transactions on Information and System Security* 4 (2000). DOI: 10.1145/382912.382923.

[79] J. McHugh, A. Christie, and J. Allen. "Defending Yourself: The Role of Intrusion Detection Systems". In: *IEEE Software* 5 (Sept. 2000). ISSN: 1937-4194. DOI: 10.1109/52.877859.

[80] P. Mell, V. Hu, R. Lippmann, J. Haines, and M. Zissman. "An overview of issues in testing intrusion detection systems". In: (2003).

[81] T. Menzel, G. Bagschik, and M. Maurer. "Scenarios for Development, Test and Validation of Automated Vehicles". In: *IEEE Intelligent Vehicles Symposium*. 2018. DOI: 10.1109/IVS.2018.8500406.

[82] T. Merino, M. Stillwell, M. Steele, M. Coplan, J. Patton, et al. "Expansion of Cyber Attack Data from Unbalanced Datasets Using Generative Adversarial Networks". In: *Software Engineering Research, Management and Applications*. 2020. ISBN: 978-3-030-24344-9. DOI: 10.1007/978-3-030-24344-9_8.

[83] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne. "Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices". In: *ACM Computing Surveys* (2015). DOI: 10.1145/2808691.

[84] C. Miller and C. Valasek. "Adventures in automotive networks and control units". In: *Def Con* (2013).

[85] C. Miller and C. Valasek. "Remote exploitation of an unaltered passenger vehicle". In: *Black Hat USA* (2015).

[86] I. Moisejevs. "Adversarial Attacks and Defenses in Intrusion Detection Systems: A Survey". English. In: *International Journal of Artificial Intelligence and Expert Systems* (2019). Ed. by Editor. ISSN: 2180-124X.

[87] N. Moustafa and J. Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". In: *Military Communications and Information Systems Conference*. 2015. DOI: 10.1109/MilCIS.2015.7348942.

[88] G. E. Mullins, P. G. Stankiewicz, and S. K. Gupta. "Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles". In: *IEEE International Conference on Robotics and Automation*. 2017. DOI: 10.1109/ICRA.2017.7989173.

[89] P.-S. Murvay and B. Groza. "Source identification using signal characteristics in controller area networks". In: *IEEE Signal Processing Letters* 4 (2014). DOI: 10.1109/LSP.2014.2304139.

[90] M. Müter and N. Asaj. "Entropy-Based Anomaly Detection for In-Vehicle Networks". In: IEEE Intelligent Vehicles Symposium. 2011. ISBN: 978-1-4577-0891-6. DOI: 10.1109/IVS.2011.5940552.

[91] J. Nehinbe. "A critical evaluation of datasets for investigating IDSs and IPSs researches". In: *IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*. 2011. DOI: 10.1109/CIS.2011.6169141.

[92] P. Nevavuori and T. Kokkonen. "Requirements for training and evaluation dataset of network and host intrusion detection system". In: *New Knowledge in Information Systems and Technologies*. 2019. DOI: 10.1007/978-3-030-16184-2_51.

[93] F. Nielsen. "Hierarchical Clustering". In: *Introduction to HPC with MPI for Data Science*. Cham: Springer International Publishing, 2016. ISBN: 978-3-319-21903-5. DOI: 10.1007/978-3-319-21903-5_8.

[94] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom. "SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing". In: *IEEE Transactions on Vehicular Technology* 2 (Feb. 2020). ISSN: 0018-9545. DOI: 10.1109/TVT.2019.2961344.

[95] T. J. Ostrand and M. J. Balcer. "The Category-Partition Method for Specifying and Generating Fuctional Tests". In: *Commun. ACM* (1988). ISSN: 0001-0782. DOI: 10.1145/62959.62964.

[96] R. Panigrahi and S. Borah. "Rank Allocation to J48 Group of Decision Tree Classifiers using Binary and Multiclass Intrusion Detection Datasets". In: *Procedia Computer Science* (2018). ISSN: 1877-0509. DOI: 10.1016/j.procs.2018.05.186.

[97] M. Pendleton and S. Xu. "A dataset generator for next generation system call host intrusion detection systems". In: *IEEE Military Communications Conference*. 2017. DOI: 10.1109/MILCOM.2017.8170835.

[98] Y. Peng, G. Kou, G. Wang, and Y. Shi. "FAMCDM: A fusion approach of MCDM methods to rank multiclass classification algorithms". In: *Omega* 6 (2011). ISSN: 0305-0483. DOI: 10.1016/j.omega.2011.01.009.

[99] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. "Systematic mapping studies in software engineering". In: 2008. DOI: 10.14236/ewic/ease2008.8.

[100] H. E. Poston. "A brief taxonomy of intrusion detection strategies". In: *IEEE National Aerospace and Electronics Conference (NAECON)*. 2012. DOI: 10.1109/NAECON.2012.6531064.

[101] M. Pujari, Y. Pacheco, B. Cherukuri, and W. Sun. "A Comparative Study on the Impact of Adversarial Machine Learning Attacks on Contemporary Intrusion Detection Datasets". In: *SN Computer Science* (2022). DOI: 10.1007/s42979-022-01321-8.

[102] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, et al. "Adversarial Attacks Against Network Intrusion Detection in IoT Systems". In: *IEEE Internet of Things Journal* (2021). ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.3048038.

[103] G. Rajbahadur, A. Malton, A. Walenstein, and A. Hassan. "A Survey of Anomaly Detection for Connected Vehicle Cybersecurity and Safety". In: 2018. DOI: 10.1109/IVS.2018.8500383.

[104] A. Rehman, S. Ur Rehman, M. Khan, M. Alazab, and T. G. "CANintelliIDS: Detecting In-Vehicle Intrusion Attacks on a Controller Area Network using CNN and Attention-based GRU". In: *IEEE Transactions on Network Science and Engineering* (2021). DOI: 10.1109/TNSE.2021.3059881.

[105] R. R. Rejimol Robinson and C. Thomas. "Ranking of machine learning algorithms based on the performance in classifying DDoS attacks". In: *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. Dec. 2015. DOI: 10.1109/RAICS.2015.7488411.

[106] V. Renjith and A. Pradeepkumar. "Citations of the Top 100 Most-cited Papers of the Journal Scientometrics in Web of Science and its Association and Correlation with Scopus and Google Scholar Citations". In: *Library Philosophy and Practice* (2021).

[107] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer. "Survey on Scenario-Based Safety Assessment of Automated Vehicles". In: *IEEE Access* (2020). ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2993730.

[108] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho. "Flow-based benchmark data sets for intrusion detection". In: *European Conference on Cyber Warfare and Security*. 2017.

[109] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho. "A survey of network-based intrusion detection data sets". In: *Computers & Security* (2019). ISSN: 0167-4048. DOI: 10.1016/j.cose.2019.06.005.

[110] Robert Bosch GmbH. *CAN Specification*. Version 2.0. 1991.

[111] M. Rumez, D. Grimm, R. Kriesten, and E. Sax. "An Overview of Automotive Service-Oriented Architectures and Implications for Security Countermeasures". In: *IEEE Access* (2020). ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3043070.

[112] F. Sabahi and A. Movaghar. "Intrusion Detection: A Survey". In: *2008 Third International Conference on Systems and Networks Communications*. Oct. 2008. DOI: 10.1109/ICSNC.2008.44.

[113] SAE International. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2021. DOI: 10.4271/j3016_202104.

[114] F. Sakiz and S. Sen. "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV". In: *Ad Hoc Networks* (2017). DOI: 10.1016/j.adhoc.2017.03.006.

[115] S. Sapre, K. Islam, and P. Ahmadi. "A Comprehensive Data Sampling Analysis Applied to the Classification of Rare IoT Network Intrusion Types". In: *IEEE 18th Annual Consumer Communications & Networking Conference*. 2021. DOI: 10.1109/CCNC49032.2021.9369617.

[116] M. Al-Saud, A. Eltamaly, M. Mohamed, and A. Kavousi-Fard. "An Intelligent Data-Driven Model to Secure Intravehicle Communications Based on Machine Learning". In: *IEEE Transactions on Industrial Electronics* 6 (2020). DOI: 10.1109/TIE.2019.2924870.

[117] K. Scarfone and P. Mell. "Guide to Intrusion Detection and Prevention Systems (IDPS)". In: (2007). DOI: 10.6028/nist.sp.800-94.

[118] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 3 (July 1948). ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[119] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani. "A Detailed Analysis of the CICIDS2017 Data Set". In: *Information Systems Security and Privacy*. Ed. by P. Mori, S. Furnell, and O. Camp. Cham: Springer International Publishing, 2019. ISBN: 978-3-030-25109-3. DOI: 10.1007/978-3-030-25109-3_9.

[120] I. Sharafaldin, A. Habibi Lashkari., and A. A. Ghorbani. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization". In: *International Conference on Information Systems Security and Privacy*. 2018. ISBN: 978-989-758-282-0. DOI: 10.5220/0006639801080116.

[121] S. Sharma and A. Kaul. "A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET Cloud". In: *Vehicular Communications* (2018). DOI: 10.1016/j.vehcom.2018.04.005.

[122] R. Singh, H. Kumar, and R. Singla. "A Reference Dataset for Network Traffic Activity Based Intrusion Detection System". In: *International Journal Of Computer Communications & Control* (2015). ISSN: 1841-9844. DOI: 10.15837/ijccc.2015.3.1924.

[123] R. Sommer and V. Paxson. "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection". In: *2010 IEEE Symposium on Security and Privacy*. May 2010. DOI: 10.1109/SP.2010.25.

[124] H. Song, J. Woo, and H. Kim. "In-vehicle network intrusion detection using deep convolutional neural network". In: *Vehicular Communications* (2020). DOI: 10.1016/j.vehcom.2019.100198.

[125] H. M. Song, H. R. Kim, and H. K. Kim. "Intrusion Detection System Based on the Analysis of Time Intervals of CAN Messages for In-Vehicle Network". In: *International Conference on Information Networking (ICOIN)*. 2016. DOI: 10.1109/ICOIN.2016.7427089.

[126] N. Stakhanova and A. A. Cárdenas. "Analysis of Metrics for Classification Accuracy in Intrusion Detection". In: *Empirical Research for Software Security: Foundations and Experience*. 2017. DOI: 10.1201/9781315154855-6.

[127] M. Sugeno and T. Yasukawa. "A Fuzzy-Logic-Based Approach to Qualitative Modeling". In: *IEEE Transactions on Fuzzy Systems* 1 (1993). DOI: 10.1109/TFUZZ.1993.390281.

[128] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, et al. "Intriguing properties of neural networks". In: *International Conference on Learning Representations* (2014).

[129] S. Tariq, S. Lee, H. Kim, and S. Woo. "CAN-ADF: The controller area network attack detection framework". In: *Computers and Security* (2020). DOI: 10.1016/j.cose.2020.101857.

[130] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani. "Toward Credible Evaluation of Anomaly-Based Intrusion-Detection Methods". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* (2010). DOI: 10.1109/TSMCC.2010.2048428.

[131] A. Taylor, S. Leblanc, and N. Japkowicz. "Anomaly detection in automobile control network data with long short-term memory networks". In: *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 2016. DOI: 10.1109/DSAA.2016.20.

[132] C. Thomas and N. Balakrishnan. "Modified evidence theory for performance enhancement of Intrusion Detection Systems". In: *2008 11th International Conference on Information Fusion*. June 2008.

[133] A. Tomlinson, J. Bryans, and S. A. Shaikh. "Towards viable intrusion detection methods for the automotive controller area network". In: *2nd ACM Computer Science in Cars Symposium*. 2018.

[134] C. Tucker, S. Furnell, B. Ghita, and P. Brooke. "A new taxonomy for comparing intrusion detection systems". In: *Internet Research* 1 (2007). DOI: 10.1108/10662240710730515.

[135] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, et al. "Intra-Vehicle Networks: A Review". English. In: *IEEE Transactions on Intelligent Transportation Systems* 2 (Apr. 2015). ISSN: 15249050. DOI: 10.1109/TITS.2014.2320605.

[136] G.-H. Tzeng and J.-J. Huang. *Multiple attribute decision making: Methods and applications*. 2011.

[137] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer. "Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving". In: *IEEE 18th International Conference on Intelligent Transportation Systems*. 2015. DOI: 10.1109/ITSC.2015.164.

[138] J. Ulvila and J. Gaffney. "Evaluation of Intrusion Detection Systems". In: *Journal of Research of the National Institute of Standards and Technology* (Nov. 2003). DOI: 10.6028/jres.108.040.

[139] A. Verma and V. Ranga. "Statistical analysis of CIDDS-001 dataset for Network Intrusion Detection Systems using Distance-based Machine Learning". In: *Procedia Computer Science* (2018). ISSN: 1877-0509. DOI: 10.1016/j.procs.2017.12.091.

[140] A. Warzyński and G. Kołaczek. "Intrusion detection systems vulnerability on adversarial examples". In: *Innovations in Intelligent Systems and Applications*. 2018. DOI: 10.1109/INISTA.2018.8466271.

[141] C. Wohlin. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: 2014. DOI: 10.1145/2601248.2601268.

[142] W. Wu, R. Li, G. Xie, J. An, Y. Bai, et al. "A survey of intrusion detection for in-vehicle networks". In: *IEEE Transactions on Intelligent Transportation Systems* 3 (2020). DOI: 10.1109/TITS.2019.2908074.

[143] L. Xiao, X. Lu, T. Xu, W. Zhuang, and H. Dai. "Reinforcement Learning-Based Physical-Layer Authentication for Controller Area Networks". In: *IEEE Transactions on Information Forensics and Security* (2021). ISSN: 1556-6021. DOI: 10.1109/TIFS.2021.3056206.

[144] Y. Xie, Y. Zhou, J. Xu, J. Zhou, X. Chen, et al. "Cybersecurity protection on in-vehicle networks for distributed automotive cyber-physical systems: State-of-the-art and future challenges". In: *Software - Practice and Experience* (2021). DOI: 10.1002/spe.2965.

[145] Z. Yang, X. Liu, T. Li, D. Wu, J. Wang, et al. "A systematic literature review of methods and datasets for anomaly-based network intrusion detection". In: *Computers & Security* (2022). ISSN: 0167-4048. DOI: 10.1016/j.cose.2022.102675.

[146] C. Young, J. Zambreno, H. Olufowobi, and G. Bloom. "Survey of automotive controller area network intrusion detection systems". In: *IEEE Design and Test* 6 (2019). DOI: 10.1109/MDAT.2019.2899062.

[147] M. Zbakh, K. Elmahdi, R. Cherkaoui, and S. Enniari. "A multi-criteria analysis of intrusion detection architectures in cloud environments". In: *2015 International Conference on Cloud Technologies and Applications (CloudTech)*. June 2015. DOI: 10.1109/CloudTech.2015.7336967.

# A Full Mapping of the Selected Candidate IDSs to the Taxonomy

| | | [30] | [31] | [32] | [42] | [47] | [50](1) | [50](2) | [50](3) | [56] | [61] | [74] | [90] | [89] | [94] | [104] | [116] | [124] | [125] | [129] | [131] | [143] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S1 | 1 | X | ✓ | ✓ | X | X | X | X | ✓ | X | ✓ | X | X | ✓ | X | X | X | X | X | X | X | ✓ |
| | 2 | ✓ | X | X | ✓ | X | ✓ | ✓ | X | ✓ | ? | X | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? |
| | 3 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 4 | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 5 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| S2 | 1 | ✓ | ✓ | ✓ | X | ? | X | ✓ | ✓ | ? | ? | ? | ? | ✓ | X | ? | X | X | X | X | ? | ✓ |
| | 1.1 | ✓ | ✓ | ✓ | X | ? | X | ✓ | ✓ | ? | ? | ? | ? | ✓ | X | ? | X | X | X | X | X | ? |
| | 2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2.1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ? | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2.2 | ✓ | ✓ | ✓ | ? | ✓ | X | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | X | ? | X | X | X | ? | ? | X |
| S3 | 1 | X | ? | X | X | X | X | X | X | X | ? | ? | X | X | X | ? | X | X | X | X | X | X |
| | 2 | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| S4 | 1 | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ? | ✓ |
| | 2 | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ? | ? | ✓ |
| | 3 | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ? | ? | ✓ |
| | 4 | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ? | ✓ |
| | 5 | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ? | ✓ | ✓ |
| | 6 | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| S5 | 1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2 | ✓ | ✓ | ✓ | X | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ? | X | ✓ | X | ✓ | ✓ | ? |
| | 3 | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 4 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| | | [30] | [31] | [32] | [42] | [47] | [50](1) | [50](2) | [50](3) | [56] | [61] | [74] | [90] | [89] | [94] | [104] | [116] | [124] | [125] | [129] | [131] | [143] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | 1 | ✓ | X | X | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ? | ✓ | ? | ? | ✓ | ? | ? | ? | ✓ |
| | 2 | ✓ | ✓ | ✓ | X | ? | ✓ | ✓ | X | ✓ | ✓ | ? | ✓ | ? | X | ? | ? | X | ? | X | ? | ✓ |
| | 2.1 | ✓ | ✓ | X | X | ? | X | X | X | ? | ✓ | ? | ? | ? | X | ? | ? | X | ? | X | ? | ✓ |
| | 2.2 | ? | ✓ | ✓ | X | ? | ✓ | ✓ | X | ? | ✓ | ? | ? | ? | X | ? | ? | X | ? | X | ? | ✓ |
| | 3 | ? | ? | X | X | ? | X | X | X | X | ✓ | ? | ? | ? | X | X | ? | X | ? | ? | X | ? |
| | 4 | X | X | X | X | X | X | X | X | X | X | ? | X | X | X | X | ? | X | ? | X | X | X |
| A2 | 1 | ✓ | ✓ | X | X | ? | ? | ? | ? | ? | X | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | X |
| | 2 | ✓ | ✓ | X | X | ? | ? | ? | ? | ? | X | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | X |
| | 3 | ✓ | ✓ | ✓ | ✓ | ? | ? | ? | ? | ? | ✓ | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ✓ |
| A3 | / | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a |
| A4 | 1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 1.1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 1.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 2 | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ? | ✓ | ✓ | X | ✓ | ✓ | ✓ | ? | ✓ |
| | 3 | ✓ | ? | ✓ | ? | ✓ | ✓ | ✓ | ✓ | X | ? | ✓ | ✓ | ? | ✓ | ✓ | ✓ | X | ✓ | ✓ | ? | ✓ |
| | 3.1 | ✓ | ? | ✓ | X | ✓ | ✓ | ✓ | ✓ | X | ? | X | X | ? | X | X | X | X | X | X | ? | X |
| | 3.2 | X | ? | ? | ? | ✓ | X | X | X | X | ? | ✓ | ✓ | ? | ✓ | ✓ | ✓ | X | ✓ | ✓ | ? | ✓ |
| A5 | 1 | ✓ | ✓ | X | X | X | ✓ | ✓ | ✓ | X | X | X | X | X | ✓ | X | X | ✓ | X | X | X | X |
| | 1.1 | ✓ | ✓ | X | X | X | ✓ | ✓ | ✓ | X | X | X | X | X | ✓ | X | X | ✓ | X | X | X | X |
| | 1.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2.1 | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2.1.1 | X | X | X | X | X | X | X | X | X | X | X | X | ? | X | X | X | X | ✓ | X | X | X |
| | 2.1.2 | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2.1.2.1 | X | X | ✓ | ? | X | X | X | ✓ | ? | ✓ | X | X | ✓ | X | ✓ | X | X | X | ? | ✓ | X |
| | 2.1.2.2 | X | ✓ | ✓ | ✓ | X | X | X | ✓ | ? | ✓ | X | ? | ? | X | ? | X | X | X | ? | ? | X |
| | 2.1.2.3 | X | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | X | X | X | ? | ✓ | X | X | X | ✓ | ✓ | ✓ | ✓ |
| | 2.1.2.4 | X | ✓ | ✓ | ✓ | ✓ | X | ✓ | ✓ | ? | X | X | ? | ? | X | ✓ | X | ✓ | ? | ✓ | ? | ? |
| | 2.1.3 | X | X | ✓ | ? | ✓ | ✓ | ✓ | ✓ | X | X | ✓ | ✓ | ? | X | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| | 2.2 | ✓ | X | ✓ | X | X | X | X | X | X | X | X | ? | ? | X | X | X | X | X | X | ✓ | ? |
| | 2.2.1 | ✓ | X | ✓ | X | X | X | X | X | X | X | X | ? | ? | X | X | X | X | X | X | X | ? |
| | 2.3 | X | ✓ | ✓ | X | ✓ | X | X | X | ✓ | X | X | ? | ? | X | X | X | X | X | ✓ | ✓ | X |

| | | [30] | [31] | [32] | [42] | [47] | [50](1) | [50](2) | [50](3) | [56] | [61] | [74] | [90] | [89] | [94] | [104] | [116] | [124] | [125] | [129] | [131] | [143] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A6 | 1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 4 | ✓ | X | X | ✓ | ✓ | ✓ | X | X | ? | X | ✓ | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 5 | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | X | X | X | X | X | X | ✓ | ✓ |
| | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| A7 | 1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2 | ✓ | ✓ | ✓ | ? | ? | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ? | ✓ |
| | 3 | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ? | ✓ |
| | 4 | ✓ | ✓ | ✓ | ? | X | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ? | ✓ |
| | 5 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 6 | ✓ | ✓ | ✓ | ? | X | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ? | ✓ |
| D1 | 1 | X | X | X | X | X | X | X | X | X | ? | ✓ | X | X | X | X | X | X | X | X | X | X |
| | 2 | X | ✓ | ✓ | ✓ | ? | X | X | X | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 3 | X | X | X | X | ? | ✓ | X | X | ? | ✓ | X | X | X | X | X | ? | X | X | X | X | ? |
| | 4 | X | ✓ | X | X | ? | ✓ | ? | ✓ | ? | ✓ | X | X | X | ? | X | ? | X | X | X | X | ? |
| | 5 | X | X | X | X | X | X | ? | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 6 | ✓ | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| D2 | 1 | X | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X |
| | 2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| D3 | 1 | ✓ | ? | X | X | ? | X | ✓ | X | ✓ | X | ✓ | X | X | X | ✓ | ✓ | ✓ | X | X | ? | X |
| | 2 | X | X | X | X | X | X | X | X | X | ? | X | X | X | X | X | X | X | X | X | X | X |
| | 3 | X | ? | ✓ | ✓ | ? | ✓ | X | ✓ | X | ? | X | ✓ | ✓ | ✓ | X | X | X | ✓ | ✓ | X | ✓ |
| | 4 | X | ✓ | ✓ | ✓ | X | ? | X | ? | X | ✓ | X | ✓ | ✓ | X | X | X | X | ? | X | ✓ | ✓ |
| D4 | 1 | X | X | X | X | X | X | ✓ | ✓ | ✓ | ✓ | X | ? | X | X | X | X | X | X | X | X | X |
| | 2 | ✓ | ? | ✓ | ✓ | ✓ | ✓ | X | X | X | X | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2.1 | ✓ | ? | ✓ | ✓ | X | ✓ | X | X | X | X | X | ? | ✓ | ✓ | ? | X | ✓ | X | ✓ | ✓ | ✓ |
| | 2.2 | X | X | X | X | ✓ | X | X | X | X | X | ✓ | ? | X | X | ? | ✓ | X | ✓ | X | X | X |

| | | [30] | [31] | [32] | [42] | [47] | [50](1) | [50](2) | [50](3) | [56] | [61] | [74] | [90] | [89] | [94] | [104] | [116] | [124] | [125] | [129] | [131] | [143] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D5 | 1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 2 | X | ✓ | ✓ | X | X | X | X | ✓ | X | ✓ | ✓ | X | ✓ | X | X | X | X | X | X | X | ✓ |
| | 2.1 | X | ✓ | ✓ | X | X | X | X | ✓ | X | ✓ | X | X | ✓ | X | X | X | X | X | X | X | ✓ |
| | 2.2 | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 3 | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | X | ✓ | X | X | ✓ | X | ✓ | ✓ | X | ✓ | X | ✓ | ✓ | ✓ |
| | 3.1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 3.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 3.3 | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 3.4 | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | X | ✓ | X | X | ✓ | X | X | ✓ | X | ✓ | X | ✓ | X | ✓ |
| | 3.5 | X | X | X | ✓ | ✓ | X | X | X | ✓ | X | X | ✓ | X | ✓ | ✓ | X | X | X | ✓ | ✓ | X |
| | 3.5.1 | X | X | X | ✓ | X | X | X | X | ✓ | X | X | X | X | ✓ | ✓ | X | X | X | ✓ | ✓ | X |
| | 3.5.2 | X | X | X | X | ✓ | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X |
| | 4 | X | ✓ | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X |
| | 5 | X | X | X | ✓ | ✓ | ✓ | X | X | X | X | ✓ | X | X | ✓ | X | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| | 5.1 | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X |
| | 5.2 | X | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X |
| | 5.3 | X | X | X | ✓ | X | ✓ | X | X | X | X | ✓ | X | X | ✓ | X | ✓ | X | ✓ | ✓ | X | ✓ |
| | 5.4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 6 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 7 | X | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X |
| D6 | 1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

| | | [30] | [31] | [32] | [42] | [47] | [50](1) | [50](2) | [50](3) | [56] | [61] | [74] | [90] | [89] | [94] | [104] | [116] | [124] | [125] | [129] | [131] | [143] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D7 | 1 | X | ✓ | X | X | ✓ | X | X | X | X | ? | X | ✓ | X | ✓ | X | X | X | X | X | X | X |
| | 1.1 | X | X | X | X | X | X | X | X | X | ? | X | X | X | ✓ | X | X | X | X | X | X | X |
| | 1.1.1 | X | X | X | X | X | X | X | X | X | ? | X | X | X | ✓ | X | X | X | X | X | X | X |
| | 1.1.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 1.1.3 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 1.2 | X | X | X | X | ✓ | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X |
| | 1.2.1 | X | X | X | X | ✓ | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X |
| | 1.2.2 | X | X | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X |
| | 1.2.3 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 1.3 | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 1.3.1 | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 1.3.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| D8 | 1 | ✓ | X | X | X | X | ✓ | ✓ | ✓ | ✓ | X | ? | X | X | ✓ | X | ✓ | ✓ | ✓ | X | ✓ | X |
| | 2 | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X | ✓ | X | X |
| | 3 | X | ✓ | ✓ | X | X | X | X | X | X | X | X | X | ✓ | X | ✓ | X | X | X | ✓ | X | ✓ |
| | 4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | ✓ | X | X |
| | 5 | X | X | X | ✓ | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X |
| D9 | 1 | X | X | ✓ | X | X | X | X | X | ✓ | ✓ | ✓ | X | X | ✓ | ✓ | X | ✓ | X | ✓ | X | ? |
| | 2 | X | ✓ | X | ✓ | ✓ | ✓ | X | ✓ | X | X | X | ✓ | ✓ | X | X | ✓ | X | ✓ | X | ✓ | ? |
| D10 | 1 | ✓ | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | ✓ | X | X |
| | 2 | X | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2.1 | X | ✓ | ✓ | ✓ | X | ✓ | X | ✓ | ✓ | X | X | ✓ | ✓ | ✓ | X | ✓ | X | ✓ | X | X | X |
| | 2.2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 2.3 | X | ✓ | X | X | ✓ | X | X | X | ✓ | ✓ | ✓ | X | X | X | ✓ | X | ✓ | X | ✓ | ✓ | ✓ |
| D11 | 1 | X | ✓ | ✓ | X | X | X | X | ✓ | ? | ✓ | ✓ | ✓ | ✓ | X | ? | X | X | X | X | ✓ | ✓ |
| | 2 | X | X | X | ✓ | ✓ | ✓ | X | X | ? | ✓ | ? | X | X | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 3 | X | X | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X |
| | 4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

| | | [30] | [31] | [32] | [42] | [47] | [50](1) | [50](2) | [50](3) | [56] | [61] | [74] | [90] | [89] | [94] | [104] | [116] | [124] | [125] | [129] | [131] | [143] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D12 | 1 | ✓ | X | X | X | ✓ | X | X | X | ✓ | X | ✓ | X | X | X | ✓ | X | ✓ | X | ✓ | ✓ | ? |
| | 2 | X | ✓ | ✓ | ✓ | X | ✓ | ✓ | ✓ | X | ✓ | X | ✓ | ✓ | ✓ | X | ✓ | X | ✓ | X | X | ? |
| | 3 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 5 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| D13 | 1 | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2 | ✓ | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | ✓ | X | X | X |
| D14 | 1 | X | ? | X | ✓ | ✓ | X | X | X | ? | ✓ | ✓ | ? | X | X | ✓ | X | ✓ | X | ✓ | X | ? |
| | 2 | ✓ | ? | ✓ | X | X | ✓ | ✓ | ✓ | ? | X | X | ? | ✓ | ? | X | ✓ | X | ✓ | ? | ✓ | ? |
| D15 | 1 | ✓ | ✓ | X | X | ? | X | ? | X | ? | X | ✓ | X | X | X | ? | X | X | X | ? | ? | X |
| | 2 | X | X | ✓ | ✓ | X | ✓ | ? | ✓ | ? | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | ? | X | ? |
| | 3 | X | X | X | X | ? | X | X | X | ? | X | X | X | X | X | ? | X | ✓ | X | X | ? | ? |
| D16 | 1 | X | ? | ✓ | X | X | X | X | X | X | X | X | ✓ | ✓ | X | X | X | X | X | X | X | X |
| | 2 | X | ? | X | X | X | X | X | X | X | ✓ | X | ✓ | X | X | X | X | X | X | X | X | X |
| | 3 | X | ? | X | X | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | X | X | X | X |
| | 4 | X | ? | X | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | X | X | ✓ | X | ✓ | X | ✓ | ✓ | ? | ✓ |
| | 5 | X | ? | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 6 | ✓ | ? | X | ✓ | X | X | X | X | X | X | ✓ | X | X | X | ✓ | X | ✓ | X | X | ? | X |
| D17 | 1 | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ? | ? | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 1.1 | X | ? | ✓ | X | X | X | ✓ | ✓ | ? | ✓ | X | ? | ? | X | X | X | X | X | X | X | ✓ |
| | 1.2 | ✓ | ? | X | ✓ | ✓ | ✓ | X | X | ? | ✓ | X | ? | ? | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | X |
| | 2 | X | ? | X | X | X | X | X | X | X | X | ✓ | ? | ? | X | ? | X | X | X | X | X | X |
| D18 | 1 | ✓ | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ? | ? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 2 | X | ? | ? | X | ✓ | X | X | X | ? | ? | ? | ? | ? | X | X | X | ? | ? | ? | ? | X |
| | 3 | X | ? | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| D19 | 1 | ✓ | X | ✓ | X | X | X | X | X | ? | X | ? | ? | X | X | ✓ | ✓ | ? | ✓ | ✓ | X | X |
| | 2 | X | X | ✓ | X | X | X | X | X | ? | X | X | ? | ? | X | X | ✓ | X | ✓ | ? | ? | ✓ |
| | 3 | X | ✓ | X | X | X | X | X | ✓ | ? | ✓ | X | X | ? | X | X | ✓ | X | X | X | X | ? |
| | 4 | X | X | X | X | X | X | X | X | ? | X | X | X | X | X | X | X | X | X | X | X | X |

| | | [30] | [31] | [32] | [42] | [47] | [50](1) | [50](2) | [50](3) | [56] | [61] | [74] | [90] | [89] | [94] | [104] | [116] | [124] | [125] | [129] | [131] | [143] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D20 | 1 | X | X | X | X | X | ✓ | ✓ | ✓ | X | X | X | X | X | X | X | X | X | X | ✓ | X | X |
| | 2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 3 | ✓ | ✓ | X | X | X | X | X | X | X | ✓ | X | X | X | X | X | ✓ | X | X | X | X | ✓ |
| D21 | 1 | X | ✓ | ✓ | X | X | X | ✓ | ✓ | X | ✓ | X | X | ✓ | X | X | X | X | X | X | X | ✓ |
| | 2 | X | X | X | ✓ | ✓ | X | X | X | ✓ | X | X | X | X | X | ✓ | X | X | X | ✓ | ✓ | X |
| | 3 | X | ? | ✓ | ✓ | ✓ | X | X | ✓ | ? | ? | X | X | ✓ | ✓ | X | ? | ✓ | X | X | X | ✓ |
| | 4 | X | ✓ | ✓ | X | X | X | X | ✓ | ? | ✓ | X | X | ✓ | X | X | ? | X | X | X | X | ✓ |
| D22 | 1 | ✓ | X | ✓ | ✓ | X | ✓ | X | X | X | ✓ | X | ✓ | X | ✓ | X | X | X | X | ✓ | ✓ | X |
| | 2 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 3 | X | ✓ | ✓ | X | ✓ | ✓ | X | X | X | ✓ | ✓ | X | X | ✓ | X | X | ✓ | X | ✓ | X | X |
| | 4 | X | X | ✓ | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |