Doctoral theses at NTNU, 2023:364

Jin Zhang

Evaluating Artificial Neural Network Robustness for Safety-Critical Systems

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science
Technical University of Denmark
Department of Civil and Mechanical Engineering
Section of Engineering Design and
Manufacturing Systems

NTNU
Norwegian University of
Science and Technology

DTU

Jin Zhang

Evaluating Artificial Neural Network Robustness for Safety-Critical Systems

Thesis for the Degree of Philosophiae Doctor

Trondheim, December 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Technical University of Denmark
Department of Civil and Mechanical Engineering
Section of Engineering Design and Manufacturing Systems

NTNU
Norwegian University of
Science and Technology

DTU

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Technical University of Denmark
Department of Civil and Mechanical Engineering
Section of Engineering Design and Manufacturing Systems

ISBN 978-82-326-7428-2 (printed ver.)
ISBN 978-82-326-7427-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2023:364

Printed by NTNU Grafisk senter

Abstract

With the power to perform more complex tasks than humans, artificial neural networks (ANNs) have been applied to execute tasks in safety-critical systems (SCSs), such as object detection, image recognition, and navigation. An ANN should provide consistent performance when input deviates from the training data. This corresponds to the attribute of robustness in the ANN.

The obstacles to developing robust ANN-based safety-critical systems (ANN-SCSs) encompass four interrelated aspects: 1) the inherent complexity and nonlinearity of ANNs that call for innovative testing and verification (T&V) techniques; 2) the need to establish a well-defined connection between robustness and safety by considering various factors; 3) the vital nature of addressing the immaturity of robustness evaluation and measurement to ensure the seamless integration of ANNs in safety-critical applications in operation; and 4) the development of precise and practical robustness measurement in operation without labeled data. It is vital to have methods to accommodate the ever-changing nature of real-world data and the diversity of ANN architectures and use cases. Consequently, addressing these four challenges holistically is essential to facilitate a safe and reliable transition toward incorporating ANNs in SCSs.

This thesis provides knowledge on ANN robustness evaluation in the context of SCSs. It develops new knowledge, methods, and guidance, combining traditional risk analysis concepts with convolutional neural network theory and robustness studies. Four main research papers have been published and submitted as a result of the work in this thesis. These papers together provide scientific contributions to 1) the systematization of knowledge and understanding for T&V of ANN-SCSs; 2) a new method for analyzing the influence of ANN robustness on the safety of autonomous vehicles; 3) a systematic summary of methods and metrics to measure ANN-SCS robustness in operation; and 4) empirical results that demonstrate the applicability of distance metrics in selecting more robust ANN models from several alternatives using unlabeled data in operation. The systematization of knowledge, the method to evaluate ANN robustness, and insights on the advantages and disadvantages of the corresponding metrics pave the way for a future where the robustness and safety of ANN-SCSs can be quantified and enhanced, ensuring improved operational safety and effectiveness in real-world scenarios.

Preface

This doctoral thesis, a result of a joint research project financed by the Norwegian University of Science and Technology (NTNU) and the Technical University of Denmark (DTU), is submitted to both NTNU and DTU for partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The primary objective of this project, titled *Management of Safety and Security Risks for Cyber-Physical Systems*, was to explore various methodologies for conducting a holistic and cost-effective risk analysis of safety-critical systems. The specific scope of this thesis focused on the development of innovative methods suitable for integrated and quantitative analysis of the safety and security aspects of these systems. The ultimate goal was to offer valuable recommendations to the industry.

The majority of the Ph.D. work was conducted at the Department of Computer Science at NTNU, Trondheim, Norway, under the supervision of Professor Jingyue Li, the primary supervisor. Further direction was provided by Associate Professor Josef Oehmen (DTU), Senior Research Scientist Igor Kozin (University of Southern Denmark, SDU), and in the early phases, Professor Mary Ann Lundteigen (NTNU).

A pivotal aspect of this collaborative project was the partnership with my tandem partner, Nelson Guzman, at DTU. This collaboration involved regular meetings and workshops to facilitate communication, discussions, and the consolidation of research progress, issues, and findings. Over the course of the year-long collaboration, Nelson Guzman and I co-authored a paper and developed a research proposal.

As part of my Ph.D. journey, I also had the opportunity to contribute to the IDUN project initiated by Professor Letizia Jaccheri (NTNU), under the mentorship of Professor Nirmalie Wiratunga (Robert Gordon University). My academic involvement at NTNU extended to teaching assistant roles in several courses, including Software Security and Customer-Driven Project, and a course coordinator role for Empirical Software Engineering. Further, I took on roles as an examiner of master's theses and a paper reviewer.

Acknowledgements

I am glad I was unaware of the challenges of starting a Ph.D. The journey has transformed me in ways I could not have imagined. Importantly, this Ph.D. would not have been possible without the support of a few key people.

First and foremost, I am extremely grateful to my supervisors. Jingyue, you have supported and pushed me throughout these years with great patience. Thank you for your invaluable guidance and continuous support during my Ph.D. study. Igor, it has been an honor to be your Ph.D. student and work with you for the past few years. I could not forget your guidance with great warmth and wisdom. Mary Ann, you inspire me with your positivity, structure, and never-ending compassion. Second, I am especially thankful to Josef Oehmen and Zhirong Yang for jumping aboard the ship to help steer this Ph.D. toward land. Your experience and knowledge have been of great help to make this Ph.D. come together. I also thank Prof. Xiaomeng Su for your enlightening talks. Thank you for both challenging and supporting me in times of need.

I am honored to have been part of the NTNU-DTU joint project and would like to thank my tandem partner Nelson for your collaboration, shared experiences, and inspirational work. It has been great fun! I also give a huge thanks to Robert Taylor for your mentoring and companionship throughout this Ph.D., especially during the year in the home office. I truly appreciate all our talks and discussions and look forward to many more in future collaborations.

I want to thank my dearest office mates and friends—Nektaria, Elnaz, Hong, Jingji, Yue, and Hongxia. Your kind help and support have made my study and life in Norway a wonderful time.

I want to thank my parents, Haiming and Qirong; my husband, Lei; and my daughter, Ruojia, for your tremendous understanding and encouragement over the past few years. Thank you for your unconditional love, encouragement, and simply always believing in me. For that, I am forever grateful.

Lastly, I reserve my most intimate acknowledgment for myself. I am proud of myself for never giving up and for making it through all the dark moments to the finish line.

Jin Zhang
14 July 2023

Contents

List of Figures

List of Tables

Abbreviations

ADS Autonomous Driving System

AI Artificial Intelligence

AI RMF AI Risk Framework

ALR Ad hoc Literature Review

ANN Artificial Neural Network

ANN-SCS ANN-classifier-based Safety Critical System

AV Autonomous Vehicle

CPS Cyber Physical System

DNN Deep Neural Network

DRM Design Research Methodology

FAA Federal Aviation Administration

FDA Food and Drug Administration

FFTA Fuzzy Fault Tree Analysis

FGSM Fast Gradient Sign Method

FMEA Failure Modes and Effects Analysis

FTA Fault Tree Analysis

GTSRB German Traffic Sign Recognition Benchmark

HL Hellinger Distance

IMDRF International Medical Device Regulators Forum

KL Kullback-Leibler Divergence

KS Kolmogorov–Smirnov statistic

MMD maximum mean discrepancy

ML Machine Learning

MLops Machine Learning Operations

MS Medical System

NIST National Institute of Standards and Technology

OOD out-of-distribution

PFH Average Frequency of Dangerous Failures per Hour

RAM Reliability Assessment Model

SAE Society of Automotive Engineers

SCS Safety Critical System

SIL Safety Integrity Level

SOTIF Safety of the Intended Functionality

SLA Service-level Agreement

SLR systematic literature review

STPA Systems Theoretic Process Analysis

TEVV Test, Evaluation, Verification, and Validation

T&V Testing and Verification

TPLC Total Product Lifecycle

UAS Unmanned Aircraft System

UMAP Uniform Manifold Approximation and Projection for Dimension Reduction

V&V Verification and Validation

WD Wasserstein Distance

XAI Explainable AI

Part I

Overview

1 INTRODUCTION

This chapter introduces the thesis subject area and problem statement and describes the research motivation, research questions, main contributions, and thesis outline. It provides an overview of the challenges in testing and verifying ANN-SCSs, the need to establish a connection between robustness and safety, and the gaps in current research.

1.1 Problem Statement

Safety-critical systems (SCSs) must be robust against failures that cause harm to people and lead to further economic loss and environmental and/or reputational damages [1]. With the power to perform more complex tasks than humans, artificial neural networks (ANNs) have been successfully adopted in several SCSs, such as autonomous vehicles (AVs), drones, and health care devices [2]. Within these applications, ANNs have shown remarkable performance in object detection, image recognition, navigation, control, etc. However, an SCS that uses an ANN may perform poorly or catastrophically misbehave due to incorrectly comprehending the sensor input variations or under diverse environmental conditions. For instance, the self-driving Uber that killed a pedestrian did not realize she was a human [3]. In addition, Tesla's Autopilot lane-keeping assist system failed to recognize a crash attenuator [4] and a stationary obstacle hidden by the leading car [5]. Assuring the prediction accuracy of the ANN classifier in a tolerable range with atypical data is a crucial need in SCSs [6].

Robustness, i.e., the ability to maintain performance in the face of perturbations and uncertainty, is a long-recognized key property of SCSs, according to relevant safety standards, such as IEC 61508[1] and ISO 26262.[2] Testing and verification (T&V) of ANNs for SCSs can help form the foundation to trust the decisions made by ANN algorithms at both the design and the operational stages. The existing studies to assure the robustness of ANNs fall into two main categories: 1) robustness measurement and 2) robustness enhancement. In a traditional safety analysis, robustness is calculated by obtaining components' failure probabilities

[1] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems

[2] ISO 26262: Road vehicles—Functional safety

(generally by observing the components over a long period or looking them up in failure rate databases collected from observation [7, 8]). In this thesis, we treat ANN as a component in an SCS. An ANN's robustness is defined as its ability to maintain a similar prediction accuracy under different conditions [9]. The robustness of ANN-classifier-based safety-critical systems (ANN-SCSs) can be put at risk due to adversarial attacks, natural perturbations, and random failures [10, 11]. Some studies have tried to understand the characteristics and impacts of adversarial inputs [12, 13], identify the intrinsic features of robust classifiers [14, 15, 16, 17], and develop methods to detect and mitigate the effects of adversarial examples [18, 19, 20, 21, 22]. To improve the robustness of ANNs, researchers commonly employ data augmentation techniques [23] and increase the complexity of the models [24]. For example, Zheng et al. [25] and Shaham et al. [26] proposed methods that leverage adversarial training to enhance the robustness of ANNs. Adversarial training involves generating a large number of adversarial examples and training the ANN to be resilient against these examples. Another approach called AugMix [27] randomly selects different augmentations, applies them to a training image, and then combines the augmented image with the original to improve robustness against natural perturbations. Ensemble learning techniques, which involve training multiple models and combining their outputs, show promise in achieving robustness in natural perturbation scenarios [28, 29, 30]. To overcome ANN hardware failure, a typical approach is to use redundancy, i.e., using two or more devices to achieve the same safety function [31].

Although academic and industrial efforts to address T&V of ANN-SCSs are increasing, there remains a gap between industry needs and state-of-the-art T&V methods. Research institutions and industry T&V practitioners are working on different aspects of this problem. Nevertheless, to the best of our knowledge, at the early stage of this Ph.D. study, there was limited research available explicitly focusing on the risk analysis of ANNs. Additionally, it appeared that existing safety standards did not explicitly address the unique challenges associated with testing and verifying ANN-SCSs. Recently, three relevant standards focused on testing AI-based systems,[3] assessing the trustworthiness of AI systems,[4] and assuring the robustness of ANNs[5] were (partially) released. However, there is currently a weak connection between potentially useful methods for robustness evaluations of ANNs and relevant safety standards. This weak connection reflects a lack of comprehensive understanding regarding the underlying mechanisms and influencing factors of ANN robustness. Consequently, the field of robustness evaluation and measurement remains immature, which poses a significant challenge in seamlessly integrating ANNs into safety-critical applications. Operationalizing ANN systems in harsh operational environments, including industrial settings, presents unique robustness challenges that require specific solutions [32]. To illustrate the importance of robustness, let us consider the incident involving a Tesla Model S P85 car crashing into a stationary fire truck in 2018 [33]. This collision occurred while the car was in Autopilot mode

[3] ISO/IEC TR 29119-11: Guidelines on the testing of AI-based systems

[4] ISO24028: Overview of trustworthiness in artificial intelligence

[5] ISO/IEC TR 24029-1: Assessment of the robustness of neural networks

with traffic-aware cruise control, and it resulted from the system's inability to detect a stationary obstacle hidden by a preceding vehicle. This incident serves as a clear reminder of the urgent need to improve the robustness of ANN-based advanced driver assistance systems to effectively handle complex real-world scenarios and minimize reliance on inattentive drivers.

1.2 Research Motivation

There is a recognized and well-documented interest in the T&V of ANN-SCSs across the fields of machine learning (ML), software engineering, and safety communities [34, 35, 36]. This is demonstrated by the growing demand for research guidance in testing and verifying ANN-SCSs. Notably, 11 automotive industry leaders have established a framework for developing, testing, verifying, and validating the safety of automated passenger vehicles (SaFAD) [37] aimed at creating standards for automated driving. Moreover, Waymo, a leader in AV technology, releases an annual analysis of events from AV operation on public roads [38], highlighting the importance of T&V.

However, there is a lack of comprehensive research investigating the influence of ANN robustness on system safety. This highlights an essential problem: without effective measurement of the robustness of ANN-SCSs, there is no structured path to improving robustness, as the gap between desired and realized performance remains unknown. Existing metrics, evaluation methods, and robustness challenges of ANN models have been discussed in several surveys [39, 40, 41, 42]. However, these surveys cover a wide range of focus areas, from adversarial robustness [40, 41] and corruption robustness [43] to distributional robustness [10]. The diversity complicates the creation of a unifying taxonomy, evaluation metrics, measurement techniques, and evaluation framework applicable in real-world scenarios. Furthermore, most evaluation metrics and methods are designed for the model development stage [44, 42, 43, 45], making them unsuitable for ANN-SCSs in operation. While many studies on T&V approaches for ANN-SCSs focus on algorithm verification, ensuring that the algorithms are correctly programmed [46, 47, 48], they often do not align with a more holistic view of the safety of the whole SCSs.

Moreover, ANN models deployed in operation are susceptible to input data changes from the training data [49], known as out-of-distribution (OOD) shifts. One possible solution to this challenge is to use a multi-model decision-making approach [50]. This approach involves using different models to perform the same task, leveraging the diversity of models to provide robust and reliable predictions in dynamic and changing conditions [30]. As the recent literature emphasizes [50, 51], a crucial research direction for effective AI risk management is continuous monitoring and validation of AI systems. This implies that practitioners need to evaluate and choose optimal models under shifting conditions dynamically, which highlights the critical need to develop effective methodologies to address these challenges.

This thesis, therefore, aspires to bridge a notable gap in contemporary research by intertwining elements of SCS and ML. It places special emphasis on understanding how ANN impacts the performance of SCSs, addressing ANN robustness in real-world operational environments, and effectively managing AI risks in dynamic scenarios. ANNs encompass a broad category of deep learning algorithms that utilize deep neural networks (DNNs) with multiple layers of nonlinear processing units for feature extraction and transformation [52]. This study specifically examines SCSs that employ ANNs for classification tasks, allowing us to address a wide range of real-world use cases and provide valuable insights to a broad audience.

1.3 Research Questions

This thesis addresses the research motivations by answering the research questions step by step. The research questions (RQs) investigated by this thesis are:

> **RQ1:** What are the challenges associated with testing and verifying the robustness of ANN classifiers for SCSs?

> **RQ2:** How can we analyze the influence of the ANN classifier's robustness on SCSs' safety?

> **RQ3:** What are the perceptions and practices of robustness evaluation in ANN-SCSs in operation?

> **RQ4:** How can we compare and rank the robustness of multiple ANN classifiers using unlabeled input during operation, supposing OOD shifts may happen at any time during operation?

Figure 1.1 shows the mapping of the RQs with the interactions of the main research areas. The above RQs follow a sequential order in which the study of the latter relies upon the results of the former. First, RQ1 aims to ground the research by systematically reviewing the state of the art of T&V approaches for ANN-SCSs to understand the challenges in assuring the robustness of ANNs. RQ2 aims to propose a new methodology to address one of the challenges identified from RQ1, i.e., analyzing the influence of ANN robustness on system safety. Lastly, RQ3 and RQ4 narrow the scope by focusing on the robustness evaluation of ANN classifier in operation. RQ3 aims to systematize the knowledge of operational robustness evaluation (challenges and solutions) and propose a framework for the robustness evaluation of ANN-SCSs. Building directly on the foundation laid by RQ3, RQ4 takes a step further, targeting the empirical assessment of metrics that can be used to dynamically rank the robustness of multiple ANN classification models under OOD shifts utilizing unlabeled data.

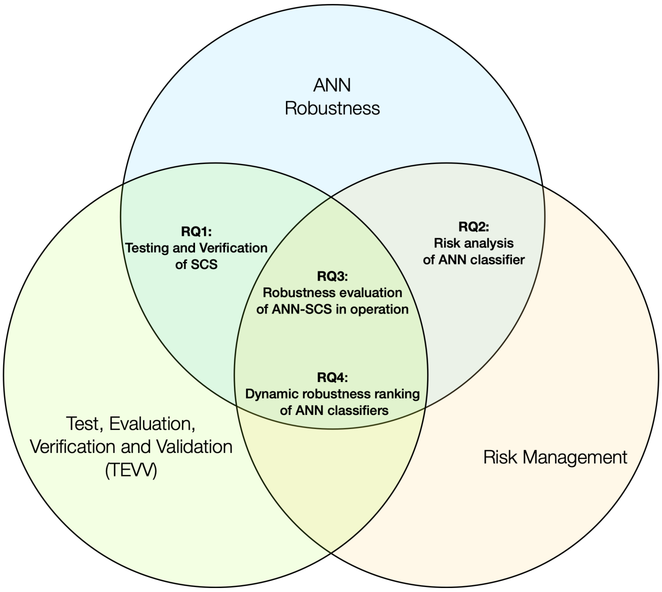


Figure 1.1: Coherence between our research questions.

1.4 Research Outcomes

This thesis builds upon studies across ML and safety analysis with inspiration from design science and safety engineering in a holistic view. The RQs are addressed in four published/submitted papers in peer-reviewed journals and conference proceedings.

1.4.1 Research Papers

The research papers that address the RQs are listed below. The connections between the research papers and the RQs are illustrated in Table 1.1.

P1 J. Zhang and J. Li, **'Testing and verification of neural network-based safety-critical control software: A systematic literature review,'** *Journal of Information and Software Technology*, vol. 123, 2020, Art. no. 106296.

My contribution: I was the leading author and developed the research design. Li supervised this process through regular consensus meetings with me. I performed the keyword search process and selected papers based on the inclusion and exclusion criteria. I extracted the data, thematically categorized the findings, and prepared the research results. Li contributed to this process. Li and I discussed the results. I wrote the paper, and Li commented on the paper.

Table 1.1: Mapping of main research papers and research questions.

	P1	P2	P3	P4
RQ 1	•			
RQ 2		•		
RQ 3			•	
RQ 4				•

Relevance to the thesis: This paper systematically reviews the recent literature on **T&V methods for ANN-SCSs**. The paper contributes to addressing RQ1. Our findings in this paper helped formulate RQ2–RQ4 and conduct P2–P4. The paper has two main contributions: 1) classification of T&V approaches in academia and industry for ANN-SCSs and 2) identification of challenges for advancing state-of-the-art T&V for ANN-SCSs. To conclude, the T&V approaches were categorized into five higher-order themes: assuring the robustness of ANNs, improving the failure resilience of ANNs, measuring and ensuring test completeness, assuring the safety properties of ANN-SCSs, and improving the interpretability of ANNs. From the industry perspective, assuring the robustness of ANNs is a crucial need in safety-critical applications.

P2 J. Zhang, J. R. Taylor, I. Kozin, and J. Li, **'Analyzing influence of robustness of neural networks on the safety of autonomous vehicles,'** in *31st European Safety and Reliability Conference*, 2021, pp. 2276–2283.

My contribution: I was the leading author and developed the research design and conceptualization. Kozin and Li supervised this process through regular meetings with me. I proposed a novel methodology, and Taylor contributed to this process. I developed the algorithms, designed and performed the experiments, and analyzed the findings. All authors discussed the results, and I wrote the paper based on the findings. Kozin, Taylor, and Li commented on the paper. Finally, I presented the paper at the online conference.

Relevance to the thesis: This paper presents an extended fault tree analysis (FTA) to represent combinations of failure causes in multidimensional space, i.e., two variables influencing whether an image is classified correctly. First, the paper shows that ANNs and vision algorithms can be included in overall risk analysis as a fault tree (FT) by using the concept of exceeding the robustness of ANNs as FT events alongside the traditional component failure probabilities. Following this, the extended FTA is demonstrated in the traffic sign recognition module of AVs theoretically and in practice. It demonstrates how an FT can include failure events from multiple small parameter deviations influencing image recognition, resulting in unsafe performance. The paper contributes to RQ2 by providing a method to analyze the influence of the ANN classifier's robustness on the safety of AVs.

P3 J. Zhang, J. Li, and J. Oehmen, **'Robustness evaluation for safety-critical systems utilising artificial neural network classifiers in operation: A survey,'** *In review, Manuscript submitted to the International Journal of Engineering Application of Artificial Intelligence*, 2023.

My contribution: I was the leading author and developed the research design and conceptualization, data collection and analysis, and research findings. Li and Oehmen exchanged ideas and commented on the draft.

Relevance to the thesis: This paper summarizes the definitions of ANN-SCS robustness in the system-, ANN models-, and input- levels, respectively, and presents the classification of approaches and remaining challenges for evaluating ANN-SCS robustness in operation. First, the paper synthesizes the findings from literature and standards of ML robustness to develop the ANN-SCSs robustness evaluation framework. Following this, the robustness assessment framework was used to map the remaining challenges at each level. The paper contributes to RQ3 by systematizing the robustness evaluation process of ANN-CSs in operation from a holistic perspective and pointing out the challenges to measuring the robustness of ANN-CSs in operation.

P4 J. Zhang, J. Li, and Z. Yang, **'Dynamic robustness evaluation for automated model selection in operation,'** *In review, Manuscript submitted to the International Journal of Information and Software Technology*, 2023.

My contribution: I was the leading author and developed the research design, data collection, experiments implementation, result analysis, and research findings. Li supervised this process through regular meetings. All authors discussed the results, and I wrote the paper based on the findings. Li and Yang commented on the paper.

Relevance to the thesis: This paper proposes using distance metrics to measure the robustness of multiple pretrained models on unlabeled inputs in operation to help ANN classifier end users choose a more robust model dynamically. The study compares and analyzes five candidate distance metrics applicable for ranking robustness. The results show that the Wasserstein distance (WD) [53] outperforms others when ranking multiple ANN models for CIFAR10-based models, while the Kullback-Leibler (KL) Divergence [54] demonstrates superior performance for ImageNet-based models. Maximum mean discrepancy (MMD) [55] can be used as the second option for both datasets. We have also found that the metrics assumptions and characteristics of the data to classify shall be considered when selecting the most appropriate metric. The paper contributes to RQ4 by going in-depth into one robustness evaluation scenario, i.e., addressing the challenge of ranking multiple ANN models' robustness in operation, identified in RQ3.

Furthermore, five secondary papers/technical reports were produced:

SP1 J. Li, J. Zhang, and N. Kaloudi, **'Could we issue driving licenses to autonomous vehicles?'** in *International Conference on Computer Safety, Reliability, and Security*, Sep. 2018, pp. 473–480.

SP2 N. H. C. Guzman, J. Zhang, J. Xie, and J. A. Glomsrud, **'A comparative study of STPA-extension and the UFoI-E method for safety and security co-analysis,'** *Reliability Engineering and System Safety*, vol. 211, 2021, Art. no. 107633.

SP3 J. R. Taylor, J. Zhang, I. Kozin, and J. Li, **'Safety and security analysis for autonomous vehicles,'** DTU Orbit, Lyngby, Denmark, 2021. [Online]. Available: https://orbit.dtu.dk/en/publications/safety-and-security-analysis-for-autonomous-vehicles

SP4 A. M. Staff, J. Zhang, J. Li, J. Xie, E. A. Traiger, J. A. Glomsrud, and K. B. Karolius. **'An empirical study on cross-data transferability of adversarial attacks on object detectors,'** in *41st Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence (SGAI 2021), AI-CyberSec 2021 Workshop.*

SP5 J. Zhang, J. Oehmen, and I. Kozin, **'Monitoring the robustness of safety critical artificial neural networks,'** *European Safety and Reliability Association Newsletter*, vol. 3, 2022, 4–5.

All the secondary papers provided complementary perspectives to this thesis. Secondary Paper SP1 explores the field of safety assurance for deep-learning-powered AVs by reviewing corresponding literature and industry standards. SP1 contributes to RQ1 by rooting the thesis in testing and verifying ANN-SCSs. SP2 comprehensively compares safety and security analysis methods for cyber-physical systems (CPSs). Two independent teams conducted the study to compare two different analysis methods. SP2's scope is complementary to the topic of this thesis due to the close relationship between the safety and robustness of CPSs. SP3 is the background report that extends the findings in main paper P2 by providing a framework for theoretical developments, actual risk analyses, hazard analysis methods' effectiveness, and quality studies of the safety of AVs. SP4 provides an example of devising transferable attacks on object detectors, addressing the challenge of assuring the robustness of a state-of-the-art object detector in practice. SP5 is a newsletter to summarize my thesis and communicate it to a broader European safety and reliability community audience. The newsletter also addresses the challenges and potential solutions in monitoring the robustness of an ANN classifier during system operations.

For SP1, I contributed to the literature review and paper writing by providing a background analysis on how to implement AV perception using deep learning methods and investigating approaches to verify the safety of AVs. For SP2, I contributed by co-conducting the analysis using the uncontrolled flows of information and energy (UFoI-E) method in the case study, co-evaluating the comparison of results, and writing a review of comparative studies of safety and security co-analysis methods. SP3 is a technical report on safety and security analysis for AVs; I contributed to the paper writing by providing a background analysis of ANN in the design of AVs and exploring how to perform risk analysis for ANNs. I also contributed to SP4 in the paper writing and revision. Regarding SP5, I was the lead author and developed the research design and conceptualization, proposed potential solutions to existing research gaps, and wrote the paper. However, as these papers only contribute indirectly to the research questions, they are left out of the main narrative of this thesis. See Table 1.2 for their connection to the research questions.

Table 1.2: Mapping of secondary papers/technical reports and research questions.

	SP1	SP2	SP3	SP4	SP5
RQ1	•				
RQ2		•	•		
RQ3				•	
RQ4					•

1.4.2 Research Contributions

This highly interdisciplinary thesis establishes a link between ML, safety, and system engineering. More specifically, the contributions of this thesis affect two research disciplines, i.e., Risk management and T&V of SCSs. The connections between the research questions, research papers, contributions, and the domain are illustrated in Figure 1.2. This thesis has four major contributions:

> ***C1:*** *The systematization of knowledge and understanding for T&V of ANN-SCSs.* This categorizes the state-of-the-art T&V methods for ANN-SCSs and identifies the challenges for advancing the state-of-the-art in T&V for ANN-SCSs.

***C2:** A new method for analyzing the influence of ANN robustness on the safety of AVs.* This integrates ANN robustness analysis into an overall safety and security analysis of SCSs.

***C3:** A systematization of knowledge and a framework for assessing ANN-SCSs' robustness in operation.* This includes a structured approach to summarize the concepts and methods of the robustness evaluation of ANN-SCSs in operation. From the state-of-the-art knowledge, we derive a framework that can be used to facilitate the systematic evaluation process by structuring the collection, integration, validation, and analysis of relevant data, including operational data, performance metrics, and environmental factors.

***C4:** New knowledge and understanding of how the robustness of multiple ANN models can be ranked using unlabeled data.* This empirically validates metrics that can effectively compare the robustness of ANN classification models in operation. Specifically, it highlights the effectiveness of distance-based metrics in ranking ANN classifier robustness for automated model selection. The findings shed light on the significance of advancing research in the area of dynamic robustness evaluation, which has been relatively overlooked but holds great importance in ensuring the robustness and performance of ANN-SCSs.

1.5 Structure of the Thesis

The thesis is composed of two parts. **Part I (Overview)** presents an introduction to the research work and provides an overview of the background, related work, research methods used, results achieved, and contributions made by the thesis. **Part II (Research Papers)** contains the four main research papers in full length and the abstracts of the secondary papers.

The rest of **Part I** is organized as follows:

- **Chapter 2** gives the theoretical background and context of this thesis. It provides an overview of SCS, risk management in ANN-SCS, and the relationship between ANN robustness and SCS risks. It establishes the foundation for understanding the challenges and research questions addressed in the thesis.
- **Chapter 3** reviews the existing literature and research efforts related to T&V of ANN-SCSs. It identifies the gaps and limitations in current approaches.
- **Chapter 4** depicts the research methodology and the approaches followed to address the research questions, including the systematic literature review process, the descriptive study, and case studies.

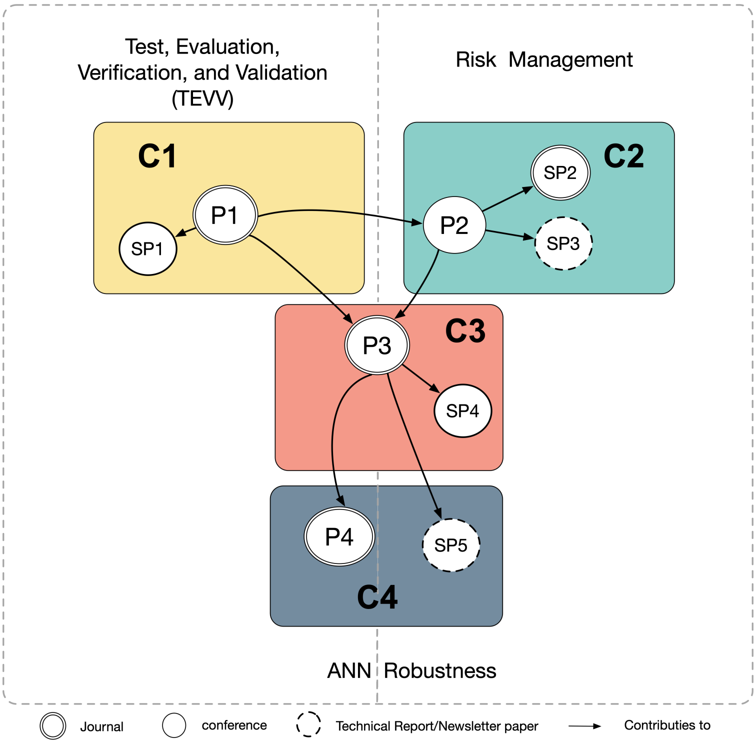


Figure 1.2: A schema of the research papers, contributions, and the domain.

- **Chapter 5** presents the findings and results of the research questions. It discusses the contributions made by each research paper and their relevance to the overall research objectives.
- **Chapter 6** discusses the implications of the research findings for academia and industry, explores the limitations of the research, and provides insights into future research directions in the field of ANN-SCSs.
- **Chapter 7** concludes the thesis by summarizing the main contributions, highlighting the key findings, and outlining potential avenues for future research in the area of ANN-SCSs.

Part II contains the main research papers in full length, along with the abstracts of the secondary papers, which provide complementary perspectives to the research.

2 Background and Context

This chapter introduces key concepts central to this thesis. We first introduce the concept of SCS, which relates to many ANN application domains investigated by this thesis, to provide context for the focus of this thesis. We then address the importance of risk management in ANN-SCSs, followed by an overview of testing, evaluation, verification, and validation (TEVV) procedures. Lastly, we highlight the relationship between ANN robustness and SCS risks, which shows that the ANN's robustness significantly contributes to the trustworthiness of ANN-SCSs.

2.1 Safety-Critical Systems (SCSs) and Safety Standards

SCSs are those for which failure may harm people, lead to economic loss, and/or cause environmental damages [1]. SCSs are used in many application areas, such as the automotive, process, and nuclear power industries and medical devices. Many SCSs are based on electrical, electronic, or programmable electronic (E/E/PE) technology. The essential system safety standard IEC61508 [56] provides a basis for the specification, design, testing, and operation of SCSs. The main goal of IEC61508 is to reduce the risk of failure to a tolerable level. It designates a safety integrity level (SIL) to determine the performance required to maintain and achieve safety. SIL ratings correlate to the frequency and severity of hazards. There are four SILs, i.e., SIL 1, SIL 2, SIL 3, and SIL 4. Each SIL represents an order of magnitude of risk reduction required, in which SIL 4 is the most demanding.

In the context of SCSs, safety-related functions can operate in two different modes: demanded mode and continuous mode. In demanded mode, a safety-related function is invoked only when a problem arises or is about to occur, such as the deployment of an airbag. On the other hand, in continuous mode, a safety-related function plays an active role, and a hazardous event may occur almost immediately if a dangerous failure of the function happens, as in the case of braking systems. Demanded mode can be further classified into two sub-modes: low-demand mode and high-demand mode, each of which corresponds to different operating conditions or usage scenarios that influence the required SIL.

Low-demand mode refers to situations where the system operates under normal conditions most of the time and only occasionally needs to respond to critical events or demands. Examples of low-demand mode applications include backup systems

Table 2.1: SIL target for low-demand SCSs [1].

Safety integrity level	**Average probability of failure on demand ($\mathbf{PFD_{avg}}$)**
SIL 4	$\geq 10^{-5}$ to $< 10^{-4}$
SIL 3	$\geq 10^{-4}$ to $< 10^{-3}$
SIL 2	$\geq 10^{-3}$ to $< 10^{-2}$
SIL 1	$\geq 10^{-2}$ to $< 10^{-1}$

Table 2.2: SIL for high-demand and continuous mode SCSs [1].

Safety integrity level	**Average frequency of dangerous failures per hour (PFH)**
SIL 4	$\geq 10^{-9}$ to $< 10^{-8}$
SIL 3	$\geq 10^{-8}$ to $< 10^{-7}$
SIL 2	$\geq 10^{-7}$ to $< 10^{-6}$
SIL 1	$\geq 10^{-6}$ to $< 10^{-5}$

that are rarely activated or emergency systems that are infrequently triggered. Conversely, high-demand mode denotes scenarios where the system is frequently exposed to safety-critical situations or demands. In high-demand mode, the system must continuously monitor, respond, and perform safety-critical functions to ensure the desired level of safety. Active safety systems in AVs or aircraft control systems are typical examples of high-demand mode applications.

The SIL targets for low-demand mode and high-demand mode define the specific SILs that must be achieved in each respective mode. These targets establish the required reliability and performance levels of the SCS to ensure its proper functioning and mitigate risks under different operating conditions. Table 2.1 presents the SIL targets for low-demand mode. For instance, achieving SIL 3 indicates a risk reduction factor of 1000 or more. For instance, achieving SIL 3 indicates that the SCS has been designed to significantly reduce the probability of hazardous events by a factor of 1000 or more compared to the baseline level of risk. Table 2.2 illustrates the SIL requirements and the associated range of average frequency of dangerous failures per hour (PFH) for high-demand and continuous mode SCSs.

2.2 Risk Management for ANN-Classifier-Based Safety Critical Systems (ANN-SCSs)

Risk Management for Traditional SCSs. The risk management process for traditional SCSs is illustrated in Figure 2.1, which outlines the steps of communication, consultation, establishing context, assessing, treating, monitoring, reviewing, recording, and reporting risk [57]. Risk management for ANN-SCSs represents an advancement over traditional SCSs. In this thesis, we focus on the first two steps of the procedure, namely establishing the context and assessing the risk.

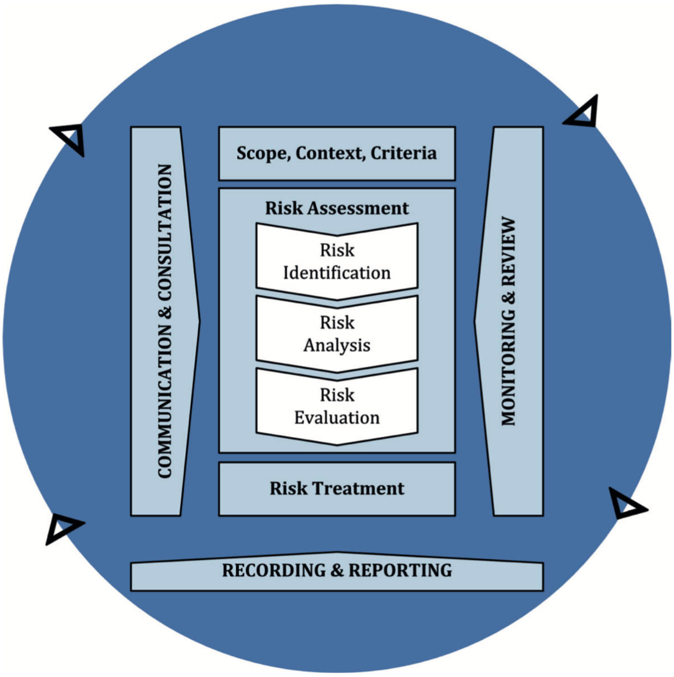


Figure 2.1: The risk management process from ISO 31000:2018 [57].

To establish the context, it is necessary to define the scope and purpose of the risk management process. For example, in the case of an autonomous car, the scope could be limited to normal driving or expanded to include emergency response and driver behavior in problematic situations. The scope determines the appropriate hazard identification methods and the frequency or probability of hazardous events. The hazard identification stage involves identifying events that could lead to an accident or system failure using various methods.

The risk assessment stage includes risk identification, analysis, and evaluation [11]. Risk analysis involves a detailed consideration of uncertainty, risk sources, consequences, likelihood, events, scenarios, controls, and their effectiveness. Analysis techniques may be qualitative, quantitative, or a combination, depending on the context and intended use. Finally, risk evaluation compares the risk analysis results with established risk criteria.

One commonly used method is failure mode and effects analysis (FMEA) [58].

FMEA involves identifying failure modes for each system component, determining their potential causes and effects on the overall system, and developing safety measures for each identified failure mode. This method helps in systematically analyzing and mitigating potential risks.

For more complex systems, various methods like fault tree analysis (FTA) [59], cause consequence analysis [60], hazard and operability analysis [61], and systems-theoretic process analysis (STPA) [62] are used. FTA uses event and logic symbols to construct a logic diagram that represents the failure logic of a system, facilitating communication with managers, designers, and operators. By incorporating failure rates and relevant data, FTA can estimate the frequency of occurrence of undesired events. FTA is a deductive approach that can trace system failures back to one or more failures at lower levels. For example, Gupta et al. [63] utilized FTA to identify combinations of component failures and human errors that could result in specific undesired events at the system level. STPA considers the system as a hierarchy of control loops and takes into account unsafe interactions between components. It provides a broader perspective by including “emergent” failure types associated with control loops as a whole, rather than focusing solely on individual components [64]. This extension of STPA enhances the understanding of potential failures and their implications for system safety.

It is important to note that no single method can identify all hazards in a wholly engineered SCS. For example, Taylor et al. [11] used a range of ways to analyze the safety and security of AVs, concluding that brainstorming was effective in identifying different accident types, and FTA was effective in identifying accident causes.

Difference between ANN-SCS Risks and Conventional Software Risks. According to Leveson’s work on software safety analysis [65], software can be analyzed from a safety perspective like physical components. Software FTA [66, 67] is a structured approach for identifying and analyzing potential faults in SCSs. However, traditional software safety analysis requires a detailed design representation and a list of hazards or safety risks to be analyzed. Accidents in SCSs can result from software errors that occur when incorrect assumptions are made regarding the correctness and completeness of the software specification. These assumptions are meant to define the expected behavior of the software under all possible scenarios [68].

ANN algorithms offer substantial benefits in addressing complex design issues for SCSs, yet, like conventional software, ANN-SCSs have their specific risks. ANN-SCSs’ unique challenges require an expansion of traditional risk frameworks and methodologies. Two key challenges are noteworthy. Firstly, ANN-based systems fundamentally rely on data. Hence, if the collected data to train the model fail to represent real-world scenarios accurately, ANN algorithms could perform unpredictably. Secondly, the inherent nondeterministic and non-robust nature of ANNs can complicate the creation of reproducible tests. Given these challenges, the focus of risk assessment for ANN-SCS has expanded beyond just the algorithm itself to encompass broader aspects of the process, specifically data management [69, 70] and ANN model design and verification [71, 72].

Table 2.3: Examples of real-world incidents caused by lacking robustness in operation.

Case	**Incident Description**	**Cause**	**Affected Attribute**
1	IBM Watson for Oncology frequently gave unsafe and erroneous cancer treatment advice to patients[78].	Lacking distributional robustness: a few synthetic cancer patient data were used for training instead of real patient data.	Safety
2	Apple's facial recognition ID system was fooled by 3D-printed masks [76].	Lacking adversarial robustness: the anti-spoofing neural network only considers cosmetic changes, wearing a scarf, or the presence of glasses on the face.	Security
3	Tesla autopilot failed to recognize a white truck against a bright sky [75].	Lacking corruption robustness: Image contrast	Safety
4	Amazon's facial recognition software mistakenly identified members of the U.S. congress [77].	Lacking distributional robustness: the facial identification system demonstrated better performance for lighter-skinned faces but encountered difficulties in recognizing darker-skinned faces.	Reliability

Traditional safety assurance for SCSs is facilitated by well-established industry standards, prescriptive development processes, and verification techniques/tools that provide engineers with evidence to demonstrate adequate system safety. However, the incorporation of ANN algorithms in SCSs complicates the process of estimating the probability of failures or accidents. Consequently, there is an inevitable shift towards greater dependence on empirical demonstrations of safety through both simulated and operational testing [73, 38].

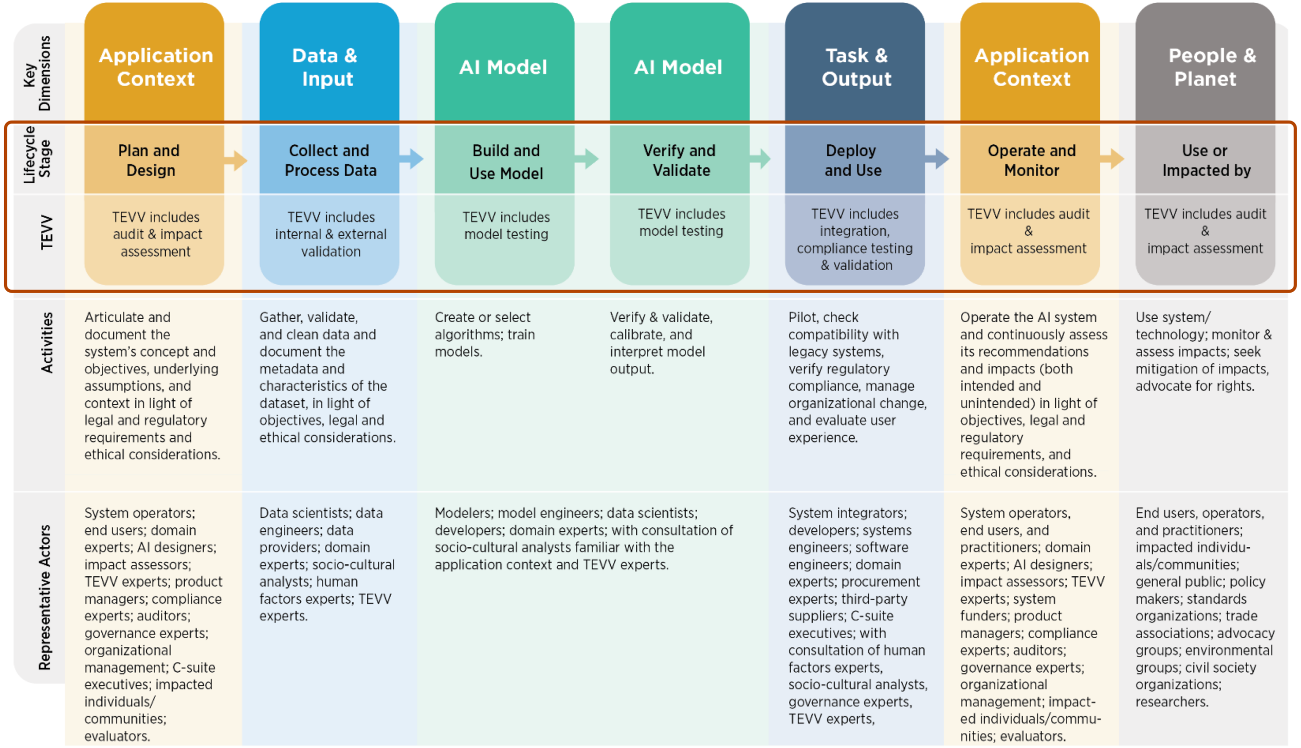
Relations between ANN Robustness and SCS Risks As stated in the international standard on trustworthiness in AI [74], an AI system's ability can be assessed using several attributes, including reliability, resilience, and robustness. Robustness is defined as *a system's ultimate ability to maintain its performance level under any circumstances, including external interference or harsh environmental conditions* [74]. Robustness is a crucial property that poses new challenges in the context of ANN classifier-based systems. The robustness of an ANN-SCS can affect other adjacent system attributes, as ANN robustness is necessary to maintain properties such as safety (e.g., for AVs) [75], security (e.g., access control) [76], and reliability (e.g., commercial facial recognition software) [77]. In Table 2.3, we present examples of real-world incidents due to the lack of robustness in ANN models.

2.3 Machine Learning (ML) Lifecycle and Test, Evaluation, Verification, and Validation (TEVV) Tasks

ML is a subfield of AI that aims to mimic how humans learn using data and statistical modeling techniques [79]. Amershi et al. [80] proposed a nine-stage workflow for ML, which starts with defining model requirements and then progresses through data collection, data cleaning, data labeling, feature engineering, model training, model evaluation, model deployment, and model monitoring. This workflow involves feedback loops where different stages may be revisited, such as model training returning to feature engineering and model evaluation or monitoring looping back to previous stages.

Given the growing interest in AI technology for SCSs, risk management for AI systems (e.g., ANN-SCSs) should be continuous, timely, and performed throughout the entire system lifecycle. National Institute of Standards and Technology (NIST) [1] published an AI risk management framework (Figure 2.2) that highlighted the importance of test, evaluation, verification, and validation (TEVV) processes for AI-based system design, development, deployment, and operation (e.g., ongoing monitoring) [51].

[1] https://www.nist.gov/



Key Dimensions	Application Context	Data & Input	AI Model	AI Model	Task & Output	Application Context	People & Planet
Lifecycle Stage	Plan and Design	Collect and Process Data	Build and Use Model	Verify and Validate	Deploy and Use	Operate and Monitor	Use or Impacted by
TEVV	TEVV includes audit & impact assessment	TEVV includes internal & external validation	TEVV includes model testing	TEVV includes model testing	TEVV includes integration, compliance testing & validation	TEVV includes audit & impact assessment	TEVV includes audit & impact assessment
Activities	Articulate and document the system's concept and objectives, underlying assumptions, and context in light of legal and regulatory requirements and ethical considerations.	Gather, validate, and clean data and document the metadata and characteristics of the dataset, in light of objectives, legal and ethical considerations.	Create or select algorithms; train models.	Verify & validate, calibrate, and interpret model output.	Pilot, check compatibility with legacy systems, verify regulatory compliance, manage organizational change, and evaluate user experience.	Operate the AI system and continuously assess its recommendations and impacts (both intended and unintended) in light of objectives, legal and regulatory requirements, and ethical considerations.	Use system/technology; monitor & assess impacts; seek mitigation of impacts, advocate for rights.
Representative Actors	System operators; end users; domain experts; AI designers; impact assessors; TEVV experts; product managers; compliance experts; auditors; governance experts; organizational management; C-suite executives; impacted individuals/communities; evaluators.	Data scientists; data engineers; data providers; domain experts; socio-cultural analysts; human factors experts; TEVV experts.	Modelers; model engineers; data scientists; developers; domain experts; with consultation of socio-cultural analysts familiar with the application context and TEVV experts.		System integrators; developers; systems engineers; software engineers; domain experts; procurement experts; third-party suppliers; C-suite executives; with consultation of human factors experts, socio-cultural analysts, governance experts, TEVV experts,	System operators, end users, and practitioners; domain experts; AI designers; impact assessors; TEVV experts; system funders; product managers; compliance experts; auditors; governance experts; organizational management; impacted individuals/communities; evaluators.	End users, operators, and practitioners; impacted individuals/communities; general public; policy makers; standards organizations; trade associations; advocacy groups; environmental groups; civil society organizations; researchers.

Figure 2.2: Test, evaluation, verification, and validation processes throughout an AI's lifecycle [51].

This framework underscores the necessity of robustness evaluation in both the developmental and operational stages, enhancing our understanding of ANN-SCSs' inherent robustness. In the NIST AI Risk Management Framework, a key insight brought to light is the discrepancy between risk assessments conducted in controlled environments and those in real-world operational settings. While laboratory measurements can offer valuable insights prior to deployment, they may fail to represent the risks involved in real-world applications accurately.

During the pre-deployment phase, data scientists engage in rigorous model design, development, and evaluation. The typical approach is to perform offline evaluations, testing a model's performance based on metrics such as accuracy, precision, recall, etc., against a test set that mirrors the training data. However, this test set might not capture all possible scenarios the model might encounter in real-world settings. For instance, road conditions in a snowy environment or scenarios involving malicious perturbations might be underrepresented in the test set, leading to potential performance gaps and vulnerabilities in the model's robustness.

Given the inherently unpredictable and varying nature of real-world data, ensuring that ML models are trained on comprehensive and relevant data that can generalize their predictions to rare and potentially catastrophic scenarios is challenging. The divergence between training data and real-world data is recognized under various terms in academic literature, including distribution shift [81], nonstationarity [82], or training-serving skew [83]. These issues highlight the need for robustness evaluations that extend beyond laboratory settings and into the operational phase.

2.4 Existing Guidelines, Standards, and Regulations, and Their Applicability to Evaluating ANN-SCSs

This section provides an overview of the various existing guidelines, standards, and regulations that govern the evaluation of ANN-SCSs. These standards serve to manage the risks associated with deploying ANNs, covering aspects like autonomous driving, adaptive systems, healthcare, and data safety.

ISO/PAS21448: Road Vehicles—Safety of the Intended Functionality (SOTIF) [84]. SOTIF is designed to complement ISO 26262. SOTIF offers guidance on identifying hazardous situations that may arise from the limitations of autonomous driving systems (ADSs). It also provides recommendations for verification and validation activities, such as analyzing triggering events, accounting for sensors' limitations, analyzing environmental conditions and operational use cases, analyzing boundary values, and examining algorithms and their decision paths.

"Safety First for Automated Driving" White Paper [37]. In June 2019, a group of 11 prominent automotive and automated driving stakeholders, including Audi, Baidu, BMW, Intel, Daimler, and VW, published a white paper titled "Safety First for Automated Driving." The report emphasizes the importance of safety in the design, verification, and validation of ADSs, particularly at SAE[2] levels 3–4 (conditional/high automation), and the need for continuous performance monitoring. Additionally, the report explores the use of DNNs in safety-critical scenarios, focusing on 3D object detection as a prominent example. It provides a valuable resource for understanding safety considerations in automated driving.

UL4600: Standard for Safety for the Evaluation of Autonomous Products [86]. UL4600 is a safety standard that outlines the safety case framework for AVs. The safety case [87] is a structured argument, supported by evidence, that demonstrates that a system is acceptably safe for its intended use in its intended operational environment. UL4600 is dedicated to ML to provide an understanding of the risks of modern ML methods and ways to mitigate them in the following topics (quoted from [88]):

- Definition of operational design domain (e.g., weather, scenarios)
- Machine learning faults (e.g., training data gaps, brittleness)
- External operational faults (e.g., other vehicles violating traffic rules)
- Faulty behavior by non-driver humans (e.g., pedestrians, lifecycle participants)
- Nondeterministic, variable system behavior (e.g., test planning, acceptance criteria)
- High residual unknowns (e.g., requirements gaps and post-deployment surprises)
- Lack of human oversight (e.g., operational fault handling, passenger handling)
- System-level safety metrics (e.g., use of leading and lagging metrics)
- Transitioning the system to degraded modes and minimum risk conditions

FAA TC-16/4 Verification of Adaptive Systems [89]. The U.S. Federal Aviation Administration (FAA) report aims to analyze adaptive systems' certifiability. In this report, adaptive systems are defined as "software having the ability to change behavior at runtime in response to changes in the operational environment, system configuration, resource availability, or other factors." FAA TC-16/4 highlights the challenges of verifying and assuring the safety of using ANNs in aircraft products such as air traffic control systems. However, it does not provide explicit verification guidelines for developing and using ANNs.

FDA April 2019 Report [90]. The U.S. Food and Drug Administration's (FDA's) Proposed Regulatory Framework for Modifications to AI/ML-based Software as a Medical Device (SaMD) focuses on risks associated with software modifications in

[2] The Society of Automotive Engineers (SAE) proposed six levels of autonomous driving [85]. A level 0 vehicle has no autonomous capabilities, and the human driver is responsible for all aspects of the driving task. For level 5 vehicles, only the ADS manages the driving tasks.

ML-based systems. It offers general information on the regulation and certification of AI software. The International Medical Device Regulators Forum (IMDRF) defines SaMD as software that serves a medical purpose and is not part of a hardware medical device, such as diagnostic software for identifying tumors or biometric signal processing software. The proposed framework suggests a total product lifecycle (TPLC) regulatory approach to evaluate and monitor SaMD from premarket development to post-market performance.

Data Safety Guidance [91]. The Data Safety Initiative Working Group of the SCSs Club has created the Data Safety Guidance to offer recommendations for using data in SCSs. The guide includes definitions, principles, processes, objectives, and advice, and it summarizes the challenges related to applying data safety techniques in ML/AI SCSs. These challenges include poor representation of rare cases in test data, degradation of sensor data, and composite data generated from merging lidar, radar, and camera.

3 Related Work

This chapter presents a concise overview of the relevant research conducted in the field, aligning with the research questions addressed in this thesis. It offers a comprehensive review of the existing literature and research efforts related to the T&V of ANN-SCSs, emphasizing the gaps and limitations in addressing the distinct challenges posed by ANN-SCSs.

3.1 Testing and Verification of ANN Classifiers in SCSs - RQ1

Despite extensive research on the testing and verification (T&V) of ANNs in the past decade, there have been relatively few review articles published on this topic [34, 35, 92, 93, 94, 95]. Many of these studies, including Taylor et al. [34], have concentrated on specific domains such as flight control systems or analyzed existing standards in a specific industry, such as the automotive industry [35, 92]. Taylor et al. [34] categorized verification and validation (V&V) methods before 2003 into five groups: automated testing and testing data generation methods, run-time monitoring, formal methods, cross-validation, and visualization. However, their review did not cover new T&V methods for modern neural networks developed post-2011. Moreover, many of these traditional V&V techniques have proven inadequate for verifying modern ANNs in several instances.

Regarding the literature review methodologies, several studies, including [34, 35, 92, 93], utilized ad hoc literature review (ALR) approaches, while [94] and [95] opted for the systematic literature review (SLR) approach. Nonetheless, these SLR-focused works concentrated more on interpreting AI rather than specifically examining T&V techniques for ANNs.

To the best of our knowledge, when we worked on RQ1, no SLR was available to provide a comprehensive and structured review of our defined study context (i.e., testing and verifying ANN-SCSs). A more comprehensive SLR covering the key aspects of T&V activities for ANN-SCSs can help researchers identify the research gaps in this area and help industrial practitioners choose proper verification and certification methods for safety purposes. In Section 5.1 of the thesis, we address this challenge by providing a systematic summary of knowledge and understanding related to the challenges in testing and verifying ANN for SCSs.

3.2 Risk Analysis of ANN-SCSs - RQ2

A comprehensive understanding of risk factors, both qualitatively and quantitatively, is crucial for constructing a robust risk model. This understanding necessitates identifying potential hazards and outlining the accident scenarios that may stem from them. It also involves gaining an in-depth knowledge of all factors influencing these scenario outcomes. However, it's often impractical to incorporate every system detail and scenario into the risk model due to the excessive time and resources required. The challenge is exacerbated when applying ANN in SCSs, as these systems can make it more difficult to estimate failure or accident probabilities [96, 97].

Formal techniques are being developed to represent and reason about systems that include learning-based algorithms [98, 21, 71]. However, these techniques are in their early stages and cannot yet be integrated into conventional risk assessment methodologies. Therefore, designers and regulators have a limited ability to use deductive inference to demonstrate that ANN-SCSs are adequately safe. Similarly, there is a lack of inductive tools for incorporating these approaches into statistical risk assessment methodologies. The difficulty lies in predicting how complex training data will shape the future operation of these systems when exposed to their operating environments, as past interactions might not provide accurate hazard predictions.

FTA, introduced in 1962 at Bell Telephone Laboratories during a safety evaluation of the Minuteman Missile launch control system [59], is one of the most widely used methods for ensuring reliability and safety in complex systems. FTA provides an overall framework for hazard identification. Supplemental techniques like FMEA [58] can further detail mechanical and electrical component failures, while STPA [62] can be used to analyze control hierarchies.

If an ANN's performance solely depended on independent variations in input parameters, conventional FTA could be utilized with discrete events, such as "perturbation exceeds the performance threshold." However, in many instances, an ANN's performance depends on two or more continuously varying disturbance parameters. In such cases, fuzzy fault tree analysis (FFTA) [99] can be considered. Still, establishing fuzzy membership functions can be a challenging and subjective process. Therefore, it is necessary to enhance FTA methodologies to effectively incorporate ANNs into comprehensive hazard identification and risk analysis. In Section 5.2 of this thesis, we address this challenge by proposing an extended FTA method that enables the analysis of the impact of ANN robustness on the safety of AVs.

3.3 Robustness Evaluation of ANN-SCSs in Operation - RQ3

There is a significant body of research exploring the robustness of ANN models [100, 101, 102]. However, none of these studies offers a dedicated overview of metrics and corresponding methods for evaluating these models' robustness in operational environments.

França et al. [103] evaluated techniques used for measuring the robustness of DNN models. Their research predominantly focused on methods employed to test image classifiers' robustness, particularly in the context of AVs. One technique often used is fuzz testing [104], which leverages invalid or unexpected inputs to test system resilience. Fault injection has also been proposed as a robustness testing method. In the survey on ML testing [2], several fault injection-based methods are identified to simulate hardware errors of AVs to evaluate their robustness.

In laboratory settings, robustness analysis has been extensively studied in relation to adversarial ML [39, 40, 41, 42]. Adversarial ML aims to guard against attacks on the system, evaluate ML algorithms' worst-case robustness, and measure these algorithms' progress towards human-level capabilities [105]. For instance, Carlini et al. [105] outlined common evaluation pitfalls when assessing robustness, including neglecting different attack methods or natural perturbations, such as noise variants [106, 107, 108].

In contrast, non-adversarial robustness has received less attention. Drenkow et al. [43] conducted a systematic review on non-adversarial robustness within the computer vision domain, categorizing their findings based on robustness tactics such as architecture, data augmentation, and optimization.

Many of these methods may not be directly applicable in operational environments as they often rely on the availability of labeled data for robustness evaluation. However, these labeled data may not always be accessible or might be delayed due to the high cost of labeling in operation. Furthermore, these techniques typically focus on the robustness of a single ML model, such as an ANN classifier. For ANN-SCSs, it's crucial to consider robustness at both the individual model and the system level when assessing operational contexts.

Microsoft researchers have considered robustness across several application domains [10], proposing a unified taxonomy and framework to address ANN model failures. Although this work provides a comprehensive perspective on ML robustness risk in operation, it does not specify metrics and methods tailored to each operating environment [10].

To date, to our knowledge, no review paper has explicitly organized definitions, metrics, and methods specifically targeting the robustness evaluation of ANN-SCSs in operation. In Section 5.3 of this thesis, we tackle this challenge by providing a systematic summary of knowledge and a framework for assessing the robustness of ANN-SCSs in operational settings. The differences between the focus of our survey from existing surveys is illustrated in Table 3.1.

Table 3.1: Chronological comparison of focuses of previous surveys.

Year	Survey	Studying Definitions	Studying Metrics and Methods	Identifying Challenges	Operation
2019	Kumar et al. [10]				✓
2019	Carlini et al. [105]	✓	✓	✓	
2020	Zhang et al. [44]		✓	✓	
2020	Huang et al. [42]		✓	✓	
2021	França et al. [103]		✓	✓	
2021	Drenkow et al. [43]	✓	✓	✓	
2022	Mohseni et al. [45]		✓	✓	
	Our survey	✓	✓	✓	✓

3.4 Dynamic Ranking of ANN Robustness in Operation - RQ4

We foresee a growing trend of operating multiple versions of models, such as ChatGPT [109], for three primary reasons: 1) rapid AI/ML advancements promote flexibility and experimentation; 2) multiple models reduce the risk of failure or underperformance; and 3) diverse environments and contexts necessitate optimized models for specific needs. This has been demonstrated in many non-safety-critical applications. For instance, cloud providers switch between different models based on service-level agreements (SLAs) in operational settings to trade between computational cost and service accuracy [110]. In light of this emerging trend, it is crucial to develop dedicated methods for continuously comparing the robustness of multiple ANN models and deciding which model should be used within the SCS in operation. Methods developed for dynamic robustness evaluation of ANN classifiers in operation should consider the following two challenges: 1) the ground truth (labels) is often not accessible or delayed; 2) the acceptable level of performance for ANN models must be determined.

Several studies have proposed test-selection-based methods [111, 112] to rank multiple models with minimum labeling effort. For instance, Ma et al. [111] proposed various metrics based on model uncertainty to identify data likely to cause misclassification. Meng et al. [112] combined majority voting [113] and item discrimination [114] techniques to measure the discrimination of inputs and select a set of "error-inducing inputs" to differentiate the robustness of multiple ANN models. However, test-selection-based methods, which rely on labeling a subset of data, are not suitable for addressing our specific problem of unlabeled robustness ranking in operation.

Labeling-free model performance estimation is a task aimed at predicting the accuracy of models on test sets without access to ground-truth labels. For example, AutoEval [115] and SelfChecker [116] propose learning an accuracy regression model using a synthetic meta-dataset, resulting in accurate predictions of model accuracy for real-world unlabeled test datasets. However, the methods presented in [115, 116], require a separate supervisor model to monitor and predict the performance of a single deployed ANN model. While it is technically possible to train multiple supervisor models to monitor and predict the performance of multiple deployed ANN models, there are several practical challenges associated with this approach. Firstly, training and maintaining multiple supervisor models can be computationally expensive and time-consuming. Secondly, each supervisor model may have its own biases and limitations, leading to inconsistent and incomparable results across different models.

Regarding determining the acceptable level of performance for ANN models, industry best practices often involve detecting drift to indicate whether an ANN model's performance is above the acceptable level [117]. Data drift detection primarily focuses on identifying changes in the input data, while model shift detection aims to detect shifts in the output of deployed ANN classifiers. Measuring distribution differences between input data to derive model robustness is unreliable since data shifts can often have trivial impacts on model performance [118]. There are two main approaches to detecting model shifts: statistical-based and distance-based. Statistical-based methods rely on a given confidence level, usually 95%, to determine if a shift is detected. However, this approach does not measure the magnitude of shift and provides only a binary (Yes/No) result, making it unsuitable for ranking multiple models. Distance-based approaches measure the distance between the distributions that generate the training and test data. Goldenberg and Webb [119] assessed the practical application of several state-of-the-art distance metrics for estimating the magnitude of model shifts. Their study showed that distance-based methods offer an alternative for estimating performance degradation. However, further investigation is needed to determine whether these techniques can effectively compare and rank the robustness of multiple ANN classifiers. This motivates us to explore various distance-based metrics and examine their effectiveness in ranking the robustness of multiple ANN classifiers during operation. In Section 5.4 of this thesis, we contribute new knowledge and understanding by demonstrating how the robustness of multiple ANN models can be ranked using unlabeled data.

4 Research Methodology

While the choices of specific methods in this thesis are separately justified within the methodological section of each paper, this chapter is devoted to how the methodologies of the individual papers fit together in response to the research questions of the thesis.

4.1 Overview

In this thesis, we mainly focus on the **robustness of ANN-SCSs** and associated domains. We have adopted a design research methodology (DRM) as a structured process [120] to systematically address our RQs in sequential order. In particular, we follow two distinct research strategies within this methodology, inspired by Robson's classification [121]:

- *Exploratory:* This strategy is about discovering what is happening, seeking new insights, and generating ideas and hypotheses for future research.
- *Improving:* This strategy aims to enhance a specific aspect of the studied phenomenon.

The initial stage, research clarification, aims to explain the research problem at hand and formulate a clear and realistic overall research plan. To answer RQ1, we performed SLR to gain a deeper understanding of the T&V of ANN-SCSs. This enabled us to identify the gaps and challenges in the current state of the art, leading to the formulation of RQ2 and guiding our subsequent research steps.

The second step of the thesis aims to address the research problem by leveraging the findings from the previous stages for problem-solving and empirical development. We focused on addressing RQ2 by filling a gap in risk analysis methods for assessing the influence of ANN robustness on SCS safety.

Based on the insights acquired when answering RQ1 and RQ2, it became apparent that an additional systematization of knowledge study was necessary to gather more information about the robustness evaluation of ANN-SCSs in operation. This additional study focused on answering RQ3, aimed to improve our understanding

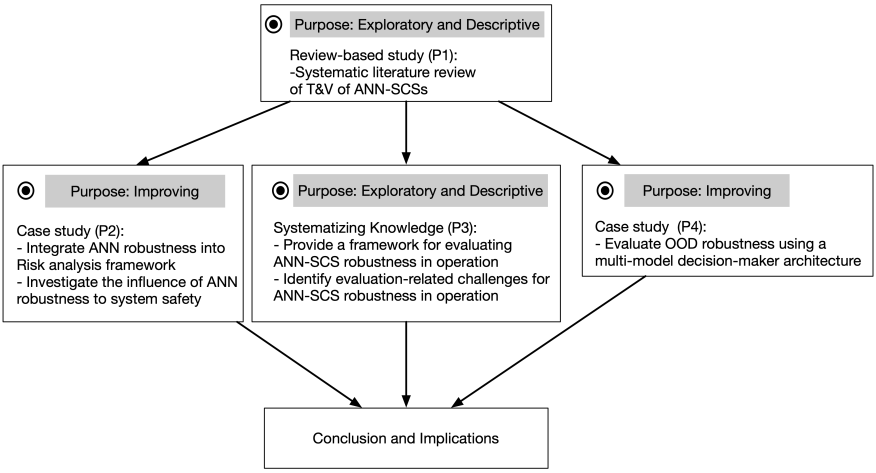

Purpose: Exploratory and Descriptive
Review-based study (P1):
-Systematic literature review
of T&V of ANN-SCSs
Purpose: Improving
Case study (P2):
- Integrate ANN robustness into Risk analysis framework
- Investigate the influence of ANN robustness to system safety
Purpose: Exploratory and Descriptive
Systematizing Knowledge (P3):
- Provide a framework for evaluating ANN-SCS robustness in operation
- Identify evaluation-related challenges for ANN-SCS robustness in operation
Purpose: Improving
Case study (P4):
- Evaluate OOD robustness using a multi-model decision-maker architecture
Conclusion and Implications

Figure 4.1: Research design of this thesis.

of evaluating the robustness of ANN-SCSs by systematizing the current state of the art and providing insights into the definition, methods, and metrics of robustness evaluation during operation.

The findings from answering RQ3 revealed that the existing support for evaluating and comparing the robustness of multiple ANNs for automated model selection was ineffective or insufficient. Consequently, we decided to concentrate on a systematic evaluation of the use and usefulness of the existing support to answer RQ4. In the study, we evaluated the applicability and usefulness of distance-based metrics for ranking the OOD robustness of multiple ANN models and automating the selection of the best model during operation in the context of multi-model decision-makers.

The subsequent sections will provide detailed accounts of the research activities undertaken to address the research questions in this thesis. The logical structure and research strategies used in this thesis are illustrated in Figure 4.1.

4.2 Literature Reviews—RQ1

In the first stage of exploration, the thesis focuses on RQ1. The objective of this research was to shape the extensive research problem addressed in this thesis and recognize the necessity and challenges of T&V for ANN-SCS. Hence, the focus of this research was further refined into three sub-research questions as follows:

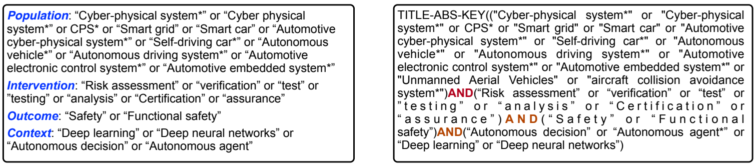
- RQ 1.1: *What are the profiles of the studies focusing on testing and verifying ANN-SCSs?*
- RQ 1.2: *What approaches and associated tools have been proposed to test and verify ANN-SCSs?*

- RQ 1.3: *What are the limitations of current studies with respect to testing and verifying ANN-SCSs?*

Research method. To address RQ 1.1-1.3, we conducted an SLR of T&V approaches for ANN-based SCSs (i.e., Paper P1 in Part II). The review protocol was based on established guidelines by Kitchenham [122] and others [123, 124, 125] and included four stages: developing a search strategy, setting inclusion and exclusion criteria, outlining the selection process, and establishing a framework for data extraction and synthesis. We employed the population, intervention, outcome, context (PIOC) criteria [122] for defining search terms and used a five-step thematic analysis [126] for data analysis. This structured approach helped us ensure the reliability of our findings and effectively structure the vast research field of ANN robustness evaluation in SCSs.

Research steps. We first collected relevant research papers based on the following steps:

- *Search strategy.* In this SLR, the search terms were formulated to identify relevant papers that address system/component T&V for improving the safety or functional safety of ANN-SCSs. Fig. 4.2 presents the search terms formulated based on the PIOC criteria. These search terms were refined through trial searches, and the final search was conducted in six digital libraries: Scopus, IEEE Xplore, Compendex EI, ACM Digital Library, SpringerLink, and Web of Science (ISI).



Population: “Cyber-physical system*” or “Cyber physical system*” or CPS* or “Smart grid” or “Smart car” or “Automotive cyber-physical system*” or “Self-driving car*” or “Autonomous vehicle*” or “Autonomous driving system*” or “Automotive electronic control system*” or “Automotive embedded system*”

Intervention: “Risk assessment” or “verification” or “test” or ”testing” or “analysis” or “Certification” or “assurance”

Outcome: “Safety” or “Functional safety”

Context: “Deep learning” or “Deep neural networks” or “Autonomous decision” or “Autonomous agent”

TITLE-ABS-KEY(("Cyber-physical system*" or "Cyber-physical system*" or CPS* or "Smart grid" or "Smart car" or "Automotive cyber-physical system*" or "Self-driving car*" or "Autonomous vehicle*" or "Autonomous driving system*" or "Automotive electronic control system*" or "Automotive embedded system*" or "Unmanned Aerial Vehicles" or "aircraft collision avoidance system*")**AND**(“Risk assessment” or “verification” or “test” or ”testing” or “analysis” or “Certification” or “assurance”)**AND**(“Safety” or “Functional safety”)**AND**(“Autonomous decision” or “Autonomous agent*” or “Deep learning” or “Deep neural networks”)

Figure 4.2: Search terms.

- *Inclusion and exclusion criteria.* The inclusion criteria require papers to have a context in SCSs, focus on T&V approaches for ANN-SCSs, and address modern ANNs. Exclusion criteria include non-peer-reviewed papers, papers not written in English, papers without full-text availability, and papers not relevant to modern ANNs. These criteria were applied to ensure the relevance and quality of the selected papers for the SLR.

- *Selection process.* The selection process for the SLR involved multiple stages. Initially, a search string was used to retrieve papers from six digital libraries, resulting in 950 papers after deduplication. These papers were then filtered based on title, keywords, and abstracts, resulting in 105 potential papers. After reading the introduction and conclusion of these papers, 27 papers were selected. Further snowballing was conducted, resulting in the inclusion of

56 additional papers. In total, 83 primary studies were selected for detailed analysis, and the selection process was verified through cross-checking and discussions between the authors.

The steps of the data analysis to answer RQ1.1-1.3 were as follows:

- *RQ1.1: Examining the profiles of the studies focusing on testing and verifying ANN-SCSs.* In this phase, we analyzed the distribution of the surveyed studies, considering their publication years, research types, and application domains. This detailed exploration provided us with a comprehensive understanding of the current landscape.
- *RQ1.2: Categorizing approaches for testing and verifying ANN-SCSs.* We employed the thematic analysis method [126], through which we identified five higher-order themes along with several subthemes. To strike a balance between accuracy and simplicity in categorization, we chose to assign each study to only one category, reflecting its primary contribution. This approach facilitated a structured representation of the T&V techniques for ANN-SCSs.
- *RQ1.3: Identifying challenges in testing and verifying ANN-SCSs.* Despite significant efforts from academia and industry, there is a notable gap between existing T&V methods for ANNs and safety standards such as IEC 61508 [56] and ISO 26262 [127]. To bridge this gap, we first mapped the methods identified in our review to these standards, using IEC 61508 [56] as the reference standard due to its influence on ISO 26262 [127]. We employed the safety integrity properties outlined in IEC 61508-3 and IEC 61508-7 as indicators to assess the extent to which current methods fulfill the T&V requirements for ANN-SCSs. Through this mapping process, we identified and summarized the key challenges involved in testing and verifying ANN-SCSs, providing valuable insights into the current landscape and the obstacles that need to be overcome.

Further details on the research design are available in Section 3 of Paper P1 in Part II.

4.3 Case Study—RQ2

The second stage of this thesis focuses on RQ2. The objective of this research was to gain a deeper understanding of the interplay between ANN robustness and overall system safety, ultimately facilitating a more integrated method of risk analysis.

Research method. To fulfill this objective, we selected the case study as our research methodology. This choice allowed us to develop a new risk analysis method and carry out an empirical evaluation based on real-world data. The case study methodology is ideal for extracting detailed insights from complex scenarios, especially when the goal is to comprehend the subject [128] systematically. In our work, presented in

Paper P2 of Part II, we aimed to introduce a new method for assessing the impact of ANN classifier robustness on the total system's safety. By following the research protocol suggested in [129], which includes designing the case study, collecting data, analyzing data, and formalizing results, we can ensure the reliability and validity of our findings.

Research steps. The steps of the research performed to answer RQ2 were as follows:

- *Analyzing safety and security threats in ANN-based perception for AVs.* We conducted a comprehensive analysis of various factors that have the potential to impact the performance of an ANN. This analysis was carried out through morphological brainstorming and by reviewing accident reports to identify key safety and security threats. This qualitative investigation allowed us to understand the vulnerabilities of ANNs, including incorrect deductions from perception systems, which have proven to be a major cause of disengagement incidents in AVs. By examining these potential threats, we constructed a risk profile encapsulating both safety and security aspects, providing us with a deep understanding of the risks and threats associated with the application of ANNs in AVs.

- *Proposing the extended fault tree analysis (FTA) including ANN components.* We introduced an extended FTA method that enables the risk analysis of ANN components in SCS. We believe that reasoning about the probability and consequences of adverse events is needed when ANN components are used in SCSs. Given the challenges in analyzing ANNs and the impracticality of extensive real-world testing (as highlighted by Kalra and Paddock's statistical assessment [130]), we propose a new approach that links reliability and safety through risk analysis and component reliability assessments. By incorporating reliability assessments into the risk analysis process, we can identify potential failures or weaknesses in the ANN components that may impact the system's overall safety. This method integrates ANN failures into the FTA by considering the ANN as a "black box" with specific functional requirements, such as recognizing a stop sign in AVs. Failure modes are then defined based on these functions, including failures like incorrect identification or classification of an image. The network's failure probability is considered as the likelihood of an observed image falling outside the network's reliable domain or in a domain where the network lacks robustness. Using this approach, we conduct hazard analysis using standard methods to identify the causes of such occurrences. This method places emphasis on developing measures of ANN robustness against different threats. The goal is to understand ANNs' performance limits and the conditions in the operating environment that may challenge these limits. The model we propose also includes the potential influence of ANN failures on the overall system risk. The aim is to evaluate the impact of these failures on the safety of the SCS in which the ANN is embedded. This comprehensive approach to risk analysis goes beyond examining the ANN in isolation and instead considers the broader system context.

- *Evaluating the proposed methodology using a case study of a traffic sign recognition neural network.* To evaluate the effectiveness of our proposed methodology, we conducted a case study on a traffic sign recognition neural network. The focus of our evaluation was on a specific top event, namely "Wrong classification of a road sign," which served as the basis for our analysis. We considered two important input variations, namely contrast and light intensity, as potential factors influencing the accuracy of the neural network's predictions. In order to demonstrate the impact of small deviations on prediction accuracy, we also examined the combined effect of contrast and brightness on the network's performance. By systematically varying these parameters, we were able to observe that even slight deviations in contrast and brightness levels can lead to failures in the classification of road signs.

4.4 Systematization of Knowledge—RQ3

The purpose of RQ3 was to systematize knowledge of the perception and current treatment of robustness evaluation in ANN-SCSs in operation. Hence, the focus of this research was further refined into three sub-research questions as follows:

- RQ 3.1: *What are the definitions of ANN-SCSs' robustness in operation?*
- RQ 3.2: *What metrics and methods are used to measure the robustness of ANN-SCSs?*
- RQ 3.3: *What are the challenges of measuring ANN-SCSs' robustness in operation?*

Research method. To address RQ 3.1–3.3, we conducted a review analysis (i.e., Paper P3 in Part II) that aimed to systematize and contextualize the existing knowledge on the robustness evaluation of ANN-SCSs in operation. The field of robustness evaluation is still in its early stages, making it challenging to automatically find relevant papers based on predefined search terms. Therefore, instead of using an SLR approach, we opted for a review analysis approach. This allowed us to delve deeper into the subject matter and offered more flexibility in exploring underexplored connections between different metrics and methods for the robustness evaluation of ANN-SCSs. By conducting a thorough examination of academic literature and industry standards, we aimed to gain valuable insights into the problem at hand. This comprehensive exploration enabled us to identify potential research gaps and incorporate the latest advancements in the field into practical applications. Our goal was to uncover the underexplored link between the available metrics and methods for deploying and operating ANN-SCSs, providing a deeper understanding of the robustness evaluation landscape in operation.

Research steps. We first collected relevant research papers based on the following steps:

- *Search terms.* The chosen search terms were based on their relevance to the research questions. The terms included "robust*," "classification," "deep learning" or "deep neural network," or "artificial neural network," "operation" or "industry," and "safety-critical system" or specific systems like "unmanned aircraft system (UAS)," "medical system (MS)," and "autonomous driving system (ADS)." These terms aimed to cover the metrics and methods for evaluating ANN-SCSs in operation.
- *Digital library search.* Papers were searched in digital libraries such as the ACM Digital Library, IEEE Xplore, SpringerLink, Scopus, and Web of Science, as well as Google Scholar. This comprehensive search aimed to identify relevant studies on the evaluation of ANN-SCSs in operation.
- *Standard.no search.* To specifically identify robustness definitions, the Norwegian portal of international standards (Standard.no) was searched. This portal provides free access to IEC, ISO, and Norwegian standards, which are relevant to the research.
- *Inclusion and exclusion criteria.* The inclusion criteria were defined to select papers that address robustness conceptually, propose metrics for measuring ANN-SCS robustness, perform explicit robustness evaluation, and focus on robustness in operation. Exclusion criteria included papers published before 2018, non-peer-reviewed papers, and papers not in the English language.
- *Filtering process.* The manual search initially returned 298 papers. After evaluating the title and abstract, 216 obviously irrelevant papers were excluded. The full content of the remaining ones (i.e., 82 papers) was thoroughly examined, leading to the exclusion of 69 additional papers. Snowballing techniques were then applied, resulting in the identification of 10 new related papers from the examination of 13 remaining papers.

We then use different approaches to analyze the data to answer RQ3.1 - 3.3, respectively.

- *RQ3.1 Summarizing definitions of ANN-SCS robustness in operation.* To answer RQ3.1, constant comparison [131] was adopted to identify similarities and differences in the ANN robustness definitions we found. The definitions of ANN robustness in operation were compared and summarized from three scales, i.e., system/component level, ANN classifier level, and data level. Further, we extracted key components associated with a robustness evaluation technique for the ANN-SCSs.
- *RQ3.2 Systematizing the knowledge of robustness evaluation for ANN-SCSs in operation.* To answer RQ3.2, we follow the typical workflow to assess robustness described in international standard ISO/IEC TR 24029-1 [132]. More precisely, for each selected paper, we identify their application domain, robustness goals, operational context, data source, and metrics, methods to measure robustness.

- *RQ3.3 Identifying challenges of measuring ANN-SCSs' robustness in operation.* To answer RQ3.3, we extracted metrics and methods-related challenges for each selected paper. We then use thematic analysis [133] to analyze the extracted information.

4.5 Case Study—RQ4

Building upon the exploratory study associated with RQ3, we proceeded with an empirical evaluation to delve deeper into one of the complex scenarios highlighted in Paper 3 of Part II, namely, evaluating OOD robustness using a multi-model decision-maker architecture. Our goal here was to tackle the challenge of assessing the robustness of multiple ANN models in dynamic environments, where operational data can significantly deviate from training data due to unforeseen OOD instances.

Research method. To address RQ4, we conducted a case study to empirically evaluate the effectiveness of adapting distance-based metrics to select more robust ANN models among several operational alternatives. As a research strategy, the case study method focuses on a specific case, providing a more profound and comprehensive understanding of the subject under investigation. In our work (i.e., Paper 4 in Part II), we employed a comparative case study approach, examining different distance metrics. Particular emphasis was placed on the applicability of these metrics to high-dimensional data, such as images, in the context of ranking the robustness of multiple ANN classifiers.

Research steps. The steps of the research performed to answer RQ4 were as follows:

- *Selecting the candidate distance-based metrics.* Distance-based approaches measure the distance between the distributions that generate the training and test data. Previous studies [119, 118] demonstrated the usefulness of distance-based methods for identifying issues with model performance degradation. The distance-based metrics we considered were drawn from the drift detection literature, including Wasserstein distance (WD) [53], maximum mean discrepancy (MMD) [55], Kolmogorov–Smirnov Statistic (KS) [134], Hellinger distance (HL) [135], and Kullback-Leibler (KL) divergence [54] (refer to Paper 4 in Part II for more information). Although these metrics were not explicitly designed for robustness ranking, they serve as natural starting points for this study.

- *Evaluating the effectiveness of distance-based metrics on ranking models.* To ensure a comprehensive evaluation of the robustness of multiple models, several factors need to be considered, including the selection of distance metrics, the nature and extent of input perturbations, and the sample size. The relationship between these factors can introduce complexities and dependencies that may impact the effectiveness of distance-based metrics in robustness comparison. To address these considerations, we begin by formulating two evaluation questions as follows:

- RQ4.1 (Effectiveness under OOD shifts): How well do the selected metrics rank multiple ANN classifiers when provided with various types of OOD data and their combinations?
- RQ4.2 (Sample size impact): What is the minimum sample size required for selected metrics to achieve over 0.50 precision in ranking the robustness of multiple ANN classifiers under varying levels of corruption?

RQ4.1 explores the effectiveness of the selected metrics in ranking multiple ANN classifiers using OOD test data. By considering various conditions such as corruption types, varying percentages of corrupted input, and a mixture of corruption types, we aim to provide a comprehensive evaluation of the metrics' performance in scenarios that simulate real operational settings. This is important because in practical applications, models may encounter unknown corruptions or a combination of different types of corruptions, and it is crucial to assess their robustness under such conditions. RQ4.2 targets at providing insights into the practical feasibility of using these metrics in real-world scenarios where the amount of labeled data for evaluation may be limited.

We then carefully selected datasets and models to ensure the representativeness of our evaluation. Ten state-of-the-art robust ANN classifiers against natural corruption (Models 1–10 in Table 4.1) were chosen from RobustBench [136]. RobustBench is a standardized robustness benchmark. It contains a robustness evaluation of 40+ models in image classification on natural corruptions. Here, we selected five robust models from the CIFAR10 leaderboard and five from the ImageNet leaderboard, respectively, because these models have demonstrated strong performance and robustness against a wide range of natural corruptions in the RobustBench benchmark. By choosing models from the leaderboard, we ensure that we are evaluating state-of-the-art models that have undergone rigorous testing and evaluation, making them reliable candidates for our study.

Table 4.1: Datasets and models used in our experiments.

No.	Dataset	Model ID	Source	Clean Accuracy
1	CIFAR10-C,Corruptions	Diffenderfer2021Winning_LRR_CARD_Deck	[30]	0.97
2		Diffenderfer2021Winning_LRR	[30]	0.97
3		Diffenderfer2021Winning_Binary_CARD_Deck	[30]	0.95
4		Hendrycks2020AugMix_ResNeXt	[27]	0.96
5		Hendrycks2020AugMix_WRN	[27]	0.95
6	ImageNet-3DCC,Corruptions	Tian2022Deeper_DeiT-B	[137]	0.81
7		Tian2022Deeper_DeiT-S	[137]	0.80
8		Erichson2022NoisyMix_new	[138]	0.77
9		Hendrycks2020Many	[139]	0.77
10		Erichson2022NoisyMix	[138]	0.77

Model robustness is sensitive to input variations [107]. The choice of corruption datasets to use in our study was made to simulate OOD scenarios in operation. To thoroughly evaluate the consistency of selected distance-based metrics for robustness ranking, we consider a variety of natural corruptions and their mixtures. We utilize the CIFAR10-C dataset [107], which consists of 15 corruption types. These corruptions include Gaussian noise, motion blur, brightness variations, etc. Additionally, we employ the ImageNet dataset with 3D Common Corruptions (ImageNet-3DCC) [140], which introduces corruptions that align with real-world scenarios, such as camera motion, weather conditions, occlusions, depth of field, and lighting. Besides, each type of corruption in CIFAR10-C and ImageNet-3DCC has five levels of severity.

In evaluating the ranking results, we used robust accuracy as the reference ranking, which is measured based on the correct labels. Robust accuracy is a well-established measure in the ML literature for assessing the performance of ANN models under various corruptions. To compare the rankings produced by the distance metrics with the ground truth, we employed the average precision at k (AP@k) metric [141]. This metric, commonly used in recommendation systems and ranking-related problems, evaluates the relevance of recommended items and their positioning in the ranking. In our study, we selected k=1 to focus on selecting the best model, considering the context of a multi-model decision-maker.

5 Results

This chapter presents the main findings of the thesis, which are organized according to the research questions and corresponding research steps. The findings encompass four aspects: a systematic review of T&V methods for ANN-SCSs, an extended fault tree analysis for analyzing the influence of ANN robustness on system safety, a systematization of knowledge and a framework for assessing ANN-SCSs' robustness in operation, and an empirical assessment of metrics for ranking the robustness of ANN models under OOD shifts.

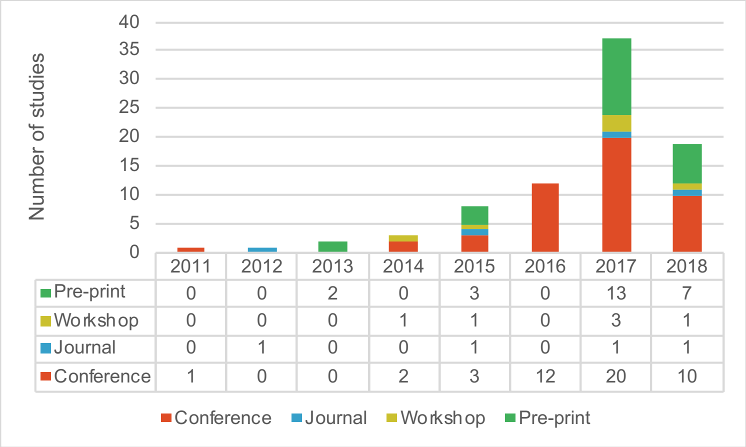
5.1 Testing and Verification (T&V) of ANN Classifiers in SCSs - RQ1

We conducted an SLR to uncover techniques made so far in the field of T&V of ANN-based SCSs, as well as to identify the research gaps. The high-level summaries of the results of RQ1 are as follows, and Section 4 of Paper P1 in Part II provides a more detailed explanation of the findings.

5.1.1 Results of RQ1.1: Profiles of the Studies Focusing on Testing and Verifying ANN-SCSs

Study distribution. We covered papers from January 2011 to November 2018. Figure 5.1 shows the distribution of selected papers based on the publication year and type of work. There have been 68 papers (81.9%) published between 2016 and 2018, indicating that researchers are paying more attention to the T&V of ANN-based SCSs. Conference was the most popular publication type with 48 papers (57.8%), followed by pre-print (25 papers, 30.1%), workshop (6 papers, 7.2%), and journal (4 papers, 4.8%).

We also examined the geographic distribution of the reviewed studies to identify the leading countries in research related to T&V of ANN-SCSs. The analysis revealed that researchers from the USA had contributed the most primary studies, with 56 publications, followed by researchers from Germany and the UK, with ten and nine publications, respectively. It is noteworthy that 47 out of the 83 publications (56.6%) involved collaboration with industry partners.



	2011	2012	2013	2014	2015	2016	2017	2018
🟩 Pre-print	0	0	2	0	3	0	13	7
🟨 Workshop	0	0	0	1	1	0	3	1
🟦 Journal	0	1	0	0	1	0	1	1
🟥 Conference	1	0	0	2	3	12	20	10

Figure 5.1: Publication year and types of selected papers.

Research types. We categorized the selected papers into six research types, namely evaluation research, solution proposal, validation research, philosophical papers, opinion papers, and experience papers, based on the criteria proposed by Kai et al. [123]. The majority of the selected papers fell into the categories of evaluation research (31.3%, 26 papers) and validation research (61.4%, 51 papers). It is not surprising that the percentage of solution proposal papers was relatively low (6 papers) because most of the reviewed papers focused on presenting and demonstrating their T&V approaches through academic and industrial case studies, simulations, and controlled experiments. The other three types of research papers (i.e., philosophical papers, opinion papers, and experience papers) were not found in the selected studies, as our inclusion criteria specifically targeted papers that addressed testing/verification approaches.

Application domains. To provide valuable insights into the domain-specific aspects of the approaches, we conducted an analysis of the application domains covered in the selected studies. Our findings indicate that a significant amount of research focuses on utilizing ANN algorithms for general-purpose control logic (59 papers, 71.1%). Additionally, considerable attention is given to the application of ANN algorithms in automotive CPSs, particularly AVs (13 papers, 15.7%). Furthermore, there are also several studies that explore the use of ANN algorithms in autonomous aerial systems, specifically airborne collision avoidance systems for unmanned aircraft (5 papers, 6%).

5.1.2 Results of RQ1.2: Classification of T&V Approaches

The initial literature review on T&V approaches and associated tools for ANN-based SCSs resulted in 79 research papers, which we classified into five high-order themes based on the research goals:

Table 5.1: A classification of approaches to test and verify ANN-based SCSs

Themes	**Subthemes**	**Papers**	**#**
Assuring robustness of ANNs	Understanding the characteristics and impacts of adversarial examples	[12],[13],[14],[142],[15], [16], [17]	17
	Detecting adversarial examples	[18],[143], [19], [20], [21], [22]	
	Mitigating impact of adversarial examples	[144], [145]	
	Improving robustness of ANNs through using adversarial examples	[25], [26]	
Improving failure resilience of ANNs		[146],[147],[148],[149],[150],[151],[152],[153],[154], [155],[156]	11
Measuring and ensuring test completeness		[157],[158],[159],[160],[161],[162],[163]	7
Assuring safety properties of ANN-based CPSs		[164],[46],[165],[166],[167],[168],[169],[98],[170], [171],[172],[173],[174]	13
Improving interpretability of ANNs	Understanding how a specific decision is made	[175],[176],[177],[178],[179],[180],[181],[182], [183],[184],[185],[186],[187],[188],[189], [190], [191],[192],[193]	31
	Facilitating understanding of the internal logic of ANNs	[194],[195],[196],[197],[198],[199]	
	Visualizing internal layers of ANNs to help identify errors in ANNs	[47],[48],[200],[201],[202],[203]	

- **CA1: Assuring the robustness of ANNs**, i.e., an ANN can cope with erroneous inputs, where the erroneous inputs can be an adversarial example (i.e., an input that adds a small perturbation intentionally to mislead an ANN's classification), or benign but misleading input data.
- **CA2: Improving the failure resilience of ANNs**, so that the ANN-SCSs are more tolerant of possible hardware and software failures.
- **CA3: Measuring and ensuring test completeness** to ensure good coverage when testing ANNs.
- **CA4: Assuring the safety property of ANN-SCSs** by providing formal verification or mathematical proof that a system satisfies some desired safety properties (e.g., the system should always stay within some allowed region, namely a safe region).
- **CA5: Improving the interpretability of ANNs** to facilitate a better understanding of how ANNs generate outputs from inputs.

In Table 5.1, we summarize the reviewed papers according to the themes and subthemes. Section 4.2 of Paper P2 in Part II provides a more detailed analysis of the identified methods and tools for T&V of ANNs.

5.1.3 Results of RQ1.3: Challenges for Testing and Verifying ANN - SCSs

In order to assess the extent to which state-of-the-art methods for testing and verifying ANN-SCSs fulfill the desired safety integrity properties, we mapped the identified challenges onto relevant properties and major T&V phases in the software safety lifecycles of IEC 61508-3. Table 5.2 presents the grouping of challenges and the corresponding safety integrity properties. Among these properties, correctness, completeness, freedom from intrinsic faults, and fault tolerance have received significant attention from the research community. However, achieving repeatability and addressing common cause failure have been relatively overlooked. Notably, no reviewed study specifically focused on precisely defined testing configurations and defense against common cause failure, which are crucial aspects for ensuring the safety of production-ready ANN-SCSs [36]. For a more detailed analysis of the identified limitations and corresponding suggestions based on the required safety integrity properties, please refer to Section 4.3 of Paper P1 in Part II.

Table 5.2: A detailed mapping of reviewed approaches to the IEC 61508 safety lifecycle

Phase	**Property**	**Relevant primary studies**	**Category**	**Remaining challenges**
Software architecture design	Completeness	None		N/A
	Correctness	[168]	CA4	Training process of ANN-based algorithm is time-consuming.
	Freedom from intrinsic faults	[13, 142, 15, 17, 20], [22] - [26]	CA1	❶ Limited to specific model classes or tasks (e.g., image classifier), or small size ANNs [142]; ❷ Not immune to adversarial adaptation [20]; ❸ Lack of understanding of how the system can be free from different kinds of attacks other than adversarial examples.
	Understandability	[175] - [203]	CA5	❶ Limited to specific model classes or tasks (e.g., image classifier), or small size ANN models [194]; ❷ Not able to provide real-time explanations; ❸ Lack of evaluation method for the explanation of ANNs.
	Verifiable and testable design	[157]	CA3	❶ Lack of integrated computer-aided toolchains to support verification activities; ❷ Limited to specific models, tasks, or ANN sizes.
		[46]	CA4	❶ Limited to specific ANN architectures (i.e., piece-wise linear activation functions), need a better understanding of ANN architectures; ❷ Trade-off between efficient verification and linear approximation of the ANN behavior is not studied sufficiently.

Table 5.2 — continued from the previous page

Phase	Property	Relevant primary studies	Category	Remaining challenges
	Fault tolerance	[147, 148, 152, 155, 156]	CA2	❶ Decouple the fault tolerance from the classification performance [148]; ❷ Lack of studies on unexpected environmental failures.
	Defense against common cause failure	None		N/A
Software module testing and integration	Completeness	[16, 26]	CA1	Lack of comprehensive criteria to evaluate testing adequacy.
		[158] - [163]	CA3	Low fidelity of testing cases compared with real-world cases [159].
	Correctness	[12, 14, 16, 18, 143] [19, 21]	CA1	❶ Vulnerable to the variation of adversarial examples; ❷ Limited to specific ANN model classes or tasks.
		[151]	CA2	Insufficient validation of input raw data.
	Repeatability	[157, 158, 159]	CA3	Testing cases generated by automated tools may be biased.
	Precisely defined testing configuration	None		N/A
Programmable electronics integration (hardware and software)	Completeness	None		N/A
	Correctness	[146, 149, 150, 153]	CA2	Insufficient testing of hardware accelerator.

Table 5.2 — continued from the previous page

Phase	**Property**	**Relevant primary studies**	**Category**	**Remaining challenges**
	Repeatability	None		N/A
	Precisely defined testing configuration	None		N/A
Software verification	Completeness	[167, 169]	CA4	❶ Limited to specific ANN models; ❷ Lack of scalability.
	Correctness	[154]	CA2	❶ Automatic generation of complete testing scenario sets.
		[164, 165, 166, 204, 170] [171] - [173]	CA4	❶ Scalability and computational performance need to improve; ❷ SMT encoding for large-scale ANN model; ❸ Lack of model-agnostic verification methods; ❹ Automatic generation of feature space abstractions [173].
	Repeatability	None		N/A
	Precisely defined testing configuration	None		N/A

5.2 The Influence of ANN Robustness on the Safety of Autonomous Vehicles (AVs) - RQ2

The findings of RQ2 contribute to the preservation of risk assessment as a valuable tool for safety engineering in the context of developing safety-critical applications, with AVs serving as a representative example. For a more comprehensive explanation of these findings, please refer to Paper P2 in Part II.

5.2.1 Safety and Security Threats in ANN-Based Perception for AVs

ANNs have exhibited outstanding performance in AV's perception applications. However, ANNs inherently exhibit vulnerability to perturbations such as instances outside their training sets, scene noise, instrument noise, image translation or rotation, or minor changes deliberately added to the original image, referred to as adversarial perturbations. Incorrect deductions from perception systems, including missing objects, incorrect classification, and traffic sign misdetection or misreading, have been identified as significant causes of disengagement incidents in AVs.

In this context, we categorize the failure modes of these perturbations in ANN-based perception methods into two major groups: safety threats and security threats. Safety threats cover a broad range of circumstances that may impact an ANN's performance in AV control. These include environmental factors, obscurations, training deficiencies, and inherent limitations of the ANN. Security threats, on the other hand, emerge in adversarial contexts, where the ANN may be exposed to intentional manipulations designed to exploit its vulnerabilities.

Safety threats. We identified a wide range of situations that can affect the performance of an ANN for AV control, ranging from environmental conditions to training deficiencies.

- Fundamental functional omissions (such as lack of training to recognize road diversion signs or failure to recognize vehicles crossing a roadway due to lack of trajectory prediction)
- Sensitivity to ambient conditions, especially low lighting
- Sensitivity to low-contrast conditions
- Sensitivity to misleading patterns (such as camouflage) or to textures
- Obscuration: intended objects hidden behind others, behind a blind curve, or behind vegetation
- Obscuration by snow, blown sand, frost, or ice
- Reduction in visibility due to fog, snow, or sandstorm
- Inadequate training set
- Poor separability of different object types due to similar feature sets
- Interference with well-trained recognition by extensions to the training set
- Orientation of the objects to be recognized ("pose")
- Unusual elevation of objects to be recognized (such as lane markings on a transition to a steep hill)
- Road reflectance, such as lights reflected from wet roads
- Strong backlight (e.g., driving into a sunset)
- Mirage effects in reflections from roadways
- Shadows

Further threats were identified from the validation studies, that is, by reviewing accident reports:

- Unstable object recognition (with consequent erroneous or absent of emergency response)
- Obscuration by leading vehicle
- Vehicles crossing the roadway (vehicle not in the detection field or failure to measure velocity)

Security threats. In an adversarial context, threats to the ANN could arise from:

- Training data poisoning. Training data poisoning refers to deliberately introducing false data during the training process.
- ANN model attack. An ANN model attack takes advantage of the model's flaws to fool the system.
- Adversarial example. An adversarial example is small changes intentionally added to the original input that are invisible to human eyes. There is a long history of work on understanding, detecting, mitigating the impact, and increasing the robustness of ANNs by using adversarial examples [2].
- Physical adversarial attack. A physical adversarial attack aims to fool ANN models by creating perturbations on physical objects.
- Sensor sabotage. Sensor sabotage can be conducted by using spotlights to blind cameras or laser-targeting of cameras.

In this study, we focus on the practical consequences of adversarial examples on the design of AV perception models. Evaluating the security threats to ANNs is a safety consideration, and adversarial examples can further be used to improve ANN robustness.

5.2.2 The Extended fault tree analysis (FTA), Including ANN Components

In order to assess the robustness and performance limits of the ANN components in the context of AV control, we proposed an extension to the traditional FTA methodology. The core of this extended FTA methodology is to determine the conditions under which the operating environment could potentially challenge the performance limits of the ANNs. Additionally, we aimed to identify the extent to which additional robustness enhancements and other safety measures could compensate for any deficiencies in the ANNs' performance. We developed a generic template that integrates ANN failure into the fault tree analysis, as depicted in Figure 5.2. This template serves as a structured framework for systematically

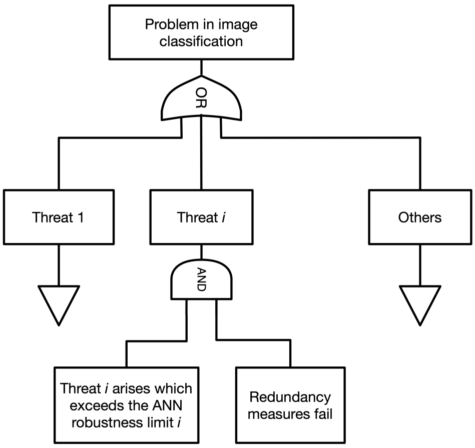

Problem in image classification
OR
Threat 1
Threat *i*
Others
AND
Threat *i* arises which exceeds the ANN robustness limit *i*
Redundancy measures fail

Figure 5.2: General template for an ANN failure subtree in an FTA (for independent threats).

analyzing the potential failure modes and their corresponding causes within the ANN components of the AV control system.

Functional failures of the ANN can then be integrated into FTs as multiple subtrees related by an OR relationship. The probability of the ANN failing in any given subtree is then expressed as:

$$\mathrm{P}_{\text{function failure } i} = \mathrm{P}_{\text{robustness limit } i \text{ exceeded}} \times \mathrm{P}_{\text{redundancy measures fail}} \tag{5.1}$$

In this equation, the events for mechanical and electrical components in the fault tree represent functional failures identified by failure analysis. Their probabilities could be determined through testing or field observations. The functional failure probabilities for the ANNs could be established by conducting robustness tests and determining the likelihood of threats exceeding the robustness threshold. For example, to understand the threat to performance posed by low illumination levels, we can drive along selected routes at different times and in various weather conditions and record the illumination brightness levels.

5.2.3 Evaluation of the Methodology

To evaluate the extended FTA methodology, we applied it to a traffic sign recognition task both theoretically and practically.

Problem formalism. We considered a possible hazardous event triggered by a decision made by the ANN, namely, “Wrong classification of a road sign.” Robustness can be measured as above by the prediction accuracy given perturbed inputs as a function of, for example, the lighting level or the contrast in the image. Assume that two variables influence whether the sign is classified correctly. One is contrast intensity,

C, and the other is light intensity (i.e., brightness), L. Suppose T_C stands for the lower limit for C, below which the sign cannot be classified correctly. In that case, we can define the event $E_C = \{E_C : c < T_C\}$ that is "too low contrast to recognize correctly." Similarly, $E_L = \{E_L : l < T_L\}$ is the event "too low lighting to recognize correctly". The third misclassification event is defined by the following condition: $E_{LC} = \{E_{LC} : (l, c) < f(c, l), c > T_C, l > T_L\}$. This should be understood as follows: "while contrast and lighting both lie in the correct classification region, their combination may belong to the misclassification region." The border dividing the two regions is determined by the function $f(c, l)$. Usually, this type of event occurs when variables (parameters) lie in the vicinity of the border points. That is to say, the effect of small deviations results in failure. A possible region of misclassification is shown in Figure 5.3. In the case where there are two variables that influence whether the sign is classified correctly, the performance can be represented as in Figure 5.3.

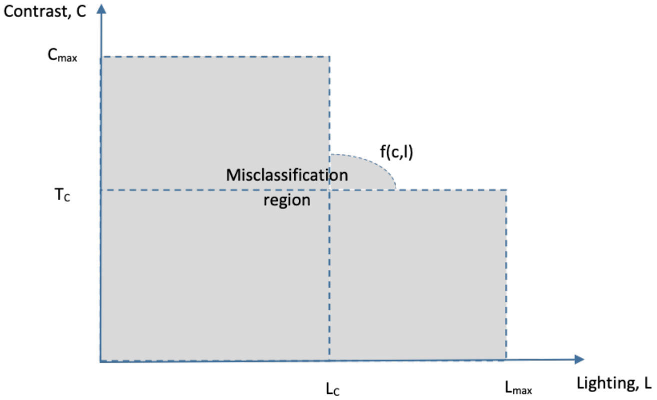


Figure 5.3: Misclassification region(conceptual).

The region of misclassification can formally be written as follows:
$\Omega = \{(c < \mathrm{T_C}) \bigcup (l < \mathrm{T_L}) \bigcup ((l, c) < f(c, l), c > \mathrm{T_C}, l > \mathrm{T_L})\}$

As soon as the misclassification events are determined, a simple fault sub-tree can be constructed in Figure 5.4.

Given that C and L are independent random variables and their probability density functions are known, $f_C(x)$ and $f_L(y)$, the probability of misclassification $P_{\text{misclassification}}$ can be calculated:

$$\mathrm{P_{misclassification}} = \iint_{\Omega} f_C(x) f_L(y) dx dy \tag{5.2}$$

An AV example of misclassification. To demonstrate the influence of the perturbations and their combination, we trained a five-layer CNN with the German Traffic

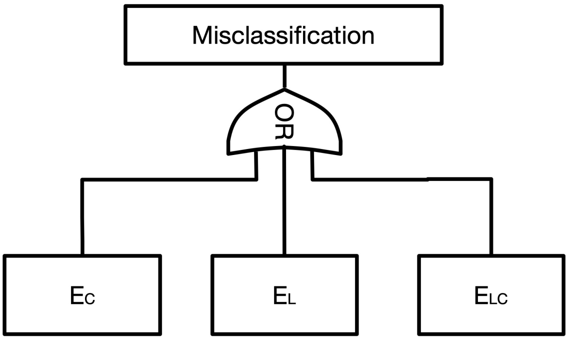
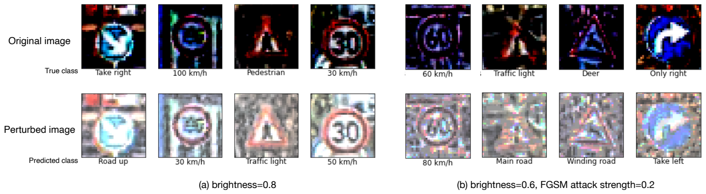


Figure 5.4: A simple fault sub-tree for misclassification (with interacting threats).

Sign Recognition Benchmark (GTSRB) dataset for the traffic sign classification [205]. The GTSRB dataset has 43 different traffic signs in various sizes and lighting conditions and is very similar to real-life data. The prediction accuracy for clean test images is 98.97%.

We adopt the algorithm from [206] to emulate the deviation of brightness and contrast and the algorithm from [207] to implement the fast gradient sign method (FGSM) attack. Figure 5.5 presents a) a set of misclassified images with brightness=0.8, in which case the prediction accuracy dropped to 84.8%; b) brightness=0.6, FGSM attack with attack strength=0.2, in which case the prediction accuracy dropped to 18.76%.



(a) brightness=0.8

(b) brightness=0.6, FGSM attack strength=0.2

Figure 5.5: Examples of misclassified traffic signs.

Then we test the combination of brightness and contrast reduction. In this experiment, we set the prediction accuracy at 90% as the benchmark for model robustness. We first examined the impact of increased brightness and contrast reduction, given the naturally low brightness/contrast characteristics of the GTSRB dataset. Figure 5.6 shows prediction accuracy curves corresponding to a) brightness variations and b) contrast variations. It shows that the upper limit for brightness increase is 0.66 in Figure 5.6 a), and the upper limit for contrast reduction is 0.54 in Figure5.6 b).

Subsequently, we tested the combined effect of brightness and contrast reduction. The aim of this experiment was to explore how the small deviation of contrast and brightness affects prediction accuracy. The brightness level is set from 0.01 to 1, and the contrast reduction is from 0.01 to 1. In Figure 5.7, the values of prediction accuracy are represented as colors. The lighter the color, the higher the prediction

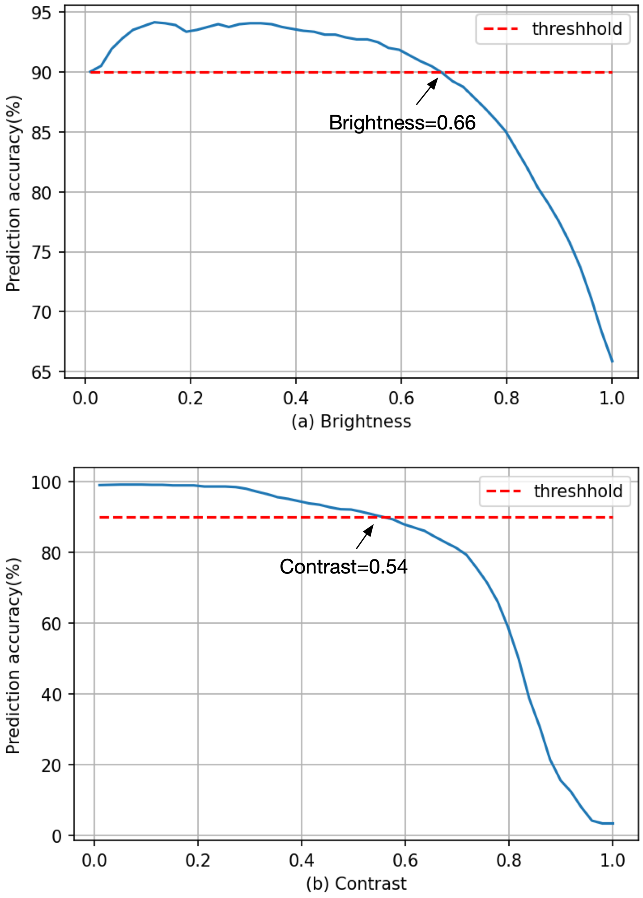
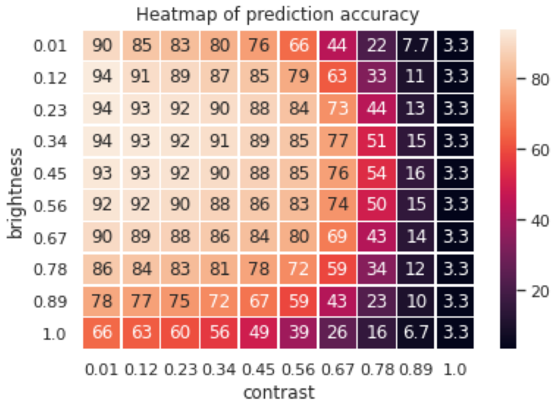


Figure 5.6: Examples of prediction accuracy curves when brightness and contrast vary.

accuracy. It shows that even brightness level and contrast reduction do not exceed their upper limits (i.e., in the correct classification region). Their combination can fall into the misclassification region (i.e., prediction accuracy is lower than 90%). A detailed analysis is provided in Section 6.2 in Paper P2 in Part II.

The experimental results highlighted that even slight deviations in pairs of threats can, when they occur together, worsen the impact on the ANN's performance more than the effect of a single deviation. Notably, brightness and contrast represent merely two of the myriad challenges to ANN performance necessitating the use of a hybrid fault tree approach. Indeed, almost all of the threats identified in Section 5.2.1 present variable intensities. In several cases, such as the complex scenario of combined obscurations and shadows, these threats interact in a way that amplifies their impact on the ANN. Some threats, like adversarial examples, are particularly



brightness \ contrast	0.01	0.12	0.23	0.34	0.45	0.56	0.67	0.78	0.89	1.0
0.01	90	85	83	80	76	66	44	22	7.7	3.3
0.12	94	91	89	87	85	79	63	33	11	3.3
0.23	94	93	92	90	88	84	73	44	13	3.3
0.34	94	93	92	91	89	85	77	51	15	3.3
0.45	93	93	92	90	88	85	76	54	16	3.3
0.56	92	92	90	88	86	83	74	50	15	3.3
0.67	90	89	88	86	84	80	69	43	14	3.3
0.78	86	84	83	81	78	72	59	34	12	3.3
0.89	78	77	75	72	67	59	43	23	10	3.3
1.0	66	63	60	56	49	39	26	16	6.7	3.3

Figure 5.7: Prediction accuracy matrix with small deviation of brightness and contrast in combination.

challenging because they are difficult for humans to perceive. To shed light on the impact of such threats on ANN performance, we can utilize methods from the field of explainable AI (XAI) [2].

Procedure of extended FTA for ANNs This study demonstrated that we could carry out comprehensive hazard identification for autonomous vehicles, which includes both hardware and ANN components. Building on the previous evaluation, we have outlined the procedure for the extended FTA as follows:

- Complete the overall high-level hazard identification using an FTA approach;
- Identify the functional failures of ANNs' which contribute to the overall FTA;
- Identify the challenges which can cause the ANN functional failure, e.g., using the checklist in Section 5.2.1;
- Determine the robustness of the ANNs when challenged by both perturbations of single parameters and a combination of parameter perturbations via testing ANN performance;
- Determine the probability of the occurrence of parameter perturbations;
- Incorporate the contribution of ANNs to the FTA using the templates given in Figure 5.2. and Figure 5.4.

5.3 Robustness Evaluation for SCSs Utilizing ANN Classifiers in Operation - RQ3

Building upon the pivotal insights from previous findings, the importance of measuring and enhancing the robustness of ANNs in the design and operation of SCSs utilizing ANN classifiers, like AVs, becomes paramount. The widespread integration of ANN classifiers in various safety-critical sectors, such as AVs, aircraft control systems, smart grids, and healthcare services [2], presents significant challenges that demand immediate attention. Accordingly, we focused on evaluating the robustness of ANN-SCSs in operation, considering the potential for compromised model performance when input data deviates from training data. Our study systematically outlines the robustness evaluation of ANN-SCSs at the system, ANN model, and input levels. We classified evaluation methods and metrics and identified opportunities and gaps for future research. More detailed insights from this study are presented in Paper P3 in Part II.

5.3.1 Results of RQ3.1: Definitions of ANN-SCS Robustness in Operation

Despite the popularity of the term “robustness” in the literature, a limited portion of papers addresses this system attribute from a conceptual point of view. We identified nine definitions from scientific papers and industry standards. Table 5.3 summarizes the identified robustness definitions at different granularity levels, i.e., the system, ANN model, and input data levels.

- *Robustness definitions at the system level.* The identified system-level definitions are generally concerned with the system’s ability to maintain its performance and function correctly when facing exceptional or unforeseen conditions. These conditions can include unavailability of resources, communication failures, environmental disturbances [208], invalid inputs [208], and changes in the system’s operating conditions [132].

- *Robustness definition at the ANN model level.* All of the identified definitions at the ANN model level refer to the model’s ability to maintain its performance when faced with inputs or conditions that differ from what it was trained on. The most commonly studied input deviations include malicious perturbations (i.e., an input that adds a small, intentional perturbation to mislead the classification of an ANN) [207] and natural perturbations [107]. Malicious perturbations usually have the purpose of making the perturbation invisible, while natural perturbations have no such constraint. Natural perturbations are noises that exist in natural environments. They may be more noisy and visible than malicious perturbations. In addition to adversarial robustness and robustness to natural perturbations, a specific concern in operation is the impact of the mismatch between the training data distribution and the operational distribution (referred to as distributional shift [74] on the model’s performance).

- *Robustness definition at the data level.* This level of robustness definition generally states that a robust model should be able to correctly classify inputs that are similar to the inputs it was trained on. In this case, the “similar inputs” are defined as the neighbors of the original data point.

Table 5.3: Definitions of robustness in literature.

Level	Ref.	Definition of robustness
System	[208]	[Robustness] is the degree to which a system or component can function correctly with invalid inputs or in stressful environmental conditions.
	[209]	[Robustness] is the ability of a system to maintain its level of performance under a variety of circumstances.
	[132]	[Robustness] is the ability of an AI system to maintain its performance level under any circumstances (domain change, hardware failure, etc.).
	[127]	[Robustness] provides safe behavior at boundaries (corner case, core event, extreme case).
ANN model	[207]	[Robustness] is the classifier's worst-case performance on small, additive, classifier-tailored perturbations.
	[175, 210]	An ANN classifier is robust if it achieves correct classification on a testing sample that is "close" to a training sample.
	[107]	Robustness is the classifier's average-case performance on small, general, classifier-agnostic corruptions or perturbations.
	[74, 211, 212]	An ANN classifier is robust if it achieves "consistent" classification (i.e., prediction accuracy) on known and unknown inputs as long as the unknowns are not too different from the known inputs.
Data	[213]	An original data point is strong (robust) concerning the ANN model under test if its neighbor accuracy is higher than a predefined threshold.

Based upon the existing definitions identified in the literature, we found that most of the existing robustness definitions and corresponding evaluations include several factors, i.e., the scale of the system architecture, the operational context, and the nature of the data (covering both input and output). Often, these factors are determined by the application domain in which the robustness needs to be evaluated. Figure 5.8 shows the key components associated with a robustness evaluation technique for ANN-SCSs.

5.3.2 Results of RQ3.2: Methods and Metrics to Measure ANN-SCS Robustness in Operation

Proposed framework to organize the categories and illustrate the knowledge. Based on the different levels of robustness definitions and the five elements of robustness evaluation, which are explained in Section 5.3.1, we proposed a framework that adopts a hierarchical conceptual approach to categorize and illustrate existing methods and metrics in evaluating the robustness of ANN-SCSs (see Figure 5.9).

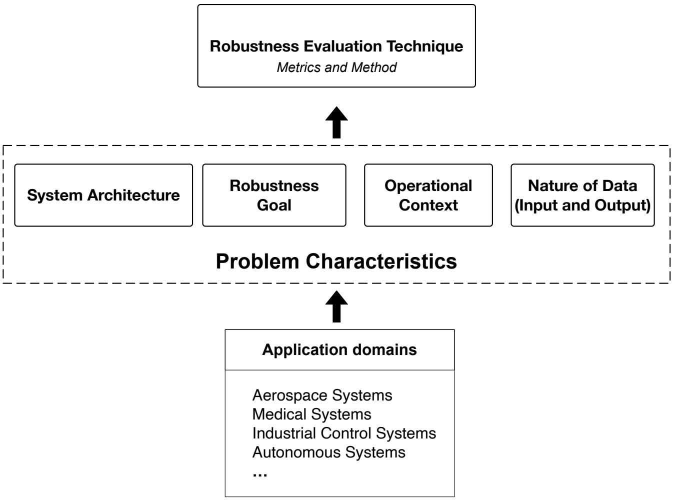


Figure 5.8: Key components associated with a robustness evaluation technique for ANN-SCSs.

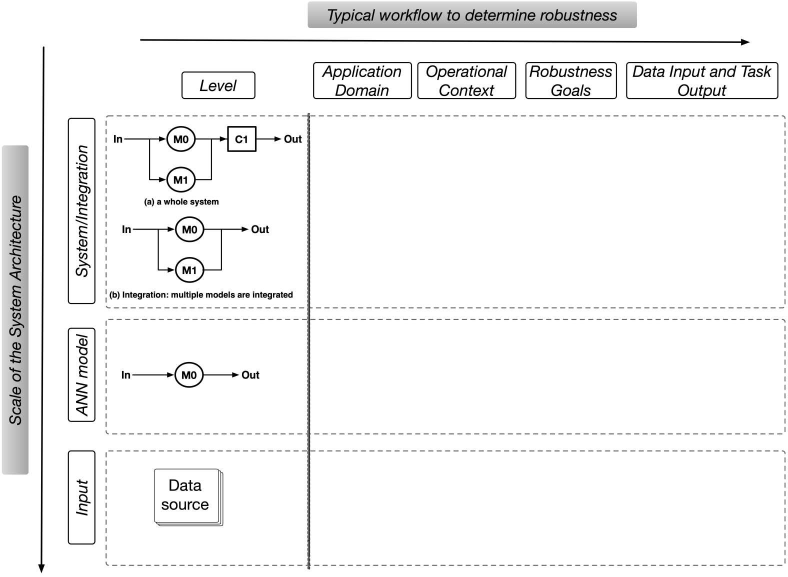


Figure 5.9: Template for the proposed multidimensional framework.

- *Scale of the System Architecture.* The system architecture scale refers to the level at which the ANN-SCS is, which includes: 1) system level, where the ANN-SCS is evaluated as a whole within its operational environment; 2) ANN model level, which involves evaluating a single ANN model independently; and 3) input level, where the input data utilized in operation are evaluated.
- *Application Domains and Context.* Recognizing the application domain and context is crucial for selecting suitable metrics and methods to assess a model or system's robustness. SCS application domains analyzed in this study include ADS, MS, and UAS.
- *Robustness Goals.* These include performance requirements such as maintaining consistent performance when dealing with altered inputs, generalizing effectively within and across domains, and resisting adversarial attacks [43].
- *Data Input and Task Output.* This study specifically focuses on using ANN for classification tasks, with image data as the primary input and class prediction as the primary output, and thus limits its discussion to data and task outputs related to this particular focus.

Categories of methods and metrics to measure ANN-SCS robustness in Operation. We analyzed five system-level, 15 ANN model-level, and eight input-level studies that focus on evaluating the robustness of ANN-SCS in operation. In Tables 5.4, 5.5, and 5.6, we summarize the identified methods and metrics in the corresponding three levels. The results show that classification accuracy is the primary metric used for robustness evaluation at all three levels. Additionally, sensitivity-based evaluation methods are popular across these levels. For system-level assessments, simulation-based evaluation is commonly employed. In contrast, input-level assessments use coverage-based metrics to evaluate the effectiveness of various scenarios and conditions in the dataset. Utilizing a combination of complementary methods and metrics can help that the robustness of the system is thoroughly analyzed and potential vulnerabilities are identified under various conditions and scenarios. Section 5.2 of Paper P3 in Part II provides a complete analysis of the identified methods and metrics.

5.3.3 Results of RQ3.3: Challenges of Measuring ANN-SCS Robustness in Operation

There are many metrics to evaluate robustness in operation. However, the focus should not be on the number of metrics but rather on effectively integrating or selecting the appropriate metrics to capture various aspects of robustness and address genuine concerns in real-world application scenarios. Building upon the results of the state-of-the-art knowledge, we unfold challenges related to the application domain, robustness goal, and methods/metrics at each level.

- *Challenges of robustness evaluation at the system level.* Research on the robustness evaluation of ANN-SCSs in operation is still in its early stages.

Table 5.4: Techniques for evaluating the robustness of ANN-SCSs in operation (system level).

SCS	Operational	Robustness	Measurement Method	Robustness Metrics	Ref.
Domain	Context	Goal			
ADS	End-to-end steering	Min. MSE of steer angle in the presence of adversarial examples and synthetic noisy input	Input-output evaluation	Likelihood-based surprise adequacy, Distance-based surprise adequacy	[214]
				Attacking strength, Average angle error, Percentage of frames whose angle error exceeds a predefined threshold	[215]
	Object perception	Ensuring safe driving in rare failure scenarios	Simulation-based Fault injection	Minimum time to collision, Failure probability	[216]
MS	Diagnose	Accurate and reliable diagnosis	Field testing	False positive rate False negative rate	[217]
UAS	VLG	Reliable landing	Fault tree analysis	Failure probability	[218]

ADS: Autonomous Driving System; MS: Medical System; UAS: Unmanned Aircraft System; VLG: Visual Landing Guidance

Table 5.5: Techniques for evaluating the robustness of ANN-SCSs in operation (ANN model level).

SCS Domain	Operational Context	Robustness Goal	Method	Metric	Ref.
		(i.e., be robust against)			
Image classification	Generic	Pixel perturbations	Sensitivity-based	Level-threshold-safe, Level-pixel-safe	[219]
		Spatial deformations	Sensitivity-based	Attack success rate	[220]
		Natural corruptions	Simulation-based	Threat severity, Minimal perturbations, fooling success rate	[206]
			Sensitivity-based	Accuracy loss	[221]
		Hardware and software faults	Simulation-based fault injection	Bit error rate-accuracy curves	[222]
				SDC rate	[223]
ADS	Traffic sign recognition	Natural corruptions	Sensitivity-based	Classification accuracy	[224]
	Object detection	Natural corruptions	Sensitivity-based	AP and AP@50; mAP	[225]
MS	Gastroenterology	Distributional shift	Sensitivity-based	Sensitivity, specificity	[226]
UAS	VLG	Distributional shift	Sensitivity-based	Classification accuracy	[218]

ADS: Autonomous Driving System; MS: Medical System; UAS: Unmanned Aircraft System; VLG: Visual Landing Guidance

We distinguish two types of system architecture variations based on different

Table 5.6: Techniques for evaluating the robustness of ANN-SCSs in operation (input level).

SCS	**Operational**	**Robustness Goal**	**Method**	**Metric**	**Ref.**
Domain	**Context**	**(i.e., be robust against)**			
Generic	Generic	Semantic diversity	Coverage-based	Importance-driven coverage (IDC)	[227]
Generic	Generic	Natural corruptions	Sensitivity-based	Neighbor accuracy, Neighbor diversity score	[213]
Generic	Generic	Distributional shift	Sensitivity-based	F-measure for threshold values	[228]
			Adversarial filtration	Top1 accuracy	[139, 229]
Generic	Generic	Triggering misclassifications	Test input selection and prioritization	Maximum mean discrepancy-critic	[230]
				Model uncertainty-based	[111]
				Sample discrimination-based	[112]

evaluation goals: 1) an ANN-SCS as a black box, with the aim of assessing potential performance degradation due to input changes, and 2) an ANN-SCS with redundant ANN models, with the objective of comparing the performance of multiple models and recommending the optimal one for use.

- *Challenges of robustness evaluation at ANN model level.* The reviewed papers evaluate ANN models focusing on adversarial perturbations, which are mainly generated based on the l_p norm distance (i.e., measures the distance between an instance and its neighboring points in the input space, p value can be 0, 1, 2, and ∞.), realistic environment lighting and geometry, spatial transformations, and distributional shifts, respectively. However, the goal of robustness in practice is more comprehensive, as highlighted by Hendrycks and Dietterich [231]. For example, to assess adversarial robustness, it is crucial to consider perceptible attacks, as attackers may not only construct small l_p perturbations to deceive the system. They may also rotate the adversarially modified images or apply other novel distortions to the image [232]. The areas of adversarial robustness and corruption robustness, distributional shift, and unusual events should be considered in a unified manner. In addition, ANN model-level robustness evaluation still lacks definitions of acceptable levels of performance.

- *Challenges of robustness evaluation at input level.* Mincu et al. [233] addressed the challenges of obtaining high-quality healthcare datasets due to privacy-preserving considerations. They suggested techniques such as federated learning to encourage reproducibility while retaining data privacy. Liu et al. [234] pointed out that it can be difficult to define which cases are in-distribution and which are out-of-distribution given the complexity of most medical data. We envision that the use of fine-grained, actionable taxonomies of perturbations, collaborative documentations of domain-specific perturbations, libraries to generate such perturbations semi-automatically, and frameworks and metrics to uncover new types of perturbations in the wild are necessary studies to be performed in the future.

In sum, we have identified three types of challenges, namely, identifying comprehensive abnormal conditions, standardizing the definition of an acceptable level of performance, and acquiring sufficient labeled data, at the system, ANN model, and input levels. Furthermore, we have highlighted an emerging need to assess ANN-SCSs using redundant models, which has been overlooked. Section 5.3 of Paper P3 in Part II provides a complete analysis of the identified challenges.

5.4 Dynamic Robustness Evaluation for Automated Model Selection - RQ4

Building on one of the challenges identified from RQ3, our research focused on exploring the applicability of existing distance metrics for evaluating and ranking model robustness in real-world operational scenarios. Through this investigation, we contribute new knowledge and understanding of dynamic robustness evaluation during operation. P4 in Part II provides a comprehensive and detailed explanation of the empirical findings.

5.4.1 The Selected Distance-based Metrics

As explained in Section 4.5, we have chosen five distance-based metrics for our analysis. While these metrics were not specifically designed for robustness ranking, they have demonstrated utility in identifying issues with model performance degradation in previous studies [117, 118].

The first metric we considered is the Wasserstein distance (WD) [53], which measures the first- and second-order distance between two distributions. Another metric is the maximum mean discrepancy (MMD) [55], a kernel-based technique that distinguishes between two probability distributions based on their mean embeddings in a reproducing kernel Hilbert space. The Kolmogorov–Smirnov (KS) statistic [134] is a statistical test that is sensitive to differences in the mean and dispersion of two distributions. The Hellinger distance (HL) [135] measures the similarity between two probability distributions. HL is symmetric, well-defined for categorical and numerical features, and widely accepted in the industry. A larger HL value indicates greater dissimilarity between the distributions, while a smaller value indicates higher similarity or overlap. Lastly, we considered the Kullback-Leibler (KL) divergence [54], which is a widely used measure that captures the information-based disparity between two distributions. KL divergence assesses how much information is lost when one distribution is used to approximate another.

We believe that this set of selected metrics covers different assumptions about the underlying data and captures various deviations between the output features of trained and operational data. For example, WD considers the mean and standard deviation of the distributions, while MMD measures the discrepancy between distribution features in a reproducing kernel Hilbert space. HL is symmetric and has a clear analogy to Euclidean distance, making it widely accepted in the industry for capturing the dissimilarity between probability distributions. KS test compares each dimension separately and identifies the largest difference across all dimensions. Although KL divergence is not strictly a distance-based metric, we included it in our study because it can quantify the difference between the distributions of model outputs on operation data and training data in terms of information content. By incorporating KL divergence alongside other distance-based metrics, we can obtain a more comprehensive understanding of the distributional differences and their impact on model performance.

5.4.2 Evaluation Results

Result of RQ4.1: Effectiveness under OOD shifts. We assessed the suitability of different metrics in ranking the robustness of both CIFAR10 and ImageNet models. The impact of corruption types, varying percentages of corrupted input, and a mixture of corruption types can be summarized as follows:

- **Corruption type.** For CIFAR10 models, WD and MMD emerged as the top two performing metrics, whereas, MMD and KS outperformed other metrics when ranking ImageNet models.
- **Percentage of corruptions.** For CIFAR10 models, all five metrics exhibited an increasing trend in ranking accuracy as the percentage of corrupted data rose. However, for ImageNet models, WD and MMD metrics showed an increasing trend in ranking accuracy with increasing corruption percentage, where KS shows almost constant ranking accuracy.
- **Mixtures of corruptions.** For CIFAR10 models, WD and MMD showed effectiveness in ranking robustness under mixed input scenarios, while KS and MMD achieved satisfactory performance for ImageNet models.

Result of RQ4.2: Sample size impact. Our empirical results indicate that a minimum of 200 samples is required to achieve reliable ranking performance with a mean AP@1 score of over 0.50 for CIFAR10 (M1–M5) models. In the case of ImageNet models, the MMD metric consistently outperforms others even with a smaller sample size of 50. while the KS metric shows satisfactory performance in some cases but lacks stability. We observed that with a sample size of 500 samples, the ranking results tended to be more consistent and satisfactory. The impact of sample size on model ranking may vary depending on the dataset and the specific metric being used. However, to ensure a reliable model ranking, we recommend a minimum sample size of 500.

Our findings demonstrate that the WD metric performs best in ranking the robustness of CIFAR10 models, while the KS metric is optimal for ranking the robustness of ImageNet models. In contrast, the MMD metric is found to be sub-optimal for both datasets. To further understand these findings, a visualizing technique called UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction) [235] was employed. The UMAP visualizations (Figure 5.10) showed that the softmax outputs of CIFAR10 models exhibited a scattered stripe-like distribution, indicating more diverse and spread-out predictions across the output space. This characteristic was effectively captured by the WD metric, which considers both the mean and standard deviation of the distributions. In contrast, the softmax outputs of ImageNet models exhibited a cluster-based distribution, indicating more concentrated and less variable predictions. The KS metric, sensitive to differences in mean and dispersion, performed well in detecting variations in these clustered distributions. The MMD metric, which considers mean embeddings, was able to capture variations in distributions regardless of their specific patterns or structures. It suggests that the metrics assumptions and characteristics of the data to be analyzed shall be considered when selecting the most appropriate metric. Section 4 of Paper P4 in Part II provides a comprehensive analysis of the evaluation results.

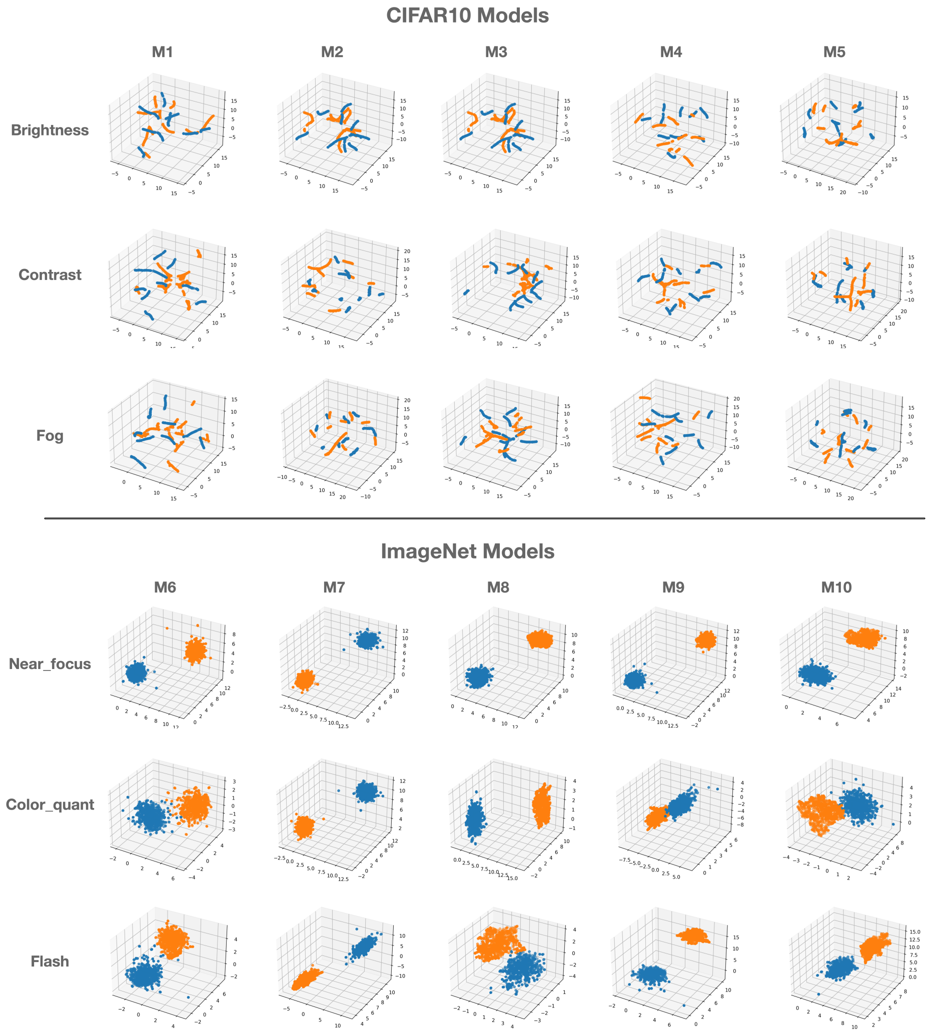


Figure 5.10: 3D UMAP visualization: Some examples of the softmax output of Models M1-M10 given training data and corrupted operation data. Data source: blue – training data, orange – operation data.

6 Discussion

This chapter consolidates the research findings from the previous chapter, discussing the contributions made to ensure the trustworthiness and reliability of ANN-SCSs. It provides implications for both academia and industry. Furthermore, the chapter examines the social impact of robustness evaluation and acknowledges the limitations of the study, emphasizing the need for further research.

6.1 Comparison with Related Work

The *first contribution* of this Ph.D. thesis, related to RQ1, is to provide a comprehensive overview of the current research on the T&V of ANN-SCSs.

The T&V of ANN-SCSs has been inadequately investigated, especially modern ANNs using deep learning techniques. This gap reflects increasingly serious concerns to assure robustness, improve failure resilience, ensure test completeness, and ensure safety properties since ANNs are increasingly being used in safety-critical domains. Existing research at the time of writing this thesis has referred to the T&V of ANN-SCSs from the traditional V&V perspective [34]. This perspective categorizes methods into automated testing and testing data generation methods, run-time monitoring, formal methods, cross-validation, and visualization. At the same time, many traditional T&V techniques are no longer effective for verifying ANNs in many cases. Hence, the complete investigation of T&V approaches for modern ANN-SCSs was essential in the evolution and progress of this field. The prerequisite to deal with any potential unexpected performance of ANN models in critical systems is a comprehensive understanding of approaches and tools to verify ANNs' performance. The results of P1 revealed that ANN robustness evaluation is a crucial aspect that needs to be addressed in order to ensure the reliability and safety of SCSs. The research community has become increasingly aware of the importance of ANN robustness, recognizing it as a widespread issue affecting the trustworthiness of ANN-SCSs. This highlights the need for further research and investigation to pave the way for the wider adoption and deployment of ANN models in SCSs.

The *second contribution* of this thesis, related to RQ2, is to shed light on the impact of ANN robustness on the identification and management of risks of autonomous systems [236].

At the time of this study, little research addressed the risk management of AVs, not to mention conducting a quantitative risk analysis. It is clear that detailed hazard identification can partly be done using traditional hazard identification methods for AVs. Determining the probabilities of failure and the risk from failures depends in part on identifying errors and weaknesses in the design and in part on determining the performance limits of the AV controller components. Many of the potential accidents involve the demands on the controllers being outside the domains of robust controller performance. Determining the limits of robust performance requires testing, preferably at the component level rather than on-road entire system testing. The probabilities of failure and the resultant risk will in many cases be the probabilities of circumstances arising outside the robust performance domains of the components. But robustness determination of ANN components in AVs is not often taken into consideration in the risk analysis. Contributing to this gap, we proposed an extended FTA to perform functional failure analysis for the ANN components of the AV control. After our work, Dong et al. [71] introduced a reliability assessment model (RAM) for ANN classifiers that utilizes the operational profile and robustness verification evidence. The scope of RAM [71] is complemented by our work. They focus on predicting the reliability of a necessarily flawed classifier, accepting that perfect robustness is not attainable. RAM [71] avoids considerations of rare but extreme classifications, focusing on average performance assuming that there will normally be errors in classification and that the probability of these must be reduced, since they cannot be eliminated. Our study (Paper P2 in Part II) focused on the probability of a robust classifier or recognizer failing due to operation at the limits of its design. For good risk analysis, different methods to deal with both sources of risk (rare and average events) are needed.

The *third contribution* of this thesis, related to RQ3, focuses on the critical analysis of a specific robustness evaluation context by investigating how the robustness of ANN-SCSs can be evaluated during operation and what factors should be included in the evaluation.

Tocchetti et al. [237] surveyed the terminology of concepts around AI robustness. They introduced three taxonomies: 1) robustness by methods and approaches in different phases of the machine learning pipeline; 2) robustness for specific model architectures, tasks, and systems; and 3) robustness assessment methodologies and insights from a fundamental and applied point of view. They also identified the lack of human perspective in evaluating AI robustness and the need to better understand practices and develop supportive tools for AI practitioners. Surveys such as the ones by Zhang et al. [44], Riccio et al. [238], and Ashmore et al. [69] provide a detailed view of the state-of-the-art techniques for evaluating particular properties of ML systems and obtaining assurances. While existing reviews offer useful insights into the robustness research considering ANN models, our study is different because, to our knowledge, it is the first comprehensive study that analyzes existing robustness evaluation approaches and metrics applicable to ANN-SCSs in operation. Recently, researchers have highlighted the need to consider the entire system and the interactions of various components within the system [239]. Furthermore, our study goes beyond the ANN model or input data aspect by investigating the robustness evaluation at the system, model, and input levels.

The *fourth contribution* of this thesis, in relation to RQ4, is to evaluate the suitability of different distance-based metrics for dynamically ranking the robustness of multiple ANN models within an SCS during operation. This involves identifying appropriate distance-based metrics and comparatively assessing these metrics based on their capability to rank the robustness of multiple ANN models.

With the growing trend of using multiple versions of AI models in operation, such as in ChatGPT, we expect redundant ANN models to become standard in ANN-SCS to increase reliability. This is similar to traditional SCSs, where redundant hardware and software are used to enhance reliability by intentionally duplicating critical components or functions. However, there is currently a lack of research on robustness evaluation for ANN-SCS with redundant ANN models. We understand that different ANN models can be fooled by the same input perturbations, like adversarial examples, and often fail silently. Increased uncertainty and risk are becoming inherent in ANN-SCS during operation, which necessitates the development of dedicated evaluation methods to compare the robustness of multiple ANN models and automatically decide which one to use during operation.

The dynamic ranking of multiple ANN models faces two main challenges. First, evaluation metrics typically require the ground truth, which might not be available during inference. This leads to using sample-selection-based methods or techniques like AutoEval [115] for automated model ranking, but these methods still require labeling or training a separate supervisor model. Second, determining the acceptable performance level for ANN models is another challenge. Standard industry practices include drift detection, but existing approaches like statistical-based and distance-based methods have limitations in directly comparing multiple models or estimating the magnitude of performance degradation. Thus, further investigation is needed to determine if these techniques can effectively compare and rank the robustness of multiple ANN classifiers. Through an extensive comparative evaluation of distance-based metrics in the context of robustness ranking for multiple ANN models, the study of this thesis provides novel valuable insights into a better understanding of robustness evaluation for ANN-SCS using multi-model decision-maker architecture. It guides researchers and practitioners in selecting appropriate metrics for their studies, ultimately improving the reliability and trustworthiness of utilizing optimal ANN models in SCSs.

6.2 Implications

The results of this thesis have several implications that can benefit both the research community and industry and provide suggestions for future research.

6.2.1 Implications for Academia

The results of the thesis outline the pre-paradigmatic nature of this research field and the need for further research through both qualitative and quantitative studies.

With regard to the contribution made by this thesis, several opportunities for future research have been identified, as follows:

- **AI quality testing and framework.** The development of a comprehensive AI quality framework is essential to ensure that ANN-SCSs meet the required robustness and reliability standards. Recently, Germany's first AI Quality and Testing Hub was established by the Hessian Minister of Digital Strategy and Development to enhance AI quality through standardization, certification schemes, and testing capabilities. Similar initiatives to promote AI quality are also being pursued in other countries such as the United States, France, China, and Australia [240]. This framework should address both model robustness and resilience in the face of real-world changes and the security of AI models.

- **Model management and evaluation.** As the complexity of ANN-SCSs grows, managing and evaluating the various ANN models becomes increasingly challenging. For example, the fatal accident involving Uber's self-driving car demonstrates the consequences of inadequate model robustness evaluation. Moreover, in many ANN-SCSs, the system operator can choose which ANN model to use for a particular task [31]. This, however, may lead to potential issues with model maintenance and evaluation. As the number of ANN models increases, some may become outdated (e.g. having lower robust accuracy) due to insufficient training data or an outdated algorithm. This could result in life-threatening consequences if an obsolete model is chosen. Additionally, choosing the appropriate ANN model for a task can be difficult. Future research should support system operators in selecting the optimal ANN model for a task under various conditions (e.g., different operational profiles), aid in the maintenance of multiple variants of the models by ML engineers, and assist in the evaluation of the model's performance. By developing strategies for effective model selection and maintenance that prioritize robustness, researchers can help ensure that ANN-SCSs maintain reliable performance even when faced with unexpected or challenging inputs. This includes methods for evaluating model robustness and techniques for updating and maintaining models to ensure they remain robust throughout their lifecycle.

- **Monitoring and observability for models.** ANN models must have been trained and tested rigorously before deployment. While an ANN model is being used in operation, the model's performance may deteriorate due to a change in data distribution or adversarial attacks. Therefore, the ANN-SCS should be regularly monitored to detect any significant mismatch between the current operational distribution and the data distribution the system was last trained on [132]. Performance degradation can occur in the deployment stage of any model and degrades community trust in the models' validity. For example, in 2021, OpenAI's GPT-3 [241], a state-of-the-art language model, showcased its limitations when it produced incorrect or nonsensical answers to specific queries. These issues were attributed to distribution shifts and the model's inability to generalize well to some new inputs. Unlike traditional model monitoring, which focuses on aggregated metrics and alerts, model observability goes beyond surface-level monitoring [110, 242]. By examining the model's predictions, explainability information, production feature data, and training data, model observability aims to uncover insights and understand the factors driving regressions or anomalies in model behavior. It provides a deeper understanding of the model's actions and informs workflows for improvement. To enhance model observability, researchers can explore techniques for diagnosing, foreseeing, and managing performance degradation. Strategies for enhancing model robustness and generalizability should also be developed. By doing so, researchers can contribute to increased trust and confidence in ANN-SCSs.

- **Understanding error propagation and robustness in hierarchical ANN-SCS.** We found that current system-level evaluation usually ignores knowledge from the low-level components (such as ANN models). While understanding the error propagation from ANN models to the system output is crucial for evaluating the robustness of an ANN-SCS. The reason is that such an understanding helps us to identify the potential sources of errors and evaluate their impact on the overall system's performance. By considering error propagation, we can assess how the errors in low-level components affect the system's output and how these errors can propagate and accumulate to cause failures or safety hazards. Recognizing that system-level evaluation is a multi-stage process, it is essential to consider how variations from one stage are accumulated and transmitted to subsequent stages. Some stages may generate variations, while others may absorb them. Though it might be possible to develop robustness metrics for individual stages, such as an ANN model or non-ANN component, aggregating these metrics to evaluate a hierarchical ANN-SCS is a complex and unexplored challenge. This highlights the significance of addressing error propagation from low-level components to enhance system-level evaluation and improve overall system performance.

6.2.2 Implications for Industry

Predicting the behavior of the complex ANN-SCS in active interaction with natural environments and humans, such as AV, is difficult. To overcome this challenge, TEVV practitioners may use various techniques, including simulation, modeling, testing, and analysis, to ensure that the ANN-SCS meets specifications and performs as expected. This thesis focuses on the TEVV tasks of ANN-SCSs and makes contributions to the industry as follows:

- The mapping of reviewed T&V approaches to the software safety lifecycle in IEC 61508 can be used by practitioners to make informed decisions about testing and verifying their ANN-SCSs. It provides practitioners with a valuable resource for understanding the best practices and challenges in the field. Our literature review identified robustness evaluation as one of the biggest challenges in testing and verifying ANN-SCSs. Robustness evaluation must be a key consideration in the development and deployment of ANN-SCSs. It requires changes in industry standards and regulations to ensure their safe and trustworthy deployment. This research could lead to the development of new standards and regulations for testing and verifying ANN-SCSs, enhancing their safety.

- We developed an extended FTA to better understand and address the risks of ANN classifiers used in AVs. This enables TEVV practitioners to perform functional failure analysis for ANN classifiers of AV control, determine the robustness and performance limits of these ANNs, identify the conditions that challenge the ANNs' performance limits in the operating environment, and assess the extent to which additional robustness enhancements and other safety measures can compensate for any performance deficiencies in the ANNs. The implications of this research for the industry include the need to address potential vulnerabilities and failures in ANNs used in safety-critical applications, such as AVs, resulting in improved design and testing of these systems and a better understanding of potential vulnerabilities and failure modes.

- The proposed multidimensional robustness evaluation framework offers comprehensive and consistent knowledge and a roadmap for practitioners to assess the robustness of the system and its ANN models for deployment and to evaluate model performance during operation. The industry could have greater assurance in their deployment models to increase customer confidence in the product.

- We empirically compare and evaluate the effectiveness of several distance metrics to continuously rank the robustness of multiple ANN classifiers used in SCSs. This opens up potential avenues for future research on the ongoing monitoring of ANN-SCSs without labeling efforts for operation. To enable this, a collaborative effort between practitioners and those responsible for operating the ANN-SCS is essential. This involves assessing system output, periodically updating the ANN models, and tracking and improving the robustness of the ANN-SCS.

6.2.3 Social Impact

AI has rapidly become a critical part of our lives and is expected to impact society substantially. To be trusted, AI must be robust, a key characteristic highlighted by organizations such as NIST, the Organisation for Economic Co-operation and Development (OECD)[1], the European Union Digital Strategy[2], and U.S. Executive Order 13960[3]. The social impact of this thesis lies in its contribution towards the development and deployment of robust and trustworthy AI systems, which have

[1] https://www.oecd.org/digital/ai/principles/

[2] https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

[3] https://www.federalregister.gov/documents/2020/12/08/2020-27065/promoting-the-use-of-trustworthy- artificial-intelligence-in-the-federal-government

the potential to bring positive outcomes, mitigate negative consequences, uphold ethical considerations, and shape regulatory frameworks.

- **Trustworthy AI Systems**: By addressing the robustness evaluation of ANN-SCSs, the thesis addresses a critical aspect of AI safety. Robust AI systems inspire trust, particularly in safety-critical domains like healthcare, transportation, finance, and national security.
- **Ethical Considerations**: Ethical AI systems are built on principles such as fairness, transparency, and accountability [50]. Robustness plays a crucial role in upholding these ethical considerations. By focusing on robustness evaluation, the thesis contributes to the development of AI systems that are not only technically competent but also aligned with ethical guidelines. This ensures that AI technologies are developed and deployed in a manner that respects human values and promotes fairness and transparency.
- **Regulatory and Policy Implications**: The research findings and implications of this thesis can inform regulatory bodies and policymakers in shaping guidelines and standards for AI systems. Organizations such as NIST, OECD, and the European Union have already emphasized the importance of robustness in AI systems. By providing insights and recommendations for robustness evaluation, this thesis can support the development of regulations and policies that promote the deployment of trustworthy AI systems.

6.3 Limitations

This thesis serves as a first step in understanding and addressing the robustness evaluation of ANN-SCSs throughout their entire lifecycle, including design, development, deployment, and operation. Despite the contributions made, there are some limitations. First, an inherent limitation of the research (related to RQ1 with Paper P1 and RQ3 with Paper P3 in Part II) was the scarcity of publicly available information. To address this, we combined knowledge from various sources and used snowballing procedures to identify as many relevant studies as possible. Second, our case study (related to RQ2, Paper P2 in Part II) only relied on simplified mathematical formalism regarding failure causes in multidimensional space. A comprehensive safety analysis and reliability modeling for ANN components in SCSs are needed. However, this study is a foundation for further applying risk analysis methods to assess ANN component reliability, as confirmed by a recently published paper [71]. Finally, for the last part of this thesis (related to RQ4, Paper P4 in Part II), one limitation is that we only validate OOD robustness, which may have affected our conclusions' generalization. Robustness in practice is more comprehensive, as highlighted by Hendrycks and Dietterich [231]. For example, assessing adversarial robustness requires considering perceptible attacks and other distortions. The areas of adversarial robustness and OOD robustness and unusual events should be considered in a unified manner. Future work could involve evaluating system robustness in multiple scenarios and testing against diverse attacks and disturbances.

Additionally, the research could explore ways to increase the interpretability and transparency of ANNs to understand their behavior better and improve robustness. Overall, the focus should be on developing robustness-enhancing techniques and incorporating robustness evaluation as an integral part of the design and deployment process for ANN-SCSs.

7 Conclusions and Future Work

This chapter presents the final remarks along with avenues for future work.

7.1 Conclusions

The increasing reliance on ANNs in decision-making systems has highlighted the need for robust and reliable models operating in real-world environments. This Ph.D. thesis has provided a comprehensive view on evaluating ANN robustness for SCSs by identifying and addressing the challenges, proposing a systematic robustness evaluation framework, and investigating methods for analyzing and measuring ANN robustness in operation. The research was motivated by the new challenges in assuring the robustness of ANN-SCSs, particularly in the context of operation. In addition, the research could increase public trust in autonomous systems by enabling the safe and trustworthy deployment of ANN-based systems in safety-critical applications. This was done through the following four main contributions:

> **C1:** The systematization of knowledge and understanding for T&V of ANN-SCSs.This includes a systematic classification of T&V approaches for ANN-SCSs and the identification of challenges for advancing the state-of-the-art in T&V for ANN-SCSs.
>
> **C2:** A new method for analyzing the influence of ANN robustness on the safety of AVs.This new method allows for a more comprehensive and quantitative analysis of the relationship between ANN robustness and the safety and reliability of autonomous systems.
>
> **C3:** A systematization of knowledge and a framework for assessing ANN-SCSs' robustness in operation.The knowledge is organized by a framework, which allows for a more effective and efficient evaluation of the robustness of ANN models under real-world conditions.
>
> **C4:** New knowledge and understanding of how the robustness of multiple ANN models can be ranked using unlabeled data.This comprehensive

empirical investigation enhances the understanding of utilizing distance-based metrics for automated model selection and ranking the robustness of multiple models in operation.

7.2 Future Work

This thesis has shown that the robustness of ANN-SCSs can be evaluated through appropriate techniques. Furthermore, this thesis has provided important insights into the challenges of evaluating and assuring the robustness of ANNs for SCSs. Despite the challenges, it is important to continue researching and developing methods to ensure the safe and reliable deployment of ANNs in SCSs. The proposed framework and methods presented in this thesis can be a starting point for future research.

Building on the insights from this research, there is a clear need for a more in-depth understanding and resolution of the potential vulnerabilities of ANNs in SCSs. In future work, I intend to further explore the extension of FTA to handle continuum disturbances, not just discrete events. This novel approach has significant implications for risk analysis, particularly in relation to ANNs. While risk analysts and ANN experts will benefit from this insight, the potential to apply this method in a wider context could revolutionize risk analysis across various domains.

In addition, my future work will delve deeper into the critical challenges faced by AVs, particularly the question of the system's response when it fails to interpret a scenario correctly, or when interpretations are unstable or conflicting. Two potential responses exist: revert the control to the human driver or trigger an automated emergency procedure, with each choice presenting its unique complexities. If control is handed back to the driver, we must ensure continued situational awareness, which is an intricate challenge to be addressed. Alternatively, if the decision is to automate emergency responses, there's a significant need for improvement in risk analysis methods for potential accident scenarios. This would involve handling a broad range of situations and establishing automated responses that prioritize the safety of all involved.

References

[1] M. Rausand, *Reliability of safety-critical systems: theory and applications*. John Wiley & Sons, 2014.

[2] J. Zhang and J. Li, 'Testing and verification of neural-network-based safety-critical control software: A systematic literature review', *Information and Software Technology*, p. 106 296, 2020.

[3] S. Levin and J. C. Wong, *Self-driving uber kills arizona woman in first fatal crash involving pedestrian*, https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe, 2018.

[4] N. M. Relations, *Collision between a sport utility vehicle operating with partial driving automation and a crash attenuator*, https://www.ntsb.gov/investigations/Pages/HWY18FH011.aspx, accessed: 2022-02-15, 2018.

[5] N. M. Relations, *Collision between a sport utility vehicle operating with partial driving automation and a crash attenuator*, https://www.ntsb.gov/investigations/Pages/HWY18FH004.aspx, accessed: 2022-02-15, 2018.

[6] ISO/IEC/TR/29119-11, 'Software and systems engineering — software testing — part 11: Guidelines on the testing of AI-based systems', International Organization for Standardization, Standard, Nov. 2020.

[7] S. OREDA, *Offshore reliability data handbook (oreda)*, 2002.

[8] S. Hauge, M. A. Lundteigen, P. Hokstad and S. Håbrekke, 'Reliability prediction method for safety instrumented systems–pds method handbook, 2010 edition', *SINTEF report STF50 A*, vol. 6031, p. 460, 2010.

[9] H. Xu and S. Mannor, 'Robustness and generalization', *Machine learning*, vol. 86, no. 3, pp. 391–423, 2012.

[10] R. S. S. Kumar, D. O. Brien, K. Albert, S. Viljöen and J. Snover, 'Failure modes in machine learning systems', *arXiv preprint arXiv:1911.11034*, 2019.

[11] R. Taylor, J. Zhang, I. Kozin and J. Li, 'Safety and Security Analysis for Autonomous Vehicles', Technical University of Denmark, Tech. Rep., 2021, Technical Report. [Online]. Available: https://orbit.dtu.dk/en/publications/safety-and-security-analysis-for-autonomous-vehicles.

[12] A. Nguyen, J. Yosinski and J. Clune, 'Deep neural networks are easily fooled: High confidence predictions for unrecognizable images', pp. 427–436.

[13] I. J. Goodfellow, J. Shlens and C. Szegedy, ‘Explaining and harnessing adversarial examples’, *arXiv preprint arXiv:1412.6572*, 2014.

[14] M. Melis, A. Demontis, B. Biggio, G. Brown, G. Fumera and F. Roli, ‘Is deep learning safe for robot vision? adversarial examples against the icub humanoid’, in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 751–759. DOI: 10.1109/ICCVW.2017.94.

[15] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin and N. Usunier, ‘Parseval networks: Improving robustness to adversarial examples’, *arXiv preprint arXiv:1704.08847*, 2017.

[16] N. Carlini and D. Wagner, ‘Towards evaluating the robustness of neural networks’, in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. DOI: 10.1109/SP.2017.49. [Online]. Available: https://ieeexplore.ieee.org/ielx7/7957740/7958557/07958570.pdf?tp=&arnumber=7958570&isnumber=7958557.

[17] S. Gu and L. Rigazio, ‘Towards deep neural network architectures robust to adversarial examples’, *arXiv preprint arXiv:1412.5068*, 2014.

[18] M. Wu, M. Wicker, W. Ruan, X. Huang and M. Kwiatkowska, ‘A game-based approximate verification of deep neural networks with provable guarantees’, *arXiv preprint arXiv:1807.03571*, 2018.

[19] F. Reuben, R. R. Curtin, S. Saurabh and A. B. Gardner, ‘Detecting adversarial samples from artifacts’, *arXiv preprint arXiv:1703.00410*, 2017.

[20] W. Xu, D. Evans and Y. Qi, ‘Feature squeezing: Detecting adversarial examples in deep neural networks’, *arXiv preprint arXiv:1704.01155*, 2017.

[21] M. Wicker, X. Huang and M. Kwiatkowska, ‘Feature-guided black-box safety testing of deep neural networks’, in D. Beyer and M. Huisman, Eds., vol. 10805 LNCS, Springer Verlag, 2018, pp. 408–426, ISBN: 03029743 (ISSN); 9783319899596 (ISBN). DOI: 10.1007/978-3-319-89960-2_22.

[22] J. H. Metzen, T. Genewein, V. Fischer and B. Bischoff, ‘On detecting adversarial perturbations’, *arXiv preprint arXiv:1702.04267*, 2017.

[23] D. Yin, R. Gontijo Lopes, J. Shlens, E. D. Cubuk and J. Gilmer, ‘A fourier perspective on model robustness in computer vision’, *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[24] C. Xie and A. Yuille, ‘Intriguing properties of adversarial training at scale’, in *International Conference on Learning Representations*, 2019.

[25] S. Zheng, Y. Song, T. Leung and I. Goodfellow, ‘Improving the robustness of deep neural networks via stability training’, pp. 4480–4488.

[26] U. Shaham, Y. Yamada and S. Negahban, ‘Understanding adversarial training: Increasing local stability of neural nets through robust optimization’, *arXiv preprint arXiv:1511.05432*, 2015.

[27] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer and B. Lakshminarayanan, ‘Augmix: A simple data processing method to improve robustness and uncertainty’, *arXiv preprint arXiv:1912.02781*, 2019.

[28] T. G. Dietterichl, ‘The handbook of brain theory and neural networks-ensemble learning’, *MIT Press*, vol. 40, 2002.

[29] C. Liu *et al.*, ‘Ensembles of natural language processing systems for portable phenotyping solutions’, *Journal of biomedical informatics*, vol. 100, p. 103 318, 2019.

[30] J. Diffenderfer, B. Bartoldson, S. Chaganti, J. Zhang and B. Kailkhura, ‘A winning hand: Compressing deep networks can improve out-of-distribution robustness’, *Advances in Neural Information Processing Systems*, vol. 34, pp. 664–676, 2021.

[31] Z. Peng, J. Yang, T.-H. Chen and L. Ma, ‘A first look at the integration of machine learning models in complex autonomous driving systems: A case study on apollo’, in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 1240–1250.

[32] S. Shankar, R. Garcia, J. M. Hellerstein and A. G. Parameswaran, ‘Operationalizing machine learning: An interview study’, *arXiv preprint arXiv:2209.09125*, 2022.

[33] T. Giallella, ‘Incident number 320’, *AI Incident Database*, K. Lam, Ed., 2018. [Online]. Available: https://incidentdatabase.ai/cite/320.

[34] B. J. Taylor, M. A. Darrah and C. D. Moats, ‘Verification and validation of neural networks: A sampling of research in progress’, in *Intelligent Computing: Theory and Applications*, International Society for Optics and Photonics, vol. 5103, 2003, pp. 8–17.

[35] F. Falcini and G. Lami, ‘Challenges in certification of autonomous driving systems’, in *2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 286–293. DOI: 10.1109/ISSREW.2017.45.

[36] A. Arpteg, B. Brinne, L. Crnkovic-Friis and J. Bosch, ‘Software engineering challenges of deep learning’, in *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 50–59, ISBN: 1538673835.

[37] Mercedes-Benz, *Safety first for automated driving (SaFAD)*, https://group.mercedes-benz.com/innovation/case/autonomous/safety-first-for-automated-driving-2.html, Accessed: 2022-02-06, 2019.

[38] M. Schwall, T. Daniel, T. Victor, F. Favaro and H. Hohnhold, ‘Waymo public road safety performance data’, *arXiv preprint arXiv:2011.00038*, 2020.

[39] W. Rawat and Z. Wang, ‘Deep convolutional neural networks for image classification: A comprehensive review’, *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[40] S. Thomas and N. Tabrizi, ‘Adversarial machine learning: A literature review’, in *International Conference on Machine Learning and Data Mining in Pattern Recognition*, Springer, 2018, pp. 324–334.

[41] N. Akhtar and A. Mian, ‘Threat of adversarial attacks on deep learning in computer vision: A survey’, *Ieee Access*, vol. 6, pp. 14 410–14 430, 2018.

[42] X. Huang *et al.*, ‘A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability’, *Computer Science Review*, vol. 37, p. 100 270, 2020.

[43] N. Drenkow, N. Sani, I. Shpitser and M. Unberath, ‘Robustness in deep learning for computer vision: Mind the gap?’, *arXiv preprint arXiv:2112.00639*, 2021.

[44] J. M. Zhang, M. Harman, L. Ma and Y. Liu, ‘Machine learning testing: Survey, landscapes and horizons’, *IEEE Transactions on Software Engineering*, 2020.

[45] S. Mohseni, H. Wang, C. Xiao, Z. Yu, Z. Wang and J. Yadawa, ‘Taxonomy of machine learning safety: A survey and primer’, *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, 2022.

[46] R. Ehlers, ‘Formal verification of piece-wise linear feed-forward neural networks’, Springer, 2017, pp. 269–286.

[47] M. D. Zeiler and R. Fergus, ‘Visualizing and understanding convolutional networks’, Springer, 2014, pp. 818–833.

[48] M. T. Ribeiro, S. Singh and C. Guestrin, ‘Why should i trust you?: Explaining the predictions of any classifier’, ACM, 2016, pp. 1135–1144, isbn: 1450342329.

[49] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney and D. Song, ‘Anomalous example detection in deep learning: A survey’, *IEEE Access*, vol. 8, pp. 132 330–132 347, 2020.

[50] Q. Lu, L. Zhu, X. Xu, J. Whittle and Z. Xing, ‘Towards a roadmap on software engineering for responsible ai’, in *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, 2022, pp. 101–112.

[51] NIST, *AI RISK MANAGEMENT FRAMEWORK*, https://www.nist.gov/itl/ai-risk-management-framework, accessed:2023-01-31, 2023.

[52] Y. LeCun, Y. Bengio and G. Hinton, ‘Deep learning’, *nature*, vol. 521, no. 7553, p. 436, 2015.

[53] Y. Rubner, C. Tomasi and L. J. Guibas, ‘The earth mover’s distance as a metric for image retrieval’, *International journal of computer vision*, vol. 40, no. 2, p. 99, 2000.

[54] J. M. Joyce, ‘Kullback-leibler divergence’, in *International encyclopedia of statistical science*, Springer, 2011, pp. 720–722.

[55] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf and A. Smola, ‘A kernel method for the two-sample-problem’, *Advances in neural information processing systems*, vol. 19, 2006.

[56] IEC61508, ‘Functional safety of electrical/electronic/programmable electronic safety-related systems’, International Electrotechnical Commission, Standard, 2005.

[57] ISO31000, ‘Risk management — guidelines’, International Organization for Standardization, Standard, Feb. 2018.

[58] K. D. Sharma and S. Srivastava, ‘Failure mode and effect analysis (FMEA) implementation: A literature review’, *J Adv Res Aeronaut Space Sci*, vol. 5, pp. 1–17, 2018.

[59] H. Watson, ‘Launch control safety study’, NJ, Tech. Rep, Tech. Rep., 1961.

[60] D. S. Nielsen, *The cause/consequence diagram method as a basis for quantitative accident analysis.* Risø National Laboratory, 1971.

[61] C. Swann and M. Preston, ‘Twenty-five years of HAZOPs’, *Journal of loss prevention in the Process Industries*, vol. 8, no. 6, pp. 349–353, 1995.

[62] N. Leveson, ‘A new accident model for engineering safer systems’, *Safety science*, vol. 42, no. 4, pp. 237–270, 2004.

[63] S. Gupta and J. Bhattacharya, ‘Reliability analysis of a conveyor system using hybrid data’, *Quality and Reliability Engineering International*, vol. 23, no. 7, pp. 867–882, 2007.

[64] R. Taylor and I. Kozin, *Design for emergent safety problems in handbook of engineering systems design*, 2021.

[65] N. G. Leveson, ‘Software safety: Why, what, and how’, *ACM Computing Surveys (CSUR)*, vol. 18, no. 2, pp. 125–163, 1986.

[66] J. Taylor, ‘Fault tree and cause consequence analysis for control software validation’, 1982.

[67] N. G. Leveson and P. R. Harvey, ‘Software fault tree analysis’, *Journal of Systems and Software*, vol. 3, no. 2, pp. 173–181, 1983.

[68] N. Leveson, *Engineering a safer world: Systems thinking applied to safety.* MIT press, 2011.

[69] R. Ashmore, R. Calinescu and C. Paterson, ‘Assuring the machine learning lifecycle: Desiderata, methods, and challenges’, *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–39, 2021.

[70] G. Sutherland and A. Hessami, ‘Safety critical integrity assurance in large datasets’, in *Assuring Safe Autonomy, Proceedings of the 28th Safety-Critical Systems Symposium(SSS’20)*, 2020, ISBN: 9781713305668.

[71] Y. Dong *et al.*, ‘Reliability assessment and safety arguments for machine learning components in system assurance’, *ACM Transactions on Embedded Computing Systems*, 2022.

[72] R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu and I. Habli, ‘Guidance on the assurance of machine learning in autonomous systems (amlas)’, *arXiv preprint arXiv:2102.01564*, 2021.

[73] N. Webb *et al.*, ‘Waymo’s safety methodologies and safety readiness determinations’, *arXiv preprint arXiv:2011.00054*, 2020.

[74] ISO/IEC/TR24028-1, ‘Overview of trustworthiness in artificial intelligence’, International Organization for Standardization, Standard, May 2020.

[75] E. N. Boudette, *Tesla's self-driving system cleared in deadly crash*, https://www.nytimes.com/2017/01/19/business/tesla-model-s-autopilot-fatal-crash.html/, accessed:2023-01-16, 2017.

[76] D. Oberhaus, *iPhone X's face ID can be fooled with a 3D-printed mask*, https://www.vice.com/en/article/qv3n77/iphone-x-face-id-mask-spoof, accessed:2023-01-27, 2017.

[77] J. Snow, *Amazon's face recognition falsely matched 28 members of congress with mugshots*, https://www.aclu.org/news/privacy-technology/amazons-face-recognition-falsely-matched-28, accessed:2023-01-27, 2018.

[78] C. Ross and I. Swetlitz, *IBM's watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show*, https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments/, accessed:2023-01-16, 2018.

[79] S. Russel and P. Norvig, *Artificial intelligence-a modern approach 2 edition*, 2003.

[80] S. Amershi *et al.*, 'Software engineering for machine learning: A case study', in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, IEEE, 2019, pp. 291–300.

[81] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodrıguez, N. V. Chawla and F. Herrera, 'A unifying view on dataset shift in classification', *Pattern recognition*, vol. 45, no. 1, pp. 521–530, 2012.

[82] M. Sugiyama, M. Yamada and M. C. du Plessis, 'Learning under nonstationarity: Covariate shift and class-balance change', *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 5, no. 6, pp. 465–477, 2013.

[83] E. Breck, S. Cai, E. Nielsen, M. Salib and D. Sculley, 'The ml test score: A rubric for ml production readiness and technical debt reduction', in *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, 2017, pp. 1123–1132.

[84] ISO/PAS21448, 'Road vehicles — Safety of the intended functionality', International Organization for Standardization, Standard, Jan. 2019.

[85] SAE, 'J3016:taxonomy and definitions for terms related to on-road motor vehicle automated driving systems', Standard, 2014. DOI: https://doi.org/10.4271/J3016_201401. [Online]. Available: https://doi.org/10.4271/J3016_201401.

[86] Koopman, Philip, *The UL 4600 guidebook*, https://safeautonomy.blogspot.com/2022/11/blog-post.html, Accessed: 2023-02-23, 2022.

[87] Ministry of Defence, 'Defence standard 00–56 issue 4 part 1, safety management requirements for defence systems–requirements', 2007.

[88] P. Koopman, U. Ferrell, F. Fratrik and M. Wagner, ‘A safety standard approach for fully autonomous vehicles’, in *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings 38*, Springer, 2019, pp. 326–332.

[89] C. Wilkinson, J. Lynch, R. Bharadwaj and K. Woodham, ‘Verification of adaptive systems’, *Federal Aviation Administration, DOT/FAA/TC-16/4*, 2016.

[90] US Food and Drug Administration, ‘Artificial intelligence and machine learning in software as a medical device’, *Silverspring: US Food and Drug Administration*, 2019.

[91] The Data Safety Initiative Working Group, *Data safety guidance*, https://scsc.uk/r127G:2?t=1, Accessed: 2023-02-23, 2022.

[92] F. Falcini and G. Lami, ‘Deep learning in automotive: Challenges and opportunities’, in *Software Process Improvement and Capability Determination*, A. Mas, A. Mesquida, R. V. O’Connor, T. Rout and A. Dorling, Eds., Springer International Publishing, 2017, pp. 279–288, ISBN: 978-3-319-67383-7.

[93] G. Hains, A. Jakobsson and Y. Khmelevsky, ‘Towards formal methods and software engineering for deep learning: Security, safety and productivity for dl systems development’, in *Systems Conference (SysCon), 2018 Annual IEEE International*, IEEE, 2018, pp. 1–5.

[94] A. Adadi and M. Berrada, ‘Peeking inside the black-box: A survey on explainable artificial intelligence (xai)’, *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018, ISSN: 2169-3536.

[95] F. M. Hohman, M. Kahng, R. Pienta and D. H. Chau, ‘Visual analytics in deep learning: An interrogative survey for the next frontiers’, *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018, ISSN: 1077-2626. DOI: 10.1109/TVCG.2018.2843369.

[96] E. E. Alves, D. Bhatt, B. Hall, K. Driscoll, A. Murugesan and J. Rushby, ‘Considerations in assuring safety of increasingly autonomous systems’, Tech. Rep., 2018.

[97] S. Burton, I. Habli, T. Lawton, J. McDermid, P. Morgan and Z. Porter, ‘Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective’, *Artificial Intelligence*, vol. 279, p. 103 201, 2020.

[98] G. Katz, C. Barrett, D. L. Dill, K. Julian and M. J. Kochenderfer, ‘Reluplex: An efficient smt solver for verifying deep neural networks’, in *International Conference on Computer Aided Verification*, Springer, 2017, pp. 97–117.

[99] Y. A. Mahmood, A. Ahmadi, A. K. Verma, A. Srividya and U. Kumar, ‘Fuzzy fault tree analysis: A review of concept and application’, *International Journal of System Assurance Engineering and Management*, vol. 4, no. 1, pp. 19–32, 2013.

[100] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori and A. Criminisi, ‘Measuring neural net robustness with constraints’, *Advances in neural information processing systems*, vol. 29, 2016.

[101] I. Buzhinsky, A. Nerinovsky and S. Tripakis, ‘Metrics and methods for robustness evaluation of neural networks with generative models’, *Machine Learning*, pp. 1–36, 2021.

[102] F. Yu, Z. Qin, C. Liu, L. Zhao, Y. Wang and X. Chen, ‘Interpreting and evaluating neural network robustness’, *arXiv preprint arXiv:1905.04270*, 2019.

[103] H. L. França, C. Teixeira and N. Laranjeiro, ‘Techniques for evaluating the robustness of deep learning systems: A preliminary review’, in *2021 10th Latin-American Symposium on Dependable Computing (LADC)*, IEEE, 2021, pp. 1–5.

[104] X. Xie *et al.*, ‘Deephunter: A coverage-guided fuzz testing framework for deep neural networks’, in *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2019, pp. 146–157.

[105] N. Carlini *et al.*, ‘On evaluating adversarial robustness’, *arXiv preprint arXiv:1902.06705*, 2019.

[106] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt and A. Madry, ‘Exploring the landscape of spatial robustness’, in *International Conference on Machine Learning*, 2019, pp. 1802–1811.

[107] D. Hendrycks and T. Dietterich, ‘Benchmarking neural network robustness to common corruptions and perturbations’, *arXiv preprint arXiv:1903.12261*, 2019.

[108] J. Gilmer, N. Ford, N. Carlini and E. Cubuk, ‘Adversarial examples are a natural consequence of test error in noise’, in *International Conference on Machine Learning*, PMLR, 2019, pp. 2280–2289.

[109] OpenAI, *ChatGPT: Language models as virtual assistants*, https://openai.com/blog/chatgpt, Accessed: May 4, 2023, 2021.

[110] C. Chen, N. Murphy, K. Parisa, D. Sculley and T. Underwood, *Reliable Machine Learning: Applying SRE Principles to ML in Production.* O’Reilly Media, Incorporated, 2022, isbn: 9781098106225. [Online]. Available: https://books.google.no/books?id=1rvHzgEACAAJ.

[111] W. Ma, M. Papadakis, A. Tsakmalis, M. Cordy and Y. L. Traon, ‘Test selection for deep learning systems’, *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 30, no. 2, pp. 1–22, 2021.

[112] L. Meng *et al.*, ‘Measuring discrimination to boost comparative testing for multiple deep learning models’, in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, IEEE, 2021, pp. 385–396.

[113] O. Sagi and L. Rokach, ‘Ensemble learning: A survey’, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.

[114] R. L. Ebel, ‘Procedures for the analysis of classroom tests’, *Educational and Psychological Measurement*, vol. 14, no. 2, pp. 352–364, 1954.

[115] W. Deng and L. Zheng, ‘Are labels always necessary for classifier accuracy evaluation?’, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 069–15 078.

[116] Y. Xiao *et al.*, ‘Self-checking deep neural networks in deployment’, in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, IEEE, 2021, pp. 372–384.

[117] G. A. Lewis, S. Echeverrıa, L. Pons and J. Chrabaszcz, ‘Augur: A step towards realistic drift detection in production ml systems’, 2022.

[118] S. Rabanser, S. Günnemann and Z. Lipton, ‘Failing loudly: An empirical study of methods for detecting dataset shift’, *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[119] I. Goldenberg and G. I. Webb, ‘Survey of distance measures for quantifying concept drift and shift in numeric data’, *Knowledge and Information Systems*, vol. 60, no. 2, pp. 591–615, 2019.

[120] L. T. Blessing and A. Chakrabarti, *DRM: A design research methodology.* Springer, 2009.

[121] C. Robson and K. McCartan, *Real world research.* Wiley Global Education, 2016.

[122] S. Keele *et al.*, ‘Guidelines for performing systematic literature reviews in software engineering’, Technical report, ver. 2.3 ebse technical report. ebse, Tech. Rep., 2007.

[123] K. Petersen, S. Vakkalanka and L. Kuzniarz, ‘Guidelines for conducting systematic mapping studies in software engineering: An update’, *Information and Software Technology*, vol. 64, pp. 1–18, 2015, ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2015.03.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584915000646.

[124] M. Shahin, M. A. Babar and L. Zhu, ‘Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices’, *IEEE Access*, vol. 5, pp. 3909–3943, 2017, ISSN: 2169-3536.

[125] P. H. Nguyen, S. Ali and T. Yue, ‘Model-based security engineering for cyber-physical systems: A systematic mapping study’, *Information and Software Technology*, vol. 83, pp. 116–135, 2017, ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2016.11.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584916303214%20https://www.sciencedirect.com/science/article/pii/S0950584916303214?via%5C%3Dihub.

[126] D. S. Cruzes and T. Dyba, ‘Recommended steps for thematic synthesis in software engineering’, in *2011 International Symposium on Empirical Software Engineering and Measurement*, 2011, pp. 275–284, ISBN: 1949-3789. DOI: 10.1109/ESEM.2011.36.

[127] ISO26262, ‘Road vehicles – Functional safety’, International Organization for Standardization, Standard, Nov. 2011.

[128] C. Voss, ‘Case research in operations management’, in *Researching operations management*, Routledge, 2010, pp. 176–209.

[129] R. K. Yin, *Case study research: Design and methods.* sage, 2009, vol. 5.

[130] N. Kalra and S. M. Paddock, ‘Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?’, *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[131] B. G. Glaser and A. L. Strauss, *The discovery of grounded theory: Strategies for qualitative research.* Routledge, 2017.

[132] ISO/IEC TR 24029-1, ‘Assessment of the robustness of neural networks Part1: Overview’, International Organization for Standardization, Standard, Mar. 2021.

[133] D. S. Cruzes and T. Dyba, ‘Recommended steps for thematic synthesis in software engineering’, in *2011 international symposium on empirical software engineering and measurement*, IEEE, 2011, pp. 275–284.

[134] J. W. Pratt, J. D. Gibbons, J. W. Pratt and J. D. Gibbons, ‘Kolmogorov-smirnov two-sample tests’, *Concepts of nonparametric theory*, pp. 318–344, 1981.

[135] D. A. Cieslak, T. R. Hoens, N. V. Chawla and W. P. Kegelmeyer, ‘Hellinger distance decision trees are robust and skew-insensitive’, *Data Mining and Knowledge Discovery*, vol. 24, pp. 136–158, 2012.

[136] F. Croce *et al.*, ‘Robustbench: A standardized adversarial robustness benchmark’, *arXiv preprint arXiv:2010.09670*, 2020.

[137] R. Tian, Z. Wu, Q. Dai, H. Hu and Y. Jiang, ‘Deeper insights into vits robustness towards common corruptions’, *arXiv preprint arXiv:2204.12143*, 2022.

[138] N. B. Erichson, S. H. Lim, F. Utrera, W. Xu, Z. Cao and M. W. Mahoney, ‘Noisymix: Boosting robustness by combining data augmentations, stability training, and noise injections’, *arXiv preprint arXiv:2202.01263*, vol. 1, 2022.

[139] D. Hendrycks *et al.*, ‘The many faces of robustness: A critical analysis of out-of-distribution generalization’, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8340–8349.

[140] O. F. Kar, T. Yeo, A. Atanov and A. Zamir, ‘3d common corruptions and data augmentation’, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 963–18 974.

[141] Y.-M. Tamm, R. Damdinov and A. Vasilev, ‘Quality metrics in recommender systems: Do we calculate metrics consistently?’, in *Proceedings of the 15th ACM Conference on Recommender Systems*, 2021, pp. 708–713.

[142] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori and A. Criminisi, ‘Measuring neural net robustness with constraints’, pp. 2613–2621.

[143] D. Gopinath, G. Katz, C. S. Pasareanu and C. Barrett, ‘Deepsafe: A data-driven approach for checking adversarial robustness in neural networks’, *arXiv preprint arXiv:1710.00486*, 2017.

[144] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, ‘Distillation as a defense to adversarial perturbations against deep neural networks’, *arXiv preprint arXiv:1511.04508*, 2015.

[145] N. Papernot and P. McDaniel, ‘Extending defensive distillation’, *arXiv preprint arXiv:1705.05264*, 2017.

[146] C. Schorn, A. Guntoro and G. Ascheid, ‘Accurate neuron resilience prediction for a flexible reliability management in neural network accelerators’, in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 979–984. DOI: 10.23919/DATE.2018.8342151. [Online]. Available: https://ieeexplore.ieee.org/ielx7/8337149/8341968/08342151.pdf?tp=&arnumber=8342151&isnumber=8341968.

[147] Q. Zhang, T. Wang, Y. Tian, F. Yuan and Q. Xu, ‘ApproxANN: An approximate computing framework for artificial neural network’, EDA Consortium, 2015, pp. 701–706, ISBN: 3981537041.

[148] J.-C. Vialatte and F. Leduc-Primeau, ‘A study of deep learning robustness against computation failures’, *arXiv preprint arXiv:1704.05396*, 2017.

[149] G. Li, K. Pattabiraman, C.-Y. Cher and P. Bose, ‘Understanding error propagation in gpgpu applications’, IEEE, 2016, pp. 240–251, ISBN: 1467388157.

[150] F. F. d. Santos, L. Draghetti, L. Weigel, L. Carro, P. Navaux and P. Rech, ‘Evaluation and mitigation of soft-errors in neural network-based object detection in three gpu architectures’, in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 169–176. DOI: 10.1109/DSN-W.2017.47. [Online]. Available: https://ieeexplore.ieee.org/ielx7/8020727/8023676/08023727.pdf?tp=&arnumber=8023727&isnumber=8023676.

[151] S. R. Manikandasriram, C. Anderson, R. Vasudevan and M. Johnson-Roberson, ‘Failing to learn: Autonomously identifying perception failures for self-driving cars [arxiv]’, *arXiv*, vol. arXiv:1707.00051, 8 pp. 2017. [Online]. Available: http://arxiv.org/abs/1707.00051.

[152] E. M. E. Mhamdi, R. Guerraoui and S. Rouault, ‘On the robustness of a neural network’, in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pp. 84–93. DOI: 10.1109/SRDS.2017.21. [Online]. Available: https://ieeexplore.ieee.org/ielx7/8067712/8068302/08069071.pdf?tp=&arnumber=8069071&isnumber=8068302.

[153] G. Li *et al.*, ‘Understanding error propagation in deep learning neural network (dnn) accelerators and applications’, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ACM, 2017, p. 8.

[154] A. H. M. Rubaiyat, Y. Qin and H. Alemzadeh, ‘Experimental resilience assessment of an open-source driving agent’, *CoRR*, vol. abs/1807.06172, 2018. arXiv: 1807.06172. [Online]. Available: http://arxiv.org/abs/1807.06172.

[155] K. Rhazali, B. Lussier, W. Schön and S. Geronimi, ‘Fault tolerant deep neural networks for detection of unrecognizable situations’, *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 31–37, 2018, issn: 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2018.09.525. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S240589631832216X.

[156] S. Daftry, S. Zeng, J. A. Bagnell and M. Hebert, ‘Introspective perception: Learning to predict failures in vision systems’, in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1743–1750. doi: 10.1109/IROS.2016.7759279.

[157] M. O’Kelly, H. Abbas and R. Mangharam, ‘Computer-aided design for safe autonomous vehicles’, in *Resilience Week (RWS) 2017*, ser. 2017 Resilience Week (RWS), IEEE, 2017, pp. 90–6. doi: 10.1109/RWEEK.2017.8088654.

[158] K. Pei, Y. Cao, J. Yang and S. Jana, ‘DeepXplore: Automated whitebox testing of deep learning systems’, Association for Computing Machinery, Inc, 2017, pp. 1–18, isbn: 9781450350853 (ISBN). doi: 10.1145/3132747.3132785.

[159] Y. Tian, K. Pei, S. Jana and B. Ray, ‘Deeptest: Automated testing of deep-neural-network-driven autonomous cars’, in *Proceedings of the 40th International Conference on Software Engineering*, ser. ICSE ’18, Gothenburg, Sweden: ACM, 2018, pp. 303–314, isbn: 978-1-4503-5638-1. doi: 10.1145/3180155.3180220. [Online]. Available: http://doi.acm.org/10.1145/3180155.3180220.

[160] S. Raj, S. K. Jha, A. Ramanathan and L. L. Pullum, ‘Work-in-progress: Testing autonomous cyber-physical systems using fuzzing features from convolutional neural networks’, in *2017 International Conference on Embedded Software (EMSOFT)*, pp. 1–2. doi: 10.1145/3125503.3125568.

[161] L. Ma *et al.*, ‘DeepGauge: Multi-granularity testing criteria for deep learning systems’, in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, Montpellier, France: ACM, 2018, pp. 120–131, isbn: 978-1-4503-5937-5. doi: 10.1145/3238147.3238202. [Online]. Available: http://doi.acm.org/10.1145/3238147.3238202.

[162] M. Zhang, Y. Zhang, L. Zhang, C. Liu and S. Khurshid, ‘DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems’, in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, ACM, 2018, pp. 132–142.

[163] J. Guo, Y. Jiang, Y. Zhao, Q. Chen and J. Sun, ‘DLFuzz: Differential fuzzing testing of deep learning systems’, in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ACM, 2018, pp. 739–743.

[164] L. Pulina and A. Tacchella, ‘Challenging smt solvers to verify neural networks’, *AI Communications*, vol. 25, no. 2, pp. 117–135, 2012, issn: 0921-7126.

[165] L. Pulina and A. Tacchella, ‘NeVer: A tool for artificial neural networks verification’, *Annals of Mathematics and Artificial Intelligence*, vol. 62, no. 3-4, pp. 403–425, 2011, ISSN: 1012-2443.

[166] S. Dutta, S. Jha, S. Sanakaranarayanan and A. Tiwari, ‘Output range analysis for deep neural networks’, *arXiv preprint arXiv:1709.09130*, 2017.

[167] W. Xiang, H. D. Tran and T. T. Johnson, ‘Output reachable set estimation and verification for multilayer neural networks’, *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–7, 2018, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2018.2808470.

[168] K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen and M. J. Kochenderfer, ‘Policy compression for aircraft collision avoidance systems’, IEEE, 2016, pp. 1–10, ISBN: 1509025235.

[169] W. Xiang, H.-D. Tran and T. T. Johnson, ‘Reachable set computation and safety verification for neural networks with relu activations’, *arXiv preprint arXiv:1712.08163*, 2017.

[170] X. Huang, M. Kwiatkowska, S. Wang and M. Wu, ‘Safety verification of deep neural networks’, in *International Conference on Computer Aided Verification*, Springer, 2017, pp. 3–29.

[171] N. Narodytska, S. P. Kasiviswanathan, L. Ryzhyk, M. Sagiv and T. Walsh, ‘Verifying properties of binarized deep neural networks’, *arXiv preprint arXiv:1709.06662*, 2017.

[172] C.-H. Cheng, G. Nührenberg and H. Ruess, ‘Verification of binarized neural networks’, *arXiv preprint arXiv:1710.03107*, 2018.

[173] T. Dreossi, A. Donzé and S. A. Seshia, ‘Compositional falsification of cyber-physical systems with machine learning components’, ser. NASA Formal Methods, Springer International Publishing, 2017, pp. 357–372, ISBN: 978-3-319-57288-8.

[174] P. Mallozzi, P. Pelliccione and C. Menghi, ‘Keeping intelligence under control’, in *Proceedings of the 1st International Workshop on Software Engineering for Cognitive Services*, Gothenburg, Sweden: ACM, 2018, pp. 37–40, ISBN: 978-1-4503-5740-1. DOI: 10.1145/3195555.3195558.

[175] C. Szegedy *et al.*, ‘Intriguing properties of neural networks’, *arXiv preprint arXiv:1312.6199*, 2013.

[176] M. T. Ribeiro, S. Singh and C. Guestrin, ‘Anchors: High-precision model-agnostic explanations’, in *Proceedings of the 32rd AAAI Conference on Artificial Intelligence*, 2018.

[177] M. Sundararajan, A. Taly and Q. Yan, ‘Axiomatic attribution for deep networks’, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3319–3328.

[178] S. Bach, A. Binder, K.-R. Müller and W. Samek, ‘Controlling explanatory heatmap resolution and semantics via decomposition depth’, in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2271–2275, ISBN: 1467399612.

[179] K. Simonyan, A. Vedaldi and A. Zisserman, ‘Deep inside convolutional networks: Visualising image classification models and saliency maps’, *arXiv preprint arXiv:1312.6034*, 2013.

[180] G. Montavon, S. Lapuschkin, A. Binder, W. Samek and K.-R. Müller, ‘Explaining nonlinear classification decisions with deep taylor decomposition’, *Pattern Recognition*, vol. 65, pp. 211–222, 2017, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2016.11.008.

[181] D. Linsley, D. Scheibler, S. Eberhardt and T. Serre, ‘Global-and-local attention networks for visual recognition’, *arXiv preprint arXiv:1805.08819*, 2018.

[182] Y. Dong, H. Su, J. Zhu and B. Zhang, ‘Improving interpretability of deep neural networks with semantic information’, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4306–4314.

[183] R. C. Fong and A. Vedaldi, ‘Interpretable explanations of black boxes by meaningful perturbation’, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437.

[184] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, ‘Learning deep features for discriminative localization’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929.

[185] M. A. K.-R. M. Dumitru, E. B. K. S. D. Pieter, J. Kindermans and K. T. Schütt, ‘Learning how to explain neural networks: Patternnet and patternattribution’, in *Proceedings of the International Conference on Learning Representations (2018).*

[186] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini and F. Giannotti, ‘Local rule-based explanations of black box decision systems’, *arXiv preprint arXiv:1805.10820*, 2018.

[187] A. Shrikumar, P. Greenside and A. Kundaje, ‘Learning important features through propagating activation differences’, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 3145–3153.

[188] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller and W. Samek, ‘On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation’, *PLOS ONE*, vol. 10, no. 7, pp. 1–46, Jul. 2015. DOI: 10.1371/journal.pone.0130140.

[189] P. Dabkowski and Y. Gal, ‘Real time image saliency for black box classifiers’, in *Advances in Neural Information Processing Systems*, pp. 6967–6976.

[190] A. S. Ross, M. C. Hughes and F. Doshi-Velez, ‘Right for the right reasons: Training differentiable models by constraining their explanations’, *arXiv preprint arXiv:1703.03717*, 2017.

[191] A. Santoro *et al.*, 'A simple neural network module for relational reasoning', in *Advances in neural information processing systems*, pp. 4967–4976.

[192] D. Smilkov, N. Thorat, B. Kim, F. Viégas and M. Wattenberg, 'Smoothgrad: Removing noise by adding noise', *arXiv preprint arXiv:1706.03825*, 2017.

[193] S. M. Lundberg and S.-I. Lee, 'A unified approach to interpreting model predictions', in *Advances in Neural Information Processing Systems*, pp. 4765–4774.

[194] N. Frosst and G. Hinton, 'Distilling a neural network into a soft decision tree', *arXiv preprint arXiv:1711.09784*, 2017.

[195] Z. Che, S. Purushotham, R. Khemani and Y. Liu, 'Distilling knowledge from deep networks with applications to healthcare domain', *arXiv preprint arXiv:1512.03542*, 2015.

[196] G. Hinton, O. Vinyals and J. Dean, 'Distilling the knowledge in a neural network', *arXiv preprint arXiv:1503.02531*, 2015.

[197] K. Xu, D. H. Park, C. Yi and C. Sutton, 'Interpreting deep classifier by visual distillation of dark knowledge', *arXiv preprint arXiv:1803.04042*, 2018.

[198] S. Tan, R. Caruana, G. Hooker, P. Koch and A. Gordo, 'Learning global additive explanations for neural nets using model distillation', *arXiv preprint arXiv:1801.08640*, 2018.

[199] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox and J. Clune, 'Synthesizing the preferred inputs for neurons in neural networks via deep generator networks', in *Advances in Neural Information Processing Systems*, pp. 3387–3395.

[200] W. Guo, D. Mu, J. Xu, P. Su, G. Wang and X. Xing, 'Lemna: Explaining deep learning based security applications', in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2018, pp. 364–379.

[201] O. Bastani, C. Kim and H. Bastani, 'Interpretability via model extraction', *arXiv preprint arXiv:1706.09773*, 2017.

[202] J. J. Thiagarajan, B. Kailkhura, P. Sattigeri and K. N. Ramamurthy, 'Treeview: Peeking into deep neural networks via feature-space partitioning', *arXiv preprint arXiv:1611.07429*, 2016.

[203] A. Mahendran and A. Vedaldi, 'Understanding deep image representations by inverting them', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196.

[204] G. Katz, C. Barrett, D. L. Dill, K. Julian and M. J. Kochenderfer, 'Reluplex: An efficient smt solver for verifying deep neural networks', in *Computer Aided Verification. CAV 2017*, Springer, 2017, pp. 97–117.

[205] J. Stallkamp, M. Schlipsing, J. Salmen and C. Igel, 'Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition', *Neural networks*, vol. 32, pp. 323–332, 2012.

[206] Z. Zhong, Z. Hu and X. Chen, 'Quantifying dnn model robustness to the real-world threats', in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, IEEE, 2020, pp. 150–157.

[207] I. Goodfellow, J. Shlens and C. Szegedy, 'Explaining and harnessing adversarial examples', in *International Conference on Learning Representations*, 2015.

[208] IEEE, 'IEEE Standard Glossary of Software Engineering Terminology', *IEEE Std 610.12-1990*, pp. 1–84, 1990. doi: 10.1109/IEEESTD.1990.101064.

[209] 'ISO/IEC TS 5723:2022 Trustworthiness — Vocabulary', no. ISO/IEC TS 5723, 2022. [Online]. Available: https://www.iso.org/standard/81608.html.

[210] D. Diochnos, S. Mahloujifar and M. Mahmoody, 'Adversarial risk and robustness: General definitions and implications for the uniform distribution', *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[211] S. Zheng, Y. Song, T. Leung and I. Goodfellow, 'Improving the robustness of deep neural networks via stability training', in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 4480–4488.

[212] J. Wang *et al.*, 'Robot: Robustness-oriented testing for deep learning systems', in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, IEEE, 2021, pp. 300–311.

[213] Z. Zhong, Y. Tian and B. Ray, 'Understanding local robustness of deep neural networks under natural variations', in *Fundamental Approaches to Software Engineering: 24th International Conference, FASE 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27–April 1, 2021, Proceedings 24*, Springer International Publishing, 2021, pp. 313–337.

[214] J. Kim, R. Feldt and S. Yoo, 'Guiding deep learning system testing using surprise adequacy', vol. 2019-May, IEEE Computer Society, May 2019, pp. 1039–1049, isbn: 9781728108698. doi: 10.1109/ICSE.2019.00108.

[215] H. Zhou *et al.*, 'Deepbillboard: Systematic physical-world testing of autonomous driving systems', in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 347–358.

[216] J. Norden, M. O'Kelly and A. Sinha, 'Efficient black-box assessment of autonomous vehicle safety', *arXiv*, Dec. 2019. [Online]. Available: http://arxiv.org/abs/1912.03618.

[217] E. Beede *et al.*, 'A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy', *Conference on Human Factors in Computing Systems - Proceedings*, Apr. 2020. doi: 10.1145/3313831.3376718.

[218] D. E. A. T. Force and A. Daedalean, 'Concepts of design assurance for neural networks (codann)', *Concepts of Design Assurance for Neural Networks (CoDANN). EASA, Daedalean*, 2020.

[219] S. Kotyan and D. V. Vargas, ‘Adversarial robustness assessment: Why both L_0 and L_∞ attacks are necessary’, *arXiv preprint arXiv:1906.06026*, 2019.

[220] H.-T. D. Liu, M. Tao, C.-L. Li, D. Nowrouzezahrai and A. Jacobson, ‘Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer’, *arXiv preprint arXiv:1808.02651*, 2018.

[221] A. Laugros, A. Caplier and M. Ospici, ‘Are adversarial robustness and common perturbation robustness independant attributes?’, in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[222] B. Reagen *et al.*, ‘Ares: A framework for quantifying the resilience of deep neural networks’, in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6.

[223] Z. Chen, N. Narayanan, B. Fang, G. Li, K. Pattabiraman and N. DeBardeleben, ‘Tensorfi: A flexible fault injection framework for tensorflow applications’, in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, IEEE, 2020, pp. 426–435.

[224] C. Berghoff, P. Bielik, M. Neu, P. Tsankov and A. Von Twickel, ‘Robustness testing of ai systems: A case study for traffic sign recognition’, in *Artificial Intelligence Applications and Innovations: 17th IFIP WG 12.5 International Conference, AIAI 2021, Hersonissos, Crete, Greece, June 25–27, 2021, Proceedings 17*, Springer, 2021, pp. 256–267.

[225] C. Michaelis *et al.*, ‘Benchmarking robustness in object detection: Autonomous driving when winter is coming’, *arXiv preprint arXiv:1907.07484*, 2019.

[226] S. A. Hicks *et al.*, ‘On evaluation metrics for medical applications of artificial intelligence’, *Scientific Reports*, vol. 12, no. 1, p. 5979, 2022.

[227] S. Gerasimou, H. F. Eniser, A. Sen and A. Cakan, ‘Importance-driven deep learning system testing’, in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 702–713.

[228] S. Dola, M. B. Dwyer and M. L. Soffa, ‘Distribution-aware testing of neural networks using generative models’, in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, IEEE, 2021, pp. 226–237.

[229] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt and D. Song, ‘Natural adversarial examples’, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 262–15 271.

[230] J. Chen, Z. Wu, Z. Wang, H. You, L. Zhang and M. Yan, ‘Practical accuracy estimation for efficient deep neural network testing’, *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 29, no. 4, pp. 1–35, 2020.

[231] D. Hendrycks, N. Carlini, J. Schulman and J. Steinhardt, ‘Unsolved problems in ml safety’, *arXiv preprint arXiv:2109.13916*, 2021.

[232] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen and G. E. Dahl, 'Motivating the rules of the game for adversarial example research', *arXiv preprint arXiv:1807.06732*, 2018.

[233] D. Mincu and S. Roy, 'Developing robust benchmarks for driving forward ai innovation in healthcare', *Nature Machine Intelligence*, pp. 1–6, 2022.

[234] X. Liu, B. Glocker, M. M. McCradden, M. Ghassemi, A. K. Denniston and L. Oakden-Rayner, 'The medical algorithmic audit', *The Lancet Digital Health*, 2022.

[235] L. McInnes, J. Healy and J. Melville, 'Umap: Uniform manifold approximation and projection for dimension reduction', *arXiv preprint arXiv:1802.03426*, 2018.

[236] J. Zhang, R. Taylor, I. Kozin and J. Li, 'Analyzing influence of robustness of neural networks on the safety of autonomous vehicles', in *31st European Safety and Reliability Conference*, 2021, p. 2276.

[237] A. Tocchetti *et al.*, 'AI Robustness: a Human-Centered Perspective on Technological Challenges and Opportunities', *arXiv preprint arXiv:2210.08906*, 2022.

[238] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss and P. Tonella, 'Testing machine learning based systems: A systematic mapping', *Empirical Software Engineering*, vol. 25, pp. 5193–5254, 2020.

[239] S. Li, J. Guo, J.-G. Lou, M. Fan, T. Liu and D. Zhang, 'Testing machine learning systems in industry: An empirical study', in *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice*, 2022, pp. 263–272.

[240] L. Madzou, 'Deploying High-Quality and Trustworthy AI', *Artificial Intelligence*, vol. LYTX, no. 2023, pp. 01–01, Dec. 2022. doi: 10.1287/LYTX.2023.01.01.

[241] T. Brown *et al.*, 'Language models are few-shot learners', *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[242] Arize AI, *Machine learning observability 101*, https://arize.com/resource/ebook-machine-learning-observability-101/, Accessed 2023.

Part II

Research Papers

Research Papers

Main research papers added in full length:

P1 Zhang, J., and Li, J. (2020) **'Testing and verification of neural network-based safety-critical control software: A systematic literature review,'** *Journal of Information and Software Technology,123, 106296.*

P2 Zhang, J., Taylor, J. R., Kozin, I., and Li, J. (2021) **'Analyzing influence of robustness of neural networks on the safety of autonomous vehicles,'** *In 31st European Safety and Reliability Conference (ESREL), pp. 2276-2283.*

P3 Zhang, J., Li, J. and Oehmen, J. (2023) **'Robustness evaluation for safety-critical systems utilising artificial neural network classifiers in operation: A survey,'** *In review, Manuscript submitted to the International Journal of Engineering Application of Artificial Intelligence.*

P4 Zhang, J., Li, J., and Yang, Z. (2023) **'Dynamic robustness evaluation for automated model selection in operation,'** *In review, Manuscript submitted to the International Journal of Information and Software Technology.*

Secondary research papers added with abstract.

SP1 Li, J., Zhang, J., and Kaloudi, N. (2018, September). **'Could we issue driving licenses to autonomous vehicles?'** *In Conference on Computer Safety, Reliability, and Security (SAFECOMP), pp. 473-480.*

SP2 Guzman, N. H. C., Zhang, J., Xie, J., and Glomsrud, J. A.(2021) **'A comparative study of STPA-extension and the UFoI-E method for safety and security co-analysis,'** *Journal of Reliability Engineering and System Safety, 211,107633.*

SP3 Taylor, J. R., Zhang, J., Kozin, I., and Li, J. **'Safety and security analysis for autonomous vehicles,'** *Technical Report, DTU orbit, 2021.*

SP4 Staff, A., Zhang, J., Li, J., Xie, J., Traiger, E., Glomsrud, J., and Karolius, K. (2021) **'An empirical study on cross-data transferability of adversarial attacks on object detectors,'** *In the 41 Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence (SGAI 2021), AI-CyberSec 2021 workshop.*

SP5 Zhang, J., Oehmen, J., and Kozin, I. (2022) **'Monitoring the robustness of safety critical artificial neural networks,'** *European Safety and Reliability Association Newsletter, 3, 4-5.*

Paper 1

Testing and verification of neural network-based safety-critical control software: A systematic literature review,

Jin Zhang, and Jingyue Li.

In: Journal of Information and Software Technology,123, 106296.

Information and Software Technology 123 (2020) 106296

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Testing and verification of neural-network-based safety-critical control software: A systematic literature review

Jin Zhang[a,b], Jingyue Li[a,*]

[a] Computer Science Department, Norwegian University of Science and Technology, Trondheim, Norway
[b] School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China

ARTICLE INFO

Keywords:
Software testing and verification
Neural network
Safety-critical control software
Systematic literature review

ABSTRACT

Context: Neural Network (NN) algorithms have been successfully adopted in a number of Safety-Critical Cyber-Physical Systems (SCCPSs). Testing and Verification (T&V) of NN-based control software in safety-critical domains are gaining interest and attention from both software engineering and safety engineering researchers and practitioners.

Objective: With the increase in studies on the T&V of NN-based control software in safety-critical domains, it is important to systematically review the state-of-the-art T&V methodologies, to classify approaches and tools that are invented, and to identify challenges and gaps for future studies.

Method: By searching the six most relevant digital libraries, we retrieved 950 papers on the T&V of NN-based Safety-Critical Control Software (SCCS). Then we filtered the papers based on the predefined inclusion and exclusion criteria and applied snowballing to identify new relevant papers.

Results: To reach our result, we selected 83 primary papers published between 2011 and 2018, applied the thematic analysis approach for analyzing the data extracted from the selected papers, presented the classification of approaches, and identified challenges.

Conclusion: The approaches were categorized into five high-order themes, namely, assuring robustness of NNs, improving the failure resilience of NNs, measuring and ensuring test completeness, assuring safety properties of NN-based control software, and improving the interpretability of NNs. From the industry perspective, improving the interpretability of NNs is a crucial need in safety-critical applications. We also investigated nine safety integrity properties within four major safety lifecycle phases to investigate the achievement level of T&V goals in IEC 61508-3. Results show that correctness, completeness, freedom from intrinsic faults, and fault tolerance have drawn most attention from the research community. However, little effort has been invested in achieving repeatability, and no reviewed study focused on precisely defined testing configuration or defense against common cause failure.

1. Introduction

Cyber-Physical Systems (CPSs) are systems involving networks of embedded systems and strong human-machine interactions [1]. Safety-critical CPSs (SCCPSs) are a type of CPSs that highlights the severe non-functional constraints (e.g., safety and dependability). The failure of SC-CPSs could result in loss of life or significant damage (e.g., property and environmental damage). Typical applications of SCCPSs are in nuclear systems, aircraft flight control systems, automotive systems, smart grids, and healthcare systems.

In the last few years, advances in Neural Networks (NNs) have boosted the development and deployment of SCCPSs. The NN is considered the most viable approach to meet the complexity requirements of Safety-Critical Control Softwares (SCCSs) [2,3]. In this study, we refer to NN-based SCCS as SCCS that heavily use NNs (e.g., to implement controller). For example, in the transportation industry, deep-learning-based NNs have been widely used to developing self-driving cars [4] and collision avoidance systems [5]. It is also worth noting that several safety incidents caused by autonomous vehicles have been presented in media, e.g., Uber car's fatal incident [6], Tesla's fatal Autopilot crash [7], and Google's self-driving car crash [8]. In addition to the safety incidents caused by failures of the autonomous system, security breaches of autonomous vehicles can potentially lead to safety issues, e.g., a demo showed that autonomous vehicles can be remotely controlled and hijacked [9]. How can we ensure that an SCCS containing NN technology

[*] Corresponding author.
E-mail addresses: jin.zhang@ntnu.no (J. Zhang), jingyue.li@ntnu.no (J. Li).

https://doi.org/10.1016/j.infsof.2020.106296
Received 20 April 2019; Received in revised form 5 December 2019; Accepted 6 March 2020
Available online 7 March 2020

will behave correctly and consistently when system failures or malicious attacks occur?

Increasing interest in the migration of Industrial Control Systems (ICSs) towards SCCPSs has encouraged research in the area of safety analysis of SCCPSs. Kriaa et al. [10] surveyed existing approaches for an integrated safety and security analysis of ICSs. The approaches cover both the design stage and the operational stage of the system lifecycle. Some approaches (such as [11,12]) are aimed at combining safety and security techniques into a single methodology. Others (such as [13,14]) are trying to align safety and security techniques. These approaches are either generic, which consider both safety and security at a very high level, or model-based, which build upon the formal or semi-formal representation of the system's functions.

There are many studies that focus on the T&V of NNs in the past decade. Several review articles [15–18] on this topic have been published. Studies [15,19] have reviewed methods focusing on verification and validation of NNs for aerospace systems. Studies [17,18] are limited in automotive applications. None of these review articles have applied the Systematic Literature Review (SLR) [20] approach.

Recently there has been more concern about Artificial Intelligence (AI) safety. The state-of-the-art advancements in the T&V of NN-based SCCS are increasingly important; hence, there is a need to have a thorough understanding of present studies to incentivize further discussion. This study aimed to summarize the current research on **T&V methods for NN-based control software in SCCPSs**. We have systematically identified and reviewed 83 papers focusing on the T&V of NN-based SCCSs and synthesized the data extracted from those papers to answer three research questions.

- RQ1 What are the profiles of the studies focusing on testing and verifying NN-based SCCSs?
- RQ2 What approaches and associated tools have been proposed to test and verify NN-based SCCSs?
- RQ3 What are the limitations of current studies with respect to testing and verifying NN-based SCCSs?

To our best knowledge, our study is the first SLR on testing and verifying NN-based control software in SCCPSs. The results of these research questions can help researchers identify the research gaps in this area, and help industrial practitioners choose proper verification and certification methods.

The main contributions of this work are:

- We made a classification of T&V approaches in both academia and industry for NN-based SCCSs.
- We identified and proposed challenges for advancing state-of-the-art T&V for NN-based SCCSs.

The remainder of this paper is organized as follows: In Section 2, we define terminologies related to NN-based SCCPSs and summarize related work from academia and industry. Section 3 describes the SLR process and the review protocol. The results of the research questions are reported in Section 4. Section 5 discusses the industry practice of T&V of NN-based SCCSs, and the threats to validity of our study. Section 6 concludes the study.

2. Background

In this section, we first introduce terminology related to CPSs and modern NNs and show how NN algorithms have been used in SCCPSs. Then, we present the current state of practice of T&V of SCCSs.

2.1. Cyber-physical systems

As defined in Rajkumar et al. [1], "*cyber-physical systems (CPSs) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core.*" Several other systems, such as Internet of Things (IoTs) and ICSs have very similar features compared to CPSs, since they are all systems used to monitor and control the physical world with embedded sensor and actuator networks. In general, CPSs are perceived as the new generation of embedded control systems, which can involve IoTs and ICSs [21,22].

In this SLR, we adopted the CPS conceptual model in Griffor et al. [23] as a high-level abstraction of CPSs to describe the different perspectives of CPSs and the potential interactions of devices and systems in a system of systems (SoS) as shown in Fig. 1. From the perspective of unit level, a CPS at least includes one or several controllers, many actuators, and sensors. A CPS can also be a system consisting of one or more cyber-physical devices. From the SoS perspective, a CPS is composed of multiple systems that include multiple devices. In general, a CPS must contain the decision flow (from controller to actuators), information flow (from sensors to controller), and action flow (actuators impacting the physical state of the physical world).

In the context of SCCPS, safety and performance are dependent on the system (to be more specific, the controller of the system) making the right decision according to the measurement of the sensors, and operating the actuators to take the right action at the right time. Thus, verification of the process of decision-making is vital for a SCCPS.

2.2. Modern neural networks

The concept of "*neural network*" was first proposed in 1943 by Warren McCullough and Walter Pitts [24], and Frank Rosenblatt in 1957 designed the first trainable neural network called "*the Perceptron*" [25]. A perceptron is a simple binary classification algorithm with only one layer and output decision of "0" or "1." By the 1980s, neural nets with more than one layer were proposed to solve more complex problems, i.e., multilayer perceptron (MLP). In this SLR, we regard multilayer NNs that emerged after the 1980s as modern NNs.

Artificial Neural Network (ANN) is the general name of computing systems designed to mimic how the human brain processes information [26]. An ANN is composed of a collection of interconnected computation nodes (namely "artificial neurons"), which are organized in layers. Depending on the directions of the signal flow, an ANN can have feed-forward or feedback architectures. Fig. 2 shows a simplified feed-forward ANN architecture with multiple hidden layers. Each artificial neuron has weighted inputs, an activation function, and one output. The weights of the interconnections are adjusted based on the learning rules. There are three main models of learning rules, which are unsupervised learning, supervised learning, and reinforcement learning. The choice of learning rules corresponds to the particular learning task. The common activation functions contain sigmoid, hyperbolic tangent, radial bases function (RBF), and piece-wise linear transfer function, such as Rectified Linear Unit (ReLU) [27]. In a word, an ANN can be defined by three factors: the interconnection structure between different layers, activation function type, and procedure for updating the weights.

Multi-Layer Perceptron (**MLP** [28]) represents a class of feed-forward ANN. An MLP consists of an input layer, one or several hidden layers, and an output layer. Each neuron of MLP in one layer is fully connected with every node in the following layer. An MLP employs a back-propagation technique (which belongs to supervised learning) for training.

Convolutional Neural Network (**CNN** [29]) is a special type of multi-layer NN with one or more convolutional layers. A convolutional layer includes "*several feature maps with different weight vectors. A sequential implementation of a feature map would scan the input image with a single unit that has a local receptive field, and store the states of this unit at corresponding locations in the feature map. This operation is equivalent to a convolution, followed by an additive bias and squashing function, hence the name convolutional network*"[29]. CNNs are superior for processing two-dimensional data (particular camera images) because of the convolution operations, which are capable of detecting features in images. CNNs are now widely applied to develop partially-autonomous and fully-autonomous vehicles.

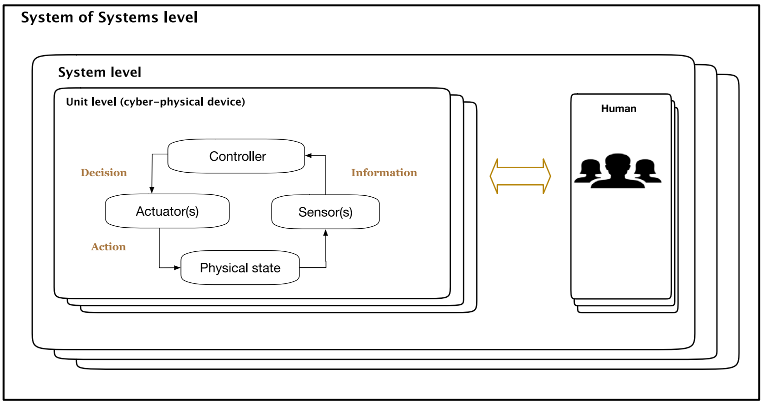


Fig. 1. CPS conceptual model.

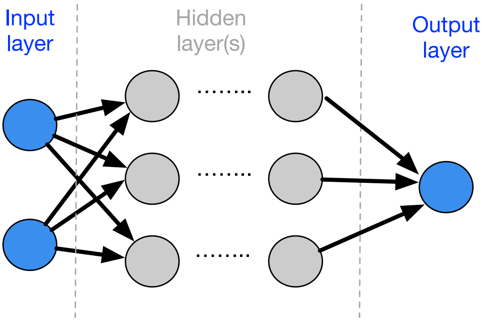


Fig. 2. A simplified feed-forward ANN architecture.

Deep Neural Networks (**DNNs** [30]) represent an ANN with multiple hidden layers between the input and output layers. DNNs (e.g., a MLP with more than three layers or a CNN) differ from shallow NNs (e.g., a three-layer MLP) in the number of layers, the activation functions that can be employed, and the arrangement of the hidden layer. Compared to shallow NNs, DNNs can be trained more in-depth to find patterns with high performance even for complex nonlinear relationships.

An NN could be trained offline or online. An NN trained offline means it only learns during development. After training, the weights of the NN will be fixed and the NN will act deterministically. Therefore static verification methods could be possible. In contrast, online training will allow the NN to keep learning and evolving during operation, which requires run-time verification methods. In some applications, such as the Intelligent Flight Control System developed by NASA [15], both offline and online training strategies are employed to meet the system requirements.

NNs are fundamentally different with algorithmic programs, but a formal development methodology can still be derived for an NN system. Development process of an NN system can include six phases [31]:

1. Formulation of requirements and goals;
2. Selection of training and test data sets;
3. Selection of the NN architecture;
4. Training of the network;
5. Testing of the network; and
6. Acceptance and use by the customer.

Like [31], Falcini et al. introduced a similar development lifecycle for DNNs in automotive software [32] and proposed a W-model integrated data development with standard software development to highlight the importance of data-driven in DNN development. Falcini et al. [32] also summarized that the DNN's functional behavior depends on both its architecture and its learning outcome through training.

2.3. *The trends of using NN algorithm in SCCPSs*

From 1940s automated range finders (developed by Norbert Wiener for anti-aircraft guns) [164] to today's self-driving cars, AI, especially NN algorithms, is widely applied in both civilian (e.g., autonomous cars) and military domains (e.g., military drones). Boosted by the advances of AI, state-of-the-art CPSs can plan and execute more and more complex operations with less human interaction. Here we present the applications of NNs in the following four representative SCCPSs.

2.3.1. *Autonomous cars*

For automobile, the Society of Automotive Engineers (SAE) proposed six levels of autonomous driving [33]. A level 0 vehicle has no autonomous capabilities, and the human driver is responsible for all aspects of the driving task. For level 5 vehicle, the driving tasks are only managed by the autonomous driving system. When developing autonomous vehicles targeting a high level of autonomy, one industry trend is to use DNNs to implement vehicle control algorithms. The deep-learning-based approach enables vehicles to learn meaningful road features from raw input data automatically and then output driving actions. The so-called end-to-end learning approach can be applied to resolve complex real-world driving tasks. When using deep-learning-based approaches, the first step is to use a large number of training data sets (images or other sensor data) to train a DNN. Then a simulator is used to evaluate the performance of the trained network. After that, the DNN-based autonomous vehicle will be able to *"execute recognition, prediction, and planning"* driving tasks in diverse conditions [10]. Nowadays, CNNs are the most widely adopted deep-learning model for fully autonomous vehicles [5–8]. NVIDIA introduced an AI supercomputer for autonomy [34]. The development flow using NVIDIA DRIVE PX includes four stages: 1) data acquisition to train the DNN, 2) deployment of the output of a DNN in a car, 3) autonomous application development, and 4) testing in-vehicle or with simulation.

One essential characteristic of deep-learning-based autonomy is that the decision-making part of the vehicle is almost a black box. This means that in most cases, we as human drivers must trust the decisions made by the deep-learning algorithms without knowing exactly why and how the decisions are made.

2.3.2. Industrial control systems

Industrial Control System (ICS) is the general term for control systems, also called Supervisory Control and Data Acquisition (SCADA) systems. ICSs make decisions based on the specific control law (such as lookup table and non-linear mathematical model) formulated by human designers. In contrast to the classical design procedure of control law, reinforcement-learning-based approaches learn the control law simply from the interaction between the controller and the process, and then incrementally improving control behavior. Such approaches and NNs have been used in process control two decades ago [35]. Concerning the recent progress in AI and the success of DNNs in making complex decisions, there are high expectations for the application of DNNs in ICSs. For instance, DNNs and reinforcement learning can be combined to develop continuous control [36]. Spielberg et al. extended the work in Lillicrap et al. [36] to design control policy for process control [37]. Even though the proposed approach in Spielberg et al. [37] is only tested on linear systems, it shows a practical solution for applying DNNs in non-linear ICSs.

2.3.3. Smart grid systems

The smart grid is designed as the next generation of electric power system, dependent on information communications technology (ICT). There is tremendous initiative of research activities in automated smart grid applications, such as FLISR (which is a smart grid multi-agent automation architecture based on decentralized intelligent decision-making nodes) [38]. NNs have been considered for solving many pattern recognition and optimization problems, such as fault diagnosis [39], and control and estimation of flux, speed [2], and economical electricity distribution to consumers. MLP is one of the most commonly used topology in power electronics and motor drives [2].

2.3.4. Healthcare

Medical devices is another emerging area where research and industry practitioners are seeking to integrate AI technologies to improve accuracy and automation. ANNs and other machine learning approaches have been proposed to improve the control algorithms for diabetes treatment in recent decades [40,41]. In 2017, an AI-powered device for automated and continuous delivery of basal insulin (named MiniMed 670G system [42]) was approved by the U.S. Food and Drug Administration. In the same year, it was reported that GE Healthcare had integrated the NVIDIA AI platform into their computerized tomography scanner to improve speed and accuracy for the detection of liver and kidney lesions [43]. Using deep learning solutions, such as CNNs, in the medical computing field has proven to be effective since CNNs have excellent performance in object recognition and localization in medical images [44].

2.4. Testing and verification of safety-critical control software

IEC 61508 and ISO 26262 are two standards highly relevant to the T&V of SCCS. IEC 61508 is an international standard concerning *Functional safety of electrical/electronic/programmable electronic safety-related systems*. It defines four safety integrity levels (SILs) for safety-critical systems [45]. The higher the SIL level a SCCPS requires, the more time and effort for verification are needed. In IEC 61508, formal methods are highly recommended techniques for verifying high SIL systems. Because formal methods can be used to construct the specification and provide a mathematical proof that the system matches some formal requirements, this is quite a strong commitment for the correctness of a system.

ISO 26262, titled *Road vehicles – functional safety*, is an international standard for the functional safety of electrical and/or electronic systems in production automobiles [46]. Besides using classical safety analysis methods such as Fault Tree Analysis (FTA) and Failure Mode and Effects Analysis (FMEA), ISO 26262 explicitly states that the production of a safety case is mandated to assure system safety. It defines a safety case as “an argument that the safety requirements for an item are complete and satisfied by evidence compiled from work products of the safety activities during development” [46].

The development of suitable approaches, which can verify the system behavior and misbehavior of a SCCPS is always challenging. Not to mention that the architecture of NNs (especially DNNs) makes it even harder to decipher how the algorithmic decisions were made. The current version of IEC 61508 is not applicable for the verification of NN-based SCCSs because AI technologies are not recommended there. The latest version of ISO 26262 and its extension, ISO/PAS 21448, which is also known as safety of the intended functionality (SOTIF) [47], will likely provide a way to handle the development of autonomous vehicles. However, SOTIF will only provide guidelines associated with SAE Level 0–2 autonomous vehicles [48], which are not ready for the verification of NN-based autonomous vehicles.

In practice, in order to reduce test and validation costs, high-fidelity simulation is a commonly used approach in the automotive domain. The purpose of using a simulator is to predict the behavior of an autonomous car in a mimicked environment. NVIDIA and Apollo distributed their high-fidelity simulation platforms for testing autonomous vehicles. CARLA [49] and Udacity’s Self-Driving Car Simulator [50] are two popular open-source simulators for autonomous driving research and testing.

3. Research method

We conducted our SLR by following the SLR guidelines in Kitchenham and Charters [20] as well as consulting other relevant guidelines in Petersen et al. [51] and Shahin et al. [52], Nguyen et al. [53]. Our review protocol consisted of four parts: 1) search strategy, 2) inclusion and exclusion criteria, 3) selection process, and 4) data extraction and synthesis.

3.1. Search strategy

Based on guidelines provided in Kitchenham and Charters [20], we use the Population, Intervention, Outcome, Context (PIOC) criteria to formulate search terms. In this SLR,

- The population should be an application area (e.g., general CPS) or specific CPS (e.g., self-driving car).
- The intervention is methodology, tools and technology that address system/component testing or verification.
- The outcome is the improved safety or functional safety of CPSs.
- The context is the NN-based SCCPSs in which the T&V take place.

Fig. 3 shows the search terms formulated based on the PIOC criteria. We first used these search terms to run a series of trial searches and verify the relevance of the resulting papers. We then revised the search string to form the final search terms. The final search terms were composed of synonyms and related terms.

We executed automated searches in six digital libraries, namely, Scopus, IEEE Xplore, Compendex EI, ACM Digital library, SpringerLink, and Web of Science (ISI).

3.2. Inclusion and exclusion criteria

Table 1 presents our inclusion and exclusion criteria. We set three inclusion criteria to restrict the application domain, context, and outcome type. We excluded papers that were not peer-reviewed, such as keynotes, books, and dissertations, and papers not written in English. It should be clarified that, unlike most other SLR studies, we did not directly exclude short papers (less than six pages), work-in-progress papers, and pre-print papers. The reason is that this research area is far

Population: "Cyber-physical system*" or "Cyber physical system*" or CPS* or "Smart grid" or "Smart car" or "Automotive cyber-physical system*" or "Self-driving car*" or "Autonomous vehicle*" or "Autonomous driving system*" or "Automotive electronic control system*" or "Automotive embedded system*"

Intervention: "Risk assessment" or "verification" or "test" or "testing" or "analysis" or "Certification" or "assurance"

Outcome: "Safety" or "Functional safety"

Context: "Deep learning" or "Deep neural networks" or "Autonomous decision" or "Autonomous agent"

TITLE-ABS-KEY(("Cyber-physical system*" or "Cyber-physical system*" or CPS* or "Smart grid" or "Smart car" or "Automotive cyber-physical system*" or "Self-driving car*" or "Autonomous vehicle*" or "Autonomous driving system*" or "Automotive electronic control system*" or "Automotive embedded system*" or "Unmanned Aerial Vehicles" or "aircraft collision avoidance system*")**AND**("Risk assessment" or "verification" or "test" or "testing" or "analysis" or "Certification" or "assurance")**AND**("Safety" or "Functional safety")**AND**("Autonomous decision" or "Autonomous agent*" or "Deep learning" or "Deep neural networks")

Fig. 3. Search terms.

Table 1
Inclusion and exclusion criteria.

Inclusion criteria	
I1	The paper must have a context in SCCPSs, either in general or in a specific application domain
I2	The paper must be aimed at testing/verification approaches for NN-based SCCSs
I3	The paper must be aimed at modern neural networks
Exclusion criteria	
E1	Papers not peer-reviewed
E2	Not written in English
E3	Full-text is not available
E4	Not relevant to modern neural networks

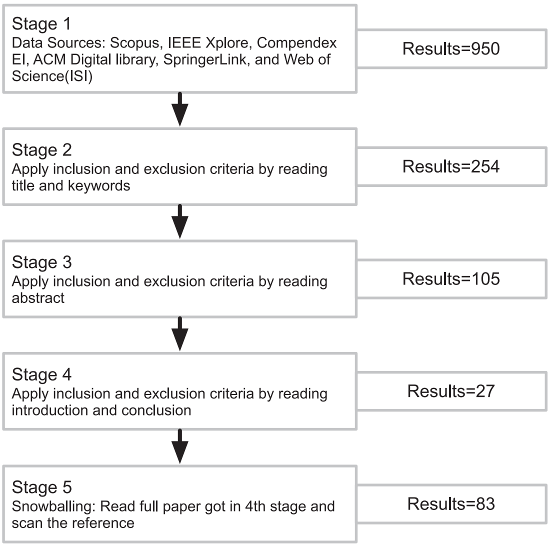

Stage 1
Data Sources: Scopus, IEEE Xplore, Compendex EI, ACM Digital library, SpringerLink, and Web of Science(ISI)
Results=950
Stage 2
Apply inclusion and exclusion criteria by reading title and keywords
Results=254
Stage 3
Apply inclusion and exclusion criteria by reading abstract
Results=105
Stage 4
Apply inclusion and exclusion criteria by reading introduction and conclusion
Results=27
Stage 5
Snowballing: Read full paper got in 4th stage and scan the reference
Results=83

Fig. 4. Search process.

from mature, so many initial thoughts or in-progress papers are still valuable to review.

3.3. Selection process

We used the inclusion and exclusion criteria to filter the papers in the following steps. We covered papers from January 2011 to November 2018. Fig. 4 shows the whole search and filtering process.

Stage 1: Ran the search string on the six digital libraries and retrieved 1046 papers. After removing those duplicated papers, we had 950 papers.

Stage 2: Excluded studies by reading title and keywords. If it was not excluded simply by reading titles and keywords, the paper was kept for further investigation. At the end of this stage, we selected 254 papers.

Stage 3: Further filtered the papers by reading abstracts and found 105 potential papers with high relevance to the research goal of our SLR.

Stage 4: Read the introduction and conclusion to decide on selection. We recorded the reasons for exclusion for each excluded paper. We excluded the papers that were irrelevant, or whose full texts were not available. Furthermore, we critically examined the quality of primary studies to exclude those that lacked sufficient information. We ended up with 27 papers.

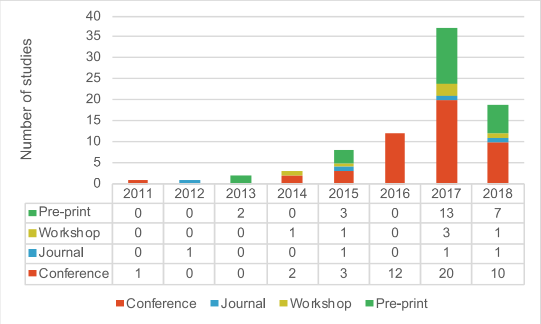
Stage 5: Read full text of the selected studies from the fourth stage, applied snowballing by scanning the reference of the selected papers. The snowballing process can be implemented in two directions: backwards (which means scanning the references of a selected paper and find any other relevant papers published before the selected paper), and forwards (which means checking if any other relevant paper was published after the selected paper and cited the selected paper). In our SLR, we adapted mainly backward snowballing to include additional papers. To limit the scope of the snowballing, we covered only references published between 2011 and 2018. From snowballing, we found 56 new relevant papers.

Finally, we selected 83 papers as primary studies for detailed analysis. We listed all of the selected studies in Appendix A. The first author conducted the selection process with face-to-face discussions with the second author. The second author performed a cross-check of each step and read all the final selected papers to confirm the selection of the papers.

3.4. Data extraction and synthesis

Data Extraction: We extracted two kinds of information from the selected papers. To answer RQ1, we extracted information for statistical analysis, e.g., publication year and research type. To answer RQ2 and RQ3, we collected information to identify key features (such as research goal, technique and tools, major contribution and limitation) of T&V approaches.

Synthesis: We used descriptive statistics to analyze the data for answering RQ1. To answer RQ2 and RQ3, we analyzed the data using the qualitative analysis method by following the five steps of thematic analysis [54]: 1) extracting data, 2) coding data, 3) translating codes into themes, 4) creating a model of higher-order themes, and 5) assessing the trustworthiness of the synthesis.



	2011	2012	2013	2014	2015	2016	2017	2018
Pre-print	0	0	2	0	3	0	13	7
Workshop	0	0	0	1	1	0	3	1
Journal	0	1	0	0	1	0	1	1
Conference	1	0	0	2	3	12	20	10

Fig. 5. Publication year and types of selected papers.

Table 2
Research type classification (T = True, F = False, ● = irrelevant or not applicable, R1–R6 refer to rules).

	R1	R2	R3	R4	R5	R6
Conditions						
Used in practice	T	●	T	F	F	F
Novel solution	●	T	F	●	F	F
Empirical evaluation	T	F	F	T	F	F
Conceptual framework	●	●	●	●	T	F
Opinion about something	F	F	F	F	F	T
Authors' experience	●	●	T	●	F	F
Decisions						
Evaluation research	√	●	●	●	●	●
Solution proposal	●	√	●	●	●	●
Validation research	●	●	●	√	●	●
Philosophical papers	●	●	●	●	√	●
Opinion papers	●	●	●	●	●	√
Experience papers	●	●	√	●	●	●

Note: Reprinted from [51],Copyright 2015 by the Elsevier.

4. Result

4.1. RQ1. What are the profiles of the studies focusing on testing and verifying NN-based SCCSs?

Studies distributions: Fig. 5 shows the distribution of selected papers based on publication year and the types of work. There has been 68 papers (81.9%) published since 2016, indicating that researchers are paying more attention to the T&V of NN-based SCCSs. Conference was the most popular publication type with 48 papers (57.8%), followed by pre-print (25 papers, 30.1%), workshop (6 papers, 7.2%), and journal (4 papers, 4.8%).

We also investigated the geographic distribution of the reviewed studies. It allowed us to identify which countries are leading the research in this domain. We considered a study to be conducted in one country if the affiliation of at least one author is in that country. Moreover, the involvement of industry would be an indicator of industry's interest in this domain. We classified the reviewed papers as industry if at least one author came from industry or the study used real-world industrial systems to test/verify the proposed approach. A paper would be categorized as academia if all authors came from academia. It shows that researchers based in the USA have been involved in the most primary studies for testing or verification of NN-based SCCSs with 56 publications, followed by the researchers based in Germany and the UK with 10 and 9 publications, respectively. It is worth noting that 47 of 83 (56.6%) publications have involvement from industry.

Research types: We classified the selected papers based on the criteria proposed by Kai et al. [51] (See Table 2). According to Table 2, the research type of the paper is governed by rules (i.e., R1-R6). Each rule is a combination of several conditions. The six research types (i.e., evaluation research, solution proposal, validation research, philosophical papers, opinion papers, and experience papers) correspond to R1-R6, respectively. For example, both evaluation research (corresponding to R1) and validation research (corresponding to R4) must present empirical evaluation. The difference between evaluation and validation research is that validation is not used in practice (e.g., experimental or simulation-based approaches), whereas evaluation studies should be conducted in a real-world context. Solution proposal means that it has to propose a new solution that may or may not be used in practice. We found that evaluation and validation research are the majority of the selected papers, corresponding to 31.3% (26 papers) and 61.4% (51 papers) of the selected papers, respectively. The low percentage of the solution proposal (6 papers) was not surprising because a majority of the reviewed papers presented and demonstrated their T&V approaches through academic and industrial case studies, simulation, and controlled experiments. The other three types of research papers (i.e., philosophical papers, opinion papers, and experience papers) do not exist in selected studies because we only included papers that aimed at testing/verification approaches (refer to inclusion criteria I2).

Table 3
Distribution of application domains of the selected studies.

Application domain	No. of studies
General SCCPSs	59
Automotive CPSs	13
Autonomous aerial systems	5
Robot system	5
Health care	1

Application domains: We analyzed the application domain of selected studies to provide useful information for researchers and practitioners who are interested in the domain-specific aspects of the approaches. The results are shown in Table 3. We found that considerable effort is now being put into using NN algorithms to accomplish control logic for general purpose (59 papers, 71.1%), automotive CPSs, such as autonomous vehicles (13 papers, 15.7), and autonomous aerial systems, such as airborne collision avoidance systems for unmanned aircrafts (5 papers, 6%).

4.2. RQ2. What approaches and associated tools have been proposed to test and verify NN-based SCCSs?

As 4 of the 83 papers focused mainly on high-level ideas and concepts without presenting detailed approaches or tools, we did not include them to answer RQ2. For the remaining 79 out of 83 (95.2%) papers, we applied the thematic analysis approach [54] and identified five high-order themes and some sub-themes. Some papers contain more than one themes. In order to balance the accuracy and the simplicity of categorization, we decided to assign each study only one category based on its major contribution. Table 4 presents the themes, sub-themes, and the corresponding papers. Fig. 6 compares the interests difference of academia and industry for the five identified themes.

4.2.1. CA1: Assuring robustness of NNs

One high-order theme of the studies is to assure the robustness of NNs. Robustness of an NN is its ability to cope with erroneous inputs. The erroneous inputs can be an adversarial example (i.e., an input that adds small perturbation intentionally to mislead classification of an NN), or benign but wrong input data. Methods under this theme can be further classified into four sub-themes.

Studies focusing on understanding the characteristics and impacts of adversarial examples Some studies tried to identify the characteristics and impacts of adversarial examples. The study [56] found the characteristics, such as the linear nature, of adversarial examples. The study

Table 4
A classification of approaches to test and verify NN-based SCCSs.

Themes	Sub-themes	Papers	#
Assuring robustness of NNs	Understanding the characteristics and impacts of adversarial examples	[55–61]	17
	Detect adversarial examples	[62–67]	
	Mitigate impact of adversarial examples	[68,69]	
	Improving robustness of NNs through using adversarial examples	[70,71]	
Improving failure resilience of NNs	[72–82]		11
Measuring and ensuring test completeness	[83–89]		7
Assuring safety properties of NN-based CPSs	[90–102]		13
Improving interpretability of NNs	Understand how a specific decision is made	[103–121]	31
	Facilitate understanding of the internal logic of NNs	[122–127]	
	Visualizing internal layers of NNs to help identify errors in NNs	[128–133]	

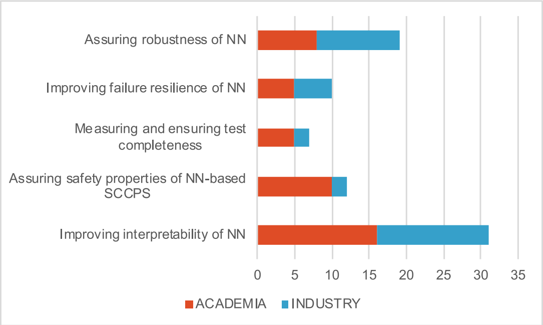


Fig. 6. Comparing the interests difference of academia and industry.

[58] measured the impact of adversarial examples by counting their frequencies and severities. Nguyen et al. [55] found that a CNN trained on ImageNet [134] is vulnerable to adversarial examples generated through Evolutionary Algorithms (EAs) or gradient ascent.

A few other studies, such as [57,59–61], tried to understand the characteristics of robust NNs. Cisse et al. [59] introduced a particular form of DNN, namely Parseval Networks, that is intrinsically robust to adversarial noise. Gu et al. [61] concluded that some training strategies, for example, training using adversarial examples or imposing contractive penalty layer by layer, are robust to certain structures of adversarial examples (e.g., inputs corrupted by Gaussian additive noises or blurring). Higher-confidence adversarial examples (i.e., adversarial instances that are extremely easy to classify into the wrong category) were used to evaluate the robustness of the state-of-the-art NN in Carlini and Wagner [60] and the robot-vision system in Melis et al. [57].

Studies focusing on methods to detect adversarial examples Detecting adversarial examples that are already inserted into training or testing data set are the primary targets of [62,64–67]. Wicker et al. [62,66] formulated the adversarial examples detection as a two-player stochastic game and used the Monte Carlo Tree Search to identify adversarial examples. Reuben [64] applied density estimates, and Bayesian uncertainty estimates to detect adversarial samples. Xu et al. [65] proposed a feature squeezing framework to detect adversarial examples, which are generated by seven state-of-the-art methods. According to [65], an advantage of feature squeezing is that it did not change the underlying model. Therefore, it can easily be integrated with other defenses methods. Metzen et al. [67] embedded DNNs with a subnetwork (called “detector”) to detect adversarial perturbations. The Deepsafe presented in Gopinath et al. [63] used clustering technology to find candidate-safe regions first and then verified whether the candidates were safe using counter-examples as a proof.

Studies focusing on methods to mitigate impact of adversarial examples Papemot et al. [68] adopted defensive distillation as a defense strategy to train DNN-based classifiers against adversarial examples. However, several powerful attacks have been proposed to defeat defensive distillation and have demonstrated that defensive distillation does not actually eliminate adversarial examples [60]. Papemot et al. [69] revisited defensive distillation and proposed a more effective way to defend against three recently discovered attack strategies, i.e., the Fast Gradient Method (FGM) [56], the Jacobian Saliency Map Approach (JSMA) [135], and the AdaDelta optimization strategy (AdaDelta) [60].

Studies focusing on increasing robustness of NNs through using adversarial examples. In studies [70,71], the authors proposed methods to leverage adversarial training (e.g., generating a large amount of adversarial examples and then training the NN not to be fooled by these adversarial examples) to increase the robustness of NNs.

4.2.2. CA2: Improving failure resilience of NNs

Studies under this theme focused on improving the resilience of NNs, so that the NN-based CPSs are more tolerant of possible hardware and software failures.

Studies [74,76,77] investigated error detection and mitigation mechanisms, while studies [75,79] focused on understanding error propagation in DNN accelerators. Vialatte et al.[74] demonstrated that faulty computations can be addressed by increasing the size of NNs. Santos et al. [76] proposed an algorithm-based fault tolerance (ABFT) strategy to detect and correct radiation-induced errors. In [77], a binary classification algorithm based on temporal and stereo inconsistencies was applied to identify errors caused by single frame object detectors. Li et al. [75] developed a general-purpose GPU (GPGPU) fault injection tool [136] to investigate error propagation patterns in twelve GPGPU applications. Later, Li et al. revealed that the error resilience of DNN accelerators depends on “*the data types, values, data reuse, and the types of layers in the design [80]*”. Based on this finding, they devised guidelines for designing resilient DNN systems and proposed two DNN protection techniques, namely Symptom-based Error Detectors (SED) and Selective Latch Hardening (SLH) to mitigate soft errors that are typically caused by high-energy particles in hardware systems [137].

Mhamdi et al. explored error propagation mechanism in an NN [78], and they theoretically and empirically proved that the key parameters that can be used to estimate the robustness of an NN are: *“Lipschitz coefficient of the activation function, distribution of large synaptic weights, and depth of the network”*. The study [80] characterized the faults propagation through an open-source autonomous vehicle control software (i.e., openpilot) to assess the failure resilience of the system. The Systems-Theoretic Process Analysis (STPA) [138] hazard analysis technique was used to guide fault injection. Existing work in Rubaiyat et al.

[80] showed that STPA is suited for an in-depth identification of unsafe scenarios, and thus, the fault injection space was reduced.

Based on the diversified redundancy strategies, the study [81] developed diverse networks in the context of different training data sets, different network parameters, and different classification mechanisms to strengthen the fault tolerance of the DNN architecture.

Studies [72,73] tried to improve computation efficiency without compromising error resilience. Studies [72,73] also predicted the error resilience of DNN accelerators to make reconfigurable NN accelerators. The study [72] demonstrated a more accurate neuron resilience assignment than the state-of-the-art techniques and provided the possibility of moving parts of the neuron computations to unreliable hardware at the given quality constraint. Zhang et al. [73] proposed a framework to increase efficiency of computation by approximating the computation of certain less critical neurons. Daftry et al. [82] provided an interesting idea about "how to enable a robot to know when it does not know?" The idea of [82] is to utilize the resulting features of the controller, which are learned from a CNN to predict the failure of the controller, and then let the system self-evaluate and decide whether to execute or discard an action.

4.2.3. CA3: Measuring and ensuring test completeness

The approaches and tools under this theme aim to ensure good coverage when testing NNs. The testing approaches include black-box testing (i.e., focusing on whether the tests cover all possible usage scenarios), white-box testing (i.e., focusing on whether the tests cover every neuron in the NN), and metamorphic testing, which focuses on both test case generation and result verification [139].

O'Kelly et al.[83] proposed methods to ensure good usage coverage through first making a formal Scenario Description Language (SDL) to create driving scenarios, and then translating the scenarios to a specification-guided automatic test generation tool named S-TALIRO to generate and run the tests. Raj et al. [86] proved the possibility of speeding up the generation of new and interesting counterexamples by introducing fuzzing patterns obtained from an unrelated DNN on a different image database, although the proposed method provides no guarantee of test completeness.

DeepXplore [84] first introduced neuron coverage as a testing metric for DNNs, and then used multiple different DNNs with similar functionality to identify erroneous corner cases. Compared to [84], DeepTest [85] and DLFuzz [89] aimed at maximizing the neuron coverage without requiring multiple DNNs. The study [85] employed metamorphic relations to identify erroneous behaviors. The study [89] proposed a differential fuzzing testing framework to generate adversarial inputs. However, methods proposed in Pei et al. [84], Tian et al. [85], Guo et al. [89] cannot guarantee the generation of test cases that can precisely reflect real-world cases (e.g., driving scenes in various weather conditions when taking a DNN-based autonomous driving system). DeepRoad [88] employed Generative Adversarial Network (GAN) based techniques and metamorphic testing to synthesize diverse real driving scenes, and to test inconsistent behaviors in DNN-based autonomous driving systems. In contrast to earlier works, DeepGauge [87] argued that the testing criteria for traditional software are no longer applicable for DNNs. Ma et al. [87] proposed neuron-level and layer-level coverage criteria for testing DNNs and for measuring the testing quality.

4.2.4. CA4: Assuring safety property of NN-based SCCPSs

Formal verification can provide a mathematical proof that a system satisfies some desired safety properties (e.g., the system should always stay within some allowed region, namely a safe region). Formal verification usually presents NNs as models and then apply a model checker, such as Boolean satisfiability (SAT) solvers (e.g., Chaff [140], SATO [141], GRASP [142]) to verify the safety property. Pulina et al. [92] developed NeVer ("Ne"ural networks "Ver"ifier), which solves Boolean combinations of linear arithmetic constraints, to verify safe regions of MLPs. Through adopting an abstraction-refinement mechanism, NeVer can verify real-world MLPs automatically. As an extended experiment analysis of results of [92], Pulina and Tacchella [90] compared the performance (e.g., competition-style and scalability) of state-of-the-art Satisfiability Modulo Theories (SMT) solvers [143], and demonstrated that scalability and fine-grained abstractions remain challenges for realistic size networks. The studies [91,97] verified the *"feed-forward NNs with piece-wise linear activation functions"* by encoding verification problems into solving a linear approximation exploring network behavior in a SMT solver.

The next generation of collision avoidance systems for unmanned aircrafts (ACAS Xu) adopted DNNs to compress large score table [5]. Julian et al. [95] explored the performance of ACAS Xu by measuring a set of safety and performance metrics. A simulation in study [95] shows that the system based on DNNs performed as correctly as the original large score table but with better performance. Reluplex [97] had successfully been used to verify the safety property of a DNN for the prototype of ACAS Xu. Although the outcomes of Reluplex [97] are limited to verifying the correctness of NNs with specific type of activation functions (i.e., ReLUs and max-pooling layers), the study sheds a light on which types of NN architectures are easier to verify, and thus paves the way for verifying real-world DNN-based controllers.

The method proposed in studies [99,100] verified that Binarized Neural Networks (BNNs) are efficient and scalable to moderate-sized BNNs. Study [99] represented BNNs as boolean formulas, and then verified the robustness of BNNs against adversarial perturbations. In study [100], BNNs and their input-output specifications were transferred into equivalence hardware circuits. The equivalence hardware circuits consist of a BNN structure module and a BNN property module. The authors of [100] then applied a SAT solver to verify the properties (e.g., "simultaneously classify an image as a priority road sign and as a stop sign with high confidence") of the BNN in order to identify the risk behavior of the BNN.

When verifying a SCCS, one of the fundamental concerns is to make sure that the SCCS will never violate a safety property. An example of a safety property is that the system should never reach an unsafe region. The main ideas of studies under this sub-theme are to calculate the output reachable set of MLPs, such as in studies [94,96], or DNNs in study [93], to verify if unsafe regions will be reached. Xiang et al. [96] proposed a layer-by-layer approach to compute the output reachable set assisted by polyhedron computation tools. The safety verification of a ReLU MLP is turned into checking if a non-empty intersection exists between the output reachable set and the unsafe regions. In a later work of Xiang et al. [94], they introduced maximum sensitivity to perform a simulation-based reachable set estimation with few restrictions on the activation functions. By combining local search and linear programming problems, Dutta et al. [93] developed an output bound searching approach for DNNs with ReLU activation functions, which is implemented in a tool called SHERLOCK to check whether the unsafe region is reached. Study [98] focused on the safety verification of image classification decisions. In [98], Huang et al. employed discretization to enable a finite exhaustive search for adversarial misclassifications. If no misclassifications are found in all layers after the exhaustive search, the NN is regarded as safe.

The idea of [101] was to formulate the formal verification of temporal logic properties of a CPS with Machine Learning (ML) components as the falsification problem (finding a counterexample that does not satisfy system specification). The study [101] adopted an ML analyzer to abstract the feature space of ML components (which approximately represents the ML classifiers). The identified misclassifying features are then used to drive the process of falsification. The introduction of the ML analyzer narrowed down the searching space for counterexamples and established a connection between the ML component and the rest of the system.

Another direction to make sure the system will not violate safety properties is to use run-time monitoring. The study [102] envisioned an approach named WISEML, which combines reinforcement learning and run-time monitoring technique, to detect invariants violations. The purpose of this work was to create a safety envelope around the NN-based SCCPSs.

4.2.5. CA5: Improving interpretability of NNs

NNs have proved to be effective ways to generalize the relationship between inputs and outputs. As the models of NNs are learned from training data sets without human intervention, the relationship between the inputs and outputs of NNs is like a black box. Due to the black-box nature of NNs, it is difficult for people to understand and explain how an NN works. Studies under this theme focus on facilitating the understanding on how NNs generate outputs from inputs. Studies in this theme can be classified into the following three sub-themes, which can be overlapped. However, this can be a way to capture the different motivations for the interpretability of NNs.

Studies focusing on understanding how a specific decision is made This line of work mainly focuses on providing explanations for individual predictions (also defined as local interpretability). One study is called Local Interpretable Model-agnostic Explanations (LIME) [129]. LIME can approximate the original NN model locally to provide an explanation for a specific prediction of interest. The problem of LIME is that it assumes the local linearity of the classification boundary, which is not true for most complex NNs. The creators of LIME later extended their work by introducing high-precision rules (i.e., if-then rules), which they called *anchors* [104]. The study [130] developed an explanation system named LEMNA for security applications and Recurrent Neural Networks (RNNs). LEMNA can locally approximate a non-linear classification boundary and handle feature dependency problems and therefore is able to provide a high fidelity explanation.

In the case of an image classifier, it is also common to use gradient measurements to estimate the importance value of each pixel for the final classification. DeepLIFT [115], Integrated Gradients [105], and more recently, SmoothGrad [120] fall into this category. The study [121] proposed a unified framework, SHapley Additive exPlanations (SHAP), by integrating six existing methods (LIME [122], DeepLIFT [115], Layer-Wise Relevance Propagation, Shapley regression values, Shapley sampling values, and Quantitative Input Influence) to measure feature importance.

Several approaches attempted to decompose the classification decision (output) into the contributions of individual components of an input based on specific local decomposition rules (i.e., Pixel-Wise decomposition [106,116], and deep Taylor decomposition [108]).

Szegedy et al. [103] investigated the semantic meaning of individual units and the stability of DNNs while small perturbations were added to the input. They pointed out that the individual neurons did not contain the semantic information, while the entire space of activations does. They also experimentally proved that the same small perturbation of input can cause different DNN models (e.g., trained with different hyperparameters) to generate wrong predictions.

There are several methods for improving local explanations for NN models compared to the above-mentioned approaches. The study [113] argued that explanation approaches for NN models should provide sound theoretical support. Ross et al. [118] presented their idea as *"Right for the right reasons,"* which means that the output of NN models should be right with the right explanation. In Ross et al. [118], incorrect explanations for particular inputs can be identified, and NN models can be guided to learn alternate explanations. Both [113,117] made efforts on real-time explanations since their approaches can generate accurate explanations quickly enough.

Studies focusing on facilitating understanding of the internal logic of NNs. Studies in this sub-theme are also known as global interpretability. To help interpret how NN models work, model distillation is used in Frosst and Hinton [122], Che et al. [123], Hinton et al. [124], Tan et al. [126]. The initial intention of distillation was to reduce the computational cost. For example, Hinton et al. [124] distilled a collection of DNN models into a single model to facilitate deployment. The knowledge distilled from NN models has later been applied for interpretability. Some studies compressed information (e.g., decision rules) from deep learning models into transparent models such as decision trees [122,131] and gradient boosting trees [123] to mimic the performance of models. Others tended to explain the inner mechanisms of NN models through analyzing the feature space. Study [126] distilled the relationship between input features and model predictions (outputs of the model) as a feature shape to evaluate the feature contribution to the model.

Another attempt to produce global interpretability is to reveal the features learned by each neuron. For example, in Nguyen et al. [127], the authors leveraged deep generator networks to synthesized the input (i.e., image) that highly activates a neuron. Dong et al. [110] adopted an attentive encoder-decoder network to learn interpretable features, and then proposed an algorithm called *prediction difference maximization* to interpret the features learned by each neuron.

One interesting work [119] used an additional NN module that is fit for relational reasoning to reason the relations between the input and response of the NN models. There is also another promising line of work (e.g., [109,114]) that combined local and global interpretability to explain NN models.

Studies focusing on visualizing internal layers of NNs to help identify errors in NNs In study [128], activities, such as the operation of the classifier and the function of intermesdiate feature layers within the CNN model, were visualized by using a multi-layered deconvolutional network (named DeconvNet). These visualizations are useful to interpret model problems. Unlike [128], which visually depicted neurons in a convolutional layer, the study [107] visualized neurons in a fully connected layer. Zhou et al. [112] proposed *Class Activation Mapping (CAM)* for CNNs to visualize the discriminative object parts on any given image. Fong and Vedaldi [111] highlighted the most responsible part of an image for a decision by perturbing meaningful images. DarkSight [125] combined the ideas of model distillation and visualization to visualize the prediction of an NN model. Thiagarajan et al. [132] built a *TreeView* representation via feature-space partitioning to interpret the prediction of an NN. Mahendran et al. [133] reconstructed semantic information (images) in each layer of CNNs by using information from the image representation.

4.3. RQ3. What are the limitations of current research with respect to testing and verifying NN-based SCCSs?

Analyzing failure modes and how the system reacts to failures are crucial parts of the safety analysis, especially in safety-critical domains. When testing and verifying the safety of NN-based SCCPSs, we need to rethink how to perform failure mode and effect analysis, how to analyze inter-dependencies between sub-systems of SCCPSs, and how to analyze the resilience of the system. We need to ensure that even if some of the system's hardware or software do not behave as expected, the system can sense the risk, avoid the risk before the incident, and mitigate the risk effectively when an incident happens. Looking into T&V activities through software development, the ideal situation is that we would find appropriate T&V methods to verify whether the design and implementation are consistent with the requirements, construct complete test criteria and test oracle, and generate test data and test any objects (such as code modules, data structures) that are necessary for the correct development of software [144]. Unfortunately, the fact is that complete T&V is hard to guarantee. In order to investigate the gap between industry needs for T&V of NN-based SCCPS and state-of-the-art T&V methods, we performed a mapping of identified approaches to the relevant standard.

4.3.1. Mapping of reviewed approaches to the software safety lifecycles in IEC 61508

An increased interest in the application of NNs within safety-critical domains has encouraged research in the area of T&V of NN-based SCCSs. Research institutions and industry T&V practitioners are working on different aspects of this problem. However, we have not found strong

Table 5
A mapping of reviewed approaches to IEC 61508 safety lifecycle.

Phase	Property	Relevant primary studies	Category	Remaining challenges
Software architecture design	Completeness	None		N/A
	Correctness	[95]	CA4	Training process of NN-based algorithm is time-consuming.
	Freedom from intrinsic faults	[56,58,59,61,65,67–71]	CA1	❶ Limited to specific model classes, or tasks (e.g., image classifier), or small size NNs [58]; ❷ Not immune to adversarial adaptation [65]; ❸ Lack of understanding on how system can be free from different kinds of attacks other than adversarial examples.
	Understand- ability	[103–133]	CA5	❶ Limited to specific model classes, or tasks (e.g., image classifier), or small size NN models [122]; ❷ Not able to provide real-time explanations; ❸ Lack of evaluation method for the explanation of NNs.
	Verifiable and testable design	[83]	CA3	❶ Lack of integrated computer- aided toolchains to support the verification activities; ❷ Limited to specific models, tasks or NN size.
		[91]	CA4	❶ Limited to specific NN architectures (i.e., piece-wise linear activation functions), need better understanding of NN architectures; ❷ Trade-off between efficient verification and linear approximation of the NN behavior is not studied sufficiently.
	Fault tolerance	[73,74,78,81,82]	CA2	❶ Decouple the fault tolerance from the classification performance [74]; ❷ Lack of studies on unexpected environmental failures.
	Defense against common cause failure	None		N/A
Software module testing and integration	Completeness	[60,71]	CA1	Lack of comprehensive criteria to evaluate testing adequacy.
		[84–89]	CA3	Low fidelity of testing cases compared with real-world cases [85].
	Correctness	[55,57,60,62–64,66]	CA1	❶ Vulnerable to the variation of adversarial examples; ❷ Limited to specific NN model classes or tasks.
		[77]	CA2	Insufficient validation of input raw data.
	Repeatability	[83–85]	CA3	Testing cases generated by automated tools may be biased.
	Precisely defined testing configuration	None		N/A
Programmable electronics integration (hardware and software)	Completeness	None		N/A
	Correctness	[72,75,76,79]	CA2	Insufficient testing of hardware accelerator.
	Repeatability	None		N/A
	Precisely defined testing configuration	None		N/A
Software verification	Completeness	[94,96]	CA4	❶ Limited to specific NN models; ❷ Lack of scalability.
	Correctness	[80]	CA2	❶ Automatic generation of complete testing scenarios sets.
		[90,92,93,97–101]	CA4	❶ Scalability and computational performance need to improve; ❷ SMT encoding for large-scale NN model; ❸ Lack of model-agnostic verification methods; ❹ Automatic generation of feature space abstractions [101].
	Repeatability	None		N/A
	Precisely defined testing configuration	None		N/A

connections between those potentially useful methods for T&V of NNs and relevant safety standards (such as IEC 61508 [45] and ISO 26262 [46]).

We hereby adopt IEC 61508 [45] as a reference standard to execute the mapping analysis since ISO 26262 [46] is the adaptation of IEC 61508 [45]. We found that the major T&V activities listed in the software safety lifecycles of IEC 61508-3 (including evaluation of software architecture design, software module testing and integration, programmable electronics integration, and software verification) are still valid when conducting T&V for NN-based SCCSs. But for most of them, new techniques/measures for supporting the T&V of NN-based software are demanded. Therefore, we decided to employ safety integrity properties (which are explained in IEC 61508-3 Annex C and Annex F of IEC 61508-7) as indicators to justify to what extent these desirable properties have been achieved by the state-of-the-art methods for T&V of NN-based SCCSs. The detailed mapping information can be found in Table 5.

In Table 5, we mapped existing T&V methods for NN-based SCCSs (column 3 and column 4) into relevant properties (column 2) of four major T&V phases (column 1) in the software safety lifecycles of IEC 61508-3. For column 5 in Table 5, we summarized the remaining challenges in testing and verifying NN-based SCCSs based on reviewed papers. The overviews of these remaining challenges can potentially inspire researchers to look for a focus in the future.

4.3.2. Limitations and suggestions for testing and verifying NN-based SCCSs

In Table 5, we show the limitations and gaps of state-of-the-art T&V approaches for NN-based SCCSs. In this section, we will take two T&V phases (evaluation of software architecture design and software module testing and integration) as examples to provide detailed analysis of identified limitations and corresponding suggestions on the basis of required safety integrity properties. For the other two T&V phases (programmable electronics integration and software verification), only summaries of limitations and suggestions will be presented to avoid duplication.

4.3.2.1. Evaluation of software architecture design. The top three properties that have been addressed are: *simplicity and understandability* (31 papers), *freedom from intrinsic design faults* (10 papers), and *fault tolerance* (5 papers). *Correctness with respect to software safety requirements specification* (1 paper) and *verifiable and testable design* have drawn little attention (2 papers) for reviewed studies. There are two properties, i.e., *completeness with respect to software safety requirements specification* and *Defense against common cause failure from external events*, which have not been addressed in reviewed papers.

4.3.2.1.1. Completeness with respect to software safety requirements specification. No study contributes to the achievement of completeness, which requires the architecture design to be able to address all the safety needs and constraints. The achievement of completeness depends on the achievement of other properties, such as fully understanding the behavior of NN models. The design and deployment of NN-based SCCSs are in its infancy stage. When NN-based SCCS design becomes more practical, more studies may address this topic.

4.3.2.1.2. Correctness with respect to software safety requirements specification. To achieve correctness, software architecture design needs to respond to the specified software safety requirements appropriately. Study [95] reported their successful design of a DNN-based compression algorithm for aircraft collision avoidance systems. Even though they demonstrated that the DNN-based algorithm preserves the required safety performance, the training process is still time-consuming.

4.3.2.1.3. Freedom from intrinsic design faults. Intrinsic design faults can be interpreted as failures derived from the design itself. State-of-the-art NNs have proved to be vulnerable to adversarial perturbations due to some intriguing properties of NNs [56]. Most of the studies in this category were aimed at understanding, detecting, and mitigating adversarial examples. Study [98] reported that their approach could generalize well on several state-of-the-art NNs to find adversarial examples successfully. However, the verification process of founded features is time-consuming, especially for larger images. In this sense, the scalability and computational performance of adversarial robustness are expected to be addressed in the future. In addition, adversarial robustness does not imply that the NN model is truly free from intrinsic design faults. How to assure freedom from interferences (e.g., signal-noise ratio degradation) other than adversarial perturbations is a research gap that needs to be filled.

4.3.2.1.4. Understandability. This property can be interpreted as the predictability of system behavior, even in erroneous and failure situations. In this category, studies focusing on providing explanations for individual prediction (e.g., [103]) and on visualizing internal layers of NN (e.g., [128–130]) are not meaningful for safety assurance. Studies focusing on facilitating understanding of the internal logic of NNs (such as presenting NNs as decision trees [122]) could be a solution to improve the understandability of NN-based architecture design. However, this line of work is rare, and most methods are only applied to small-scale DNNs with image input, or specific NN models. Besides, assuming the explanation of NN is available, confirming the correctness of the explanation is still a challenge. Interpretability of NNs is undoubtedly a crucial need in safety-critical applications. Methods in this line should capable of explaining different types of sensor data (e.g., image, text, and point data) and both local and global decisions.

4.3.2.1.5. Verifiable and testable design. The evaluation metrics of verifiable and testable design may be derived from modularity, simplicity, provability, and so on. We observed that existing verifiable and testable designs are limited to specific NN architectures (e.g., [91]) or specific tasks (e.g., [83]). There is no standard procedure for determining which type of NNs will be easier to verify. Ehlers [91] argued that NNs that adopt piece-wise linear activation functions are easier to verify, but their method still need to face the conflict between efficient verification and accuracy of linear approximation for the NN behavior.

4.3.2.1.6. Fault tolerance. Fault tolerance implies that the architecture design can assure the safe behavior of the software whenever internal or external errors occur. To achieve fault tolerance, features like failure detection and failure impact mitigation of both internal and external errors should be included in the design. Existing methods showed that unexpected environmental failures are hard to detect and mitigate. Besides, many of the proposed approaches in this category have not yet been evaluated in the real-world. Some studies formulated approximated computational models to represent real-world systems (e.g., [73]). The study [82] did not use any test oracle when executing system flight tests. Some studies used simulation models to verify the performance of the original NN (e.g., [74]). They are not able to prove the fidelity of the model compared with the real-world system.

4.3.2.1.7. Defense against common cause failure from external events. Software common cause failure is a type of concurrent failure of two or more modules in the software, which is caused by software design defects and triggered by external events such as time, unexpected input data, or hardware abnormalities [145]. Many safety critical systems adopt redundant architectures (meaning two or more independent subsystems have identical functions to back-up each other) to prevent a single point of failure. However, redundant architectures are vulnerable considering common cause failure. In the context of NN-based SCCSs, it is common to employ multiple NNs with similar architectures in order to improve the accuracy of prediction. If a common cause failure occurs in this kind of software design, the prediction might be totally wrong, and thus the control software might make the wrong decision. DeepXplore, reported in Pei et al. [84], used more than two different DNNs with the same functionality to automatically generate a test case. If all the DNNs in DeepXplore are affected by common cause failure, such as if a sensor failure causes all the DNNs to make the same misclassification, then it will not be able to generate the corresponding test case. No method is found in reviewed papers that can identify common cause failure modes and defend against such failures. In order to effectively defend against common cause failure, designers need to inspect the completeness and correctness of the safety requirements specification, trace the implementation of the safety requirements specification, and make a thorough T&V plan to reveal the common cause failure modes in the early stage.

4.3.2.2. Software module testing and integration. The top two properties that have been addressed are: *completeness of testing and integration with respect to the design specifications* (9 papers) and *correctness of testing and integration with respect to the design specifications* (8 papers). *Repeatability* has drawn little attention (3 papers) from the reviewed studies. There is one property, *precisely defined testing configuration*, which has not been addressed in the reviewed papers. This property aims to evaluate the precision of T&V procedures, which is not in the scope of our selected papers. Therefore, we will not give more explanation on this property.

4.3.2.2.1. Completeness of testing and integration with respect to the design specifications. We observed some efforts that tried to find a systematic way to generate testing cases (e.g., [85,88]) to measure testing quality (e.g., [87]) or to connect different T&V stages in the development of SCCSs (e.g., [146]). As analyzed in Section 4.2, we can infer that an NN-based control software is instinctually different in design workflow and software development compared to the design of traditional control software. We suggest that the testing criteria should thoroughly align with the software design. To be more specific, the instinctive features of NN-based softwares (e.g., NN model's architectural details and the working mechanism of NNs) should be carefully considered when setting the testing criteria. That is testing criteria should be defined comprehensively and explicitly under the consideration of not only test case coverage but also the robustness of NN-based system performance (for instance, test how an NN will respond when input data change slightly) and the features of training data sets, such as the data density issue mentioned in Ashmore and Hill [147].

4.3.2.2.2. Correctness of testing and integration with respect to the design specifications. Several studies (e.g., [55,62,63]) reported that their methods are vulnerable to the variation of adversarial examples. Another common limitation is that most methods are model-specific, meaning that they can only apply to a single type or class of NN model. To achieve correctness of testing and integration, the module testing task should be completed, which means the testing should cover both NN models and external input. However, few studies focused on the validation of input data. One study [77] identified that sufficient validation of input raw data remains a challenge.

4.3.2.2.3. Repeatability. The complexity and un-interpretable feature of NNs make manual testing almost infeasible. In order to be able to generate consistent results from testing repeatedly, some studies were dedicated to achieving automatic test execution or even automatic test generation. We found three papers (i.e., [83–85]) addressing automatic test generation. However, generating test cases automatically is still a challenge. For instance, studies [84,85] claimed that the test cases generated by an automated testing tool may not cover all real-world cases.

4.3.2.3. Programmable electronics integration. The major limitation of this line of work is insufficient testing for hardware accelerators. NN-based SCCPSs requires typically high-performance computing systems, such as Graphics Processing Units (GPUs). Some industry participants have provided specialized hardware accelerators to accelerate NN-based computations. For example, Google deployed a DNN accelerator (called Tensor Processing Unit) in its data centers for DNN applications [148]. NVIDIA introduced an automotive supercomputing platform named DRIVE PX 2 [34], which now has been used by over 370 companies and research institutions in the automotive industry [149]. However, little research effort has been put into the T&V of the reliability of using hardware accelerators for NN applications. We found seven studies (i.e. [72–77,79]) addressing the evaluation of the error resilience of hardware accelerators. However, the testing is limited to specific type errors (e.g., radiation-induced soft errors, which are presented in Schorn et al. [72], Santos et al. [76], Li et al. [79]). The mitigation method proposed in Santos et al. [76] (called ABFT: Algorithm-Based Fault Tolerance) can only protect portions of the accelerator (e.g., sgemm kernels, which is one kind of matrix multiplication kernels). The study [77] identified errors made by single frame object detectors, but the result showed that the method is not capable of detecting all mistakes. The studies [72,79] investigated the propagation characteristic of soft errors in the DNN system, but they used a DNN simulator instead of a real DNN accelerator for fault injection.

4.3.2.4. Software verification. In general, there is a lack of a comprehensive and standardized framework for verifying the safety of NN-based SCCSs. Formal verification procedures are highly demanding. The common limitation of formal verification approaches is the scalability issues. Most proposed methods are limited to a specific NN structure and size (e.g., [91,92,97,99,100]). The study [92] reported that their approaches can only verify small-scale systems (i.e., the layer of NN is 3 and the maximum amount of input neurons is 64). One approach reported in Narodytska et al. [99] can verify medium size NNs. The verification of large-scale NNs is still a challenge. Another limitation is that proposed approaches are not robust to NN variations. For example, verification methods in studies [91,97] are only adapted to specific network types and sizes.

5. Discussion

In this section, we first discuss industry practices for T&V of NN-based SCCPSs. Then, we compare this SLR with related works. At the end of this section, we present the threats to the validity of our study.

5.1. Industry practice

Our findings on the research questions (RQ1 to RQ3) mainly reflected the academic efforts addressing T&V of NN-based SCCPSs. NN-based applications have drawn a lot of attention from industry practitioners. Taking the automotive industry as an example, several car makers (e.g., GM, BMW, and Tesla) and some high technology companies (e.g., Waymo and Baidu) are leading the revolution in autonomous driving safety.

5.1.1. Safety of the intended functionality

At the beginning of this year, ISO/PAS 21448:2019 [47] was published. It listed recommended methods for deriving verification and validation activities (See ISO/PAS 21448:2019 Table 4). In Table 6, we highlighted six of the recommended methods, which shared similar verification interests with existing academic efforts.

5.1.2. Safety reports

In 2018, three companies (Waymo, General Motor, and Baidu Apollo) published their annual safety reports. As a pioneer in the development of self-driving cars, Waymo proposed the "*Safety by Design*" [150] approach, which entails the processes and techniques they used to face safety challenges of a level 4 autonomous car on the road. For the cybersecurity consideration, Waymo adopted Google's security framework [151] as the foundation. After that, General Motor (GM) released their safety report [152] for Cruise AV (also level 4). GM's safety process combined conventional system validation (such as vehicle performance tests, fault injection testing, intrusive testing, and simulation-based software validation) with SOTIF validation through iterative design. Baidu adopted the Responsibility-Sensitive Safety model [153] proposed by Mobileye [154] (an Intel company) to design the safety process for the Apollo Pilot for a passenger car (level 3).

In addition, we noticed that Tesla started releasing quarterly safety data since October 2018 [155]. It seemed that Tesla has a completely different approach to self-driving cars than other companies. According to TESLA NEWS [156], AutoPilot will rely for its self-driving function on cameras, not on LIDAR; the AutoPilot software is trained online (which means that the NN keeps learning and evolving during operation). The Autopilot's safety features are continuously evolved and enhanced through understanding real-world driving data from every Tesla.

Referring to these safety reports of existing autonomous cars, we should be aware that when testing DNN-based control software (the core part of autonomous vehicles), black-box system level testing (by observing inputs and its corresponding outputs, e.g., closed course testing and real-world driving) is still the leading method. More systematic T&V criteria and approaches are needed for more complete and reliable testing results.

5.2. Comparison with related work

5.2.1. Verification and validation of NNs

Taylor et al. [15] conducted a survey on the Verification and Validation (V&V) of NNs used in safety-critical domains in 2003. Study [15] is the closest work we found, although they did not adopt an SLR approach. Our study covered new studies from 2011 to 2018. The authors of [15] also made a classification of methods for the V&V of NNs. They grouped the methods into five traditional V&V technique categories, namely, automated testing and testing data generation methods, run-time monitoring, formal methods, cross validation, and visualization. In contrast to [15], our study adopted a thematic analysis approach [54] and identified five themes based on the research goals of the selected studies. We thought it was better to classify the proposed T&V methods of NNs based on their aims rather than on the traditional technique categories since many traditional V&V techniques are no longer effective for verifying NNs in many cases. New methods and tools should be explored and developed without being limited by the traditional V&V

Table 6
Shared verification interests of ISO/PAS 21448 and academic efforts.

ISO/PAS 21448	Academic efforts
Analysis of triggering events	CA1: Assuring robustness of NNs
Analysis of sensors design and their known potential limitations	CA2: Improving failure resilience of NNs
Analysis of environmental conditions and operational use cases	CA3: Measuring and ensuring test completeness
Analysis of boundary values	CA4: Assuring safety property of NN-based SCCPSs
Analysis of algorithms and their decision paths	CA5: Improving interpretability of NNs
Analysis of system architecture	CA1–CA5

categories. Another difference is our study specialized more in the T&V of modern NNs, such as MLP and DNN, whereas the study [15] provided more in-depth analysis of V&V methodologies for NNs used in flight control system, such as Pre-Trained Neural Network (PTNN) and Online Learning Neural Network (OLNN). Our study and [15] have some common findings. For example, one category, named *Visualization* in Taylor et al. [15], falls into our category CA5 Improving interpretability of NNs.

5.2.2. Surveys of security, safety, and productivity for deep learning (DL) systems development

Hains et al. [16] surveyed existing work on "*attacks against DL systems; testing, training, and monitoring DL systems for safety; and the verification of DL systems.*" Our study and [16] shared a similar motivation. The critical difference between our SLR and [16] are threefold: 1) We conducted our literature review on 83 selected papers based on specific SLR guidelines, while they used an ad hoc literature review (ALR) approach and reviewed only 21 primary papers. 2) They only focused on DL systems, whereas our scope covered modern NN-based software systems, which embodies DL-based software systems. 3) They inferred that formal methods and automation verification are the two promising research directions based on the reviewed works. In contrast, we focused more on safety issues, and found more categories to be addressed for safety purposes.

5.2.3. Surveys of certification of AI technologies in automotive

Falcini et al. [17,18] reviewed the existing standards in the automotive industry and pointed out the related applicability issues of automotive software development standards to deep learning. Although our SLR takes the automotive industry as an example, we are concerned with SCCPSs in general. This concern is reflected in the distribution of the selected papers (only 13 of the 83 selected papers are oriented to automotive CPSs).

5.2.4. SLR of explainable artificial intelligence (XAI)

There are two very recent SLRs, Adadi and Berrada [157] and Hohman et al. [158], on the interpretation of artificial intelligence. Both [157,158] employed similar commonly accepted guidelines to conduct their SLRs. The fundamental difference between our study and [157,158] is the scope. Adadi and Berrada [157] reviewed 381 papers on existing XAI approaches from interdisciplinary perspectives. As reported in Hohman et al. [158], the scope of their SLR is visualization and visual analytics for deep learning. The study [158] focused on studies that adopted visual analytics to explain NN decisions. Our study has a more comprehensive coverage of T&V approaches that were employed to not only interpret NN behaviors but also to assure the robustness of NNs, to improve the failure resilience of NNs, to ensure test completeness, and to assure the safety property of NN-based SCCPSs. In a summary, our SLR tried to provide an overview of key aspects related to T&V activities for NN-based SCCSs.

5.3. Threats to validity

In this section, we discuss some threats to the validity of our study.

5.3.1. Search strategy

The most possible threat in this step is missing or excluding relevant papers. To mitigate this threat, we used six of the most relevant digital libraries to retrieve papers. Additionally, we employed two strategies to mitigate potential limitations in the search terms: 1) adopted an PIOC criteria to ensure the coverage of search terms; and 2) improved search terms iteratively. Further, we conducted an extensive snowballing process on references of the selected papers to identify related papers. The search keywords were cross-checked and agreed on by both authors.

5.3.2. Study selection

Researchers' subjective judgment could be a threat to the study selection. We strictly followed the pre-defined review protocol to mitigate this threat. For example, we started recording the inclusion and exclusion reasons from the 3rd stage. We validated the inclusion and exclusion criteria with two authors on the basis of the pilot search. Furthermore, the second author performed a cross-check of all selected papers. Any paper that raised doubts about its inclusion or exclusion decision was discussed between the first and second authors. For example, the *"smart grid"* is included in the search term, but no relevant papers were found after the 3rd stage. Then, we conducted a snowballing search to identify papers that presented how to use NNs in smart grids. We found out that AI is mainly used to solve the economically relevant problems [159] of the smart grid system (e.g., prediction of energy usage and efficient use of resources). AI is not involved in the safety-critical applications (e.g., decision making on optimal provision of power) of smart grids. Therefore, there were no relevant papers addressing safety analysis or testing/verification (refer to Inclusion criteria I2).

5.3.3. Data extraction

The first author was responsible for designing the data extraction form and conducting the data extraction from selected papers. In order to avoid the first author's bias in data extraction, the two authors continuously discussed the data extraction issues. The extracted data were verified by the second author.

5.3.4. Data synthesis

Data analysis outcomes could vary with different researchers. To reduce the subjective impact on data synthesis, besides strictly following the thematic synthesis steps [54], the data synthesis was first agreed on by both authors. We disseminated our preliminary findings to two internal research groups at our university (i.e., the autonomous vehicle lab and autonomous ships lab) and presented at a Ph.D. seminar on IoT, Machine Learning, Security, and Privacy for comments and feedback. In summary, the audiences agreed with our research design and results, and they thought that the mapping of reviewed approaches to the IEC61508 is a valuable attempt. Several researchers working in formal verification and safety verification thought that safety cases would be a promising direction to address the challenges of T&V of NN-based SCCSs. One suggestion is adding information about self-driving car simulators. Based on these comments and feedback, we revised our paper accordingly.

6. Conclusion and future work

In this paper, we have presented the results of a Systematic Literature Review (SLR) of existing approaches and practices on T&V methods for neural-network-based safety critical control software (NN-based SCCS). The motivation of this study was to provide an overview of the state-of-the-art T&V of safety-critical NN-based SCCSs and to shed some light on potential research directions. Based on pre-defined inclusion and exclusion criteria, we selected 83 papers that were published between 2011 and 2018. A systematic analysis and synthesis of the data extracted from the papers and comprehensive reviews of industry practices (e.g., technical reports, standards, and white papers) related to our RQs were performed. Results of the study show that:

1. The research on T&V of NN-based SCCSs is gaining interest and attention from both software engineering and safety engineering researchers/practitioners according to the impressive upward trend in the number of papers on T&V of NN-based SCCSs (See Fig. 5). Most of the reviewed papers (68/83, 81.9%) have been published in the last three years.
2. The approaches and tools reported for the T&V of NN-based control software have been applied to a wide variety of safety-critical domains, among which “automotive CPSs” has received the most attention.
3. The approaches can be classified into five high-order themes, namely, assuring robustness of NNs, improving failure resilience of NNs, measuring and ensuring test completeness, assuring safety properties of NN-based SCCPSs, and improving interpretability of NNs.
4. The activities listed in the software safety lifecycles of IEC 61508-3 are still valid when conducting safety verification for NN-based control software. However, most of the activities need new techniques/measures to deal with the new characteristics of NNs.
5. Four safety integrity properties within the four major safety lifecycle phases, namely, correctness, completeness, freedom from intrinsic faults, and fault tolerance, have drawn the most attention from the research community. Little effort has been put on achieving repeatability. No reviewed study focused on precisely defined testing configuration and defense against common cause failure, which are extremely crucial for assuring the safety of a production-ready NN-based SCCS [160].
6. It is common to combine standard testing techniques with formal verification when testing and verifying large-scale, complex safety-critical software [15,144]. As explained in Section 4.3, we found that an increasing concern of the reviewed works is the integration of different T&V techniques in a systematic manner to gain assurance for the whole lifecycle of the NN-based control software.

This SLR is just a starting point in our studies to test and verify NN-based SCCPSs. In the future, we will focus on improving the interpretability of NNs. To be more specific, we plan to develop a method for explaining why an NN model is more (or less) robust than other models. It can guide software designers to design an NN model with an appropriate robustness level, which will greatly support safety by design.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Weifeng Liu for commenting on and improving this paper. This work is supported by the Safety, autonomy, remote control and operations of industrial transport systems (SAREPTA) project, which is financed by the Norwegian Research Council with grant no. 267860. This work is also supported by the Management of safety and security risks for cyber-physical systems project, which is financed by the Norwegian University of Technolgy in collaboration with the Technical University of Denmark, grant no. IE-053-2017.

Appendix A. Selected studies (sorted based on publication year)

S_ID	Author(s)	Year	Title	Publication venue
[92]	Pulina, L. and A. Tacchella	2011	NeVer: a tool for artificial neural networks verification	Annals of Mathematics and Artificial Intelligence
[90]	Pulina, L. and A. Tacchella	2012	Challenging SMT solvers to verify neural networks	AI Communications
[107]	Simonyan, K., A. Vedaldi and A. Zisserman	2013	Deep inside convolutional networks: Visualising image classification models and saliency maps	arXiv preprint
[103]	Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus	2013	Intriguing properties of neural networks	arXiv preprint
[56]	Goodfellow, I. J., J. Shlens and C. Szegedy	2014	Explaining and Harnessing Adversarial Examples	International Conference on Learning Representations (ICLR)
[61]	Gu, S. and L. Rigazio	2014	Towards deep neural network architectures robust to adversarial examples	International Conference on Learning Representations (ICLR)
[128]	Zeiler, M. D. and R. Fergus	2014	Visualizing and understanding convolutional networks	European conference on computer vision
[73]	Zhang, Q., T. Wang, Y. Tian, F. Yuan and Q. Xu	2015	ApproxANN: an approximate computing framework for artificial neural network	Design, Automation & Test in Europe Conference & Exhibition
[123]	Che, Z., S. Purushotham, R. Khemani and Y. Liu	2015	Distilling knowledge from deep networks with applications to healthcare domain	arXiv preprint
[124]	Hinton, G., O. Vinyals and J. Dean	2015	Distilling the knowledge in a neural network	arXiv preprint
[55]	Nguyen, A., J. Yosinski and J. Clune	2015	Deep neural networks are easily fooled: High confidence predictions for unrecognizable images	IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
[116]	Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller and W. Samek	2015	On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation	PloS one
[161]	Scheibler, K., L. Winterer, R. Wimmer and B. Becker	2015	Towards Verification of Artificial Neural Networks	Workshop on Methods and Description Languages for Modeling and Verification of Circuits and Systems (MBMV)
[71]	Shaham, U., Y. Yamada and S. Negahban	2015	Understanding adversarial training: Increasing local stability of neural nets through robust optimization	arXiv preprint
[133]	Mahendran, A. and A. Vedaldi	2015	Understanding deep image representations by inverting them	IEEE conference on computer vision and pattern recognition
[106]	Bach, S., A. Binder, K.-R. Müller and W. Samek	2016	Controlling explanatory heatmap resolution and semantics via decomposition depth	IEEE International Conference on Image Processing (ICIP)
[68]	Papernot, N., P. McDaniel, X. Wu, S. Jha and A. Swami	2016	Distillation as a defense to adversarial perturbations against deep neural networks	IEEE Symposium on Security & Privacy
[70]	Zheng, S., Y. Song, T. Leung and I. Goodfellow	2016	Improving the robustness of deep neural networks via stability training	IEEE conference on computer vision and pattern recognition
[82]	Daftry, S., S. Zeng, J. A. Bagnell and M. Hebert	2016	Introspective perception: Learning to predict failures in vision systems	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

(continued on next page)

S_ID	Author(s)	Year	Title	Publication venue
[112]	Zhou, B., A. Khosla, A. Lapedriza, A. Oliva and A. Torralba	2016	Learning deep features for discriminative localization	IEEE conference on computer vision and pattern recognition
[58]	Bastani, O., Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori and A. Criminisi	2016	Measuring neural net robustness with constraints	Advances in neural information processing systems
[115]	Shrikumar, A., P. Greenside, A. Shcherbina and A. Kundaje	2016	Not just a black box: Interpretable deep learning by propagating activation differences	arXiv Preprint
[95]	Julian, K. D., J. Lopez, J. S. Brush, M. P. Owen and M. J. Kochenderfer	2016	Policy compression for aircraft collision avoidance systems	IEEE/AIAA international conference on Digital Avionics Systems Conference (DASC)
[127]	Nguyen, A., A. Dosovitskiy, J. Yosinski, T. Brox and J. Clune	2016	Synthesizing the preferred inputs for neurons in neural networks via deep generator networks	Advances in Neural Information Processing Systems
[132]	Thiagarajan, J. J., B. Kailkhura, P. Sattigeri and K. N. Ramamurthy	2016	TreeView: Peeking into deep neural networks via feature-space partitioning	arXiv preprint
[75]	Li, G., K. Pattabiraman, C.-Y. Cher and P. Bose	2016	Understanding error propagation in GPGPU applications	International Conference on High Performance Computing, Networking, Storage and Analysis
[129]	Ribeiro, M. T., S. Singh and C. Guestrin	2016	Why should i trust you?: Explaining the predictions of any classifier	ACM SIGKDD International Conference on Knowledge Discovery and Data Mining
[105]	Sundararajan, M., A. Taly and Q. Yan	2017	Axiomatic attribution for deep networks	International Conference on Machine Learning
[83]	O'Kelly, M., H. Abbas and R. Mangharam	2017	Computer-aided design for safe autonomous vehicles	Resilience Week (RWS)
[101]	Tommaso DreossiAlexandre DonzSanjit A. Seshia	2017	Compositional Falsification of Cyber-Physical Systems with Machine Learning Components	NASA Formal Methods
[85]	Tian, Y., K. Pei, S. Jana and B. Ray	2017	DeepTest: Automated testing of deep-neural-network-driven autonomous cars	arXiv preprint
[64]	Reuben, F., R. R. Curtin, S. Saurabh and A. B. Gardner	2017	Detecting Adversarial Samples from Artifacts	arXiv preprint
[122]	Frosst, N. and G. Hinton	2017	Distilling a Neural Network Into a Soft Decision Tree	arXiv preprint
[84]	Pei, K., Y. Cao, J. Yang and S. Jana	2017	DeepXplore: Automated Whitebox Testing of Deep Learning Systems	ACM Symposium on Operating Systems Principles (SOSP)
[63]	Gopinath, D., G. Katz, C. S. Pasareanu and C. Barrett	2017	Deepsafe: A data-driven approach for checking adversarial robustness in neural networks	arXiv preprint
[108]	Montavon, G., S. Lapuschkin, A. Binder, W. Samek and K.-R. Müller	2017	Explaining nonlinear classification decisions with deep Taylor decomposition	Pattern Recognition
[76]	Santos, F. F. d., L. Draghetti, L. Weigel, L. Carro, P. Navaux and P. Rech	2017	Evaluation and Mitigation of Soft-Errors in Neural Network-Based Object Detection in Three GPU Architectures	IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)

(*continued on next page*)

S_ID	Author(s)	Year	Title	Publication venue
[69]	Papernot, N. and P. McDaniel	2017	Extending defensive distillation	arXiv preprint
[91]	Ehlers, R.	2017	Formal verification of piece-wise linear feed-forward neural networks	International Symposium on Automated Technology for Verification and Analysis
[77]	Manikandasriram, S. R., C. Anderson, R. Vasudevan and M. Johnson-Roberson	2017	Failing to learn: autonomously identifying perception failures for self-driving cars	arXiv preprint
[65]	Xu, W., D. Evans and Y. Qi	2017	Feature squeezing: Detecting adversarial examples in deep neural networks	Network and Distributed Systems Security Symposium (NDSS)
[110]	Dong, Y., H. Su, J. Zhu and B. Zhang	2017	Improving interpretability of deep neural networks with semantic information	IEEE Conference on Computer Vision and Pattern Recognition
[131]	Bastani, O., C. Kim and H. Bastani	2017	Interpretability via model extraction	arXiv preprint
[111]	Fong, R. C. and A. Vedaldi	2017	Interpretable explanations of black boxes by meaningful perturbation	IEEE International Conference on Computer Vision
[57]	Melis, M., A. Demontis, B. Biggio, G. Brown, G. Fumera and F. Roli	2017	Is Deep Learning Safe for Robot Vision? Adversarial Examples Against the iCub Humanoid	IEEE International Conference on Computer Vision Workshops (ICCVW)
[146]	Vishnukumar, H. J., B. Butting, C. Muller and E. Sax	2017	Machine learning and deep neural network - artificial intelligence core for lab and real-world test and validation for ADAS and autonomous vehicles: AI for efficient and quality test and validation	Intelligent Systems Conference (IntelliSys)
[78]	Mhamdi, E. M. E., R. Guerraoui and S. Rouault	2017	On the Robustness of a Neural Network	IEEE Symposium on Reliable Distributed Systems (SRDS)
[67]	Metzen, J. H., T. Genewein, V. Fischer and B. Bischoff	2017	On detecting adversarial perturbations	International Conference on Learning Representations (ICLR)
[93]	Dutta, S., S. Jha, S. Sanakaranarayanan and A. Tiwari	2017	Output range analysis for deep neural networks	arXiv preprint
[59]	Cisse, M., P. Bojanowski, E. Grave, Y. Dauphin and N. Usunier	2017	Parseval networks: Improving robustness to adversarial examples	arXiv preprint
[96]	Xiang, W., H.-D. Tran and T. T. Johnson	2017	Reachable set computation and safety verification for neural networks with ReLU activations	arXiv preprint
[97]	Katz, G., C. Barrett, D. L. Dill, K. Julian and M. J. Kochenderfer	2017	Reluplex: An efficient SMT solver for verifying deep neural networks	International Conference on Computer Aided Verification (CAV)
[117]	Dabkowski, P. and Y. Gal	2017	Real time image saliency for black box classifiers	Advances in Neural Information Processing Systems (NIPS)
[118]	Ross, A. S., M. C. Hughes and F. Doshi-Velez	2017	Right for the right reasons: Training differentiable models by constraining their explanations	arXiv preprint
[74]	Vialatte, J.-C. and F. Leduc-Primeau	2017	A Study of Deep Learning Robustness Against Computation Failures	arXiv preprint
[119]	Santoro, A., D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia and T. Lillicrap	2017	A simple neural network module for relational reasoning	Advances in Neural Information Processing Systems (NIPS)
[98]	Huang, X. W., M. Kwiatkowska, S. Wang and M. Wu	2017	Safety Verification of Deep Neural Networks	International Conference on Computer Aided Verification
[120]	Smilkov, Daniel and Thorat, Nikhil and Kim, Been and Viégas, Fernanda and Wattenberg, Martin	2017	Smoothgrad: removing noise by adding noise	arXiv preprint
[60]	Carlini, N. and D. Wagner	2017	Towards Evaluating the Robustness of Neural Networks	IEEE Symposium on Security and Privacy (SP)
[162]	Katz, G., C. Barrett, D. L. Dill, K. Julian and M. J. Kochenderfer	2017	Towards proving the adversarial robustness of deep neural networks	arXiv Preprint
[79]	Li, G., S. K. S. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer and S. W. Keckler	2017	Understanding error propagation in deep learning neural network (DNN) accelerators and applications	International Conference for High Performance Computing, Networking, Storage and Analysis

(*continued on next page*)

S_ID	Author(s)	Year	Title	Publication venue
[121]	Lundberg, S. M. and S.-I. Lee	2017	A unified approach to interpreting model predictions	Advances in Neural Information Processing Systems (NIPS)
[99]	Narodytska, N., S. P. Kasiviswanathan, L. Ryzhyk, M. Sagiv and T. Walsh	2017	Verifying properties of binarized deep neural networks	arXiv preprint
[86]	Raj, S., S. K. Jha, A. Ramanathan and L. L. Pullum	2017	Work-in-progress: testing autonomous cyber-physical systems using fuzzing features from convolutional neural networks	International Conference on Embedded Software (EMSOFT)
[72]	Schorn, C., A. Guntoro and G. Ascheid	2018	Accurate neuron resilience prediction for a flexible reliability management in neural network accelerators	Design, Automation & Test in Europe Conference & Exhibition (DATE)
[104]	Ribeiro, M. T., S. Singh and C. Guestrin	2018	Anchors: High-precision model-agnostic explanations	AAAI Conference on Artificial Intelligence
[87]	Ma, L., F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li and Y. Liu	2018	DeepGauge: multi-granularity testing criteria for deep learning systems	ACM/IEEE International Conference on Automated Software Engineering
[88]	Zhang, M., Y. Zhang, L. Zhang, C. Liu and S. Khurshid	2018	DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems.	ACM/IEEE International Conference on Automated Software Engineering
[89]	Guo, J., Y. Jiang, Y. Zhao, Q. Chen and J. Sun	2018	DLFuzz: differential fuzzing testing of deep learning systems	ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering
[80]	Rubaiyat, A. H. M., Q. Yongming and H. Alemzadeh	2018	Experimental Resilience Assessment of An Open-Source Driving Agent	arXiv preprint
[81]	Rhazali, K., B. Lussier, W. Schön and S. Geronimi	2018	Fault Tolerant Deep Neural Networks for Detection of Unrecognizable Situations	IFAC-PapersOnLine
[66]	Wicker, M., X. Huang and M. Kwiatkowska	2018	Feature-guided black-box safety testing of deep neural networks	International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)
[109]	Linsley, D., D. Scheibler, S. Eberhardt and T. Serre	2018	Global-and-local attention networks for visual recognition	arXiv preprint
[62]	Wu, M., M. Wicker, W. Ruan, X. Huang and M. Kwiatkowska	2018	A Game-Based Approximate Verification of Deep Neural Networks with Provable Guarantees	arXiv preprint
[125]	Xu, K., D. H. Park, C. Yi and C. Sutton	2018	Interpreting Deep Classifier by Visual Distillation of Dark Knowledge	arXiv preprint
[102]	Mallozzi, P., P. Pelliccione and C. Menghi	2018	Keeping intelligence under control.	International Workshop on Software Engineering for Cognitive Services
[114]	Guidotti, R., A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini and F. Giannotti	2018	Local rule-based explanations of black box decision systems	arXiv preprint
[130]	Guo, W., D. Mu, J. Xu, P. Su, G. Wang and X. Xing	2018	LEMNA: Explaining Deep Learning based Security Applications	ACM SIGSAC Conference on Computer and Communications Security
[126]	Tan, S., R. Caruana, G. Hooker, P. Koch and A. Gordo	2018	Learning Global Additive Explanations for Neural Nets Using Model Distillation	arXiv preprint
[113]	Dumitru, M. A. K.-R. M., E. B. K. S. D. Pieter, J. Kindermans and K. T. Schütt	2018	Learning how to explain neural networks: Patternnet and patternattribution	International Conference on Learning Representations (ICLR)
[94]	Xiang, W., H. D. Tran and T. T. Johnson	2018	Output Reachable Set Estimation and Verification for Multilayer Neural Networks	IEEE Transactions on Neural Networks and Learning Systems
[163]	Kuper, L., G. Katz, J. Gottschlich, K. Julian, C. Barrett and M. Kochenderfer	2018	Toward scalable verification for safety-critical deep networks	arXiv preprint
[100]	Cheng, C.-H., G. Nührenberg and H. Ruess	2018	Verification of binarized neural networks	arXiv preprint

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.infsof.2020.106296.

References

[1] R. Rajkumar, I. Lee, L. Sha, J. Stankovic, Cyber-physical systems: the next computing revolution, in: Design Automation Conference (DAC), 2010 47th ACM/IEEE, IEEE, 2010, pp. 731–736.

[2] B.K. Bose, Neural network applications in power electronics and motor drives–an introduction and perspective, IEEE Trans. Ind. Electron. 54 (1) (2007) 14–33.

[3] P. Ongsulee, Artificial intelligence, machine learning and deep learning, in: ICT and Knowledge Engineering (ICT&KE), 2017 15th International Conference on, 2017, pp. 1–6.

[4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L.D. Jackel, M. Monfort, U. Muller, J. Zhang, et al., End to end learning for self-driving cars, arXiv preprint arXiv:1604.07316 (2016).

[5] K.D. Julian, J. Lopez, J.S. Brush, M.P. Owen, M.J. Kochenderfer, Policy compression for aircraft collision avoidance systems, in: Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th, IEEE, 2016, pp. 1–10.

[6] S. Levin, J.C. Wong, Self-driving uber kills arizona woman in first fatal crash involving pedestrian, 2018, (https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe), Accessed: 2018-07-27.

[7] D. Yadron, D. Tynan, Tesla driver dies in first fatal crash while using autopilot mode, 2016, (https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk), Accessed: 2018-07-27.

[8] D. Lee, Google self-driving car hits a bus, 2016, (https://www.bbc.com/news/technology-35692845), Accessed:18-12-2018.

[9] Valasek, Chris, Miller, Charlie, Who's behind the wheel? exposing the vulnerabilities and risks of high tech vehicles, 2015, (https://trid.trb.org/view/1370158), Accessed: 2018-07-27.

[10] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, Y. Halgand, A survey of approaches combining safety and security for industrial control systems, Reliab. Eng. Syst. Saf. 139 (2015) 156–178.

[11] T. Aven, A unified framework for risk and vulnerability analysis covering both safety and security, Reliab. Eng. Syst. Saf. 92 (6) (2007) 745–754, doi:10.1016/j.ress.2006.03.008.

[12] G. Stoneburner, Toward a unified security-safety model, Computer 39 (8) (2006) 96–97.

[13] T. Novak, A. Treytl, Functional safety and system security in automation systems—A life cycle model, in: 2008 IEEE International Conference on Emerging Technologies and Factory Automation, 2008, pp. 311–318, doi:10.1109/ETFA.2008.4638412.

[14] P. Bieber, J.-P. Blanquart, G. Descargues, M. Dulucq, Y. Fourastier, E. Hazane, M. Julien, L. Léonardon, G. Sarouille, Security and safety assurance for aerospace embedded systems, in: Proceedings of the 6th International Conference on Embedded Real Time Software and Systems, Toulouse, France, 2012, pp. 1–10.

[15] B.J. Taylor, M.A. Darrah, C.D. Moats, Verification and validation of neural networks: a sampling of research in progress, in: Intelligent Computing: Theory and Applications, 5103, International Society for Optics and Photonics, 2003, pp. 8–17.

[16] G. Hains, A. Jakobsson, Y. Khmelevsky, Towards formal methods and software engineering for deep learning: security, safety and productivity for dl systems development, in: Systems Conference (SysCon), 2018 Annual IEEE International, IEEE, 2018, pp. 1–5.

[17] F. Falcini, G. Lami, Challenges in certification of autonomous driving systems, in: 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2017, pp. 286–293, doi:10.1109/ISSREW.2017.45.

[18] F. Falcini, G. Lami, Deep learning in automotive: challenges and opportunities, in: A. Mas, A. Mesquida, R.V. O'Connor, T. Rout, A. Dorling (Eds.), Software Process Improvement and Capability Determination, Springer International Publishing, 2017, pp. 279–288.

[19] P. Van Wesel, A.E. Goodloe, Challenges in the Verification of Reinforcement Learning Algorithms, Technical Report, 2017. https://ntrs.nasa.gov/search.jsp?R=20170007190

[20] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, 2007.

[21] E. Lee, The past, present and future of cyber-physical systems: a focus on models, Sensors 15 (3) (2015) 4837–4869.

[22] A. Humayed, J. Lin, F. Li, B. Luo, Cyber-physical systems security—A survey, IEEE Internet Things J. 4 (6) (2017) 1802–1831, doi:10.1109/JIOT.2017.2703172.

[23] E.R. Griffor, C. Greer, D.A. Wollman, M.J. Burns, Framework for Cyber-Physical Systems: Volume 1, Overview, Technical Report, 2017. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-201.pdf

[24] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133.

[25] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain., Psychol. Rev. 65 (6) (1958) 386.

[26] G. Katz, C. Barrett, D.L. Dill, K. Julian, M.J. Kochenderfer, Reluplex: an efficient SMT solver for verifying deep neural networks, in: International Conference on Computer Aided Verification, Springer, 2017, pp. 97–117.

[27] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher, P. Held, Multi-layer perceptrons, in: Computational Intelligence, Springer, 2013, pp. 47–81.

[28] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.

[29] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[30] M. van Gerven, S. Bohte, Artificial Neural Networks as Models of Neural Information Processing, Frontiers Media SA, 2018.

[31] D.M. Rodvold, A software development process model for artificial neural networks in critical applications, in: IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339), 5, 1999, pp. 3317–3322, doi:10.1109/IJCNN.1999.836192.

[32] F. Falcini, G. Lami, A.M. Costanza, Deep learning in automotive software, IEEE Softw. 34 (3) (2017) 56–63, doi:10.1109/MS.2017.79.

[33] SAE, J3016:Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, Standard, 2014, doi:10.4271/J3016_201401.

[34] NVIDIA, Nvidia drive:scalable ai platform for autonomous driving, 2018, (https://www.nvidia.com/en-us/self-driving-cars/drive-platform/), Accessed:18-12-2018.

[35] J.C. Hoskins, D.M. Himmelblau, Process control via artificial neural networks and reinforcement learning, Comput. Chem. Eng. 16 (4) (1992) 241–251, doi:10.1016/0098-1354(92)80045-B.

[36] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971 (2015).

[37] S.P.K. Spielberg, R.B. Gopaluni, P.D. Loewen, Deep reinforcement learning approaches for process control, in: 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP), 2017, pp. 201–206, doi:10.1109/ADCONIP.2017.7983780.

[38] G. Zhabelova, V. Vyatkin, Multiagent smart grid automation architecture based on IEC 61850/61499 intelligent logical nodes, IEEE Trans. Ind. Electron. 59 (5) (2012) 2351–2362, doi:10.1109/TIE.2011.2167891.

[39] B.K. Bose, Artificial intelligence techniques in smart grid and renewable energy systems–some example applications, Proc. IEEE 105 (11) (2017) 2262–2273, doi:10.1109/JPROC.2017.2756596.

[40] G. Robertson, E.D. Lehmann, W. Sandham, D. Hamilton, Blood glucose prediction using artificial neural networks trained with the AIDA diabetes simulator: a proof-of-concept pilot study, J. Electr. Comput. Eng. 2011 (2011) 1–11, doi:10.1155/2011/681786.

[41] M.K. Bothe, L. Dickens, K. Reichel, A. Tellmann, B. Ellger, M. Westphal, A.A. Faisal, The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas, Expert Rev. Med. Devices 10 (5) (2013) 661–673.

[42] Medtronic, Medtronic initiates u.s. launch of world's first hybrid closed loop system for type 1 diabetes, 2017, (http://newsroom.medtronic.com/phoenix.zhtml?c=251324&p=irol-newsArticle&ID=2279529), Accessed: 2018-08-25.

[43] K. Sennaar, Ai in medical devices – three emerging industry applications, 2018, (https://www.techemergence.com/ai-medical-devices-three-emerging-industry-applications/). Accessed: 2018-08-16.

[44] H. Greenspan, B. Van Ginneken, R.M. Summers, Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique, IEEE Trans. Med. Imaging 35 (5) (2016) 1153–1159.

[45] IEC61508:2005, Functional safety of electrical/electronic/programmable electronic safety-related systems, Standard, International Electrotechnical Commission, 2005.

[46] ISO 26262:2011, Road vehicles – Functional safety, Standard, International Organization for Standardization, 2011.

[47] G. Griessnig, A. Schnellbach, Development of the 2nd edition of the ISO26262, in: J. Stolfa, S. Stolfa, R.V. O'Connor, R. Messnarz (Eds.), Systems, Software and Services Process Improvement, Springer International Publishing, 2017, pp. 535–546.

[48] Hansen, Standardization Efforts on Autonomous Driving Safety Barely Under Way, Technical Report, 2017.

[49] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, Carla: an open urban driving simulator, arXiv:1711.03938 (2017).

[50] Udacity, An open source self-driving car, 2016, (https://github.com/udacity/self-driving-car). Accessed:2018-12-19.

[51] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: an update, Inf. Softw. Technol. 64 (2015) 1–18, doi:10.1016/j.infsof.2015.03.007.

[52] M. Shahin, M.A. Babar, L. Zhu, Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices, IEEE Access 5 (2017) 3909–3943.

[53] P.H. Nguyen, S. Ali, T. Yue, Model-based security engineering for cyber-physical systems: a systematic mapping study, Inf. Softw. Technol. 83 (2017) 116–135, doi:10.1016/j.infsof.2016.11.004.

[54] D.S. Cruzes, T. Dyba, Recommended steps for thematic synthesis in software engineering, in: 2011 International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 275–284, doi:10.1109/ESEM.2011.36.

[55] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: high confidence predictions for unrecognizable images, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 427–436.

[56] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv:1412.6572 (2014).

[57] M. Melis, A. Demontis, B. Biggio, G. Brown, G. Fumera, F. Roli, Is deep learning safe for robot vision? Adversarial examples against the iCub humanoid, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 2017, pp. 751–759, doi:10.1109/ICCVW.2017.94.

[58] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, A. Criminisi, Measuring neural net robustness with constraints, Advances in Neural Information Processing Systems (2016) 2613–2621.

[59] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, N. Usunier, Parseval networks: improving robustness to adversarial examples, arXiv preprint arXiv:1704.08847 (2017).

[60] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 39–57, doi:10.1109/SP.2017.49.

[61] S. Gu, L. Rigazio, Towards deep neural network architectures robust to adversarial examples, arXiv preprint arXiv:1412.5068 (2014).

[62] M. Wu, M. Wicker, W. Ruan, X. Huang, M. Kwiatkowska, A game-based approximate verification of deep neural networks with provable guarantees, arXiv preprint arXiv:1807.03571 (2018).

[63] D. Gopinath, G. Katz, C.S. Pasareanu, C. Barrett, Deepsafe: a data-driven approach for checking adversarial robustness in neural networks, arXiv preprint arXiv:1710.00486 (2017).

[64] F. Reuben, R.R. Curtin, S. Saurabh, A.B. Gardner, Detecting adversarial samples from artifacts, arXiv preprint arXiv:1703.00410 (2017).

[65] W. Xu, D. Evans, Y. Qi, Feature squeezing: detecting adversarial examples in deep neural networks, arXiv preprint arXiv:1704.01155 (2017).

[66] M. Wicker, X. Huang, M. Kwiatkowska, Feature-guided Black-Box Safety Testing of Deep Neural Networks, in: LNCS, 10805, Springer Verlag, 2018, pp. 408–426, doi:10.1007/978-3-319-89960-2_22.

[67] J.H. Metzen, T. Genewein, V. Fischer, B. Bischoff, On detecting adversarial perturbations, arXiv preprint arXiv:1702.04267 (2017).

[68] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, arXiv preprint arXiv:1511.04508 (2015).

[69] N. Papernot, P. McDaniel, Extending defensive distillation, arXiv preprint arXiv:1705.05264 (2017).

[70] S. Zheng, Y. Song, T. Leung, I. Goodfellow, Improving the robustness of deep neural networks via stability training, in: Proceedings of the IEEE conference on computer vision and pattern Recognition, 2016, pp. 4480–4488.

[71] U. Shaham, Y. Yamada, S. Negahban, Understanding adversarial training: increasing local stability of neural nets through robust optimization, arXiv preprint arXiv:1511.05432 (2015).

[72] C. Schorn, A. Guntoro, G. Ascheid, Accurate neuron resilience prediction for a flexible reliability management in neural network accelerators, in: 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018, pp. 979–984, doi:10.23919/DATE.2018.8342151.

[73] Q. Zhang, T. Wang, Y. Tian, F. Yuan, Q. Xu, Approxann: An Approximate Computing Framework for Artificial Neural Network, EDA Consortium, 2015, pp. 701–706.

[74] J.-C. Vialatte, F. Leduc-Primeau, A study of deep learning robustness against computation failures, arXiv preprint arXiv:1704.05396 (2017).

[75] G. Li, K. Pattabiraman, C.-Y. Cher, P. Bose, Understanding error Propagation in GPGPU Applications, IEEE, 2016, pp. 240–251.

[76] F.F.d. Santos, L. Draghetti, L. Weigel, L. Carro, P. Navaux, P. Rech, Evaluation and mitigation of soft-errors in neural network-based object detection in three GPU architectures, in: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2017, pp. 169–176, doi:10.1109/DSN-W.2017.47.

[77] S.R. Manikandasriram, C. Anderson, R. Vasudevan, M. Johnson-Roberson, Failing to learn: autonomously identifying perception failures for self-driving cars [arxiv], arXiv:1707.00051 (2017) 8 pp.

[78] E.M.E. Mhamdi, R. Guerraoui, S. Rouault, On the robustness of a neural network, in: 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), 2017, pp. 84–93, doi:10.1109/SRDS.2017.21.

[79] G. Li, S.K.S. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer, S.W. Keckler, Understanding error propagation in deep learning neural network (DNN) accelerators and applications, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, 2017, p. 8.

[80] A.H.M. Rubaiyat, Y. Qin, H. Alemzadeh, Experimental resilience assessment of an open-source driving agent, CoRR abs/1807.06172 (2018).

[81] K. Rhazali, B. Lussier, W. Schön, S. Geronimi, Fault tolerant deep neural networks for detection of unrecognizable situations, IFAC-PapersOnLine 51 (24) (2018) 31–37, doi:10.1016/j.ifacol.2018.09.525.

[82] S. Daftry, S. Zeng, J.A. Bagnell, M. Hebert, Introspective perception: learning to predict failures in vision systems, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 1743–1750, doi:10.1109/IROS.2016.7759279.

[83] M. O'Kelly, H. Abbas, R. Mangharam, Computer-aided design for safe autonomous vehicles, in: Resilience Week (RWS) 2017, 2017 Resilience Week (RWS), IEEE, 2017, pp. 90–96, doi:10.1109/RWEEK.2017.8088654.

[84] K. Pei, Y. Cao, J. Yang, S. Jana, Deepxplore: Automated Whitebox Testing of Deep Learning Systems, Association for Computing Machinery, Inc, 2017, pp. 1–18, doi:10.1145/3132747.3132785.

[85] Y. Tian, K. Pei, S. Jana, B. Ray, Deeptest: automated testing of deep-neural-network-driven autonomous cars, in: Proceedings of the 40th International Conference on Software Engineering, in: ICSE '18, ACM, New York, NY, USA, 2018, pp. 303–314, doi:10.1145/3180155.3180220.

[86] S. Raj, S.K. Jha, A. Ramanathan, L.L. Pullum, Work-in-progress: testing autonomous cyber-physical systems using fuzzing features from convolutional neural networks, in: 2017 International Conference on Embedded Software (EMSOFT), 2017, pp. 1–2, doi:10.1145/3125503.3125568.

[87] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, J. Zhao, Y. Wang, Deepgauge: multi-granularity testing criteria for deep learning systems, in: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ACM, 2018, pp. 120–131, doi:10.1145/3238147.3238202.

[88] M. Zhang, Y. Zhang, L. Zhang, C. Liu, S. Khurshid, Deeproad: gan-based metamorphic testing and input validation framework for autonomous driving systems, in: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ACM, 2018, pp. 132–142.

[89] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, J. Sun, Dlfuzz: differential fuzzing testing of deep learning systems, in: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ACM, 2018, pp. 739–743.

[90] L. Pulina, A. Tacchella, Challenging SMT solvers to verify neural networks, AI Commun. 25 (2) (2012) 117–135.

[91] R. Ehlers, Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks, Springer, 2017, pp. 269–286.

[92] L. Pulina, A. Tacchella, Never: a tool for artificial neural networks verification, Ann. Math. Artif. Intell. 62 (3–4) (2011) 403–425.

[93] S. Dutta, S. Jha, S. Sanakaranarayanan, A. Tiwari, Output range analysis for deep neural networks, arXiv preprint arXiv:1709.09130 (2017).

[94] W. Xiang, H.D. Tran, T.T. Johnson, Output reachable set estimation and verification for multilayer neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2018) 1–7, doi:10.1109/TNNLS.2018.2808470.

[95] K.D. Julian, J. Lopez, J.S. Brush, M.P. Owen, M.J. Kochenderfer, Policy Compression for Aircraft Collision Avoidance Systems, IEEE, 2016, pp. 1–10.

[96] W. Xiang, H.-D. Tran, T.T. Johnson, Reachable set computation and safety verification for neural networks with ReLU activations, arXiv preprint arXiv:1712.08163 (2017).

[97] G. Katz, C. Barrett, D.L. Dill, K. Julian, M.J. Kochenderfer, Reluplex: an efficient SMT solver for verifying deep neural networks, in: Computer Aided Verification. CAV 2017, Springer, 2017, pp. 97–117.

[98] X. Huang, M. Kwiatkowska, S. Wang, M. Wu, Safety verification of deep neural networks, in: International Conference on Computer Aided Verification, Springer, 2017, pp. 3–29.

[99] N. Narodytska, S.P. Kasiviswanathan, L. Ryzhyk, M. Sagiv, T. Walsh, Verifying properties of binarized deep neural networks, arXiv preprint arXiv:1709.06662 (2017).

[100] C.-H. Cheng, G. Nührenberg, H. Ruess, Verification of binarized neural networks, arXiv preprint arXiv:1710.03107 (2018).

[101] T. Dreossi, A. Donzé, S.A. Seshia, Compositional falsification of cyber-physical systems with machine learning components, in: NASA Formal Methods, Springer International Publishing, 2017, pp. 357–372.

[102] P. Mallozzi, P. Pelliccione, C. Menghi, Keeping intelligence under control, in: Proceedings of the 1st International Workshop on Software Engineering for Cognitive Services, ACM, 2018, pp. 37–40, doi:10.1145/3195555.3195558.

[103] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[104] M.T. Ribeiro, S. Singh, C. Guestrin, Anchors: high-precision model-agnostic explanations, in: Proceedings of the 32rd AAAI Conference on Artificial Intelligence, 2018.

[105] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 3319–3328.

[106] S. Bach, A. Binder, K.-R. Müller, W. Samek, Controlling explanatory heatmap resolution and semantics via decomposition depth, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 2271–2275.

[107] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, arXiv preprint arXiv:1312.6034 (2013).

[108] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, Explaining nonlinear classification decisions with deep taylor decomposition, Pattern Recognit. 65 (2017) 211–222, doi:10.1016/j.patcog.2016.11.008.

[109] D. Linsley, D. Scheibler, S. Eberhardt, T. Serre, Global-and-local attention networks for visual recognition, arXiv preprint arXiv:1805.08819 (2018).

[110] Y. Dong, H. Su, J. Zhu, B. Zhang, Improving interpretability of deep neural networks with semantic information, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4306–4314.

[111] R.C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3429–3437.

[112] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

[113] M.A.K.-R.M. Dumitru, E.B.K.S.D. Pieter, J. Kindermans, K.T. Schütt, Learning how to explain neural networks: patternnet and patternattribution, in: Proceedings of the International Conference on Learning Representations (2018), 2018.

[114] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, arXiv preprint arXiv:1805.10820 (2018).

[115] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 3145–3153.

[116] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One 10 (7) (2015) 1–46, doi:10.1371/journal.pone.0130140.

[117] P. Dabkowski, Y. Gal, Real time image saliency for black box classifiers, in: Advances in Neural Information Processing Systems, 2017, pp. 6967–6976.

[118] A.S. Ross, M.C. Hughes, F. Doshi-Velez, Right for the right reasons: training differentiable models by constraining their explanations, arXiv preprint arXiv:1703.03717 (2017).
[119] A. Santoro, D. Raposo, D.G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, T. Lillicrap, A simple neural network module for relational reasoning, in: Advances in Neural Information Processing Systems, 2017, pp. 4967–4976.
[120] D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, Smoothgrad: removing noise by adding noise, arXiv preprint arXiv:1706.03825 (2017).
[121] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.
[122] N. Frosst, G. Hinton, Distilling a neural network into a soft decision tree, arXiv preprint arXiv:1711.09784 (2017).
[123] Z. Che, S. Purushotham, R. Khemani, Y. Liu, Distilling knowledge from deep networks with applications to healthcare domain, arXiv preprint arXiv:1512.03542 (2015).
[124] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 (2015).
[125] K. Xu, D.H. Park, C. Yi, C. Sutton, Interpreting deep classifier by visual distillation of dark knowledge, arXiv preprint arXiv:1803.04042 (2018).
[126] S. Tan, R. Caruana, G. Hooker, P. Koch, A. Gordo, Learning global additive explanations for neural nets using model distillation, arXiv preprint arXiv:1801.08640 (2018).
[127] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, in: Advances in Neural Information Processing Systems, 2016, pp. 3387–3395.
[128] M.D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, Springer, 2014, pp. 818–833.
[129] M.T. Ribeiro, S. Singh, C. Guestrin, Why Should I Trust you?: Explaining the Predictions of any Classifier, ACM, 2016, pp. 1135–1144.
[130] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, X. Xing, Lemna: explaining deep learning based security applications, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2018, pp. 364–379.
[131] O. Bastani, C. Kim, H. Bastani, Interpretability via model extraction, arXiv preprint arXiv:1706.09773 (2017).
[132] J.J. Thiagarajan, B. Kailkhura, P. Sattigeri, K.N. Ramamurthy, Treeview: peeking into deep neural networks via feature-space partitioning, arXiv preprint arXiv:1611.07429 (2016).
[133] A. Mahendran, A. Vedaldi, Understanding deep image representations by inverting them, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5188–5196.
[134] J. Deng, W. Dong, R. Socher, L. Li, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, doi:10.1109/CVPR.2009.5206848.
[135] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The Limitations of Deep Learning in Adversarial Settings, IEEE, 2016, pp. 372–387.
[136] Q. Lu, M. Farahani, J. Wei, A. Thomas, K. Pattabiraman, Llfi: An intermediate code-level fault injection tool for hardware faults, in: 2015 IEEE International Conference on Software Quality, Reliability and Security, IEEE, 2015, pp. 11–16.
[137] S. Borkar, Designing reliable systems from unreliable components: the challenges of transistor variability and degradation, IEEE Micro 25 (6) (2005) 10–16.
[138] N. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety, MIT Press, 2011.
[139] T.Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, T.H. Tse, Z.Q. Zhou, Metamorphic testing: a review of challenges and opportunities, ACM Comput. Surv. 51 (1) (2018) 4:1–4:27, doi:10.1145/3143561.
[140] M.W. Moskewicz, C.F. Madigan, Y. Zhao, L. Zhang, S. Malik, Chaff: Engineering an Efficient Sat Solver, ACM, 2001, pp. 530–535.
[141] H. Zhang, Sato: an efficient prepositional prover, in: International Conference on Automated Deduction, Springer, 1997, pp. 272–275.
[142] J.P. Marques-Silva, K.A. Sakallah, Grasp: a search algorithm for propositional satisfiability, IEEE Trans. Comput. 48 (5) (1999) 506–521.
[143] C. Barrett, C. Tinelli, Satisfiability Modulo Theories, in *Handbook of Model Checking*: Springer, 2018, pp. 305–343.
[144] W.R. Adrion, M.A. Branstad, J.C. Cherniavsky, Validation, verification, and testing of computer software, ACM Comput. Surv. (CSUR) 14 (2) (1982) 159–192.
[145] Protecting Against Common Cause Failures in Digital I&C Systems of Nuclear Power Plants, Number NP-T-1.5, Nuclear Energy Series, International Atomic Energy Agency, Vienna, 2009.
[146] H.J. Vishnukumar, B. Butting, C. Muller, E. Sax, Machine learning and deep neural network - artificial intelligence core for lab and real-world test and validation for ADAS and autonomous vehicles: ai for efficient and quality test and validation, in: 2017 Intelligent Systems Conference, 2017, pp. 714–721, doi:10.1109/IntelliSys.2017.8324372.
[147] R. Ashmore, M. Hill, “Boxing clever”: practical techniques for gaining insights into training data and monitoring distribution shift, in: International Conference on Computer Safety, Reliability, and Security, Springer, 2018, pp. 393–405.
[148] N. Jouppi, Google supercharges machine learning tasks with TPU custom chip, 2017, (https://cloud.google.com/blog/products/gcp/google-supercharges-machine-learning-tasks-with-custom-chip). Accessed: 2018-08-25.
[149] NVIDIA, Partner innovation:accelerating automotive breakthroughs, 2018, (https://www.nvidia.com/en-us/self-driving-cars/partners/). Accessed:2018-12-19.
[150] WAYMO, Waymo Safety Report : On the Road to Fully Self-Driving, Technical Report, 2017. https://www.bbc.com/news/technology-35692845
[151] GoogleCloud, Google Infrastructure Security Design Overview, Technical Report, 2017. https://cloud.google.com/security/infrastructure/design/
[152] GM, Self-driving safety report, Technical Report, 2018. https://www.gm.com/our-stories/self-driving-cars.html
[153] S. Shalev-Shwartz, S. Shammah, A. Shashua, On a Formal Model of Safe and Scalable Self-driving Cars, arXiv e-prints arXiv:1708.06374 (2017).
[154] Mobileye, Mobileye: sensing the future, 2018, (https://www.mobileye.com/). Accessed:2018-12-19.
[155] Tesla, Tesla vehicle safety report, 2018, (https://www.tesla.com/VehicleSafetyReport). Accessed: 2019-11-01.
[156] Tesla, Your tesla is learning to drive by itself, 2019, (https://evannex.com/blogs/news). Accessed: 2019-11-01.
[157] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence (XAI), IEEE Access 6 (2018) 52138–52160.
[158] F.M. Hohman, M. Kahng, R. Pienta, D.H. Chau, Visual analytics in deep learning: An interrogative survey for the next frontiers, IEEE Transactions on Visualization and Computer Graphics (2018), doi:10.1109/TVCG.2018.2843369. 1–1
[159] S. Khan, D. Paul, P. Momtahan, M. Aloqaily, Artificial Intelligence Framework for Smart City Microgrids: State of the Art, Challenges, and Opportunities, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 283–288, doi:10.1109/FMEC.2018.8364080.
[160] A. Arpteg, B. Brinne, L. Crnkovic-Friis, J. Bosch, Software engineering challenges of deep learning, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2018, pp. 50–59.
[161] K. Scheibler, L. Winterer, R. Wimmer, B. Becker, Towards verification of artificial neural networks, In: 18th MBMV Workshop, 2015, pp. 30–40
[162] G. Katz, C. Barrett, D.L. Dill, K. Julian, M.J. Kochenderfer, Towards proving the adversarial robustness of deep neural networks, arXiv preprint arXiv:1709.02802 (2017).
[163] L. Kuper, G. Katz, J. Gottschlich, K. Julian, C. Barrett, M. Kochenderfer, Toward scalable verification for safety-critical deep networks, arXiv preprint arXiv:1801.05950 (2018).
[164] J. L. Heilbron, The Oxford Companion to the History of Modern Science, Oxford University Press, 2003.

Paper 2

Analyzing influence of robustness of neural networks on the safety of autonomous vehicles,

Jin Zhang, Robert Taylor, Igor Kozin, and Jingyue Li.

In: 31st European Safety and Reliability Conference (ESREL), pp. 2276-2283.

Analyzing Influence of Robustness of Neural Networks on the Safety of Autonomous Vehicles

Jin Zhang[1,2,3]

[1] *Computer Science Department,Norwegian University of Science and Technology(NTNU), Norway.*
[2] *Engineering Systems Design Group, Technical University of Denmark (DTU), Denmark.*
[3] *School of Information Science and Technology, Southwest Jiaotong University(SWJTU), China.*
E-mail: jin.zhang@ntnu.no

J.Robert Taylor

Independent consultant and researcher, Denmark. E-mail: roberttayloritsa@gmail.com

Igor Kozin

Independent consultant and researcher, Denmark. E-mail: igor.o.kozin@gmail.com

Jingyue Li

Computer Science Department, Norwegian University of Science and Technology(NTNU), Norway.
E-mail: jingyue.li@ntnu.no

Neural networks (NNs) have shown remarkable performance of perception in their application in autonomous vehicles (AVs). However, NNs are intrinsically vulnerable to perturbations, such as occurrences outside of the training sets, scene noise, instrument noise, image translation, and rotation, or small changes intentionally added to the original image (called adversarial perturbations). Incorrect conclusions from the perception systems (e.g., missing objects, wrong classification, and traffic sign misdetection or misreading) have been a major cause of disengagement incidents in AVs. In order to explore the dynamic nature of hazardous events in AVs, we develop a range of methods to analyze AV safety and security. This work is part of the project and is devoted to analyzing the influence of robustness in the NN-based perception system by using fault tree analysis (FTA). We extend the traditional FTA to represent combinations of failure causes in the multi-dimensional space, i.e., two variables that influence whether the image is classified correctly. The extended FTA is demonstrated on the traffic sign recognition module of AV theoretically and in practice.

Keywords: safety, neural network, autonomous vehicles, robustness, failure mode, hazard identification.

1. Introduction

The development of Autonomous Vehicles (AVs) is proceeding rapidly and promises safer and more efficient roads. However, safety and security problems remain, and disengagement incidents, that is, the handover of vehicle control to a human driver, present a major problem Banerjee et al. (2018). ISO 26262:2011 (2011) and ISO/PAS21448:2019 (2019) intended to address the growing complexity of vehicle systems. However, ISO 26262 does not clearly specify the methods for safety analysis. In the automotive domain, traditional hazard analysis techniques such as Fault Tree Analysis (FTA) and Failure Mode and Effects Analysis (FMEA) or Hazard and Operability Analysis (HAZOP) are generally used for the complex system. In this study, the methods are extended to cover problems arising particularly in Neural Networks (NNs).

One of the major problems in analyzing AV controllers is that of NN components. Deep Neural Networks (DNNs) have been widely used for object detection, image recognition, navigation, and control in AVs. Although DNNs are powerful methods for performing complex tasks compared to humans, they are extremely vulnerable to natural noise Hendrycks and Dietterich (2019) and to small perturbations intentionally added to the input to cause mispredictions Szegedy et al. (2013). A DNN is different from traditional human written programs with certain intended behaviors. Risk analysis of the use of DNNs is at present challenging due to its black-box nature. Analyzing the internal working of a NN with no underlying design is computationally hard Shalev-Shwartz et al. (2017); Johnson (2018). This sets a limit on what can be achieved by hazard identification.

Kalra and Paddock of Rand Corporation made

Proceedings of the 31st European Safety and Reliability Conference
Edited by Bruno Castanier, Marko Cepin, David Bigaud, and Christophe Berenguer

Published by Research Publishing, Singapore.
ISBN: 978-981-18-2016-8; doi:10.3850/978-981-18-2016-8_518-cd

a statistical assessment on the number of miles of driving that would be needed for AV safety Kalra and Paddock (2016). Their results show that demonstrating with 95% confidence that the AV failure rate is 20% better than the human driver failure rate would require 11 billion miles of on-road driving (equivalent to 500 billion vehicle years to complete the requisite miles). This level of testing is impractical. Therefore, it is desirable to analyze safety in the same way that other rare hazards are analyzed, that is, by risk analysis based on component reliabilities and by in-depth assessment of defense. This does not mean that on-road testing would not be needed. On-road testing is an evidence-based way of performing this validation. The risk analysis provides a way of amplifying the value of on-road testing, allowing near miss and partial failure cases to be included in the evidence base while providing a framework for assessing such less serious incidents Taylor et al. (2021).

This paper describes the part of the study that investigates the influence of perturbations in NNs in the context of AVs from an integrated perspective. We consider both safety hazards due to natural perturbations and security threats due to adversarial perturbations as part of an entire system risk assessment. We analyze the failure modes of perturbations in the NN-based perception system by using various hazard identification methods and a combination of methods, i.e., the use of dynamic fault tree methods to explicit reliability analysis of NNs. We also use the Systems Theoretic Process Analysis (STPA) of control loops but include emergent hazards Taylor and Kozin (2021a) as well as component functional failures and the semi-automated fault tree construction to help obtain completeness and consistency in the FTAs. The proposed methods are tested using a design for a 1/4 scale AV. The physical model enables the effects of "real world" problems such as camera resolution, processing response times, the field of view, camera alignment etc., to be investigated in the context of NN performance. An FTA was made for the entire vehicle, including physical, control, and sensor components. The design used as an example for the analysis includes vision algorithms and NNs for control of steering, acceleration, and braking. Due to the space limits, we present the whole FTA in a technical report Taylor et al. (2021).

Our main contribution is to show that NNs and vision algorithms can be included in overall risk analysis in the form of a Fault Tree (FT) by using the concept of exceeding robustness of NNs as FT events alongside the traditional component failure probabilities. The second contribution is that we demonstrate how an FT can include failure events that stem from multiple small deviations of parameters influencing image recognition. These failure events are a very special class of failures that are difficult to identify and quantify. The difficulty is rooted in the phenomenon that arises when all parameters - considered one by one - lie in operational regions. While multiple small variations occur together, they cause performance to fall in a region where the image can be misclassified.

The remainder of the paper is organized as follows: In section 2, we introduce background related to the AV hazard analysis. Section 3 summarizes the hazard identification methods we used for this study. In section 4, we identify both safety and security threats to the NN performance. Section 5 discusses robustness determination and robustness enhancement. Section 6 demonstrates our extended FTA for the traffic sign recognition network both theoretically and in practice. Section 7 concludes the study.

2. AV hazards analysis

AVs are composed of many functional modules – physical, electronic, and software. Since the most important safety issues involve crashes, FTAs provided the overall framework for hazard identification. Still, FMEA was used to provide details of mechanical and electrical component failure, STPA was used to analyze the control hierarchies, and emergent hazard analysis was used for control loop failures. Dynamic methods, including cause consequence analysis and dynamic FTs, were needed, especially for the navigation procedures, such as lane changing navigation functions and emergency response functions. A major problem has previously been that risk analysis of the NNs used for the vision systems and some control functions could not be included in the overall risk analysis. It is, therefore, necessary to extend FTA to incorporate NNs into the overall hazard identification and risk analysis.

Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep reinforcement learning (DRL) are the three most common deep learning methodologies used in AVs Grigorescu et al. (2020). CNNs are widely adopted for AV perception. The perception algorithms are the most critical module to detect objects and make image classification. Any incorrect conclusions from the perception algorithms, such as missing objects, wrong classification, and traffic sign misdetection, may lead to potentially fatal incidents. RNNs are suitable for trajectory prediction, and DRL is for path planning, for example, learning driving trajectories.

A vital impact factor for NN hazards is the selection of the training set. Any omission of essential phenomena in the training set will result in a system that may fail to recognize critical cases. This results in the strategy of using massive training sets. Waymo, for example, trains its vision systems for AVs with millions of real traffic scenarios and billions of simulated scenarios Schwall et al.

(2020). However, meeting new phenomena can lead to accidents. The existing AV incidents indicate the difficulties in developing safe AI systems. Even if a system is empirically demonstrated to be safe with millions of tests, there is no guarantee that it will not fail when new situations arise. The selection of test cases needs to consider the wide range of challenges to performance identified by explicit hazard identification.

3. Methodologies for hazard identification of AVs

The overall risk assessment for the AV was made using FMEA for the components and sequential and dynamic FTA Taylor (1975). Sequential FTs are needed to deal with the sequence and timing of responses to hazardous situations versus the dynamic development of the accident situation. If the performance of the NN only depended on independent variations in input parameters, conventional FTs with discrete events could be used, such as ”perturbation exceeds the performance threshold.” Hybrid events are needed because, in many cases, NN's performance depends on two or more continuously varying disturbance parameters. For this reason, we introduce hybrid events in FTs Taylor and Kozin (2021b) that can be interpreted as a point in a multi-parameter space belonging to the region where safety issues may occur with a rather high probability. The probability of failure is dependent on the probability of challenges to NN robustness. For example, a failure to function is often the result of deviations of two or more parameters, such as a braking force, vehicle speed, and distance to an obstacle at the start of braking. These must be determined empirically (as must failure rates in physical systems). The frequency of challenges can be observed by actually driving typical AV routes at different times and under different conditions. The NN robustness can be measured by the probability of correct image classification (i.e.,prediction accuracy) given perturbed inputs.

4. NN functional failures

One challenge of analyzing NNs is that of seeming randomness in the design of NNs. When the reverse analysis is performed on most NNs trained with a given set of test images, the features that are recognized seem to be distributed in inexplicable ways among the network layers Bengio et al. (2013).

4.1. *Safety threats to NNs*

There is a wide range of situations that can affect the performance of a neural network for AV control:

- Fundamental functional omissions (such as lack of training to recognize road diversion signs)
- Sensitivity to ambient conditions, especially low lighting
- Sensitivity to low contrast conditions
- Sensitivity to patterns (such as camouflage) or textures
- Obscuration due to intended objects hidden behind others or a blind curve or vegetation
- Obscuration by snow, blown sand, frost or ice
- Interference with well-trained recognition by extensions to the training set
- Orientation of the objects to be recognized (“pose”)
- Unusual elevation of objects to be recognized (such as lane markings on a transition to a steep hill)
- Road reflectance lights reflected from wet roads

A straightforward solution is to improve the vision system by data augmentation, sensor fusion, etc. Hendrycks and Dietterich (2019) evaluated NN robustness to common corruptions and perturbations, such as Gaussian noise, motion blur, and snow. They found that as accuracy of NN architectures improves, for instance, from AlexNet to ResNet, corruption robustness has no significant changes. All tested NN models are surprisingly vulnerable to common perturbations. Zhong et al. (2020) reported robustness of thirteen image classifiers and three object detectors to five real-world perturbations, i.e., luminance, spatial transformation, blur, corruption, and weather. Based on their results, some models outperform others for a particular perturbation, and a more complex NN architecture does not necessarily lead to a more robust model. Their results also showed that object detectors are more robust than image classifiers across various real-world perturbations.

4.2. *Security threats to NNs*

In an adversarial context, threats to the neural network could arise from:

- Training data poisoning
- NN model attack
- Adversarial example
- Physical adversarial attack
- Sensor sabotage

Training data poisoning refers to deliberately introduce false data during the training process. NN model attack takes advantage of the model flaws to fool the system. An adversarial example is small changes intentionally added to the original input that are invisible to human eyes. There is a long history of work on understanding, detecting, and mitigating impact of adversarial examples Zhang and Li (2020). Physical adversarial attack aims to fool NN models by creating perturbations on physical objects. Sensor sabotage can be conducted by using spotlights to blind cameras or laser-targeting of cameras. In this study, we focus on the practical consequences of adversarial examples on the design of AV perception models. Evaluating the security threats to NNs is a safety consideration, and adversarial examples can further be used to improve the model robustness.

5. NN robustness measures

Each of the threats to NN performance (introduced in Section 4) requires robustness testing and the probability that each threat will arise needs to be determined. For instance, the likelihood of poor illumination can be determined by driving representative routes at different times using a recording photometer.

5.1. *Robustness determination*

In a traditional risk analysis, the probability of an adverse consequence is determined by obtaining failure probabilities for components (generally by observing over a long period or looking them up in failure rate databases collected from observation). Here, failure probability for a component is derived by determining the robustness against perturbations or attacks, that is, the probability that the robustness limits will be challenged and exceeded. The probability of the AV failing must take account of redundancy in the whole AV system. The contribution of the NN to the AV FT will then be as shown in Fig. 1.

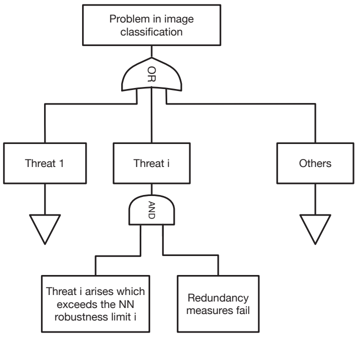

Problem in image classification
OR
Threat 1
Threat i
Others
AND
Threat i arises which exceeds the NN robustness limit i
Redundancy measures fail

Fig. 1. General template for an NN failure subtree in an FTA (for independent threats)

Functional failures of the NN can then be incorporated into fault trees in the form of multiple subtrees in an OR relationship. The probability of failure of the NN in any subtree is then:

$$P_{\text{functional failure i}} = P_{\text{robustness limit i exceeded}} \times P_{\text{redundancy measures fail}} \tag{1}$$

Robustness metrics can be developed to determine the functional range of NNs during testing. Most of the previous works propose accuracy-based metrics to measure NN robustness, i.e., the accuracy (fraction of intended targets recognized) of the NN when inputs are perturbed Hendrycks and Dietterich (2019); Zhong et al. (2020). In an adversarial setting, the minimum perturbation distance (i.e., size of deviation for a loss of function) and adversarial accuracy (i.e., the accuracy of the model when an attack takes place) are two standard metrics to evaluate NN robustness Moosavi-Dezfooli et al. (2016); Zhang et al. (2019). The AV we analyze in this study is relatively simple. Still, the perceptional module of our testing car has over 50 NNs and vision algorithms for different purposes and different navigation situations. There are tens of potential disturbances for each of these, which will affect performance, most being continuous factors rather than discrete yes/no influences. Each of these, and in many cases combinations of these, require robustness tests. Each test can involve hundreds or even thousands of test cases in order to obtain a stable measure of robustness. Laboratory testing is used for robustness determination because it seems doubtful that on-road testing could generate sufficient cases to explore the space of potential failures fully. Laboratory testing has been found to be practicable because the components can be set up and tested automatically.

5.2. *Robustness enhancement*

Data augmentation and increasing model complexity are commonly used approaches for improving NN robustness. However, robustness improvement is not uniform across perturbation types. For instance, increasing performance in the presence of Gaussian noise may cause reduced performance on other perturbations Hendrycks and Dietterich (2019). In Table 1, we identified robustness enhancements to perturbations based on perturbation types. We also map these robustness enhancements into appropriate safety strategies, i.e., inherently safe design, fail-safe design, and safety margins on components Varshney (2016). The inherently safe design aims to exclude potential hazards from the system. Fail-safe design is to keep the system in a safe state at the time of failure. Safety margins on a component are to reserve extra space for achieving safety.

Some defense mechanisms cannot enhance the robustness. For instance, Henriksson et al. (2019) used probability values from a normalized output layer of NNs as anomaly scores because they hypothesize that samples from an outlier distribution will have uncertain class results. This will not be true when the outlier is an adversarial example. Some methods (e.g., adversarial logit pairing Kannan et al. (2018) are less valuable to increase adversarial robustness. But they can be used to remarkably enhance common perturbation

Table 1. Robustness enhancements to perturbations

Perturbation type	Method/Example	Safety strategy
Natural perturbation	Multiscale networks Ke et al. (2017)	Inherently Safe Design
	Feature aggregating Xie et al. (2017)	Inherently Safe Design
	Adversarial Logit Pairing Kannan et al. (2018)	Inherently Safe Design
	Run-time out-of-distribution detection Henriksson et al. (2019)	Fail-safe design
	Histogram equalization Pizer et al. (1987)	Safety Margin
Adversarial perturbation	Adversarial training Madry et al. (2019)	Inherently Safe Design
	Randomized smoothing Lecuyer et al. (2019)	Inherently Safe Design
	Adversarial detection Smith and Gal (2018)	Fail-safe design

robustness Hendrycks and Dietterich (2019).

6. FTA for the traffic sign recognition network

The starting point and basis for safety analysis of AVs is a functional block diagram picturing all top-level functions and connections between them. A hazard identification analysis can be made by analyzing each function and indicating components/subsystems for their failure modes and effects (functional FMEA analysis). To identify more complex failure scenarios caused by several failures, degraded performances, and other internal and external factors, like weather and road conditions, causal models are needed. In this paper, we focus on FTs that, if properly analyzed, can generate a comprehensive set of hazard scenarios and provide the basis for the use of probabilistic reasoning to estimate the probabilities of the identified scenarios. However, constructing FTs for NN-controlled AVs is not a standard procedure and requires a substantial modification of classical FTA. This is due to two reasons. One is a possible malfunction of the NN and the difficulty of constructing the internal causal structure, resulting in outputting erroneous decisions. The second is that continuously changing processes (variables) influencing a vehicle's performance (possibly in combination) can result in safety issues and eventually in crashes. The second point motivates us to introduce failure events that manifest themselves when continuously evolving variables in a multi-dimensional space enter the "prohibited region". This is like in structural reliability – a failure occurs when stress exceeds the strength of the construction.

Given that the functional requirement placed on a NN is that of a simple function, such as recognizing a traffic sign, the NN can be considered a black box. The failure modes can be defined as failure to identify an image, incorrect classification of an image, or in some cases, wrong estimation of an image parameter. The NN will have a certain correct performance set and a certain level of robustness against image imperfections or distortions. The probability of failure of the NN is then the probability of the observed image lying in a domain outside the NN's capability or in a domain for which the NN is not robust. The hazard analysis can then be completed using standard methods (e.g., conventional FTA) to determine the possible causes of the inputs lying outside the NN's reliable domain. Our emphasis is placed on developing robustness measures for NNs against different types of threats.

6.1. *Problem formalism*

We propose a mathematical formalism to be able to calculate the probabilities of failure states. One of the possible hazardous events triggered by a decision made by the NN is the "Wrong classification of a traffic sign". This event can occur because of inadequate robustness of the NN, which in turn can be caused by naturally or intentionally perturbed inputs.

Robustness can be measured by the prediction accuracy given perturbed inputs. The prediction accuracy is unlikely to achieve unity, and there is a threshold of $T_r < 1$ where, if achieved, the NN decides that the image in question is recognized. Hence there is always a probability of misclassification that is greater than 0.

Assume that two variables influence whether the sign is classified correctly. One is contrast intensity, C, and the other is light intensity (i.e., brightness), L. If T_C stands for the lower limit for C, below which the sign cannot be classified correctly, we can define the event $E_C = \{E_C : c < T_C\}$ that is "too low contrast to recognize correctly". Similarly, $E_L = \{E_L : l < T_L\}$ is the event "too low lighting to recognize correctly". The third misclassification event is defined by the following condition: $E_{LC} = \{E_{LC} : (l, c) < f(c, l), c > T_C, l > T_L\}$. This should be understood as follows: while contrast and lighting both lie in the correct classification region, their combination may belong to the misclassification region. The border dividing the two regions is determined by function $f(c, l)$. Usually, this type of event occurs when variables (parameters) lie in the vicinity of the border points. That is to say, the effect of small deviations results in a failure. A possible region of misclassification is shown in Fig. 2.

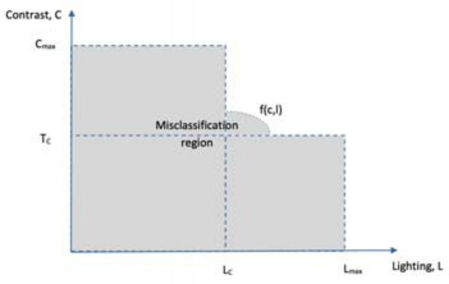


Fig. 2. Misclassification region(Conceptual)

The region of misclassification can formally be written as follows:
$\Omega = \{(c < T_C) \bigcup (l < T_L) \bigcup ((l, c) < f(c, l), c > T_C, l > T_L)\}$

As soon as the misclassification events are determined, a simple fault sub-tree can be constructed (see Fig. 3).

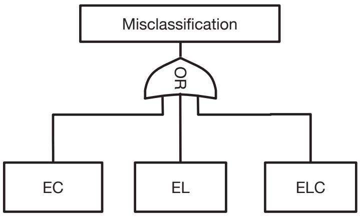


Fig. 3. A simple fault sub-tree for misclassification (with interacting threats)

Given C and L are independent random variables and their probability density functions are known, $f_C(x)$ and $f_L(y)$, the probability of misclassification $P_{\text{misclassification}}$ can be calculated:

$$P_{\text{misclassification}} = \iint_{\Omega} f_C(x) f_L(y) dx dy \quad (2)$$

6.2. *An AV example of misclassification*

To demonstrate the influence of the perturbations and their combination, we trained a 5-layer-CNN with the German Traffic Sign Recognition Benchmark (GTSRB) dataset for the traffic sign classification Stallkamp et al. (2012). The GTSRB dataset has 43 different traffic signs in various sizes and lighting conditions and is very similar to real-life data. The prediction accuracy for clean test images is 98.97%.

We adopt the algorithm from Zhong et al. (2020) to emulate the deviation of brightness and contrast, and algorithm from Goodfellow et al. (2014) to implement the FGSM attack. Fig. 4 presents: (a) a set of misclassified images with brightness=0.8. In this case, the prediction accuracy dropped to 84.8%, (b) brightness=0.6, FGSM attack with attack strength=0.2, the prediction accuracy dropped to 18.76%.

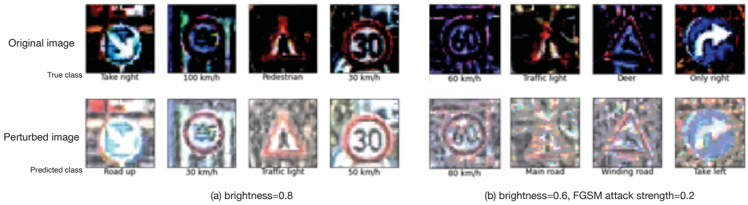
1) Brightness $X^{'} = Clip(X + l)$, where X is the original test image, l is a constant value to be added, $X^{'}$is the resulting new image, Clip is a function to make sure $X^{'}$ is in a valid pixel intensity range of [0,255] or [0,1].

2) Contrast Reduction $X^{'} = Clip((1 - c) \cdot X + c \cdot C)$, where X is the original test image, c is the contrast level, C is a constant factor.

In this experiment, we set prediction accuracy at 90% as the acceptance level of model robustness. Instead of showing the case of low brightness/contrast, we test the influence of increasing brightness and contrast reduction due to the low brightness/contrast nature of the GTSRB dataset. Fig. 5 shows prediction accuracy curves corresponding to (a) brightness variations, and (b) contrast variations. It shows that the upper limit for brightness increase is 0.66 in Fig. 5 (a), and the upper limit for contrast reduction is 0.54 in Fig.5 (b).

Then we test the combination of brightness and contrast reduction. The brightness level is set from 0.01 to 1, and contrast reduction is from 0.01 to 1, respectively. This experiment is intended to show how the small deviation of contrast and brightness affects prediction accuracy. In Fig. 6, the values of prediction accuracy are represented as colors. The lighter the color, the higher the prediction accuracy. It shows that even brightness level and contrast reduction do not exceed their upper limits (i.e., in the correct classification region). Their combination can fall into the misclassification region (i.e., prediction accuracy is lower than 90%).

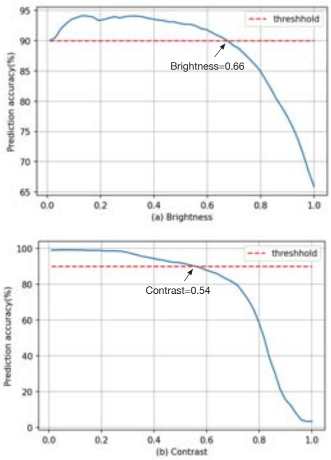
It is worth noting that contrast and lighting are just two of the challenges to the NN performance, which require a hybrid fault tree approach. In fact, almost all of the threats listed in Sections 4.1 and 4.2 have continuously varying intensities. In most cases, pairs of threats can interact to make the joint deviation worse than any single deviation alone. A particularly difficult example that was found is obscurations coupled with shadows. Some of the threats (e.g., adversarial examples) are hard for a human to understand. Methods in the field of explainable AI (XAI) can be employed to identify the influence of threats on the NN performance Zhang and Li (2020). We include more results and discussions in a technical report Taylor et al. (2021) due to the page limits.



(a) brightness=0.8

(b) brightness=0.6, FGSM attack strength=0.2

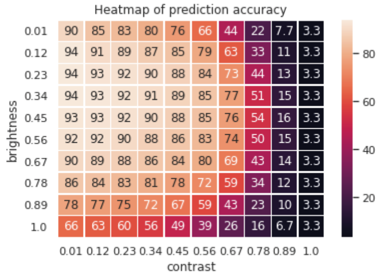
Fig. 4. Examples of misclassified traffic signs



(a) Brightness

(b) Contrast

Fig. 5. Examples of prediction accuracy curves when brightness and contrast vary



brightness \ contrast	0.01	0.12	0.23	0.34	0.45	0.56	0.67	0.78	0.89	1.0
0.01	90	85	83	80	76	66	44	22	7.7	3.3
0.12	94	91	89	87	85	79	63	33	11	3.3
0.23	94	93	92	90	88	84	73	44	13	3.3
0.34	94	93	92	91	89	85	77	51	15	3.3
0.45	93	93	92	90	88	85	76	54	16	3.3
0.56	92	92	90	88	86	83	74	50	15	3.3
0.67	90	89	88	86	84	80	69	43	14	3.3
0.78	86	84	83	81	78	72	59	34	12	3.3
0.89	78	77	75	72	67	59	43	23	10	3.3
1.0	66	63	60	56	49	39	26	16	6.7	3.3

Fig. 6. Prediction accuracy matrix with small deviation of brightness and contrast in combination

7. Conclusion

From this study, it became clear that detailed hazard identification can be made for AVs, including both hardware and NN components. The procedure is:

(1) Complete the overall high-level hazard identification using an FTA approach.
(2) Identify the functional failures of the NNs which contribute to the overall FTA.
(3) Identify the challenges which can cause the NN functional failure, e.g., using the checklist in Sections 4.1 and 4.2.
(4) Determine the robustness of the NNs when challenged by perturbations of single parameters or by the combination of parameter perturbations via testing NN performance and making a heatmap as in Fig. 6.
(5) Determine the probability of the occurrence of parameter perturbations.
(6) Incorporate the contribution of NNs into the FTA using the templates given in Fig. 1 and Fig 3.

A further conclusion is that a detailed hazard assessment can be essential in determining the scope of controller component testing.

One of the key findings of the studies described here is that safety and security analysis becomes much easier when an integrated approach is taken. There are many potential cases where individual controller components (e.g., NN for image recognition) can fail due to an attack, but where accidents can be avoided by other components taking over. This is particularly an issue where there is a possibility of a crash and poor visibility conditions. In these cases, lidar and radar provide less informative but more robust detection of hazards.

Safety in AVs is not ensured by hazard detection alone. It is not safe, for example, to simply stop the vehicle when a crash potential is detected in fast-moving traffic. Policies, strategies, plans, and algorithms for safe state recovery are needed. Our next challenge, then, is to carry out hazard identification and risk assessment on these recovery plans.

References

Banerjee, S. S., S. Jha, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer (2018). Hands off the wheel in autonomous vehicles?: A systems perspective on over a million miles of field data. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 586–597. IEEE.

Bengio, Y., G. Mesnil, Y. Dauphin, and S. Rifai (2013). Better mixing via deep representations.

In *International conference on machine learning*, pp. 552–560. PMLR.

Goodfellow, I. J., J. Shlens, and C. Szegedy (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Grigorescu, S., B. Trasnea, T. Cocias, and G. Macesanu (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics 37*(3), 362–386.

Hendrycks, D. and T. Dietterich (2019). Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint*.

Henriksson, J., C. Berger, M. Borg, L. Tornberg, S. R. Sathyamoorthy, and C. Englund (2019). Performance analysis of out-of-distribution detection on various trained neural networks. In *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 113–120. IEEE.

ISO 26262:2011 (2011, November). Road vehicles – Functional safety. Standard, International Organization for Standardization.

ISO/PAS21448:2019 (2019, January). Road vehicles — Safety of the intended functionality. Standard, International Organization for Standardization.

Johnson, C. (2018). The increasing risks of risk assessment: On the rise of artificial intelligence and non-determinism in safety-critical systems. In *the 26th Safety-Critical Systems Symposium*, pp. 15. Safety-Critical Systems Club York, UK.

Kalra, N. and S. M. Paddock (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice 94*, 182–193.

Kannan, H., A. Kurakin, and I. Goodfellow (2018). Adversarial logit pairing. *arXiv preprint*.

Ke, T.-W., M. Maire, and S. X. Yu (2017). Multigrid neural architectures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6665–6673.

Lecuyer, M., V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana (2019). Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE.

Madry, A., A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu (2019). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Moosavi-Dezfooli, S.-M., A. Fawzi, and P. Frossard (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582.

Pizer, S. M., E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld (1987). Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing 39*(3), 355–368.

Schwall, M., T. Daniel, T. Victor, F. Favaro, and H. Hohnhold (2020). Waymo public road safety performance data. *arXiv preprint arXiv:2011.00038*.

Shalev-Shwartz, S., S. Shammah, and A. Shashua (2017). On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*.

Smith, L. and Y. Gal (2018). Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*.

Stallkamp, J., M. Schlipsing, J. Salmen, and C. Igel (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks 32*, 323–332.

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Taylor, R. (1975). Sequential effects in failure mode and fault tree analysis. *Reliability and Fault Tree Analysis*.

Taylor, R. and I. Kozin (2021a). Design for emergent safety problems in handbook of engineering systems design. In-press, Springer.

Taylor, R. and I. Kozin (2021b). Hybrid fault trees for continuous systems. Unpublished manuscript.

Taylor, R., J. Zhang, I. Kozin, and J. Li (2021). Safety And Security Analysis for Autonomous Vehicles. `https://github.com/safe-ai-tech/Reports_Papers`. [Technical report].

Varshney, K. R. (2016). Engineering safety in machine learning. In *2016 Information Theory and Applications Workshop (ITA)*, pp. 1–5. IEEE.

Xie, S., R. Girshick, P. Dollár, Z. Tu, and K. He (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500.

Zhang, H., Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan (2019). Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR.

Zhang, J. and J. Li (2020). Testing and verification of neural-network-based safety-critical control software: A systematic literature review. *Information and Software Technology*, 106296.

Zhong, Z., Z. Hu, and X. Chen (2020). Quantifying dnn model robustness to the real-world threats. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 150–157. IEEE.

Paper 3

Robustness evaluation for safety-critical systems utilising artificial neural network classifiers in operation: A survey,

Jin Zhang, Jingyue Li, and Josef Oehmen.

In review, Manuscript submitted to the International Journal of Engineering Application of Artificial Intelligence.

Robustness Evaluation for Safety-Critical Systems Utilizing Artificial Neural Network Classifiers in Operation: A Survey

Jin Zhang[a,b,c], Jingyue Li[a,*], Josef Oehmen[b]

[a] *Computer Science Department, Norwegian University of Science and Technology, Trondheim, Norway*
[b] *Section of Engineering Design and Product Development, Department of Civil and Mechanical Engineering, Technical University of Denmark, Denmark*
[c] *School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China*

Abstract

Artificial neural networks (ANNs) have become increasingly prevalent in various industries, with applications in safety-critical domains such as image recognition, medical diagnosis, and autonomous vehicles. Assessing the robustness of ANN classifier-based safety-critical systems (ANN-SCSs) in operation is crucial, as model performance can be compromised when input data deviate from the training data. While existing reviews provide useful insights into the robustness research considering ANN models, a structured discussion on ANN-SCS robustness evaluation in operation is still missing. This study aims to systematize how to evaluate the robustness of ANN-SCSs on different granularity levels of a system in operation, classify evaluation methods and metrics, and identify challenges and gaps for future research. We analyzed five system-level, 15 ANN model-level, and eight input-level studies that focus on evaluating the robustness of ANN-SCS in operation. Our results provide a summary of nine ANN-SCS robustness definitions at the system, ANN model, and input levels, respectively; present a classification of eight major evaluation approaches and 30 metrics; and identify three categories of research gaps, namely, defining abnormal conditions, determining acceptable performance levels, and obtaining labeled data.

Keywords: artificial neural network, safety-critical systems, robustness evaluation, operation.

1. Introduction

Artificial neural network (ANN) classifiers are being used in various safety-critical application sectors, such as autonomous cars, aircraft control systems, smart grids, and healthcare services (Zhang and Li, 2020). The increased complexity and connectivity of these systems can make them more fragile to disturbances and attacks. If a failure occurs in an ANN model, it may progressively trigger physical damage, cause harm to people, and lead to further economic loss and/or environmental or reputational damages. Several incidents have been specifically tied to the robustness of ANN classifier-based safety-critical systems (ANN-SCSs), such as fatal incident involving a Tesla self-driving car in which the deployed model failed to differentiate between a white truck and the bright sky (Boudette, 2017). IBM Watson for Oncology is another well-known example of an ANN-SCS failure, where it frequently gave unsafe and erroneous cancer treatment advice to patients (Ross and Swetlitz, 2018). The potential consequences of damages necessitate enhancing the robustness of ANN-SCSs and considering effective evaluation methods for the general robustness of any type of machine learning (ML) system (Chen et al., 2022).

Unless the robustness of ANN-SCS can be measured effectively, there is no path to structured robustness improvement, as the disparity between the desired performance and the actual performance remains unidentified. Existing metrics, evaluation methods, and challenges surrounding the robustness of ANN models have been discussed in several surveys (Rawat and Wang, 2017; Thomas and Tabrizi, 2018; Akhtar and Mian, 2018; Huang et al., 2020). However, the surveys' focuses are quite diverse, targeting adversarial robustness (Thomas and Tabrizi, 2018; Akhtar

[*] Corresponding author
Email addresses: jin.zhang@ntnu.no (Jin Zhang), jingyue.li@ntnu.no (Jingyue Li), jooehm@dtu.dk (Josef Oehmen)

Preprint submitted to Engineering Applications of Artificial Intelligence

and Mian, 2018), corruption robustness (Drenkow et al., 2021), or distributional robustness (Kumar et al., 2019). First, this diversity makes it challenging to produce a unifying taxonomy, evaluation metrics, measurement techniques, and evaluation framework for real-world applications. Second, most evaluation metrics and methods are designed for the model development stage (Zhang et al., 2020; Huang et al., 2020; Drenkow et al., 2021; Mohseni et al., 2022) and thus cannot be directly applied to ANN-SCSs in operation. **The robustness challenges for operationalizing ANN-SCSs require unique solutions considering harsh operational (including industrial) environments (Shankar et al., 2022).** Besides, ANN models are often a small part of a large system (Sculley et al., 2015; Li et al., 2022). The failure of an ANN model may propagate to or be mitigated by other components, including dedicated backup systems (Peng et al., 2020). Thus, the robustness of ANN systems should be measured on three levels: 1) ANN-related input data; 2) the ANN model itself; and 3) the whole system, including other relevant components. Understanding how the failure of ANN models affects the robustness of the whole system is essential. To our knowledge, no existing studies have summarized the robustness of ANN-SCSs on different granularity levels of a system in operation.

Regarding ANN-SCSs' robustness in operation, there have been some efforts in the literature. For example, Kumar et al. presented a joint taxonomy of intentional and unintentional robustness challenges for ML systems (Kumar et al., 2019), and Mohseni et al. (Mohseni et al., 2022) reviewed dependability limitations for ML algorithms and methods for improving model performance and robustness.

Furthermore, there have been a number of conceptualizations of system-level robustness and its relationship to component robustness without specifically focusing on ANN safety components, e.g. the impact of centralization vs decentralization in decision-making architectures (Boss and Gralla, 2023), the robustness of complex system architectures specifically against catastrophic cascading failures (Potts et al., 2020), or the non-safety impact of designing robust systems and implications for, e.g., flexibility and adaptability to future needs (Ross et al., 2008). However, a concise discussion of the robustness evaluation of ANN-SCSs in operation is still missing.

To fill this research gap, this study aims to answer the following **main research question (MRQ): What are the perceptions and practices of robustness evaluation in ANN-SCSs in operation?**

We refine this MRQ into three sub-research questions as follows:

- RQ1: What are the definitions of ANN-SCSs' robustness in operation?
- RQ2: What metrics and methods are used to measure the robustness of ANN-SCSs?
- RQ3: What are the challenges of measuring ANN-SCSs' robustness in operation?

RQ1 aims to bridge this gap by extracting and analyzing various robustness definitions to provide the basis for measuring robustness. RQ2 focuses on reviewing methods and metrics employed to assess the robustness of ANN-SCSs at the system, ANN model, and input levels, aiming to establish a consistent evaluation framework for researchers and practitioners. RQ3 identifies remaining challenges in evaluating ANN-SCSs' robustness in operation.

This study focuses on SCSs that use ANNs for classification tasks. Figure 1 illustrates the categories of ML methodologies and our focus. By concentrating on classification tasks, we can tackle a substantial portion of real-world use cases and provide valuable insights to a wide audience. Although ANNs are only one of the machine learning (ML) approaches for SCSs, they serve as a representative example to demonstrate the challenges we address.

We present a comprehensive analysis of the different methods and metrics proposed in the literature over the past five years for conducting robustness evaluation of ANN-SCSs based on well-established guidelines (Molléri et al., 2016). We examine 23 studies (five system-level, ten ANN model-level, and eight input-level) and offered a summary of ten ANN-SCS robustness definitions, and we classify eight major evaluation approaches and 30 metrics. For example, at the system level, one robustness definition focuses on maintaining system reliability under abnormal conditions. An evaluation approach may involve injecting faults into the system to assess its response. At the ANN model level, a robustness definition may involve the model's ability to maintain accurate predictions despite adversarial attacks. A possible evaluation approach could involve the use of adversarial training and testing. A robustness definition at the input level could encompass the system's ability to manage noisy or corrupted inputs effectively. One evaluation approach might involve measuring test coverage to assess how well the representation of data inputs captures various input scenarios and conditions. Furthermore, we identify three categories of research gaps, namely, defining abnormal conditions, determining an acceptable level of performance, and obtaining labeled data.

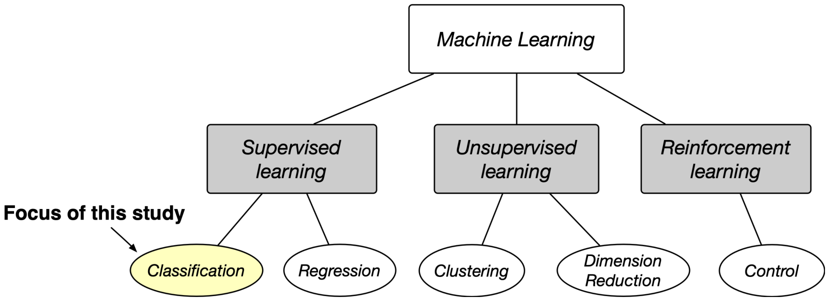


Figure 1: Machine learning categories (Zhang et al., 2020) and our focused task.

This work is the first attempt to provide a structured discussion on ANN-SCS robustness evaluation in operation, covering aspects like definitions, metrics, methods, challenges, and future directions. Moreover, our paper investigates the robustness of entire ANN-SCSs, considering different levels rather than just focusing on individual components or models.

The rest of the paper is organized as follows. In Section 2, we provide an overview of standards related to ANN-SCSs and the relations between ANN robustness and SCS risks. Section 3 describes related work. Section 4 introduces the research methodology used in this work. Section 5 presents the results of the research questions. Section 6 discusses the results. Section 7 contains the conclusions and outlines our future work.

2. Background

ANNs are increasingly utilized in various safety-critical sectors, including transportation, healthcare, finance, and the military, to enhance the performance and safety of systems. For instance, ANNs have been applied for predicting wing deformations in aircraft control systems (Yasuda and Yang, 2022), object detection, lane-keeping, and decision-making in autonomous vehicles (AVs) (Grigorescu et al., 2020), as well as image recognition, target tracking, and decision-making in unmanned aerial vehicles (UAVs) (Kyrkou and Theocharides, 2019). Ensuring that ANN classifiers maintain accurate results even with noisy data is vital for SCSs.

2.1. Standards Related to Safety-critical Systems Containing AI Elements

Amershi et al. (2019) proposed nine stages of the ML workflow. We grouped the ML lifecycle stages into two phases, namely, “before deployment” and “in operation” (as shown in Figure 2). Robustness evaluation is critical and complements both the “before development” and “in operation” phases. Before deployment, the ANN classifier is normally systematically evaluated on different robustness benchmarks to ensure model performance on known inputs (Hendrycks and Dietterich, 2019; Croce et al., 2020). In operation, the ANN is integrated into a system that operates in an environment that may differ from the testing environment (e.g., low-quality of input data (Javier et al., 2019) or unseen attacks on the model (Zhu et al., 2021)). This means that a model benchmarked before deployment may not guarantee the same level of performance in a real-world environment since the robustness evaluation during the development stage may not sufficiently cover all realistic threat models. How to deal with the risks of the system’s deployment has been the subject of many regulations, such as:

- European medical devices, including those using AI, need to comply with good clinical practice (ISO 14155) and undergo “clinical investigations” (Beede et al., 2020a).
- For non-medical systems using AI, such as facial recognition (BUNDESAMT, 2004), and self-driving car (UK Government, 2022; Schwall et al., 2020), field trials are one of the recognized means of comparing and assessing the robustness at the system level.

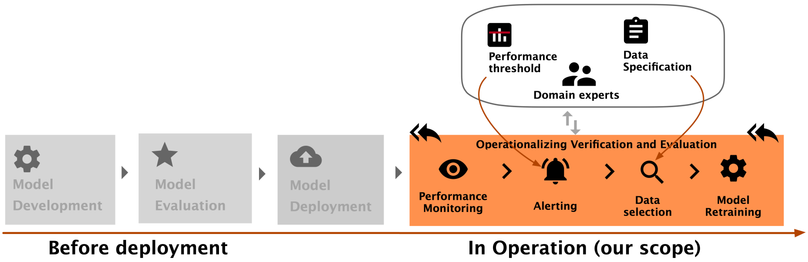


Figure 2: Paper scope.

- In the field of unmanned aircraft systems (UASs), international standard ISO 21384-3 has defined minimum covering elements for robustness evaluation of unmanned aircraft systems, which include "procedures to evaluate environmental conditions before and during the mission (i.e., real-time evaluation), procedures to cope with unintended adverse operating conditions, and contingency procedures to cope with abnormal situations."
- The Food and Drug Administration (FDA) of the United States identified the necessity for enhanced techniques to evaluate algorithm robustness in the face of evolving clinical inputs and conditions in the "Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device Action Plan" (US Food and Drug Administration, 2019).
- Developers of automated driving systems in the automation industry validate their laboratory testing by conducting controlled field testing on public roads (Schwall et al., 2020; Webb et al., 2020).

2.2. *Relations between ANN Robustness and SCS Risks*

The international standard on trustworthiness in AI (ISO/IEC TR 24028-1) states that an AI system's ability can be assessed based on **robustness, reliability, and resilience**. Robustness is "*a system's ultimate ability to maintain its performance level under any circumstances, including external interference or harsh environmental conditions*arXiv preprint (ISO/IEC TR 24028-1). The National Institute of Standards and Technology (NIST) AI Risk Management Framework (NIST, 2023) highlighted the discrepancy between laboratory and real-world risk assessment, i.e., measuring AI risks in a laboratory or controlled environment could provide valuable pre-deployment insights. Still, these measurements may not accurately reflect the risks that occur in operational settings. The robustness of an ANN-SCS can affect other adjacent system attributes, such as safety (e.g., for autonomous vehicles (Boudette, 2017)), security (e.g., access control (Oberhaus, 2017)), and reliability (e.g.,commercial facial recognition software (Snow, 2018)). In Table 1, we present examples of real-world incidents due to the lack of robustness in ANN models.

3. Related Work

Despite numerous studies on the robustness of ANN models, e.g., (Bastani et al., 2016; Yu et al., 2019; Buzhinsky et al., 2021), there has been no dedicated study summarizing the methods and corresponding metrics for assessing the robustness of these models in operation.

Reviews on robustness evaluation of ML model in general or a single ANN model in a lab environment. França et al. (2021) reviewed state-of-the-art techniques for evaluating the robustness of DNN models. Their work focuses on methods employed to test the robustness of image classifiers associated with AVs. For example, fuzz testing (Xie et al., 2019) is reported as one methodology to test robustness as it employs invalid or unexpected inputs in the testing process. Alternatively, fault injection can be used to test robustness. In the survey on ML testing (Zhang and Li, 2020), several fault injection-based methods are identified to simulate hardware errors of AVs to evaluate their robustness.

In controlled environments like laboratories, robustness has been extensively explored and evaluated in the context of adversarial ML (Rawat and Wang, 2017; Thomas and Tabrizi, 2018; Akhtar and Mian, 2018; Huang et al., 2020)

Table 1: Examples of real-world incidents caused by lacking robustness in operation.

Case	Incident Description	Cause	Affected Attribute
1	IBM Watson for Oncology frequently gave unsafe and erroneous cancer treatment advice to patients(Ross and Swetlitz, 2018).	Lacking distributional robustness: a few synthetic cancer patient data were used for training instead of real patient data .	Safety
2	Apple's facial recognition ID system was fooled by 3D-printed masks (Oberhaus, 2017).	Lacking adversarial robustness: the anti-spoofing neural network only considers cosmetic changes, wearing a scarf, or the presence of glasses on the face.	Security
3	Tesla autopilot failed to recognize a white truck against a bright sky (Boudette, 2017).	Lacking corruption robustness: Image contrast	Safety
4	Amazon's facial recognition software mistakenly identified members of the U.S. congress (Snow, 2018).	Lacking distributional robustness: the facial identification system demonstrated better performance for lighter-skinned faces but encountered difficulties in recognizing darker-skinned faces.	Reliability

to defend against an adversary who will attack the system to test ML algorithms' worst-case robustness, and to measure the improvements in ML algorithms towards human-level abilities (Carlini et al., 2021). Carlini et al. (2021) developed a checklist of common evaluation pitfalls when evaluating adversarial robustness. Examples of the pitfalls include "not using the right attack method," as the robustness of an ANN to one attack method may not necessarily indicate its robustness to other attack methods. Another example is "not considering the effect of different types of natural perturbations," as the robustness of a ANN model can be affected by various types of noise, such as rotations and translations (Engstrom et al., 2019), common corruptions and perturbations (Hendrycks and Dietterich, 2019), and Gaussian noise (Gilmer et al., 2019). Non-adversarial robustness has received disproportionately less attention than adversarial robustness. Drenkow et al. (2021) performed a systematic review to measure computer vision's non-adversarial robustness. The identified papers were categorized based on different robustness tactics, i.e., architecture, data augmentation, and optimization robustness. Their work also pointed out the absence of formal definitions of robustness in operation.

Most methods in this direction are not directly applicable in the operational context, as they require labeled data for robustness evaluation. Labeled data may be inaccessible or delayed due to the high cost of labeling cost in operation. Besides, these techniques are typically employed to evaluate the robustness of a single ML model (e.g., an ANN classifier). However, suitable methods that can measure robustness at both single model levels and the entire system level must be investigated in the operational context.

Reviews on robustness evaluation in operation. Microsoft researchers discussed robustness in several application domains (Kumar et al., 2019). Their work (Kumar et al., 2019) provides a unified taxonomy and framework that covers both intentional and unintentional failures of an ANN model. The classification in (Kumar et al., 2019) aimed to summarize all possible risks associated with ML systems in one place. Although their work gives an overall view of ML robustness risk in operation, they didn't identify metrics and methods that apply to each operating environment.

Assessment methods for the robustness of cyber-physical systems. ANN or ML-based SCSs are a specific subset of safety risk assessment methods for cyber-components in cyber-physical systems (e.g., industrial control systems, ICSs). Existing methods stress the need for a system-level approach that considers the role and interaction of physical, cyber-physical, and cyber components (e.g., (Carreras Guzman et al., 2020; Guzman et al., 2021)). These are embedded in a broader literature of designing SCSs, driven by specific models of accidents causation that reflect

Table 2: Chronological comparison of focuses of previous surveys.

Year	Survey	Studying Definitions	Studying Metrics and Methods	Identifying Challenges	Operation
2019	Kumar et al. (2019)				✓
2019	Carlini et al. (2021)	✓	✓	✓	
2020	Zhang et al. (2020)		✓	✓	
2020	Huang et al. (2020)		✓	✓	
2021	França et al. (2021)		✓	✓	
2021	Drenkow et al. (2021)	✓	✓	✓	
2022	Mohseni et al. (2022)		✓	✓	
	Our survey	✓	✓	✓	✓

current real-life complexity due to technology changes, accident nature, new hazards, decreased tolerance for even single accidents, increased system complexity, increased complexity of human-automation interaction, and evolving safety standards and public views (e.g., Leveson (2004)). The complexities addressed in safety and robustness assessments in the cyber-physical domain as well as in the general safety of complex systems domain are currently only incompletely reflected in the conversations on the robustness of ANN-based SCSs.

Additionally, no current review paper explicitly organizes definitions, metrics, and methods specifically targeting the robustness evaluation of ANN-SCSs in operation. The distinctions between the focus of our survey and existing surveys are illustrated in Table 2.

4. Research Methodology

We conducted our survey-based research utilizing the methodology proposed by Molléri et al. (2016), which involves defining research questions, designing a collection strategy, and analyzing and reporting findings. The research steps are shown in Figure 3.

Data collection. To answer the research questions, we searched papers published in digital libraries, including the ACM Digital Library, IEEE Xplore, SpringerLink, Scopus, Web of Science, and Google Scholar. To identify relevant robustness definitions, we also searched the Norwegian portal of international standards (Standard.no), which provides free access to us. Standard.no contains all active IEC (International Electrotechnical Commission) and ISO (International Standards Organization) standards, and some Norwegian standards in full text. The search terms as below were selected according to their relevance to the research questions and the scope to explore the metrics and methods for evaluating ANN-SCSs in operation. The terms "robust*," "classification," "artificial neural network," and "operation" were used as the main search terms, as they are central to the investigated topic. In addition, the term "safety-critical system" and the three typical safety-critical systems, namely, unmanned aircraft system (UAS), medical system (MS), and autonomous driving system (ADS), were included to ensure that relevant studies in the context of SCS were included in the final analysis.

(robust) AND (classification) AND (deep learning OR deep neural network OR artificial neural network) AND (operation OR industry) AND (safety-critical system OR autonomous driving systems OR medical system OR unmanned aircraft system)*

Inclusion and exclusion criteria. The inclusion criteria are:

- Studies that address robustness from a conceptual point of view, i.e., provide a concrete definition of robustness;
- Studies that propose metrics to measure ANN-SCS robustness;
- Studies that perform an explicit robustness evaluation;
- Studies that focus on robustness in operation (as opposed to robustness evaluation before deployment).

The exclusion criteria are:

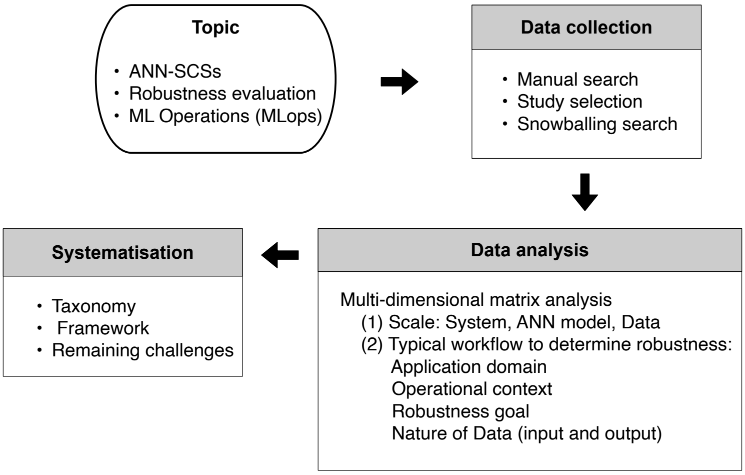


Figure 3: Research steps overview.

- Studies published before 2018;
- Studies that are not peer-reviewed;
- Studies that are not in the English language.

We included studies published between 2018 and 2022. Studies published before 2018 were excluded since the concept of MLOps (machine learning operations)[1] gained significant traction and recognition around 2018-2019 (Treveil et al., 2020). Our survey focuses on robustness evaluation for ANN-SCSs in operation, and this exclusion criterion helps ensure the relevance of the included papers.

Filtering process. The manual search returned 298 papers. After reading the title and abstract, we excluded 216 papers that were obviously irrelevant. After reading the full content of the remaining ones (i.e., 82 papers), we excluded 69 papers. To achieve a comprehensive coverage of relevant studies, we supplemented the manual search with the backward and forward snowballing procedure, adhering to the procedure guidelines proposed by Wohlin (2014). After examining the remaining 13 papers, we identified an additional ten related papers.

Data analysis To answer RQ1, constant comparison (Glaser and Strauss, 2017) was adopted to identify similarities and differences in the robustness definitions we found. Constant comparison is used in qualitative data analysis by continually comparing and contrasting the data. It is used to identify patterns, themes, and relationships in the data and to develop an understanding of the data systematically and rigorously. To answer RQ2, we followed the typical workflow to assess robustness described in international standard ISO/IEC TR 24029-1. More precisely, for each selected paper, we identified its application domain, robustness goals, operational context, data source, metrics, and methods to measure robustness. To answer RQ3, we extracted methods and metrics-related challenges for each selected paper. Thematic analysis (Cruzes and Dyba, 2011) was then used to analyze the extracted information. To ensure the accuracy of the data analysis, two rounds of data analyses were performed for each RQ, and minor corrections were made during the second round.

[1] MLOps combines the best practices from software development (DevOps) and data engineering with ML, streamlining and managing the machine learning lifecycle from development to deployment and monitoring in production.

Table 3: Definitions of robustness in literature.

Level	Ref.	Definition of robustness
System	IEEE Std 610.12	[Robustness] is the degree to which a system or component can function correctly with invalid inputs or in stressful environmental conditions.
	ISO/IEC TS 5723	[Robustness] is the ability of a system to maintain its level of performance under a variety of circumstances.
	ISO/IEC TR 24029-1	[Robustness] is the ability of an AI system to maintain its performance level under any circumstances (domain change, hardware failure, etc.).
	ISO 26262	[Robustness] provides safe behavior at boundaries (corner case, core event, extreme case).
ANN model	Goodfellow et al. (2015)	[Robustness] is the classifier's worst-case performance on small, additive, classifier-tailored perturbations.
	Diochnos et al. (2018); Szegedy et al. (2013)	An ANN classifier is robust if it achieves correct classification on a testing sample that is "close" to a training sample.
	Hendrycks and Dietterich (2019)	Robustness is the classifier's average-case performance on small, general, classifier-agnostic corruptions or perturbations.
	ISO/IEC TR 24028-1; Zheng et al. (2016); Wang et al. (2021)	An ANN classifier is robust if it achieves "consistent" classification (i.e., prediction accuracy) on known and unknown inputs as long as the unknowns are not too different from the known inputs.
Data	Zhong et al. (2021)	An original data point is strong (robust) concerning the ANN classifier being tested if the accuracy of its neighboring points exceeds a predefined threshold.

5. Results of Research Questions

5.1. Results of RQ1: Definitions of ANN-SCS Robustness in Operation

Despite the popularity of the term "robustness" in literature, a limited portion of papers addresses this system attribute from a conceptual point of view. We identified nine definitions from scientific papers and industry standards. Table 3 summarizes the identified robustness definitions at different granularity levels, i.e., the system, an ANN model, and input data levels. More specifically, we use the results of the constant comparative technique to analyze the focus of the identified definitions and explain their links and relationships (what needs to be measured) with robustness evaluation methods and metrics (how it can it be measured).

Robustness definitions at the system level. In an SCS, each component is designed to carry out a specific function or set of functions that are essential to the overall operation of the system. Identified definitions at the system level are generally concerned with the system's ability to maintain its performance and function correctly when facing exceptional or unforeseen conditions. These conditions can include unavailability of resources, communication failures, environmental disturbances (IEEE Std 610.12), invalid inputs (IEEE Std 610.12), and changes in the system's operating conditions (ISO/IEC TR 24029-1). Most definitions state that robustness requirements must be met under any circumstances. A recent study (ISO/IEC TS 5723) on the trustworthiness of systems changed the word "any" to "a variety of," which eliminates the ambiguity in understanding the requirement to meet robustness under specific circumstances and allows for a more focused approach rather than an all-encompassing definition. The specific focus of each definition varies. For example, some definitions emphasize that a system's performance must be *stable* (ISO/IEC TR 24029-1) and remain on an *acceptable level* (IEEE Std 610.12), while the definition in ISO 26262 emphasizes the significance of robustness in preserving *safe* behavior under a variety of operating conditions.

Robustness definition at the ANN model level. All the identified definitions at the ANN model level refer to the model's ability to maintain its performance when faced with inputs or conditions that differ from what it was trained

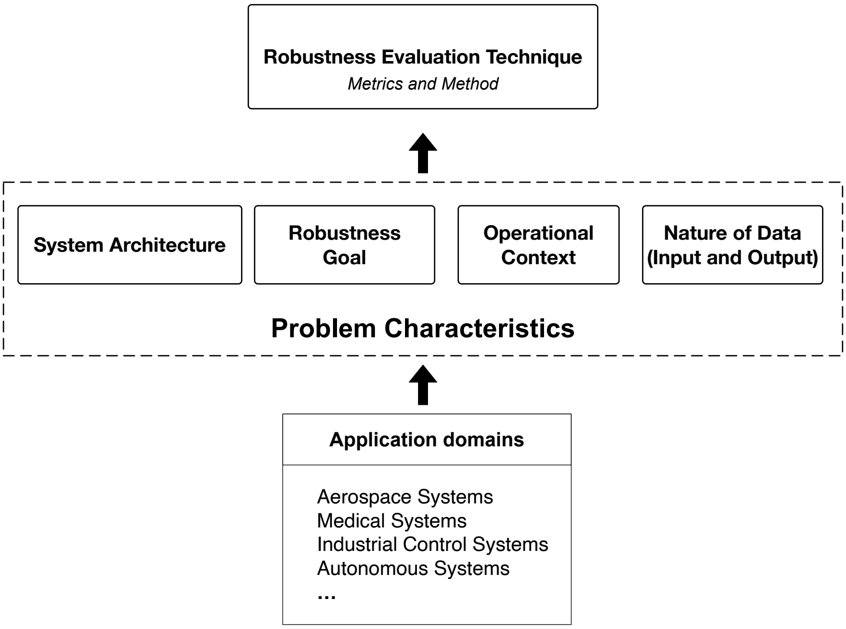


Figure 4: Key components associated with a robustness evaluation technique for ANN-SCSs.

on. The most commonly studied input deviations include malicious perturbations (i.e., an input that incorporates a subtle, deliberate perturbation with the aim of causing misclassification by an ANN) (Goodfellow et al., 2015) and natural perturbations (Hendrycks and Dietterich, 2019). Malicious perturbations are typically designed to be invisible, while natural perturbations have no such constraint. Natural perturbations are noises that exist in natural environments and may be noisier and more noticeable than malicious perturbations. Some studies (Szegedy et al., 2013; Goodfellow et al., 2015; Diochnos et al., 2018) emphasizes the model's robustness to adversarial examples, while (Hendrycks and Dietterich, 2019) focuses on the model's robustness to natural perturbations. In addition to adversarial robustness and robustness to natural perturbations, a specific concern in operation is the impact of the discrepancy between the training data distribution and the operational distribution (referred to as distributional shift (ISO/IEC TR 24028-1)) on the model's performance.

Robustness definition at the data level. An original data point can be considered robust concerning a ANN classifier being tested if the accuracy of its neighboring points exceeds a predefined threshold (Zhong et al., 2021). This means that the model's performance on inputs similar to the original data point (i.e., its neighbors) is above a certain level of accuracy. This definition of robustness is closely related to the second definition (Diochnos et al., 2018; Szegedy et al., 2013) in the category of the ANN model level, which states that a robust model should be able to correctly classify inputs that are similar to the inputs it was trained on. In this case, the "similar inputs" are defined as the neighbors of the original data point, and the threshold represents the acceptable performance level. Note that robustness is a relative concept, and the threshold can differ depending on the application and use case.

Key components associated with robustness evaluation for ANN-SCSs. Based upon the existing definitions identified in the literature, we summarized that *ANN robustness in operation is the ability of an ANN-SCS to maintain a stable performance at an acceptable level even in the face of unexpected or adverse conditions to continue to function correctly.* First, the performance must be stable. The specific threshold for what is considered a minor deviation must be determined for each SCS. Second, the performance must always remain at least at an acceptable level. This aspect raises the need for dynamic robustness evaluation to guarantee that the system continues to perform correctly in operation. An acceptable performance level does not require that the system works at its best, as it would in a

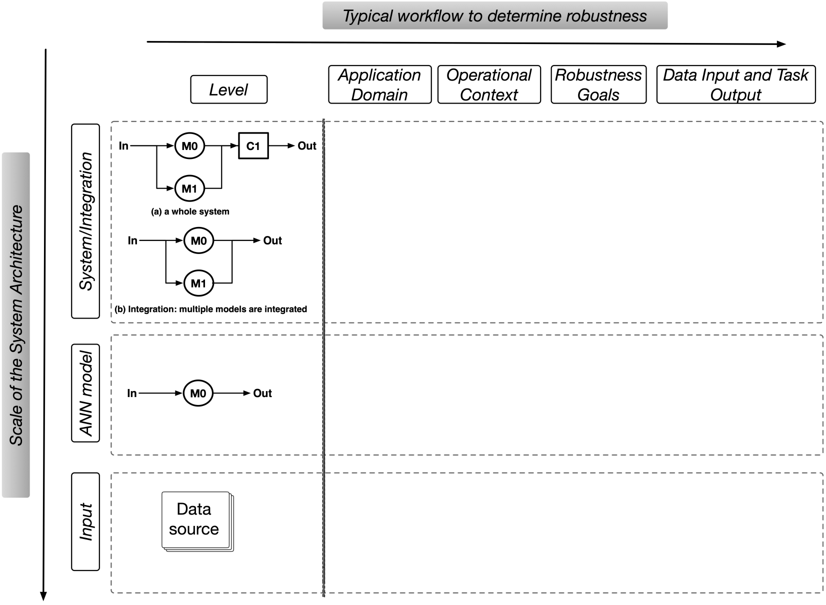


Figure 5: Template for the proposed multidimensional framework.

reference situation, but the performance must stay above a certain threshold.

The constant comparative analysis of the robustness definitions also revealed that most of the existing robustness definitions and corresponding evaluations include several factors, i.e., the scale of the system architecture, the operational context, the nature of the data (covering both input and output), etc. The determination of these factors often depends on the specific application domain in which the evaluation of robustness is required. Figure 4 illustrates the key components related to a robustness evaluation technique for ANN-SCSs.

Answer to RQ1: The existing studies define robustness in operation on the system, model, and data levels. Five essential elements for evaluating ANN-SCS in operation are identified: system architecture, application domain, operational context, robustness goal, and the nature of data.

5.2. Results of RQ2: Methods and Metrics to Measure ANN-SCS Robustness in Operation

5.2.1. Proposed Framework

Based on the different levels of robustness definitions and the five elements of robustness evaluation, which were explained in Section 5.1, we proposed a framework that adopts a hierarchical conceptual approach to categorize and illustrate existing methods and metrics in evaluating the robustness of ANN-SCSs (see Figure 5).

Scale of the system architecture. The system architecture scale refers to the level at which the ANN-SCS is: 1) At the system level, where the ANN-SCS is evaluated as a whole within its operational environment. In this context, multiple models can be combined either serially, as in a traffic light perception task that requires both detection and recognition or in parallel, where several alternative models address a problem. For example, in the camera perception component of Apollo, a cutting-edge autonomous vehicle system, four models are associated with obstacle detection (Peng et al., 2020). Developers must decide which of these four models to employ when identifying obstacles. 2) The ANN model level involves evaluating a single ANN model independently. 3) The input level is where the input data utilized in operation are evaluated.

Application domains and context. Recognizing the application domain and context is crucial for selecting suitable metrics and methods to assess a model or system's robustness. SCS application domains analyzed in this study include ADSs, MSs, and UASs.

Robustness goals. These include performance requirements such as maintaining consistent performance when dealing with altered inputs, generalizing effectively within and across domains, and resisting adversarial attacks (Drenkow et al., 2021).

Data input and task output. This paper specifically focuses on using ANNs for classification tasks, with image data as the primary input and class prediction as the primary output, and thus limits its discussion to data and task outputs related to this particular focus.

In the following sections, we present how the aforementioned factors influence the selection of methods and metrics.

5.2.2. Methods and Metrics of Robustness Evaluation at the System Level

Robustness measurement techniques vary at the system level, with many being application domain-specific. Table 4 summarizes studies, measurement methods, and metrics for system-level robustness evaluation in operation.

Table 4: Methods and Metrics for evaluating the robustness of ANN-SCSs in operation (system level).

SCS Domain	**Operational Context**	**Robustness Goal**	**Method**	**Metrics**	**Ref.**
ADS	End-to-end steering	Min. MSE of steer angle in the presence of adversarial examples and synthetic noisy input	Input-output evaluation	Likelihood-based surprise adequacy, Distance-based surprise adequacy	Kim et al. (2019)
				Attacking strength, Average angle error, Percentage of frames whose angle error exceeds a predefined threshold	Zhou et al. (2020)
	Object perception	Ensuring safe driving in rare failure scenarios	Simulation-based fault injection	Minimum time to collision, Failure probability	Norden et al. (2019)
MS	Diagnose	Accurate and reliable diagnosis	Field testing	False positive rate, False negative rate	Beede et al. (2020b)
UAS	VLG	Reliable landing	Fault tree analysis	Failure probability	Cluzeau et al. (2020)

ADS: Autonomous Driving System; MS: Medical System; UAS: Unmanned Aircraft System; VLG: Visual Landing Guidance

Autonomous driving systems (ADSs). A crucial element of an AV is its perception module, governed by the underlying ANN (Peng et al., 2020; Grigorescu et al., 2020). The ANN processes input from a variety of sensors, including cameras, LiDAR, and infrared sensors, to analyze the surroundings and produce outputs such as steering directions or class predictions. For end-to-end steering tasks, the robustness goals aim for minimal mean squared error (MSE) between predicted and actual steering directions.

Kim et al. (2019) assessed ADS output correctness in the presence of adversarial examples and synthetic noisy input. They introduced two metrics to measure how surprising an input is to an ADS. They quantified surprise by evaluating the deviation in the system's behavior (i.e., predicted steering angle) between the training data and the test input. Likelihood-based surprise adequacy measures the surprise by calculating the probability of the system encountering a similar input during training. In contrast, distance-based surprise adequacy calculates the Euclidean distance between neuron activation representations of the given input in operation and the training data.

Zhou et al. (2020) developed DeepBillboard, a systematic input-output evaluation method for dynamic driving conditions, such as varying viewing angles, lighting, and distances. The objective is to maximize the likelihood, degree, and duration of steering-angle errors in an AV caused by a generated adversarial billboard. The study proposed several metrics to assess perturbation effectiveness in both digital and physical domains, including attacking strength, attack possibility, average angle error, and the percentage of frames with angle error exceeding a predefined threshold.

The primary goal of object perception is to ensure safe driving during rare events (Norden et al., 2019). As validating performance requires driving the vehicle billions of miles to test it, many studies suggest simulation-based evaluations to efficiently assess AV systems and identify rare failure scenarios (Webb et al., 2020; Zhou et al., 2020; Yamaguchi et al., 2016). A simulation-based testing framework should prioritize scenarios, evaluate coverage of failure modes, and rank them by importance. Considering the AV system as a black box, Norden et al. (2019)

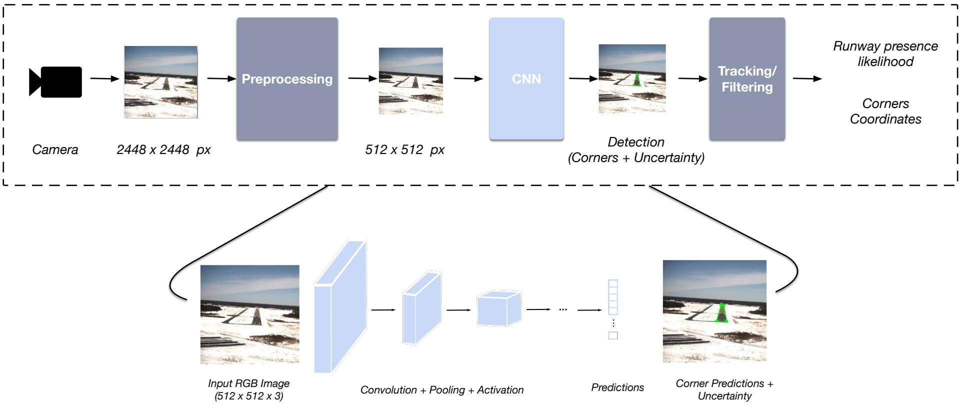


Figure 6: System architecture overview of visual landing guidance (VLG) (Cluzeau et al., 2020).

developed a simulation testing framework to measure accident probabilities based on standard traffic behavior. The testing framework can efficiently identify and rank failure scenarios using adaptive importance-sampling methods. They proposed two metrics: minimum time-to-collision (TTC) and failure probability. Their findings revealed that a widely deployed ADS, OPENPILOT (Comma AI, 2019), often fails due to uncertain perception of other vehicles or lane boundaries in specific weather conditions. Additionally, they found that collisions can be either glancing or high-impulse, depending on the vehicles' relative velocities.

Medical systems (MSs). Clinical investigations or trials (Beede et al., 2020b) are common methods for assessing the robustness of deployed MSs. Although there is no current requirement for ANN systems to undergo observational clinical studies, the success of an ANN model depends not only on its accuracy but also on its impact on patient care and outcomes (Shah et al., 2019). Beede et al. (2020b) conducted a user-centered field trial to evaluate the robustness of an ANN-based diabetic eye disease detection system used in clinical environments. The primary users of the system were nurses and retinal specialists. The robustness goal was to accurately and reliably detect diabetic eye disease and provide appropriate diagnoses to improve patient care. The study used live patient data in the form of eye images, either taken in a dedicated screening room with controlled lighting conditions for high-quality fundus photos or in the nurse's office with less controlled lighting conditions, which may have resulted in low-quality images the ANN system could not recognize. The study found that lighting conditions were significant factors in using the ANN system, and ungradable images and user frustration often resulted from suboptimal lighting conditions. Although the ANN system aimed to reduce the time required for patients to receive care, its deployment sometimes caused unnecessary delays or misdiagnoses. The study emphasized the importance of evaluating ANN-system performance using live data generated at the clinical site.

Unmanned aircraft systems (UASs). Object detection from UASs equipped with cameras has been increasingly deployed in many industrial applications. The mobility of UAS-mounted cameras brings greater challenges in robustness evaluation, such as variations in altitude and object scale, view angles, weather, and illumination (Yu et al., 2020). Visual landing guidance (VLG) facilitates landing an aircraft on a runway or vertiport (Cluzeau et al., 2020). It comprises traditional software and an ANN (see Figure 6). A quantitative ANN failure mode and effect analysis (FMEA) is proposed to estimate the failure rates of ANN-SCS, given adequate error metrics and failure definitions (Cluzeau et al., 2020).

5.2.3. Methods and Metrics of Robustness Evaluation at the ANN Model Level

The model-level evaluation focuses on evaluating an ANN model in isolation without considering any other components of the ANN-SCS. Consequently, most of the papers in this category did not conduct evaluations tailored to specific applications of the model. Thus, we list their application domains in Table 5 as "image classification." Additionally, we identified ANN model-level evaluations for particular domains, i.e., ADS, MS, and UAS. The robustness goals of each study vary, including being robust to pixel perturbations, spatial transformations, natural corruptions, distribution shifts, and hardware and software faults. Depending on the different robustness goals, various methods are employed to conduct robustness evaluations. One category of technique was sensitivity analysis (Saltelli, 2002), which attributes uncertainty in the output to diverse factors contributing to uncertainty in the model input. The sensitivity-based method requires zero knowledge of ANN classifiers, making it more applicable for assessing the robustness of ANN models in operation. Another category of technique was simulation-based, such as injecting fault (Hsueh et al., 1997) into systems to evaluate their behavior in fault scenarios. Table 5 summarizes the research identified on the methods and metrics of robustness evaluation at the ANN model level. **Note that these methods and metrics originate from lab evaluations but can be applied when labeled data are available in operational settings.**

Table 5: Methods and Metrics for evaluating the robustness of ANN-SCSs in operation (ANN model level).

SCS Domain	Operational Context	Robustness Goal (i.e., be robust against)	Method	Metric	Ref.
Image classification	Generic	Pixel perturbations	Sensitivity-based	Level-threshold-safe, Level-pixel-safe	Kotyan and Vargas (2019)
		Spatial deformations	Sensitivity-based	Attack success rate	Liu et al. (2018)
		Natural corruptions	Simulation-based	Threat severity, Minimal perturbations, Fooling success rate	Zhong et al. (2020)
			Sensitivity-based	Accuracy loss	Laugros et al. (2019)
		Hardware and software faults	Simulation-based fault injection	Bit error rate-accuracy curves	Reagen et al. (2018)
				SDC rate	Chen et al. (2020b)
ADS	Traffic sign recognition	Natural corruptions	Sensitivity-based	Classification accuracy	Berghoff et al. (2021)
	Object detection	Natural corruptions	Sensitivity-based	AP and AP@50, mAP	Michaelis et al. (2019)
MS	Gastroenterology	Distributional shift	Sensitivity-based	Sensitivity, specificity	Hicks et al. (2022)
UAS	VLG	Distributional shift	Sensitivity-based	Classification accuracy	Cluzeau et al. (2020)

ADS: Autonomous Driving System; MS: Medical System; MS: Medical System, UAS: Unmanned Aircraft System; VLG: Visual Landing Guidance

Across-domain evaluation methods and metrics for image classification. ANNs can be susceptible to misclassification when small disturbances, known as adversarial samples, are added to original samples (Kotyan and Vargas, 2019). Pixel perturbation refers to introducing minor changes to individual pixels in an image. The extent of these changes is determined by the L_p (p=0, 1, 2, inf) norm, which limits the number or magnitude of pixel alterations. Researchers have employed sensitivity-based methods to evaluate pixel perturbation robustness. Sensitivity-based methods investigate how input attacks can affect output (Kotyan and Vargas, 2019). Kotyan and Vargas (2019) suggested a dual evaluation method using L_0 and L_1 metrics to generate adversarial samples that humans do not misclassify. The concept of robustness levels was introduced to represent the degree of disruption added to the original sample and thresholds were defined to limit the spatial distribution of noise in the adversarial sample. They utilized two metrics, *level-threshold-safe* and *level-pixel-safe*, to assess an ANN model's resilience against adversarial attacks.

Pixel norm perturbations in model evaluation have limited practicality. Liu et al. (2018) proposed a parametric norm-ball attack that directly alters physical parameters such as lighting and geometry to create adversarial examples. Their differentiable rendering approach effectively assesses the sensitivity of model output to spatial transformations in real-world inputs. Zhong et al. (2020) introduced a simulation-based framework that measures the robustness of ANN models against natural perturbations in five categories: luminance, spatial transformation, blur, corruption, and weather. They used various metrics to evaluate the smallest perturbation needed to cause misclassification and the fooling success rate, indicating model robustness. Additionally, they introduced a threat severity metric that measures the minimum real-world perturbation required to change model predictions, quantified using L_p norm-based distance. A larger L_p norm-based distance signifies lower threat severity (Zhong et al., 2020). Laugros et al. (2019) examined the potential correlations between adversarial robustness and robustness to natural corruptions. They assessed the accuracy loss, which measures the ratio of the ANN model's accuracy on perturbed data to its accuracy on non-perturbed data, for each model against each perturbation. The robustness of the ANN model to unseen perturbations was evaluated through experiments, such as assessing the accuracy loss of an adversarially trained model against an unseen natural corruption or vice versa. The authors concluded that adversarial robustness and robustness to common perturbations are distinct attributes, suggesting that ANN robustness should be evaluated more broadly.

In addition to focusing on input data perturbations, a few studies have evaluated ANN model robustness against hardware and software faults (Reagen et al., 2018; Chen et al., 2020b). Reagen et al. (2018) introduced *Ares*, an ANN-specific fault injection framework designed to assess the relationship between hardware fault rate and model accuracy. Their research found that ANN fault tolerance varies significantly depending on the model, layer type, and structure. Chen et al. (2020b) introduced TensorFi to investigate faults at the interfaces of TensorFlow operators, assuming that faults within operators only affect their outputs. TensorFI considers output corruption in the form of "random value replacement" or "single bit-flip" for any data processed by the ANN system, such as weights, biases, or inputs. TensorFI evaluates ANN model robustness based on the occurrence of one type of output corruption, specifically, silent data corruption (SDC).

Domain-specific evaluation methods and metrics for ADSs. In ADSs, ANN models are used for traffic sign recognition and object detection. Despite the different operational contexts, environmental changes, such as weather conditions, have been identified as a key challenge for state-of-the-art ADSs. Typically, sensitivity-based methods are employed to analyze how domain-specific noises affect the output (Temel et al., 2017; Michaelis et al., 2019; Berghoff et al., 2021).

Berghoff et al. (2021) used a set of robustness properties (including image noise, pixel perturbations, geometric transformations, and color transformations) to evaluate ANN models trained on the German Traffic Sign Recognition Benchmark dataset under various environmental conditions. Their method helps identify failure modes that require attention in operation. For example, they discovered that the models perform well under normal conditions but have weaknesses when faced with direct sunlight or similar backgrounds. Similarly, Michaelis et al. (2019) assessed the robustness of object detection models under different image distortions and weather conditions, focusing on ADSs. They proposed three benchmark datasets (COCO-C, Cityscapes-C, and PASCAL-C) containing corrupted versions of commonly used object detection datasets. They demonstrated that a variety of object detection models undergo significant degradation in performance on corrupted images. The authors used dataset-specific performance measures to evaluate the robustness of object detection models. They adopted the PASCAL average precision (AP) metric at 50% intersection over union (IoU) for the PASCAL VOC dataset (Everingham et al., 2010) and the COCO AP metric, which averages over IoUs between 50% and 95%, for the MS COCO (Lin et al., 2014) and Cityscapes (Cordts et al., 2016) datasets. The challenge of applying sensitivity-based methods in the ADS domain is the cost of manually labeling thousands or even millions of inputs.

Domain-specific evaluation methods and metrics for MSs. Hicks et al. (2022) comprehensively discussed the evaluation metrics for binary ANN models in gastroenterology. In addition to commonly used metrics such as accuracy, precision, and recall, they emphasized the importance of incorporating clinically relevant metrics, like sensitivity and specificity. Sensitivity is determined by calculating the proportion of correctly classified positive samples to all samples assigned to the positive class. A high sensitivity value indicates the model's effectiveness in identifying the majority of positive cases, which is crucial in medical research. Specificity, conversely, represents the rate at which negative samples are accurately classified and serves as the negative class counterpart to sensitivity.

Domain-specific evaluation methods and metrics for UASs. In a study by Cluzeau et al. (2020), an ANN model was used for the perception component of a VLG system, which helps land aircraft on runways or vertiports. They assessed the ANN model's performance using traditional metrics such as accuracy, precision, recall, and F1 score. Yet, they emphasized that evaluation metrics should be selected considering the context of the entire system and its specific use case rather than in isolation. For example, a system designed to identify runways might prioritize minimizing false positives, which occur when the system makes incorrect predictions regarding the presence of runways. In contrast, a system focused on detecting other aircraft to prevent collisions might prefer tolerating false positives, ensuring that it does not overlook any aircraft in the vicinity. In light of these considerations, decision thresholds should be established based on different predefined rates of false negatives and positives during operation.

5.2.4. Methods and Metrics of Robustness Evaluation at the Input Data Level

The input-level evaluation aims to examine the representativeness of data inputs. A trend involves using large-scale realistic data across all application domains. Large volumes of data can help uncover patterns and trends, whereas small volumes may not provide enough information for accurately evaluating system robustness. For ADSs, large-scale public datasets primarily consist of annotated frames from LiDAR, radar, and stereo cameras, offering various city scenarios, weather conditions, times of day, and scene types (Yates, 2022). The variety of data used is also crucial.

If the data cover only a narrow range of scenarios, the evaluation may not be robust enough to handle unexpected events or outliers. Different applications can have various types of adverse conditions to consider. For instance, in ADSs, perturbations could include luminance, spatial transformation, blur, corruption, and weather (Zhong et al., 2020). In contrast, the UAS benchmark focuses on complex scenarios with viewpoint changes, fast motion, rare weather conditions, and flying altitude changes (Yu et al., 2020). Table 6 summarizes the identified research on methods and metrics for robustness evaluation at the input level. As input-level evaluation is domain-independent, we categorized the identified papers based on their robustness goals.

Table 6: Methods and Metrics for evaluating the robustness of ANN-SCSs in operation (input level).

SCS Domain	**Operational Context**	**Robustness Goal (i.e., be robust against)**	**Method**	**Metric**	**Ref.**
Generic	Generic	Semantic diversity	Coverage-based	Importance-driven coverage (IDC)	Gerasimou et al. (2020)
Generic	Generic	Natural corruptions	Sensitivity-based	Neighbor accuracy, Neighbor diversity score	Zhong et al. (2021)
Generic	Generic	Distributional shift	Sensitivity-based	F-measure for threshold values	Dola et al. (2021)
			Adversarial filtration	Classification accuracy	Hendrycks et al. (2021c,a)
Generic	Generic	Triggering misclassifications	Test input selection and prioritization	Maximum mean discrepancy-critic	Chen et al. (2020a)
				Model uncertainty-based	Ma et al. (2021)
				Sample discrimination-based	Meng et al. (2021)

Gerasimou et al. (2020) proposed DeepImportance to evaluate the adequacy of test cases in ANN systems. They introduced an importance-driven criterion (IDC) to assess the semantic adequacy of an input dataset by measuring how well it activates various combinations of important neurons' behaviors. A high IDC score implies a diverse input set that effectively triggers numerous combinations of significant neuron clusters.

ANN models can be easily deceived by slight changes in input data, making it crucial to identify data points that negatively impact robustness. Zhong et al. (2021) investigated the robustness of individual inputs when subjected to natural variations, such as rotations or rain in the original input. They determined neighbor accuracy as the percentage of a data point's neighbors, including the data point itself, that can be accurately classified by the ANN model being tested. A data point is considered robust with respect to the ANN model being tested if the accuracy of its neighboring points surpasses a pre-established threshold. On the contrary, a input point is non-robust if its neighbor accuracy falls below a predefined threshold. To quantify the diversity of classes that a data point's neighbors belong to, they calculated the neighbor diversity score (λ) using the simpson diversity index (Simpson, 1949) for each input data point. Then, the test images are ranked based on their λ values, and the top k images (where k is chosen according to the user requirements) are marked as potentially the most non-robust inputs.

Existing coverage-based methods struggle to differentiate between invalid and valid test cases, which can result in test suites being biased towards incorporating more invalid inputs to achieve higher coverage. Dola et al. (2021) proposed a deep generative model-based input validation approach to determine if test inputs are valid. Valid inputs are those from a dataset under test that follows the same distribution as the training data of the ANN being tested. They utilized a variational auto-encoder (VAE) model to classify test inputs generated by ANN test generation techniques. Test inputs with reconstruction probabilities lower than a specified threshold were identified as invalid. To find the optimal reconstruction probability threshold for identifying invalid inputs, they employed the F-measure, which is the harmonic mean of precision and recall.

Hendrycks et al. (2021c,a) introduced various robustness benchmark datasets to reveal the failure modes of ANN models. These benchmarks include ImageNet-A, a dataset comprising images that belong to ImageNet classes but are more challenging and can cause errors in different models. Other benchmark datasets, such as StreetView StoreFronts (SVSF), DeepFashion Remixed (DFR), and ImageNet-Renditions (ImageNet-R) (Hendrycks et al., 2021a), capture naturally occurring data distribution shifts in aspects like image style, geographical settings, and camera operation. These datasets were created using a straightforward adversarial filtration technique to eliminate spurious cues and examine model performance with easy-to-classify examples removed.

Several works focus on selecting an efficient subset of samples to save the labeling effort in operational contexts. Chen et al. (2020a) proposed PACE (practical accuracy estimation) to precisely estimate the accuracy of an ANN model for the entire testing set by using a selected subset of test inputs. PACE incorporates clustering to divide test inputs into distinct groups, uses the maximum mean discrepancy (MMD) measure to select representative prototypes, and employs adaptive random testing to ensure diverse coverage with the specified number of test inputs. Ma et al. (2021) proposed various metrics based on model uncertainty to identify data likely to cause misclassification. They used the maximum probability score to measure the highest prediction probability for a specific input across various mutant models. They also calculated the variance score, representing the variation in prediction probabilities. Samples with higher variance scores are considered more prone to causing misclassification. Furthermore, Ma et al. (2021) employed the Kullback-Leibler score to compare the actual class prediction distribution of an input with a worst-case scenario where class predictions are evenly distributed across all classes. Their findings revealed that model uncertainty-based metrics are highly effective in identifying misclassified inputs, surpassing the performance of coverage-based metrics. Meng et al. (2021) combined majority voting (Sagi and Rokach, 2018) and item discrimination (Ebel, 1954) techniques to measure the discrimination of inputs and choose "error-inducing inputs" to discriminate the robustness of multiple ANN models.

Answer to RQ2: Classification accuracy is the primary metric used for robustness evaluation at all three levels. Additionally, sensitivity-based evaluation methods are quite popular across these levels. For system-level assessments, simulation-based evaluation is commonly employed. In contrast, input-level assessments use coverage-based metrics to evaluate the effectiveness of various scenarios and conditions in the dataset. Utilizing a combination of complementary methods and metrics can help ensure that the robustness of the system is thoroughly analyzed and potential vulnerabilities are identified under various conditions and scenarios.

5.3. Results of RQ3: Challenges of measuring ANN-SCS Robustness in Operation

There are many metrics to evaluate robustness in operation. However, the focus should not be on the number of metrics but rather on effectively integrating or selecting the appropriate metrics to capture various aspects of robustness and address genuine concerns in real-world application scenarios. Building upon the results of RQ1 and RQ2, we unfold challenges related to the application domain, robustness goal, and methods/metrics at each level.

5.3.1. Challenges of Robustness Evaluation at the System Level

Research on the robustness evaluation of ANN-SCS in operation is still in its early stages. Studies of robustness evaluation at the system level are rarely reported in the literature. We distinguish two types of system architecture variations based on different evaluation goals: 1) an ANN-SCS as a black box, with the aim of assessing potential performance degradation due to input changes; and 2) an ANN-SCS with redundant ANN models, with the objective of comparing the performance of multiple models and recommending the optimal one for use.

Evaluating robustness at the system level as a black box. Three out of five studies (Kim et al., 2019; Zhou et al., 2020; Norden et al., 2019) have used simulated abnormal inputs to ensure the comprehensiveness of the abnormal conditions. While measuring the robustness of systems in a controlled laboratory environment can offer valuable insights before deployment, such measurements may differ from robustness measurements taken in operational, real-world settings. Although some studies have targeted user-centered field testing using operational data (Beede et al., 2020b), no metrics are currently available to quantify the comprehensiveness of defined abnormal conditions. To minimize the likelihood that ANN-SCSs will fail in scenarios involving rare failures, Hendrycks et al. (2021b) argued that systems must exhibit *unusual* robustness. They recommended creating more benchmarks, including unusual and extreme distribution shifts and rare failure scenarios, to stress-test systems.

Due to the absence of a system specification governing ANN's training inference mechanism and the use of data-centric approach, it becomes challenging to explicitly determine the expected performance of the ANN-SCS and evaluate whether it meets the required standards and regulations. For an SCS, a safety integrity level (SIL) specifies the level of performance required to maintain and achieve safety (IEC61508). The acceptable level of stable performance of an ANN-SCS directly contributes to the determination of the SIL of a system. However, no study provides a specific answer to deciding the acceptable level of robustness for an ANN-SCS. Indeed, the acceptable level would likely vary depending on the specific application and potential consequences of failure. We would suggest that a methodology to guide the decision of an acceptable level of stable performance should be developed for ANN-SCSs.

Although it is likely to obtain a considerable number of labeled data during the development phase, getting a similar amount of labeled data in operation in real time could be challenging. Data labeling requires significant manual effort. Labeling requiring particular expertise, e.g., labeling medical images (Liu et al., 2022), can be more costly. This implies that robustness evaluation methods in operation ideally should use unlabeled data. While some sample-based methods (Chen et al., 2020a; Ma et al., 2021) can estimate the robustness of ANN models based on a selected subset from unlabeled data, they still require manual labeling for the chosen samples, which can be a drawback. We didn't find studies addressing this challenge in system-level evaluation.

Evaluating robustness at the system level with redundant ANN models. In this scenario, multiple models are connected in a standby mode, waiting to be activated to perform some specific task or function. By having multiple models in standby mode, the system can continue to operate even if one model experiences an issue, providing high reliability and stability (Peng et al., 2020). Currently, it is popular to store multiple model variants in a dedicated cluster and serve the optimal model in operation (Vittal, 2021; Barla, 2023). This refers to **homogeneous redundancy** (Lu et al., 2022), a common strategy of using multiple identical components or models to perform the same task in a system.

Redundant hardware and software have been extensively used in traditional SCSs to increase reliability by intentionally duplicating critical components or functions of a system (Johnson, 1996; Jain and Gupta, 2011), and majority voting or fail-safe criteria are often employed to determine the active component. However, the study of robustness evaluation for ANN-SCSs with redundant ANN models has not yet been reported in the literature. Majority voting criteria can be ineffective due to the susceptibility of ANN models to the same input perturbations, such as adversarial examples. Furthermore, ANN models often fail silently when facing invalid inputs, indicating the ineffectiveness of fail-safe criteria.

Dedicated evaluation methods should be developed to compare the robustness of multiple ANN models and decide which model should be used within the SCS. While accuracy is a common and acceptable metric, it is hardly measurable when data are not labeled in operation. Consequently, industry best practices often involve drift detection as an alternative approach. However, drift detection faces two key issues when comparing the robustness of multiple ANN models: it does not measure the extent of degradation and provides a binary result (Yes/No), making it unfit for comparing multiple models, and it is complex to measure data drift in high-dimensional data like images. Developing reliable unsupervised metrics is an essential solution to address the challenge of data labeling in operation.

5.3.2. Challenges of Robustness Evaluation at the ANN Model Level

The reviewed papers evaluate ANN models focusing on adversarial perturbations, which are mainly generated based on the L_p norm distance, realistic environment lighting and geometry, spatial transformations, natural corruptions, and distributional shifts, respectively. However, the goal of robustness in practice is more comprehensive, as highlighted by Hendrycks et al. (2021b). For example, to assess adversarial robustness, it is crucial to consider perceptible attacks, as attackers may not only construct small L_p perturbations to deceive the system. They may also rotate the adversarially modified images or apply other novel distortions to them (Gilmer et al., 2018). The areas of adversarial robustness and corruption robustness, distributional shift, and unusual events should be considered in a unified manner.

ANN model-level robustness evaluation still lack definitions of acceptable levels of performance. For instance, mean absolute percentage error (MAPE) is an evaluation metric used to measure the accuracy of predictions across industries. A lower MAPE value indicates a more accurate prediction. A MAPE of 20% may be considered good or bad, depending on the situation. There is no industry standard for what the acceptable level of MAPE should be for a good model.

5.3.3. Challenges of Robustness Evaluation at the Input Level

Mincu and Roy (2022) addressed the challenges of obtaining high-quality healthcare datasets due to privacy-preserving considerations. They suggested techniques, such as federated learning, to encourage reproducibility while retaining data privacy. Liu et al. (2022) pointed out the challenge of differentiating between in-distribution and out-of-distribution cases given the complexity of most medical data. We envision that the use of fine-grained, actionable taxonomies of perturbations, collaborative documentations of domain-specific perturbations, libraries to generate such perturbations semi-automatically, and frameworks and metrics to uncover new types of perturbations in the wild must be studied in the future.

Answer to RQ3: We have identified three types of challenges, namely, identifying comprehensive abnormal conditions, standardizing the definition of an acceptable level of performance, and acquiring sufficient labeled data, at the system, ANN model, and input levels. Furthermore, we have highlighted an emerging need to assess ANN-SCSs using redundant models, which has been overlooked.

6. Discussion

In this section, we compare our results with related work and analyze the impacts of the results for the industry and validity threats to our studies.

6.1. Comparison with Related Work

Tocchetti et al. (2022) surveyed the terminology of concepts around AI robustness. They introduced three taxonomies: 1) methods and approaches that ensure robustness at different stages of the ML pipeline; 2) robustness tailored to specific model architectures, tasks, and systems; and 3) methodologies for assessing robustness from both a theoretical and practical perspective. They also highlighted the lack of a human perspective in evaluating AI robustness. They emphasized the urgent need to understand AI practitioners' practices and develop tools that assist in enhancing the robustness of AI systems. Surveys conducted by Riccio et al. (2020), Zhang et al. (2020), and Ashmore et al. (2021) offer an in-depth perspective on existing methods for evaluating ML systems' properties and obtaining assurances.

While existing reviews provide useful insights into the robustness research considering ANN models, our study, to the best of our knowledge, is the first comprehensive study to analyzes existing robustness evaluation approaches and metrics applicable to ANN-SCSs in operation. Our paper goes beyond the ANN model or input data aspect by investigating robustness evaluation at the system, model, and input levels. Recently, researchers have started to highlight the need to consider the entire system and the interactions of various components within the system (Li et al., 2022).

6.2. Implications

The lack of precise mathematical definitions for real-world robustness results in unclear and potentially misleading uses of the term within the research community. Robustness evaluations vary not only because robustness goals and available data are different from one test to another, but also because, there is no widely accepted guideline. The CONSORT-AI[2] and SPIRIT-AI[3] Steering Group (con, 2019) reported emerging issues in clinical trials involving AI interventions, including the study setting, the criteria for inclusion at the input data level, and the interaction between the human, and the algorithm. Liu et al. (2022) proposed a medical algorithmic audit framework to identify potential algorithmic errors, map the components contributing to errors, and anticipate their consequences. They suggested several methodologies for assessing these algorithmic errors, including "exploratory error analysis, subgroup testing, and adversarial testing." Considering the issues discussed earlier, we suggest that more guidelines aimed at enhancing future research on robustness evaluation in operations should be developed.

Existing literature states robustness goals at either the system level or the ANN model level. However, based on the technical reports we have reviewed, industry practitioners consider robustness goals from the top down (Cluzeau et al., 2020). That said, during analysis, it is often useful for analysts to break down system robustness objectives into their component or model-level goals to better understand the specific need for robustness evaluation and how to perform it. In turn, understanding the error propagation from low-level components (such as ANN models) to the system output is crucial. We should treat the system-level evaluation as a multi-stage process, given that variations introduced in earlier stages may accumulate and propagate to subsequent stages. Although it is possible to develop robustness metrics for individual stages (i.e., an ANN model or non-ANN component), how to combine these metrics to assess the robustness of a hierarchical ANN-SCS remains underexplored.

Assessing the robustness of ANN-SCSs by simply adopting principles recommended by standards and regulations is not straightforward. Business owners tend to care more about business KPIs. An adjacent system attribution of

[2] CONSORT-AI: Consolidated Standards of Reporting Trials—Artificial Intelligence

[3] SPIRIT-AI: Standard Protocol Items: Recommendations for Interventional Trials—Artificial Intelligence

robustness, i.e., resilience, is considered in operation. A resilient ANN model will perform well on a wide range of datasets beyond just the training set. It will also perform better for a longer period, as it is more robust and less overfitted. While no single KPI measures ANN model resilience, industry practitioners in machine learning operations (MLOps) have suggested a few ways to evaluate the resiliency of models (Chen et al., 2022):

- Smaller standard deviations in a cross-validation run
- Similar error rates for longer times in production models
- Less discrepancy between error rates of test and validation datasets
- How much input drift impacts the model

We believe that the robustness evaluation of ANN-SCSs in operation should also address business owners' concerns. This could involve finding a solution to aggregate evaluation results from the system, model, and input levels into a meta-index reflecting the system's robustness.

While researchers have made significant strides in proposing various metrics to evaluate robustness, the lack of consensus on which metric to use has limited their adoption by practitioners. As a result, ML teams often rely on ad hoc approaches when testing and evaluating the robustness of their models (Shankar et al., 2022), which may lead to inconsistent and unreliable assessments. To address this issue, researchers and industry professionals must collaborate to identify the key factors that define robustness across a broad range of ML models and applications.

Our literature review also shows that significant work remains to integrate our knowledge on ANN data, component and system-level robustness assessment into existing frameworks of cyber-physical safety assessment (Guzman et al., 2021) or broader approaches to designing robust complex SCSs (Leveson, 2004). However, we believe that our understanding of ANN-SCSs has now reached a point where we can begin integrating them into (and adaptating) established safety design and assessment frameworks and methods.

6.3. Threats to Validity

External Validity. First, the term "robustness" tends to be overused and is subject to a wide array of interpretations, targeting adversarial, corruption, and distributional robustness. As a result, many methods and metrics were initially designed to address only one of these robustness objectives. To mitigate this threat, we considered all robustness objectives during the paper selection process to ensure that our findings were relevant to a wide range of robustness challenges in operation. Another issue is that research on the robustness evaluation of ANN-SCSs in operation is still in its early stages, and studies of robustness evaluation at the system level are rarely reported in the literature. To address this, we combined knowledge from various sources, such as scientific papers and industry standards to provide a comprehensive understanding of the current state of robustness evaluation of ANN-SCSs in operation. Additionally, we are aware of the potential limitation in external validity arising from our focus on a limited number of application domains. To address this concern, we carefully selected three representative domains: ADSs, MSs, and UASs. These domains were chosen based on their well-established status and the availability of relevant studies. It is crucial to highlight that the insights and findings derived from these domains can be extrapolated to enhance robustness evaluation in other application domains.

Internal Validity. To minimize the risk of overlooking pertinent studies, we utilized six of the most relevant digital libraries. We also executed an extensive process of snowballing on the references of the selected papers. To ensure our search was thorough and appropriate, the authors cross-checked and reached an agreement on the search keywords. The primary author designed the data extraction template, and conducted the data extraction from the selected papers. To counter potential bias, both authors engaged in ongoing discussions about issues related to data extraction. Furthermore, to ensure accuracy and consistency, the second author verified the data that was extracted.

7. Conclusion and Future Work

In this survey, we gathered, organized, and analyzed existing literature on the evaluation of robustness for ANN-SCS in operation. Based on the identified literature, we first summarized the definitions of ANN-SCS robustness. We identified the key factors—application domain, operational context, system architecture, robustness goal, and nature of data—which are associated with robustness evaluation in operation. We then proposed a multi-dimensional framework to demonstrate the application of the reviewed methods and metrics to evaluate the robustness of ANN-SCSs. The study includes an in-depth analysis of the robustness evaluation methods and metrics for ANN-SCSs at the system, ANN model, and input levels. We provide insights to industry by describing the remaining research gaps in defining abnormal conditions, determining an acceptable level of performance, and obtaining labeled data. We believe that our findings will be a starting point for future studies focusing on the continuous evaluation of robustness in operational ANN-SCSs. In our future work, we plan to propose an evaluation methodology that enables the continuous assessment of multi-model robustness and facilitates the automated selection of the most robust model in operational settings. Our objective is to explore the impact of utilizing different distance metrics to establish a practical approach for ranking the robustness of multiple models in real-world ANN-SCSs. This research endeavor aims to address the dynamic risk assessment requirements of ANN-SCSs in operation. Additionally, we will focus on developing runtime decision-making support tools. These tools will involve monitoring and evaluating the output of large language models, such as ChatGPT, to ensure the generation of trustworthy outcomes and enhance overall system reliability.

Acknowledgments

This work is supported by the Management of Safety and Security Risks for Cyber-Physical Systems project, which is financed by the Norwegian University of Science and Technology and the Technical University of Denmark.

CRediT authorship contribution statement

Jin Zhang: Conceptualization; methodology; writing—original draft; validation. **Jingyue Li:** Supervision; project administration; writing—review and editing. **Josef Oehmen:** Writing—review and editing.

References

, 2019. Reporting guidelines for clinical trials evaluating artificial intelligence interventions are needed. Nature Medicine 25, 1467–1468.

Akhtar, N., Mian, A., 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. Ieee Access 6, 14410–14430.

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., Zimmermann, T., 2019. Software engineering for machine learning: A case study, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), IEEE. pp. 291–300.

Ashmore, R., Calinescu, R., Paterson, C., 2021. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. ACM Computing Surveys (CSUR) 54, 1–39.

Barla, N., 2023. Model Deployment Strategiesr. https://neptune.ai/blog/model-deployment-strategies. Accessed: 2023-02-19.

Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A., Criminisi, A., 2016. Measuring neural net robustness with constraints. Advances in neural information processing systems 29.

Beede, E., Baylor, E., Hersch, F., Iurchenko, A., Wilcox, L., Ruamviboonsuk, P., Vardoulakis, L.M., 2020a. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy, in: Proceedings of the 2020 CHI conference on human factors in computing systems, pp. 1–12.

Beede, E., Baylor, E., Hersch, F., Iurchenko, A., Wilcox, L., Ruamviboonsuk, P., Vardoulakis, L.M., 2020b. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. Conference on Human Factors in Computing Systems - Proceedings URL: https://dl.acm.org/doi/10.1145/3313831.3376718, doi:10.1145/3313831.3376718.

Berghoff, C., Bielik, P., Neu, M., Tsankov, P., Von Twickel, A., 2021. Robustness testing of ai systems: a case study for traffic sign recognition, in: Artificial Intelligence Applications and Innovations: 17th IFIP WG 12.5 International Conference, AIAI 2021, Hersonissos, Crete, Greece, June 25–27, 2021, Proceedings 17, Springer. pp. 256–267.

Boss, L.N., Gralla, E.L., 2023. Robustness of decentralized decision-making architectures in command and control systems. Systems Engineering 26, 149–161.

Boudette, E.N., 2017. Tesla's Self-Driving System Cleared in Deadly Crash. https://www.nytimes.com/2017/01/19/business/tesla-model-s-autopilot-fatal-crash.html/. Accessed:2023-01-16.

BUNDESAMT, F., 2004. An investigation into the performance of facial recognition systems relative to their planned use in photo identification documents–biop i. Bundesamt fur Sicherheit in der Informationstechnik .

Buzhinsky, I., Nerinovsky, A., Tripakis, S., 2021. Metrics and methods for robustness evaluation of neural networks with generative models. Machine Learning , 1–36.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A., 2021. On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705.

Carreras Guzman, N.H., Wied, M., Kozine, I., Lundteigen, M.A., 2020. Conceptualizing the key features of cyber-physical systems in a multi-layered representation for safety and security analysis. Systems Engineering 23, 189–210.

Chen, C., Murphy, N., Parisa, K., Sculley, D., Underwood, T., 2022. Reliable Machine Learning: Applying SRE Principles to ML in Production. O'Reilly Media, Incorporated. URL: https://books.google.no/books?id=1rvHzgEACAAJ.

Chen, J., Wu, Z., Wang, Z., You, H., Zhang, L., Yan, M., 2020a. Practical accuracy estimation for efficient deep neural network testing. ACM Transactions on Software Engineering and Methodology (TOSEM) 29, 1–35.

Chen, Z., Narayanan, N., Fang, B., Li, G., Pattabiraman, K., DeBardeleben, N., 2020b. Tensorfi: A flexible fault injection framework for tensorflow applications, in: 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), IEEE. pp. 426–435.

Cluzeau, J.M., Henriquel, X., Rebender, G., Soudain, G., Dijk, L.v., Gronskiy, A., Haber, D., Perret-Gentil, C., Polak, R., 2020. Concepts of Design Assurance for Neural Networks (CoDANN). Technical Report. European Union Aviation Safety Agency. URL: https://www.easa.europa.eu/en/document-library/general-publications/concepts-design-assurance-neural-networks-codann.

Comma AI, 2019. Openpilot. https://github.com/commaai/openpilot. Accessed: 2023-03-31.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3213–3223.

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., Hein, M., 2020. Robustbench: a standardized adversarial robustness benchmark. arXiv preprint arXiv:2010.09670.

Cruzes, D.S., Dyba, T., 2011. Recommended steps for thematic synthesis in software engineering, in: 2011 international symposium on empirical software engineering and measurement, IEEE. pp. 275–284.

Diochnos, D., Mahloujifar, S., Mahmoody, M., 2018. Adversarial risk and robustness: General definitions and implications for the uniform distribution. Advances in Neural Information Processing Systems 31.

Dola, S., Dwyer, M.B., Soffa, M.L., 2021. Distribution-aware testing of neural networks using generative models, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE. pp. 226–237.

Drenkow, N., Sani, N., Shpitser, I., Unberath, M., 2021. Robustness in deep learning for computer vision: Mind the gap? arXiv preprint arXiv:2112.00639.

Ebel, R.L., 1954. Procedures for the analysis of classroom tests. Educational and Psychological Measurement 14, 352–364.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A., 2019. Exploring the landscape of spatial robustness, in: International Conference on Machine Learning, pp. 1802–1811.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88, 303–338.

França, H.L., Teixeira, C., Laranjeiro, N., 2021. Techniques for evaluating the robustness of deep learning systems: A preliminary review, in: 2021 10th Latin-American Symposium on Dependable Computing (LADC), IEEE. pp. 1–5.

Gerasimou, S., Eniser, H.F., Sen, A., Cakan, A., 2020. Importance-driven deep learning system testing, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, pp. 702–713.

Gilmer, J., Adams, R.P., Goodfellow, I., Andersen, D., Dahl, G.E., 2018. Motivating the rules of the game for adversarial example research. arXiv preprint arXiv:1807.06732.

Gilmer, J., Ford, N., Carlini, N., Cubuk, E., 2019. Adversarial examples are a natural consequence of test error in noise, in: International Conference on Machine Learning, PMLR. pp. 2280–2289.

Glaser, B.G., Strauss, A.L., 2017. The discovery of grounded theory: Strategies for qualitative research. Routledge.

Goodfellow, I., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples, in: International Conference on Learning Representations.

Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G., 2020. A survey of deep learning techniques for autonomous driving. Journal of Field Robotics 37, 362–386.

Guzman, N.H.C., Kozine, I., Lundteigen, M.A., 2021. An integrated safety and security analysis for cyber-physical harm scenarios. Safety science 144, 105458.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al., 2021a. The many faces of robustness: A critical analysis of out-of-distribution generalization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8340–8349.

Hendrycks, D., Carlini, N., Schulman, J., Steinhardt, J., 2021b. Unsolved problems in ml safety. arXiv preprint arXiv:2109.13916.

Hendrycks, D., Dietterich, T., 2019. Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D., 2021c. Natural adversarial examples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15262–15271.

Hicks, S.A., Strümke, I., Thambawita, V., Hammou, M., Riegler, M.A., Halvorsen, P., Parasa, S., 2022. On evaluation metrics for medical applications of artificial intelligence. Scientific Reports 12, 5979.

Hsueh, M.C., Tsai, T.K., Iyer, R.K., 1997. Fault injection techniques and tools. Computer 30, 75–82.

Huang, X., Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E., Wu, M., Yi, X., 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Computer Science Review 37, 100270.

IEC61508, 2005. Functional safety of electrical/electronic/programmable electronic safety-related systems. Standard. International Electrotechnical Commission.

IEEE Std 610.12, 1990. IEEE Standard Glossary of Software Engineering Terminology. Standard. International Organization for Standardization.

ISO 14155, 2020. Clinical investigation of medical devices for human subjects — Good clinical practice. Standard. International Organization for Standardization.

ISO 21384-3, 2019. Unmanned aircraft systems - Operational procedures. Standard. International Organization for Standardization.

ISO 26262, 2011. Road vehicles – Functional safety. Standard. International Organization for Standardization.

ISO/IEC TR 24028-1, 2020. ISO/IEC TR 24028:2020 Overview of trustworthiness in artificial intelligence. Standard. International Organization for Standardization.

ISO/IEC TR 24029-1, 2021. Assessment of the robustness of neural networks Part1: Overview. Standard. International Organization for Standardization.

ISO/IEC TS 5723, 2022. ISO/IEC TS 5723:2022 Trustworthiness — Vocabulary. Standard. International Organization for Standardization. URL: https://www.iso.org/standard/81608.html.

Jain, M., Gupta, R., 2011. Redundancy issues in software and hardware systems: an overview. International Journal of Reliability, Quality and Safety Engineering 18, 61–98.

Javier, G.H., Pasquale, F., Rudolf, H., Apostolos, P., Laurent, B., 2019. Study on face identification technology for its implementation in the schengen information system .

Johnson, D.M., 1996. A review of fault management techniques used in safety-critical avionic systems. Progress in Aerospace Sciences 32, 415–431.

Kim, J., Feldt, R., Yoo, S., 2019. Guiding deep learning system testing using surprise adequacy, IEEE Computer Society. pp. 1039–1049. doi:10.1109/ICSE.2019.00108.

Kotyan, S., Vargas, D.V., 2019. Adversarial robustness assessment: Why both l_0 and l_∞ attacks are necessary. arXiv preprint arXiv:1906.06026.

Kumar, R.S.S., Brien, D.O., Albert, K., Viljöen, S., Snover, J., 2019. Failure modes in machine learning systems. arXiv preprint arXiv:1911.11034.

Kyrkou, C., Theocharides, T., 2019. Deep-learning-based aerial image classification for emergency response applications using unmanned aerial vehicles., in: CVPR Workshops, pp. 517–525.

Laugros, A., Caplier, A., Ospici, M., 2019. Are adversarial robustness and common perturbation robustness independant attributes?, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

Leveson, N., 2004. A new accident model for engineering safer systems. Safety science 42, 237–270.

Li, S., Guo, J., Lou, J.G., Fan, M., Liu, T., Zhang, D., 2022. Testing machine learning systems in industry: an empirical study, in: Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice, pp. 263–272.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. European Conference on Computer Vision (ECCV) , 740–755.

Liu, H.T.D., Tao, M., Li, C.L., Nowrouzezahrai, D., Jacobson, A., 2018. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. arXiv preprint arXiv:1808.02651.

Liu, X., Glocker, B., McCradden, M.M., Ghassemi, M., Denniston, A.K., Oakden-Rayner, L., 2022. The medical algorithmic audit. The Lancet Digital Health .

Lu, Q., Zhu, L., Xu, X., Whittle, J., Xing, Z., 2022. Towards a roadmap on software engineering for responsible ai, in: Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, pp. 101–112.

Ma, W., Papadakis, M., Tsakmalis, A., Cordy, M., Traon, Y.L., 2021. Test selection for deep learning systems. ACM Transactions on Software Engineering and Methodology (TOSEM) 30, 1–22.

Meng, L., Li, Y., Chen, L., Wang, Z., Wu, D., Zhou, Y., Xu, B., 2021. Measuring discrimination to boost comparative testing for multiple deep learning models, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE. pp. 385–396.

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W., 2019. Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484.

Mincu, D., Roy, S., 2022. Developing robust benchmarks for driving forward ai innovation in healthcare. Nature Machine Intelligence , 1–6.

Mohseni, S., Wang, H., Xiao, C., Yu, Z., Wang, Z., Yadawa, J., 2022. Taxonomy of machine learning safety: A survey and primer. ACM Computing Surveys 55, 1–38.

Molléri, J.S., Petersen, K., Mendes, E., 2016. Survey guidelines in software engineering: An annotated review, in: Proceedings of the 10th ACM/IEEE international symposium on empirical software engineering and measurement, pp. 1–6.

NIST, 2023. AI RISK MANAGEMENT FRAMEWOR. https://www.nist.gov/itl/ai-risk-management-framework. Accessed:2023-01-31.

Norden, J., O'Kelly, M., Sinha, A., 2019. Efficient black-box assessment of autonomous vehicle safety. arXiv URL: http://arxiv.org/abs/1912.03618.

Oberhaus, D., 2017. iPhone X's Face ID Can Be Fooled With a 3D-Printed Mask. https://www.vice.com/en/article/qv3n77/iphone-x-face-id-mask-spoof. Accessed:2023-01-27.

Peng, Z., Yang, J., Chen, T.H., Ma, L., 2020. A first look at the integration of machine learning models in complex autonomous driving systems: a case study on apollo, in: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 1240–1250.

Potts, M.W., Sartor, P.A., Johnson, A., Bullock, S., 2020. A network perspective on assessing system architectures: Robustness to cascading failure. Systems Engineering 23, 597–616.

Rawat, W., Wang, Z., 2017. Deep convolutional neural networks for image classification: A comprehensive review. Neural computation 29, 2352–2449.

Reagen, B., Gupta, U., Pentecost, L., Whatmough, P., Lee, S.K., Mulholland, N., Brooks, D., Wei, G.Y., 2018. Ares: A framework for quantifying the resilience of deep neural networks, in: Proceedings of the 55th Annual Design Automation Conference, pp. 1–6.

Riccio, V., Jahangirova, G., Stocco, A., Humbatova, N., Weiss, M., Tonella, P., 2020. Testing machine learning based systems: a systematic mapping. Empirical Software Engineering 25, 5193–5254.

Ross, A.M., Rhodes, D.H., Hastings, D.E., 2008. Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. Systems engineering 11, 246–262.

Ross, C., Swetlitz, I., 2018. IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show. https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments/. Accessed:2023-01-16.

Sagi, O., Rokach, L., 2018. Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, e1249.

Saltelli, A., 2002. Sensitivity analysis for importance assessment. Risk analysis 22, 579–590.

Schwall, M., Daniel, T., Victor, T., Favaro, F., Hohnhold, H., 2020. Waymo public road safety performance data. arXiv preprint arXiv:2011.00038.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F., Dennison, D., 2015. Hidden technical debt in machine learning systems. Advances in neural information processing systems 28.

Shah, N.H., Milstein, A., Bagley, S.C., 2019. Making machine learning models clinically useful. Jama 322, 1351–1352.

Shankar, S., Garcia, R., Hellerstein, J.M., Parameswaran, A.G., 2022. Operationalizing machine learning: An interview study. arXiv preprint arXiv:2209.09125.

Simpson, E., 1949. Measurement of diversity. Nature 163, 688–688. doi:10.1038/163688a0.

Snow, J., 2018. Amazon's Face Recognition Falsely Matched 28 Members of Congress With Mugshots. https://www.aclu.org/news/privacy-technology/amazons-face-recognition-falsely-matched-28. Accessed:2023-01-27.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Temel, D., Kwon, G., Prabhushankar, M., AlRegib, G., 2017. Cure-tsr: Challenging unreal and real environments for traffic sign recognition. arXiv preprint arXiv:1712.02463.

Thomas, S., Tabrizi, N., 2018. Adversarial machine learning: A literature review, in: International Conference on Machine Learning and Data Mining in Pattern Recognition, Springer. pp. 324–334.

Tocchetti, A., Corti, L., Balayn, A., Yurrita, M., Lippmann, P., Brambilla, M., Yang, J., 2022. Ai robustness: a human-centered perspective on technological challenges and opportunities. arXiv preprint arXiv:2210.08906.

Treveil, M., Omont, N., Stenac, C., Lefevre, K., Phan, D., Zentici, J., Lavoillotte, A., Miyazaki, M., Heidmann, L., 2020. Introducing MLOps. O'Reilly Media.

UK Government, 2022. Code of Practice: automated vehicle trialling. Department for Transportation. https://www.gov.uk/government/publications/trialling-automated-vehicle-technologies-in-public/code-of-practice-automated-vehicle-trialling. Accessed:2023-01-16.

US Food and Drug Administration, 2019. Artificial intelligence and machine learning in software as a medical device. Silverspring: US Food and Drug Administration .

Vittal, R., 2021. Deploy shadow ML models in Amazon SageMaker. https://aws.amazon.com/blogs/machine-learning/deploy-shadow-ml-models-in-amazon-sagemaker/. Accessed: 2023-02-19.

Wang, J., Chen, J., Sun, Y., Ma, X., Wang, D., Sun, J., Cheng, P., 2021. Robot: Robustness-oriented testing for deep learning systems, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE. pp. 300–311.

Webb, N., Smith, D., Ludwick, C., Victor, T., Hommes, Q., Favaro, F., Ivanov, G., Daniel, T., 2020. Waymo's safety methodologies and safety readiness determinations. arXiv preprint arXiv:2011.00054.

Wohlin, C., 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th international conference on evaluation and assessment in software engineering, pp. 1–10.

Xie, X., Ma, L., Juefei-Xu, F., Xue, M., Chen, H., Liu, Y., Zhao, J., Li, B., Yin, J., See, S., 2019. Deephunter: a coverage-guided fuzz testing framework for deep neural networks, in: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 146–157.

Yamaguchi, T., Kaga, T., Donzé, A., Seshia, S.A., 2016. Combining requirement mining, software model checking and simulation-based verification for industrial automotive systems, in: 2016 Formal Methods in Computer-Aided Design (FMCAD), IEEE. pp. 201–204.

Yasuda, H., Yang, J., 2022. Wingtip deflection monitoring and prediction based on digital image correlation and machine learning techniques, in: European Workshop on Structural Health Monitoring: EWSHM 2022-Volume 2, Springer. pp. 409–416.

Yates, G., 2022. Autonomous Driving Open Datasets Released To Date (2022). https://apera.io/l/autonomous-driving-open-datasets-release. Accessed: 2023–02-16.

Yu, F., Qin, Z., Liu, C., Zhao, L., Wang, Y., Chen, X., 2019. Interpreting and evaluating neural network robustness. arXiv preprint arXiv:1905.04270.

Yu, H., Li, G., Zhang, W., Huang, Q., Du, D., Tian, Q., Sebe, N., 2020. The unmanned aerial vehicle benchmark: Object detection, tracking and baseline. International Journal of Computer Vision 128, 1141–1159.

Zhang, J., Li, J., 2020. Testing and verification of neural-network-based safety-critical control software: A systematic literature review. Information and Software Technology , 106296.

Zhang, J.M., Harman, M., Ma, L., Liu, Y., 2020. Machine learning testing: Survey, landscapes and horizons. IEEE Transactions on Software Engineering .

Zheng, S., Song, Y., Leung, T., Goodfellow, I., 2016. Improving the robustness of deep neural networks via stability training, in: Proceedings of the ieee conference on computer vision and pattern recognition, pp. 4480–4488.

Zhong, Z., Hu, Z., Chen, X., 2020. Quantifying dnn model robustness to the real-world threats, in: 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE. pp. 150–157.

Zhong, Z., Tian, Y., Ray, B., 2021. Understanding local robustness of deep neural networks under natural variations, in: Fundamental Approaches to Software Engineering: 24th International Conference, FASE 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27–April 1, 2021, Proceedings 24, Springer International Publishing. pp. 313–337.

Zhou, H., Li, W., Kong, Z., Guo, J., Zhang, Y., Yu, B., Zhang, L., Liu, C., 2020. Deepbillboard: Systematic physical-world testing of autonomous driving systems, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, pp. 347–358.

Zhu, Y., Cheng, Y., Zhou, H., Lu, Y., 2021. Hermes attack: Steal dnn models with lossless inference accuracy., in: USENIX Security Symposium, pp. 1973–1988.

Paper 4

Dynamic robustness evaluation for automated model selection in operation,

Jin Zhang, Jingyue Li, and Zhirong Yang.

In review, Manuscript submitted to the International Journal of Information and Software Technology.

Dynamic Robustness Evaluation for Automated Model Selection in Operation

Jin Zhang[a,b,c], Jingyue Li[a,*], Zhirong Yang[a]

[a] *Computer Science Department, Norwegian University of Science and Technology, Trondheim, Norway*
[b] *Section of Engineering Design and Product Development, Technical University of Denmark, Denmark*
[c] *School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China*

Abstract

Context: The increasing use of artificial neural network (ANN) classifiers in systems, especially safety-critical systems (SCSs), requires ensuring their robustness against out-of-distribution (OOD) shifts in operation, which are changes in the underlying data distribution from the data training the classifier. However, measuring the robustness of classifiers in operation with only unlabeled data is challenging. Additionally, machine learning engineers may need to compare different models or versions of the same model and switch to an optimal version based on their robustness.
Objective: This paper explores the problem of dynamic robustness evaluation for automated model selection. We aim to find efficient and effective metrics for evaluating and comparing the robustness of multiple ANN classifiers using unlabeled operational data.
Method: To quantitatively measure the differences between model outputs and assess robustness under OOD shifts using unlabeled data, we choose distance-based metrics. An empirical comparison of five such metrics, suitable for higher-dimensional data like images, is performed. The selected metrics include Wasserstein Distance (WD), Maximum Mean Discrepancy (MMD), Hellinger distance (HL), the Kolmogorov-Smirnov Statistic (KS), and Kullback-Leibler (KL) divergence, known for their efficacy in quantifying distribution differences. We evaluate these metrics on ten state-of-the-art models (five CIFAR10-based models and five ImageNet-based models) from a widely used robustness benchmark (**RobustBench**) using data perturbed with various types and magnitudes of corruptions to mimic real-world OOD shifts.
Results: Our findings reveal that the WD metric outperforms others when ranking multiple ANN models for CIFAR10-based models, while the KS metric demonstrates superior performance for ImageNet-based models. MMD can be used as a reliable second option for both datasets.
Conclusion: This study highlights the effectiveness of distance-based metrics in ranking models' robustness for automated model selection. It also emphasizes the significance of advancing research in dynamic robustness evaluation.

Keywords: Artificial neural network classifier, automated model selection, robustness, dynamic evaluation, distance-based metrics

[*] Corresponding author
Email addresses: jin.zhang@ntnu.no (Jin Zhang), jingyue.li@ntnu.no (Jingyue Li), zhirong.yang@ntnu.no (Zhirong Yang)

Preprint submitted to Information and Software Technology

1. Introduction

Artificial neural network (ANN) models in operation are susceptible to input data changes from the training data [1], which are commonly referred to as *out-of-distribution (OOD) shifts*. Due to the diverse underlying mechanisms that cause OOD shifts, the best methods for enhancing models' OOD robustness differ across different datasets and shifts [2, 3]. As emphasized by [4, 5], one crucial research direction for effective artificial intelligence (AI) risk management is continuously monitoring and validating the outcomes of AI systems. This highlights the need for practitioners to *dynamically evaluate and choose optimal models for deployment under changing conditions*. In addition, employing different models to perform the same task, known as the **multi-model decision-maker** [4], presents a promising solution for maintaining system performance and accuracy when facing OOD shifts [6], as it leverages the diversity of models to provide robust predictions in dynamic and changing conditions.

The multi-model decision-maker has been observed in various AI applications, including AWS fraud detection[1] and IBM Watson natural language understanding.[2] However, naive averaging, or taking the models' majority decision, followed by most of today's multi-model decision-makers, is not optimal as models may be sensitive to different OOD shifts, resulting in misleading conclusions when averaging the models' outputs or using votes. *Automated ranking to choose the best model regularly* in the context of a multi-model decision-maker can be a better strategy, particularly in the presence of OOD shifts. However, implementing such a strategy is challenging due to several factors. Firstly, the introduction of OOD shifts in the data brings about uncertainties and variations, posing challenges in accurately assessing the performance of each model. Secondly, selecting the best model requires an effective evaluation metric that can capture the model's robustness under different shifts. Third, unlike model training data, most data in model operation are unlabeled. Thus, we aim to answer the following question:

How can we compare and rank the robustness of multiple ANN models using unlabeled input during operation, supposing OOD shifts may happen at any time during operation?

This study focuses on ANNs for classifying high-dimensional data, such as images, to serve as a representative example demonstrating the challenges we are addressing. We consider OOD (also referred to as natural corruption [6]) robustness since natural corruption is the main input that influences ANN classifiers' performance in real-world scenarios [7], e.g., autonomous vehicles (AVs) [8].

Although the scientific community has developed methods to facilitate the testing of ANN models without relying on data labels [9, 10], these methods typically require the training of a dedicated supervisor model to monitor the performance of the deployed ANN model individually. While the approach is effective for assessing a single model, it becomes impractical when comparing multiple ANN models because the training of a dedicated supervisor model for each deployed model is resource-intensive, especially when dealing with a large number of models. Additionally, each supervisor model may have its own biases and limitations, leading to inconsistent and incomparable results across different models. Another promising category of research uses distance-based metrics [11] to estimate the performance degradation of a single model given the changing input data. The single-model scenario measures performance degradation given an

[1] AWS fraud detection: https://aws.amazon.com/cn/solutions/implementations/fraud-detection-using-machine-learning/

[2] IBM Watson natural language understanding: https://www.ibm.com/au-en/cloud/watson-natural-language-understanding

identical model but different input data. In contrast, the robustness comparisons in the multi-model decision-maker scenario need to compare the performance difference of selected models given identical input data. The differences in ANN models' architectural, training scheme, and inherent properties can result in variations in how the models respond to OOD shifts. Therefore, it is uncertain whether the performance degradation measured by distance-based metrics can effectively differentiate the robustness of multiple models.

Research comparing the robustness of multiple models using these distance-based metrics is relatively scarce. The lack of extensive experimental evidence makes it challenging to determine these metrics' suitability for robustness comparison among multiple models. Comparing the robustness of multiple models must consider various factors, such as the choice of distance-based metrics, the nature and extent of input perturbations, and the sample size. The interplay between these factors can introduce complexities and dependencies that may affect the effectiveness of distance-based metrics in robustness comparison.

The distance-based metrics that we consider are drawn mainly from the drift detection literature [11, 12, 13], including Wasserstein distance (WD) [14], maximum mean discrepancy (MMD) [15], the Kolmogorov-Smirnov statistic (KS) [16], Hellinger distance (HL) [17], and Kullback-Leibler (KL) divergence [18]. Our main experiments were carried out on two OOD shift datasets, i.e., CIFAR10-C [19] and ImageNet-3DCC [20]. The experiments provide evidence about the effectiveness of the selected distance-based metrics under different types of shifts, and for what percentage of the shifts they are effective, as well as the minimum number of samples needed to make reliable rankings. Our empirical findings demonstrate the superiority of WD and KS over other metrics in ranking multiple ANN models for CIFAR10-based models and ImageNet-based models, respectively. MMD is a suboptimal option for both datasets. Based on our empirical findings and analysis, we recommend a minimum sample size of 500 to achieve stable ranking accuracy of over 0.50.

Our **main contribution** is novel empirical evidence on the applicability of using distance-based metrics to dynamically select the best model under OOD shift using only unlabeled data in the context of multi-model decision-makers. Furthermore, our research has revealed the importance of considering the metrics' assumptions and characteristics of the data to be analyzed when selecting the most appropriate metric.

The paper is organized as follows: Section 2 provides an overview of the related work. Section 3 presents the problem formulation and our research design. In Section 4, we describe the evaluation design and results. Section 5 discusses our results' implications and the limitations of our work. The conclusions and future work are presented in Section 6.

2. Related Work

Methods developed for dynamic robustness evaluation of ANN classifiers in operation should consider the following two challenges: 1) the ground truth (labels) is often inaccessible or delayed; 2) the types of shifts in machine learning (ML) operation can be unknown. Three categories of related work have been proposed to address the two challenges.

Model evaluation using a subset of test data. Several test selection-based methods [21, 22] have been proposed to rank multiple models with minimum labeling effort. For instance, Ma et al. [21] proposed various metrics based on model uncertainty to identify data likely to cause misclassification. Meng et al. [22] combined majority voting [23] and item discrimination [24] techniques to measure the discrimination of inputs and select a set of "error-inducing inputs" to differentiate the robustness of multiple ANN models. These methods still rely on labeling a subset of data, making them unsuitable for addressing our specific problem of using only unlabeled data for robustness ranking in operation.

Labeling-free model performance estimation. For example, AutoEval [10] and SelfChecker [25] propose learning an accuracy regression model using a synthetic meta-dataset, resulting in accurate predictions of model accuracy for real-world unlabeled test datasets. However, the methods in [10, 25] require a separate supervisor model to monitor and predict the performance of a single deployed ANN model. Although it is technically possible to train multiple supervisor models to monitor and predict the performance of multiple deployed ANN models, there are several practical challenges associated with this approach. Firstly, training and maintaining multiple supervisor models can be computationally expensive and time-consuming. Secondly, each supervisor model may have its own biases and limitations, leading to inconsistent and incomparable results across different models.

Shift detection of a single model. Data shift detection primarily focuses on identifying changes in the input data, while model shift detection aims to detect shifts in the output of ANN classifiers. Measuring distribution differences between input data to derive model robustness is unreliable since data shifts can often have trivial impact on model performance [12]. There are two main approaches to detecting model shifts: statistical-based and distance-based. Statistical-based methods rely on a given confidence level, usually 95%, to determine if a model shift is detected. However, this approach does not measure the magnitude of shift and provides only a binary (Yes/No) result, making it unsuitable for ranking multiple models. Distance-based approaches measure the distance between the distributions that generate the training and test data. Igor et al. [11] assessed the practical application of several state-of-the-art distance-based metrics for estimating the magnitude of model shifts. Their study showed that distance-based methods offer an alternative for estimating performance degradation. However, further investigation is needed to determine whether these techniques can effectively compare and rank the robustness of multiple ANN classifiers. This motivates us to explore various distance-based metrics and examine their effectiveness in ranking the robustness of multiple ANN classifiers during operation.

3. Research Methodology

In this section, we begin by introducing the problem we aim to address, which is the comparison of robustness among multiple models using distance-based metrics. We then provide an overview of our research design, outlining the methodology and specific distance-based metrics employed in our study.

3.1. Problem Formulation

This paper focuses on image classification tasks using the multi-model decision-maker architecture. We aim to address the problem of assessing the robustness of ANN models in dynamic environments where the operation data can be different from the training data due to OOD shift. First, we differentiate two types of data:

- The modeling data X_{model}, which include X_{tr} (training datasets) and X_{te} (test datasets), are the data used in the modeling (model development) stage.
- The operation data X_{op} are the input data to the model in the operation stage.

Second, **we assume that the model predicts accurately in the modeling stage**. This assumption prevents meaningless random or underfitting models, which holds in practice as people usually only deploy models with high accuracy [26].

Considering the context of operational environments, the acceptable level of performance can be case-specific. We prefer to adopt the contrastive measures strategy, which measures the dissimilarity between the feature distribution of a model's soft predicted labels on a reference dataset and operational data. In this study, we use the training dataset as the reference dataset. Therefore, the precise formulation of the problem is as follows. Let $X_{\text{tr}} \in \mathbb{R}^{D_1 \times D_2 \times N}$ denote N training images of dimension $D_1 \times D_2$, $A = \{a_k\}_{k=1}^{m}$ denote a set of m different data augmentation schemes (e.g., adding random cropping or Gaussian noise to training images [6]), and $S_a(X_{\text{tr}})$ denote training datasets where augmentation $a \in A$ has been applied to X_{tr}. To address the problem of ranking models within a multi-model decision-maker system F^N, where F^N consists of N models, we consider the output of the softmax layer of each model, denoted as $f^a(\cdot)$. Our goal is to select the most robust model f^* from the set of models in F^N using a distance-based metric `dist` applied to the augmented training data $S_a(X_{\text{tr}})$ and the unlabeled operation data X_{op}. The ranking problem can be formulated as follows:

$$f^* = \arg \min_{f^a \in F^N} \texttt{dist}(f^a(S_a(X_{\text{tr}})), f^a(X_{\text{op}})), \tag{1}$$

In Equation (1), the distance-based metric `dist` compares the outputs of each model when applied to the augmented training data $S_a(X_{\text{tr}})$ and the unlabeled operation data X_{op}. The model with the minimum distance (f^*) is selected as the most robust model because the distance reflects the similarity between the model's predictions on familiar data (training data) and unseen data (operation data). The smaller the distance, the better the model's ability to generalize and handle variations or shifts in the data distribution. Figure 1 illustrates the above-described workflow.

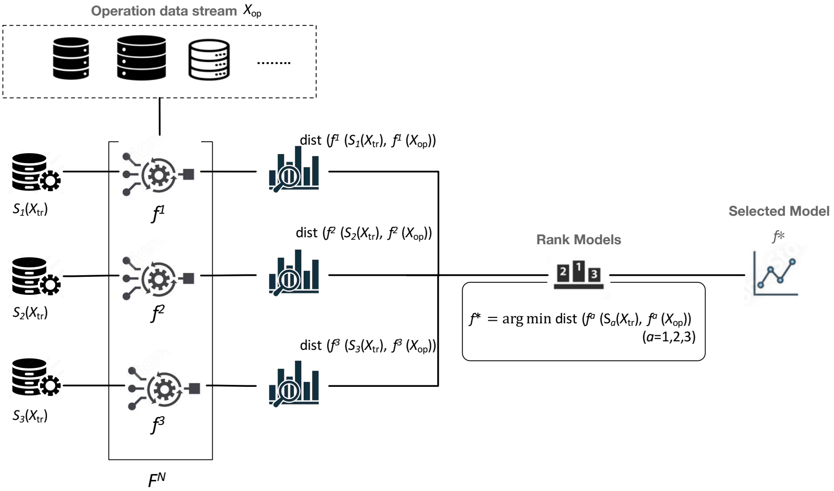

Operation data stream X_{op}
....
dist (f^1 ($S_1(X_{tr})$, f^1 (X_{op}))
$S_1(X_{tr})$
f^1
dist (f^2 ($S_2(X_{tr})$, f^2 (X_{op}))
$S_2(X_{tr})$
f^2
Rank Models
Selected Model
f^*
f^* = arg min dist (f^a ($S_a(X_{tr})$, f^a (X_{op}))
(a=1,2,3)
dist (f^3 ($S_3(X_{tr})$, f^3 (X_{op}))
$S_3(X_{tr})$
f^3
F^N

Figure 1: Illustration of the studied problem.

Terminology for distance-based metrics. A statistical distance is a quantitative measure of dissimilarity between two objects [27]. A distance-based metric is a non-negative function $D(x, y)$ that defines the distance between objects X and Y that satisfies the following axioms [28]:

- Non-negativity: $D(x, y) \geq 0$.
- Identity of indiscernibles: $D(x, y) = 0 \Leftrightarrow x \equiv y$.
- Symmetry: $D(x, y) = D(y, x)$
- Triangle inequality: $D(x, y) + D(y, z) \geq D(x, z)$.

Theoretical justifications for using distance-based metrics for robustness ranking. We are inspired by AutoEval [10], which showed a very strong negative correlation between robustness and distribution difference. This finding indicates that it is feasible to estimate robustness with distribution statistics. Furthermore, Theorem 1 in [29] states that the OOD robustness of classifiers can be bounded as the conditional Wasserstein distance between the training data distribution and corrupted data distribution. It provides theoretical support for the promising use of distance-based metrics to solve our robustness ranking problem.

3.2. Research Design

Previous papers have shown that the last few layers of ANN classifiers before the output contain valuable information to represent ANN robustness [30, 31]. Among the layers, we find that softmax appears in all popular ANN classifiers because it outputs class probabilities and provides a differentiable surrogate for hard classification. This key observation motivates us to employ the information in the output of the softmax layer (i.e., $f^a(\cdot)$ in Eq. 1) to compare the robustness of ANN classifiers. This choice avoids domain-specific feature representation. Furthermore, the neural network forward pass naturally gives the soft predicted labels. Therefore, using such information to evaluate the ANN classifier's robustness can reduce the demand to manually label the prediction results.

Selecting the candidate distance-based metrics. As explained in Section 1, we have chosen five distance-based metrics for our analysis. While these metrics were not specifically designed for robustness ranking, they have demonstrated utility in identifying issues with model performance degradation in previous studies [12, 13].

The first metric we considered is the Wasserstein distance (WD) [14], which measures the first- and second-order distance between two distributions. Another metric is the maximum mean discrepancy (MMD) [15], a kernel-based technique that distinguishes between two probability distributions based on their mean embeddings in a reproducing kernel Hilbert space. The Kolmogorov-Smirnov (KS) statistic [16] is a statistical test that is sensitive to differences in the mean and dispersion of two distributions. The Hellinger distance (HL) [17] measures the similarity between two probability distributions. HL is symmetric, well-defined for categorical and numerical features, and widely accepted in the industry. A larger HL value indicates greater dissimilarity between the distributions, while a smaller value indicates higher similarity or overlap. Lastly, we considered the Kullback-Leibler (KL) divergence [18], which is a widely used measure that captures the information-based disparity between two distributions. KL divergence assesses how much information is lost when one distribution is used to approximate another.

By selecting this set of metrics, we ensure coverage of various assumptions about the underlying data and the ability to capture different deviations between the output features of trained and operational data. For example, WD considers the mean and standard deviation of the distributions, while MMD measures the discrepancy between distribution features in a reproducing kernel Hilbert space. HL is symmetric and has a clear analogy to Euclidean distance, making it widely accepted in the industry for capturing the dissimilarity between probability distributions. KS test compares each dimension separately and identifies the largest difference across all dimensions. Although KL divergence is not strictly a distance-based metric, we included it in our study because it can quantify the difference between the distributions of model outputs on operation data and training data in terms of information content. By incorporating KL divergence alongside other distance-based metrics, we can obtain a more comprehensive understanding of the distributional differences and their impact on model performance.

4. Evaluation of the Effectiveness of Distance-Based Metrics on Ranking Models

4.1. Evaluation Design

Evaluation questions. This study addresses the following two evaluation questions:

- RQ1 (Effectiveness under OOD shifts): How well do the selected metrics rank multiple ANN classifiers when provided with various types of OOD data and their combinations?

  RQ1 is broken down into three sub-questions as follows:

  - RQ 1.1: How do the selected metrics perform in ranking the robustness of multiple ANN classifiers under different types of corruptions?
  - RQ 1.2: What is the impact of varying percentages of corrupted input on the effectiveness of the selected metrics in ranking the robustness of multiple ANN classifiers?
  - RQ 1.3: How well do the selected metrics rank the robustness of multiple ANN classifiers when faced with mixed combinations of corruption types?

- RQ2 (Sample size impact): What is the minimum sample size required for the selected metrics to achieve over 50% precision in ranking the robustness of multiple ANN classifiers under varying levels of corruption?

RQ1 explores the effectiveness of the selected metrics in ranking multiple ANN classifiers using OOD test data. By considering various conditions such as corruption types (RQ 1.1), varying percentages of corrupted input (RQ 1.2), and a mixture of corruption types (RQ 1.3), we aim to provide a comprehensive evaluation of the metrics' performance in scenarios that simulate real operational settings. This is important because in practical applications, models may encounter unknown corruptions or a combination of different types of corruptions, and it is crucial to assess their robustness under such conditions. RQ2 aims to provide insights into the practical feasibility of using these metrics in real-world scenarios where the amount of labeled data for evaluation may be limited.

Model selection. The variability in robustness among ANN classifiers can be attributed to various factors, including the classifier's architecture, training methods, and evaluation settings. For example, ANN models can be trained using standard training datasets or augmented inputs through data augmentation techniques [32]. The proportion of augmented inputs used during training, as well as the type and strength of augmentations, can significantly impact the robustness of the trained models. Additionally, some ANN classifiers may perform well against certain corruptions but poorly against others. To ensure the representativeness of our evaluation, we carefully consider all these factors during the model selection process.

Ten state-of-the-art ANN classifiers robust against natural corruption (Models 1-10 in Table 1) were chosen from RobustBench [32]. RobustBench is a standardized robustness benchmark. It contains a robustness evaluation of 40+ models in image classification on natural corruptions. Here, we selected five robust models from the CIFAR10 leaderboard and five from the ImageNet leaderboard, respectively, because these models have demonstrated strong performance and robustness against a wide range of natural corruptions in the RobustBench benchmark. By choosing models from the leaderboard, we ensured that we were evaluating state-of-the-art models that have undergone rigorous testing and evaluation, making them reliable candidates for our study. After selecting the models, we measured their clean accuracy to classify the images in the training dataset. The results are shown in the last column of Table 1 and illustrate that all selected models have been trained to a satisfactory accuracy. Among them, Models 1, 2, and 3 are trained based on a backbone network WideResNet-18-2, while Models 4 and 5 are trained on Augmix [33]. In Augmix, diverse augmentations are randomly selected and applied to a training image, followed by the mixture of the augmented image with the original. Models 6 and 7 are two vision transformers (ViTs)-based models. They are the top two state-of-the-art models on the leaderboard of ImageNet. Models 8, 9, and 10 are ResNet-50-based models. Models 8 and 10 are trained by leveraging noisy augmentations in input and feature space to achieve high OOD robustness on ImageNet-C.

Table 1: Datasets and models used in our experiments.

No.	**Dataset**	**Model ID**	**Source**	**Clean Accuracy**
1	CIFAR10-C,Corruptions	Diffenderfer2021Winning_LRR_CARD_Deck	[6]	0.97
2		Diffenderfer2021Winning_LRR	[6]	0.97
3		Diffenderfer2021Winning_Binary_CARD_Deck	[6]	0.95
4		Hendrycks2020AugMix_ResNeXt	[33]	0.96
5		Hendrycks2020AugMix_WRN	[33]	0.95
6	ImageNet-3DCC,Corruptions	Tian2022Deeper_DeiT-B	[34]	0.81
7		Tian2022Deeper_DeiT-S	[34]	0.80
8		Erichson2022NoisyMix_new	[35]	0.77
9		Hendrycks2020Many	[2]	0.77
10		Erichson2022NoisyMix	[35]	0.77

Corruption datasets. Model robustness is sensitive to input variations [19]. The choice of the corruption datasets in our study was made to simulate OOD scenarios in operation. To thoroughly evaluate the consistency of selected distance-based metrics for robustness ranking, we considered a variety of natural corruptions and their mixtures. We utilized the CIFAR10-C dataset [19], which consists of 15 corruption types. These corruptions include Gaussian noise, motion blur, brightness variations, etc. Additionally, we employed the ImageNet dataset with 3D Common Corruptions (ImageNet-3DCC) [20], which introduces 12 corruption types that align with real-world scenarios, such as lighting, weather conditions, and camera motion. Besides, each type of corruption in CIFAR10-C and ImageNet-3DCC has five levels of severity.

Determining the ground truth for comparing ranking results. To evaluate the ranking results, we utilized ranking based on robust accuracy measured using the correct labels as the ground truth. Robust accuracy is a widely accepted measure in the ML literature for evaluating the performance of ANN models under corruptions [36, 37, 38]. In order to compare the rankings produced by the distance-based metrics with the ground truth, we employed the average precision at k (AP@k) metric [39] commonly used in evaluating recommendation systems and ranking-related problems. AP@k evaluates two aspects: 1) the relevance of the recommended items and 2) whether the most relevant items are placed at the top.

In our study, we selected k=1 to focus on selecting the best model, considering the context of a multi-model decision-maker. Precision@1 checks if the model in the top position matches the ground truth. AP@1 provides a measure of how accurately the ranking generated by a specific distance-based metric, aligns with the ground truth for a specific type of corruption across all severity levels. The mean AP@1 calculates the average AP@1 for recommendations across different corruption types, providing an overall measure of ranking accuracy.

We implemented the proposed approach and carried out experiments using a state-of-the-art framework, i.e., PyTorch 1.7.1, and toolbox, i.e., RobustBench [32].

4.2. Result of RQ1: Effectiveness under OOD Shifts

RQ 1.1 Corruption type. The robustness of ANN classifiers is corruption-dependent [3, 6, 19]. A model that is robust against a certain corruption can still be vulnerable to other corruptions. This implies that the ranking of model robustness could change when testing environments vary (e.g., new testing inputs, corruption type, application field, etc.). For each experimental setting and distance-based metric, we evaluated how well different metrics performed in ranking the robustness of CIFAR10 and ImageNet models under a single type of corruption.

CIFAR10 models. As introduced in Section 4.1, we compared the ranking accuracy of the five selected distance-based metrics by using the mean AP@1 score. Table 2 shows the evaluation result for the CIFAR10 models (Model 1-5) under 15 corruptions. For each corruption type, the AP@1 in the table is the average ranking across CIFAR10-C severity levels 1 through 5, and the bold number indicates which metric achieves the highest performance on that corruption. Both WD and MMD achieve a mean AP@1 exceeding 0.50, indicating their effectiveness in robustness ranking under most types of corruption. WD demonstrates superior results with 14 out of 15 corruption types. MMD emerges as the second-best metric and can serve as a viable alternative to WD. On the other hand, HL, KS, and KL exhibit relatively poor AP@1 scores, with values below or equal to 0.47.

ImageNet models. Note we only evaluated 11 ImageNet-3DCC corruptions (excluding corruption type: xy_motion_blur) due to download errors encountered with the original source. Table 3 presents the evaluation results for the ImageNet models (Model 6-10) under 11 ImageNet-3DCC corruptions. The AP@1 values in the table represent the average performance across severity levels 1 through 5 for each corruption type, and the bold numbers indicate the metric with the highest average performance for that corruption. MMD achieves the best performance, and KS and WD also achieve a mean AP@1 exceeding 0.50, demonstrating their effectiveness in robustness ranking across most corruption types. However, HL and KL exhibit relatively poor mean AP@1 scores, below 0.40.

Answer to RQ 1.1: For CIFAR10 models, WD and MMD emerged as the top two performing metrics, whereas, MMD and KS outperformed other metrics when ranking ImageNet models.

RQ 1.2: Percentage of corruptions. Building on the insights from Rabanser et al. [12], which emphasized the significance of considering varying percentages of affected data in de-

Table 2: Ranking precision of different distance-based metrics on CIFAR10-C. Bold font indicates the more accurate ranking estimation across five metrics, i.e., WD, MMD, HL, KS, and KL.

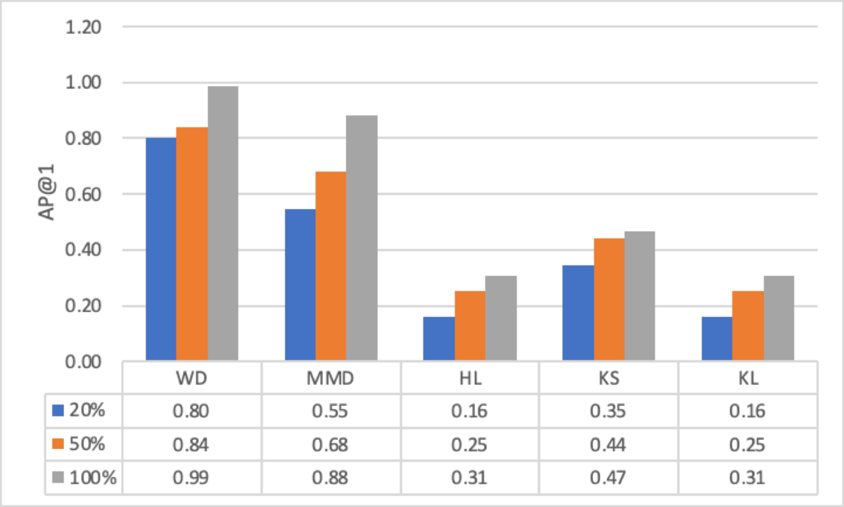
Corruption	**WD**	**MMD**	**HL**	**KS**	**KL**
shot_noise	**1.00**	**1.00**	0.80	0.00	0.80
motion_blur	**1.00**	**1.00**	0.00	0.60	0.00
snow	**1.00**	0.80	0.00	0.80	0.00
pixelate	**1.00**	0.80	0.40	0.60	0.40
gaussian_noise	**1.00**	**1.00**	**1.00**	0.00	**1.00**
defocus_blur	**1.00**	0.80	0.00	0.20	0.00
brightness	**1.00**	0.60	0.00	0.20	0.00
fog	**1.00**	**1.00**	0.00	**1.00**	0.00
zoom_blur	**1.00**	**1.00**	0.00	0.40	0.00
frost	**1.00**	**1.00**	0.00	**1.00**	0.00
glass_blur	**1.00**	**1.00**	0.80	0.20	0.80
impulse_noise	**1.00**	0.60	0.60	0.00	0.60
contrast	0.80	0.80	0.00	**1.00**	0.00
jpeg_compression	**1.00**	**1.00**	**1.00**	0.00	1.00
elastic_transform	**1.00**	0.80	0.00	**1.00**	0.00
Mean AP@1	**0.99**	0.88	0.31	0.47	0.31

Table 3: Ranking precision of different distance-based metrics on ImageNet-3DCC. Bold font indicates the most accurate ranking estimation across five metrics, i.e., WD, MMD, HL, KS, and KL.

Corruption	**WD**	**MMD**	**HL**	**KS**	**KL**
near_focus	0.80	**1.00**	0.00	**1.00**	0.00
far_focus	**1.00**	**1.00**	0.00	**1.00**	0.00
bit_error	0.40	**1.00**	0.00	**1.00**	0.00
color_quant	0.20	0.40	0.20	0.40	0.20
flash	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
fog_3d	0.60	0.00	0.60	0.00	**0.80**
h265_abr	0.40	**1.00**	0.00	**1.00**	0.00
h265_crf	0.00	**1.00**	0.00	**1.00**	0.00
iso_noise	**1.00**	0.80	1.00	0.60	**1.00**
low_light	**1.00**	0.40	**1.00**	0.20	**1.00**
z_motion_blur	**1.00**	**1.00**	0.40	**1.00**	0.00
Mean AP@1	0.67	**0.78**	0.38	0.75	0.36

tecting shifts, we adopted a similar approach to examine the influence of different corruption percentages on the performance of our selected metrics. Specifically, we explored multiple corruption percentages, ranging from $\delta = 0.2$ to $\delta = 1.0$, for each corruption type. In the following sections, we analyze the effects of different corruption percentages on the performance of our selected metrics separately for the CIFAR10 and ImageNet models.

CIFAR10 models. We present the analysis of the mean AP@1 across all corruption types to explore the relationship between the amount of affected data and the accuracy of ranking estimation using distance-based metrics. The results in Figure 2 indicate that WD consistently achieves a mean AP@1 score of over 0.50 across all three levels of affected data. Furthermore, we observe an increasing trend in ranking accuracy for all five metrics as the percentage of affected data increases. Notably, ranking models becomes more challenging when the input data exhibit only 20% corruption.



	WD	MMD	HL	KS	KL
■ 20%	0.80	0.55	0.16	0.35	0.16
■ 50%	0.84	0.68	0.25	0.44	0.25
■ 100%	0.99	0.88	0.31	0.47	0.31

Figure 2: Ranking precision of varying corruption percentages (CIFAR10-C).

ImageNet models. Figure 3 provides valuable insights into the impact of different corruption percentages on the ranking accuracy of the selected metrics for ImageNet models, in comparison to the findings for CIFAR10 models. Notably, MMD and KS consistently achieve a mean AP@1 score of over 0.50 across all three levels of affected data, indicating their effectiveness in capturing the robustness of ImageNet models under different levels of data corruption. Conversely, the other metrics demonstrate relatively lower ranking accuracy across varying corruption percentages. In addition, the WD and MMD metrics show an increasing trend in ranking accuracy as the percentage of affected data rises.

Answer to RQ1.2: For CIFAR10 models, all five metrics exhibited an increasing trend in ranking accuracy as the percentage of corrupted data rose. However, for ImageNet models, WD and MMD metrics showed an increasing trend in ranking accuracy with increasing corruption percentage, where KS showed almost constant ranking accuracy.

RQ1.3: Mixtures of corruptions. To evaluate the robustness of models in scenarios with unknown types and combinations of corruptions, we conducted evaluations using mixed perturbations. This approach allowed us to account for real-world operations where the specific corruption types and their combinations are uncertain. To simulate this unknown corruption sce-

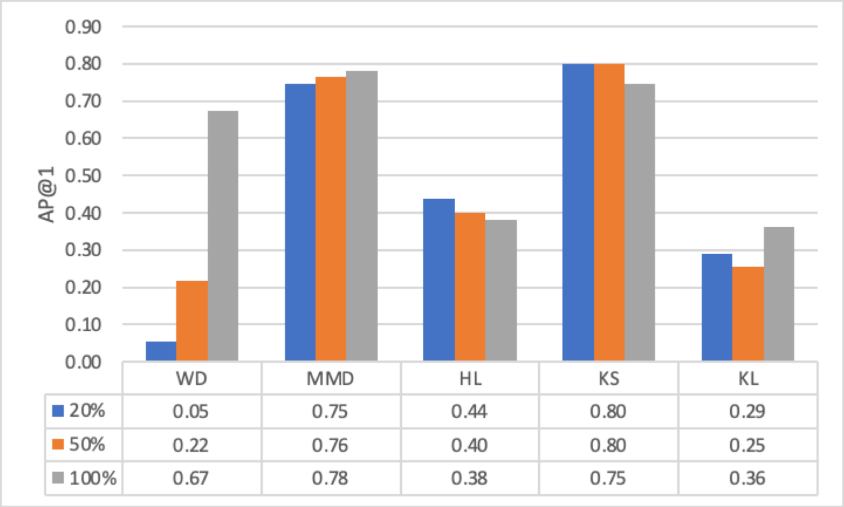


Figure 3: Ranking precision AP@1 of varying corruption percentages (ImageNet-3DCC).

nario, we employed the Poisson distribution [40], which describes the probability distribution of random events occurring over time.

In our evaluation, we utilized the CIFAR10-C dataset, which consists of 15 types of corruptions, each with five severity levels. Each severity level contains 10 000 samples, resulting in a total of 50 000 images per corruption type. We created a perturbed data pool by incorporating 750 000 perturbed inputs from the CIFAR10-C dataset. We focused on Models 1-5 for this analysis. In each experiment run, we randomly selected 1000 mixtures of corrupted samples as a test data batch using Poisson and uniform processes to emulate the corruption types, severity levels, and probabilities encountered in real-world scenarios. After generating the test data batches, we calculated the mean AP@1 for the WD, MMD, HL, and KS metrics. We repeated this evaluation process for 50 batches to obtain robust performance estimates. This approach allowed us to assess the ranking accuracy of the distance-based metrics under mixed perturbations and evaluate their ability to handle unknown corruption scenarios.

To simulate mixtures of corruptions for Models 6-10, we employed a similar strategy using the ImageNet-3DCC dataset. The ImageNet-3DCC dataset comprises 12 types of corruptions, each with five severity levels. Each severity level provides 5000 samples for testing purposes. We created a perturbed data pool consisting of 275 000 perturbed inputs from the ImageNet-3DCC dataset, covering 11 types of corruptions (excluding xy_motion_blur) due to download errors encountered with the original source. The results in Table 4 align with our findings for RQ 1.1, demonstrating that WD and MMD are effective in ranking the robustness of multiple ANN classifiers under mixed input scenarios for CIFAR10 models. Additionally, MMD and KS show satisfactory performance in ranking the robustness of multiple ANN classifiers for ImageNet models. This suggests that different metrics may be more suitable for assessing model robustness depending on the dataset characteristics and the nature of the input data.

Table 4: Ranking accuracy AP@1 under mixed perturbations.

Data source	**Batches**	**WD**	**MMD**	**HL**	**KS**	**KL**
CIFAR10C	50	0.9	0.72	0.48	0.42	0.48
ImageNetC	50	0.4	0.86	0.14	0.98	0.02

Answer to RQ 1.3: For CIFAR10 models, WD and MMD showed effectiveness in ranking robustness under mixed input scenarios, while KS and MMD achieved satisfactory performance for ImageNet models.

4.3. Result of RQ2: Sample Size Impact

We conducted model ranking experiments using different sample sizes of corrupted images from the operation set, ranging from 10 to 1000. The corruption type was randomly selected, and the severity level was fixed at 5. As the HL and KL metrics were found to be ineffective in ranking multiple models based on the results of RQ1, they were excluded from this experiment.

Table 5: Ranking accuracy AP@1 of varying sample sizes.

Model	**Metric**	**Sample size**						
		10	**20**	**50**	**100**	**200**	**500**	**1000**
M1-M5	WD	0.60	0.20	0.80	0.40	0.60	1.00	1.00
	MMD	0.40	0.20	0.60	0.40	0.60	0.60	1.00
	KS	0.20	0.60	0.20	0.20	0.40	0.60	0.40
M6-M10	WD	0.09	0.00	0.09	0.18	0.00	0.36	0.40
	MMD	0.64	0.36	0.82	0.82	0.73	0.73	0.91
	KS	0.45	0.55	0.82	0.73	0.45	0.64	0.91

The results presented in Table 5 verify the effectiveness of WD and MMD for ranking CIFAR10 models, and of MMD and KS for ranking ImageNet models given 1000 samples. Further, the results show that at least 200 samples are required to achieve reliable ranking performance with a mean AP@1 score of over 0.50 for CIFAR10 (M1-M5) models. In the case of ImageNet models, the MMD metric consistently outperformed the others even with a smaller sample size of 50, while the KS metric showed satisfactory performance in some cases but lacked stability. We observed that with a sample size of 500 samples, the ranking results tended to be more consistent and satisfactory.

Answer to RQ2: The impact of sample size on model ranking may vary depending on the dataset and the specific metric being used. However, to ensure a reliable model ranking, we recommend a minimum sample size of 500.

5. Discussion

5.1. Comparison with Related Work

To address the issue of estimating model performance using unlabeled data, we have adopted a different approach from the studies conducted by Schelter et al. [9] and Deng et al. [10]. While their methods relied on synthetic perturbations and model prediction information to train performance predictors for each pretrained model to measure models' performance with unlabeled data, we have chosen a straightforward strategy by directly employing distance measurements of model output statistics. In contrast to [9, 10], distance-based estimation does not require a separate training step. It directly measures performance degradation based on the soft predicted label. The benefit is that a distance-based metric is simple and effective.

5.2. Implications for Academia

Our results show that the performance of different metrics varied depending on the complexity of the datasets. CIFAR10 [41] is a smaller dataset that consists of 60 000 32×32 color images in 10 classes, with 6,000 images per class. ImageNet [42] is a more complex dataset containing over 1.2 million high-resolution images distributed among 1000 different classes. The images in ImageNet cover a wide range of object categories and exhibit greater diversity in terms of visual appearance, background complexity, and object scales.

The reason that CIFAR10 and ImageNet models require different distance-based metrics to achieve accurate ranking results could be attributed to the characteristics of these metrics and the nature of the classification task. For CIFAR10-based models, the superiority of the WD and MMD metrics in accurately predicting the best model suggests that these models align well with the assumptions and capabilities of the WD and MMD metrics. In contrast, for ImageNet-based models, the KS and MMD metrics outperformed others, indicating that these models' characteristics were better captured by the KS and MMD metrics in terms of the observed distribution differences in their softmax outputs.

We used a visualizing technique named UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction) [43] to further understand the rationale behind our findings. UMAP enables the compression of high-dimensional data/features into two or three dimensions for visualization. For instance, researchers from Google and OpenAI used UMAP to analyze the space of activations of a neural network [44]. We fed the softmax outputs produced by training data and operation data through UMAP to reduce them to two dimensions. They were then plotted, with similar softmax outputs placed near each other. Figure 4 shows the three-dimensional UMAP illustrations of some examples of CIFAR10 models and ImageNet models under several OOD shifts. According to the UMAP visualization, we observed that the softmax outputs of CIFAR10 models have a scattered stripe-type distribution, while ImageNet models' outputs have a cluster-based distribution. The scattered stripe-like distribution of CIFAR10 models' softmax outputs suggests that the models' predictions are more diverse and spread out across the output space. The WD metric, which measures the first- and second-order distance between two distributions, can effectively capture these variations by considering both the mean and standard deviation of the distributions. In the case of ImageNet models, the cluster-based distribution implies that the models' softmax output distributions are more concentrated and have less variability compared to CIFAR10 models. The KS metric, being sensitive to differences in mean and dispersion, can effectively detect variations in the distributions even when they are clustered together. The MMD metric, which considers the mean embeddings of the distributions, allows it to capture variations in distributions regardless of their specific patterns or structures.

5.3. Implication for Industry

Dynamic robustness evaluation directly aligns with the ISO 26262 automotive standard [45] for functional safety, which emphasizes the importance of emergency operation as a mode activated when transitioning to a safe state is not feasible within a specified timeframe. By regularly

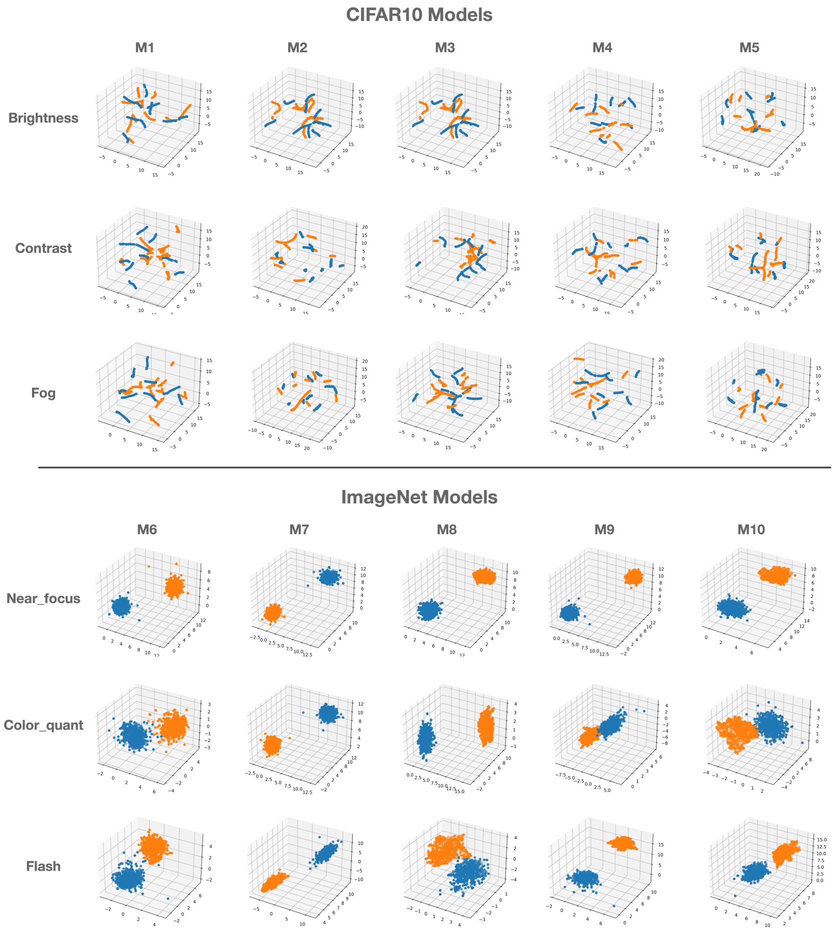


Figure 4: 3D UMAP visualization: Some examples of the softmax output of Models M1-M10 given training data and corrupted operation data. Data source: blue: training data; orange: operation data.

assessing backup model robustness and ensuring fail-operational behavior, stakeholders in the automotive industry can develop AI-based SCSs that adhere to the ISO 26262 standard and uphold ethical and responsible AI principles.

In another popular scenario, a large-scale ML-powered system typically utilizes diverse ML models and leverages their interactions to enable complex functionalities. Dynamic robustness evaluation offers a comparative measure for assessing the robustness of multiple models or versions within the same operational environment. It can facilitate automated monitoring and evaluation of ANN-SCSs against robustness requirements, triggering risk mitigation strategies when necessary. In Figure 5, we envision the application of dynamic robustness evaluation in three phases. In the initial phase, the performance of the main model is continuously monitored to determine if it meets the required level of robustness. If the requirements are not satisfied, it automatically proceeds to the second phase, which involves evaluating backup models to ensure their acceptable performance. In the third phase, it ranks the models to select the optimal one for deployment, enabling a seamless transition from the main model. The dynamic robustness evaluation offers a systematic and automated approach to assess the robustness of ANN-SCSs and choose the most robust model for operation, particularly in SCSs.

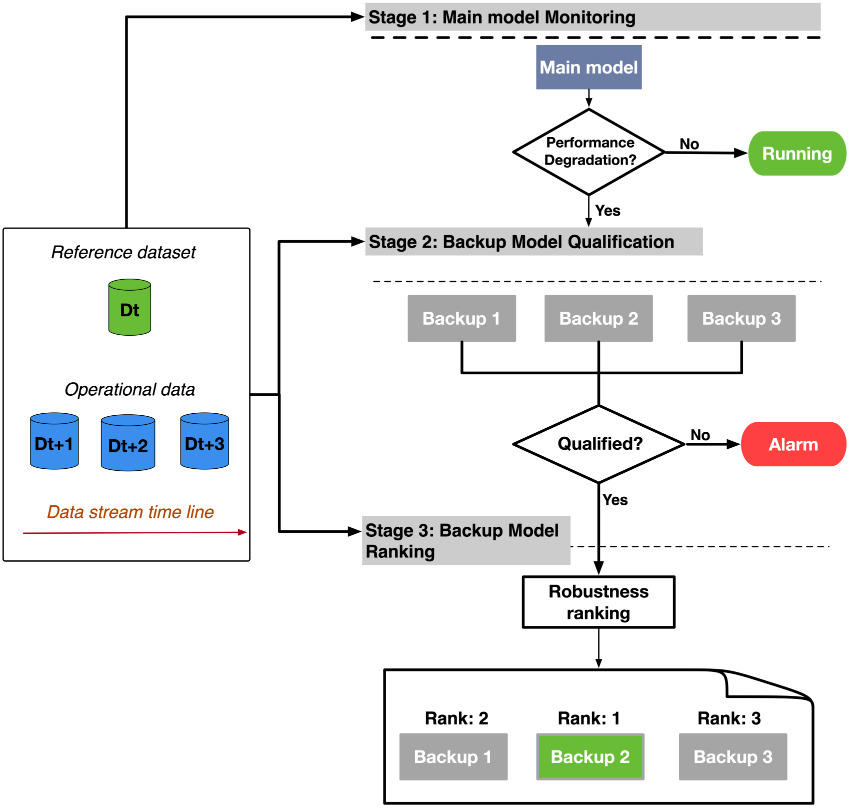


Figure 5: Envisioned use case of dynamic robustness evaluation.

5.4. Threats to Validity

Although distance-based metrics can be applied to scenarios when the type of corruptions from the target operational circumstances are unknown, we piloted only individual known corruptions to verify the correlation between the distance metric scores and state-of-the-art evaluation methods using labeled inputs because we had to use these known corruptions on existing methods to build the ground truth for comparisons. Another construct validity threat originates from the selection of distance-based metrics. To mitigate this issue, we chose metrics widely used in the industry. The possible threat to external validity is that we only piloted our experiments with limited numbers of datasets, ANN classifiers, and natural corruptions. To address this validity, we chose datasets, ANN classifiers, and corruptions popularly used in ANN robustness studies. distance-based metrics apply to ANN classifiers of any size and type of corruption because it compares the dissimilarity of the soft predicted label information in the modeling and operation settings.

6. Conclusion and Future Work

This paper presents a comprehensive empirical investigation aimed at enhancing our understanding of the utilization of distance-based metrics for robustness ranking and automated model selection. By considering various factors such as the dataset, model, corruption type, corruption percentage, and sample size impact, we conducted a thorough evaluation to assess the performance of these metrics. Among the five selected distance-based metrics, we assessed which metrics perform best under OOD shifts. Our findings demonstrate that the WD metric performs best in ranking the robustness of CIFAR10 models, while the KS metric is optimal for ranking the robustness of ImageNet models. In contrast, the MMD metric is found to be suboptimal for both datasets.

The complexity of model robustness and the diverse nature of OOD shifts make it difficult to derive a universally applicable theory. However, our study highlights the importance of considering the matches between the assumptions and characteristics of the metrics and the profile of the possible data using, e.g., UMAP analysis. This knowledge can guide future research toward developing more robust and comprehensive theories for model ranking and robustness assessment. In the future, research efforts should aim to extend the evaluation to other application domains and explore additional evaluation techniques. This will help improve the generalizability of our findings.

Acknowledgments

This work is supported by the Management of Safety and Security Risks for Cyber-Physical Systems project, which is financed by the Norwegian University of Science and Technology and the Technical University of Denmark. This work is also supported by the Norwegian Research Council with Grant No. 287284. The computations were performed on resources provided by the NTNU IDUN/EPIC computing cluster [46].

CRediT authorship contribution statement

Jin Zhang: Conceptualization; methodology; software; writing—original draft; validation. **Jingyue Li:** Supervision; project administration; Writing—review and editing. **Zhirong Yang:** Conceptualization; review and editing.

References

[1] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, D. Song, Anomalous example detection in deep learning: A survey, IEEE Access 8 (2020) 132330–132347.

[2] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al., The many faces of robustness: A critical analysis of out-of-distribution generalization, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, pp. 8320–8329.

[3] O. Wiles, S. Gowal, F. Stimberg, S. Alvise-Rebuffi, I. Ktena, K. Dvijotham, T. Cemgil, A fine-grained analysis on distribution shift, arXiv preprint arXiv:2110.11328 (2021).

[4] Q. Lu, L. Zhu, X. Xu, J. Whittle, Z. Xing, Towards a roadmap on software engineering for responsible ai, in: Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, pp. 101–112.

[5] NIST, AI RISK MANAGEMENT FRAMEWORK, https://www.nist.gov/itl/ai-risk-management-framework, 2023. Accessed:2023-01-31.

[6] J. Diffenderfer, B. Bartoldson, S. Chaganti, J. Zhang, B. Kailkhura, A winning hand: Compressing deep networks can improve out-of-distribution robustness, Advances in Neural Information Processing Systems 34 (2021) 664–676.

[7] Z. Zhong, Z. Hu, X. Chen, Quantifying dnn model robustness to the real-world threats, in: 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE, pp. 150–157.

[8] R. Taylor, J. Zhang, I. Kozin, J. Li, Safety And Security Analysis for Autonomous Vehicles, https://github.com/safe-ai-tech/Reports_Papers, 2021. [Technical report].

[9] S. Schelter, T. Rukat, F. Bießmann, Learning to validate the predictions of black box classifiers on unseen data, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pp. 1289–1299.

[10] W. Deng, L. Zheng, Are labels always necessary for classifier accuracy evaluation?, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15069–15078.

[11] I. Goldenberg, G. I. Webb, Survey of distance measures for quantifying concept drift and shift in numeric data, Knowledge and Information Systems 60 (2019) 591–615.

[12] S. Rabanser, S. Günnemann, Z. Lipton, Failing loudly: An empirical study of methods for detecting dataset shift, Advances in Neural Information Processing Systems 32 (2019).

[13] G. A. Lewis, S. Echeverría, L. Pons, J. Chrabaszcz, Augur: A step towards realistic drift detection in production ml systems, in: Proceedings of the 1st Workshop on Software Engineering for Responsible AI, pp. 37–44.

[14] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover's distance as a metric for image retrieval, International journal of computer vision 40 (2000) 99.

[15] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, A. Smola, A kernel method for the two-sample-problem, Advances in neural information processing systems 19 (2006).

[16] J. W. Pratt, J. D. Gibbons, J. W. Pratt, J. D. Gibbons, Kolmogorov-smirnov two-sample tests, Concepts of non-parametric theory (1981) 318–344.

[17] D. A. Cieslak, T. R. Hoens, N. V. Chawla, W. P. Kegelmeyer, Hellinger distance decision trees are robust and skew-insensitive, Data Mining and Knowledge Discovery 24 (2012) 136–158.

[18] J. M. Joyce, Kullback-leibler divergence, in: International encyclopedia of statistical science, Springer, 2011, pp. 720–722.

[19] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, arXiv preprint arXiv:1903.12261 (2019).

[20] O. F. Kar, T. Yeo, A. Atanov, A. Zamir, 3d common corruptions and data augmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18963–18974.

[21] W. Ma, M. Papadakis, A. Tsakmalis, M. Cordy, Y. L. Traon, Test selection for deep learning systems, ACM Transactions on Software Engineering and Methodology (TOSEM) 30 (2021) 1–22.

[22] L. Meng, Y. Li, L. Chen, Z. Wang, D. Wu, Y. Zhou, B. Xu, Measuring discrimination to boost comparative testing for multiple deep learning models, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE, pp. 385–396.

[23] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (2018) e1249.

[24] R. L. Ebel, Procedures for the analysis of classroom tests, Educational and Psychological Measurement 14 (1954) 352–364.

[25] Y. Xiao, I. Beschastnikh, D. S. Rosenblum, C. Sun, S. Elbaum, Y. Lin, J. S. Dong, Self-checking deep neural networks in deployment, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE, pp. 372–384.

[26] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, T. Zimmermann, Software engineering for machine learning: A case study, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), IEEE, pp. 291–300.

[27] S.-H. Cha, Comprehensive survey on distance/similarity measures between probability density functions, City 1 (2007) 1.

[28] D. Burago, I. D. Burago, Y. Burago, S. Ivanov, S. V. Ivanov, A Course in Metric Geometry, volume 33, American Mathematical Soc., 2001.

[29] V. Sehwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, P. Mittal, Improving adversarial robustness using proxy distributions, arXiv preprint arXiv:2104.09425 1 (2021).

[30] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, Adversarial examples are not bugs, they are features, in: Advances in Neural Information Processing Systems, pp. 125–136.

[31] A. Rahnama, A. T. Nguyen, E. Raff, Robust design of deep neural networks against adversarial attacks based on lyapunov theory, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8178–8187.

[32] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, M. Hein, Robustbench: a standardized adversarial robustness benchmark, arXiv preprint arXiv:2010.09670 (2020).

[33] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, B. Lakshminarayanan, Augmix: A simple data processing method to improve robustness and uncertainty, arXiv preprint arXiv:1912.02781 (2019).

[34] R. Tian, Z. Wu, Q. Dai, H. Hu, Y. Jiang, Deeper insights into vits robustness towards common corruptions, arXiv preprint arXiv:2204.12143 (2022).

[35] N. B. Erichson, S. H. Lim, F. Utrera, W. Xu, Z. Cao, M. W. Mahoney, Noisymix: Boosting robustness by combining data augmentations, stability training, and noise injections, arXiv preprint arXiv:2202.01263 1 (2022).

[36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: 6th International Conference on Learning Representations, ICLR.

[37] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, M. Jordan, Theoretically principled trade-off between robustness and accuracy, in: International Conference on Machine Learning, PMLR, pp. 7472–7482.

[38] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, Z. Wang, Adversarial robustness: From self-supervised pre-training to fine-tuning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 699–708.

[39] Y.-M. Tamm, R. Damdinov, A. Vasilev, Quality metrics in recommender systems: Do we calculate metrics consistently?, in: Proceedings of the 15th ACM Conference on Recommender Systems, pp. 708–713.

[40] S. Katti, A. V. Rao, Handbook of the poisson distribution, Taylor & Francis, 1968.

[41] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, pp. 248–255.

[43] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).

[44] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, C. Olah, Exploring neural networks with activation atlases, Distill. (2019).

[45] ISO 26262:2011, Road vehicles – Functional safety, Standard, International Organization for Standardization, 2011.

[46] M. Själander, M. Jahre, G. Tufte, N. Reissmann, EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure, 2019.

Secondary Paper 1

Could we issue driving licenses to autonomous vehicles?

Jingyue Li, Jin Zhang, and Nektaria Kaloudi

In: Conference on Computer Safety, Reliability, and Security (pp. 473-480).

Abstract: Many companies are studying autonomous vehicles. One trend in the development of control algorithms for autonomous vehicles is the use of deep-learning approaches. The general idea is to simulate a human driver's decision-making and behavior in various scenarios without necessarily knowing why the decision is made. In this position paper, we first argue that traditional safety analysis methods need to be extended to verify deep-learning-based autonomous vehicles. Then, we propose borrowing ideas from the process of issuing driving licenses to human drivers to verify autonomous vehicles. Verification of autonomous vehicles could focus on sufficient training as well as mental and physical health checks. Based on this position, we list several challenges that need to be addressed.

Secondary Paper 2

A comparative study of STPA-extension and the UFoI-E method for safety and security co-analysis,

Nelson H. Carreras Guzman, Jin Zhang, Jing Xie and Jon Arne Glomsrud

In: Journal of Reliability Engineering and System Safety, 211, 107633.

Abstract: Emerging challenges in cyber-physical systems (CPSs) have been encouraging the development of safety and security co-analysis methods. These methods aim at mitigating the new risks associated with the convergence of safety-related systemic flaws and security-related cyber-attacks that have led to major losses in CPSs. Although several studies have reviewed existing safety and security co-analysis methods, only a few empirical studies have attempted to compare their strengths and limitations to guide risk analysis in practice. This paper bridges the gap between two novel safety and security co-analysis methods and their practical implementations. Namely, this paper compares a novel extension of the System-Theoretic Process Analysis (STPA-Extension) and the Uncontrolled Flows of Information and Energy (UFoI-E) method through a common case study. In our case study, the CPS under analysis is a conceptual autonomous ship. We conducted our comparative study as two independent teams to guarantee that the implementation of one method did not influence the other method. Furthermore, we developed a comparative framework that evaluates the relative completeness and the effort required in each analysis. Finally, we propose a tailored combination of these methods, exploiting their unique strengths to achieve more complete and cost-effective risk analysis results.

Secondary Paper 3

Safety and security analysis for autonomous vehicles,

Robert Taylor, Jin Zhang, Igor Kozin, and Jingyue Li. *Technical University of Denmark*

Abstract: This technical report has been composed to provide a consolidated framework for these extensive studies. The purpose is twofold. Firstly, we wish to avoid the repetitiveness of reiterating the safety analysis background every time we publish a detailed study. Secondly, we aim to eliminate the need to start from scratch for every new project on AV safety and security. In this report, we present the development of a comprehensive range of methods to assess and improve the safety and security of AVs. The methods proposed are not only theoretical but have also been practically tested with the help of an actual AV design project, specifically a 1/4 scale vehicle design.

Secondary Paper 4

An empirical study on cross-data transferability of adversarial attacks on object detectors,

Alexander Michael Staff, Jin Zhang, Jingyue Li, Jing Xie, Elizabeth Ann Traiger, Jon Arne Glomsrud, and Kristian Bertheussen Karolius.

In the 41 Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence (SGAI 2021), AI-CyberSec 2021 workshop.

Abstract: Object detectors are increasingly deployed in safety-critical systems, including autonomous vehicles. Recent studies have found that object detectors based on convolutional neural networks are fundamentally vulnerable to adversarial attacks. Adversarial attacks on object detectors involve adding a carefully chosen perturbation to the input, which causes the object detector to make mistakes. The potential consequences of adversarial attacks must be known to make sure these safety-critical systems are reliable. This paper investigates the influence of transfer attacks on object detectors, where the attacker does not access the target detector and its training set. Devising an attack with this assumption requires the attacker to train their model on data that resembles the target detector's training set. Using their model as a surrogate, attackers can generate adversarial attacks without accessing the target detector. Our study investigates whether one can effectively attack a black box model using publicly available data. We have performed targeted objectness gradient attacks on the state-of-the-art object detector (i.e., YOLO V3). Initial transferability between the attacking and target model is low. However, increasing attack strength from 8 to 24 strengthens transferability and reduces the target detector performance by about half. Transferability is also studied when the datasets for the attacking and the target model intersect. Attack performance is proportional to the size of the intersection. With the stronger transferability caused by intersecting datasets, attack strength can be dropped to 16 and retain the attack performance.

Secondary Paper 5

Monitoring the robustness of safety critical artificial neural networks,

Jin Zhang, Josef Oehmen, and Igor Kozin.

In: European Safety and Reliability Association Newsletter, 3, 4-5.

Abstract: ANNs play a crucial role in executing safety-related tasks such as object detection, image recognition, navigation, and control in AVs. However, there have been instances where AVs using ANNs have misbehaved due to incorrectly comprehending sensor input variations or diverse environmental conditions, leading to accidents and failures. This newsletter highlights the challenges in monitoring the performance of ANN models in real-world operational domains. Changes in data-gathering modules, operational data shifts, and adversarial attacks can affect the performance of ANN models over time, leading to degraded predictions and reduced trustworthiness of the system. Additionally, the decision-making process of ANNs is often considered a black box, making it difficult for humans to understand why and how decisions are made. To address these challenges, the newsletter presents a research initiative aimed at building decision-making support tools for understanding the reliability of ANNs in safety-critical systems. The research focuses on analyzing how ANNs interact with other components and human operators. It identifies three key aspects of ANN reliability and robustness: component reliability of ANNs, system reliability, and interaction with human operators. Furthermore, the newsletter highlights the lack of automated methods/tools to help safety operators interpret and trust ANN predictions during system operations.

ISBN 978-82-326-7428-2 (printed ver.)
ISBN 978-82-326-7427-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

DTU

NTNU
Norwegian University of
Science and Technology