Safety through Machine Learning Applications

A Safety Case Analysis

Freek Jacobs
Master Thesis
October 2018

TU Delft Delft University of Technology

Safety Through Machine Learning Applications

A Safety Case Analysis

by

Freek Jacobs

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday October 26, 2018.

Student number:	4106792	
Project duration:	March, 2018 – October, 2018	
Thesis committee:	Prof. dr. ir. A. Verbraeck,	Chair
	Dr. S. W. Cunningham,	First supervisor
	Prof. dr. ir. P. H. A. J. M. van Gelder,	Second supervisor

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Acknowledgements

I would first like to thank the members of my graduation committee for their great support and enthusiasm throughout the completion of this thesis. There was no joint meeting that did not end in a passionate discussion where new ideas or angles followed each other in quick succession. This thesis largely consists of these ideas. Alexander Verbraeck, thank you for your excitement, involvement and patience. I genuinely appreciate how you always took the time to answer my questions and doubts during meetings, how you rapidly responded in detail when I asked you for your thoughts, and how you motivated me with your enthusiasm to keep giving my best. Pieter van Gelder, I would like to thank you for your close involvement throughout the process. Your interest and ideas helped me to add a whole different angle to my thesis, and your down-to-earth-approach enabled me to put matters into perspective. Scott Cunningham, I would like to give special thanks to you. Without your elaborate guidance, I would not have been able to complete this thesis the way it is. You always took your time to schedule our meetings, provided me with valuable feedback and ideas whenever I needed it, and remained enthusiastic, patient and involved during the whole process.

I would also like to thank my fellow students and friends for lending an ear whenever I needed it. Special thanks are for Quinn and Arushi who were present with me at De Wijnhaven during the summer break, so I did not have to eat my wok alone. I would also like to thank the members of my band De Klittenband and my roommates for being supportive of me through times where combining thesis and other activities was challenging.

Finally, I must express my very profound gratitude to my parents and brothers for providing me with endless support and unceasing encouragement not only during the process of writing this thesis but also throughout the rest of my years of study. This accomplishment would not have been possible without them. A big thank you for this.

Freek Jacobs
The Hague, October 2018

Summary

During the past decade, machine learning has developed from a buzzword to a technology that is increasingly influencing our daily lives. Applications of machine learning algorithms can be found in the industrial sector, but also in healthcare, financial markets, social media, transportation, law enforcement, and many other sectors. Part of the machine learning market are algorithms in safety-critical applications. Systems that were once controlled by humans are increasingly taken over by machine learning algorithms. Ensuring safety in machine learning applications is not straightforward. A variety of standards exist to ensure safety for mechanical devices, but not for machine learning applications. Standards and conventions are missing to validate the safety of safety-critical socio-technical machine learning applications.

Research goal

The question that is addressed in this research is:

> What are safety considerations when using machine learning for socio-technical safety-critical applications?

Four sub-questions break the research question down into smaller parts:

1. What is revealed by looking at machine learning and safety in an integrated manner with four different approaches?
2. How can a framework be developed with the findings of SQ1?
3. Based on the findings of SQ1 and SQ2, what conclusions can be drawn concerning machine learning capabilities?
4. Based on the findings of SQ1, SQ2 and SQ3, what conclusions can be drawn concerning organisational capabilities?

Methods

To answer the research questions, a mixed-method research design is chosen, consisting of conceptual analysis, interviews, content analysis and simulation. These methods were chosen in a way that weak points of one method are balanced out by strong points of another method. The conceptual analysis is intended to unravel the theoretical combination of machine learning and safety. Semi-structured interviews are used as a way to validate whether concepts found during the conceptual analysis are relevant in practice. The content analysis takes the findings of the conceptual analysis to examine whether two commonly used safety standards, the ISO 31000 and the IEC 61580, are fit for assessing machine learning applications. Lastly, the simulation adds a quantitative part and is intended to reveal error trade-offs on a software level.

Findings and conclusions

The mixed-method approach revealed a number of insights. Firstly, it became clear that dealing with epistemic uncertainty is inherent to machine learning applications and complicates the validation of these systems. Strategies from safety engineering and management theory were taken as a guideline to analyse ways to increase the safety of machine learning applications. A second insight is the challenge to cope with type I, type II and type III errors in cyber-physical. It was shown that both cyber systems and physical systems separately inherit serious safety risks. When combined as is the case with machine learning applications, these risks are even greater than the sum of both. The conceptual analysis also showed that, although research is done about practical implementations of machine learning applications and safety, actual implementation into practice remains troublesome.

Two interviews were conducted: one with an expert in autonomous vehicles and one with an expert in chemical industries. Both validated the findings from the concept analysis about machine learning application challenges such as inherent uncertainty, error trade-offs, impact or errors and strategies to increase safety.

The formal analysis concluded that current general safety standards are not yet adapted to suit machine learning applications. The ISO31000: 2018 and the IEC61508 were evaluated in light of earlier findings concerning the character of machine learning applications. The fact that machine learning algorithms can make wrong predictions is something that the aforementioned safety standards do not deal with. For standard-setting bodies, it remains a challenge to include the ability of wrong predictions and non-deterministic behaviour into general standards.

The simulation showed that classifier choice, feature size, sample size and performance metrics all have an impact on the type I and type II error trade-off. Combined with the findings concerning practical consequences of these errors, the importance of choosing an adequate trade-off is shown. Algorithms that perform equally might make different trade-offs between types of errors to achieve this performance. Therefore, a metric was proposed that explicitly includes this trade-off instead of just a performance score.

The conceptual framework for the international classification for patient safety was taken as a starting point for developing a framework to address machine learning and safety. Management and engineering safety strategies that were found after answering SQ1 were used to form a basis for the framework. The type of error, and environmental, social and organisational harm were included in the framework as factors since their relevance was pointed out in SQ1. The framework starts from the occurrence of an error, detection of the error, reduction of the impact of the error, and finally dealing with the consequences of the error. Following the steps of the framework offers a new way for the risk analysis of machine learning applications since it provides a structured way to gain insight in the prolongation of an error in these systems.

Answering SQ1 and SQ2 made it possible to reflect on how far we can take machine learning in socio-technical safety-critical applications. Currently, one can conclude that machine learning algorithms are never flawless. This posed the question about considerations for choosing the right context- or domain-specific solution for machine learning applications. A synthesis across the chapters of this thesis resulted in a list of considerations. The first consideration is the cost of type I and type II errors. Since algorithms will inevitably fail at a point in time, knowing the cost of errors is crucial. Consequently, one needs to consider what rates are acceptable, i.e., what is the desired trade-off between type I and type II errors? Following up this consideration, one has to estimate whether these rates can be approached with the current state of algorithms. And finally, what strategies can be put in place in case an error is made? All of these considerations play an important role in determining whether machine learning can increase safety for given applications.

Stage-gate models were used as a structured guideline to discuss organisational capabilities. In stage 1, the risk manager should specify what tasks machine learning applications have to take over. The next stage involves a risk analysis by following the steps in the framework of SQ2. During stage 3 and 4, the software and hardware are developed and tested. Since validation of machine learning applications is still an open challenge and proper safety protocols are missing, companies carry a great responsibility to come up with adequate safety strategies.

Recommendations

Recommendations for further research include:

- *Expansion towards specific contexts.* The findings of this study with regard to design, validation and error trade-offs can be expanded to specific industries. This way, safety hazards can be mapped and solutions can be sought in a structural way.

- *Adaption of existing safety standards.* In this thesis, it was concluded that existing safety standards are not suited for validating or certifying machine learning applications. Although some suggestions were made to improve the IEC and ISO standards, this field is open for research.

- *Legislation.* Government legislation in the area of safety-critical socio-technical machine learning applications is a field that needs further research. As was shown in this thesis, legislation is underdeveloped. It will be a major challenge for governments to legislate the coming technological proceedings in the field of machine learning applications. Findings in this thesis offer a start for solving this challenge.

- *Type I and II error trade-off analysis.* It is opted in this thesis that a cost-benefit analysis can give insight into the desired type I and type II error trade-off. Moreover, Bayesian decision theory can then address uncertainty and can help to find robust policy options in light of social values. Working out this method is open for further research.

Contents

List of Figures

List of Tables

1 Introduction

During the past decade, machine learning (ML) has developed from a buzzword to a technology that is increasingly influencing our daily lives. Applications of ML algorithms can be found in the industrial sector, but also in healthcare, financial markets, social media, transportation, law enforcement, and many other sectors. The market share of artificial intelligence reflects this increase. [47] expects the artificial intelligence market share to grow from 357.89 million U.S. dollars in 2016 to 31,236.92 million U.S. dollars in 2025 (see figure 1.1).

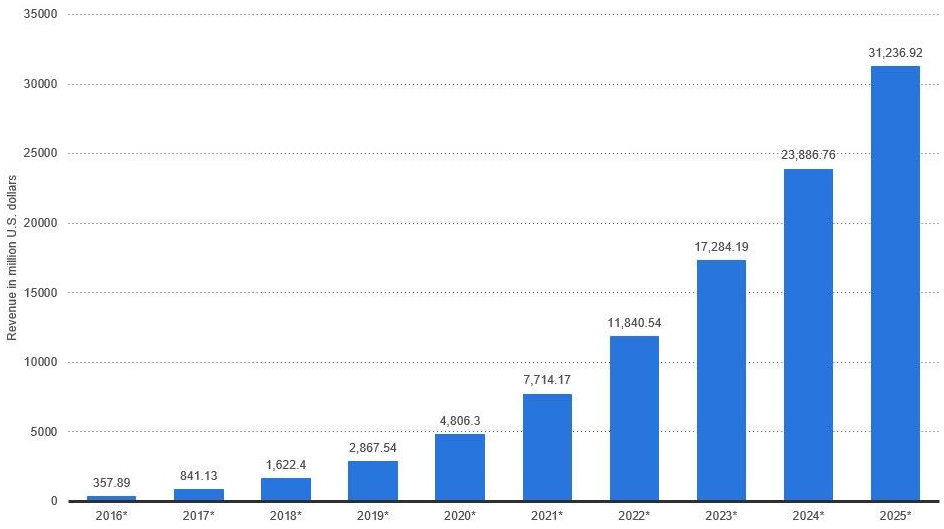


Figure 1.1: Global annual revenues from applications of artificial intelligence for enterprises, from 2016 to 2025. Reprinted from [47].

Part of the ML market are algorithms in safety-critical applications. Systems that were once controlled by humans are increasingly taken over by ML algorithms. We rely on these algorithms in fields like (semi-)autonomous cars, medical diagnosing, nuclear plant control and cybersecurity. Lots of research is done in these fields, and practical implementations are already existent.

Relying on these new algorithms brings opportunities, but also risks. This new technology is good at taking over tasks which humans are bad at, such as finding patterns in a lot of data or doing repetitive work. Algorithms cannot be tired, unmotivated or drunk like their human counterparts. However, algorithms have their weak sides too. These weak sides might inherit other risks than the ones we are already familiar with. Therefore, it is crucial to know potential safety problems when we rely on ML applications in safety-critical systems.

Ensuring safety of ML applications is not straightforward. Self-learning algorithms are often characterised as "black boxes", illustrated in figure 1.2. For example, an image of a dog is being fed as input into a labelling algorithm. The algorithm outputs the label "cat", which is incorrect. However, the cause of this incorrect

Figure 1.2: Schematic drawing of the black box principle. Reprinted from [106].

output is often unknown. Once training data have been fed to such an algorithm, it is complicated to figure out why it gives a particular response to a given input [129]. This can be troublesome for general applications of data science. [161] stated that applications of data science are usually divided into two extremes:

1. To discover new knowledge from data to inform decision makers.
2. To analyse data that is used by autonomous systems to make decisions.

Both applications demand data science outcomes to be correct and unbiased. However, it is acknowledged that biases are inherited in ML algorithms through programmer choices and the type of training data that was chosen [182]. Furthermore, other limiting factors can be the quality of the algorithm, the quality and quantity of the data, or ethical implications [106].

Practice shows that ML algorithms can indeed go wrong, sometimes with severe consequences. In 2016, a Tesla autopilot mistook a white truck for a clear sky, which led to a fatal collision [69].

In case of failure of this new technology, there are some new implications as well. One of these implications is liability in case of a wrong decision. Previously, it was the human controller. But what if it is a machine that makes a wrong decision with deadly consequence? Will the engineer be held responsible for designing a faulty code? Or will it be the manufacturer for implementing the code in its product? Another implication is risk acceptability. As a society, we know and accept that humans make errors from time to time. However, do we accept a machine to make mistakes with deadly consequences? The considerations are safety-related, but this field is not yet well-developed with regard to ML applications.

In the engineering world, safety can be defined as "a measure of absence of failures or conditions that would render the system dangerous" [66, p. 4]. A variety of standards exist to ensure safety for mechanical devices. For ML algorithms, this is not so clear-cut. Standards and conventions are missing to assure the safety of safety-critical socio-technical ML applications. At the current state of technical development, ML algorithms will inevitably fail at a point in time. Moreover, because of their black box characteristics, it is complicated or even impossible to pinpoint when this happens. The research gap of interest is, therefore:

> Standards and conventions are lacking for designing a safety case for safety-critical socio-technical ML applications.

ML software is only an element of a whole safety-critical socio-technical system. Therefore, it might seem odd to only talk about the safety of this specific element. Nevertheless, when software was introduced into safety-critical systems, it required a whole new approach to safety case design [104]. Following this analogy, ML introduces new challenges to safety case design as well. These challenges are the main topic of this thesis.

1.1. Scope

ML applications safety is a reasonably broad research field. To narrow this field down, this research is scoped based on the variables that are shown in figure 1.3. The software processing part of ML applications will be treated as a black box. One can look at these algorithms as white boxes and research ML transparency and accountability. This approach is very algorithm-specific and tends towards a computer science problem rather than an Engineering and Policy Analysis (EPA) problem. Therefore, the white-box approach is not considered here.

Technical versus social addresses the data processing chain versus a decision making part. Both social and technical parts are important when analysing decision making based on sensor data. Therefore, a middle ground was chosen for this research.

Single strata versus cross-cutting research is the consideration to look at only one part in the chain from sensor to decision making or to look at the chain as a whole. Errors or inconsistencies at the start of the

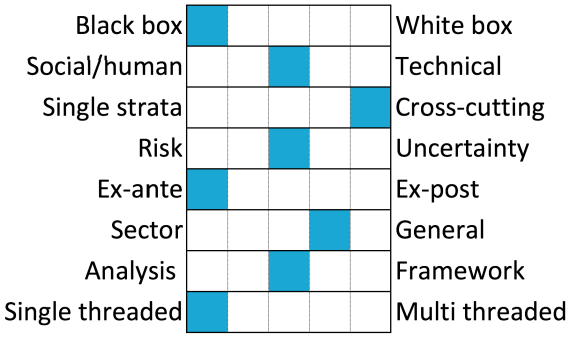


Figure 1.3: Considerations and their gradings for scoping this research. The blue squares show the tendency towards one of the two variables.

chain can cause ill-informed decisions at the end of the chain. Looking at the whole chain for this reason is important.

Risk and uncertainty are challenging topics in ML. Algorithms are trained to minimise empirical risk based on their training data. Uncertainty comes into play when the training data do not form a good representation of the real world data. To increase safety, both risk and (epistemic) uncertainty have to be minimised [180]. Thus, for this research, both terms are equally important.

This thesis will aim at analysing systems before a safety compromise has happened. It will attempt to map what can go wrong, and how this can be prevented. Therefore, an ex-ante approach is favoured over an ex-post approach.

Many sectors face the challenge of analysing safety when integrating smart systems into their processes. To address the problem for multiple sectors, a general approach is chosen for this thesis. Only in the last part, the theoretical findings will be applied to a sector-specific case.

Both analysis and the construction of a framework will be executed in this thesis. For a contribution to the identified knowledge gap, ML and safety need to be analysed. Secondly, a generalisation will be made to sketch the contours of a framework to execute a safety analysis for applications of ML in different safety-critical sectors.

To narrow the research down further, a single-threaded approach is chosen over a multi-threaded approach. The motivation for this choice is that the sensor (in this case a camera) is seen as a source. After processing and implementing data from this single source, a variety of decisions are made. Inevitably, decisions will also be made based on other information and developments. Nonetheless, this thesis will only look at the decisions that will be influenced by uncertainty or incorrectness of a sensor and its processing chain.

1.2. Goal, target audience and research question

The goal of this thesis is to give an overview of current challenges in the field of safety of safety-critical socio-technical ML applications in the form of a conceptual framework. This overview can help risk managers in safety-critical socio-technical systems to gain an understanding of the implications of introducing ML techniques for safety. Currently, it is not clear how to build a safety case. This thesis provides insight into considerations for risk managers when using ML for safety. It should provide insight into the question of whether automating safety-critical systems with smart solutions will lead to safer systems.

Secondly, the goal of this thesis helps software engineers of ML algorithms for safety-critical solutions to think about safety implications of their work. These engineers usually are only occupied with the goal to optimise algorithm performance. Practical implications of an error in case an algorithm makes a wrong decision are not their direct concern. For this reason, this thesis aims to give software engineers insight into practical implications and safety issues as a consequence of incorrect algorithm predictions.

Based on these goals and target audience, the following general research question is posed:

What are safety considerations when using ML for socio-technical safety-critical applications?

Sub-questions (SQ) are formulated as:

1. What is revealed by looking at ML and safety in an integrated manner with four different approaches?
2. How can a framework be developed with the findings of SQ1?
3. Based on the findings of SQ1 and SQ2, what conclusions can be drawn concerning ML capabilities?
4. Based on the findings of SQ1, SQ2 and SQ3, what conclusions can be drawn concerning organisational capabilities?

1.3. Research methodology

ML and safety is a broad field. To get a better grasp of the problem, this field of ML and safety is subdivided into four parts which will help to answer the research questions:

- *Conceptual.* The conceptual part is about safety, risk, uncertainty and errors. It inherits questions about how the concepts of safety and risks are defined, and how safety through ML applications can be achieved theoretically.
- *Practical.* The practical part of safety and ML incorporates how ML is currently used in industry in safety-critical socio-technical applications. What are practical challenges and best practices to govern safety?
- *Formal.* Industrial safety standards are meant to guard safety in industrial-technological applications. Are these standards ready to incorporate ML applications and their safety risks?
- *Numerical.* The output of ML applications can essentially be quantified in terms of error rates. What parameters or design choices influence those error rates, and what are the consequences?

Figure 1.4: Different angles to ML and safety in socio-technical safety-critical applications.

1.3.1. Methods

A research methodology can be designed based on the subdivision in figure 1.4. In order to include all four angles, both qualitative and quantitative methods should be combined. Therefore, a mixed method research design is used. This research design includes the following four methods: conceptual analysis, interviews, content analysis and simulation. These methods were chosen in a way that weak points of one method are balanced out by strong points of another method. This way, triangulating the methods results in findings that give more insight into the research problem than just executing them separately. The methods and their relation to each other are explained in the next subsections and illustrated in figure 1.6.

Conceptual analysis

Conceptual analysis was chosen to unravel the theoretical combination of ML and safety. This method is often used to break down or analyse concepts to gain a more thorough understanding [25]. By using conceptual analysis in this research, concepts of safety and ML can be analysed to obtain a deeper understanding of the combination. A significant advantage of this method is that an extensive collection of literature is available for analysis. However, it is a descriptive method, and it will not tell anything about the relevance or validity of concepts in practice [26]. This shortcoming is countered by conducting interviews.

[133] presents an outline for systematic concept analysis which will be followed in this thesis. Figure 1.5 illustrates this outline. The steps in this outline are in practice overlapping and not linearly followed [133]. Nonetheless, they form a guideline to execute the analysis. The goal (step 1) and preliminary framework for the analysis (step 4) will be elucidated in the introduction of the relevant chapters.

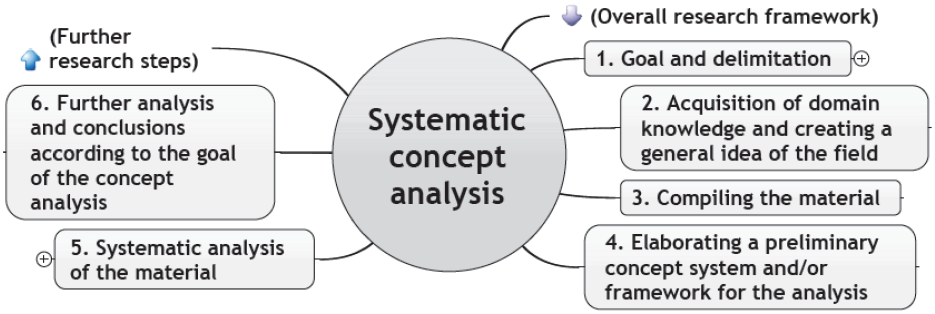


Figure 1.5: Steps for systematic concept analysis. Reprinted from [133].

Interview

Interviews will be used as a way to validate whether concepts found during conceptual analysis are relevant in practice. The appropriate interview style for this is 'semi-structured' since it allows interviewees to be open and express their view in their terms. At the same time, interviewees can be directed towards concepts that were found during the conceptual analysis. Interviews are however time-consuming, and consequently the sample size of interviewees is small. Nonetheless, since the interviews will only be used as a way of validating the conceptual analysis and not to generalise findings, the small sample size is not a problem. Another downfall is subjectivity. Interviews are the collection of subjective data, not objective facts. To make up for this, content analysis is used.

Content analysis

Content analysis allows describing characteristics of communication forms [84]. It facilitates to look directly at formal safety standards, so it transcends external interpretations or opinions. On the other hand, content analysis has limited data content, so other forms of information are missing. Conceptual analysis, which assesses a variety of data, makes up for this. A second disadvantage of content analysis is that it does not have a theoretical base or that it can be too liberal. In this research, the conceptual analysis provides a theoretical basis. Concepts that were found using this method are used to review the safety standards in this content analysis.

Simulation

All of the past three methods lack a quantitative part, while this dimension can help to achieve more insight into the quantification of errors instead of just the qualitative implications. It, therefore, provides a quantitative angle to all three research questions. To add this quantitative dimension to this research, simulation and

analysis of synthetic data is used as a method. Simulation allows for a quantitative "what happens if..." evaluation of scenarios by varying essential parameters [54, 92]. A disadvantage is that simulation, in this case, does not give practical implications of the results. Fortunately, the combination of the conceptual analysis and interviews gives a link to practice when combined with the simulation results.
The simulation will be based on a case study. The next section elaborates on the selected case.

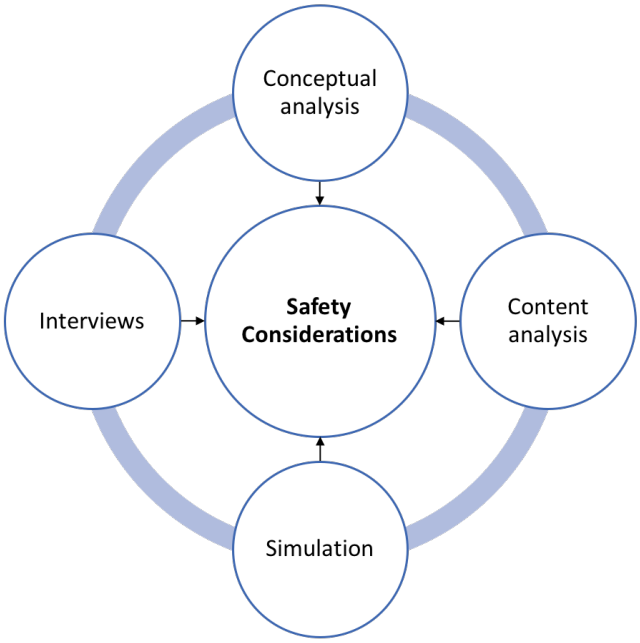


Figure 1.6: Research design for this thesis. The four outer circles represent the methods used. The inner circle represents the research question. The blue circle stands for triangulation of the four methods.

1.3.2. Case

One of the challenges of ML and safety, the trade-off between error types, will be explored in depth through a case study. Case studies are a renowned way of doing scenario analysis [17, 139]. Case studies can be classified as single- and multiple case [192], which is illustrated in figure 1.7. Due to time and data limitations of this research, the preferred method is a single case study. Apart from single- and multiple case designs, embedded and holistic design can be distinguished [192]. An embedded case study focuses on different sub-units of an entity, while a holistic case study focuses on the entity as a whole. Since a safety situation as a whole needs to be analysed in the proposed research, a holistic approach is a preferred methodology.

The Prorail case was chosen since it is very topical. Prorail is experimenting with smart cameras at this very moment since the Dutch government demanded them to use smart ways to improve carriage registration [56]. Implicitly, the assumption by the government was made that a smart system will be inherently safer than manual registration. This assumption is yet to be proven. Therefore, it will be interesting to evaluate the safety of the solution that is currently being implemented.

A well-known problem of executing case studies is generalisability [83]. While this is also true for this research proposal, valuable information can be extracted from a single case study. Findings of the Prorail case study can be transposed beyond the context of a single case. Also, readers of this case study can make their judgement whether findings correspond to the situation in their application. As noted before, there is insufficient time to do a multiple case study. Nonetheless, if future research would successfully replicate the proposed study for different sectors, the proposed framework will gain strength and credibility.
The methodology of the case study itself is described in detail in section 7.1.

1.3.3. Discussion on methodology

The methodology for this research was based on both the research questions and scope. When changing the scope (in figure 1.3), other methodologies would be appropriate. For example, let us consider ML as a white box instead of a black box, combined with a scope towards uncertainty instead of risk. This change in scope would opt for a predominantly quantitative approach where one would analyse algorithms to see

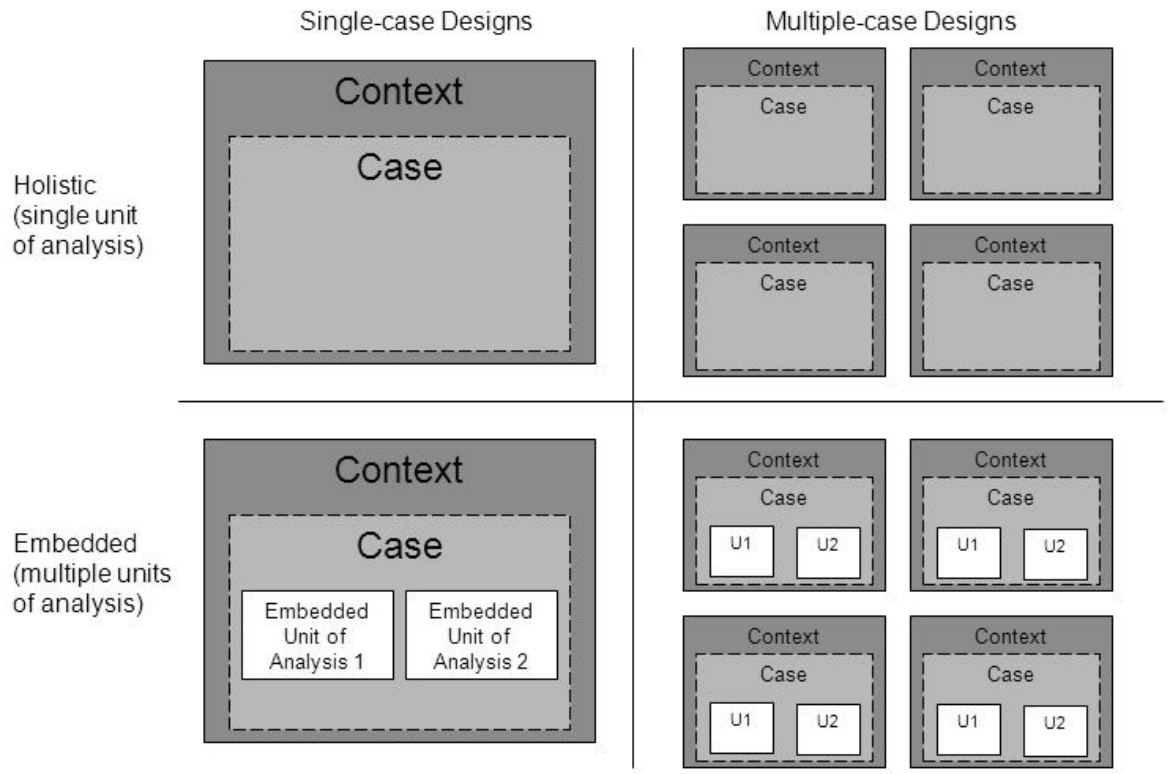


Figure 1.7: Different types of case studies. Reprinted from [193].

how confidence rates are defined, what they mean and how they are compared between different algorithms. Such research would contribute to the field of algorithm transparency. Given the same limited research time, however, it would be limited to a technical approach instead of including the social/human implications side.

Another exemplary research is to opt for a single strata approach instead of cross-cutting. One could look at the interface between man and machine to increase safety on this part. This research tends towards the social/human side of the scope. Also, a focus on a multi-threaded approach could give more insight into other factors than just algorithm performance (such as trust in technology) that influence safety and decision making based on algorithms. Again, given limited research time, this research would be more sector-specific since interfaces are different in every sector. Appropriate methods for such kind of research are pointed towards qualitative tools such as interviews.

Given the scope of this thesis, a mixed method approach cuts through all four parts of ML and safety as were illustrated in figure 1.4. Leaving one method out would take away balance to counter the negatives of another method. A thought experiment can demonstrate this. Let us start by taking out the conceptual analysis. This gap would have to be compensated by many interviews to explore the meaning of safety, ML and error types in different industries. It also takes away the theoretical basis for the formal analysis. And finally, it complicates linking results from the simulation analysis about error trade-offs to practical implications. Taking away interviews devaluates the results of the conceptual analysis since these are meant to prove practical relevance for the conceptual analysis. The results of the conceptual analysis are also used for the simulation and content analysis, so these would devaluate too. Taking away the content analysis would leave part of the problem definition of this thesis out by not stating what is lacking in current safety standards and how it should be improved. Lastly, leaving out the simulation part means leaving out the possibility to link quantitative software performance to qualitative implications, which is the essence of this research.

One sacrifice of the mixed approach is research depth. Executing four methods in the time span of this thesis will lead to a broader result than going in depth with just one method. However, the four methods triangulated will give a comprehensive overview of challenges still to overcome in the field of ML and safety. With these findings as a foundation, a different scope like the previous two examples in this section can be chosen to go more in-depth about specific topics. This thesis can then be used to place these findings into perspective.

1.4. Relevance to EPA

"Engineering and Policy Analysis (EPA) evaluates systems to deal with grand challenges" [1]. This thesis fulfils this requirement since it evaluates the safety aspect of ML, which can be used to contribute to the solution of a variety of grand challenges. The Prorail case of this thesis is a relevant example. Railway safety complies to a grand challenge, for a couple of reasons. Firstly, the public is at risk since the current lack of administration can lead to an inadequate response of firefighters in case of a hazardous goods leak, which in its turn can lead to the public being exposed to hazardous goods. Time is running out since it can happen at any moment. A perfect solution is yet to be found, since the current solution that is in the development phase (computer vision) is not flawless, and evaluation concerning safety has not happened yet.

A broader scope is found after zooming out. ML is not just an innovation, but is a piece of technology that has the potential to dictate our future lives. When safety is not ensured, this can have disastrous consequences. The technology of artificial intelligence is moving forward at a rapid pace, and algorithms are growing more and more responsible for essential decisions in safety-critical socio-technical systems. Therefore. it is important that we map safety hazards as soon as possible.

1.5. Chapter outline

In addition to the research methods section, a chapter outline was constructed to illustrate the flow of this thesis. Figure 1.8 shows this diagram.

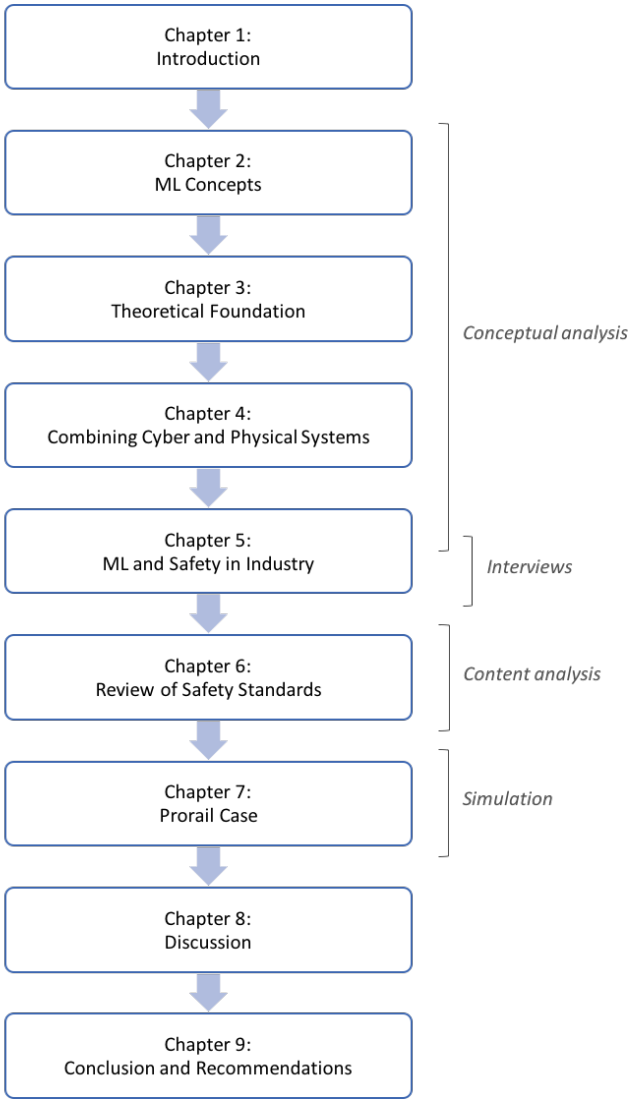


Figure 1.8: Chapter outline of this research.

The research starts with the conceptual analysis. ML related concepts such as artificial intelligence, supervised and unsupervised learning are analysed in chapter 2 since they are used throughout this research. The meaning of ML and safety, in theory, is explored in chapter 3, and open challenges are discussed. Chapter 4 outlines the risks of combining cyber and physical systems into cyber-physical systems by analysing type I, II and III error consequences. In chapter 5, conceptual analysis and interviews are used to show the risks of scaling up ML applications. Safety standards are analysed in light of ML applications in chapter 6. The last part of the methodology is executed in chapter 7, which is the Prorail case. All findings are interpreted in the discussion section, which forms an answer to SQ1. With these findings, a framework is constructed to answer SQ2. The discussion ends with a reflection on ML capabilities and organisational capabilities, which respectively answer SQ3 and SQ4. This thesis ends with a conclusion and recommendations for further research in chapter 9.

2

Concepts of Machine Learning

This chapter provides an introduction to basic ML concepts. Many of these concepts will be used throughout the rest of this thesis. When already familiar with the fundamentals of ML, the reader can skip this chapter. The goal is to sketch an overview of relevant terms and challenges within Artificial Intelligence.
For more detailed information on ML, please refer to [168] or other introductory materials available.

2.1. Artificial Intelligence

Artificial Intelligence (AI) comprises of all techniques that enable computers to show human behaviour. In recent terms, we would consider these techniques 'smart'. The idea of mechanical systems mimicking human behaviour dates back to Talos, an enormous automaton in Greek mythology. Talos had the task to guard Europa against invaders. The actual term ML was first mentioned back in 1956 on Dartmouth College during a summer workshop. People attending this workshop predicted that machines in no time would match human intelligence. However, it soon became apparent that the difficulty of AI was largely underestimated.

When progression remained absent, around 1974 funding of AI stopped and public interest in AI decreased [156]. The years that followed became known as 'AI winter'. From 1980, expert systems led to a period of growth for AI. These systems were able to solve problems for specific domains with information derived from domain experts. Although expert systems caused an AI boom, a second AI winter followed in the late 80s after the introduction of desktop computers caused a decreasing interest in AI.

During the last two decades, AI proliferated. This growth was made possible by the availability of big data, increased computational power and improved ML techniques. By 2016, the AI-related hardware and software market had grown to 8 billion dollars, and IDC predicts it to reach 47 billion dollars by 2020 [7].

AI can be classified into two groups: general AI and narrow AI. Narrow AI is more common and takes over specific tasks from humans. These tasks can include manoeuvring an autonomous car, predicting songs one might like on Spotify, or automatically recognising faces on Facebook. General AI attempts to comprehend a system that can take over any task from humans. That includes that general AIs can reason, sense, and execute things the same way humans do. General AI is the group of AI that popular figures like Elon Musk and Stephan Hawking warn about. However, general AI is still in its infancy.
AI research fields and methods include (but are not limited to) [2]:

- Machine Learning
- Natural language understanding
- Language synthesis
- Computer vision
- Robotics
- Sensor analysis
- Optimisation and simulation

One should note that these fields are often intertwined. For example, computer vision can be combined with ML techniques or optimisation and simulation frameworks to improve performance [157].

2.2. Machine Learning

ML is a subfield of AI, or in other words: it is a way to achieve AI. Back in 1959, Arthur Samuel defined it as "the ability to learn without being directly programmed" [116]. ML began to flourish in the 80s (figure 2.1). ML and AI are often used interchangeably, but it is possible to achieve AI without using ML. This would require one to manually construct complex rules and decision-trees [117].

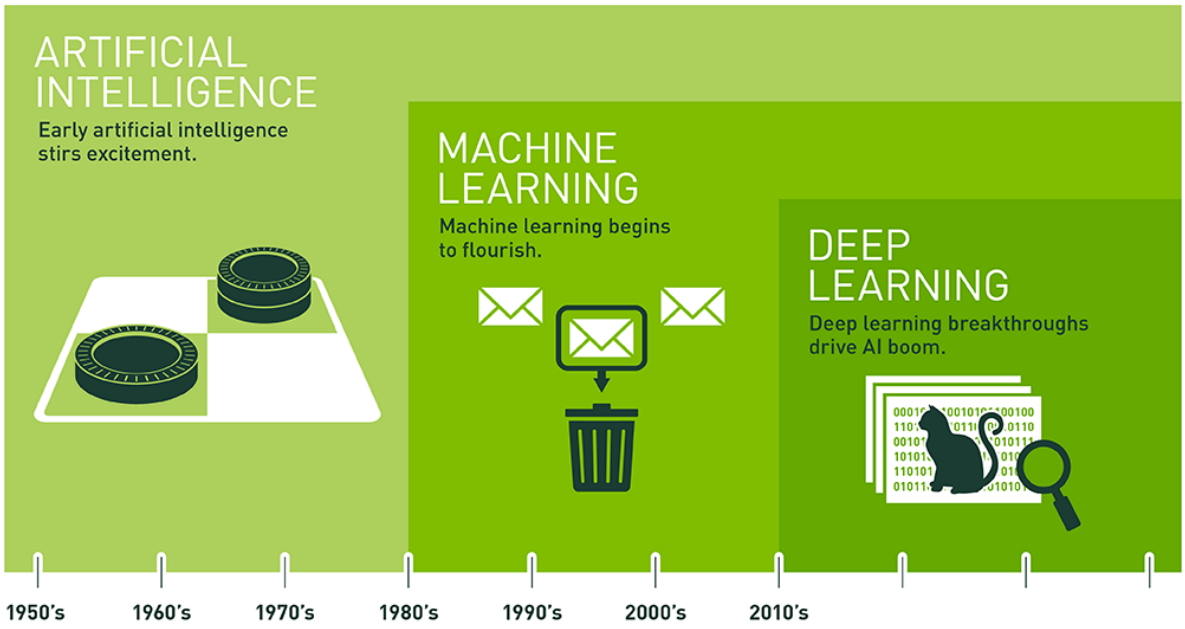


Figure 2.1: Artificial intelligence started in the 1950s. After that, ML (a subset of artificial intelligence) started developing in the 1980s, and deep learning (a subset of ML) in the 2010s. Reprinted from [50].

ML algorithms use inductive inference on their training dataset to make predictions for inputs that were not included in the training set. Instead of setting up strict rules to make predictions, ML is a way to train an algorithm, so it learns its own rules and adjusts itself to make better predictions. This learning process is based on data. Formally, [127] defines learning in the context of computer software as: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E".
Three types of ML can be distinguished [148]:

1. *Supervised learning.* Input training samples are labelled by a 'teacher', showing the algorithm what the correct output is for those specific input samples. The goal for the algorithm is to learn general rules that map the inputs x to the correct outputs y. Semi-supervised learning is a class of supervised learning where the training set typically contains a large amount of unlabelled data and only a small amount of labelled data.

2. *Unsupervised learning.* In this case, the training data is unlabelled, leaving the unsupervised learning algorithm to find patterns in its inputs. Goals of unsupervised learning can be to find patterns in the input data or to execute feature learning. Feature learning is a collection of methods to find features or representations in the input data that can be used for feature detection or classification.

3. *Reinforcement learning.* This is a reward-based approach to ML where an agent interacts with its environment to maximise its reward. Where supervised learning would reward an agent by performing a right set of actions, reinforcement learning rewards or punishes the agent for positive or negative behaviour.

Supervised, unsupervised and reinforcement learning will be further explained in the next sections.

2.2.1. Supervised learning

The concept of supervised learning is illustrated in figure 2.2. A training set [X,Y] is fed to the algorithm. It forms a hypothesis to map the input data X to the labels Y. Based on this hypothesis, it is able to map new input samples x to predictions y. More formally: a set of training samples $(x^{(i)}, y^{(i)})_{i=1...m} \in X \times Y$ is used to estimate the parameters θ which form the hypothesis $h_\theta(x) : X \implies Y$ [64]. This hypothesis is used to predict the output y for any input sample x.

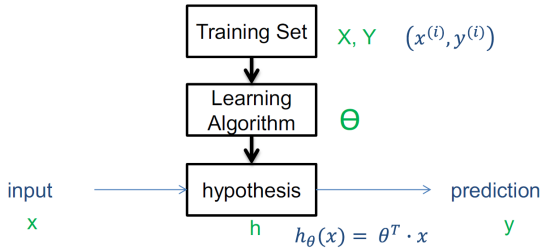


Figure 2.2: Supervised learning. Reprinted from [64].

Supervised learning is mainly used for two tasks: classification and regression. In classification, the algorithm predicts which discrete class a new input sample belongs to. E.g., the input is an image of a Ferrari, and the algorithm labels this image as "car". In regression, the algorithm tries to predict continuous valued outputs. An example of a regressional problem is trying to predict the price of a house based on its size and age.

2.2.2. Unsupervised learning

As opposed to supervised learning, unsupervised learning deals with unlabelled data. Based on the properties of this unlabelled data, unsupervised ML algorithms infer a function to describe the structure of the input data. Applications include clustering, dimensionality reduction or anomaly detection (figure 2.3). Clustering involves grouping objects with similar properties. An example is to cluster consumers in a customer database based on purchasing habits to allow more specific advertisement. Dimensionality reduction aims to reduce the number of random variables by obtaining a set of principal variables, which are a subset of the original variables and preserve (to a certain extent) the original information and structure [144].

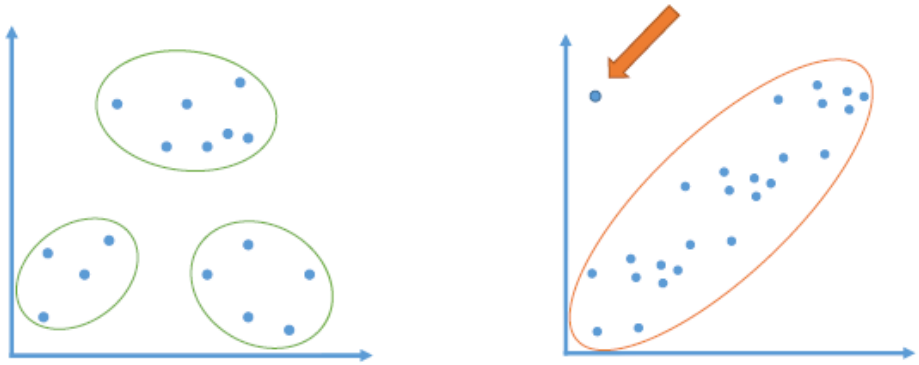


Figure 2.3: Clustering (left) and anomaly detection (right). Reprinted from [64].

2.2.3. Reinforcement learning

Reinforcement learning lies somewhere between supervised and unsupervised learning. An agent interacts with its environment and determines the ideal behaviour to maximise its performance. The agent receives feedback in the form of rewards and penalties. It learns by maximising rewards and minimising its penalties.

2.2.4. Fields and methods

ML research fields and methods include (but are not limited to) [2]:

- Deep Learning
- Support Vector Machines
- Decision trees
- Bayes learning
- K-means clustering
- Association rule learning
- Regression
- Artificial neural networks

2.2.5. Difference between Machine Learning and Statistical Modelling

To further clarify the concept ML, it is helpful to sketch the difference between ML and Statistical Modelling. A quick search on Google shows that online blogs like KDNuggets, Quora and StackExchange are regularly being used to discuss differences and similarities between the fields. In this section, an analysis of the differences is made.
Using formal definitions, Statistical Modelling can be described as:

> "a formalisation of relationships between variables in the form of mathematical equations." [189, "Introduction"]

ML is formally described as:

> "an algorithm that can learn from data without relying on rules-based programming." [132, p. 202]

Based on these definitions, one can conclude that ML is less strict on following formal mathematical rules than statistical modelling. This argument is backed up by [132].

Terminologically, there are some differences as well. Larry Wasserman states that both ML and statistical modelling are concerned with learning from data, but that they use different names for the same concepts [186]. Table 2.1 shows these differences.

Table 2.1: Terminological differences between ML and statistical modelling as identified by [186].

Machine Learning	**Statistics**
Estimation	Learning
Classifier	Hypothesis
Data point	Example/Instance
Regression	Supervised Learning
Classification	Supervised Learning
Covariate	Feature
Response	Label

Now that terminological and formal differences are stated, it is time to look at substantive differences. [35] offers a conceptual view on these differences (figure 2.4). Breiman considers a black box model for data generation, where a vector of input variables x goes in on one side, and response variables y come out on the other side. Nature associates input variables to response variables. Statistical modelling approaches the inside of the black box as a stochastic model and aims to estimate its parameters [35]. [35] states that ML, on the other hand, considers the inside of the black box to be unknown. The only aim of ML is then to find an algorithm f(x) which predicts the response y for a given x.

The differences sketched in the previous paragraphs have some practical implications. One of these is the amount of data that can be used. ML offers ways to improve predictions using the growing amount of data that is available these days, whereas statistical models reach a saturation point where accuracy does not

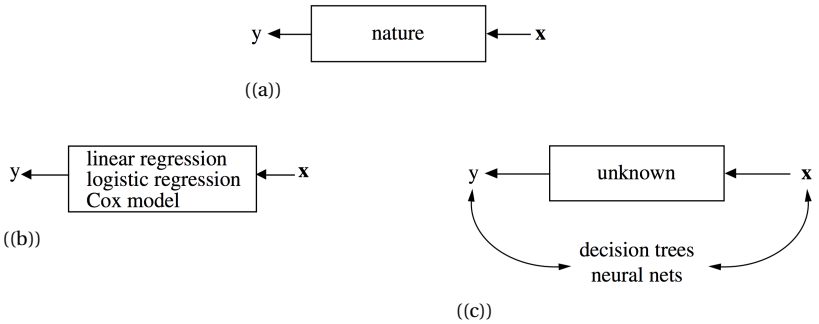


Figure 2.4: Conceptual illustration of the difference between statistical modelling and ML, where a represents the real world, b represents statistical modelling and c represents ML. Reprinted from [35].

improve after a particular feature or sample size [132]. Also, the computational cost of running ML models is much lower than running statistical models with the same feature and sample size [165].

The knowledge about the structure of the input data that is required a priori also differs. ML does not require any prior knowledge about relations between variables [160]. In statistical modelling, one must understand the collection process of the data, the underlying distribution, what happens in case the experiment is repeated, and properties of the estimator such as the p-value [160].

2.3. Deep learning

Deep learning is a subclass of ML where complex neural networks are used. Its structure is inspired by the way biological nervous systems communicate. As opposed to simple neural networks, deep learning neural networks are constructed of multiple hidden layers (figure 2.5). This construction enables more complex functions such as feature transformation and extraction [74].

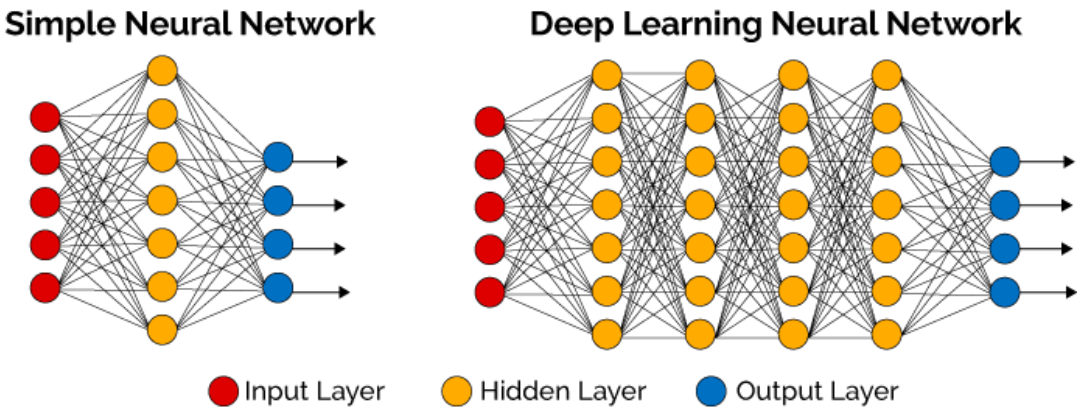


Figure 2.5: Simple neural networks versus deep neural networks. Reprinted from [74].

Deep learning research fields and methods include (but are not limited to) [2]:

- Convolutional neural networks
- Recursive neural networks
- Long short-term memory
- Deep belief networks

2.4. Challenges and next steps in research

Many challenges in the field of ML still exist. Based on recent blog posts [46, 110, 196], some of these challenges are explained in this section. Challenges regarding business and society are enumerated by [112]. Firstly, Marr mentions a lack of computer power. Although processing power has increased during the past decades, it still forms a bottleneck for ML. Quantum computing could offer a solution, but is years away from implementation. Another challenge is the lack of human resources. [112] states that a shortage of appropriately skilled people slows the growth of ML research and implementation. A third challenge that [112] states is public trust. As long as people do not understand what is inside the AI black box, they do not feel comfortable with the decisions that it makes. The final challenge that was mentioned by Marr is that AI solutions to this date are still narrow. General AI, although promising, is still miles away from implementation. Raia Hadsell, a Google DeepMind researcher, explained this as: "There is no neural network in the world, and no method right now that can be trained to identify objects and images, play Space Invaders, and listen to music" [28, p. 36].

Apart from the lack of computer power, skilled people and public trust, [46] lists some other challenges for ML projects. The first one is for people to understand the limits of ML. In many environments, ML evolved to a concept of which people think it will rapidly solve all of their problems. In practice, ML requires properly organised and structured data, skilled people, and specific goals. Secondly, the availability of data is a bottleneck. It requires time and money to collect and prepare data. Also, personal data is bound to privacy laws.

More specific challenges are mentioned by [110]. The first one is the inability so far to achieve one-shot learning, the learning from only one or a couple of examples. It offers excellent potential since one-shot learning does not need the large amount of data that ML needs typically. However, it is not there yet. A second challenge is developer bias, where ML algorithms are biased to benefit the developers. The example that is given by [110] is a medical algorithm which recommends expensive treatments over the most effective treatments. The third challenge named by [110] is including ethics into ML algorithms. ML algorithms can optimise desired variables, but taking ethical considerations into account when giving results is challenging. For example: will an automated vehicle differentiate between a child or an adult when an accident is imminent? Will it value the life of its driver over the life of a pedestrian, and will this change when the pedestrian appears to be a child?

2.5. Conclusion

This chapter aimed to clarify definitions relevant to ML. It explained the history of ML and how it differs from AI and statistical modelling. It was shown that ML can be subdivided into supervised learning, unsupervised learning and reinforcement learning. The chapter was ended with an inventorisation of remaining challenges in the field of ML. The next chapter will use the findings of this chapter to combine ML with concepts from the field of safety engineering.

3

Safety of Machine Learning Applications - Theoretical Foundation

For this thesis, it is essential to define what is meant by safety. This definition will help to conceptualise terms like risk, uncertainty and hazards. However, there is no universally accepted definition of safety. Its meaning is domain-specific, so safety regulations and requirements vary for different domains. This chapter will use the engineering definition of safety to express the safety of ML applications. It starts by defining the concept of safety. Then, the general loss function of ML algorithms is related to this definition to show safety concerns. Following these, strategies to overcome these safety concerns are stated. The chapter ends by analysing what challenges are still open for increasing the safety of ML applications, and how this thesis contributes to these challenges.

3.1. Defining safety

A quick search already leaves us with a variety of definitions:

1. Merriam Webster: "1: The condition of being safe from undergoing or causing hurt, injury, or loss. 2: a device (as on a weapon or a machine) designed to prevent inadvertent or hazardous operation." [3]
2. "Freedom from those conditions that can cause death, injury, occupational illness, or damage to or loss of equipment or property, or damage to the environment." [123]
3. "Safety is the property of a system that it will not endanger human life or the environment." [185]
4. "Safety is freedom from accidents or losses." [105]

These definitions vary in application and specificness, but they all sketch a desired situation. For a system to be safe, it is prescribed what should not happen. Therefore, safety can be defined as a system state where dangerous situations are absent.

In system engineering safety, the complete absence of dangerous situations is hard to guarantee. Instead, safety is aimed to be optimised to achieve an acceptable level of risk throughout the lifetime of a system [64]. In this context, risk is seen as the antonym of safety. According to [64], risk minimisation involves hazard identification, risk assessment and risk elimination or mitigation strategies. The success of these three measures is dependent on our knowledge of the system. Since this knowledge is never complete (e.g., external factors cannot be mapped entirely), a certain degree of epistemic uncertainty always plays a role in safety assessments [52, 103, 147].

In [130, 131], Möller proposed a definition of safety containing both risk and uncertainty. Möller defines safety as the minimisation of the risk and epistemic uncertainty of harmful events. The rest of this paragraph is used to describe the different parts of this definition, starting with harmful events. Harmful events are only safety issues when the severity is high enough to meet a certain threshold. In this context, severity can be defined as the number of casualties, the amount of damaged property, or in other ways depending on the domain. Risk in the safety domain is usually defined as the product of the severity of a harmful event, and its probability of happening. In other words: risk is the expected value of the cost of harm, which is the

weighted sum of the severity of all harmful events [130]. It is important to note that the probability distribution and the severity are assumed to be known in this definition. So despite not knowing the outcome, there is enough information to judge whether a risk is acceptable or not. This is not the case with epistemic uncertainty. With epistemic uncertainty, the probability distribution is not (entirely) known. [114] defines epistemic uncertainty as uncertainty "which results from our incomplete knowledge and could in principle be reduced, though this may be impractical, not possible in the framework of available time and resources, or many similar reasons." (p.107).
The next section shows how risk and epistemic uncertainty can be related to the general loss function of ML algorithms.

3.2. Loss function

In statistical learning, the goal is to find a function $h : X \rightarrow Y$ that maps the most probable label ŷ to the observation x. The loss function L is the prediction error, or in other words: some function of the difference between the predicted label h(x) and the true label y. The risk R(h) is then defined as the expected value of the loss.

$$\mathbf{E}[L(h(X), Y)] = \int_X \int_Y L(h(x),) f_{X,Y}(x, y) dydx \tag{3.1}$$

The objective is to find the function h that minimises R(h).

In an ML context, the probability density $f_{X,Y}$ is not accessible. What is accessible though, is a finite set of training samples drawn from the joint distribution (X,Y):{(x_1, y_1),...,(x_m, y_m)}. Using this training set, the goal is to find the function h so that the empirical risk $R_m^{emp}(h)$ is minimised. The empirical risk can be defined as:

$$R_m^{emp}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i). \tag{3.2}$$

Following this definition, [64] identified three areas of concern: the training data, model selection and the optimisation strategy.

3.2.1. Training dataset

[180] states that $R^{emp}(h)$ converges to $R(f)$ as m approaches infinity. While [64] supports this finding, it raises questions about the size and representativeness of training sets due to their finiteness. In practice, $R^{emp}(h)$ does not converge to $R(f)$ due to the following issues:

- The number of samples that is needed to form a good generalisation is undetermined and hard to define. A theoretical definition is existent (e.g. [30]), but it is questioned whether this theory is useful because the bounds are very loose in practice [57].

- In the notation given in section 3.2, it is assumed that the training samples are drawn from the true distribution of (X,Y). In practice, this might not be the case, which is a very relevant source for epistemic uncertainty in ML [180]. Some risks have not occurred in practice yet, such as black swan events, which prevent $R^{emp}(h)$ from converging to $R(f)$. Accordingly, [180] states that the true underlying distribution of (X,Y) can not be known in some situations. This prevents the use of covariate shift [164] and domain adaption techniques [53] (changing the input distribution while keeping the output distribution accurate) to increase robustness.

- Corrupted or badly curated data and sampling errors can be other reasons for training data to not accurately represent the underlying distribution of (X,Y). [64] states that data loss and corruption should be prevented by proper data management. [77] goes one step further. It claims that a backdoor can be built into ML algorithms by deliberately corrupting training data. [77] mentions the example of a street sign classifier that identifies stop signs as speed limits after special stickers have been added to the stop signs. This backdoor persisted even after the network was later retrained with a correct training set.

- The last source of epistemic uncertainty mentioned by [180], is caused by a small probability density. It might be the case that the training data distribution perfectly follows the real distribution of (X,Y),

but nonetheless, essential situations or domains in the input space are missing due to the rareness of certain training samples. This could arguably be an explanation for the Uber autonomous car accident in 2018, which killed a pedestrian crossing the road while pushing a bike (see [122]).

3.2.2. Optimisation strategy

As was shown in equation 3.2, the goal of ML algorithms is to minimise the empirical risk $R_m^{emp}(h)$. A risk of this strategy is overfitting. Overfitting is defined in statistics as "The production of an analysis which corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably" [5]. For ML, this "set of data" is the training set. To prevent overfitting, simply minimising empirical risk is not the best strategy. Therefore, a strategy called 'regularisation' is used which discourages the use of a complex function or model to reduce the possibility of overfitting. A regularisation term is added to the minimisation goal in equation 3.2:

$$\text{minimise}_h[R_m^{emp}(h) + \lambda * \text{comp(h)}]. \tag{3.3}$$

The function 'comp(h)' denotes the complexity of the hypothesis h, and the term λ denotes the weight factor that determines how big the role of comp(h) is in this equation. Increasing λ will result in less overfitting (variance), but increases the risk of underfitting (bias). These concepts are illustrated in figure 3.1.

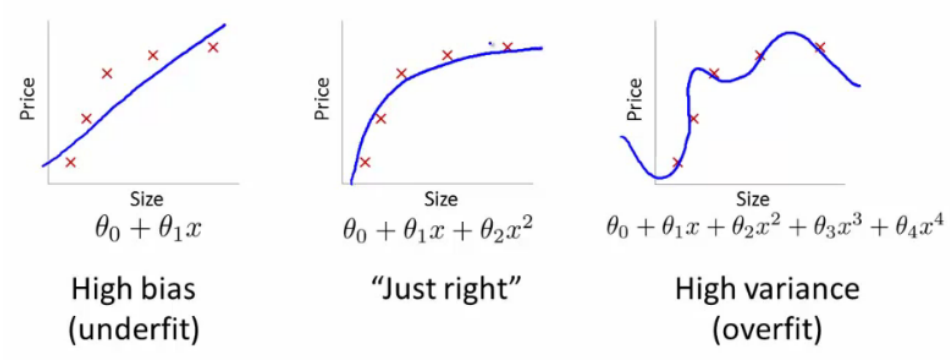


Figure 3.1: Illustrations of underfitting, the right fit, and overfitting in a regression problem. Image reprinted from [9].

Another point of interest is the loss function L(h(x)). Normally, the loss function is defined so that minimising the cost function maximises the correctness of the algorithm. Furthermore, the same loss function is used for every training sample. [179] points out that not all errors have the same impact in a safety context. He presents two suggestions to increase the application-significance of the loss function:

- The severity of a wrong prediction should be included in the loss function, as opposed to only including the difference between y and h(x);
- Instead of using only one loss function, multiple loss functions can be taken into consideration per training sample.

More solutions for increasing safety can be found when looked at safety engineering and risk management theory. These are explained in the next section.

3.3. Strategies to achieve safety

[131, 179] proposed four general principles of safety engineering:

1. *Inherently safe design.* The first principle suggests excluding hazards from the system as much as possible rather than to cope with these risks. Possible examples are to use fireproof materials instead of inflammable ones or to perform chemical reactions at lower temperatures and pressures.
2. *Safety reserves.* By implementing safety factors, structures should be designed to resist stress loads that exceed the maximum expected or intended loads. For example: by implementing a safety factor 2, a crane should be able to lift twice the weight of what is expected in practice.

3. *Safe fail.* In case a system does fail, safe fail systems should be in place to limit the impact or harm of a failure. When the air pressure in an airplane cabin drops, oxygen masks come out to prevent breathing issues due to a lack of oxygen.

4. *Procedural safeguards.* These safeguards are about procedures and protocols to maintain safety. Examples are quality control standards or proper staff training. Safeguards are important to identify new potential sources of harm (such as worn material) or to control employee behaviour which cannot be excluded from a design perspective.

In risk management theory, four similar strategies are known:

1. *Terminating risk.* Comparable to 'inherently safe design', this strategy aims to eliminate risk by removing risky practices or processes from an organisation.

2. *Treating risk.* This strategy is about decreasing the likelihood of the risk occurring, and is therefore comparable to 'safety reserves'.

3. *Transferring risk.* In risk management theory, transferring risk can be achieved through insurance or shifting responsibility to third parties. This strategy is best compared to 'safe fail', since another party or system takes care of the risk in case of failure.

4. *Tolerating risk.* In this last strategy, no effort is made to mitigate or reduce the risk. The motivation for this strategy could be for instance that measures are not cost-effective, the company lacks the appropriate resources, or the likelihood of the risk is neglectable. However, the risks are monitored in case circumstances change. This strategy is, therefore, best compared to the engineering strategy 'procedural safeguards'.

The next part of this section will analyse the literature on current industrial strategies to increase the safety of ML applications, grouped by the given four categories.

3.3.1. Inherently safe design

In an ML context, inherently safe design can be explained as robustness in case the training data is not sampled from the real distribution of (X,Y). In this case, the algorithm might show unexpected or unpredictable behaviour, which should be prevented from a safety point of view. Some strategies are posed to deal with this problem.

Interpretability

When an ML algorithm encounters an input it has not seen before in its training set, it can be of great support to understand how it forms its decision or prediction [188]. Merely trusting a well-performing algorithm is not enough. Take for instance a physician who would like to understand why the computer predicts that a patient is likely to develop cancer in the following years, and what the probability is that this prediction is correct. Or an autonomous car that shows its driver that it does not understand a situation on the road, and hands back control. A single confidence metric like the test error is not enough to describe the complex reasoning of an ML algorithm [58]. [58] argues that values like fairness, reliability, robustness and causality are missing.

However, interpretability is complex. ML algorithms do not form their decisions with the same reasoning as humans do. Many algorithms reason based on thousands or millions of input variables, making human interpretation rather complex. Some research was done in this area, e.g. [40, 109, 172]. There is a sense of urgency to this research since the new General Data Protection Regulation (GDPR) will enter into force by May 2018. Although not explicitly stated, [75] believes that the interpretability of decisions can be demanded based on the GDPR, as opposed to [183].

Formal verification

'Correct by construction and formal verification' is proposed by [64], and entails the use of formal procedures in software development. Articles like [95, 113, 159, 178] offer possible solutions or research directions for verifying ML algorithms. In this thesis, however, ML algorithms are mainly treated as black boxes, so this research falls out of scope.

Redundancy and dissimilarity
[64] argues that creating redundant architectures is a strategy for inherently safe design in ML. The concept is that multiple computers calculate results in parallel. A central voter checks what the most common answer is, and sets this answer as the final output. A comparable approach can be used with ML algorithms [23, 150].

3.3.2. Safety reserves
[179] mentions minimising the maximal test error instead of the average test error as a way to implement safety reserves.

3.3.3. Safe fail
A technique that is commonly used in ML is the reject option: when an algorithm is too uncertain about a particular prediction, it does not make a prediction and thereby fails in a safe way [181]. When this happens, a human or backup system should intervene. An intricate part of this strategy is that algorithms occasionally report high confidence while actually making wrong predictions. This typically occurs for areas in the input space X where the probability density of the training data is low [19].

3.3.4. Procedural safeguards
Non-specialists are often operators of decision-supporting ML systems. Therefore, [179] suggests user experience design to guide users to set up ML systems properly. This involves training data selection and evaluation procedures. The second procedural safeguard that [179] suggests is opening up data. Varshney argues that open data will increase the safety of ML systems through innovation and transparency. This argument is backed up by [94, 149, 162].

Another approach is the certification of software. Software certification in safety-critical systems is used to ensure that the software shows the desired behaviour under expected and unexpected circumstances. There are many different verification methods to ensure this behaviour [108]. For ML applications, verification methods are much harder to obtain, since the same algorithm can behave in unpredictable ways, depending on its previous training data. An autonomous car, for example, might show a particular behaviour when a cat crosses the road in front of it. Two months later, it might show utterly different behaviour when it encounters the exact same situation since in the meantime new training data changed the response of the automatic driving algorithm. Recently, some work was published on this topic [18, 63, 150].

3.4. Challenges
The strategies that were mentioned in the previous section offer a good start at combining ML and safety, but are mainly software related and proposed by computer scientists. These are valuable strategies, but they lack a practical part where implications are considered. E.g., none of the papers reviewed in this chapter address the implications of an ML-induced error in terms of harm. Literature in this area was not found. Thus, a challenge that remains is how to combine the fields of ML applications and safety on a systems level instead of a software level. This is the challenge that will be addressed in the rest of this thesis.

Some other challenges regarding the safety of ML applications are still open. These challenges do not necessarily fit the definition of safety, but are mentioned in this section since it can be argued that they are safety-related:

- *Privacy.* ML algorithms can deal with sensitive data sources such as patient information or valuable business information. With the output or behaviour of such algorithms, it can be possible to reason back to individual users. The differential privacy field is concerned with ensuring that private information is kept private while maximising the accuracy of the results [13, 91]. Open challenges in the field are for instance how to include public data or how to handle missing data in privacy-sensitive datasets.

- *Security.* The adversarial ML field is concerned with manipulating ML algorithms by feeding it malicious training data [85, 100]. These data are specifically designed to fool ML algorithms. The process of training algorithms to withstand malicious training data is called adversarial training. This field of research has so far mainly been applied to small problems, thus remains a challenge for future research [100].

- *Social impact.* What is the impact of ML and automation on society [38, 68]? An open issue in this field is how susceptible jobs are to the rising industrial applications of ML [68]. Frey predicts that the

replacement of labour by mechanical workforce could impact wages, employment, skill demand and the economy as a whole. Some even predict ML to cause a fourth industrial revolution since decision-making will mostly shift from humans to machines [171].

3.5. Contributions

This thesis contributes to the current scientific literature by using a new methodology design to combine the field of ML and safety on a systems level. This is an attempt to form a non-computer-science view on this field. The triangulation of conceptual analysis, interviews, content analysis and simulation, offers a new and exciting way to form a bridge between ML performance and practical safety implications. The two main contributions of this thesis are a synthesis of this bridge by providing a discussion on ML capabilities and organisational capabilities for increasing safety in socio-technical safety-critical applications. ML capabilities show how far we can bring this technology for increasing safety. Organisational capabilities show the steps and considerations when implementing ML applications for safety into an organisation. These two contributions are supported by a framework where safety strategies are structurally laid out.

3.6. Conclusion

This chapter aimed to come up with a theoretical definition of safety for ML applications and to derive potential problems and strategies to overcome these problems. ML was defined as the combination of risk and epistemic uncertainty. The notion of epistemic uncertainty arose safety-related questions regarding the training dataset and optimisation strategy. Inherently safe design, safety reserves, safe fail mechanisms and procedural safeguards were found as strategies to increase safety. The question remains if and how these strategies are executed in practice. This requires to look at the issues stated in this chapter at a system level. This will be addressed in the next chapter.

4

Combining Cyber and Physical Systems

Risk can be divided into two categories: the risk of type I and type II errors. A type I error is the rejection of a true null hypothesis (also known as 'false positive') whereas a type II error means failing to reject a false null hypothesis (also known as 'false negative') [166]. The errors are illustrated in table 4.1. In decision making, these errors can both lead to unsafe situations.

Table 4.1: Types of errors.

	When H0 is true	**When H1 is true**
Do not reject H0	Correct decision p = 1-α	Type II error p = β
Reject H0	Type I error p = α	Correct decision p = 1 - β

In this chapter, the consequences of type I and type II errors in the cyber domain and the physical domain will be explored and related to type III errors. Furthermore, it will be analysed what happens when type I and II errors in the cyber domain affect the physical domain, which is the case with ML applications. Usually, cyber systems and physical systems both have a physical and computational component. Therefore, the distinction between the two will be made based on the following typology:

1. *Cyber systems:* systems assessed with information processing and classification or regression.
2. *Physical system:* real-world tangible application.

This chapter will be limited to safety-critical applications.

4.1. Type I & II errors in the cyber domain

4.1.1. Predictive analytics

A major upcoming field in the cyber domain is predictive analytics [115, 167]. One of the applications of predictive analytics involves terrorism prediction. The aim is to predict future acts of terrorism by analysing social media and intelligence data [34, 120, 152]. Unfortunately, this research is plagued by many false positives because the group of true positives is too small according to [43, 121]. [43] illustrates that this is a common problem by using the following simplified Bayes example:
Imagine there is an algorithm which detects terrorism communication with 99% accuracy:

$$P(+|\text{terrorist}) = 0.99 \tag{4.1}$$

$$P(-|\text{no terrorist}) = 0.99 \tag{4.2}$$

In this case, the unknown variable is the chance of detecting a terrorist given a positive detection by the algorithm:

$$P(\text{terrorist}|+) = ? \tag{4.3}$$

Terrorists are fairly rare. [43] assumes that one in a million people are terrorists.

$$P(\text{terrorist}) = \frac{1}{1,000,000} \tag{4.4}$$

The unknown variable in equation 4.3 can now be determined using Bayes theorem:

$$P(\text{terrorist}|+) = \frac{P(+|\text{terrorist})P(\text{terrorist})}{P(+|\text{terrorist})P(\text{terrorist}) + P(+|\text{no terrorist})P(\text{no terrorist})} \tag{4.5}$$

$$= \frac{1}{10,102} \tag{4.6}$$

In other words, the false positive rate of this algorithm is:

$$FP = P(\text{no terrorist}|+) \tag{4.7}$$

$$= 1 - P(\text{terrorist}|+) \tag{4.8}$$

$$= 99.99\% \tag{4.9}$$

Even with an accuracy of 0.99, it can be concluded that the analyst will be confronted with more than 10,000 false positives for every real terrorist. The opportunity cost of looking at these false positives are possibly huge. The fact that a terrorist attack is characterised by a low probability but high impact makes this problem hard to solve with predictive analytics. In statistics, this problem is called the "base rate fallacy" [124, 155], where the number of samples is high, but the occurrence of the phenomenon sought for is low. The base rate fallacy dictates that an unrealistically high accuracy will likely still result in an unacceptable false positive rate. [124]. Figure 4.1 shows the relationship between accuracy and the true positive rate for the terrorist problem. The S-shaped curve shows that very high accuracy is needed for the true positive ratio to be acceptable.

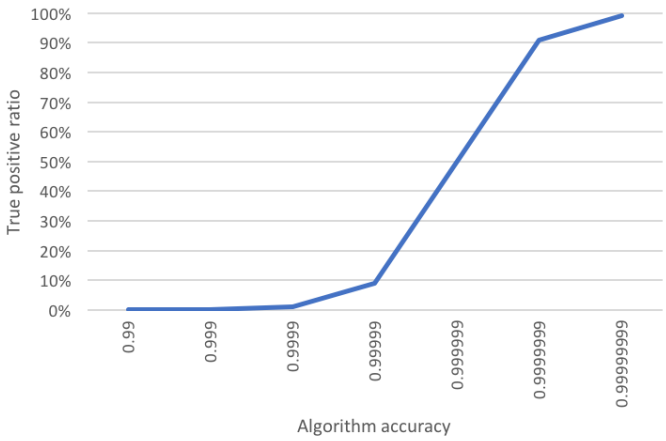


Figure 4.1: Behaviour of the true positive ratio as a function of the accuracy of the algorithm. The probability of a person being a terrorist is held constant at one in a million.

4.1.2. IT security

The IT security sector is another domain that is plagued by false positives. According to a study by the Ponemon Institute, companies spend on average 21,000 hours per year on false positive alerts, which equals about 1.3 million dollars [10]. Not only does this mean that security teams are focused on harmless notifications, but they are also distracted from actual threats that can lead to a breach. The same study stated that correspondents observed an increase in both the volume and severity of malware, respectively with 47% and 14%.

An IT breach can have severe consequences, both financially and regarding public trust. For example, nearly 70 million dollars worth of Bitcoins was stolen recently after a cyber attack on cryptocurrency website NiceHash [37]. Gemalto stated that 2017 was a record year for stolen data, with over 2.6 billion breached records [70]. Not only do these breaches have financial consequences, but they can also result in a decrease in public trust. Information privacy breaches expose companies to the risks of a decreased consumer trust and devaluation of public market value [111].

4.1.3. Informational privacy

Informational privacy issues form another problem that is caused by a large number of false positives [124, 125]. This becomes clear in predictive policing. When a large number of people are falsely marked as suspect, the personal data of these innocent people will be screened. This is arguably not in line with the proportionality principle, which states that there must be a legitimate aim for a measure [51]. The legitimacy of going through lots of false positives is questionable.

4.1.4. Subsidiary conclusion

Based on the examples in the previous sections, the consequences of errors in cyber-systems can be listed to include:

- Financial loss
- Privacy issues
- Public trust issues

These are not mutually exclusive since it can be argued that privacy issues and public trust issues lead to financial loss. The last two listed risks are mentioned separately since they can have social consequences as well, such as legal issues.

4.2. Type I & II errors in the physical domain

4.2.1. Climate science and assessment

Anthropogenic climate change is a challenging topic in both politics and science. The Intergovernmental Panel on Climate Change (IPCC) stated that "responding to climate change involves an iterative risk management process that includes both mitigation and adaptation" [89, p. 64]. Risk management in climate science starts with estimating causal factors and future projections of variables such as temperature and sea level rise. Type I and type II errors play an essential role in determining which of these changes are caused by humans [174]. Type I and Type II errors are also important to the mean, lower and upper range of projected impacts of future climate change [154]. In this context, a type I error is analogous to an overestimation of the effect, while a type II error corresponds to an underestimation (in terms of magnitude) [39, 154].

Both type I and type II errors are potentially bad for climate science. [135] states that both errors can lead to a discipline or assessment being seen as irrelevant, misinterpreted, or can lead to severe damage to society and human well-being. Type I and II errors in risk assessments can impact consequences of policy-making as well. A type I error can lead to a lack of necessary measures, which causes governments to not adequately be prepared for the effects of climate change. An example of this situation is underestimating the water level rise, which leads to a lack of investments in dykes, which results in an increased risk of floods. On the other hand, a type II error can result in a focus on the wrong area. To stick with the same example, one can think of investments in dykes that are too high because the water level rise was overestimated.

[39] argues that climate science favours the "side of least drama", thus favours decreasing type I errors over the expense of increasing type II errors. In other words: climate science underestimates rather than overestimates climate effects. [15] gives two exemplary cases where decreasing the risk of type I errors led to an increase in the risk of type II errors.

The first example is the IPCC fourth assessment report where the IPCC estimated the sea level rise [89]. The influence of melting land ice on the total sea level rise was estimated to be huge. This effect was even accelerating, but because of the complexity of this effect, the authors were unable to make accurate quantitative projections [169]. Therefore, they decided to remove the acceleration out of their calculations, only keeping the effect at a constant increase. Later, it appeared that the IPCC had indeed been too conservative in their estimations [39]. This example shows avoiding type I errors can lead to type II errors. Furthermore, [15] showed that only 31% of the media that covered the report actually mentioned that the authors left the dynamics of land ice meltage out, which can have consequences for risk management and policy making.

The second example, also from the IPCC Fourth Assessment Report, is an estimation of glacial melt. The IPCC concluded that glacial melt is possible by 2035 [89]. Although glaciers are melting at a rapid pace, 2035 is likely to be an overestimation [32], thus a possible type I error. As opposed to the first example, there was wide media and academic coverage of this potential error, even forcing IPCC to make a public statement regarding the overestimation [15].

These examples show the complexity of the trade-off between type I and type II errors in climate science. Type II errors seem to be favoured over type I errors due to the public opinion.

4.2.2. Bridge construction

An error in the design or assessment of a bridge can have severe consequences. One of the largest bridge collapses in history happened in Quebec, 1907 [143]. At one point in construction, designers were told that the bridge weighed eight million pounds more than estimated. Also, it became clear that the frame of the bridge started to bend. Unfortunately, the designers did not take these warnings seriously. Shortly after, the bridge collapsed, killing 75 people. The cause of the collapse appeared to be the construction frame, which was not able to carry the additional eight million pounds.

A recent example is the Florida International University pedestrian bridge collapse in March 2018. Six people died when the bridge came down on a public road. A few days before the collapse, cracks were spotted in the bridge surface. Bridge engineers, the construction manager, state transportation managers and a university representative discussed the issue in a joint meeting but concluded that the cracks formed no safety issue [82]. The bridge collapsed three hours after the meeting. It is not yet clear whether the cracks caused the bridge to collapse, but if they did, the conclusion of the meeting to not further address the cracks would be a type II error. A similar thing occurred at the Kolkata flyover collapse in India in 2016. Workers heard a cracking sound but did not act on it [21]. A few hours later, the bridge collapsed, killing at least 24 people [145].

These examples illustrate a clear danger of type II errors when building bridges. Warnings were not taken seriously, which caused the bridges to collapse and resulted in many casualties. A type I error in this context would be unnecessarily overdimensioning a bridge. That might have prevented the collapses, but at the expense of additional cost.

4.2.3. Railway safety

Trains are a relatively safe mode of transportation when looked at fatal collisions and derailments per train-kilometres [60]. Nonetheless, errors do occur on railroads. In 2016, two freight trains collided head-on in Texas after one of the trains missed a red stop sign [8]. The collision caused the death of three crew members, a derailment and a massive fire. A similar accident occurred with a passenger train and freight train in 2008. The train driver failed to spot a red light, resulting in a collision with a freight train which killed 25 people and injured 135 people [42].

Failing to see the red sign can be seen as a type II error, which had severe consequences in these cases. A type I error in this context would mean false signalling, letting trains stop while there is no safety threat. This would likely result in arrival delays since a train takes a considerable amount of time to come to a stop, but casualties would be less likely.

4.2.4. Industrial safety

Chemical plant safety became relevant to the world after a methyl isocyanate chemical leak in Bhopal, India, killed more than 3800 people and caused morbidity and premature deaths for a long time after [36]. One of the causes of the leak was poor monitoring. Poor monitoring can be seen as a type II error since possible deficits leading up to the disaster were not identified.

In the oil industry, errors do occur as well. During the Gulf oil spill in 2010, millions of litres of oil were spilled into the Gulf of Mexico. After the spill, it became apparent that BP, the well owner; and Transocean, the rig owner, ignored tests in the hours before the leak started [71]. The tests indicated that the safety equipment on the oil platform was faulty. Ignoring these tests is a clear type II error, leading up to an environmental disaster of immense proportions.

False alarms can occur as well. In 2000, an alarm warned the citizens of Umatilla, Oregon, that a chemical spill had occurred at the Umatilla Chemical Depot [11]. Area residents quickly left town or sought shelter. After it became apparent that it was a false alarm, government officials, emergency responders and area citizens critiqued the system. One official mentioned that citizens might not heed the sirens next time [11]. This shows an occasion and possible consequences of a type I error.

A similar situation occurred in the defence industry on Hawaii. An alert stated that there was an incoming ballistic missile threat, and advised residents to seek immediate shelter. 38 minutes later, it became apparent that there was no missile threat, rendering the warning false. The false alarm led to panic and disruption in the area. Procedures were revisited, and the responsible employee was fired [27].

4.2.5. Medical diagnostics

A well-known source of type I errors is medical diagnostics. In this context, a false positive is an occasion where a patient gets to hear that a given condition is present when it is actually absent. False positives can come with high costs for follow-up care and testing [101]. On the other side, false negatives have an impact as well. False negatives can have psychological consequences and health implications for the participant, a negative impact on public confidence in screening, and may lead to legal action being taken by the participant [138].

4.2.6. Subsidiary conclusion

Physical systems share the same risks as cyber systems, but there are two additional risks:

- Physical injury
- Environmental damage

Furthermore, the balance between type I and type II errors depends on the nature of the system. The physical systems that were examined mainly favour type I errors over type II errors since missing a threat is costly in terms of physical injury and environmental damage. Climate science forms an exception to this rule, favouring an underestimation of climate effects rather than an overestimation. Public trust is the primary motivator for this preference. It could, however, be argued that a preference for type I errors benefits climate science as well since insufficient preparation for the effects of climate change can be very costly in the long term.

4.3. Type III errors

Type I and type II errors are the classic types of error and usually form a trade-off. In addition, there is another error that can occur in both cyber and physical systems. This error, named type III error, occurs when the right answer to the wrong problem is given [98]. Although it is called type III error, it comes before type I and type II errors, and can even be argued to be more fundamental than the first two errors [128]. After all, type I and II errors can only occur after the research problem has been defined. Furthermore, how useful is it to optimise the type I and type II error trade-off when the wrong problem is being solved?

An example of a type III error in the physical domain was already implicitly given in section 4.2.1. Here, the IPCC underestimated sea level rise because they left out melting land ice. Based on the model they made, their conclusions might be right, but the model answered the wrong question. It answered how much the sea level would rise if melting land ice would be taken out of the equation. However, the question they actually wanted to answer was how much the sea level would rise if they took all factors (including melting land ice) into consideration. Thus, what started as a type III error caused an underestimation of sea level rise, i.e., a type II error.

In the cyber domain, type III errors often come in the form of overfitting (discussed in section 3.2.2). Overfitting can have different causes, such as including too many features or not enough data points to construct a model. The result is a good performance on the training dataset, but poor performance on another set. This poor performance manifests itself in type I or type II errors. To come back at the definition of type III errors: the 'wrong problem' is trying to make a model that performs well on the training set, while you are actually looking for a model that performs well on all data of interest. This model might be the 'right answer' for the specific training set but is wrong for the rest of the input space. A biased training set has the same effect [88] since it does not conform with the real world input space.

To conclude, type III errors can be seen as a cause for type I and type II errors both in the physical and cyber domain. With the rising amount of data or big data, it is tempting to include more features into a model. However, the availability of big data does not mean that everything should be included. If extra features do not sufficiently describe the variable of interest, the model is prone to overfitting and becomes overly complex. Overfitting, which was shown to classify as a type III error, can lead to type I and type II errors.

4.4. Risks of combining cyber and physical systems

Physical systems become more and more dependent on cyber-systems. When combining these two domains in cyber-physical systems, the risks of both domains are combined as well. An error in an IT-system subsequently leads to an error in its connected physical system. This means that cyber-physical systems are potentially very vulnerable since cyber-systems and physical systems both carry significant risks on their own already. Besides, it was shown in section 4.1 that IT developers in purely cyber-systems only had to worry

about financial or social losses. If their IT systems are linked to physical systems, they have to be concerned with physical loss as well.

There is also a combined risk effect. The question of responsibility becomes highly relevant in cyber-physical systems. If the software makes a bad decision, will the software developer be held responsible, the party that implemented the software into its physical system, or a human operator? This question of liability can lead to negligence of involved parties since none of them feels direct responsibility. The avoidance of responsibility can thus be an extra source of risk.

Another combined effect is dealing with type I errors. It was shown in section 4.1 that it is common in the cyber-domain to deal with an abundance of false positives. However, false positives are much more costly in the physical world, which was shown in section 4.2. This asks for a careful approach when designing IT software for physical systems.

4.5. Conclusion

The goal of this chapter was to assess the consequences of type I and type II errors in the physical and cyber-domain, and what happens when these domains are combined into the cyber-physical domain. Errors in both the cyber and physical domain can have severe consequences, but when combined, the risk seems to be even more significant than the sum of the risks of the separate domains. This makes risk-management of cyber-physical systems an imperative topic.

5

How Industries Maintain the Safety of Machine Learning Applications

Industries are coming up with their own ways of maintaining the safety of ML applications. In this chapter, it is shown that challenges arise with regard to safety when ML applications are scaled up. The first part shows a number of ML applications on a small scale. In the second part, best practices to guard safety are analysed for industries which are heavily regulated: the aviation industry, the automotive industry and the medical industry. The goal is to show safety threats when scaling up ML applications. Sections are structured as follows:

- Inherently safe design
- Safe fail
- Procedural safeguards

The chapter ends with a validation section which shows that industries are struggling with the problems that were found in this chapter and the previous chapters.

5.1. Small-scale Machine Learning applications

5.1.1. Smart thermostats

ML applications are increasingly making their way to our daily lives. In 2011, Nest Labs launched the world's first learning thermostat. This thermostat for home use learns from the user's behaviour. It tracks the user's movement, phone GPS location, and preferences to optimise indoor heating and cooling [140].

No past safety-critical situations are known for smart thermostats. This can be explained due to two reasons:

1. *An error does not have severe consequences.* Errors in learning thermostats are not likely to lead to human injury and do not impact many people.
2. *There is time to correct possible errors.* If an error occurs, there is plenty of time to correct it.

However, self-learning thermostats bring secondary risk. If the thermostat is hacked, hackers can obtain information about actual locations of owners [158]. Positional information can be used for instance to plan a house robbery when the owners are gone.

The Nest Labs thermostats have a safe fail strategy for extreme temperatures. It possesses a mechanism called "Safety Temperature" which prevents the room temperature from going above or below a certain threshold temperature. This mechanism prevents damage from extreme temperatures. The system functions independently from the central temperature control system and only operates as a safe fail.

5.1.2. Smart cleaning robots

Smart cleaning robots are another household application of ML. These robots learn their surroundings based on visual data and determine optimal trajectories to clean a room. They automatically identify when their battery is almost out of power, and know how to revert to their charging station. There is one known incident where the robot ingested the hair of a sleeping woman [118], but other than that, these robots are considered to be fairly safe. They are safe due to the same reason as the smart thermostat: an error does not have severe consequences.

Risk increases when ML is applied on a larger scale and when an error affects more people. Also, more extensive safety strategies are needed. The next section will show potential risks of applying ML in safety-critical systems on larger scale.

5.2. Large-scale safety-critical applications

5.2.1. Aviation industry

A quick literature search shows a variety of papers written about autonomous autopilots. [22] introduces an intelligent autopilot system which can land an airliner under severe weather conditions using artificial neural networks. It can also perform go-arounds in case the weather conditions are too bad to land the plane. Another paper by NASA shows a neural network that can control a manned aircraft in the presence of damage to components [177]. Or [137], which shows a comparative study for ML control of the pitch angle of earoplanes.

Despite these studies, current autopilots in commercial aviation are not as related to ML as one would think. Operational autopilots in everyday use are functioning based on Control Theory principles such as Proportional Integral Derivative controllers (PID controllers) [22]. These systems are characterised by a highly restrained functionality, only capable of executing simple tasks in non-emergency conditions.

Even the restrained functionality of these systems can lead to dangerous situations [24, 163]. In case of a situation like turbulence, the autopilot could exert undesired behaviour or simply shut off and hand control back to the pilots. An example of such a situation was the Air France flight in AF447 in 2009. The flight encountered heavy turbulence, leading to the decision for the autopilot to start a steep climb. This steep climb caused the aeroplane to stall. The autopilot handed back control to the flight crew, causing the plane to lose altitude dramatically. The flight crew did not have sufficient time to rectify the situation, leading to a crash [62, 151].

Inherently safe design

Since autopilots only perform simple tasks, the inherently safe design choice is to not use autopilot under more difficult circumstances.

Safe fail

There is no usage of ML software in current autopilots, so no safe fail mechanisms can be mentioned. Nonetheless, it should be denoted that the current mechanism of handing back control to the human pilot in case of complex situations could be a safe fail of ML software as well. For this, the challenge of transparency should be overcome. ML software can only hand back control to its human pilots when it can accurately recognise situations that it cannot handle.

Procedural safeguards

An important reason that there is no ML in autopilot applications is the lack of safety standards for intelligent aviation software [18, 29]. The certification process requires software to be completely transparent, and understandable to the people that execute the certification. [29] lists the following reasons why this can be challenging for ML software in aviation:

- *Comprehensive requirements.* Certification requires to define a comprehensive set of requirements. With dynamic systems based on ML algorithms, it is hard to specify in detail what the system will do at run-time.

- *Verifiable requirements.* Even when the requirements are defined, it remains a challenge to verify them. An important reason for this is the lack of verification methods for ML components in aviation.

- *Documented design.* Many ML systems are developed based on open source software. It has to be controlled that this software will not be updated or altered after implementation.
- *Transparent design.* The certification authorities require a transparent software design. This can be challenging for the following reasons:
  - *Deterministic behaviour.* Current certification processes expect software to be deterministic, while ML software has a non-deterministic nature; newly acquired experiences change the behaviour of the algorithm.
  - *Conventional design artefacts.* ML software contains many unconventional artefacts such as data structures, computing models or programming languages that are unknown to the certification authorities. This makes it hard for authorities to certify ML models.
  - *Complexity.* ML models can be complex, possibly containing thousands or millions of variables.
  - *No unintended functionality.* It is hard to demonstrate that the software will not show unintended functionality.

5.2.2. Automotive industry

Autonomous cars have the potential to drastically decrease the risk of driving mistakes in modern traffic [16]. Autonomous cars do not know physical states such as fatigue, drunkenness, or distraction. Furthermore, autonomous vehicles could outperform human drivers because of improved perception (sensors register everything around the car), better decision making (e.g. better planning of complex movement of the car), and better execution (quicker response and improved accuracy) [93]. Google's self-driving cars are already on the road in California and Texas, and Tesla allowed owners to switch to Autopilot mode from the second half of 2015 [76].

Inherently safe design

Autonomous cars aim to increase their robustness to uncertainty by collecting as much data as possible. Despite this effort, accidents like the one mentioned previously could still happen, and there is no known inherently safe design strategy like interpretable models to decrease this [195].

Safe fail

Raw weather conditions and unusual driver environments can cause autonomous vehicles to perform worse than their human counterparts [73]. Although Tesla's software is called "Autopilot", it is only capable of handling relatively simple environments like highways [55]. When it encounters a difficult situation, its safe fail mechanism is to hand back control to the driver. However, it has been shown that driver's attention decreases when Autopilot is engaged, which can lead to increased reaction times when the software hands back control over the vehicle to the driver [170].

Safety protocol

[93] lists three problems concerning the safety protocol of autonomous vehicles:

1. There is no current standard to validate the safety of autonomous vehicles before allowing them on the road.
2. There is no consensus on how safe an autonomous vehicle should be, or in other words: about the safety requirements that an autonomous vehicle should fulfil.
3. Real world testing is a quandary since testing improves safety, but also imposes a risk on the environment of the autonomous cars.

The international industry standard for testing commercial vehicle safety is the ISO26262 [136]. However, the ISO26262 was not designed for autonomous vehicles for the following reasons [99]:

- The ISO26262 relies on the driver being ultimately responsible for the safety of the car. While this is easily justifiable for simple systems like adaptive cruise control, responsibility becomes a more controversial topic when the autonomy of cars is increased.
- The standard does not account for the large amount of data that is coming from all sensors in autonomous vehicles.

- Decision rules of ML algorithms are not easy to interpret by humans. Furthermore, the ISO 26262 standard requires the decision rules to be defined upfront, while ML algorithms change their behaviour based on newly acquired training data. It should be denoted that current simple systems like adaptive cruise control rely on more traditional Control Theory based systems, which do not change their behaviour over time and thus can be validated using traditional methods [97].

It can be concluded that, despite the fact that autonomous cars are already on the road, there is no universally approved safety protocol to guide this.

5.2.3. Medical industry

ML is an emerging research area in the field of medical problems. ML has the potential to automate time-intensive tasks that are currently executed by experienced practitioners [33]. A typical application is classification, where software classifies patients suffering a particular illness based on medical imaging techniques. An algorithm to detect skin cancer based on images was already shown to perform on par with certified dermatologists [61]. A New England Journal of Medicine recently stated that it expects that such algorithms will soon replace the work of expert clinicians like radiologists and anatomical pathologists [134].

There is a range of risks involved with ML in the medical industry. An erroneous classification algorithm can falsely conclude that a disease is occurrent, causing the incurrence of high ineffective treatment costs and possibly emotional suffering for the patient. A recent study showed that two-thirds of the insulin-calculation apps advised incorrect insulin dosages [86] The cost of a false negative can have severe consequences for the patient as well, since not detecting a disease while there is one, is something a hospital wants to avoid at all times.

Inherently safe design

Inherently safe design in medical diagnostics boils down to interpretability. When an algorithm is used to determine if a patient has a particular disease, doctors will want to know on what rules the algorithm bases its decision. However, transparency is one of the drawbacks of ML algorithms, causing the name 'black box medicine' for this domain [141]. Up to this point, there is no known trend in developing transparent algorithms for medical use [142].

Safe fail

Safe fail mechanisms in black box medicine would lead back to the certainty measure of the algorithm. If an algorithm can show that it is uncertain about a prediction or diagnosis, a doctor can decide to use different diagnostic methods. Unfortunately, no literature concerning the prediction certainty was found in this domain.

Safety protocol

The ISO 13485 is a commonly used international standard for medical devices and software and is often seen as the first step in complying with European regulatory requirements [119]. This standard is based on the "build and freeze model" whereby medical devices are produced, tested and used in a defined and unchanged way [126]. A small design change requires a new certification process. This is troublesome for ML applications, which change their behaviour after acquiring new data [96]. Furthermore, the lack of transparency is seen as a problem since the motivation of findings is highly valued in healthcare [176]. Only recently, for the first time, the U.S. Food and Drug Administration (FDA) approved an ML application to be used in a clinical setting [112]. The application was Arterys, a medical imaging platform that can be used to help doctors diagnose heart problems.

5.2.4. Nuclear power plant control

The nuclear industry adopted a relatively high level of automation for reasons such as regulatory requirements, reduction of human errors, efficiency improvement or protection for human operators against radiation exposure [102]. There is an emphasis on automated safety functions. These functions include stopping the reactor, preventing radiation exposure, or removing excess heat. Current regulation prescribes that some safety functions need to be automated, and automatic systems are known to be more reliable, more accurate and faster than their human counterparts.

A new trend, known as small modular reactors (SMRs), even integrates a higher level of automation in its design [107]. SMRs are planned to be constructed in low population density areas that are difficult to access (such as islands, deserts, or polar regions), thus requiring more automation or even autonomous operation.

However, despite the fact that there is a vast amount of literature about ML solutions in nuclear power plants (NPP), there have not been any applications in operating nuclear power plants so far [102, 191]. Just like in modern aviation, automation consists of various independent subsystems with their own task, mainly based on traditional control theory principles. Autonomous control is intended to be used in case of unusual events like reactor failure when the situation is too complicated to assess by humans. However, the literature shows a gap between current nuclear plant control and the development of ML solutions for the nuclear power plant industry [191].

Inherently safe design
Robustness in current systems is achieved by control the use of control systems. There is no known example of inherent design since ML systems have not been implemented.

Safe fail
Safe fail mechanisms for ML applications in nuclear power plant control form one of the current research gaps [191].

Safety protocol
The IAEA software safety standard sets guidelines for the validation of software used in nuclear power plants but does not mention ML or adaptive software [78]. However, it mentions a number of things that can be difficult for ML software:

1. *Avoidance of complexity.* The safety standard mentions that complexity should be avoided, so it is easier to understand the software. The black box nature of ML algorithms makes it difficult to fulfil this requirement.

2. *Understandable and modifiable structure.* The IAEA requires the software to be understandable and modifiable, which is again one of the weak points of ML.

3. *Predictability.* Software in nuclear power plants should be deterministic. This is contradictory to the nature of machine-learning software.

4. *Verifiability.* The IAEA requires that it is possible to demonstrate that all software requirements have been met. This is not easy due to the lack of transparency and the non-deterministic nature of ML algorithms.

5.3. Validation

The findings in the last three chapters are all based on theoretical research. A short check will be executed in this section to validate whether the safety issues of ML applications addressed previously are actually relevant in practice. This part merely serves as a short inventarisation of practical concerns, not as a full validation. The research objective is formulated as follows:

> Do the theoretical findings of the past three chapters -with regard to the safety of ML applications- show resemblance with concerns found in practice?

This research question will be answered by means of two interviews. The interviews are semi-structured to give the interviewees room to talk freely about problems they encountered regarding ML applications. Topics of interest along the lines of the past three chapters are type I/II errors, testing, safety strategies and regulation. The involved sectors are automotive and chemical industry since these sectors were already discussed in theory in the previous parts. When these two interviews already show resemblance to the findings in the last three chapters, they are sufficient to answer the research question. More interviews would be needed to generalise findings across whole sectors, but since that is not the goal of this section, two interviews are sufficient.

5.3.1. Autonomous cars

Daniël Heikoop is involved in the Meaningful Human Control over Automated Driving Systems (MHC-ADS) project. The project is aimed at "guiding a responsible transition within increasingly complex and automated driving systems" [4]. Heikoop highlighted the next themes during the interview:

- *Situational awareness.* Heikoop stresses that modern autonomous cars can take over relatively simple tasks, but when situations get more complicated, control is handed back to the driver. However, due to the fact that drivers do not have to execute basic tasks, they are not part of the car control loop anymore, and their situational awareness decreases. Consequentially, when a driver suddenly needs to react to a complex situation, the chance of an error or accident increases. Heikoop questions if it would not be better to skip this phase of semi-autonomous driving.
- *The number of relevant factors.* The automotive industry is often inspired by technologies in aviation, but it is not easy to translate new technologies from aviation to the automotive industry. While an earoplane only deals with take-off, landing, and a predetermined trajectory, a car has to deal with cities, corners, pedestrians, cyclists, traffic, and other factors. The fact that the environment of a car is so much more complex makes it difficult to translate aviation technologies to the automotive industry.
- *Liability.* Heikoop states that the issue of liability is also part of autonomous driving. Who is responsible in case of an accident? Is it the software engineer who coded the control software? Is it the car producer? Is it the driver? Is it the car itself? Or is it the government that allowed the car to drive on public roads?
- *Social impact in case of errors.* Heikoop states that nothing is allowed to go wrong in autonomous driving since the impact in the media is huge. Moreover, based on the media, states can all of a sudden decide to prohibit the testing of autonomous vehicles after an incident. Heikoop finds it a pity that the media impact is so huge in case of incidents, while the technique actually seems to be safer than manual cars; fewer accidents per kilometre driven.

5.3.2. Chemical industry

Dick Nijen Twilhaar is a board member of Innovating Safety, a foundation about knowledge and technology that has the purpose of improving safety, health and environmental aspects of professional work activities. He also has an extensive record in the safety sector of Shell. Twilhaar noted the following with respect to chemical industry safety:

- *The trade-off between false positives and false negatives.* Twilhaar emphasises the importance of the balance between the detection side on the one hand, and false alarms on the other. False alarms cannot be eliminated without decreasing the detection-rate, so it is always a trade-off. Investing more money to place extra sensors will usually increase false-alarms and thus does not necessarily lead to a safer system. Furthermore, according to Twilhaar, more false alarms lead to normalisation. If false alarms keep going off, operators will get used to it, thus decreasing the chance of an accurate response in case of a true alarm.
- *Software always contains errors.* As a rule of thumb, Twilhaar states that there is always an error in every 1000 lines of software code. Furthermore, every new system has a chance of failure. Due to the high number of input parameters, it can not be proven that control software is entirely safe. As an example, he states that mobile phone operating software, used by millions of people, still contains errors.

  For these reasons, Twilhaar thinks that the focus should be on fighting the consequences of an incident rather than attempting to make the system itself 100% safe. He states that striving to achieve 100% safety is a high risk.
- *The vulnerability of digital control systems.* Twilhaar highlights the vulnerability of digital control systems to cyber threats. As an example, he mentions Stuxnet, a malicious computer worm that targeted Iranian control systems. He states that control systems that have a connection to the internet or even USB always entails a major threat. He does give two conditions where digital control systems can be used:

  1. A digital control system can be used when important system parameters are guarded by independent subsystems. E.g., an independent subsystem measures if the pressure inside a chemical reaction tank does not surpass a certain threshold. If it does, the system will be shut down without the intervention of the digital controller.
  2. The second circumstance where a digital control system can be used is when it outperforms its human equivalent. Humans are fallible. If these systems improve the safety situation, it is a good choice to use them. However, Twilhaar states that he does not take social acceptance into consideration for this advice, since it can complicate matters.

5.3.3. Interviews conclusion

These interviews confirm the following findings from this chapter and the last chapters:

- The findings regarding situational awareness (section 4.2.1).
- The complexity of current cyber-physical systems (chapter 4).
- The issue of liability (section 3.3).
- The consideration of social impact in case of errors (chapter 4)
- The trade-off between type I & type II errors (chapter 4).
- The fact that a system will always contain uncertainty (chapter 2).
- The vulnerability of digital control systems (chapter 4).
- The usage of safe fail mechanisms (chapter 2, 4).

Therefore, the answer to the research question "Do the theoretical findings of the past three chapters -with regard to the safety of ML applications- show resemblance with concerns found in practice?" can be answered with 'yes'.

5.4. Conclusion

This chapter looked at strategies currently in place for increasing safety of ML applications in heavily regulated industries. It was shown that safety issues increase when ML is applied at a larger scale. All large-scale safety-critical industries show problems in this area. In aviation, medicine and the nuclear industry, it seems that the epistemic uncertainty of ML software withholds implementation into practice, although theoretical research is present. In the automotive industry, (partly) autonomous cars are currently being employed on the road, but this is happening without convincing safety strategies or regulation. Two industry interviews confirm these findings. Without appropriate regulation, the safety-critical industry carries a big responsibility in making their products safe.

6

A Review of Safety Standards in Light of ML Applications

The ISO 31000 and IEC 61580 are industry standards that systematically address safety. They are designed to guide the construction of risk analyses, to come up with safety measures, and to implement and maintain these measures in an organisation. Although both standards are widely used in industrial settings, they are not designed to specifically include ML applications. In this chapter, it will be analysed whether both standards are fit for ML applications, and to what degree. Relevant pieces will be cited and evaluated in light of the findings in chapter 3. Consequently, some suggestions for improvement will be made. The information obtained in this chapter will help to formulate how to improve these current risk standards to incorporate the risks of ML applications.

6.1. ISO 31000:2018

The ISO 31000 was published in 2009 and is meant as a general standard for the implementation of risk management. Recently, the original document was revised. The revised ISO31000:2018 standard was published last February. As opposed to many other ISO standards, the ISO 31000 is not focused on a specific industry. It instead provides structure and best practices to all operations dealing with risk management. In this section, it will be analysed how the standard can be improved by taking the findings of the previous chapters into account with regard to risks of ML applications. The scope of this chapter will be limited to the process part of the safety standard (see figure 6.1, since this part contains the actual risk assessment. Improvements are only suggested from an ML applications perspective. Since the ISO 31000 is meant as a universal standard, suggestions in this chapter might not work for applications other than ML applications.

6.1.1. Definition of risk

In the ISO 31000 standard, risk is defined as the

> **Effect** of **uncertainty** on **objectives** [67, p. 1].

To see how this definition relates to the definition of risk in ML applications given in chapter 3, the ISO 31000 definition needs to be broken down to understand its meaning. The first term, 'effect', is defined as "a deviation from the expected. It can be positive, negative or both, and can address, create or result in opportunities and threats" [67, p. 1]. Freely interpreted, an effect is an 'unexpected change/result'. If this interpretation is substituted into the definition, it says:

> **Unexpected result** of **uncertainty** on **objectives**.

This seems a little double since 'unexpected' and 'uncertainty' are both mentioned. The meaning of 'uncertainty' is not mentioned in the ISO 31000:2018 standard, but the document refers to the ISO website for terminology. Here, uncertainty is defined as "the state, even partial, of deficiency of information related to, understanding or knowledge of, an event, its consequence, or likelihood" [6]. Filling this into the definition results in:

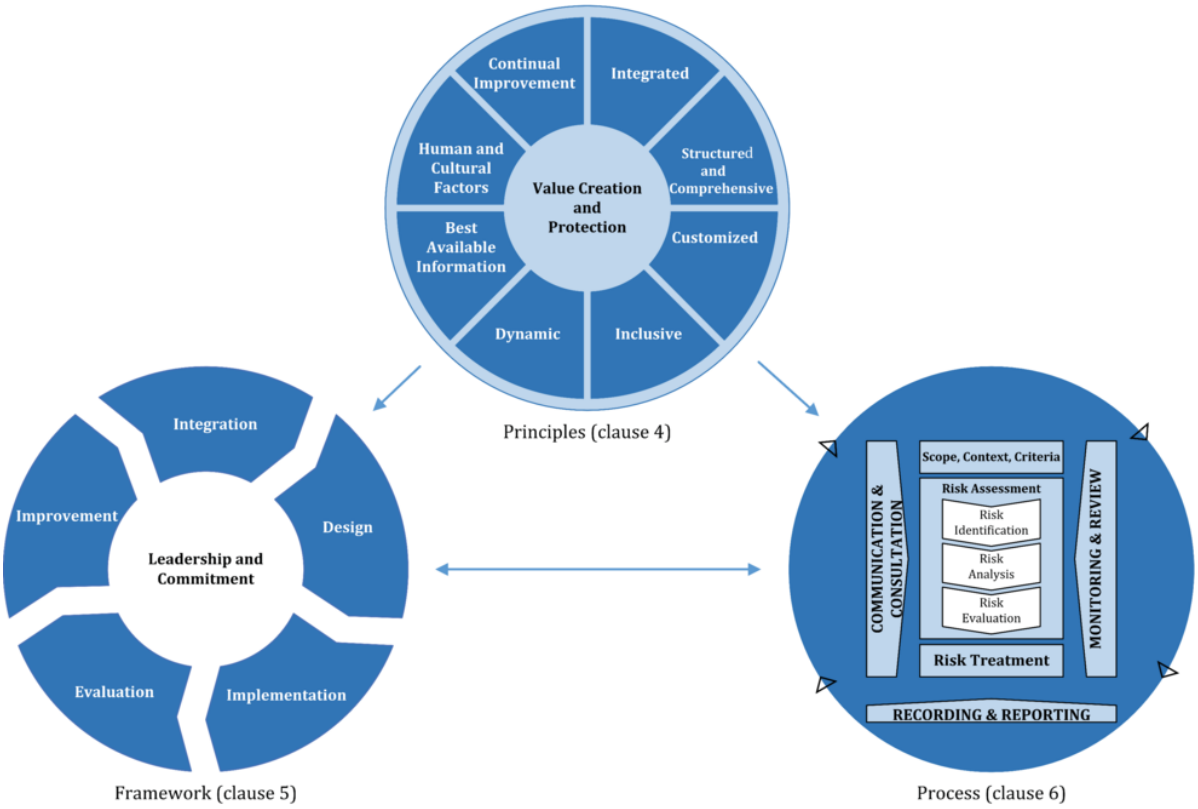


Figure 6.1: The three parts of the ISO 31000 safety standard: principles, framework, and process. Image reprinted from [67].

> **Unexpected result** of **the state of deficiency of information related to an event, its consequence, or likelihood** on **objectives**.

Although the wording seems a bit peculiar, this definition does capture the essence of epistemic uncertainty in ML applications, as was defined in chapter 3. Epistemic uncertainty was defined here as "[uncertainty] which results from our incomplete knowledge and could in principle be reduced, though this may be impractical, not possible in the framework of available time and resources, or many similar reasons" [114, p. 107]. The essential part is the first part where it is stated that epistemic uncertainty results from incomplete knowledge. This is consistent with the ISO definition, which mentions results stemming from a deficiency of information.

Furthermore, the conventional definition of risk is also mentioned by the ISO 31000 standard as a note to the definition of risk given in the previous paragraph. It states:

> Risk is usually expressed in terms of risk sources, potential events, their consequences and their likelihood [67, p. 1].

This definition of risk is more in line with the definition of risk given in chapter 3, which was the product of the severity of a harmful event, and its probability of happening. It is not clear from the ISO standard how this first and second definition relate.

Suggested improvements

Two improvements are suggested. The first one is a general suggestion and is about the phrasing of the ISO definition of risk. It can be argued that "unexpected result of uncertainty" is a tautology since a deficiency of information will not lead to expected results. Therefore, a more uniform definition is suggested, such as "the result of uncertainty".

The second suggestion is a clarification of the relation between the conventional definition of risk, and the definition as "the effect of uncertainty on objectives". It is not clear from the ISO standard how these relate. The main risk definition only seems to address epistemic uncertainty, so a link to the conventional definition of risk is needed, since both are important for safety. As a reminder, safety was defined in chapter 3 as the minimisation of both risk and epistemic uncertainty, not just epistemic uncertainty.

6.1.2. Scope, context, criteria

In this section of the ISO 31000, it is described how to design boundary conditions for the risk assessment. The purpose of following these steps is to customise the risk management process. Setting up the boundary conditions is done in three steps:

1. Defining the scope.
2. External and internal context.
3. Defining risk criteria.

These steps are context-specific and mainly involve the company goals and risk criteria. The ISO 31000 guide gives a list of requirements to be considered when setting risk criteria. Some of these requirements are listed here, with additional commentary for ML applications. What should be considered for establishing risk criteria, according to the ISO 31000 [67, p. 11]:

- *the nature and type of uncertainties that can affect outcomes and objectives (both tangible and intangible).* In short, this considers the type of uncertainty, which is generally divided into aleatory and epistemic uncertainty. It is not clear how this distinction should be used to set risk criteria.
- *how consequences (both positive and negative) and likelihood will be defined and measured.* Due to epistemic uncertainty, it is hard to measure likelihood in ML applications. There is no advice on how to measure likelihood in uncertain situations.
- *time-related factors.* This is a crucial point for ML applications due to their non-deterministic nature over time. However, how this should be considered is left open at this point in the ISO guide.
- *how the level of risk is to be determined.* If one would use the conventional definition of risk, the level of risk would be the product of the likelihood and consequence as described in the second item. If however the risk is defined as the "effect of uncertainty on objectives", which is the leading ISO definition, it will pose a problem. Epistemic uncertainty can not be determined (else it would not be epistemic uncertainty), so for ML applications this is a confusing item.

Suggested improvements
The considerations for setting risk criteria that were mentioned are a bit confusing when applied to ML applications. The suggested approach is to leave these considerations out when setting risk criteria, and take them into account when setting up the risk assessment itself. In the risk assessment, these considerations can be addressed more specifically.

6.1.3. Risk assessment

Risk identification
The purpose of this part is to find all the risks that might influence the objectives of the risk assessment. The following factors that should be considered according to the ISO 31000, form a potential problem for ML applications:

- *tangible and intangible sources of risk.* In assessing the risks of ML applications, the source of risk can be interpreted in different ways. One could argue that the source of epistemic uncertainty is the training dataset, which does not cover the whole input space. However, a theoretical training set that covers the whole input space still leaves aleatory uncertainty. It can also be argued that the source of risk is the cyber-physical system as a whole. This leaves a more open approach.
- *causes and events.* Identifying events that can occur when the system makes wrong predictions (e.g. type I and II errors) is a possible approach for setting up a list of risks for harmful events.
- *limitations of knowledge.* The limitation or lack of knowledge is the source of risk in the definition given by the ISO guide, but it does not state how to address this.

Risk analysis
Risk analysis is concerned with "a detailed consideration of uncertainties, risk sources, consequences, likelihood, events, scenarios, controls and their effectiveness" [67, p. 12].

6.1.4. Risk treatment

Selection of risk treatment options
Risk treatment options for ML applications were given in chapter 3. The ISO 31000 guide gives its own options but does mention that appropriate options are dependent on the circumstances and might be different from the ones that it mentions. The guide lists a couple of options. They are listed here and grouped into the risk strategy categories that were defined earlier:

- Inherently safe design
  - Avoid the risk by not starting the activity that is causing it.
  - Removing the risk source.
- Safety margins
  - Changing the likelihood.
- Fail-safe
  - Changing the consequences.
  - Sharing the risk (e.g. through contracts, buying insurance).
- Procedural safeguards
  - Taking or increasing the risk in order to pursue an opportunity.
  - Retaining the risk by informed decision.

Suggested improvements
Although not elaborate, all strategies are represented. Therefore, no improvements are suggested for this part.

6.2. IEC61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems

The IEC 61508 is an international safety standard that was issued by the International Electrotechnical Commission. Comparable to the ISO 31000, its intention is to serve as a basic standard for a wide range of industries, as long as the industry incorporates electrical/mechanical/electronic/programmable electronic devices. Many industry-specific standards are derived from the IEC 61508:

- ISO 26262 - Automotive software
- IEC 62279 - Rail software
- IEC 61511 - Process industry
- IEC 62061 - Machinery

Figure 6.2 shows the steps of the IEC 61508. Step 1-5 form the full risk analysis and will be taken into account in this section.

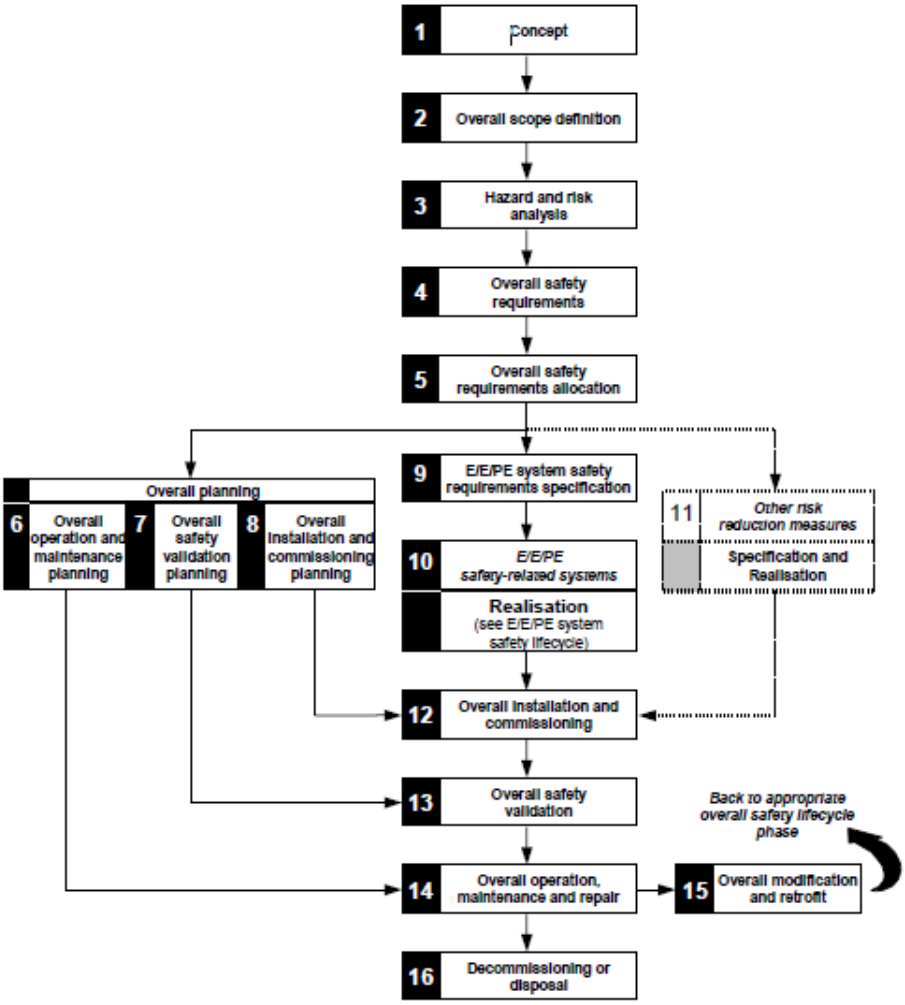


Figure 6.2: The IEC 61508 safety life cycle. Image reprinted from [12].

6.2.1. Definition of safety and risk

Part 4 of the IEC 61508 (called 'definitions and abbreviations') defines safety as follows:

> freedom from unacceptable risk.

In this same guide, risk is defined as:

> combination of the **probability of occurrence of harm** and the **severity of that harm**.

Subsequently, harm is defined as:

> physical injury or damage to the health of people either directly, or indirectly as a result of damage to property or to the environment.

Clearly, the definition of risk is the conventional one. Its meaning is obvious, but it does not contain any mention of uncertainty. The rest of part 4 of the IEC 61508 also does not mention uncertainty. Furthermore, 'probability' is not explained any further.

Suggested improvements
Since the notion of uncertainty is missing throughout, the suggested improvement is to include uncertainty in or relate this to the definition of risk.

6.2.2. Scope, context, criteria

The IEC 61508 is fairly brief and non-specific in this section. Its first step is to develop an understanding of the equipment under control (EUC). Some requirements are listed, of which the following ones might cause some trouble for ML applications:

1. *The likely sources of hazards shall be determined.* Although ML applications are meant to decrease the potential of a hazard, they are a source of hazards as well. This makes this requirement of the IEC 61508 a little ambiguous.

2. *Information about the current safety regulations (national and international) shall be obtained.* The problem with this requirement is that the regulation of ML applications is still in its infancy (as was shown in chapter 5). Therefore, decreasing the chance of a hazard will mainly have to be done by executing proper risk analysis and prevention steps.

With regard to setting the scope, the following requirement can be troublesome:

1. *The type of accident-initiating events that need to be considered (for example component failures, procedural faults, human error, dependent failure mechanisms which can cause accident sequences to occur) shall be specified.* Regarding this requirement, it should be denoted that the nature of the accident-initiating event of ML applications is not always clear. For example, a wrong prediction could be marked as 'component failure', but this might not be clear from the start since ML applications can still show a high confidence level when making wrong decisions.

Suggested improvements
The requirements given in the IEC 61508 are very general, which makes it hard to give specific suggestions to incorporate ML applications. It would be helpful to include a guideline which explains possible sources and the nature of hazards when using ML applications. This would provide insight into possible hazards.

6.2.3. Hazard and risk analysis

This part of the IEC guide notes the requirements to fulfil when setting up the hazard and risk analysis. The objectives of the requirements are to determine the hazards and hazardous events of the EUC, to determine the event sequences leading to the hazardous event, and to determine the associated risks. The next requirements raise questions:

1. *The hazards and hazardous events of the EUC and the EUC control system shall be determined under all reasonably foreseeable circumstances (including fault conditions and reasonably foreseeable misuse). This shall include all relevant human factor issues, and shall give particular attention to abnormal or infrequent modes of operation of the EUC.* The notion of 'foreseeable circumstances' is troublesome for ML applications since epistemic uncertainty plays an important role. In other words: not all circumstances can be foreseen.

2. *The likelihood of the hazardous events for the conditions specified in 7.4.2.3 shall be evaluated.* Again due to epistemic uncertainty, it is not possible to determine the likelihood of all hazardous events of ML applications. The IEC 61508 does not mention how to cope with this uncertainty.

3. *The EUC risk shall be evaluated, or estimated, for each determined hazardous event.* This is an important requirement, but it is not explained how this should be done in case of (epistemic) uncertainty.

4. *The hazard and risk analysis shall consider the following:*

   - *each determined hazardous event and the components that contribute to it;*
   - *the consequences and likelihood of the event sequences with which each hazardous event is associated.*

   If one breaks a whole system down into components, this approach works. However, software of ML applications often operates as one. An example is [31], where raw sensor data is used as input for a deep neural network that directly controls parameters like steering direction. In such systems, it is hard to break down an event sequence when a faulty steering movement is made.

Suggested improvements
The IEC 61508 assumes the EUC to be deterministic, while ML applications have a non-deterministic character. This causes some issues which became apparent in this subsection. The suggested improvement is to include uncertainty in the hazard and risk analysis requirements. Uncertainty is inherent to ML applications, so it should at least be mentioned in the requirements for setting up the risk and hazard analysis.

The second point of improvement is to include the possible non-modularity of ML applications software. What to do when software cannot be broken down into different components?

6.2.4. Overall safety requirements

The purpose of this section is to develop safety requirements to reduce or mitigate the risks that were determined during the risk analysis.

1. *The necessary risk reduction shall be determined for each determined hazardous event. The necessary risk reduction may be determined in a quantitative and/or qualitative manner.* Although this is an important requirement, it does not specify how this should be achieved. Advice on possible strategies is missing.

2. *The dangerous failure rate that can be claimed for the EUC control system shall be not lower than* 10^{-5} *dangerous failures per hour.* As in the previous examples, the IEC standard assumes deterministic behaviour, and does not explain how to deal with uncertainty.

Suggested improvements
Similar to the previous points, the notion of uncertainty should be included. Furthermore, risk reduction strategies (in particular for ML applications) would be helpful to add.

6.3. Conclusion

The goal of this chapter was to analyse how well the ISO 31000 and IEC 31508 standards are suitable for ML applications. The following conclusions can be drawn concerning suitability:

- Both standards did not combine risk and uncertainty in their definition of safety. ISO's definition of risk mainly focuses on epistemic uncertainty, whereas the IEC standard focuses on likelihood-based risk. Both are important for ML applications, but neither of the two standards combines risk and uncertainty into one definition.

- The source of risks and hazards involving ML applications is different from conventional sources. It is not the malfunctioning of a component, but complex behaviour and man-machine interactions that can inhibit safety issues. This is not captured by the ISO 31000 and IEC 16508.

- The IEC 61508 assumes deterministic behaviour, whereas ML applications are non-deterministic in character since they are trained from incomplete training sets. This can pose problems when attempting to calculate quantitative figures such as likelihood, or when mapping all sorts of hazards that can occur.

- The IEC 61505 lacks guidance on how to assess risks when ML applications cannot be separated on component level.

7

Prorail Case

Prorail, the Dutch railway infrastructure manager, is currently experimenting with smart cameras to support data-driven decision making in an effort to increase national railway safety. Smart cameras are being placed along the track to identify hazardous contents of freight trains (figure 7.1). This should lead to a full administration of carriage content and location. With this information, firefighters can increase their efficiency and adequacy in case of an accident where hazardous goods leak from their carriage.

Figure 7.1: A processed frame of a passing freight train in the Prorail project, generated by a smart camera. Image reprinted from [187].

The Prorail smart system is aimed to correctly register 100% of Dutch railway freight traffic to increase safety. However, the current system is not without flaws. Errors on various system levels can propagate through the system, possibly leading to flawed, redundant or no registration at all. An exemplary error source is a malfunctioning sensor due to rain, fog, mechanical failure, or another cause. Incomplete or faulty knowledge of hazardous goods on railway tracks can have severe consequences. Therefore, this part of the thesis aims to answer the following question:

> How do classifier and design choices influence type I and type II errors in identifying faulty freight train registrations?

This case study is presented as a proof of concept to show the influence that classifier choice and design choices can have on type I and II errors and trade-offs. Essentially, choosing the wrong classifier is a type III error, so this experiment can also be seen as testing the influence of type III errors on type I and II errors.

7.1. Methodology

To answer the research question, this problem will be treated as an anomaly detection problem. The framework in figure 7.2 will be used to determine the methodology for this case study. The application domain is explained in the introduction of this chapter. In the next section, the problem characteristics will be discussed. After this section, the anomaly detection technique will be determined.

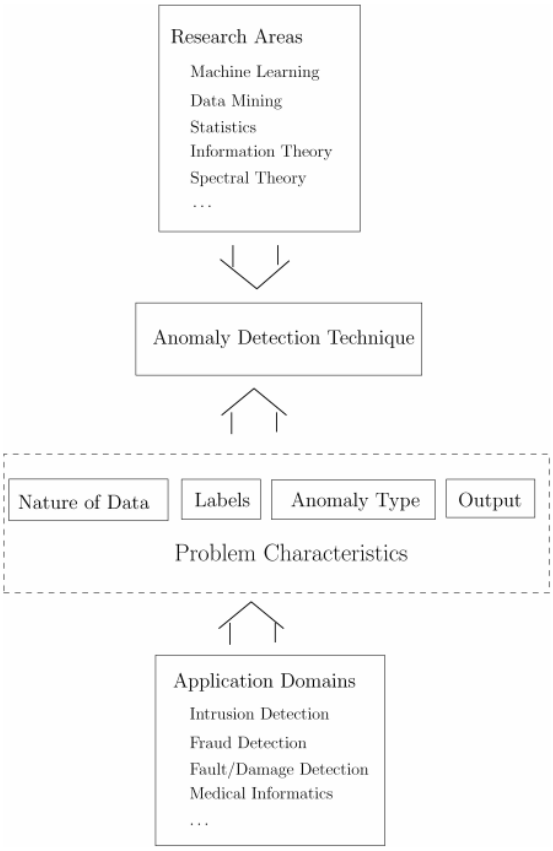


Figure 7.2: Key components associated with anomaly detection. Image reprinted from [41].

Anomaly detection or outlier detection is the process of identifying entries in data sets that are different from the norm [72]. This process is also known as outlier detection. The terms outlier detection and anomaly detection are often used interchangeably [41], but in this report, the term anomaly detection will be used. Anomaly detection is often used to enhance rule-based systems with applications in intrusion detection, fraud detection, data leakage prevention of medical applications. In these contexts, anomalies represent, for example, security breaches, credit card fraud or the occurrence of a disease. In general, anomalies can be defined by two important characteristics [72]:

1. Their characteristics deviate from the norm.

2. Their occurrence is infrequent compared to normal entries.

A visual example where anomalies occur in a two-dimensional data set is shown in figure 7.3. The data is clustered in two regions, N_1 and N_2 where the density is high. Points that lie far from these two regions are considered anomalies. The single entries o_1 and o_2 are identified as anomalies, and also the cluster O_3.

7.1.1. Problem characteristics

Different components have to be taken into account when dealing with an anomaly detection problem. [41] mapped these components in a framework, which is shown in figure 7.2. The problem characteristics depend on the application domain and will be explained in the next sections.

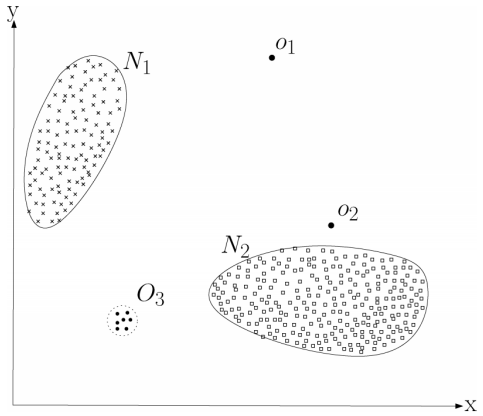


Figure 7.3: Example of anomalies in a two-dimensional data set. Image reprinted from [41].

Nature of input data

The nature of input data is essential for anomaly detection [173]. [173] breaks the nature of input data down in the following way: a dataset is a collection of data instances (such as records, events, observations). An instance can be broken down using a set of attributes (e.g., variables, features, dimensions). These attributes are of different types, such as categorical, binary or continuous. A data instance is univariate (consists of one attribute) or multivariate (consists of multiple attributes. In the multivariate case, all attributes might have identical types or might have varying types. Also, the way data points are related to each other is a way to classify data [173]. For instance, time series data should be treated differently than point data. Altogether, this defines the nature of the input data, which is important when selecting anomaly detection techniques.

The nature of the Prorail data can be determined based on the given typology. One data instance, which is the registry of a train, consists of a variety of attributes and is therefore multivariate. These attributes are not of the same type. There is categorical data such as the wagon type, weather type and the presence of a hazard. There is also continuous data such as the time of registry, length of wagons, and length of the trains. Dependencies do exist in the data, but are reasonably complex and have to be analysed in detail. For instance, the same train might pass the camera multiple times with the same composition or a slightly changed composition where only a small number of wagons have been swapped. Such dependencies will not be taken into account in this case study.

Type of anomaly

Three types of anomalies can be distinguished:

1. *Point anomalies.* Point anomalies are the simplest type of anomalies. If an individual data point can be flagged as an anomaly when compared to the other points in the dataset, it is classified as a point anomaly. For instance, o_1 and o_2 are point anomalies in figure 7.3.

2. *Contextual anomalies.* A data point is contextual anomalous when it is only anomalous in a specific context, but not otherwise. This depends on two attributes: contextual attributes and behavioural attributes. The contextual attributes determine what features comprise the context, while the behavioural attributes determine the non-contextual features of that same instance [41]. E.g., when measuring temperature, a contextual attribute can be the month of the year, and a behavioural attribute is the temperature itself.

3. *Collective anomalies.* Anomalies can be classified as collective in case a set of related data instances is anomalous, but when the separate instances within a collective anomaly are not point anomalies.

In the Prorail case, we are dealing with all three types of anomalies. For instance, a point anomaly is present when the system registers the train to have an unrealistic length. When a train is registered to be 1200 meters long, it is almost certainly an anomaly, regardless of other factors. Contextual anomalies are present when for instance a specific hazardous good does not match the wagon that carries it. Lastly, collective anomalies can be present in case the same wagon is registered in multiple different trains on an occasion where wagon swapping was not possible given the time frame.

Data labels
Labels for a data set denote whether an entry is normal or anomalous. Usually, only unlabelled data is available. Based on the availability of data labels, the following detection techniques can be selected:

- *Supervised anomaly detection.* Supervised techniques assume the availability of a training data set that contains both labels for normal data and anomalous data.
- *Semi-supervised anomaly detection.* Semi-supervised techniques assume a training set which only contains labels for normal entries.
- *Unsupervised anomaly detection.* Unsupervised methods do not require a labelled training set. These techniques make the implicit assumption that anomalies are a lot less frequently occurring than normal instances in the test data.

The data of the Prorail case is not labelled. The system registers trains, but it can not indicate whether or not these registries contain errors. Furthermore, we assume that there are no experts that have labelled any data, which makes this problem an unsupervised anomaly detection problem. It should be denoted that labels are still needed for this case study to compare classifier performance.

Output of anomaly detection algorithms
Two types of outputs of anomaly detection algorithms can be distinguished:

1. *Scores.* The detection algorithm ranks every data instance based on the degree to which it is considered an anomaly.
2. *Labels.* The algorithm simply assigns the label 'normal' or 'anomaly' to every data instance.

Scoring-based anomaly detection allows for a custom threshold for selecting anomalies. Binary labels offer less flexibility but also reduced complexity. For the sake of restraining the number of independent variables, labels are preferred over scores for the Prorail case.

Now that the requirements are specified, anomaly detection techniques can be selected.

7.1.2. Anomaly detection technique selection

[72] provides a classification of unsupervised anomaly detection algorithms for multivariate data, which is shown in figure 7.4. The selection of these algorithms is based on practical usage and appearances in scientific publications. All algorithms can be configured to output true or false labels to data points.

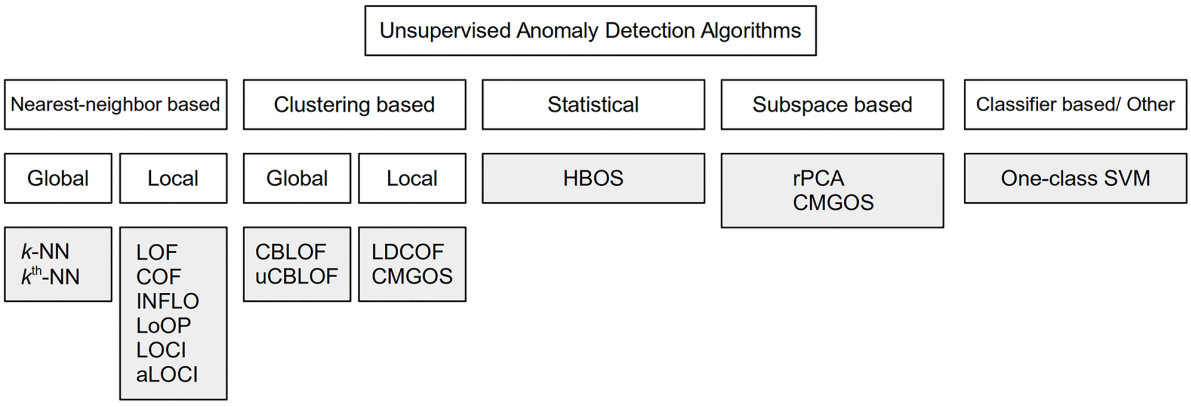


Figure 7.4: A taxonomy of unsupervised anomaly detection algorithms. Image reprinted from [72].

The first decision to be made is whether to select global or local anomaly detection algorithms. As was discussed in section 7.1.1, the data includes both types of anomalies. Therefore, one global and one local optimisation algorithm is selected.

According to [71], the LOF classifier performs best for local tasks. It is accurate, deterministic and has a high score on global detection. Its main downside is its speed, which is marked as 'average'. However, that will

not be a constraint for this research. The LOF classifier starts by finding the k nearest neighbours for every data point. Then, it calculates an anomaly score based on the distance to these neighbours. The absolute value of the anomaly score depends on normalisation, the characteristics of the dataset and the number of dimensions [72]. Therefore, different thresholds will be used to find an optimum.

The one-class Support Vector Machine (one-class SVM) classifier will be used as a second algorithm. SVMs are deemed one of the most successful and frequently used classifiers during the past years [14]. Both supervised and unsupervised variants exist, but we will use the unsupervised one. SVMs use hyperplanes to separate the class of anomalies from the normal class and are well-suited for anomaly detection [72].

7.2. Data

In figure 7.5, the different tiers of the Prorail smart camera system are schematically drawn. Data collection through sensors and initial storage takes place from tier zero to tier two. In tier three to tier six, the data is transported and analysed using a series of ML algorithms. Results of the analysis are made accessible in tier seven to tier nine. For this research, the output of the 'analysis' layer will be used since this is the information that risk managers will receive. The current system has not progressed enough for the original data to be used for this research. Therefore, a synthetic dataset was generated which emulates the system's output, including errors that can occur in tier zero to six.

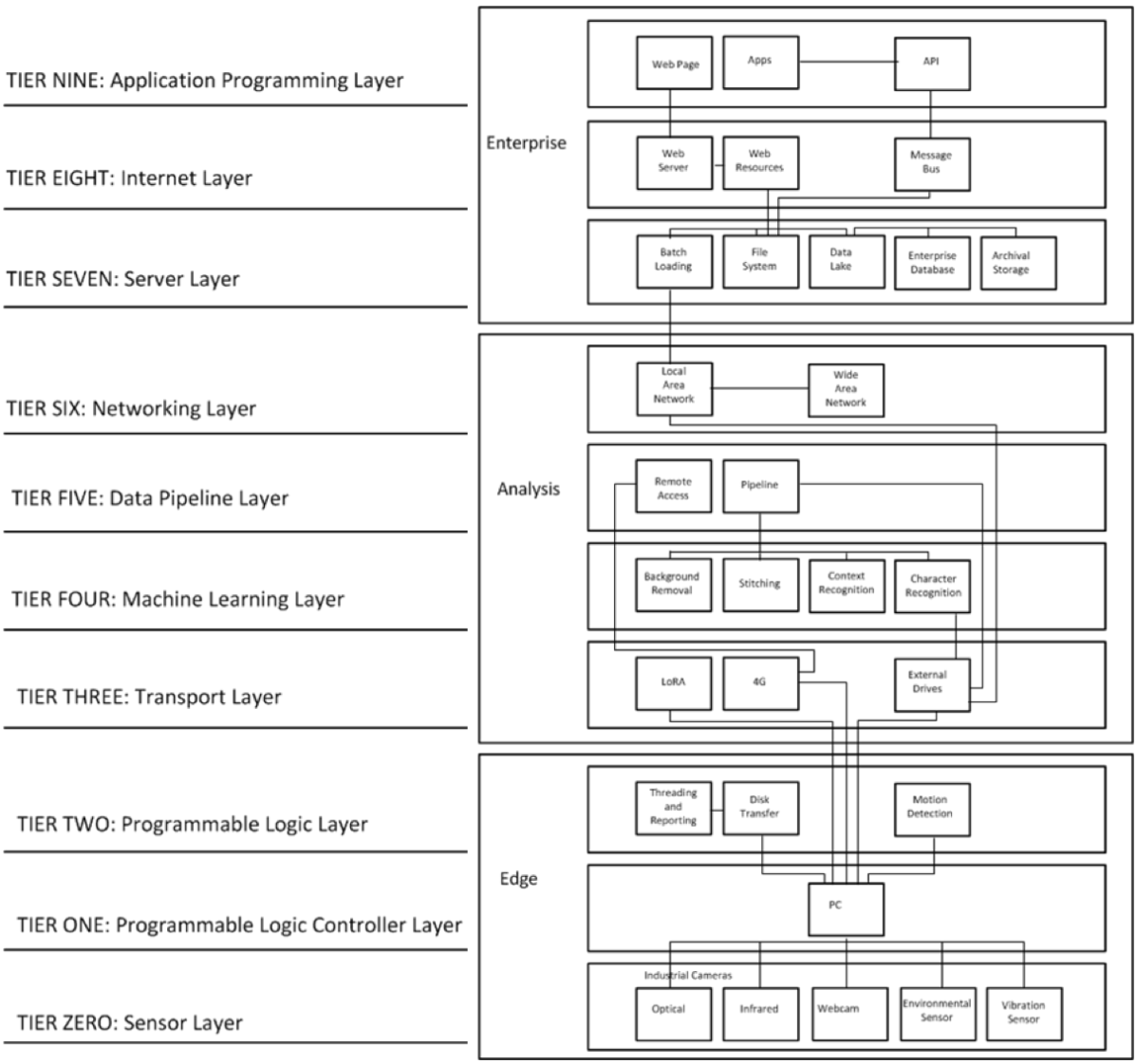


Figure 7.5: Layers of the Prorail smart system.

7.2.1. Data types

The real system saves the following data per train:

- *Operator and locomotive ID.* Both operator and locomotive are assigned with a unique ID.
- *Time.* The year, month, day, hour, minute and second that the train is first spotted.
- *External conditions.* The lighting and weather conditions are registered with the help of a small local weather station.
- *Movement.* These fields include start, velocity and acceleration.
- *Wagon information.* Every wagon behind a locomotive is registered separately. The order of the wagons is registered in a wagon ID. Wagon length, hazardous goods markers and the wagon-type are registered for every wagon.

Since this experiment is only a proof of concept, features are limited to features with regard to wagon information: wagon length, train length, number of wagons, wagon type, hazardous goods marker.

7.2.2. Data generation

This part highlights how the data for this experiment was generated. The generation process is a combination of random processes, illustrated in figure 7.6. The wagon count is generated with a uniform process of values between 1 and 20. In the next step, all wagon types are assigned to wagon types. Wagons can be of type A, B and C, and all three are equally common. In the third step, lengths are assigned to the individual wagons. Wagons of type A, B and C are respectively 10, 13 and 20 meters long. Since wagons from different manufacturers could vary a little, a Gaussian distribution with a standard deviation of 0.2 meters is applied to the wagon lengths. The total train length is the sum of all of its wagons. Finally, a hazardous goods boolean is assigned to every wagon. Wagon type A has a 0.3 chance of containing hazardous goods, whereas wagon types B and C can not contain hazardous goods.

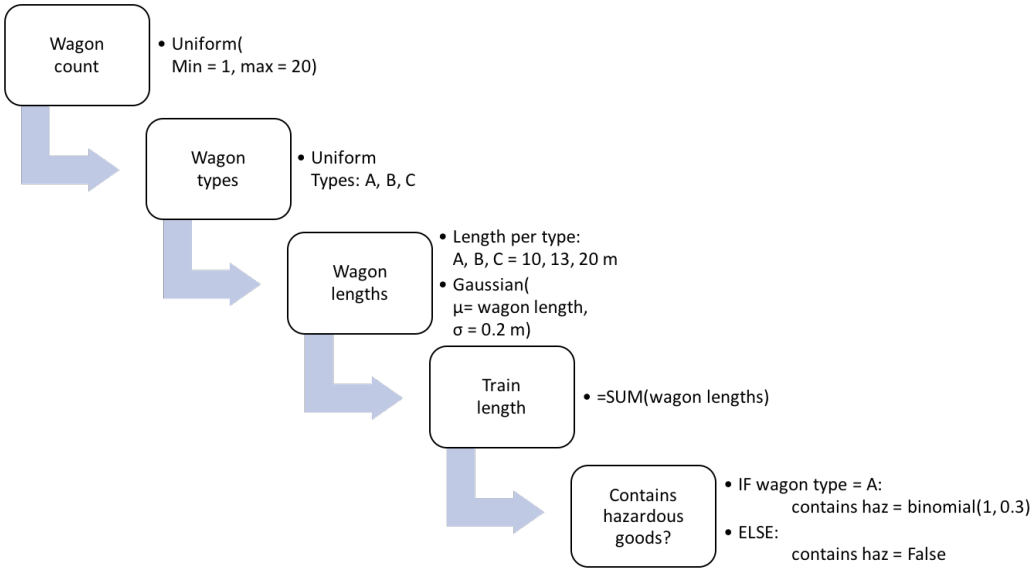


Figure 7.6: Data generation process per train.

7.2.3. Anomaly generation

In the real system, anomalies originate from two different sources:

1. *The train itself.* This can be caused by, e.g. a vague tag or a faulty tag.
2. *The registration system.* The camera or software might have issues which can cause an error in the data.

This distinction is not used for the anomaly generation process in this section since this experiment is only about detection, but one should be aware that this difference does exist.

The anomaly generation is based on a discrete Bayesian network, as was proposed in [20, 194]. Bayesian networks allow for a good structural representation of Prorail's data pipeline, which makes them suitable for usage in this research.

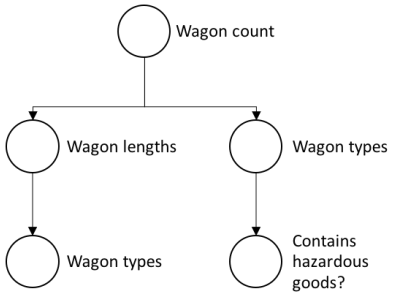


Figure 7.7: Data corruption process based on Bayesian network, where the circles represent the nodes of the model.

The structure of the corruption process is shown in figure 7.7. All nodes can have two states: 0 and 1. By default, the nodes are in state 0, which means no corruption process takes place. A Dirichlet distribution w is set as a prior for the conditional probabilities of the nodes, as is common practice in Bayesian networks [80]. This distribution is divided by 20 which results in an anomaly ratio of around 5%. Based on the prior probability w the first node can transform to state 1. If this happens, the wagon count of the train is corrupted by applying a Gaussian distribution over the ground truth with a standard deviation of 10 meters.

The rest of the nodes transform based on the following conditional logic: if the prior node is in state 1, the node has a chance w to remain in state 0. If the prior node is in state 0, it has a chance w to transform to state 1. In all other cases, the node will copy the state of the prior node.

In the case of a node being assigned to state 1, the following corruption processes will occur:

- Node 1 (wagon lengths): a Gaussian distribution will be applied with a standard deviation of 10 meters.
- Node 2 (wagon types): wagon types of a train will be shuffled.
- Node 3 (train length): a Gaussian distribution will be applied with a standard deviation of 100 meters.
- Node 4 (contains hazardous goods): the information whether a wagon contains hazardous goods will be shuffled over the concerning train.

7.3. Experiment setup

The influence of five factors on type I and type II errors will be analysed:

1. Type of classifier
2. Evaluation metric
3. Number of parameters
4. Sample size
5. Ishikawa rule-based approach

Therefore, the experiment setup consists of five subparts. All parts show results according to table 7.1.

7.3.1. Type of classifier

Based on the reasoning in section 7.1.2, SVM and LOF are used as classifiers. They are implemented in Python using the scikit-learn ML library. For the SVM classifier, ν and γ are independent variables in this research. For LOF, independent variables are the number of neighbours and the contamination rate. The optimal result is selected by running a grid search over the independent variables while taking the ROC-AUC-score as the dependent variable. The next section explains more about the evaluation metrics.

Table 7.1: Format of displaying results. Numbers will be depicted as a percentage of the total number of samples.

		Predicted	
		Yes	No
Actual	Yes	True Positives	False Negatives
	No	False Positives	True Negatives

7.3.2. Evaluation metric

Classifiers use an evaluation metric to measure their 'success'. Three commonly used metrics will be compared: the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), and the F_β score. These metrics will be shortly introduced in the next paragraphs.

ROC-AUC score

The ROC-AUC is widely used in health care since both false positives and true positives are important here [65, 79]. The score represents the chance that the classifier or test method correctly rates a randomly selected sample or patient. It measures the area under the curve when the False Positive Ratio (FPR) is plotted against the True Positive Ratio (TPR). Thus, a higher ROC-AUC score will be found when the TPR can be maximised while keeping the FPR low.

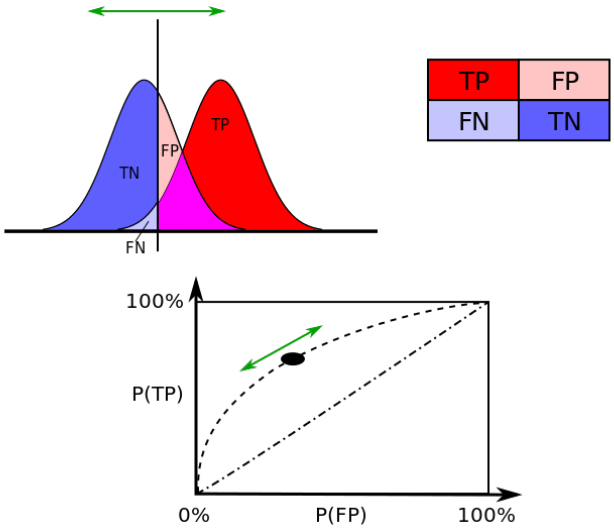


Figure 7.8: Illustration of the meaning of an ROC curve. The green arrows represent the variation of the classifier threshold. Image reprinted from [45].

ROC curves can provide insight into the FPR/TPR trade-off. Typically, an ROC curve is constructed by varying the probability score or threshold from the classifier and plotting the FPR and TPR couples for each threshold. The one-class SVM and LOF classifiers are designed in a way that they do not produce probability or confidence scores as outputs, thus varying thresholds is not possible. For this reason, a similar approach to [59] is chosen for constructing ROC-curves. The gist is that, instead of varying the threshold, the independent variables of the classifiers are varied to produce FPR and TPR couples. When executing the grid search, constructing the ROC curve comes down to plotting the FPR and TPR couples for every location on the grid. Pareto analysis can then be executed to find the Pareto frontier. For more information on the ROC-AUC score metric, please refer to [65].

F-score

The F_β score is one of the industry standards for ML. Regarding type I and type II errors, its formula is:

$$F_\beta = \frac{(1+\beta^2) * \text{true positive}}{(1+\beta^2) * \text{true positive} + \beta^2 * \text{false negative} + \text{false positive}} \tag{7.1}$$

The traditional F-score is the F_1-score (where $\beta = 1$). Two alternatives are the $F_{0.5}$ score (with more emphasis on false positives) and the F_2 score (with more emphasis on false negatives). For this experiment only, the most commonly used F_1 will be tested.

Accuracy score
The third score thas is used in this analysis is the accuracy score. Although accuracy is a commonly used metric, it is only based on the number of true positives and true negatives, or in other words: the correct predictions. The formula is:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.2}$$

All three of the metrics will be compared using the same SVM classifier and synthetic dataset. It will be evaluated how they compare with regard to false positives and false negatives.

7.3.3. Number of parameters

The small set of parameters consists of a two-dimensional dataset containing the wagon count and the total train length. To increase dimensionality, the following features will be added subsequently:

1. Wagon type
2. Hazardous goods per wagon
3. Length per wagon

Since SVM classifiers cannot handle categorical data, one-hot encoding was used to convert the wagon types to integer data.

7.3.4. Sample size

A small and large dataset will be compared. The small set consists of 5000 samples, which is equal to one year of data when nearly 14 trains pass per day. The large set is 200 times bigger and therefore contains 100,000 samples.

7.3.5. Ishikawa rule-based

Ishikawa diagrams are useful for identifying and structuring factors that contribute to the problem of interest [87, 90]. These diagrams do not include consequences or observations, which would be needed to construct a rule-based classifier. [44] proposed an extended Ishikawa diagram where an observation side is added. Figure 7.9 shows the structure of this extended Ishikawa diagram.

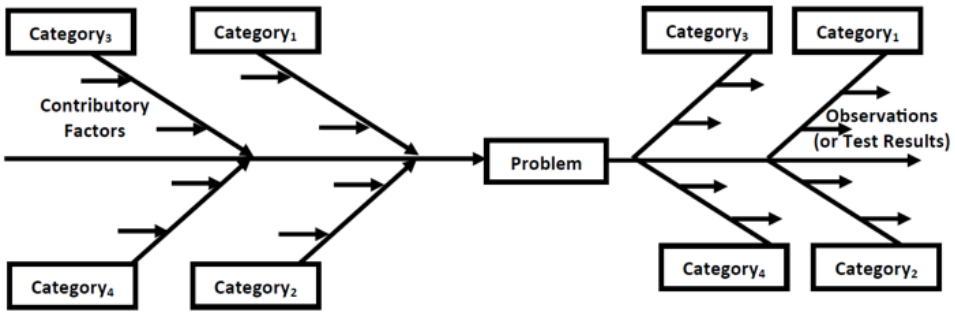


Figure 7.9: Structure of the extended Ishikawa diagram. Image reprinted from [44].

The arrows in the extended Ishikawa diagram are causal relations. The left side represents the causes of the problem of interest, split up into categories. The right side shows observations in case the problem occurs. These observations can help to reason back to the cause of the problem.

The Prorail case can be structured by first inventorising possible errors that can occur in the system. All errors and contributing factors can be entered into the left side of the extended Ishikawa diagram. With this

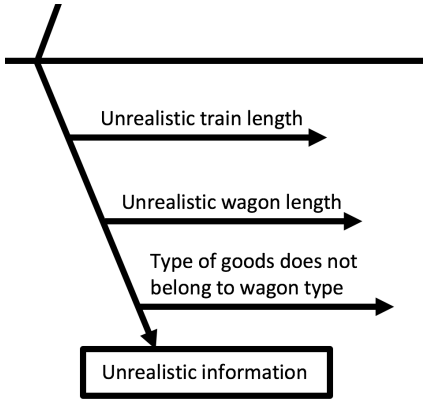


Figure 7.10: Right side branch of the extended Ishikawa diagram that is used for the analysis in this chapter.

left side, it can be reasoned how the right side of the diagram should be filled. The question to be answered is: how would an error or a combination of errors manifest itself in the system output data?

Since this research is only a proof of concept, it does not include a full Ishikawa diagram of the Prorail case. However, the structure of an Ishikawa diagram of the Prorail case might look like the one in figure 7.11. For this research we focus on the observation side, in particular, the 'incorrect information' branch which

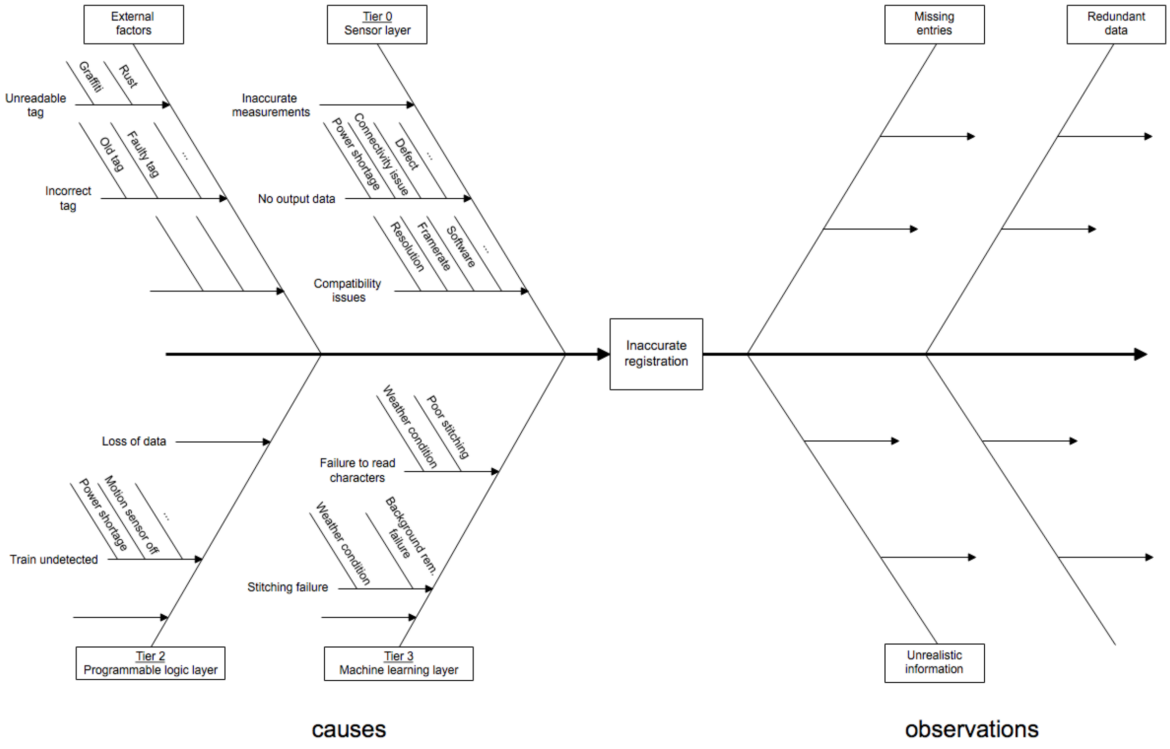


Figure 7.11: Partly filled Ishikawa diagram that serves as an example for the Prorail case.

shows data instances that are not physically possible. The rule-based classifier is based on the simplified 'incorrect information' branch shown in figure 7.10. Three rules are used:

1. Wagons of a train can not have a length of fewer than 8 meters or more than 22 meters.
2. The train length cannot be fewer than '8 meters * wagon_count' or more 'than 22 meters * wagon_count'.
3. Only wagons of type A can carry hazardous goods.

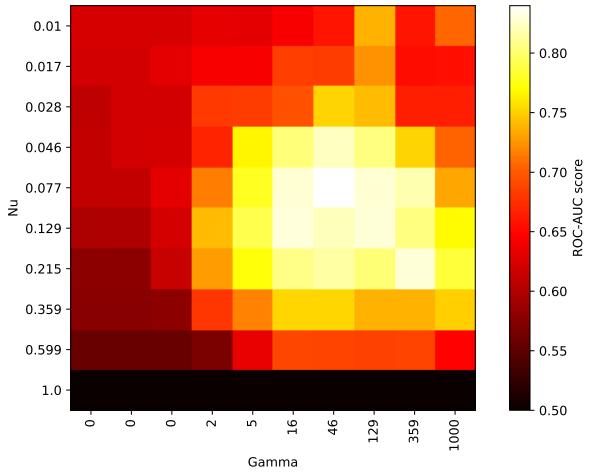
All occasions that do not fulfil one of these three rules will be flagged as anomalies.

7.4. Results

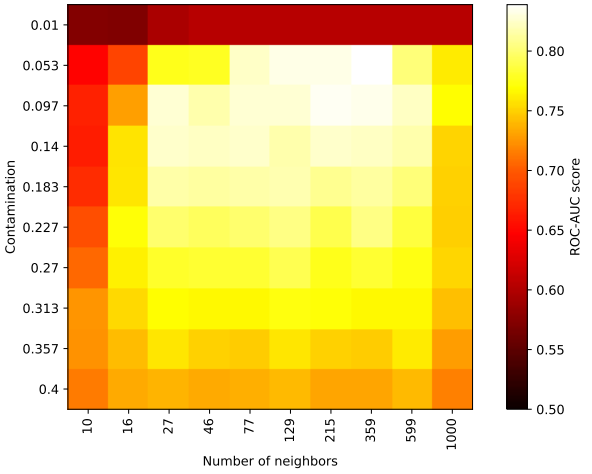
In this section, the results of the experiment will be discussed.

7.4.1. Classifier comparison

A grid search over the independent variables yielded the heat maps that are shown in figure 7.12. Both methods converged quite well. For the SVM classifier, an optimal ROC-AUC score of 0.80 was found where gamma=50 and nu=0.1. The LOF classifier performed optimally at an ROC-AUC score of 0.82 where the number of neighbours was 193 and the contamination rate was 0.066.



(a) SVM



(b) LOF

Figure 7.12: Heatmap of optimal parameters for one class SVM and LOF classifiers with sample size n=5000, where the optimal parameters have an ROC-AUC score of 0.84.

Although both classifiers performed optimally at identical ROC-AUC scores (0.84), the confusion matrix in table 7.2 shows different behaviour, specifically concerning false positives. The SVM classifier outputted 3.3% of the dataset as false positives, whereas the LOF classifier only outputted 1.9% as false positives. With a dataset of 5000 samples, this conforms to respectively 165 versus 95 false positives.

Figure 7.13 visually confirms these results. From this image, it is visible that the SVM classifier identifies more samples as outliers than the LOF classifier.

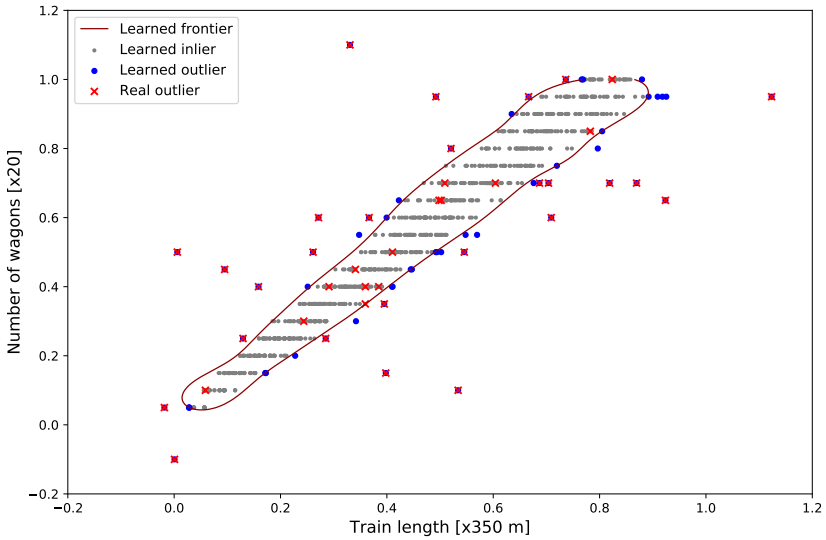
Table 7.2: Confusion matrices of comparison between the optimised SVM and LOF classifiers where the number of samples is 5000.

((a)) SVM, ROC-AUC: 0.84.

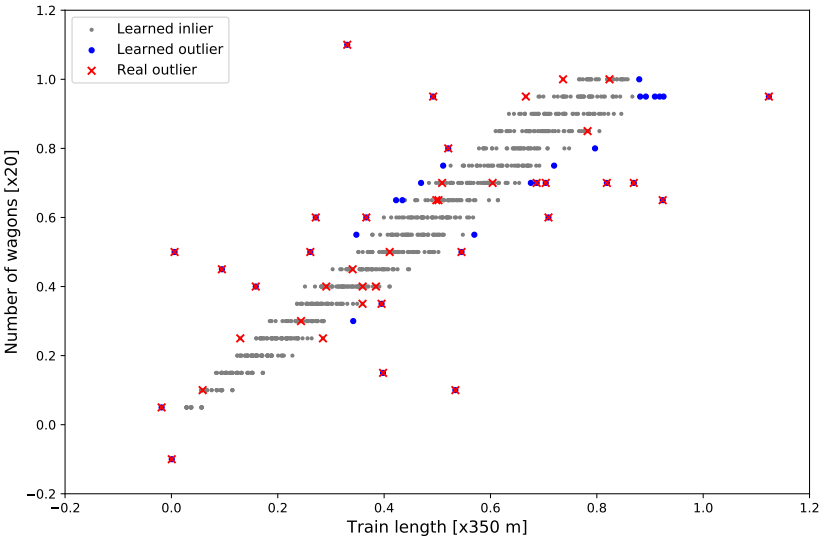
		Predicted	
		Yes	No
Actual	Yes	3.5%	1.4%
	No	3.3%	91.8%

((b)) LOF, ROC-AUC: 0.84

		Predicted	
		Yes	No
Actual	Yes	3.4%	1.5%
	No	1.9%	93.2%



(a) Scatter plot for optimal SVM parameters of gamma=50, nu = 0.1



(b) Scatter plot for optimal LOF parameters of n_neighbours=55, contamination = 0.066

Figure 7.13: Scatter plots of optimal SVM and LOF performance on the same dataset.

7.4.2. Metric comparison

The results of the metric comparison for ROC-AUC, F_1 and accuracy are depicted in table 7.3. These results show the different trade-offs that were made very well. The AUC-ROC metric causes the classifier to select the result with the most true positives while false negatives are the least out of the three metrics. However, this comes at the cost of having the highest false positive percentage. When the accuracy metric is used, the opposite trade-off is made. It leads to a low false positive percentage at the cost of the highest false negative percentage. The F_1 score falls right in between the other two metrics.

Table 7.3: Comparison of optimal results using different scoring metrics. The same dataset (5000 samples) and SVM classifier was used.

((a)) ROC-AUC score

		Predicted	
		Yes	No
Actual	Yes	3.5%	1.4%
	No	3.3%	91.8%

((b)) F_1 score

		Predicted	
		Yes	No
Actual	Yes	3.2%	1.7%
	No	0.9%	94.2%

((c)) Accuracy score

		Predicted	
		Yes	No
Actual	Yes	1.2%	3.7%
	No	0.0%	95.1%

Figure 7.14 shows the ROC plot of the metric comparison. As suspected from the confusion matrices, the optimal accuracy score is on the far left side of the Pareto optimal front. The ROC-AUC and F_1 metrics are close, where the ROC-AUC metric favours a slightly higher FPR and TPR.

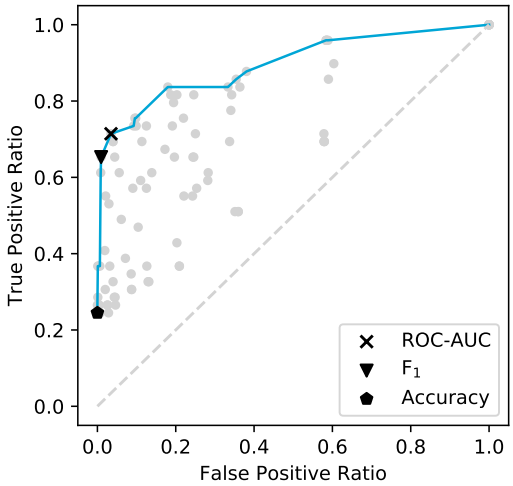


Figure 7.14: ROC plot of metric comparison. The black markers illustrate the maximum score of the corresponding metric. The blue line represents the Pareto front, and the dotted line is the case where the classifier performs randomly.

7.4.3. Number of features comparison

The initial dataset only contained the length of the train and the number of wagons. In this section, adding extra features is explored. The resulting confusion matrices can be found in table 7.4.

Table 7.4: Performance comparison of using extra features in addition to train length and number of wagons. SVM was used as a classifier for a dataset of 5000 samples.

((a)) Only train length and wagon count (AUC-ROC: 0.84).

		Predicted	
		Yes	No
Actual	Yes	3.5%	1.4%
	No	3.3%	91.8%

((b)) Added wagon type as a third feature (AUC-ROC: 0.66).

		Predicted	
		Yes	No
Actual	Yes	2.8%	2.1%
	No	24.6%	70.5%

((c)) Added hazardous goods boolean as third feature (AUC-ROC: 0.74).

		Predicted	
		Yes	No
Actual	Yes	4.1%	0.8%
	No	60.7%	34.4%

((d)) Added wagon lengths as third feature (AUC-ROC: 0.57).

		Predicted	
		Yes	No
Actual	Yes	4.7%	0.2%
	No	78.0%	17.1%

Compared to the base situation in table 7.4(a) of using two features, all three runs with an added feature show much lower performance with regard to AUC-ROC scores. This lower score is reflected in an increased false positive score and consequently a lower true negative score. When 'wagon type' was added as a third feature, false negatives dropped to 0.2%, which means that the SVM classifier found almost all anomalies. However, this came at the expense of a false positive percentage of 78.0%. For all three added feature scenarios, the following observation can be denoted: the higher the true positive percentage, the higher the false positive percentage.

7.4.4. Sample size comparison

The influence of the sample size is analysed in this section. Table 7.5 shows the confusion matrices of a sample size of 5000 versus a sample size of 200 000. The results are similar. The large dataset leads to a slightly lower true positive percentage and a slightly higher true negative percentage. With 0.84 versus 0.83, the ROC-AUC score is a little higher in favour of the small dataset. The ROC curves in figure 7.15 indeed show similar behaviour. Also, the scatter plots in figure 7.16 are nearly identical.

Table 7.5: Small versus large sample size, optimised results, SVM classifier.

((a)) Small (n=5000), ROC-AUC=0.84.

		Predicted: Yes	Predicted: No
Actual	Yes	3.5%	1.4%
	No	3.3%	91.8%

((b)) Large (n=200,000), 2 features, ROC-AUC=0.83.

		Predicted: Yes	Predicted: No
Actual	Yes	3.5%	1.4%
	No	4.3%	90.8%

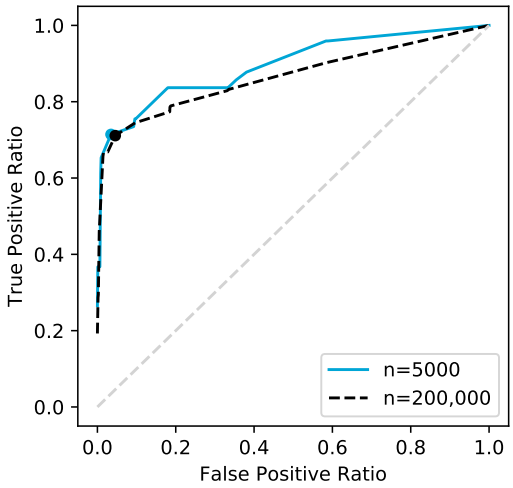


Figure 7.15: ROC curves of sample size comparison with two features and the SVM classifier.

In table 7.6 a similar comparison is shown, but with three features instead of two. The third feature here is the lengths of all the separate wagons. In this situation, adding more features shows a vast improvement since the AUC-ROC score rises from 0.57 to a score of 0.87.

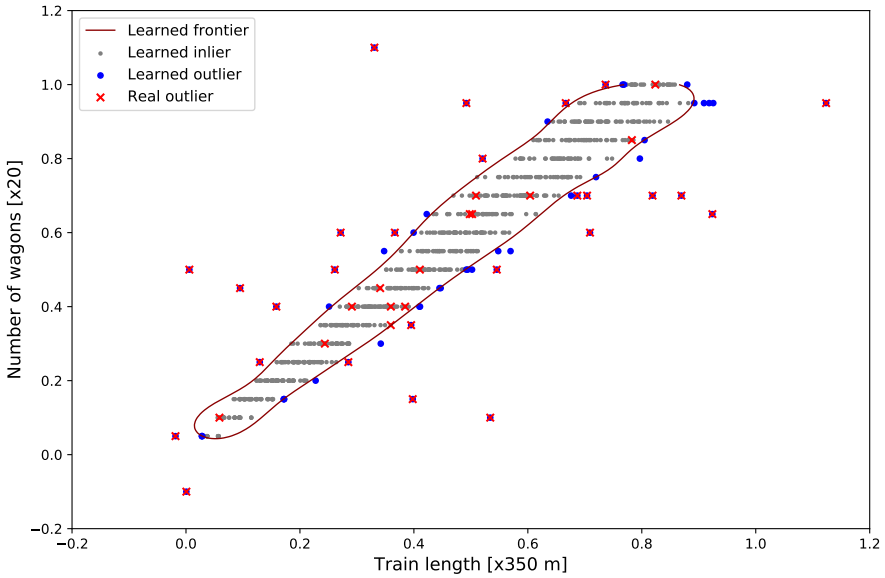
Table 7.6: Small versus large sample size, optimised results, added wagon lengths, SVM classifier.

((a)) Small (n=5000) AUC-ROC: 0.57.

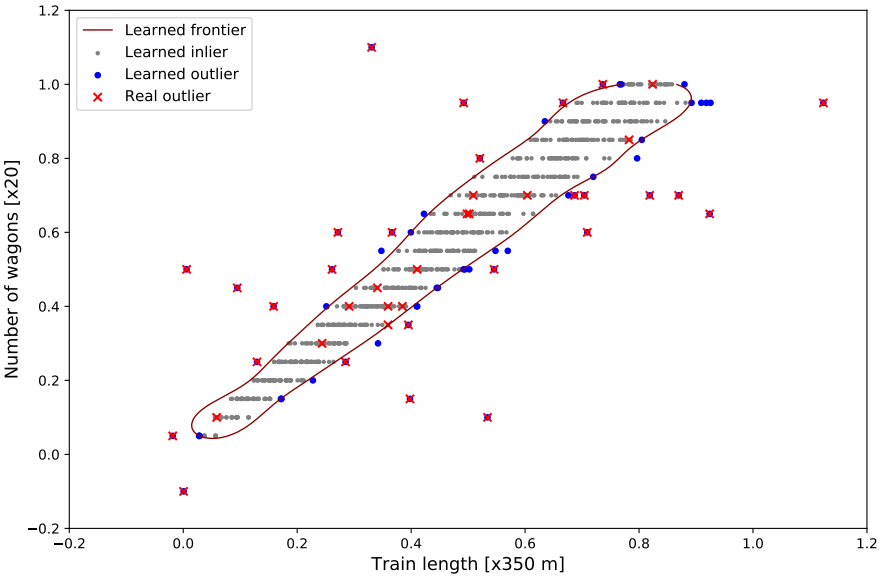
		Predicted: Yes	Predicted: No
Actual	Yes	4.7%	0.2%
	No	78.0%	17.1%

((b)) Large (n=100,000), ROC-AUC=0.87.

		Predicted: Yes	Predicted: No
Actual	Yes	3.8%	1.1%
	No	3.9%	91.2%



(a) Small dataset



(b) Large dataset

Figure 7.16: Scatter plots of small (n=5000) and large (n=200,000 number of samples using an SVM classifier. Only the first 100 samples were plotted.

7.4.5. Ishikawa rule-based

The rule-based analysis yields superior results in comparison to the SVM classifier (table 7.7). Both true positives and true negatives are higher while false positives and false negatives are lower than the SVM results. This comes with an ROC-AUC score of 0.91, which is the highest score so far.

Table 7.7: Confusion matrices of the SVM classifier and the rule-based analysis, based on an Ishikawa diagram.

((a)) SVM classifier (two features), ROC-AUC=0.84.

		Predicted	
		Yes	No
Actual	Yes	3.5%	1.4%
	No	3.3%	91.8%

((b)) Rule-based, ROC-AUC=0.91.

		Predicted	
		Yes	No
Actual	Yes	4.0%	0.8%
	No	0.0%	95.2%

7.5. Conclusion

The question to be answered in this case study was:

> How do classifier and design choices influence type I and type II errors in identifying faulty freight train registrations?

It was shown that classifier and design choices yield very different results with regard to type I and type II errors. Even with the same performance score of one metric, different trade-offs could be observed. Based on these observations, two conclusions can be drawn. The first conclusion is that not only one metric (performance score) should be used when measuring the performance of an algorithm. Both type I and type II errors should be taken into account. All of the classifier runs showed errors, so the trade-off between the two types of errors is particularly important. The second conclusion to be drawn is that an Ishikawa diagram in combination with a rule-based classifier shows potential for anomaly detection. A detailed discussion of the results is included in the next chapter.

8

Discussion

This chapter provides a discussion on the outcomes of the results of this thesis. It starts with separate discussions on all four parts of the methodology: the conceptual, practical, formal and numerical analysis. With the findings of this discussion, a framework is constructed in the next section. The chapter is concluded with the two main contributions of this thesis: a discussion about ML capabilities for safety-critical applications, and a discussion about organisational capabilities for implementing ML applications.

8.1. Conceptual analysis

The conceptual part of this thesis looked at the combination of ML, safety, risk, cyber systems and physical systems.

8.1.1. Interpretation of results

The analysis of the combination of safety and ML resulted in a distinction between risk, epistemic uncertainty and harm. It was shown how to decrease empirical risk mathematically. Also, safety engineering and risk management strategies were linked to ML to show how to improve safety. Breaking up safety and ML in these concepts gave a couple of valuable insights. Firstly, the concept of epistemic uncertainty is a crucial factor in the challenge of building a safety case for ML. Epistemic uncertainty describes the non-deterministic character of ML applications. If ML applications would only be deterministic, a conventional risk analysis would suffice when constructing a safety case. Risk managers would analyse the risk of the occurrence of an error and the implications of this error. With epistemic uncertainty, the chance of an error cannot be expressed since it depends on the ability to induce from training data to the unknown real-world distribution.

The second result was the application of safety strategies from safety engineering and risk management theory to ML applications. This provided a structured way of stating strategies to improve the safety of ML applications. Strategies from different fields than engineering and management could also have been picked since they can be expected to show the same structure. The first strategy is to eliminate risk by taking out risky parts of the system. The second strategy is decreasing the likelihood of a risk occurring. When the first two strategies did not work, and an error occurs, the third strategy is about damage control. Finally, the fourth strategy is about dealing with the consequences of the damage. Using this structure made it easier to distinguish between different safety strategies.

Thirdly, the combination of cyber systems, physical systems and risk was examined. Risk was categorised in terms of type I, type II and type III errors. This proved to be helpful to address risk in a categorical way instead of probabilistic. With this distinction, safety problems were shown of combining cyber and physical systems into cyber-physical systems, which is the case when using ML applications for safety. The impact of errors in cyber-physical systems that was shown in this part is concerning, especially when combined with earlier findings that there is little literature on the safety of ML applications.

The last part of the analysis, the application of the concepts of ML applications and safety to industries, revealed that best practices in the industry are mostly absent, while it also showed safety threats when scaling up ML applications. Epistemic uncertainty plays a significant role here since it makes validation and verification of ML applications complicated. This inherent uncertainty makes it difficult for industries to guarantee safety. Best practices for setting up a safety case for ML applications were not found, which can also be at-

tributed to the fact that industries have not found ways yet to deal with epistemic uncertainty in a robust way. Still, ML applications can be observed in practice, for instance in the automotive industry. This means that organisations carry a great responsibility with regard to safety.

8.1.2. Limitations

Executing a conceptual analysis revealed valuable insights but also had some drawbacks. As was mentioned in the methodology section, it did indeed not show practical relevance. Concepts were taken from literature and revealed insights about current academic progress, but lacked a practical aspect. A possible cause is that companies and industries do not reveal state of the art or only publish about successful results. For instance, safety protocols for ML applications in the automotive industry are not easily shared. It is therefore not easy to see if the concepts that were found, are also relevant in practice.

Another drawback is that the analysis was based on induction, and can therefore never be complete. Some strategies to increase safety (based on the engineering safety categorisation) were mentioned in chapter 3, errors in a collection of cyber and physical systems were explained in chapter 4, and some industries were examined in chapter 5. These are just subsets of all possible strategies, systems and industries. Although sufficient for the goal of this thesis, this analysis is non-exhaustive and can always be expanded.

8.2. Interviews

Interviews were conducted with an expert in autonomous cars and an expert in chemical industries.

8.2.1. Interpretation of results

The two semi-structured interviews that were conducted resulted in a validation of the findings of chapters 3, 4 and 5. The themes and concepts that were highlighted as currently relevant by Heikoop (autonomous cars) and Twilhaar (chemical industry) were mostly already mentioned in these chapters. Although Heikoop possessed general knowledge about autonomous cars, his background was in cognitive psychology, and he is specialised in the man-machine interface. This specialisation explained his focus on the social aspect of autonomous cars, highlighting concepts such as situational awareness, liability and the social impact of an error. Twilhaar had an engineering background, which explains his focus on the technical aspect of safety in the chemical industry. The two backgrounds of the interviewees resulted in a varied mix of concepts that came up during the interviews. Another explanation for the variation in concepts that came up was the choice of executing the interviews in a semi-structured style. This allowed for the interviewees to talk freely about safety considerations that they deemed relevant.

8.2.2. Limitations

The approach of taking semi-structured interviews with two experts in different fields had some drawbacks. The first one is the small sample size. This drawback was already mentioned in the methodology section and became apparent in the results as well, although not as prominent as was expected. The two interviewees mostly mentioned specific themes that were relevant to them. In the case of this thesis, this turned out well since both validated a broad range of findings from the rest of this thesis. However, it is easily imaginable that more interviews could have been needed when different interviewees with other specialisations were chosen. Besides, the implicit assumption in this research is that industries struggle with the same challenges with regard to ML applications, but more interviews would be needed to validate this assumption.

The semi-structured approach allowed for an open interview where the interviewees could freely express their opinions. The advantage of this was that interviewer bias could be limited. The limitation of this approach is that interviews cannot be compared. One concept that is seen as relevant by one interviewee might be regarded as irrelevant by another. A stricter protocol would be needed to verify this. The current way of analysing already resulted in the label 'relevant' when a concept was coined and explained by the interviewee.

8.3. Content analysis

In the content analysis, the ISO31000 and IEC61580 safety standards were analysed to check whether they were suitable for the risk analysis of ML applications.

8.3.1. Interpretation of results

The analysis concluded by stating concepts that were missing in the safety standards and recommended areas of improvement. The reason for the lack of suitability for ML applications might be the broad character of the safety standards. Both safety standards were not specifically built for one industry but aimed to be suitable for application in a range of different industries. Therefore, their definitions of risk and safety need to suit a wide range of industries. Small changes in updated versions of the standards can impact many industries, so ISO and IEC might tend to be conservative with updating their safety frameworks. Nonetheless, these standards form a basis for more specific standards, so the suitability for including ML applications can impact these specific standards as well.

The absence of the notion of non-deterministic behaviour or complex behaviour in the standards illustrates that work needs to be done here. Systems are growing increasingly complex and errors do not just originate from malfunctioning components. Also, ML applications cannot just be subdivided to trace back errors to the sub-parts where they originated. These themes require a revision of the current safety standards to suit more complex and non-deterministic behaviour.

The findings of the content analysis are congruent with the findings from the conceptual analysis and interview methods. All of these methods point towards a lack of compliance for ML applications with non-deterministic behaviour in current safety standards.

8.3.2. Limitations

One limitation of content analysis that was mentioned in the methodology section is that it can be too liberal. By applying concepts from the concept analysis such as risk, uncertainty, and risk strategies, a theoretical base for the analysis of the safety standards was intended. Based on the results, this approach was successful to show what was missing in these standards in a systematic way.

Another limitation was the number of content available for the analysis. Two general safety standards were analysed, but given more time, the analysis of some more specific standards would have led to more specific conclusions. The current conclusions have a general character, which is consistent with the character of the two safety standards. For specific industries or applications, it might provide more insight to execute the same analysis for a specific safety standard.

8.4. Simulation

This section provides a discussion on the results of the Prorail case.

8.4.1. Interpretation of results

The case study aimed to find out how data and algorithm design choices affect false positives and false negatives in the Prorail case. Based on the results, the following findings can be depicted:

Design choices

Different design choices do affect false positives and false negatives. In the majority of the results a clear trade-off can be observed: opting for a lower true negative percentage will automatically lead to a higher false positive percentage. This is in line with prior expectations since a more extensive decision boundary includes more true negatives at the cost of more false positives. The situations where this trade-off did not exist indicate better or worse classifier performance.

Classifier performance

A situation where the classifier performed flawlessly (no false positives and no false negatives) was not found. For the situation of two features (number of wagons and train length), this has a clear reason. Both LOF and SVM classifiers were able to identify the apparent outliers, e.g. trains with a registered length of -35 meters (see figure 7.13). However, anomalies with a combination of length and number of wagons that were within the dense part of the scatter plot cannot be detected based on just these two features.

Adding extra features

Adding more features while keeping the sample size constant did not result in an improvement of classifier performance. This might be attributed to the fact that adding one extra feature resulted in multiple extra feature vectors, which overcomplicated the model. For example: when adding individual wagon lengths, the result is that each train contains 20 extra features (wagon 1 length, wagon 2 length...wagon 20 length).

Even trains that only contain a low number of wagons have these 20 extra features. As a consequence, the model likely overfitted. This hypothesis is strengthened by the observation that the AUC-ROC score and complexity of the model seem to be linked. The hazardous goods boolean addition caused each train to have 20 extra features, where each feature was assigned a binary label (true or false). This gave an AUC-ROC score of 0.74. One level up concerning complexity was the addition of the wagon-type for each wagon. This was a categorical variable with four possible values and produced 0.66 as the ROC-AUC score. Lastly, the wagon lengths were the most complex features, where each variable was real-valued. This resulted in the lowest score of 0.57. Based on these observations, it can be concluded that adding model complexity while keeping the sample size constant is a possible cause of the degrading performance.

Scoring metrics

Different scoring metrics lead to different trade-offs regarding false positives and false negatives. Depending on the cost of a false positive and false negative, one could consider different metrics. Even better would be to come to an aggregate metric where certain weight factors are assigned to false positive and false negative errors.

One way of doing this is executing a cost-benefit analysis where the consequences of type I and type II errors are quantified. Consequences should contain both financial and social aspects. This way, both type of errors can be compared. The next step is to assign weight factors to the two types of errors. These weight factors translate to deterministic parameters in the aggregate metric. Converting the findings of the cross-benefit analysis into an aggregate scoring metric is a possibility to link practical consequences to the digital trade-off of between type I and type II errors.

A problem of the proposed approach is the difficulty of quantifying and distilling consequences of errors into one weight factor. Bayesian networks could form a solution here. Instead of desiring one specific number, the Bayesian approach makes it possible to specify a weight range. This is appropriate when the exact cost number is impossible to determine, but a range can be specified. When finding the optimal classifier parameters, using this metric would result in finding a section on the Pareto front instead of one specific point. Both extremes of this section can then be evaluated using a simple what-if analysis; are the false positive and false negative rates at the extremes still desirable?

Sample size

Adding more samples does not necessarily lead to a better classification of anomalies. As explained previously, this might be attributed to the fact that some anomalies cannot be detected based on just two features. In other words: underfitting occurs. Some trains contain a realistic combination of the number of wagons and total length but are still anomalous because the ground truth differs or another feature is wrongly registered. The situation where the sample size was small performed slightly better than the one which used a big sample size, which could be explained because no standard deviation was calculated. Due to underfitting, the cases of a small and large sample size in combination with two features would be expected to perform similarly.

The situation where a bigger sample size was tested in combination with three features showed a considerable improvement. This strengthens the observation that two features are not enough for this case and resulted in underfitting. However, more samples were needed when the dataset contained three features to prevent overfitting.

Ishikawa rule-based

Using the Ishikawa diagram to come up with a rule-based classifier yielded the best results. This was to be expected based on the way the synthetic dataset was generated. In the real world, more types of trains exist which might not all be known. Therefore, boundaries for the rule-based system need to be set wider, lowering the true positive percentage. Still, this rule-based approach has the potential to perform well on real datasets. ML takes many samples to infer specific rules over its input data. The more rules, the more complex the model becomes, and the more data is needed to prevent overfitting. Humans can reduce model complexity and thus data requirements by using a rule-based approach to identify samples that are not physically possible and deleting those out of the dataset. The next step of the analysis would then involve using unsupervised anomaly detection to find the remaining anomalies.

8.4.2. Limitations

Some limitations should be addressed concerning the case study:

- The statistical significance of the results was not calculated, and results were not cross-validated. This is not a problem for the purpose of pointing out that different false positive/negative trade-offs exist. Nonetheless, when the aim is to extrapolate the findings to general cases, statistical significance and cross-validation should be taken into account.
- The synthetic dataset was not validated and is a simplification of reality. In reality, more types of trains exist, wagon dimensions differ, and hazardous goods are more than just a boolean. More information could also be used such as the direction that the train is heading, its speed, and its previously registered runs. The dataset also does not represent accurately how anomalies arise in the real system. The distinction was not made whether anomalies arose from the train itself or the data processing pipeline. Also, specific anomalies can exist at different processing steps. For these reasons, the dataset has no direct value for practical implication. Despite, it was sufficient for serving the purpose of this research since it allowed performance differences in algorithms and design choices to become visible.
- A grid search was used to find optimal parameters. More advanced optimisation algorithms could yield better results. Also, a grid with higher resolution and a larger parameter space could give a more extensive overview.
- The combination of using more parameters and a large dataset could yield the best performance. Due to the lack of computational power, this was not tested.
- On a real dataset, the rule-based classifier might perform worse than on the synthetic one, since the real world is not always normally distributed. Also, oddly shaped wagons might incorrectly be marked as anomalies, so the false positive percentage will be higher when working with a real dataset.
- A labelled dataset was used to measure performance and to optimise the hyperparameters of the classifiers. In practice, due to the absence of a labelled dataset, hyperparameters should be tuned using heuristics. This could lead to decreased performance in comparison with the grid search executed in this simulation experiment.

8.4.3. Practical implications

This section discusses implications of the situation where an anomaly detection classifier would be implemented in practice.

Barring the rule-based approach, the classifiers in this experiment all outputted relatively high false positive or false negative percentages. Let's take one the best performing classifiers, which is the SVM in table 7.2. Even though the system registers 95.0% of the trains correctly, and the classifier has a false negative percentage of 1.1%, it means that 22% of incorrectly registered trains will be missed. Also, out of all the predicted positives, only 3.8%/(3.8%+3.9%)=49.4% is actually a true positive. Therefore, even though the classifier seemed to perform reasonably well on first sight, half of the positives are false alarms, and almost a quarter of the incorrectly registered trains will be missed.

These high false positive and false negative percentages do not have to be a problem, but they are when one assumes that the system is 100% accurate. In the introduction, it was already stated that the goal of the Prorail smart camera system is to achieve a 100% accuracy rate. Even if system accuracy and anomaly detection accuracy would be improved, 100% seems far away at this stage. It might even be dangerous to assume 100% accuracy and to base safety precautions on the assumption that all hazardous goods trains are correctly registered. For instance, when one thinks the system is flawless, backup measures in case of an error might be forgotten. Or employees trust the system so much that they forget that it contains flaws.

It would arguably be better when it is acknowledged that the system is not 100% flawless. From there, a safety case can be constructed where backup measures for different kinds of flaws can be implemented. For instance: what happens when an inconsistency is found in a registered train which carries hazardous goods? Will the train be stopped and manually inspected? Moreover, who will be deemed responsible for the cost of such an inspection? From here, questions related to system performance can be asked. How many of these inspections do we find practical, how many false positives can we afford? How many hazardous goods trains do we maximally accept to miss? After these questions, the final step is to relate all of the answers to the system performance. In other words: how good should the system be in terms of false positives and false

negatives before we go to the implementation phase? All of these questions can only be included in the safety case when it is acknowledged that a system like the Prorail smart camera system is not flawless.

8.5. Building a Safety Framework

With the findings of this thesis, a framework is proposed. It is meant to serve as a starting point for the risk analysis of ML applications. The framework shows how errors and strategies are placed in relation to each other. This section starts with a short analysis of how the conceptual framework for the international classification for patient safety can be used as an analogy for the ML applications risk framework. Then, the different parts of the framework will be explained.

8.5.1. Origin

Structurally, the framework that is proposed is based on the 'Conceptual Framework for the International Classification for Patient Safety', designed by the World Health Organisation [146] (figure 8.1). Despite its name, the patient safety framework is not meant for classification. Instead, it offers a structured way for grouping and understanding patient safety concepts.

One can find similarities between the structure of the patient safety framework and the safety strategies that were defined in chapter 3. The framework starts with the occurrence of an incident. In ML applications, the analogy of an incident is an error. After an incident occurred, the next step in the patient safety framework is detection. This is defined as "an action or circumstance that results in the discovery of an incident" [146, p. 5]. Detection happens in ML applications as well, so this term can be kept. Following up 'detection' is 'mitigating factors'. Mitigating factors are "actions or circumstances that prevent or moderate the progression of an incident toward harming the patient" [146, p. 6]. The concept in the safety engineering strategies that is most analogous to 'mitigating factors' is 'safe fail mechanisms'. They both aim to impede the progression from error to harmful impact. Detection can be seen as part of safe fail mechanisms as well, but is left as a separate step because of its importance. After an error caused harmful impact, the patient safety framework states that 'ameliorating actions' can be taken. These actions are aimed to compensate harm after the occurrence of an incident. The safety engineering strategy of 'procedural safeguards' matches this aim. The left and right side of the patient safety framework are comprised of 'actions taken to reduce risk'. For ML applications, this can be translated to 'Inherently safe design' strategies, since these aim to inherently reduce the risk of an error.

The next sections will provide elaboration on the different parts and connections of the framework. Examples will be given based on the Prorail case of chapter 7.

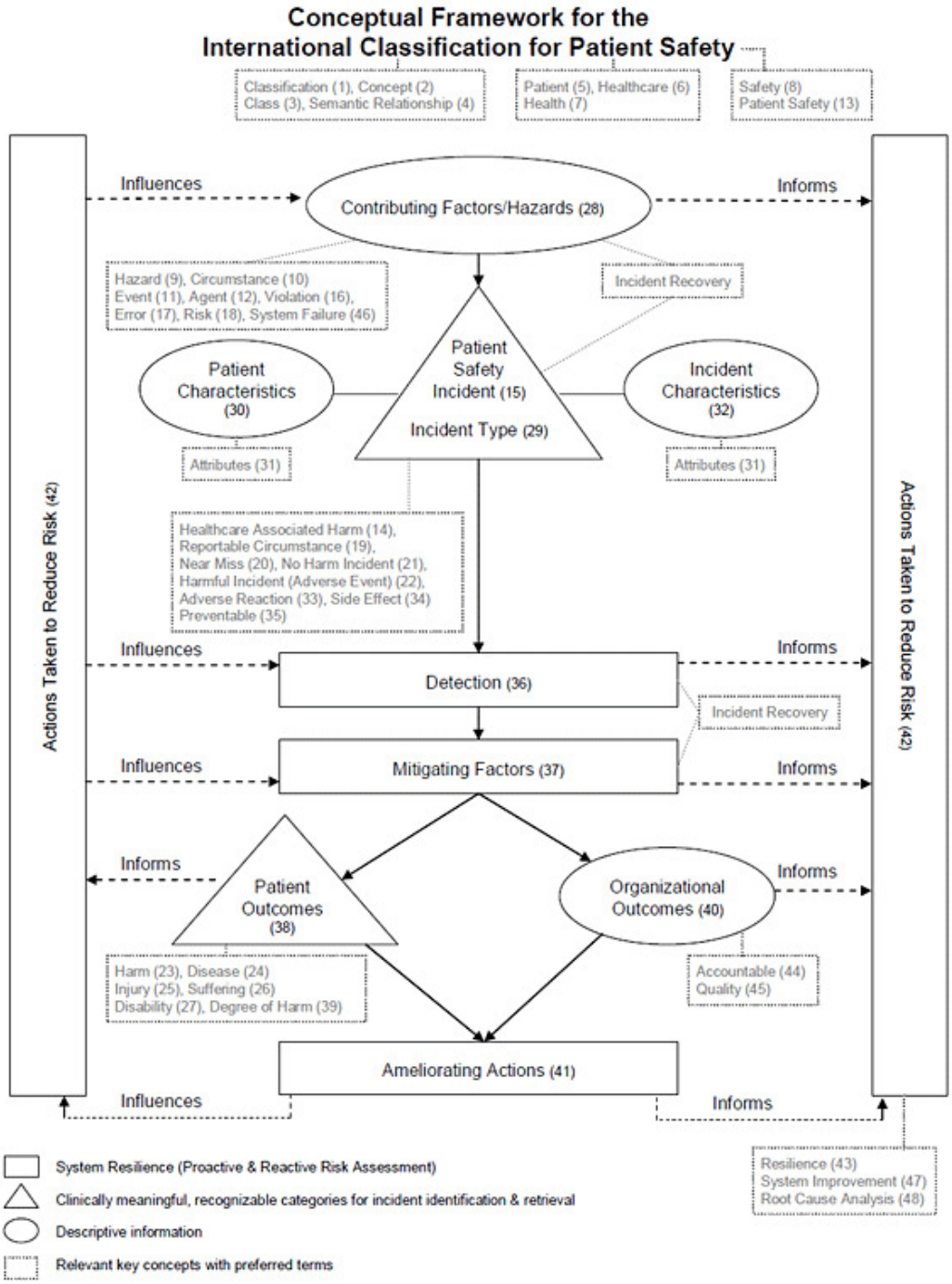


Figure 8.1: Conceptual framework for the International Classification for Patient Safety. Image reprinted from [146].

8.5.2. Parts of the ML applications risk framework

Contributing factors

Contributing factors are circumstances that play a part in or increase the risk of an incident. In [146], the distinction is made between three types of contributing factors: human factors, system factors and external factors. These can be translated to ML applications. Human factors for the Prorail case could include an unreadable or faulty hazardous goods tag. An example of a system factor is a flawed data pipeline or an unreliable sensor. External factors are factors beyond the control of the organisation, such as the weather condition.

System characteristics

Systems that use ML can be divided into three categories: cyber-physical systems, decision sciences and data products [180]. As discussed in chapter 4, cyber-physical systems are systems that integrate computational algorithms and a physical system, such as autonomous cars, smart grids, or surgical robots. Decision sciences are systems where ML is used to aid decision-making, e.g. medical treatment or predictive policing. Data products are automated products like digital advertising, media recommendations, or email spam filtering. Each type of system has its characteristics. Characteristics that describe systems include:

- *Type of interaction and interface.* The way systems interact with people differs. In the Prorail system, people interact with a digital interface where train registries are stored. The system shows relevant information for every train such as its length and content.

- *Scale of data.* The amount of data can be used as a way to characterise a system. Systems with few data points can behave in a less predictable way than systems where much data is available. For the Prorail system, the scale of data is the amount of train registries available.

- *Decision time frame.* This indicator expresses the maximum time that is allowed for a system to make decisions. In certain situations, an autonomous car needs to make decisions in tenths of seconds, while decision sciences usually have a wider time frame. The Prorail system is characterised by a large decision time frame. In case an accident occurs, emergency services could even choose to manually inspect the image of the involved trains to gain information on their contents.

- *Consequence time frame.* This variable indicates whether decisions have an immediate consequence, or whether it takes longer for a consequence to become visible. If an autonomous car makes a wrong decision, consequences can immediately show. Consequences of a faulty train registration could only show when an accident occurs, therefore being characterised by a longer consequence time frame.

- *Magnitude of consequence.* A decision might impact many people or much capital, while other decisions only have an impact on a small scale.

Error characteristics

Error characteristics can describe incidents. Firstly, the types of error can be analysed by specifying the meaning of type I/II errors in the particular system. Such an analysis would be similar to the analysis that was executed in chapter 4, where the meaning of type I and type II errors was analysed for a number of systems. This part of the framework would result in a description of the characteristics of a type I and type II error. A second characteristic is the error frequency. How often are errors expected to occur? This question can be challenging to answer and relies partly on the training dataset, as was described in chapter 3.

The error characteristics can be influenced by external factors such as weather conditions in case of smart cameras. Adverse weather conditions such as rain, snow or fog could negatively influence the performance of such cameras. Therefore, a dotted line was drawn from 'contributing factors' to 'error characteristics'. Furthermore, error characteristics can be seen as a subset of the system characteristics, hence the dotted line from 'system characteristics' to 'error characteristics' in figure 8.2.

Incident type

The box 'incident type' is comprised of the combination of 'contributing factors', 'system characteristics' and 'error characteristics' to describe an incident. It is used to map the different types of incidents, and under which circumstances they can happen.

Risk assessment steps

The risk assessment steps in the framework were already discussed in chapter 3 and 5 as strategies and will therefore only shortly be explained in this subsection.

Detection is an action or event that leads to the discovery of an incident. Accidents can be detected through manual inspection, monitoring, notification by authorities, or in other ways depending on the system. Inspection can be seen as the first step in damage control when an incident has occurred. The sooner an incident is detected, the more time there is available to intervene.

Safe fail mechanisms are constraining measures in case of an incident to minimise harm. After an incident, procedural safeguards come into place. The safeguards are meant to compensate harm after an impact has occurred.

Inherently safe design is placed along the sides of the framework since it influences or is influenced by all elements in the framework. It is the first strategy that was described in chapter 3. Also, an example of inherently safe design considerations are the ones in chapter 7. In this section, its influence on other elements in the framework will be described from top to bottom. Firstly, it influences 'contributing factors/hazards' since an inherently safe design strategy could eliminate one or multiple contributing factors, thus decrease inherent risk. The other way around, contributing factors also influence inherent risk strategies. For detection and safe fail mechanisms similar reasoning applies. Inherently safe design strategies will change the change of an incident, and therefore influence appropriate detection and safe fail mechanism strategies. Also, (a lack of) detection and safe fail strategies might require a different risk threshold, thus different inherently safe design measures. Lastly, when an incident with severe harm has occurred, procedural safeguards might feed back to inherently safe design since severe harm can require to lower inherent risk drastically. No line is drawn from 'inherently safe design' to 'procedural safeguards' since inherently safe design only influences procedural safeguards through preceding safety strategies and after the harm has already been done.

Social/environmental and organisational outcomes

It was shown in chapter 4 that social or environmental outcomes are one of the ways to describe harm. These outcomes might also be characterised as physical damage. One might look at the number of casualties and wounded, or environmental damage.

The second way to characterise harm is organisational outcomes. Organisational outcomes are consequences for the company itself. These consequences come in the form of reputational damage/media coverage, legal action, accountability, or financial cost. Chapter 4 showed examples of these organisational outcomes.

Organisational outcomes are partly dependent on the acceptability of risk. This topic was touched upon but deserves a short reflection since it is an integral part in the framework. Only when risk is deemed unacceptable, it is regarded as a safety issue. As a result, when we are talking about risk in socio-technical systems, it should be denoted that the acceptability of risk is subjective and varies across different sectors. Take for instance modes of human transportation. Fatality numbers per mode of transport can be found in table 8.1. Commercial aviation is one of the safest ways of transportation with 0.07 fatalities per billion passenger miles (2000-2009, [153]). With cars, we accept a much higher fatality ratio. Passengers or drivers in a car have a fatality rate of 7.28 per billion passenger miles, which is about a hundred times higher than in aviation [153]. Riding a motorcycle forms the highest risk with a fatality ratio of 212.57 per billion passenger miles [153]. It is clear from these examples that risk acceptability is dependent on more than just real numbers.

Table 8.1: Passenger fatalities per billion passenger miles in the USA, 2000-2009. Table reprinted from [153].

Riding a motorcycle	212.57
Driving or passenger in a car or light truck	7.28
Passenger on a local ferryboat	3.17
Passenger on commuter rail and Amtrak	0.43
Passenger on urban mass transit rail (2002-2009)	0.24
Passenger on a bus (holding more than 10 passengers)	0.11
Passenger on commercial aviation	0.07

The acceptability of risk is dependent on other factors, such as trust in technology. E.g., car crashes happen every day, but when an autonomous car causes an accident, it is all over the news. The technology is new, but people already expect it to be flawless. Therefore, when an accident occurs, drastic measures are

taken. Uber took all of its autonomous cars off the road in Arizona after a deadly accident occurred [184]. Autonomous cars are aimed to be safer than humans, but in this situation, the risk acceptability was so low that a single accident was already enough to stop the project. This example again shows that risk acceptability is not necessarily rational.

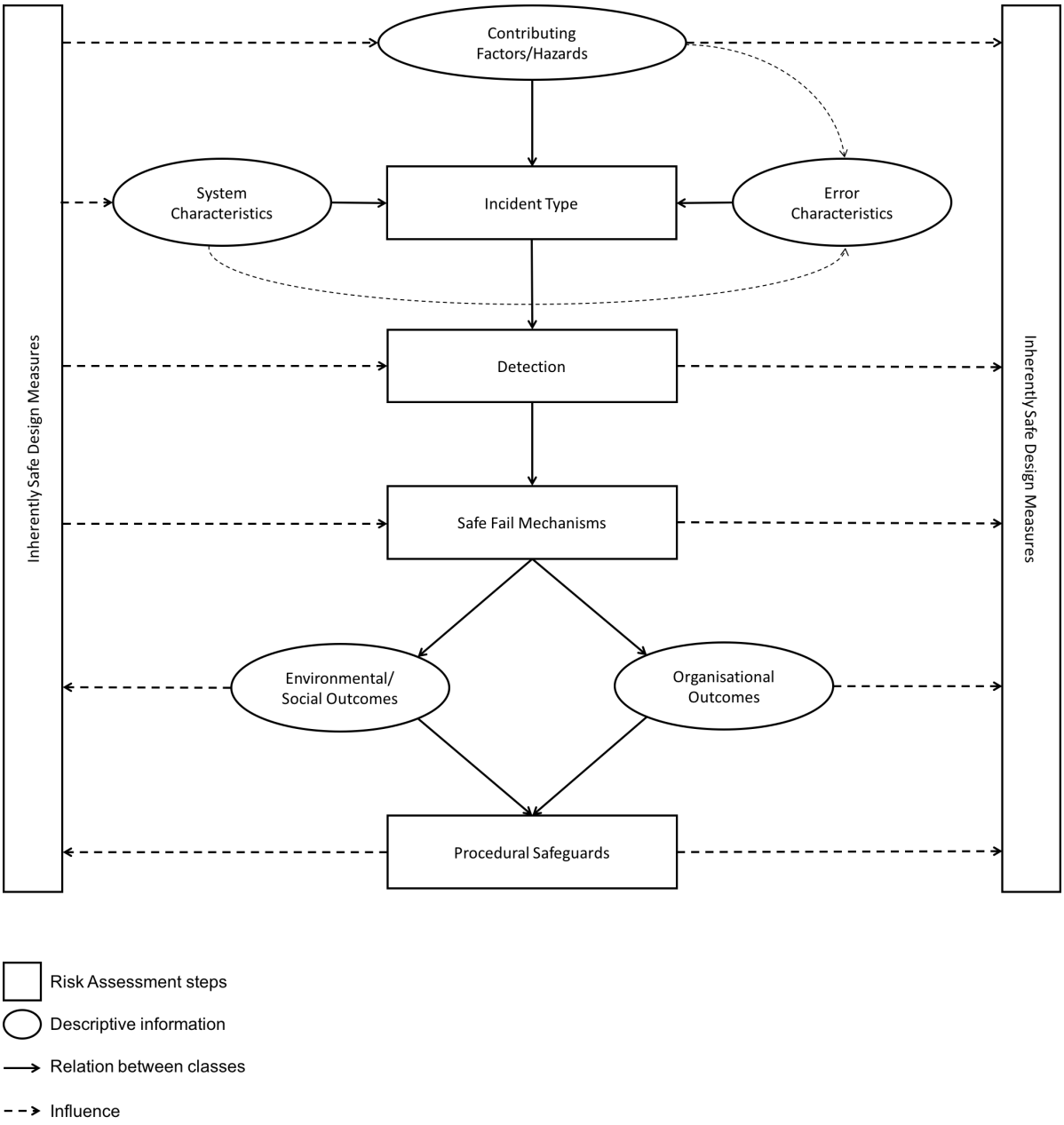


Figure 8.2: Safety framework for the risk analysis of ML applications.

8.6. Synthesis and contributions

Now that all findings are stated, discussed and structured, it is possible to reflect on ML capabilities and organisational capabilities when implementing ML applications in safety-critical socio-technical systems. These reflections synthesise the findings of the mixed-method design in this thesis and form the two main contributions of this work.

8.6.1. ML capabilities

Results of this thesis can be synthesised by addressing the no free lunch theorem. This theorem states that all classifiers perform equally well in terms of error rates when averaged over all possible data generating distributions [190]. In other words: no single classifier or ML algorithm performs universally better than any other algorithm in all situations. What this means for safety is that the right algorithm needs to be chosen for a specific domain or problem context. However, what are considerations for choosing the right solution?

Let's start with the algorithm design process. The Prorail simulation case showed how vital design choices are. A broad range of factors like sample size, feature size, type of features, type of classifier and performance metric all influence the trade-off between type I and type II errors. Just optimising an algorithm is therefore not possible since optimal solutions are about trade-offs between type I and type II errors. One should determine what the maximally acceptable false negative and false positive ratios are. To make such estimates, domain-specific knowledge is needed.

The first thing to estimate is the cost of type I and type II errors. Chapter 4 analysed the possible costs of type I and II errors in terms of consequences. It was shown how severe consequences of errors can be when combining cyber and physical systems (which is the case with ML applications). For this reason, risk managers should map possible domain-specific consequences of type I and II errors. The subsequent question that should be asked is then: how many occurrences of type I and type II errors are acceptable?

After analysing the cost and acceptable rate of type I and type II errors, one can look at strategies to increase safety. Chapter 3 analysed strategies from safety engineering and management theory that can be applied to ML applications. The notion of epistemic uncertainty was coined in this chapter as well. This notion is relevant for safety strategies since it implies that ML applications inherit uncertainty, so just optimising the algorithm itself is not a sufficient strategy for building a safe system. In other words: the algorithm will never be flawless, so also other strategies should be taken into account to increase safety. Using the framework of chapter 3, optimising the algorithm can be considered as an 'inherently safe design' or 'terminating risk' strategy. The other suggested strategies (safety reserves, safe fail, procedural safeguards) should be implemented as well. As mentioned before, this can only be properly done after accepting that algorithms are not flawless, thus realising that additional measures are needed.

When analysing the type I and type II error trade-off, it is important to realise that deciding not to use algorithms for safety is also an inherently safe design choice. It might just be that ML applications perform worse than the 'non-algorithmic' situation. E.g., an algorithm might lower the number of false negatives. If it does this at the expense of increasing false positives, it might be better not to use ML applications for safety in that specific situation. Again, this depends on the cost of both type I and type II errors for this case.

It would be needless to list all of the preceding considerations regarding ML and safety trade-offs when current safety standards already include such considerations, but they do not. The analysis in chapter 6 showed that existing safety standards are not yet suited for helping to find safety strategies for ML applications. Chapter 5 showed that industries have not yet found a comprehensive way to build a safety case either, but it also showed that safety issues increase when ML applications are applied on a larger scale.

This section started by citing the no free lunch theorem. This posed the question about considerations for choosing the right context- or domain-specific solution for ML applications. A synthesis across the chapters of this thesis resulted in a list of consecutive questions. The first question that risk managers should as themselves is: what are the costs of type I and type II errors? What rates do I find acceptable, thus what is the desired trade-off? Can I approach these rates with the current state of algorithms? And finally, what strategies do I put in place in case an error is made?

8.6.2. Organisational capabilities

Once an organisation decides to implement ML models into its practice, what does the process of integration look like? One can use the findings of this thesis, but how is the implementation process managed? Stage gate models (SGM) could offer a structured guideline. SGM will be reviewed in this section to come up with recommendations for organisations about how to use the findings of this thesis for managing ML applications.

SGM are tools for managing and controlling innovation efforts [48]. These efforts could include process improvement, business change, or software development. SGM are comprised of distinct stages (phases) which are separated by gates (decision points). Each gate marks a point where a decision is needed whether or not to proceed to the next stage. Traditionally, SGM contain a discovery phase, five stages and five gates, as shown in figure 8.3. In the next paragraphs, each phase will be explained and the findings of this thesis will be used to give recommendations regarding these phases.

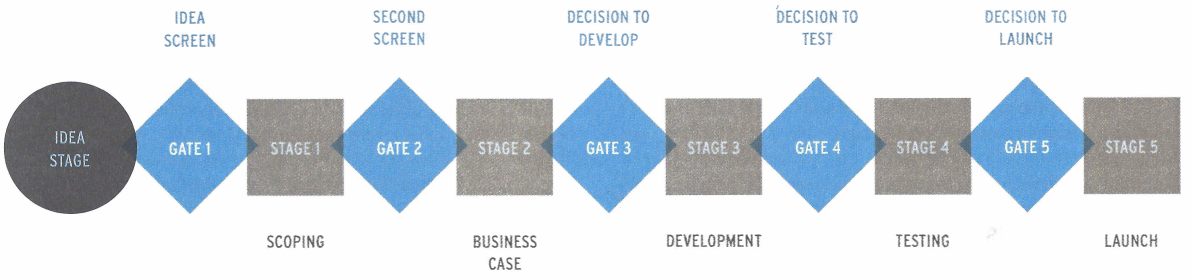


Figure 8.3: Five stages and four gates from the SGM framework as proposed in [49].

Stage 0: discovery
The discovery stage is skipped for this discussion since the concept is already clear: to apply ML applications for increasing safety.

Stage 1: scoping
This stage involves scoping the idea of stage 0. The risk manager has to decide what processes should be automated to increase safety. What are specific tasks that ML applications should fulfil? Where does the risk manager think that improvements can be made? Tasks should be specific since it was explained in chapter 2 that general AI is still far away from implementation.

Stage 2: building the business case and plan
Once the tasks are specified, the business case and plan can be built. This is the last phase of concept development. Building a plan for ML applications involves three aspects. Firstly, the steps regarding the analysis of type I and II error cost in section 8.6.1 should be followed. These steps result in constraints on type I and II error rates. When this analysis is executed, one should look at the technical feasibility of the desired ML applications in combination with the maximally allowed error rates. At this stage of the project, feasibility can only be estimated, so expertise is needed from experts in the field of ML. The last aspect that should be analysed in this second stage is what happens in case of an error, and what strategies can be implemented to avoid or minimise impact after an error has occurred. The framework of chapter 8.5 can be used as guidance for this step of the analysis. It shows that the impact of errors can be decreased at different levels, and what kind of strategies can be designed for this purpose. With this, it forms a synthesis of the risk management strategies for ML applications that were defined in this thesis.

Stage 3: development
During this stage, the plan that was developed at stage 2 is actually executed. This involves executing the development of new ML software, system architecture, and risk management strategies. One can choose to first develop a system at small scale or to develop it at large scale immediately. For example, the Prorail smart camera setup was first tested on one location and will be expanded in case of success. Therefore, stage 3 and 4 are often alternated and not linearly executed.

Stage 4: testing and validation
Validation of ML algorithms is still an open problem due to epistemic uncertainty and their black box character, as was found in theory (chapter 3) and practice (chapter 5). This emphasises the importance of proper risk management strategies in stage 2. Current safety standards are not yet suited for ML applications (as was analysed in chapter 6), so companies carry a great responsibility to validate ML applications safety strategies. This also requires extensive testing and data collection before implementation.

Stage 5: product launch
The product launch is the last stage of putting ML applications into use. Next to all technical aspects of the previous stages, this involves social aspects such as training employees how to use the new technology and what to do in case of errors or alarms. Employees should trust the outcomes of the new system and should keep their trust in case of false positives. This is a vital aspect of ML applications since usefulness is lost and safety can be compromised when employees do not trust outcomes.

9

Conclusion and Recommendations

Given the findings in the past chapters, conclusions to the research questions will be discussed in this chapter.

9.1. Answering the sub-questions

Sub-question 1

> What is revealed by looking at ML and safety in an integrated manner with four different approaches?

In this thesis, four methods were used to look at ML and safety: conceptual analysis, interviews, formal analysis and simulation. This revealed a number of insights. Firstly, it became clear that dealing with epistemic uncertainty is a difficult task at this point. The black-box nature of ML algorithms makes decision-making intransparent. Although academic interest present on a software level, analysis of a number of practical cases in this thesis showed that safety on a system level is still a problem. When accidents happen in certain industries, ML applications are temporarily put out of order, but a fundamental solution is non-existent. In other safety-critical industries, ML applications are not yet implemented due to their inherent uncertain character. This remains one of the most important challenges in this field.

A second finding is the challenge to cope with type I, type II and type III errors in cyber-physical systems using ML applications. It was shown in chapter 4 that both cyber systems and physical systems separately inherit serious safety risks. When combined, these risks are even greater than the sum of both. To become aware of this is the first challenge. To decrease these risks is a second challenge.

Chapter 5 analysed how industries maintained the safety of ML applications and coped with their uncertain character. This analysis showed that challenges remain. Although research is done about practical implementations of ML applications and safety, actual implementation into practice remains troublesome. Also, regulating bodies such as governments face the challenge of allowing innovation which might improve future safety, while maintaining current safety.

Two interviews were conducted: one with an expert in autonomous vehicles and one with an expert in chemical industries. Both validated the findings from the concept analysis about ML application challenges such as inherent uncertainty, error trade-offs, impact or errors and strategies to increase safety.

Current general safety standards are not yet adapted to suit ML applications. In chapter 6, the ISO31000: 2018 and the IEC61508 were evaluated in light of earlier findings about the character of ML applications. The fact that ML algorithms can make wrong predictions is something that the aforementioned safety standards do not deal with. For standard-setting bodies it remains a challenge to include the ability of wrong predictions and non-deterministic behaviour into general standards.

The simulation showed that classifier choice, feature size, sample size and performance metrics all have an impact on the type I and type II error trade-off. Combined with the findings about the practical consequences of these errors, the importance of choosing an adequate trade-off is shown. Algorithms that perform equally might make different trade-offs between types of errors to achieve this performance. Therefore, a metric was proposed that explicitly includes this trade-off instead of just a performance score.

Sub-question 2

> How can a framework be developed with the findings of SQ1?

The conceptual framework for the international classification for patient safety was taken as a starting point for developing a framework to address ML and safety. Management and engineering safety strategies that were found after answering SQ1 were used to form a basis for the framework. The type of error, and environmental, social and organisational harm were included in the framework as factors since their relevance was pointed out in SQ1.

The framework starts from the occurrence of an error, detection of the error, reduction of the impact of the error, and finally dealing with the consequences of the error. Following the steps of the framework offers a new way for the risk analysis of ML applications since it provides a structured way to gain insight in the prolongation of an error in these systems.

Sub-question 3

> Based on the findings of SQ1 and SQ2, what conclusions can be drawn concerning ML capabilities?

Answering SQ1 and SQ2 made it possible to reflect on how far we can take ML in socio-technical safety-critical applications. Currently, one can conclude that ML algorithms are never flawless. This posed the question about considerations for choosing the right context- or domain-specific solution for ML applications. A synthesis across the chapters of this thesis resulted in a list of consecutive considerations. The first consideration is the cost of type I and type II errors. Since algorithms will inevitably fail at a point in time, knowing the cost of errors is an integral part of finding an ML solution. Consequently, one needs to consider what rates are acceptable, i.e., what is the desired trade-off between type I and type II errors? Following up this consideration, one has to estimate whether these rates can be approached with the current state of algorithms. And finally, what strategies can be put in place in case an error is made? All of these considerations play an important role in determining whether ML can increase safety for given applications.

Sub-question 4

> Based on the findings of SQ1, SQ2 and SQ3, what conclusions can be drawn concerning organisational capabilities?

This research question was aimed to give an idea about the process of integration once an organisation decides to implement ML models into its practice. Stage gate models were used as a structured guideline to discuss organisational capabilities. In stage 1, the risk manager should specify what tasks ML applications have to take over. The next stage involves a risk analysis by following the steps in the framework of SQ2. During stage 3 and 4, the software and hardware are developed and tested. Since validation of ML applications is still an open challenge and proper safety protocols are missing, companies carry a great responsibility to come up with adequate safety strategies.

9.2. Answering the main research question

> What are safety considerations when using ML for socio-technical safety-critical applications?

This thesis used a mixed-method design and applied four methods to explore the field of ML and safety. With the results of these four methods, a framework was constructed for the risk analysis of ML applications. All findings were synthesised in two ways: by drawing conclusions about ML capabilities, and by drawing conclusions about organisational capabilities. This approach provided a way to think conceptually about how far we can take ML to increase safety in socio-technical safety-critical systems.

9.3. Recommendations for further research

Based on the conclusions and limitation of this research, some directions for further research are proposed:

- *Combine ML and Ishikawa/rule-based classifiers.* Both methods showed potential in the case study. It will be interesting to see how a combination of the two methods performs in a real case. For this, a real dataset should be used, and a full Ishikawa diagram should be constructed to come up with a set of accurate rules for the rule-based classifier.
- *Go in-depth about specific algorithms.* SVM and LOF were used in the case study, but these or other algorithms can be explored in greater depth. Specific topics can be on the influence of the parameters on type I and type II errors or how data preparation influences error rates.
- *Execute the case study with a real dataset once the system is ready.* For Prorail, it would be vital to know how high the real type I and type II error rates are, and how these can be minimised. For such a study, a real dataset is needed as well as experts to label the data.
- *Add cross-validation and statistical significance.* Cross-validation would add statistical depth to a possible follow-up study.

Concerning the full study, the following is recommended:

- *Expansion towards specific contexts.* The findings of this study with regard to design, validation and error trade-offs can be expanded to specific industries. This way, safety hazards can be mapped and solutions can be sought in a structural way.
- *Adaption of existing safety standards.* In this thesis, it was concluded that existing safety standards are not suited for validating or certifying ML applications. Although some suggestions were made to improve the IEC and ISO standards, this field is open for research.
- *Legislation.* Government legislation in the area of safety-critical socio-technical ML applications is a field that needs much further research. As was shown in this thesis, legislation is underdeveloped. It will be a significant challenge for governments to legislate the coming technological proceedings in the field of ML applications. Findings in this thesis offer a start for solving this challenge.
- *Type I and II error trade-off analysis.* In the discussion section, it was opted that a cost-benefit analysis can give insight into the desired type I and type II error trade-off. Moreover, Bayesian decision theory can then address uncertainty and can help to find robust policy options in light of social values. Working out this method is open for further research.

A

Industry interviews

A.1. Interview protocol

A.1.1. Introduction

Length: 40 minutes
Primary goal: to understand how you deal with uncertainty when implementing machine learning applications.
Method: semi-structured interview. The following questions are used as a guideline to address certain topics, but the interview might deviate from this line.

A.1.2. Background information

- Can you tell me something about your background? Guiding questions:
  - Where do you currently work?
  - What is your role?
  - What is your educational background?
- Before we start, I will tell something about myself and my research. I am a MSc. student in a program that is called 'Engineering and Policy Analysis' at the faculty of Technology, Policy and Management at the TU Delft. Currently I'm working on my thesis. It is about the safety of applying machine learning in safety-critical systems. We see a trend where more and more safety-critical systems are automated and governed by machine learning applications, but current research about the safety of these applications is lacking. Therefore, the goal of this part of my thesis is to analyse best practices in industry about safety cases for machine learning applications.

A.1.3. Context

Can you tell me about machine learning in your field of work?

- What is the current role of this technology in your field?
- In what applications is it already implemented?

A.1.4. Errors

How do control systems in your field of work deal with errors?

- What kind of errors can occur in your field of work where machine learning applications are used (false positives/false negatives)?
- What impact can these mistakes have?
- How is the trade-off made between false positives and false negatives?

A.1.5. Testing

What are best practices for testing safety?

- How is the safety of machine learning or control systems tested in your field of work?
- Are there any protocols to be used for testing?
- To what extent is it possible to test safety?

Questions about testing procedure. How are all scenarios taken into account?

A.1.6. Safety strategies

Risks analysis:

- What is done to map the risks of machine learning applications?
- Is there a standardised way for this?
- Do you think this procedure can be improved? How?

Inherently safe design:

- What design considerations are implemented for making the system more robust?
- How do you prevent biased training data?
- How do you make sure that you included all scenarios? (for instance, you train a camera to detect pedestrian, but forgot to include scenarios where it rains)
- How do you handle situations where your algorithm is confident in its prediction, but wrong nonetheless?

Fail safe:

- Are there fail safe mechanisms in place for when things go wrong?
- What are these mechanisms?
- How did you come up with those mechanisms?
- Can you think of scenarios that are not covered by these mechanisms? Which ones?

A.1.7. Regulation

What safety regulations are applicable to your ML product?
What do you think of these regulations?

- Do they make the product safer?
- Are they properly adapted to fit ML software?
- What are points of improvement?

A.1.8. Transcript Daniel

About:

- Educational background: Applied Cognitive Psychology.
- Current profession: "Postdoc as part of the interdisciplinary project "Meaningful Human Control over Automated Driving". This project entails intensive collaboration with a philosopher, a traffic engineer, and a behavioural psychologist (me). Together we will aim to address not only what the definition is of meaningful human control over automated driving, but also whether we can provide a framework to be used by a wide array of stakeholders, such as policy makers, governments and car manufacturers" [81].

Notes

D: Wij als onderzoekers van automatische auto's hebben een soort mantra dat we leren van de luchtvaart. De luchtvaart heeft al heel veel decennia bijna alles geautomatiseerd. 90% of meer van de complete vlucht is volledig geautmatiseerd. Daarom kijken wij dus naar de luchtvaart voor dat soort effecten. Wij als psychologen kijken ook naar wat het dan doet met mensen die voor lange tijd naar automatische systemen kijken. Als je alles automatiseert dan hoeft de mens niets te doen behalve opletten dat het niet fout gaat. Laat dat nou juist het ding zijn waar mensen niet zo goed in zijn. Dat is waar de fouten gebeuren en de ongelukken door ontstaan. Dat mensen denken dat alles goed gaat, en het zal ook vaak goed gaan, maar als het dan een keer niet goed gaat, hebben ze geen tijd en mogelijkheden om in te grijpen waar nodig, omdat ze dan compleet out of the loop zijn. Ze letten niet meer op waar nodig, terwijl ze dat eigenlijk wel zouden moeten doen. Maar ja, als iets heel lang goed gaat, dan zit je de hele tijd naar iets te staren. Dan gaat je allertheid naar beneden en raak je vanzelf out of the loop. Het is namelijk heel moeilijk om je aandacht bij iets te houden waar niets gebeurt. Dat is vooral klassieke literatuur uit de jaren 40 zelfs, die heeft aangetoond dat mensen dat niet lang kunnen volhouden. Hoe dat op te lossen is een heel groot vraagstuk. We zitten te denken over allerlei human machine interfaces zodat er interactie is tussen mens en systeem. Of monitors die kijken naar hoe de bestuurder zich voelt, en in welke staat hij is. Maar dat is allemaal nog erg lastig om te bekijken wat werkt. Hoeveel tijd kost het om iemand weer terug in de loop te brengen? Dat kost zodanig veel tijd dat men zich al begint af te vragen of we niet deze fase van semi-geautomatiseerd rijden moeten overslaan. Dat we niet allemaal volautomatisch moeten rijden.

F: Hoe kan automatisch rijden het verkeer veiliger maken?

D: Met volledig automatisch rijden zou je bijvoorbeeld peleton-rijden kunnen introduceren, waarbij auto's op minder dan een seconde van elkaar rijden. Hoge snelheden. Als dat allemaal volledig geautomatiseerd is, heb je geen bestuurders meer nodig. Bijvoorbeeld, vrachtwagens kunnen zich koppelen aan de platoon leader, en dan kunnen chauffeurs eigenlijk gewoon een andere baan gaan zoeken. Daar komt het dan op neer. Er zitten heel veel voordelen aan, ook op het gebied van benzine-verbruik, luchtweerstand, traffic flow, noem maar op. Er zijn schattingen gemaakt dat het voordelen tot wel 30,000 euro per truck per jaar oplevert. In die zin is automatiseren erg aanlokkelijk. Maar: tot aan volledig automatiseren heb je dus dat een mens er nog steeds controle over moet hebben. Om daar een zinnige vorm van controle over te hebben, is nog wel een lastig vraagstuk. Wat is zinnig? Hoe kun je ervoor zorgen dat een mens zinnig in controle is over zo'n systeem? Zoals ik net al noemde, is het voor een mens erg moeilijk om voor lange tijd de aandacht te houden op iets dat perfect functioneert, tot het een keer fout gaat. Dus dat is niet echt een zinnige vorm van controle. En dat is de reden dat er, bijvoorbeeld met de Tesla, ongelukken gebeuren. Mensen denken dat iets goed gaat, mensen denken dat een systeem daarmee om kan gaan, maar als ze gewoon de manual hadden gelezen. In de manual van de tesla staat dat je moet opletten, en dat de auto niet om kan gaan met dat en dat en zus en zo, en alleen onder de ideale omstandigheden goed kan functioneren. Maar zelfs dan moet je zelf ook nog op blijven letten, en je handen aan het stuur houden, voeten op de pedalen houden, en ingrijpen wanneer het nodig is. Dus de technologie staat in dat opzicht nog zodanig in de kinderschoenen. Natuurlijk zie je veel succesverhalen door allerlei autobedrijven en technologiebedrijven. Maar dat wordt alleen onder ideale omstandigheden gedaan. In die zin zijn we nog vrij ver weg van waar we moeten zijn.

F: Waardoor komt het denk je dat de technologie nog in kinderschoenen staat?

D: Volledig autonoom rijden kan alleen maar op gecontroleerde trajecten, zoals bijvoorbeeld bij de havens van Rotterdam. Daar heb je automatische constructies (volgens mij zijn het niet eens auto's die daar rijden), van die constructies die alles volautomatisch reguleren. Dat is echt een closed-loop circuit. Om dat soort technologie op de openbare weg te krijgen zijn er zo onnoemelijk veel factoren die van invloed zijn. Dat is ook iets wat ik zelf de laatste tijd heb genoemd. De analogie tussen de luchtvaart en de autoindustrie is dat luchtvaart 2d is, waarbij autorijden 3d is. Op basis van hoeveel factoren er meespelen. In de luchtvaart heb je alleen stijgen en landen, en voor de rest is het gewoon 1 track door de lucht. Je komt amper in aanraking met andere dingen. Af en toe met andere vliegtuigen, maar that's about it. Op de weg heb je te maken met steden, bochten, voetgangers, fietsers, ander verkeer, noem maar op. Het is heel veel moeilijker om technologie uit de luchtvaart die we al decennia hebben, toe te passen op automatisch rijden.

F: Je noemde eerder dat de luchtvaart een grote leerschool is. Wat voor dingen kun je dan wel meenemen uit de luchtvaart?

D: Onder andere wat voor effect automatisering heeft op de bestuurder. Daar hebben we al gezien dat de task demand effect heeft op hoe goed ze hun aandacht erbij kunnen houden. Zelfs in een cockpit heb je al twee piloten en een copiloot. Dat zou je ook op de weg moeten hebben. Maar dat is natuurlijk compleet onhaalbaar, om drie mensen in een auto te hebben, puur en alleen om de aandacht op de weg te houden. Al dat soort studies gerelateerd aan vermoeidheid, en hoe ze in de loop kunnen blijven, en ook de technologie die die dingen automatiseert. Al weet ik niet precies wat ze daar allemaal uithalen.

F: Met al die risico die er bestaan, en bijvoorbeeld worden opgevangen in de luchtvaart met drie piloten, is het dan wel verantwoord om zelfrijdende auto's in de vorm zoals ze nu zijn, los te laten op de weg?

D: je zou dat vanuit een aantal aspecten kunnen bekijken. Allereerst, vind ik het zelf (persoonlijke mening) dat het goed is om, zoals Tesla het doet, de auto's op de weg te gooien, en aldoende te leren. Aan de andere kant weet Tesla ook dondersgoed dat mensen de auto gaan misbruiken, en echt niet allemaal de manual gaan lezen. Ookal zeggen ze in de manual dat je de auto alleen onder bepaalde omstandigheden kan gebruiken, mensen gaan daar natuurlijk niet naar luisteren, en gaan het misbruiken. In die zin is het niet helemaal te verantwoorden, maarja, alle revolutionaire technologie van oudsher is ontstaan door menselijke slachtoffers. Als je de mens veilig wil houden dan kom je ook niet snel verder. Er moeten natuurlijk allerlei regels zijn van overheidsinstanties die dan bepalen wat wel en niet mag, en wat je wel en niet moet doen. Zo zijn er bijvoorbeeld in Nederland reguleringen geweest die het toestaan om te experimenteren met auto's ook op de openbare weg. Er zijn ook een aantal trajecten die speciaal vrijgesteld zijn om hiermee te experimenteren. Zodoende creeer je een veiligere omgeving voor dat soort technologieen om te experimenteren. In die zin is dat een te verantwoorden methode om dat te doen. Dus je kunt het uit verschillende aspecten bekijken. Puur vanuit de psychologische kant, hoe mensen ermee om kunnen gaan, dan zou je je sterk kunnen afvragen om het uberhaupt zinnig is om die tussenstadia te proberen. Of je moet het helemaal volbouwen met allerhande HMI's, interfaces, driver's state monitors, zodat je echt alle aspecten afvangt. Maar het menselijk brein is heel complex. Om al die dingen af te vangen is utopisch. In die zin zou je kunnen denken: het is nooit verantwoord om met dit soort dingen te experimenteren, want het gaat toch nooit weg. Maarja, mensen kunnen ook leren. Als we dan mensen leren om met automatische voertuigen om te gaan, dus echt met drive training voor het halen van je rijbewijs, dan zou het misschien wel weer kunnen. Dus het is niet eenvoudig om te beantwoorden, zo'n vraag.

F: Mensen kunnen leren, en machines kunnen tegenwoordig ook leren. Daardoor kan het lastiger zijn om een garantie van veiligheid af te geven voor een systeem. Hoe denk je daarover met betrekking tot de autoindustrie?

D: ik weet niet of het moeilijker zou zijn, want in principe is een automatisch rijdende auto die zelf leert exact hetzelfde als een mens. Een mens leert ook aldoende, alrijdende hoe je wel en niet moet rijden. Waar je op moet letten, en hoe je moet reageren op bepaalde omstandigheden. In principe zou je een automatisch rijdende auto ook eerst een rijles moeten geven en een certificaat laten behalen. In principe doet een automatisch rijdende auto niets anders dan een mens doet. En dat is het paradoxale aan een automatische auto, in ieder geval gebaseerd op de levels op automation die we nu hebben. Gebaseerd op de SAE - Society of Automobile Engineering. Die hebben zes levels bepaald, waarbij 0 volledig handmatig is. Level 1 is rij-

den met automatische assistentiesystemen, zoals adaptive cruise control. Level 2 is waarbij zowel laterale als longitudinale controle wordt overgenomen door de automaat. Level 3 is dan waarbij in principe alles wordt overgenomen, maar er wordt nog wel van de bestuurder verwacht dat je overneemt en blijft opletten. En level 4 wordt niet eens meer verwacht dat de bestuurder kan overnemen. Dat de auto echt in alle omstandigheden kan reageren, dus in principe niet meer in een ongeluk terecht kan komen. Bij volledig automatisch rijden wordt de mens helemaal buiten schot gehouden. Bij level 1 is het dus hele basale taken. Bij level 2, longitudinaal en lateraal, gaat het om sturen en remmen. Dat kan de mens ook hartstikke goed. Naarmate het level hoger wordt, worden er steeds complexere taken overgenomen. Terwijl, wat je eigenlijk wil van een automatische auto, is dat de complexere taken worden overgenomen, zodat de mens makkelijke taken kan blijven uitvoeren. Want moeilijke taken zijn de taken waarbij dingen fout gaan. Als je de moeilijke taken laat overnemen, maak je dingen veiliger. De automatische auto's maken eigenlijk dezelfde stappen als de mens. De makkelijke taken kunnen ze eerst doen, en langzaam de moeilijke taken. In die zin is het paradoxaal dat automatische auto's de makkelijke taken overnemen, terwijl we dat juist goed zelf kunnen doen. Daardoor krijg je het probleem dat mensen out of the loop raken, als je gewoon simpele taken niet meer hoeft te doen. Dat is gewoon saai.

F: Wat denk je dan dat in dit stadium van techniek de functie van automatisering is?

D: Vanuit een engineering perspectief: wat kunnen we? Engineers proberen gewoon een auto zoveel mogelijk te laten kunnen doen. Vanuit een psychologisch aspect is het de vraag. Ik zie het zo dat het niet zo zinnig lijkt te zijn. Er zijn verschillende aspecten en perspectieven.

F: Denk je wel dat het de toekomst heeft?

D: Ja, je ziet overal dat het de toekomst is en zal gaan zijn. De technologie houdt niet op. Het blijft maar doorgaan. En natuurlijk, er zitten ontzettend veel voordelen aan. In de long run zie ik het als een soort Minority Report situatie. Auto's die over elkaar heen rijden, in peletons rijden. Instappen wanneer je wil. Dat is de utopie uiteindelijk. Dan heb je geen menselijke fouten meer. In die zin zou het geweldig zijn als we die kant op gaan. Die tussenstadia is het lastige punt, en dat is waar we nu tegenaan zitten. Er zijn allerlei systemen die level 1 zijn, bepaalde systemen die level 2 zijn. Sommige systemen hikken tegen level 3 aan. Maar die tussenstadia, dat is een lastig punt.

F: Hoe zou je de techniek door die tussenstadia kunnen leiden?

D: Allereerst moeten we bepalen wat een zinnige vorm is van menselijke controle over zo'n systeem. Als we dat niet hebben, blijven we op deze voet verdergaan, en blijven we dingen automatiseren waar de mens niet mee gebaat is. Alleen maar technologische hoogstandjes zonder dat de mens er een voordeel aan heeft. Dus we moeten allereerst bepalen hoe we de mens zinnig in controle kunnen houden. Dat is ook het project waar ik mee bezig ben. Allereerst de definitie vinden: wat is zinnig, wat bepaalt dat iets zinnig is? Naar aanleiding daarvan hoop ik een zinnige transitie over verschillende levels of automation te kunnen bepalen. Dus vanuit een menselijk aspect. Voor mensen daadwerkelijk zinnig delen van de rijtaken automatiseren, zodanig dat hij nog steeds zinnig in controle is. Dat perspectief streef ik na.

F: Heb je het hier over een tijdelijke situatie?

D: Uiteindelijk, ja, maar we hebben nog geen flauw idee over hoe tijdelijk dat is. Sommigen zeggen: in 2020 hebben we volledig automatische auto's, maar ik vind dat heel erg voorbarig. Sommigen zeggen 2050, anderen zeggen 2100. Dat lijkt me lange termijn genoeg om dit nog wel belangrijk te vinden. Als je nu nog 80 jaar rond blijft rijden met auto's die in de tussenfase zitten, dan heb je toch wel zoiets nodig. Maar inderdaad, zodra we volledig automatische auto's hebben is het belang van de mens niet meer relevant, want de mens speelt dan geen rol meer.

F: Je zei iets eerder dat het een goed idee is om mensen, net als auto's, een rij-examen te geven. Maar sommigen zeggen dat hier het pijnpunt zit van zelflerende systemen; deze systemen kunnen minder goed generaliseren dan mensen, dus het is moeilijk om te checken wanneer een auto veilig genoeg is om de weg op te gaan. Hoe denk je hierover?

D: Ik denk in die zin ook weer dat zo'n automatisch systeem dezelfde stappen ondergaat als een mens. In principe bestaat de technologie van zo'n auto uit sensoren, radars, LIDAR, etc., die de omgeving scannen. Dat is gebaseerd op stimuli, data-bundels, en dat is eigenlijk niet anders dan hoe een mens stimuli binnenkrijgt. Die krijgt beelden op zijn netvlies, wat ook data is, en daarop baseer je een bepaalde reactie. Een automatisch systeem doet eigenlijk niets anders. Dus in principe kan die dezelfde stappen bewandelen als een mens. Dus in principe zou je eenzelfde soort rijles kunnen ondergaan. Want in principe, als je een mens een rij-examen laat doen, zijn dat maar een bepaald aantal scenarios waar een mens op moet reageren. En als je daar goed op reageert, ben je geslaagd. Wie zegt dat je dat in vervolgsituaties ook altijd goed doet? Er zijn genoeg mensen die hun rijbewijs hebben en die als idioten op de weg rijden. In die zin is het hetzelfde.

F: een verschil zou kunnen zijn dat een mens die overdag goed kan rijden, 's nachts ook nog wel goed presteert, terwijl een machine geen zicht meer heeft onder die omstandigheden.

D: Niet per se. Tijdens een nacht rijden is toch wel heel anders dan overdag rijden. Soweiso heb je natuurlijk veel minder zicht, wat bij een automatisch systeem niet per se van toepassing hoeft te zijn. Als die op basis van radar rijdt, maakt dag of nacht geen verschil. Maar tijdens de nacht heb je ook ander volk op straat. Mensen die een borreltje teveel hebben gedronken, daar moet je ook maar mee om zien te gaan. Het is toch wel heel anders. Onvoorspelbaar gedrag, of gelimiteerd zicht.

F: Iets anders: in welke zin hebben jullie te maken met regulering?

D: Binnen dit project hebben we er wel mee te maken, want twee van ons team zijn filosofen. Die kijken naar de morele en ethische aspecten van het rijden met automatische systemen. Op basis daarvan kunnen dus ook voorstellen gemaakt worden van hoe ethische vraagstukken en morele vraagstukken moeten worden behandeld. Dat is een van de dingen die de laatste tijd meer aandacht heeft gekregen. Bijvoorbeeld het trolley problem. Dit zal bij een automatisch systeem niet anders zijn, wanneer de auto een keuze moet maken tussen een groep schoolkinderen aanrijden, of tegen een muur aanrijden waarbij de bestuurder omkomt. Nu heb ik vernomen dat een van de bedrijven die automatische auto's maakt, garandeert dat het systeem altijd voor de bestuurder zal kiezen, wat natuurlijk hartstikke logisch is. Anders koopt niemand meer zo'n auto. Dat lijkt me een hele duidelijke keuze. Je kan er natuurlijk ontzettend complexe algorithmes aan opplakken. Wat als het nog maar een kind is met een heel leven voor zich? Of in een slechte buurt waar de kans groter is dat het een crimineel is. Je kan het zo moeilijk maken als je zelf wil. Uiteindelijk moet er een keuze gemaakt worden, en is het logisch dat je voor de bestuurder kiest. Anders raak je het product aan de straatstenen niet kwijt. Dus dat soort dingen worden ook behandeld binnen dit project. En meer ethische/morele vraagstukken. Daar kunnen dan ook regels uit rollen. Daar zijn wij niet voor. Dat zullen de stakeholders binnen het project zijn. Maar een van onze partners is juristen, advocaten, verzekermaatschappijen.

F: Hoe zit het met deze problemen met aansprakelijkheid?

D: Laatst in het nieuws was een vrouw die overstak terwijl een automatische auto kwam aanrijden. Die vrouw werd aangereden omdat de auto niet op tijd kon remmen. Daarvan werd gekeken: wie is er nou verantwoordelijk voor dit ongeluk? Het blijk uiteindelijk de vrouw te zijn, want ook als er een mens achter het stuur zat, had die niet op tijd kunnen reageren. Met dit soort scenarios is het altijd de vraag wie er verantwoordelijk was. Is het de software engineer die de software heeft gecodeerd? Of is het het bedrijf dat de auto heeft geproduceerd? Is het de bestuurder? Of is het de auto? Is het het land dat ervoor heeft gezorgd dat de auto mag rijden? Dat soort kwesties komt ook aan bod bij ons.

F: Welke bestaande reguleringen zijn er?

D: Binnen Nederland is een aantal reguleringen dat het mogelijk maakt om met automatische auto's op de weg te zijn. Maar Europees breed is er geen regulering daaromtrent. Reguleringen gaan natuurlijk traag. Er gaat een heel juridisch proces aan vooraf, en er gaan jaren overheen. Deze technologie is zo snel. Binnen een aantal jaren zijn we alweer zoveel stappen verder. Regulering houdt dat niet bij. In die zin is het dwijlen met de kraan open. Maar we proberen dus wel op basis van wat we nu weten voorstellen te maken, op de manier

van: hier kan je iets mee doen. Officieel zullen we geen voorstellen doen, maar we publiceren werk en daar verbinden we conclusies aan. Op basis van die conclusies kunnen onze stakeholders daar wat mee doen. Ond er andere een odnerdeel van dit project. Maar het is lastig met dit soort nieuwe technologie, want zeker van de hogere levels van automatisch rijden weten we eigenlijk niet zo veel. We weten niet hoe ze eruit gaan zien, of onder wat voor omstandigheden ze kunnen werken. Wat dat voor gevolgen heeft voor de maatschappij, voor het verkeer, wat we daarmee aan moeten. Moeten we er speciale regels voor aannemen? Moeten we andere regels afschaffen? Het is allemaal toekomstmuziek, glazen bol werk. Dat maakt het heel lastig.

F: Toch wordt er in de auto-industrie al veel geexperimenteerd met dit soort technieken. Waarom denk je dat dit gebeurt?

D: Nederland is een voorloper op dit gebied, en wil een wereldleider blijven. Daarom zijn ze ook progressief met het aanleggen van de wetgeving hiervoor. In andere landen, bijvoorbeeld Amerika, is het afhankelijk van de staat. Ik had vernomen dat, naar aanleiding van dat laatste ongeluk, Arizona zich heeft teruggetrokken, en dat daar niet meer getest mag worden. In andere staten mag het wel, zoals California. Duitsland, Japan, Scandinavie zijn er ook al volop mee bezig. Maar andere landen willen er niets van weten, daar mag veel gebeuren. Het ligt er ook aan wat de mogelijkheden zijn. In Nederland is de infrastructuur een van de beste ter wereld. Als je naar Rusland gaat bijvoorbeeld is het allemaal wat lastiger. Zolang de lijnen van de weg zichtbaar zijn en de weg is netjes geasfalteerd, borden zijn duidelijk zichtbaar, je kan duidelijk van A naar B, dan kan een systeem dat aardig goed. Maar zodra er lijnen ontbreken, kan hij zomaar van de weg afrijden. Als er geen borden zijn, weet hij ook niet hoe hij zich moet gedragen met alle gevolgen van dien. Een mooie, overzichtelijke snelweg kan prima, maar als het minder wordt, zelfs door regen, donker, schemer, dan wordt het al snel chaos.

Automatische auto's zijn nu al veiliger dan handmatige auto's. Het is nieuw, hot, het komt dan in het nieuws, en zodra er iets gebeurt wordt het breed uitgemeten in het nieuws. Mensen denken dat zo'n automatische auto helemaal het einde is en alles kan, hartstikke geweldig is, terwijl dit eigenlijk niet zo is. Het is heel erg gelimiteerd in zijn gebruik. Onder ideale omstandigheden werkt het, maar anders ook niet. En dan gaan mensen dat misbruiken, met alle gevolgen van dien. Filmpjes kijken op de achterbank bijvoorbeeld.

F: Een oplossing kan zijn om heel veel meters te maken tot de auto veiliger is. Zo'n zelfde aanpak in de vliegtuigindustrie is natuurlijk lastig, want daar is een fout fataal. Hoe denk je over de impact van een fout in de autoindustrie?

D: Hier mag het ook niet fout gaan natuurlijk, want zodra het fout gaat wordt het breed uitgemeten in het nieuws, en kan er zomaar een staat bedenken: dit doen we niet meer. En op een gegeven moment mag het nergens meer. Natuurlijk moet de technologie goed gebenchmarkt worden, en het wordt natuurlijk ook uitvoerig getest in simulaties en closed track circuits, en heel erg gecontroleerde trials op de openbare weg. En rijsimulatoren. Het wordt zeker uitvoerig getest, maar fouten kunnen altijd ontstaan. Het is alleen jammer dat, om dat het zo'n nieuwe techniek is, het in het nieuws erg breed wordt uitgemeten als het fout gaat. En dat terwijl het al daadwerkelijk veilig is, volgens de data. Minder ongelukken per gereden kilometer.

F: Hoe zie je deze ontwikkeling in andere industrieen, bijvoorbeeld in chemische installaties? De problemen kunnen hetzelfde zijn, bijvoorbeeld de interface zoals je aangaf. Hoe zie je deze trend in andere industrieen?

D: In principe is de trend, dat je als actieve bestuurder wordt omgevormd tot een passieve supervisor van een automatisch rijdend systeem. Dus er komen hele andere taken aan bod. Iets waar je helemaal niet voor geleerd hebt. Bijvoorbeeld, in chemische industrie waarbij mensen moeten controleren of alles goed blijft werken, die zijn er specifiek voor getraind om alle metertjes en lichtjes in de gaten te houden, en zijn er specifiek op getraind om in te grijpen wanneer het nodig is. In een automatische auto is de bestuurder getraind om een auto te besturen, niet om een automatisch systeem te monitoren. Je moet er dus voor getraind worden, daar komt het op neer. Je moet leren om te gaan met zulke systemen. Het is een compleet andere taak wat je doet, en dat moet ook als zodanig worden gezien. Die trend hoop ik in ieder geval graag gezien. Dat het wordt meegenomen in rijlessen. Aan de andere kant: de verwachting is dat er steeds meer automatisch rijdende auto's op de weg komen. De saturation op de weg gaat steeds meer naar automatisch rijden. Naarmate er meer en meer van dat soort voertuigen komen, kan je het principe van swarm intelligence toepassen. Dus dat de automatische auto's zich als een soort van zwerm/peleton gaan gedragen, en dat ze elkaar direct

kunnen vertellen: over een paar kilometer is er wat, dus bereid je vast voor. Zo kun je veel beter anticiperen op mogelijke ongelukken. Ik denk dat dat de way to go is, vanuit mijn eigen perspectief. Het lijkt mij een belangrijke stap om goed te communiceren met elkaar. Dat lijkt mij een groot voordeel om te kunnenn doen. Wij mensen kunnen niet met mensen tien kilometer achter ons communiceren als er een auto gecrasht is. Automatische systemen zouden dat wel kunnen om zo de veiligheid te verbeteren. Die trend hoop ik te zien.

F: Nog even over de situatie tot die tijd: hoe autonomer systemen worden, hoe lastiger je taak als bestuurder wordt omdat je alleen de moeilijke taken hoeft over te nemen. Als de makkelijke dingen worden overgenomen, kan het lastig zijn op het moment dat je moet reageren op een moeilijk moment.

D: dat is een psychologisch aspect waar ik bekend mee ben. Dat is een kwestie van overload en underload. Je hebt een bepaalde hoeveelheid mentale belasting die je aankan, waaronder je het beste presteert. Als je weinig te doen hebt, zit je in een underload staat. Dan gaat je prestatie ook achteruit. Het is te saai, je hebt te weinig te doen. Dan raak je out of the loop, wat er met een automatisch rijdend systeem ook gebeurt. Als je ineens teveel te doen hebt, kom je dus in een overload staat, waardoor je ook slechter gaat presteren. Je moet dus altijd eigenlijk op zoek gaan naar de juiste balans. Je moet altijd net genoeg te doen hebben, ook in een automatische auto. In een gewone auto heb je de dagelijkse rijtaken zoals sturen, kijken, gas, remmen, enzo. Dan heb je net genoeg te doen, ben je net actief genoeg, je blijft in de loop omdat je genoeg te doen hebt. Als je bijv. naar het zuiden van Frankrijk gaat met die eindeloos lange snelwegen, dan zit je voet op 1 plek, je stuur op 1 plek, en er gebeurt voor de rest niks. Dan krijg je eigenlijk hetzelfde fenomeen. Dat noemen ze dan highway blindness, waarbij je de hele tijd hetzelfde ziet en op een gegeven moment zie je de verandering niet meer. En dan kan het zomaar zijn dat er opeens wel iets verandert, zoals een auto die voor je stilstaat, en dat zie je dan niet, dus dan knal je er bovenop. Mailtjes beantwoorden, of voorbereiden van presentaties, meetings even doorlezen. Dat soort dingen zou je dan moeten kunnen doen. Of als je aan truckers denkt: die kunnen dan vast hun verplichte twee uur slapen terwijl de auto gewoon doorrijdt. Als je dus ervoor wilt zorgen dat ze ten allen tijden genoeg te doen hebben, dan kunnen ze dat soort dingen niet meer, want ze moeten altijd in de loop blijven. Het is de vraag of je wil dat mensen altijd in de loop blijven. Want dan sla je dus de plank mis. Dan kan je ze net zo goed gewoon handmatig laten doorrijden. Dus ik vraag me zelf af of we die kant wel op moeten gaan.

Je kan bestuurders over een snelweg laten rijden, en elke keer dat ze onder een brug door rijden op een knopje laten drukken. Dat was een van mijn experimenten. Op een gegeven moment gaan ze dat slechter doen. Je kan ze ook bijvoorbeeld een cognitieve taak kunnen laten doen, zodat ze ook mentaal aanwezig zijn, en niet alleen visueel. Bijvoorbeeld een n-back task. Een lastige taak, zeker na langere tijd. Dan blijf je cognitief bezig. Je blijft met je hersenen bezig. Maar zodra je dan andere dingen moet doen, dan gaat dat niet samen. Je kan dus ook niet bijvoorbeeld je laptop openen en mailtjes beantwoorden. Er zijn wel ideeen voor, maar of dat nou de ideale optie is, is de vraag.

Het zou andersom moeten zijn. Eerst de moeilijke taken zodat mensen de ruimte hebben om andere dingen te kunnen doen. En langzaamaan, als de technologie beter wordt, kan alles worden afgevangen zodat de mens niks meer hoeft te doen. Die tussenstadia zitten heel veel haken en ogen aan.

In de vliegtuigindustrie werkt het ook zo. Het vliegen in de lucht is volledig geautomatiseerd. Alleen het stijgen en landen wordt gedaan door de piloot, wat nou juist de moeilijke taken zijn. Ik denk dat het in elk systeem zo werkt. De simpele dingen worden overgenomen. In veel industrietakken is dat natuurlijk geweldig, voor lopendebandwerk. Maar voor dit soort dingen is het wat lastiger.

A.1.9. Transcript D.N. Twilhaar

Note: During this interview, the recording device failed to save the recording of the interview. Therefore, only the written notes are included in this transcript.

About:

- Educational background: Control Engineering.
- Current profession: board member of the Safety Leaders Foundation. "The Foundation wants to acquire, develop, keep current, test and apply knowledge and technology that has the purpose to improve safety, health and environmental aspects of professional work activities" [175].

Notes

Alle technieken in de regeltechniek, zoals bode-diagrammen, gaan uit van lineare systemen. Er bestaat niets om te bewijzen dat niet-lineaire systemen stabiel zijn. Dit vormt een probleem wanneer veiligheid gegarandeerd moet worden.

Rule of thumb: in goed geteste software zit gemiddeld 1 fout per 1000 regels. Wat je hebt gebouwd wijkt altijd af van wat je had willen bouwen. Andere overweging (statistical detection/measurement theory): trade-off tussen detectie-kant en vals-alarm. Voorbeeld: het kan niet dat de brandweer altijd moet uitrukken terwijl een groot deel van de meldingen vals alarm blijkt te zijn. Meer geld investeren in detectieapparatuur heeft in het algemeen als resultaat dat de vals-alarm-kans omhoog gaat, waardoor de veiligheid niet groter wordt. Voor elk nieuw systeem geldt: het kan misgaan. En: installeren kost geld, geld is werk, werk is risico. Neem het vorobeeld van het gebruik van offshore helicopters: deze zijn niet super betrouwbaar. Als oplossing worden twee monitoren gebruikt, maar er is alsnog een groot risico. Als hij neerstort, dan wordt er in ieder geval getraind hoe mensen eruit moeten. Maar al twee keer zijn er mensen omgekomen. Nul risico bestaat niet. Er is een optimaal risico.

Je kunt een softwaresysteem niet volledig testen. Er is altijd een kwetsbaarheid voor kwaadwillenden. Neem de stuxnet bug in Iran. Een systeem moet onderhouden worden, bijvoorbeeld door middel van verbinding met internet, of door usb. Deze toegang tot system vormt een grootschalig gevaar. Dit zijn risico's van digitale regeltechniek.

Je kunt wel met digitale regelsystemen werken, maar onder speciale omstandigheden:

1. Naast de digitale regelaar heb je simpele systemen die maar 1 parameter bewaken (bijv. druk). Hierbij wordt een operating envelope dus onafhankelijk bewaakt met een simpel systeem.
2. De taken die tegenwoordig door de mens worden uitgevoerd. De mens is faalbaar. Als systemen dit verminderen, wordt het veiliger, de maatschappelijke acceptatiekant daargelaten. Als er bijvoorbeeld in een huidige opstelling geen bewaking is, is een systeem niet gek. Maar wees realistisch. Streven naar 100% veiligheid is een groot risico. Probeer niet alles te detecteren. Er moet vooraf open over getallen gecommuniceerd worden. De vals-alarm-kans moet duidelijk gecommuniceerd worden.

Vals alarm en detectie is gekoppeld. Daar is niet aan te ontkomen. Wat is een redelijke vals alarm kans? Veel valse alarmen zorgen voor een normalisering van risico's. Bijvoorbeeld het geval waarbij een sticker over een waarschuwingslampje geplakt wordt dat onnodig vaak gaat knipperen. Het resultaat hiervan: de reactie in geval van een echte calamiteit wordt minder goed. Daar zit een trade-off in.

Vals positieven worden een steeds groter probleem. Alles moet gedetecteerd worden. Om dit in de hand te houden, moet je risico's laten controleren door een apart systeem. Veiligheid moet niet afhangen van autonome systemen. Tenzij de veiligheid gewaarborgd wordt door onafhankelijke subsystemen.

Door het maken van een safety-case wil je bewijzen dat de situatie veilig is, maar software kun je niet bewijzen. Dat is onmogelijk. Met zoveel parameters kun je veiligheid niet bewijzen. Hoe bewijs je dat een chauffeur geschikt is voor zijn taak? Wat je wel kan doen, is de gevolgen van een fout beter bestrijden. Neem bijvoorbeeld een autonoom remsysteem. Dat kan na veel testen nog steeds een stuk karton aanzien voor iets

anders. Maar als de auto remt op basis van deze fout, kan dat een ongeluk veroorzaken.

Hoe zou je een systeem kunnen testen? Dat kan niet. Het aantal mogelijke permutaties aan de input-kant is te groot. Neem je bijvoorbeeld ook waarden van gisteren of eergisteren mee? De rekenkracht om alles mee te nemen is niet beschikbaar. Kijk naar operating-software in veelgebruikte telefoons of computers. Daar zitten nog steeds bugs in, ookal wordt het gebruikt door miljarden mensen. Dus je kunt in je case niet bewijzen dat je de risico's onder controle hebt.

B

Raw Code

In [2]:

```
%matplotlib inline
import random

import numpy as np
import matplotlib.pyplot as plt

import matplotlib.font_manager
from sklearn import svm
from sklearn.svm import OneClassSVM
from sklearn.neighbors import NearestNeighbors
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neighbors import LocalOutlierFactor

from sklearn.preprocessing import LabelBinarizer
from sklearn.preprocessing import label_binarize
from sklearn.model_selection import StratifiedShuffleSplit, GridSearchCV
```

Create dataset

In [500]:

```
np.random.seed(333)
w = np.random.dirichlet((1,1,1,1,1))/20
```

In [501]:

```
def gating(start,prob):
    z=np.random.uniform()
    if (z < prob):
        if (start==0):
            # initial value is zero
            transform = 1
        else:
            transform = 0
    else:
        if (start==1):
            # initial value is one
            transform = 1
        else:
            transform = 0
    return transform

#lengths per type of train
types_length = [10,13,20]

def generate_trains(count, types_length):
    np.random.seed(333)
    data = {}
    observ = []

    for x in range(count):
        #number of wagons
        wagon_count = round(np.random.uniform(1,20))

        #type of wagon (0,1,2,3,...)
        wagons_type = np.random.randint(1, len(types_length)+1,size=wagon_count)

        #length of train
        wagons_length = np.copy(wagons_type)

        for t in range(len(types_length)):
            wagons_length[wagons_type==(t+1)] = types_length[t]

        #create small (natural) deviation in train lengths
        sd = 0.2
        wagons_length_normal = np.random.normal(wagons_length, sd)
        train_length = sum(wagons_length_normal)

        #hazardous goods (y/n)
```

```
        #hazardous goods (y/n)
        wagons_haz = np.copy(wagons_type)
        for i in range(len(wagons_type)):
            if wagons_type[i] == 1:
                wagons_haz[i] = np.random.binomial(1,0.3)
            else:
                wagons_haz[i] = 0

        #store results
        train_nr = 'T' + str(x)

        groundtruth = {}
        groundtruth['wagon_count'] = wagon_count
        groundtruth['wagons_type'] = wagons_type
        groundtruth['wagons_length'] = wagons_length_normal
        groundtruth['train_length'] = train_length
        groundtruth['wagons_haz'] = wagons_haz

        data[train_nr] = {}
        data[train_nr]['groundtruth'] = groundtruth


        ## CREATE ANOMALIES
        anomaly = False
        node = np.zeros(7)

        for y in range(5):
            if (y == 0):
                z=np.random.uniform()
                if (z < w[0]):
                    # initial value is one
                    node[0] = 1

                    wagon_count = round(np.random.normal(wagon_count, 10))
                    anomaly = True

                else:
                    node[0] = 0

            if (y==1):
                node[y]= gating(node[0],w[y])
                if node[y] == 1:
                    wagons_length_normal = np.random.normal(wagons_length_normal, 10)
                    anomaly = True

            if (y==2):
                node[y]= gating(node[0],w[y])
                if node[y] == 1:
                    #wagons_type = np.random.normal(wagons_type,0.2).astype(int) #incorrect, range
too low
                    random.shuffle(wagons_type)
                    anomaly = True

            if (y==3):
                node[y]= gating(node[1],w[y])
                if node[y] == 1:
                    train_length = np.random.normal(train_length, 100)
                    anomaly = True

            if (y==4):
                node[y]= gating(node[2],w[y])
                if node[y] == 1:
                    #wagons_haz = np.random.normal(wagons_haz, 0.5).astype(int)
                    random.shuffle(wagons_haz)
                    anomaly = True



    #    node = censor(c2,node)
    #    print(node)
        observ.append(node)

        dist = {}
        dist['wagon_count'] = wagon_count
        dist['wagons_type'] = wagons_type
        dist['wagons_length'] = wagons_length_normal
        dist['train_length'] = train_length
        dist['wagons_haz'] = wagons_haz
```

```
        dist['wagons_naz'] = wagons_naz
        dist['anomaly'] = anomaly

        data[train_nr]['dist'] = dist

    observ = np.array(observ)
    return data
```

In [631]:

```
def ratios(y_true, y_pred, method='SVM'):
    # type y_true is a boolean array
    # True : inlier
    # False : outlier

    # type y_pred is a -1, 1 array where:
    #  1 : inlier
    # -1 : outlier

    # convert y_pred to boolean array
    y_pred = y_pred==1

    total = len(y_true)/100
    TP = sum([not a and not b for a, b in zip(y_pred, y_true)])/total
    FP = sum([not a and b for a, b in zip(y_pred, y_true)])/total
    TN = sum([a and b for a, b in zip(y_pred, y_true)])/total
    FN = sum([a and not b for a, b in zip(y_pred, y_true)])/total

    anomaly_rate = (len(y_true)-sum(y_true))/len(y_true)
    TPR = TP/(TP+FN)
    FPR = FP/(FP+TN)
    TNR = 1-FPR
    FNR = 1-TPR

#    print("TP: " + str(TP))
    return anomaly_rate, TP, FP, TN, FN
```

Other functions

In [637]:

```
def print_ratios(y_true, y_pred, method='SVM'):
    anomaly_rate, TP, FP, TN, FN = ratios(y_true, y_pred, method)

    print("Method: " + str(method) + "\n"
          "\t Anomaly rate:         " + str("%.4f" % anomaly_rate) + "\n \n" +
          "\t True positives:  " + str("%.4f" % TP) + "\n"
          "\t True negatives:  " + str("%.4f" % TN) + "\n"
          "\t False positives: " + str("%.4f" % FP) + "\n"
          "\t False negatives: " + str("%.4f" % FN) + "\n"
         )

def nrm(x, sigma = 0.005):
    y = np.random.normal(x, sigma)
    return y

def reconstruct(flat_values,prototype):
    length_arrays = [len(s) for s in prototype]
    reconstruction = []
    for length in length_arrays:
        reconstruction.append(flat_values[:length].astype(int))
    return reconstruction

def iter_anomaly(list_of_arrays):
    anomaly_list = []
    for i in list_of_arrays:
        if -1 in i:
            anomaly_list.append(True)
        else:
            anomaly_list.append(False)
    anomaly_list = np.asarray(anomaly_list).reshape(-1,1)
    return anomaly_list
```

```
def svm_fun(data, anomalies, gamma=60, nu=0.04):
    estimator = svm.OneClassSVM(kernel='rbf', gamma=gamma, nu=nu)
    estimator.fit(data)
    y_pred = estimator.predict(data)
    print_ratios(anomalies, y_pred, 'SVM')

def lof_fun(data, anomalies, n_neighbors=35):
    neigh_estimator = LocalOutlierFactor(n_neighbors=n_neighbors)
    neigh_estimator.fit(data)
    y_pred = neigh_estimator.fit_predict(data)
    print_ratios(anomalies, y_pred, "LocalOutlierFactor")
```

In [440]:

```
def parse_data(data):
    lengths = []
    wagons = []
    anomalies = []
    wagons_types = []
    haz_perwagon = []
    wagons_lengths = []

    for i in data:
        lengths.append(data[i]['dist']['train_length'])
        wagons.append(data[i]['dist']['wagon_count'])
        anomalies.append(data[i]['dist']['anomaly'])
        wagons_types.append(data[i]['dist']['wagons_type'])
        haz_perwagon.append(data[i]['dist']['wagons_haz'])
        wagons_lengths.append(data[i]['dist']['wagons_length'])

    return lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths
```

In [441]:

```
from matplotlib.colors import Normalize
from sklearn.metrics import roc_auc_score, f1_score, fbeta_score
from sklearn.metrics import accuracy_score as accuracy

def SVM_optimal_params(X, anomalies, train_ratio, nu_range, gamma_range, visualise=True, metric='ro
c_auc_score'):

    clf = OneClassSVM()
    results_dic = {}


    if train_ratio<1:
        train_x = X[:int(train_ratio*len(X))]
        test_x  = X[int(train_ratio*len(X)):]
        y_true = np.invert(np.asarray(anomalies).reshape(-1,1))[int(train_ratio*len(X)):]

    else:
        train_x = X
        test_x  = X
        y_true = np.invert(np.asarray(anomalies).reshape(-1,1))


    #initialize timer
    pt = progress_timer(description= 'Progress', n_iter=len(gamma_range)*len(nu_range)-1)

    for gamma in gamma_range:
        for nu in nu_range:
            clf.set_params(gamma=gamma, nu=nu)

            clf.fit(train_x)

            y_pred = clf.predict(test_x)
            FP, FN, TP, TN = classification(y_true, y_pred)

            results_dic[(gamma, nu)] = {}
            results_dic[(gamma, nu)]['TPR'] = TPR(y_true, y_pred)
            results_dic[(gamma, nu)]['FPR'] = FPR(y_true, y_pred)
            results_dic[(gamma, nu)]['TNR'] = TNR(y_true, y_pred)
            results_dic[(gamma, nu)]['FNR'] = FNR(y_true, y_pred)
            results_dic[(gamma, nu)]['TP'] =  TP
```

```
            results_dic[(gamma, nu)]['FP'] =  FP
            results_dic[(gamma, nu)]['TN'] =  TN
            results_dic[(gamma, nu)]['FN'] =  FN
            results_dic[(gamma, nu)]['roc_auc_score'] = roc_auc_score(y_true, y_pred)
            results_dic[(gamma, nu)]['f1_score'] = f1_score(y_true, y_pred)
            results_dic[(gamma, nu)]['f05_score'] = fbeta_score(y_true, y_pred,0.5)
            results_dic[(gamma, nu)]['f2_score'] = fbeta_score(y_true, y_pred,2)
            results_dic[(gamma, nu)]['accuracy'] = accuracy(y_true, y_pred)

            pt.update()

    pt.finish()

    return results_dic


def LOF_optimal_params(X, anomalies, n_neighbors, contamination, metric='roc_auc_score'):

    clf = LocalOutlierFactor()
    results_dic = {}
    y_true = np.invert(np.asarray(anomalies).reshape(-1,1))

    #initialize timer
    pt = progress_timer(description= 'Progress', n_iter=len(n_neighbors)*len(contamination)-1)

    for n in n_neighbors:
        for c in contamination:
            clf.set_params(n_neighbors=n, contamination=c)

            y_pred = clf.fit_predict(X)
            FP, FN, TP, TN = classification(y_true, y_pred)

            results_dic[(n, c)] = {}
            results_dic[(n, c)]['TPR'] = TPR(y_true, y_pred)
            results_dic[(n, c)]['FPR'] = FPR(y_true, y_pred)
            results_dic[(n, c)]['TNR'] = TNR(y_true, y_pred)
            results_dic[(n, c)]['FNR'] = FNR(y_true, y_pred)
            results_dic[(n, c)]['TP'] =  TP
            results_dic[(n, c)]['FP'] =  FP
            results_dic[(n, c)]['TN'] =  TN
            results_dic[(n, c)]['FN'] =  FN
            results_dic[(n, c)]['roc_auc_score'] = roc_auc_score(y_true, y_pred)
            results_dic[(n, c)]['f1_score'] = f1_score(y_true, y_pred)
            results_dic[(n, c)]['f05_score'] = fbeta_score(y_true, y_pred,0.5)
            results_dic[(n, c)]['f2_score'] = fbeta_score(y_true, y_pred,2)
            results_dic[(n, c)]['accuracy'] = accuracy(y_true, y_pred)



            pt.update()

    pt.finish()

    return results_dic

def heatmap_params(results_dic,metric,gamma_range, nu_range, filename="heatmapSVM"):
    ##############################################################################
    # Visualization
    #
    # Draw heatmap of the validation accuracy as a function of gamma and nu

    # Utility function to move the midpoint of a colormap to be around
    # the values of interest.

    class MidpointNormalize(Normalize):

        def __init__(self, vmin=None, vmax=None, midpoint=None, clip=False):
            self.midpoint = midpoint
            Normalize.__init__(self, vmin, vmax, clip)

        def __call__(self, value, clip=None):
            x, y = [self.vmin, self.midpoint, self.vmax], [0, 0.5, 1]
            return np.ma.masked_array(np.interp(value, x, y))

    #################

    rocauc_lst = []
```

```
    for key in results_dic.keys():
        rocauc_lst.append(results_dic[key][metric])

    scores = np.asarray(rocauc_lst).reshape(len(nu_range), len(gamma_range),  order='F')


    f = plt.figure(figsize=(10, 5.5))
    plt.subplots_adjust(left=.2, right=0.95, bottom=0.15, top=0.95)
    plt.imshow(scores, interpolation='nearest', cmap=plt.cm.hot,
                       norm=MidpointNormalize(vmin=0.5, midpoint=0.7))


    plt.xlabel('Gamma')
    plt.ylabel('Nu')
    cbar = plt.colorbar()
    cbar.set_label('ROC-AUC score', rotation=90)
    plt.xticks(np.arange(len(gamma_range)), gamma_range.astype(int), rotation=90)
    plt.yticks(np.arange(len(nu_range)),    np.round(nu_range, decimals=3))
#    plt.title(str(metric) + 'for one-class SVM')
    plt.show()
    f.savefig(str(filename) + ".pdf", bbox_inches='tight')

def heatmap_params_LOF(results_dic,metric,n_neighbors_range,contamination_range, filename='plot'):

    ##############################################################################
    # Visualization
    #
    # Draw heatmap of the validation accuracy as a function of gamma and nu

    # Utility function to move the midpoint of a colormap to be around
    # the values of interest.

    class MidpointNormalize(Normalize):

        def __init__(self, vmin=None, vmax=None, midpoint=None, clip=False):
            self.midpoint = midpoint
            Normalize.__init__(self, vmin, vmax, clip)

        def __call__(self, value, clip=None):
            x, y = [self.vmin, self.midpoint, self.vmax], [0, 0.5, 1]
            return np.ma.masked_array(np.interp(value, x, y))

    ##################

    rocauc_lst = []
    for key in results_dic.keys():
        rocauc_lst.append(results_dic[key][metric])

    scores = np.asarray(rocauc_lst).reshape(len(contamination_range), len(n_neighbors_range),  orde
r='F')


    f = plt.figure(figsize=(10, 5.5))
    plt.subplots_adjust(left=.2, right=0.95, bottom=0.15, top=0.95)
    plt.imshow(scores, interpolation='nearest', cmap=plt.cm.hot,
                       norm=MidpointNormalize(vmin=0.5, midpoint=0.7))


    plt.xlabel('Number of neighbors')
    plt.ylabel('Contamination')
    cbar = plt.colorbar()
    cbar.set_label('ROC-AUC score', rotation=90)
    plt.xticks(np.arange(len(n_neighbors_range)), np.round(n_neighbors_range, decimals=0), rotation
=90)
    plt.yticks(np.arange(len(contamination_range)),    np.round(contamination_range, decimals=3))
#    plt.title(str(metric) + 'for LOF')
    f.savefig(str(filename) + ".pdf", bbox_inches='tight')
    plt.show()
```

In [442]:

```
#https://www.themarketingtechnologist.co/progress-timer-in-python/

#import libraries
import progressbar as pb
```

```
#define progress timer class
class progress_timer:

    def __init__(self, n_iter, description="Something"):
        self.n_iter         = n_iter
        self.iter           = 0
        self.description    = description + ': '
        self.timer          = None
        self.initialize()

    def initialize(self):
        #initialize timer
        widgets = [self.description, pb.Percentage(), ' ',
                   pb.Bar(marker=pb.RotatingMarker()), ' ', pb.ETA()]
        self.timer = pb.ProgressBar(widgets=widgets, maxval=self.n_iter).start()

    def update(self, q=1):
        #update timer
        self.timer.update(self.iter)
        self.iter += q

    def finish(self):
        #end timer
        self.timer.finish()
```

In [443]:

```
def classification(y_true, y_predic):
    # type y_true is a boolean array
    # True : inlier
    # False : outlier

    # type y_pred is a -1, 1 array where:
    #  1 : inlier
    # -1 : outlier

    # convert y_pred to boolean array
    y_predic = y_predic==1


    TP = sum([not a and not b for a, b in zip(y_predic, y_true)])
    FP = sum([not a and     b for a, b in zip(y_predic, y_true)])
    TN = sum([    a and     b for a, b in zip(y_predic, y_true)])
    FN = sum([    a and not b for a, b in zip(y_predic, y_true)])

    return FP, FN, TP, TN

def TPR(y_true, y_prediction):
    FP, FN, TP, TN = classification(y_true, y_prediction)
    TPR = TP/(TP+FN)
    return TPR

def FPR(y_true, y_prediction):
    FP, FN, TP, TN = classification(y_true, y_prediction)
    FPR = FP/(FP+TN)
    return FPR

def TNR(y_true, y_prediction):
    FP, FN, TP, TN = classification(y_true, y_prediction)
    TNR = TN/(FP+TN)
    return TNR

def FNR(y_true, y_prediction):
    FP, FN, TP, TN = classification(y_true, y_prediction)
    FNR = FN/(TP+FN)
    return FNR
```

In [444]:

```
from operator import itemgetter
from functional import compose
from matplotlib.ticker import FormatStrFormatter
import progressbar as pb
```

```
def best_parameters(results_dic,metric):
    key_max = max(results_dic.keys(), key=(lambda k: results_dic[k][metric]))
    best_params = results_dic[key_max]

    gamma = key_max[0]
    nu = key_max[1]

    r = (best_params['FP']+best_params['FN']+best_params['TP']+best_params['TN'])/100

    print('\nThe best parameters are (gamma: ' + str("%.4f" % gamma) + ', nu: ' + str("%.4f" % nu)
+ ')')
    print('True positives: ' +  str(best_params['TP']/r))
    print('False negatives: ' + str(best_params['FN']/r))
    print('True negatives: ' +  str(best_params['TN']/r))
    print('False positives: ' + str(best_params['FP']/r))


    print(metric + ': ' + str("%.2f" % best_params[metric]))



    return (gamma, nu)
```

In [445]:

```
from sklearn.metrics import roc_auc_score
```

In [446]:

```
def split_data(X, anomalies, train_ratio):
    train_x = X[:int(train_ratio*len(X))]
    test_x  = X[int(train_ratio*len(X)):]
    y_true = np.invert(np.asarray(anomalies).reshape(-1,1))[int(train_ratio*len(X)):]

    return train_x, test_x, y_true
```

In [447]:

```
import warnings
warnings.filterwarnings("ignore")
```

One-class SVM train length - wagon count

-Comparison of methods
-Comparison of metrics
-Comparison of number of samples

Classifier comparison

In [569]:

```
data = generate_trains(count=5000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data)
```

In [570]:

```
nu_range = np.logspace(-2, -0, 10)
#gamma_range = np.logspace(-2, 4, 7)
gamma_range = np.logspace(-1,3,10)
```

In [ ]:

```

```

In [571]:

```
x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

Method 1: SVM

In [572]:

```
results_SVM = SVM_optimal_params(X=X, anomalies=anomalies,
                  train_ratio=train_ratio,
                  nu_range=nu_range, gamma_range=gamma_range)
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 0:00:25
```

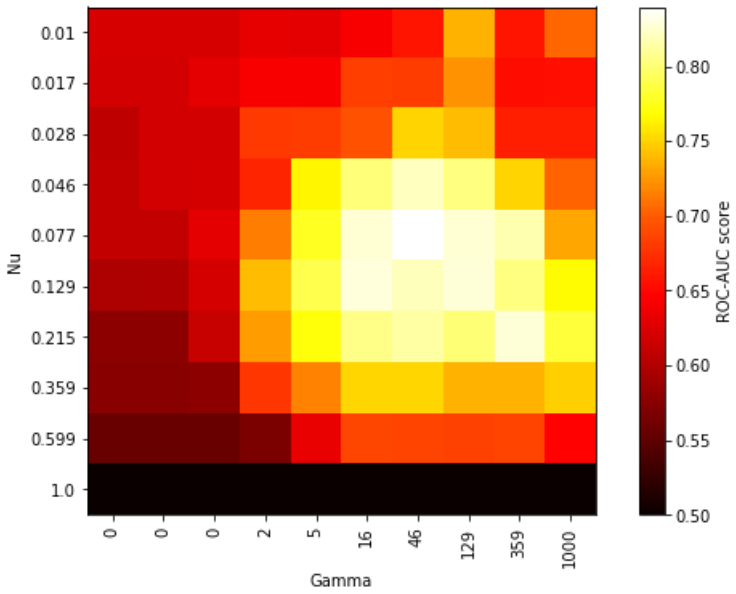
In [573]:

```
best_params = best_parameters(results_SVM,metric='roc_auc_score')
```

```
The best parameters are (gamma: 46.4159, nu: 0.0774)
True positives: [3.5]
False negatives: [1.4]
True negatives: [91.8]
False positives: [3.3]
roc_auc_score: 0.84
```

In [574]:

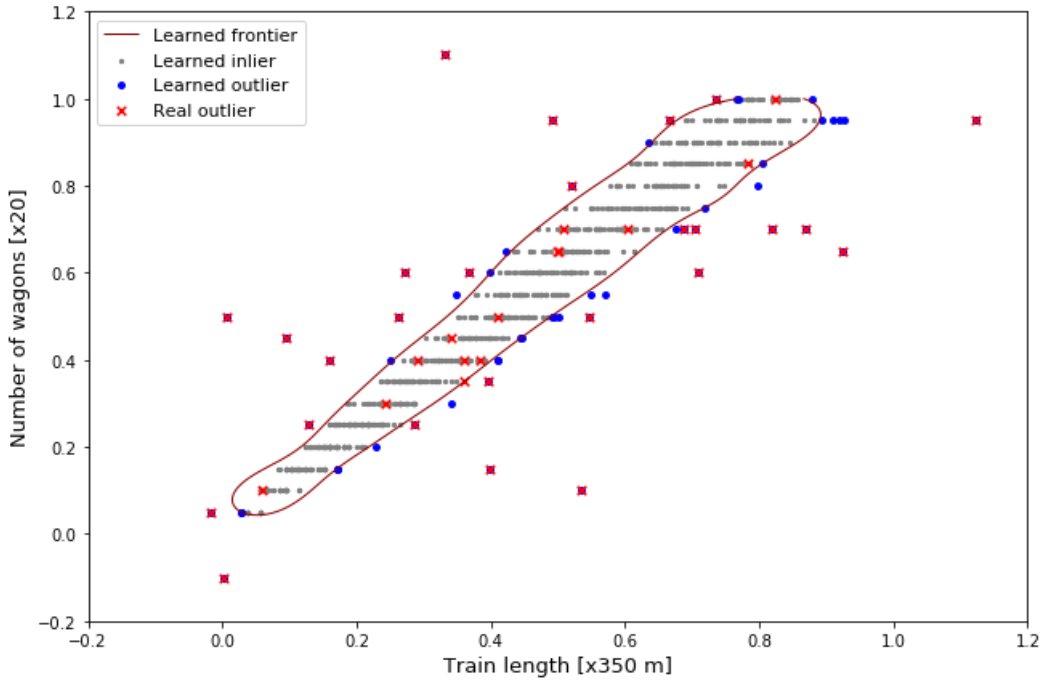
```
heatmap_params(results_SVM, metric='roc_auc_score',gamma_range=gamma_range,nu_range=nu_range,filena
me="class_comp_SVM_heatmap")
```



In [575]:

```
#plot
X_train, X_test, y_true = split_data(X, anomalies, train_ratio)
f = plt.figure(figsize=(10.5,7))

estimator = svm.OneClassSVM(kernel='rbf', gamma=best_params[0], nu=best_params[1])
estimator.fit(X_train)
y_pred = estimator.predict(X_test)


xx, yy = np.meshgrid(np.linspace(0, 1, 200), np.linspace(0, 1, 200))
# plot the line, the points, and the nearest vectors to the plane
Z = estimator.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
```

```
Z = Z.reshape(xx.shape)
l = plt.contour(xx, yy, Z, levels=[0], linewidths=1, colors='darkred')


a = plt.scatter(X_test[:,[0]],
                X_test[:,[1]],s=5,c='grey')
b = plt.scatter(X_test[:,[0]][y_pred==-1],
                X_test[:,[1]][y_pred==-1], c='blue',s=15)
c = plt.scatter(X_test[:,[0]][y_true==False],
                X_test[:,[1]][y_true==False], c='red', marker='x',s=30)

plt.legend([l.collections[0], a, b, c],
           ["Learned frontier", "Learned inlier", "Learned outlier",
            "Real outlier"],
           loc="upper left",
           prop=matplotlib.font_manager.FontProperties(size=11))


#plt.title('One-class SVM to detect anomalies in train length vs number of wagons
(scaled)',fontsize=18)
plt.xlabel('Train length [x350 m]',fontsize=13)
plt.ylabel('Number of wagons [x20]',fontsize=13)

axes = plt.gca()
axes.set_xlim([-.2,1.2])
axes.set_ylim([-.2,1.2])

f.savefig("class_comp_SVM_scatter.pdf", bbox_inches='tight')

plt.show()

#print_ratios(y_true, y_pred, method='SVM')
```



Method 2: LOF

In [576]:

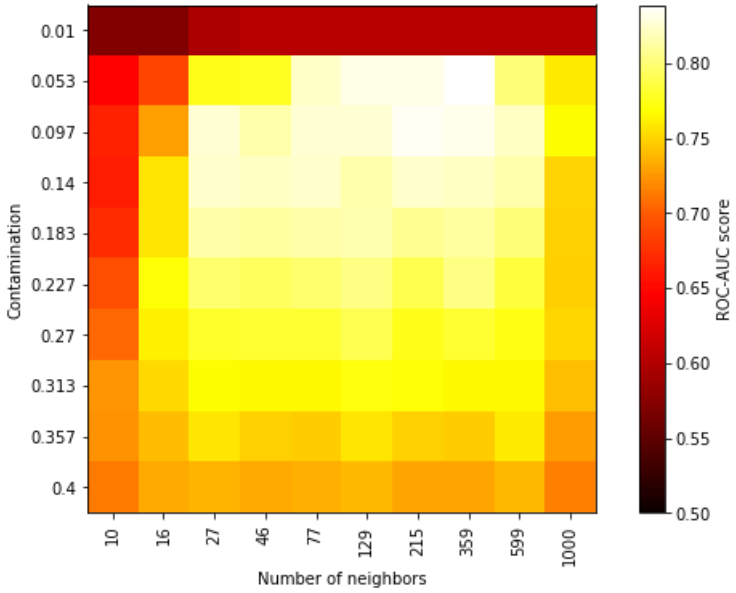
```
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))

n_neighbors_range = np.logspace(1,3,10).astype(int)
#n_neighbors_range = np.linspace(5,150,19).astype(int)
contamination_range = np.linspace(0.01,0.4,10)
results_LOF = LOF_optimal_params(X=X, anomalies=anomalies,
                  n_neighbors=n_neighbors_range, contamination=contamination_range)

best_params = best_parameters(results_LOF,metric='roc_auc_score')
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 0:00:45
```

```
The best parameters are (gamma: 359.0000, nu: 0.0533)
True positives: [3.38]
False negatives: [1.46]
True negatives: [93.2]
False positives: [1.96]
roc_auc_score: 0.84
```

In [577]:

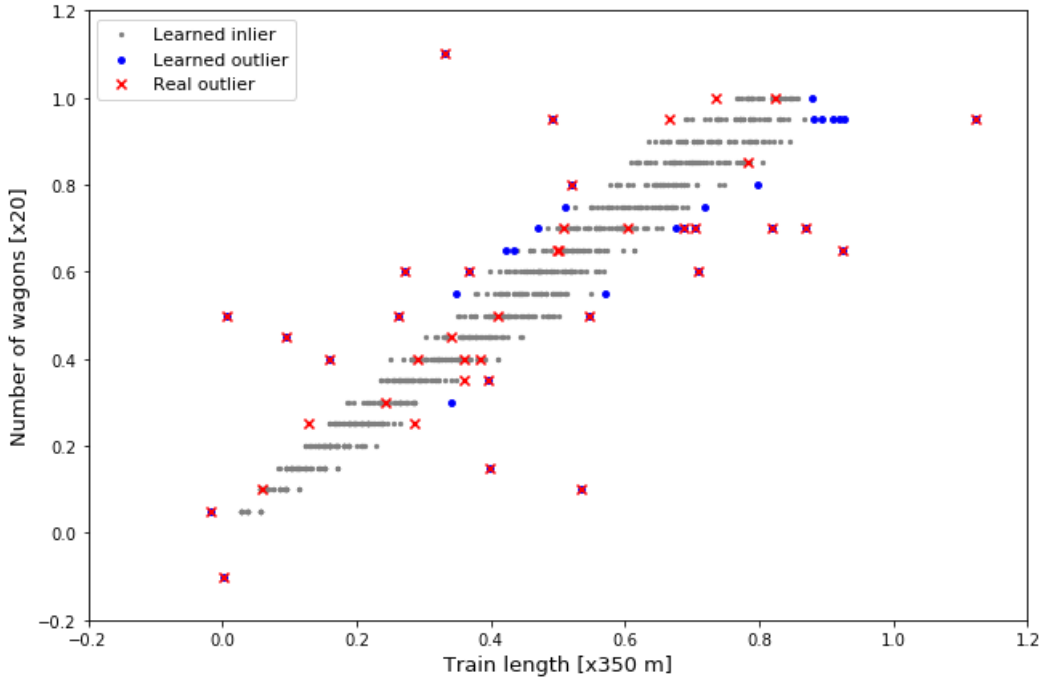
```
heatmap_params_LOF(results_dic=results_LOF, n_neighbors_range=n_neighbors_range,

contamination_range=contamination_range,metric='roc_auc_score',filename='class_comp_LOF_heatmap')
```



In [580]:

```
#plot
#X_train, X_test, y_true = split_data(X, anomalies, train_ratio)
f = plt.figure(figsize=(10.5,7))

estimator = LocalOutlierFactor(n_neighbors=best_params[0], contamination=best_params[1])
y_pred = estimator.fit_predict(X)[4000:]
X_s = X[4000:]
y_true_s = y_true[4000:]
xx, yy = np.meshgrid(np.linspace(0, 1, 200), np.linspace(0, 1, 200))
# plot the line, the points, and the nearest vectors to the plane
Z = estimator._decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
l = plt.contour(xx, yy, Z, levels=[0], linewidths=1, colors='darkred')


a = plt.scatter(X_s[:, [0]],
                X_s[:, [1]],s=5,c='grey')
b = plt.scatter(X_s[:, [0]][y_pred==-1],
                X_s[:, [1]][y_pred==-1], c='blue',s=15)
c = plt.scatter(X_s[:, [0]][y_true_s==False],
                X_s[:, [1]][y_true_s==False], c='red', marker='x')

plt.legend([a, b, c],
           ["Learned inlier", "Learned outlier",
            "Real outlier"],
           loc="upper left",
           prop=matplotlib.font_manager.FontProperties(size=11))


# plt.title('One-class SVM to detect anomalies in train length vs number of wagons
(scaled)',fontsize=18)
plt.xlabel('Train length [x350 m]',fontsize=13)
plt.ylabel('Number of wagons [x20]',fontsize=13)
```

```
axes = plt.gca()
axes.set_xlim([-.2,1.2])
axes.set_ylim([-.2,1.2])

f.savefig("class_comp_LOF_scatter.pdf", bbox_inches='tight')

plt.show()

#print_ratios(y_true, y_pred, method='LOF')
```



Comparison of metrics

In [581]:

```
best_params_rocauc = best_parameters(results_SVM,    metric='roc_auc_score')
best_params_f1 = best_parameters(results_SVM,        metric='f1_score')
best_params_f05 = best_parameters(results_SVM,        metric='f05_score')
best_params_f2 = best_parameters(results_SVM,        metric='f2_score')
best_params_accuracy = best_parameters(results_SVM, metric='accuracy')
```

```
The best parameters are (gamma: 46.4159, nu: 0.0774)
True positives: [3.5]
False negatives: [1.4]
True negatives: [91.8]
False positives: [3.3]
roc_auc_score: 0.84

The best parameters are (gamma: 46.4159, nu: 0.0464)
True positives: [3.2]
False negatives: [1.7]
True negatives: [94.2]
False positives: [0.9]
f1_score: 0.99

The best parameters are (gamma: 46.4159, nu: 0.0464)
True positives: [3.2]
False negatives: [1.7]
True negatives: [94.2]
False positives: [0.9]
f05_score: 0.98

The best parameters are (gamma: 16.6810, nu: 0.0167)
True positives: [1.8]
False negatives: [3.1]
True negatives: [95.]
False positives: [0.1]
f2_score: 0.99
```

```
The best parameters are (gamma: 0.7743, nu: 0.0100)
True positives: [1.2]
False negatives: [3.7]
True negatives: [95.1]
False positives: [0.]
accuracy: 0.95
```

Variation dataset sample size

In [ ]:

```
# Generate 100.000 trains
```

In [661]:

```
data_large = generate_trains(count=100000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data_large)

x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

In [663]:

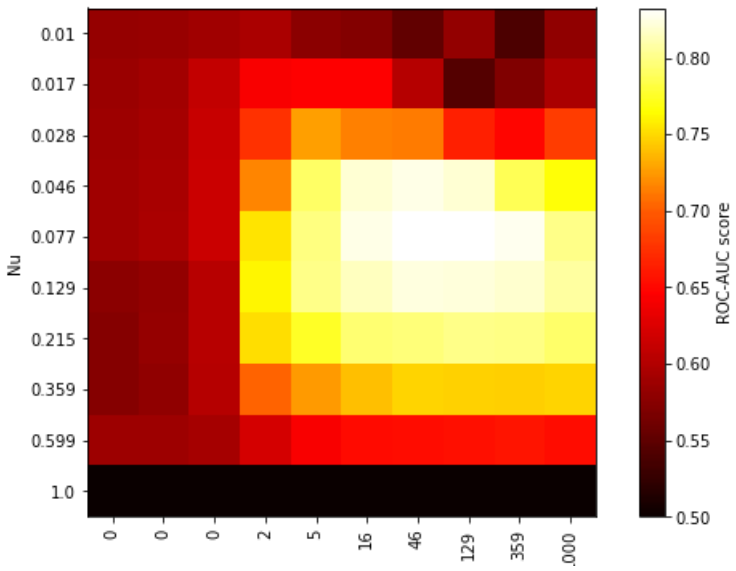
```
results_SVM_big = SVM_optimal_params(X=X, anomalies=anomalies,
                  train_ratio=train_ratio,
                  nu_range=nu_range, gamma_range=gamma_range)
best_params = best_parameters(results_SVM_big,metric='roc_auc_score')
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| Time: 2:54:25
```

```
The best parameters are (gamma: 46.4159, nu: 0.0774)
True positives: [3.47]
False negatives: [1.41]
True negatives: [90.785]
False positives: [4.335]
roc_auc_score: 0.83
```

In [664]:

```
heatmap_params(results_SVM_big,
               metric='roc_auc_score',
               gamma_range=gamma_range,nu_range=nu_range,
               filename="ss_comp_large_heatmap")
```



Gamma

In [665]:

```
np.save('results_SVM_big.npy', results_SVM_big)
```

In [666]:

```
# # Load
# read_dictionary = np.load('my_file.npy').item()
# print(read_dictionary['hello']) # displays "world"
```

In [667]:

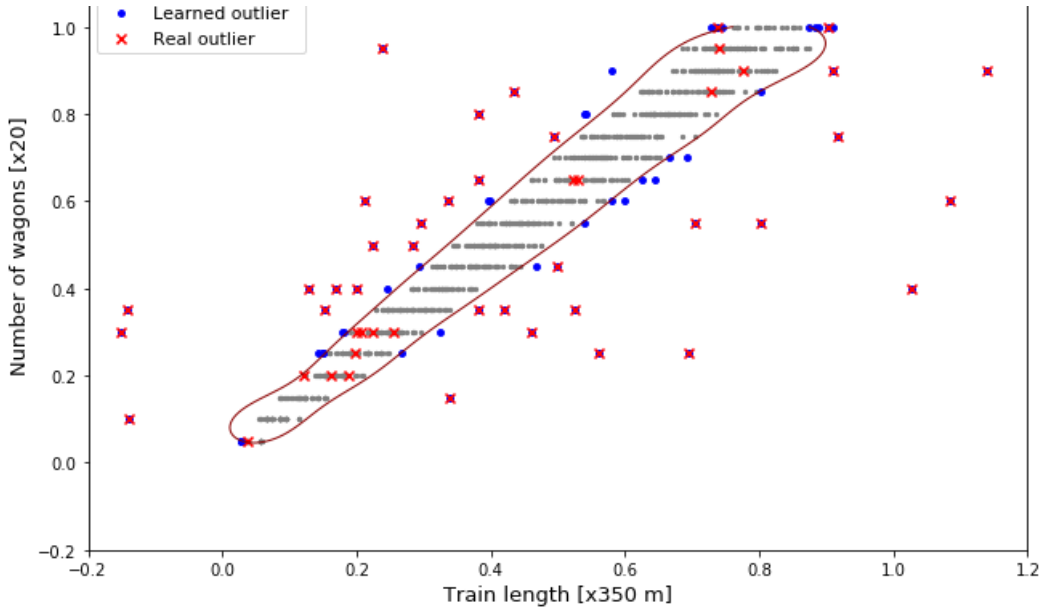
```
#heatmap_params(results_SVM_big, metric='roc_auc_score')
```

In [668]:

```
#prepare data for plotting
X_train, X_test, y_true = split_data(X, anomalies, train_ratio)

estimator = svm.OneClassSVM(kernel='rbf', gamma=best_params[0], nu=best_params[1])
estimator.fit(X_train)
y_pred = estimator.predict(X_test)[:1000]
X_s = X_test[:1000]
y_true_s = y_true[:1000]
```

In [669]:

```
#plot
plt.figure(figsize=(10.5,7))

xx, yy = np.meshgrid(np.linspace(0, 1, 200), np.linspace(0, 1, 200))
# plot the line, the points, and the nearest vectors to the plane
Z = estimator.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
l = plt.contour(xx, yy, Z, levels=[0], linewidths=1, colors='darkred')


a = plt.scatter(X_s[:,[0]],
                X_s[:,[1]],s=5,c='grey')
b = plt.scatter(X_s[:,[0]][y_pred==-1],
                X_s[:,[1]][y_pred==-1], c='blue',s=15)
c = plt.scatter(X_s[:,[0]][y_true_s==False],
                X_s[:,[1]][y_true_s==False], c='red', marker='x')

plt.legend([l.collections[0], a, b, c],
           ["Learned frontier", "Learned inlier", "Learned outlier",
            "Real outlier"],
           loc="upper left",
           prop=matplotlib.font_manager.FontProperties(size=11))


#plt.title('One-class SVM to detect anomalies in train length vs number of wagons
(scaled)',fontsize=18)
plt.xlabel('Train length [x350 m]',fontsize=13)
plt.ylabel('Number of wagons [x20]',fontsize=13)

axes = plt.gca()
axes.set_xlim([-.2,1.2])
axes.set_ylim([-.2,1.2])

f.savefig("ss_comp_large_scatter.pdf", bbox_inches='tight')

plt.show()

#print_ratios(y_true, y_pred, method='SVM')
```



Variation number of parameters in dataset

In [605]:

```
data = generate_trains(count=5000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data)

x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

Add wagon type (A, B, C) to the large dataset

In [606]:

```
def fill_zeros(data):
    # Fill list or train type arrays with zeros to be of same length
    data_filled = np.zeros([len(data),len(max(data,key = lambda x: len(x)))])
    for i,j in enumerate(data):
        data_filled[i][0:len(j)] = j
    return data_filled
```

In [617]:

```
# Fill list or train type arrays with zeros to be of same length
#types_filled = np.zeros([len(wagons_types),len(max(wagons_types,key = lambda x: len(x)))])
#for i,j in enumerate(wagons_types):
#    types_filled[i][0:len(j)] = j

types_filled = fill_zeros(wagons_types)
```

In [616]:

```
#convert list of types to integer list
types = list(range(len(types_length)+1))

# Binarize wagon types to use for classifiers
types_binarized = []
for i in range(len(types_filled)):
    a = label_binarize(types_filled[i], classes=types)
    types_binarized.append(np.concatenate(a).ravel())

# Add binarized types to data set
X1 = np.append(X,types_binarized,axis=1)
```

Check performance of SVM

In [588]:

```
results_SVM_wagontypes = SVM_optimal_params(X=X1, anomalies=anomalies,
                  train_ratio=train_ratio,
                  nu_range=nu_range, gamma_range=gamma_range)
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 0:03:39
```

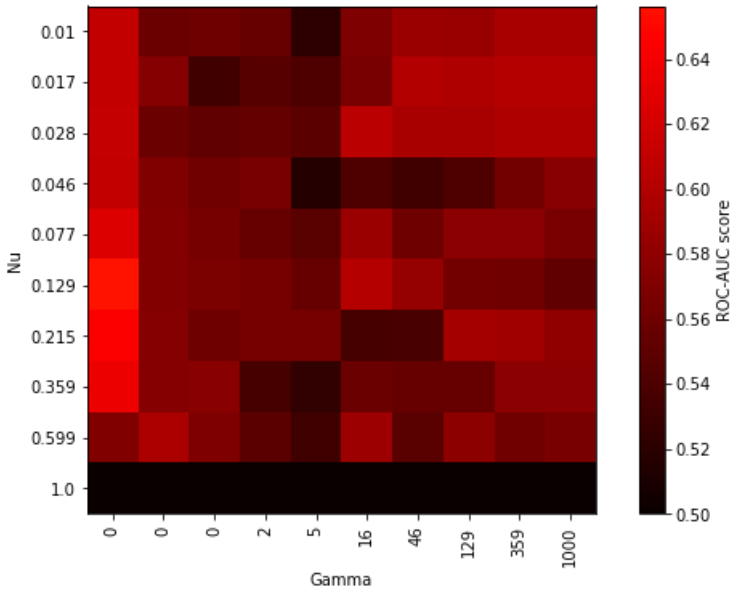
In [589]:

```
best_params = best_parameters(results_SVM_wagontypes,metric='roc_auc_score')
```

```
The best parameters are (gamma: 0.1000, nu: 0.1292)
True positives: [2.8]
False negatives: [2.1]
True negatives: [70.5]
False positives: [24.6]
roc_auc_score: 0.66
```

In [590]:

```
heatmap_params(results_SVM_wagontypes, metric='roc_auc_score',
               gamma_range=gamma_range, nu_range=nu_range,
               filename="ft_comp_wagontypes_heatmap")
```



In [245]:

```
results_LOF_wagontypes = LOF_optimal_params(X=X1, anomalies=anomalies,
                  n_neighbors=n_neighbors_range, contamination=contamination_range)
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 0:20:56
```

In [249]:

```
best_params_LOF = best_parameters(results_LOF_wagontypes, metric='roc_auc_score')
```

```
The best parameters are (gamma: 193.0000, nu: 0.1493)
True positives: [0.96]
False negatives: [3.88]
True negatives: [81.18]
False positives: [13.98]
roc_auc_score: 0.53
```

Add hazardous goods boolean

In [618]:

```
haz_filled = fill_zeros(haz_perwagon)
haz_filled[:5]

X2 = np.append(X,haz_filled, axis=1)
```

In [592]:

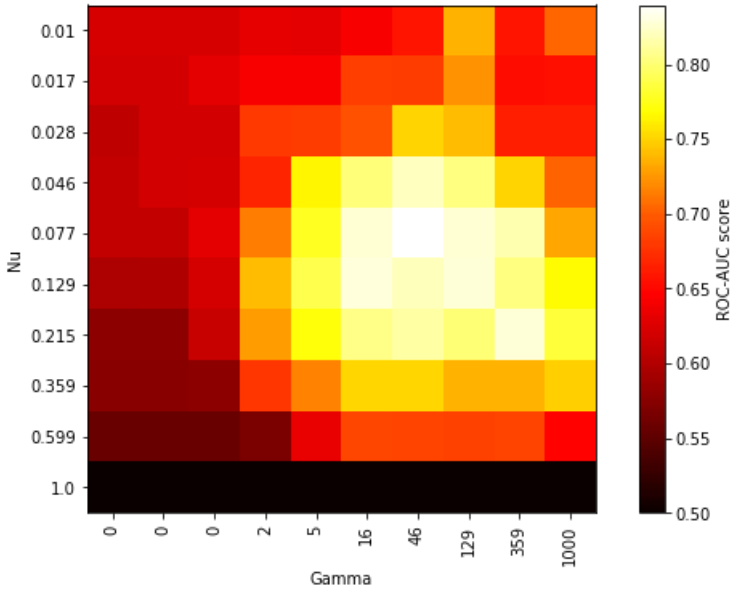
```
results_SVM_haz = SVM_optimal_params(X=X2, anomalies=anomalies,
                  train_ratio=train_ratio,
                  nu_range=nu_range, gamma_range=gamma_range)

best_params = best_parameters(results_SVM_haz,metric='roc_auc_score')
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 0:00:54
```

```
The best parameters are (gamma: 46.4159, nu: 0.3594)
True positives: [4.1]
False negatives: [0.8]
True negatives: [60.7]
False positives: [34.4]
roc_auc_score: 0.74
```

In [593]:

```
heatmap_params(results_SVM, metric='roc_auc_score',
               gamma_range=gamma_range, nu_range=nu_range,
               filename="ft_comp_hazgoods_heatmap")
```



Same experiment, more samples

In [594]:

```
data_large = generate_trains(count=20000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data_large)

x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

In [595]:

```
haz_filled = fill_zeros(haz_perwagon)
haz_filled[:5]

X2_20k = np.append(X,haz_filled, axis=1)
```

In [597]:

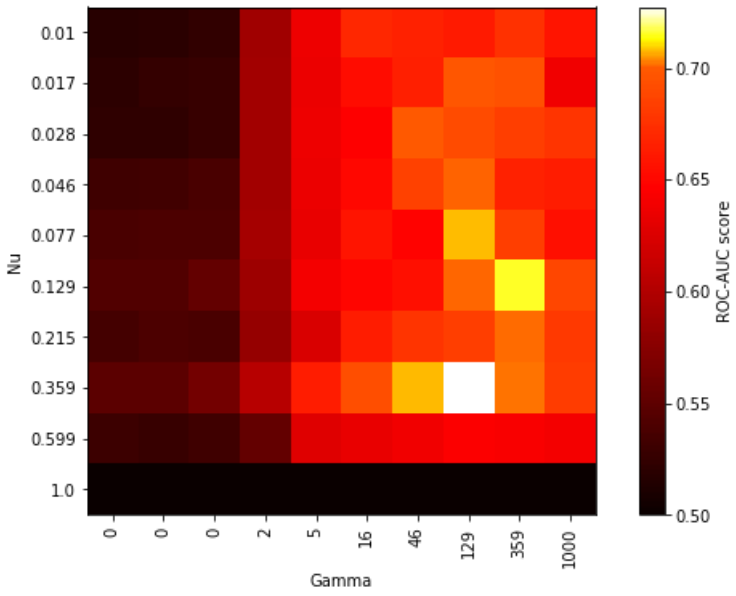
```
results_SVM_20k = SVM_optimal_params(X=X2_20k, anomalies=anomalies,
                  train_ratio=train_ratio,
                  nu_range=nu_range, gamma_range=gamma_range)

best_params = best_parameters(results_SVM_20k,metric='roc_auc_score')
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 1:16:49
```

```
The best parameters are (gamma: 129.1550, nu: 0.3594)
True positives: [4.15]
False negatives: [0.9]
True negatives: [60.025]
False positives: [34.925]
roc_auc_score: 0.73
```

In [598]:

```
heatmap_params(results_SVM_20k, metric='roc_auc_score',
               gamma_range=gamma_range, nu_range=nu_range,
               filename="ft_comp_hazgoods20k_heatmap")
```



Add individual wagon lengths

In [639]:

```
data = generate_trains(count=5000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data)

x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

In [640]:

```
wagons_lengths_filled = fill_zeros(wagons_lengths)/(wagons_lengths_filled.max())
X3 = np.append(X,wagons_lengths_filled, axis=1)
```

In [641]:

```
results_SVM = SVM_optimal_params(X=X3, anomalies=anomalies,
                  train_ratio=train_ratio,
                  nu_range=nu_range, gamma_range=gamma_range)

best_params = best_parameters(results_SVM,metric='roc_auc_score')
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 0:01:31
```

```
The best parameters are (gamma: 0.1000, nu: 0.0167)
True positives: [4.7]
False negatives: [0.2]
True negatives: [17.1]
False positives: [78.]
roc_auc_score: 0.57
```

Add individual wagon lengths + 20,000 samples

In [ ]:

```
data = generate_trains(count=20000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data)

x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

In [621]:

```
wagons_lengths_filled = fill_zeros(wagons_lengths)/(wagons_lengths_filled.max())
```

In [622]:

```
X3 = np.append(X,wagons_lengths_filled, axis=1)
```

In [601]:

```
results_SVM = SVM_optimal_params(X=X3, anomalies=anomalies,
                  train_ratio=train_ratio,
                  nu_range=nu_range, gamma_range=gamma_range)

best_params = best_parameters(results_SVM,metric='roc_auc_score')
#heatmap_params(results_SVM, metric='roc_auc_score')
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 2:44:41
```

```
The best parameters are (gamma: 5.9948, nu: 0.0774)
True positives: [4.125]
False negatives: [0.925]
True negatives: [90.325]
False positives: [4.625]
roc_auc_score: 0.88
```

Add individual wagon lengths + 100,000 samples

In [670]:

```
data = generate_trains(count=100000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data)
```

```
x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
wagons_lengths_filled = fill_zeros(wagons_lengths)/(wagons_lengths_filled.max())
X3 = np.append(X,wagons_lengths_filled, axis=1)
```

In [671]:

```
results_SVM = SVM_optimal_params(X=X3, anomalies=anomalies,
                    train_ratio=train_ratio,
                    nu_range=nu_range, gamma_range=gamma_range)

best_params = best_parameters(results_SVM,metric='roc_auc_score')
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 17:13:24
```

```
The best parameters are (gamma: 5.9948, nu: 0.0774)
True positives: [3.78]
False negatives: [1.1]
True negatives: [91.225]
False positives: [3.895]
roc_auc_score: 0.87
```

Combine all features

In [643]:

```
data = generate_trains(count=5000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data)

x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

In [623]:

```
X1_all = np.append(X,      types_binarized,       axis=1)
X2_all = np.append(X1_all, haz_filled,            axis=1)
X3_all = np.append(X2_all, wagons_lengths_filled, axis=1)
```

In [624]:

```
results_SVM_all = SVM_optimal_params(X=X3_all, anomalies=anomalies,
                    train_ratio=train_ratio,
                    nu_range=nu_range, gamma_range=gamma_range)
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 0:06:59
```

In [625]:

```
best_params_all = best_parameters(results_SVM_all, metric='roc_auc_score')
```

```
The best parameters are (gamma: 0.1000, nu: 0.0100)
True positives: [4.7]
False negatives: [0.2]
True negatives: [13.5]
False positives: [81.6]
roc_auc_score: 0.55
```

Higher sample size

In [645]:

```
data_large = generate_trains(count=20000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data_large)

x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

In [627]:

```
types_filled = fill_zeros(wagons_types)

#convert list of types to integer list
types = list(range(len(types_length)+1))

# Binarize wagon types to use for classifiers
types_binarized = []
for i in range(len(types_filled)):
    a = label_binarize(types_filled[i], classes=types)
    types_binarized.append(np.concatenate(a).ravel())

# Add binarized types to data set
X1_l = np.append(X,types_binarized,axis=1)
```

In [628]:

```
haz_filled = fill_zeros(haz_perwagon)
haz_filled[:5]

X2_l = np.append(X1_l,haz_filled, axis=1)
```

In [629]:

```
wagons_lengths_filled = fill_zeros(wagons_lengths)/(wagons_lengths_filled.max())

X3_l = np.append(X2_l,wagons_lengths_filled, axis=1)
```

In [630]:

```
results_SVM_all_l = SVM_optimal_params(X=X3_l, anomalies=anomalies,
                   train_ratio=train_ratio,
                   nu_range=nu_range, gamma_range=gamma_range)
```

```
Progress: 100% |||||||||||||||||||||||||||||||||||||||||||||||||| Time: 3:24:08
```

In [632]:

```
best_params_SVM_all_l = best_parameters(results_SVM_all_l, metric='roc_auc_score')
```

```
The best parameters are (gamma: 0.2783, nu: 0.0100)
True positives: [3.1]
False negatives: [1.95]
True negatives: [61.65]
False positives: [33.3]
roc_auc_score: 0.63
```

Rule based

In [652]:

```
data_large = generate_trains(count=5000, types_length=types_length)
lengths, wagons, anomalies, wagons_types, haz_perwagon, wagons_lengths = parse_data(data_large)

x1 = np.asarray(lengths).reshape(-1,1)/350
x2 = np.asarray(wagons).reshape(-1,1)/20
```

```
X = np.append(x1,x2, axis=1)
y_true = np.invert(np.asarray(anomalies).reshape(-1,1))
train_ratio=0.8
```

In [653]:

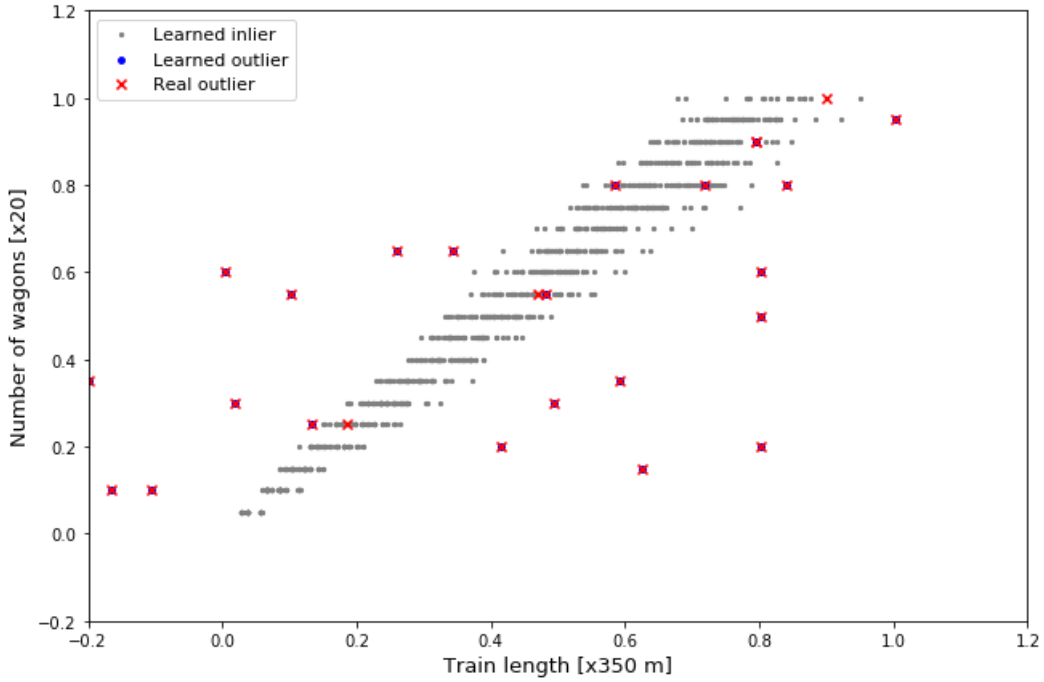
```
#check if hazardous goods are registered to the right wagon
haz_anom = []
for i in range(len(haz_perwagon)):
    #a returns true when a wagon that is not equal to type 1 still contains hazgoods
    a = np.logical_and(haz_perwagon[i]==1, wagons_types[i]!=1)
    haz_anom.append(a.any())


#check if wagon lengths make sense
wagonlength_anom = []
for i in range(len(wagons_lengths)):
    #a returns true when a wagon is shorter than 8m or longer than 21m
    a = np.logical_or(wagons_lengths[i]<9, wagons_lengths[i]>21)
    wagonlength_anom.append(a.any())

#check if train lengths make sense
trainlength_anom = []
for i in range(len(lengths)):
    #a returns true when a train is shorter than 8m*wagon_count or longer than 21m*wagon_count
    a = np.logical_or((wagons[i]*9)>lengths[i], wagons[i]*21<lengths[i])
    trainlength_anom.append(a.any())


comb1 = np.logical_or(haz_anom,wagonlength_anom)
comb2 = np.logical_or(comb1, trainlength_anom)

print(roc_auc_score(y_true,np.invert(comb2)))
print_ratios(y_true,np.invert(comb2))
```

```
0.9132231404958678
Method: SVM
  Anomaly rate:        0.0484

  True positives:  4.0000
  True negatives:  95.1600
  False positives: 0.0000
  False negatives: 0.8400
```

In [660]:

```
#plot
f = plt.figure(figsize=(10.5,7))

y_pred = np.invert(comb2)[:1000]
X_s = X[:1000]
y_true_s = y_true[:1000]
xx, yy = np.meshgrid(np.linspace(0, 1, 200), np.linspace(0, 1, 200))

a = plt.scatter(X_s[:, [0]],
                X_s[:, [1]],s=5,c='grey')
b = plt.scatter(X_s[:, [0]][y_pred==False],
                X_s[:, [1]][y_pred==False], c='blue',s=15)
c = plt.scatter(X_s[:, [0]][y_true_s==False],
                X_s[:, [1]][y_true_s==False], c='red', marker='x')

plt.legend([a, b, c],
           ["Learned inlier", "Learned outlier",
            "Real outlier"],
           loc="upper left",
           prop=matplotlib.font_manager.FontProperties(size=11))


# plt.title('One-class SVM to detect anomalies in train length vs number of wagons
(scaled)',fontsize=18)
plt.xlabel('Train length [x350 m]',fontsize=13)
plt.ylabel('Number of wagons [x20]',fontsize=13)

axes = plt.gca()
```

```
axes.set_xlim([-.2,1.2])
axes.set_ylim([-.2,1.2])

f.savefig("rulebased_comp_scatter.pdf", bbox_inches='tight')

plt.show()

#print_ratios(y_true, y_pred, method='LOF')
```



Bibliography

[1] MSc Engineering and Policy Analysis. URL https://www.tudelft.nl/onderwijs/opleidingen/masters/epa/msc-engineering-and-policy-analysis/.

[2] Kdnuggets. URL https://www.kdnuggets.com/2017/07/rapidminer-ai-machine-learning-deep-learning.html.

[3] Safety. URL https://www.merriam-webster.com/dictionary/safety.

[4] Meaningful Human Control over Automated Driving Systems (MHC-ADS). URL https://www.tudelft.nl/citg/over-faculteit/afdelingen/transport-planning/research/projects/mhc-ads/.

[5] Overfitting | Definition of overfitting in English by Oxford Dictionaries. URL https://en.oxforddictionaries.com/definition/overfitting.

[6] ISO/Guide 73:2009(en), 2009. URL https://www.iso.org/obp/ui/#iso:std:iso:guide:73:ed-1:v1:en.

[7] Worldwide semiannual artificial intelligence systems spending guide. *IDC: Analyze the Future*, 2016. URL https://www.idc.com/getdoc.jsp?containerId=IDC_P33198.

[8] Rail Insider-National Transportation Safety Board: BNSF trains collided after one train missed a red light, Jul 2016. URL https://www.progressiverailroading.com/bnsf_railway/article/NTSB-BNSF-trains-collided-after-one-train-missed-a-red-light--48799.

[9] Underfitting and overfitting in machine learning, Nov 2017. URL https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/.

[10] The cost of malware containment. 2017. URL https://www.ponemon.org/local/upload/file/Damballa%20Malware%20Containment%20FINAL%203.pdf.

[11] False alarm: looking back on chemical depot scare, Jan 2018. URL http://www.hermistonherald.com/hh/local-news/20180116/false-alarm-looking-back-on-chemical-depot-scare.

[12] IEC SC 65A. Functional safety of electrical/electronic/programmable electronic safety-related systems. Technical Report IEC 61508, The International Electrotechnical Commission, 1998.

[13] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[14] Mennatallah Amer, Markus Goldstein, and Slim Abdennadher. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, pages 8–15. ACM, 2013.

[15] William RL Anderegg, Elizabeth S Callaway, Maxwell T Boykoff, Gary Yohe, and Terr y L Root. Awareness of both type 1 and 2 errors in climate science and assessment. *Bulletin of the American Meteorological Society*, 95(9):1445–1451, 2014.

[16] James M Anderson, Kalra Nidhi, Karlyn D Stanley, Paul Sorensen, Constantine Samaras, and Oluwatobi A Oluwatola. *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation, 2014.

[17] Paul A Anderson. Decision making by objection and the cuban missile crisis. *Administrative Science Quarterly*, pages 201–222, 1983.

[18] R Ashmore and E Lennon. Progress towards the assurance of non-traditional software. In *Developments in System Safety Engineering, Proceedings of the Twenty-fifth Safety-Critical Systems Symposium, Bristol, UK, ISBN*, pages 978–1540796288, 2017.

[19] Josh Attenberg, Panagiotis G Ipeirotis, and Foster J Provost. Beat the machine: Challenging workers to find the unknown unknowns. *Human Computation*, 11(11):2–7, 2011.

[20] Martin Atzmueller, Joachim Baumeister, Mario Goller, and Frank Puppe. A datagenerator for evaluating machine learning methods. *KI*, 20(3):57–63, 2006.

[21] Suvojit Bagchi. Workers heard cracking noise, Apr 2016. URL `http://www.thehindu.com/news/cities/kolkata/workers-heard-cracking-noise/article8423585.ece`.

[22] Haitham Baomar and Peter J Bentley. An intelligent autopilot system that learns flight emergency procedures by imitating human pilots. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–9. IEEE, 2016.

[23] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999.

[24] Johannes Baumbach, André Marais, and D Gonçalves. Losing the boxes: fragmentation as a source of system complexity. In *Proc. of the 11th SA INCOSE Conference*, 2015.

[25] Michael Beaney. Susan stebbing on cambridge and vienna analysis. In *The Vienna circle and Logical Empiricism*, pages 339–350. Springer, 2003.

[26] Karen Bennett. Conceptual analysis and its limits. *Philosophic Exchange*, 46(1), Oct 2017. URL `https://digitalcommons.brockport.edu/phil_ex/vol46/iss1/1/`.

[27] Mark Berman and Brian Fung. Hawaii's false missile alert sent by troubled worker who thought an attack was imminent, officials say, Jan 2018. URL `https://www.washingtonpost.com/news/the-switch/wp/2018/01/30/heres-what-went-wrong-with-that-hawaii-missile-alert-the-fcc-says/?noredirect=on&utm_term=.2e6be69289ca`.

[28] Anurag Bhardwaj, Wei Di, and Jianing Wei. *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing Ltd, 2018.

[29] Siddhartha Bhattacharyya, Darren Cofer, D Musliner, Joseph Mueller, and Eric Engstrom. Certification considerations for adaptive systems. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 270–279. IEEE, 2015.

[30] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

[31] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[32] Tobias Bolch, Anil Kulkarni, Andreas Kääb, Christian Huggel, Frank Paul, JG Cogley, Holger Frey, Jeffrey S Kargel, Koji Fujita, Marlene Scheel, et al. The state and fate of himalayan glaciers. *Science*, 336 (6079):310–314, 2012.

[33] V Bolón Canedo, Beatriz Remeseiro López, A Alonso Betanzos, and Aurélio Campilho. Machine learning for medical applications. 2016.

[34] Martin Bouchard. *Social networks, terrorism and counter-terrorism: Radical and connected*. Routledge, 2015.

[35] Leo Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[36] Edward Broughton. The bhopal disaster and its aftermath: a review. *Environmental Health*, 4(1):6, 2005.

[37] Ryan Browne. More than $60 million worth of bitcoin potentially stolen after hack on cryptocurrency site, Dec 2017. URL https://www.cnbc.com/2017/12/07/bitcoin-stolen-in-hack-on-nicehash-cryptocurrency-mining-marketplace.html.

[38] Erik Brynjolfsson and Andrew McAfee. *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company, 2014.

[39] Keynyn Brysse, Naomi Oreskes, Jessica O'Reilly, and Michael Oppenheimer. Climate change prediction: Erring on the side of least drama? *Global environmental change*, 23(1):327–337, 2013.

[40] Aleksandar Chakarov, Aditya Nori, Sriram Rajamani, Shayak Sen, and Deepak Vijaykeerthy. Debugging machine learning tasks. *arXiv preprint arXiv:1603.07292*, 2016.

[41] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[42] Syantani Chatterjee. Train engineer was texting just before california crash, Oct 2008. URL https://www.reuters.com/article/us-usa-train-crash/train-engineer-was-texting-just-before-california-crash-idUSN0152835520081002.

[43] Corey Chivers. How likely is the NSA PRISM program to catch a terrorist?, Jan 2015. URL https://bayesianbiologist.com/2013/06/06/how-likely-is-the-nsa-prism-program-to-catch-a-terrorist/.

[44] Sabarathinam Chockalingam, Wolter Pieters, Andre Teixeira, Nima Khakzad, and Pieter van Gelder. Combining bayesian networks and fishbone diagrams to distinguish between intentional attacks and accidental technical failures. *TU Delft Repository*, Jul 2018.

[45] Chris. Machine learning interview questions – q4 – explain how a roc curve works, Jul 2017. URL http://machinelearningspecialist.com/machine-learning-interview-questions-q4-explain-how-a-roc-curve-works/.

[46] Bartek Ciszewski. 7 challenges for machine learning projects, Aug 2018. URL https://www.netguru.co/blog/7-challenges-for-machine-learning-projects.

[47] Louis Columbus. 10 charts that will change your perspective on artificial intelligence's growth, Jan 2018. URL https://www.forbes.com/sites/louiscolumbus/2018/01/12/10-charts-that-will-change-your-perspective-on-artificial-intelligences-growth/#3a2aa4954758.

[48] Robert G Cooper. Stage-gate systems: a new tool for managing new products. *Business horizons*, 33(3): 44–54, 1990.

[49] Robert G Cooper. Winning at new products: Accelerating the process from idea to launch. 2001.

[50] Michael Copeland. The difference between AI, Machine Learning, and Deep Learning? | NVIDIA Blog, Jun 2016. URL https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/.

[51] Paul Craig and Gráinne De Búrca. *EU law: text, cases, and materials*. Oxford University Press, 2011.

[52] Zhihuang Dai, Michael J Scott, and Zissimos P Mourelatos. Incorporating epistemic uncertainty in robust design. In *ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 85–95. American Society of Mechanical Engineers, 2003.

[53] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.

[54] Jason P Davis, Kathleen M Eisenhardt, and Christopher B Bingham. Developing theory through simulation methods. *Academy of Management Review*, 32(2):480–499, 2007.

[55] Matthew DeBord. The 3 biggest risks facing self-driving cars, Jun 2017. URL https://www.businessinsider.nl/self-driving-cars-risks-2017-6/?international=true&r=US.

[56] S.A.M. Dijksma. *Vervoer gevaarlijke stoffen*. 2017.

[57] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[58] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. 2017.

[59] Stephan Dreiseitl, Melanie Osl, Christian Scheibböck, and Michael Binder. Outlier detection with one-class svms: an application to melanoma prognosis. In *AMIA Annual Symposium Proceedings*, volume 2010, page 172. American Medical Informatics Association, 2010.

[60] Vojtech EKSLER. Intermediate report on the development of railway safety in the 4 European Union. *European Railway Agency. Safety Unit*, 5, 2013.

[61] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639): 115, 2017.

[62] Bureau d'Enquêtes et d'Analyses. Final report on the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro–Paris. *Paris: BEA*, 2012.

[63] José M Faria. Non-determinism and failure modes in machine learning. In *Software Reliability Engineering Workshops (ISSREW), 2017 IEEE International Symposium on*, pages 310–316. IEEE, 2017.

[64] José M Faria. Machine learning safety: An overview. 2018.

[65] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[66] T Ferrell. Engineering safety-critical systems in the 21st century. In *IEEE Central Virginia Section, Engineers Week Dinner Meeting. Charlottesville, VA*, 2010.

[67] International Organization for Standardization. *ISO 31000: 2018: Risk Management - Guidelines*. International Organization for Standardization, 2018.

[68] Carl Benedikt Frey and Michael A Osborne. The future of employment: how susceptible are jobs to computerisation? *Technological Forecasting and Social Change*, 114:254–280, 2017.

[69] Brian Fung. The technology behind the tesla crash, explained, Jul 2016. URL `https://www.washingtonpost.com/news/the-switch/wp/2016/07/01/the-technology-behind-the-tesla-crash-explained/?utm_term=.e1359c9bab2e`.

[70] Gemalto. Findings from the 2017 breach level index. https://breachlevelindex.com/assets/Breach-Level-Index-Report-2017-Gemalto.pdf, 2017.

[71] Suzanne Goldenberg. Gulf oil spill: firms ignored warning signs before blast, inquiry hears, May 2010. URL `https://www.theguardian.com/environment/2010/may/12/deepwater-gulf-oil-spill-hearing`.

[72] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[73] Lee Gomes. Hidden obstacles for google's self-driving cars: Impressive progress hides major limitations of Google's quest for automated driving. *Univ. Parma, Parma, Italy, Tech. Rep*, 2014.

[74] Anna Gomez. Deep learning in digital pathology, Feb 2018. URL `http://www.global-engage.com/life-science/deep-learning-in-digital-pathology/`.

[75] Bryce Goodman and Seth Flaxman. European Union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

[76] Nathan A Greenblatt. Self-driving cars and the law. *IEEE Spectrum*, 53(2):46–51, 2016.

[77] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[78] Safety Guide. IAEA safety standards series - software for computer based systems important to safety in nuclear power plants. *Structure*, 1:2, 2000.

[79] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[80] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[81] Daniel Heikoop. Linkedin page. URL `https://www.linkedin.com/in/daniel-heikoop-8bb13167/`.

[82] Edward Helmore. Crack in Florida bridge deemed no concern just hours before collapse, Mar 2018. URL `https://www.theguardian.com/us-news/2018/mar/17/florida-bridge-collapse-safety-meeting-crack-no-concerns`.

[83] Phil Hodkinson and Heather Hodkinson. The strengths and limitations of case study research. In *learning and skills development agency conference at Cambridge*, volume 1, pages 5–7, 2001.

[84] Ole R Holsti. Content analysis for the social sciences and humanities. *Reading. MA: Addison-Wesley (content analysis)*, 1969.

[85] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58. ACM, 2011.

[86] Kit Huckvale, Samanta Adomaviciute, José Tomás Prieto, Melvin Khee-Shing Leow, and Josip Car. Smartphone apps for calculating insulin dose: a systematic assessment. *BMC medicine*, 13(1):106, 2015.

[87] Gheorghe Ilie and Carmen Nadia Ciocoiu. Application of fishbone diagram to determine the risk of an event with multiple causes. *Management Research and Practice*, 2(1):1–20, 2010.

[88] Abhaya Indrayan and Rajeev Kumar Malhotra. *Medical biostatistics*. Chapman and Hall/CRC, 2017.

[89] C IPC and AR4 Climate Change. Synthesis report, a. *Allali, et al., Editors*, pages 23–73, 2007.

[90] Kaoru Ishikawa. *Guide to quality control: industrial engineering and technology*. Asian Productivity Organization Tokyo, Japan, 1976.

[91] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*, 2014.

[92] Paul E Johnson. Simulation modeling in political science. *American Behavioral Scientist*, 42(10):1509–1530, 1999.

[93] Nidhi Kalra. Challenges and approaches to realizing autonomous vehicle safety. 2017.

[94] Shubir Kapoor, Aleksandra Mojsilovic, Jade Nguyen Strattner, and Kush R Varshney. From open data ecosystems to systems of innovation: A journey to realize the promise of open data. In *Bloomberg Data for Good Exchange Conference*, 2015.

[95] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[96] Jonathan Kay. How do you regulate a self-improving algorithm?, Oct 2017. URL `https://www.theatlantic.com/technology/archive/2017/10/algorithms-future-of-health-care/543825/`.

[97] Roozbeh Kianfar, Paolo Falcone, and Jonas Fredriksson. Safety verification of automated driving systems. *IEEE Intelligent Transportation Systems Magazine*, 5(4):73–86, 2013.

[98] AW Kimball. Errors of the third kind in statistical consulting. *Journal of the American Statistical Association*, 52(278):133–142, 1957.

[99] Philip Koopman and Michael Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017.

[100] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[101] Jennifer Elston Lafata, Janine Simpkins, Lois Lamerato, Laila Poisson, George Divine, and Christine Cole Johnson. The economic impact of false-positive cancer screens. *Cancer Epidemiology and Prevention Biomarkers*, 13(12):2126–2132, 2004.

[102] Daeil Lee and Jonghyun Kim. Autonomous algorithm for safety systems of the nuclear power plant by using the deep learning. In *International Conference on Applied Human Factors and Ergonomics*, pages 72–82. Springer, 2017.

[103] Chris Leong, Tim Kelly, and Rob Alexander. Incorporating epistemic uncertainty into the safety assurance of socio-technical systems. *arXiv preprint arXiv:1710.03394*, 2017.

[104] Nancy G Leveson. Software safety: Why, what, and how. *ACM Computing Surveys (CSUR)*, 18(2):125–163, 1986.

[105] Nancy G Leveson. Safeware. *System Safety and Computers. Addison Wesley*, 1995.

[106] Colin Lewis. Ai machine learning black boxes: The need for transparency and accountability, 2017. URL `https://www.kdnuggets.com/2017/04/ai-machine-learning-black-boxes-transparency-accountability.html`.

[107] Zhitao Liu and Jihong Fan. Technology readiness assessment of small modular reactor (SMR) designs. *Progress in Nuclear Energy*, 70:20–28, 2014.

[108] Robyn R Lutz. Software engineering for safety: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 213–226. ACM, 2000.

[109] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. 2015.

[110] Alexey Malanov. Machine learning: 9 challenges, Aug 2018. URL `https://www.kaspersky.com/blog/machine-learning-nine-challenges/23553/`.

[111] Arvind Malhotra and Claudia Kubowicz Malhotra. Evaluating customer information breaches as service failures: An event study approach. *Journal of Service Research*, 14(1):44–59, 2011.

[112] Bernard Marr. First FDA approval for clinical cloud-based deep learning in healthcare, Jan 2017. URL `https://www.forbes.com/sites/bernardmarr/2017/01/20/first-fda-approval-for-clinical-cloud-based-deep-learning-in-healthcare/#519dc374161c`.

[113] George Rupert Mason, Radu Constantin Calinescu, Daniel Kudenko, and Alec Banks. Assured reinforcement learning with formally verified abstract policies. In *9th International Conference on Agents and Artificial Intelligence (ICAART)*. York, 2017.

[114] Hermann G Matthies. Quantifying uncertainty: modern computational representation of probability and applications. In *Extreme man-made and natural hazards in dynamics of structures*, pages 105–135. Springer, 2007.

[115] Viktor Mayer-Schönberger and Kenneth Cukier. Big data: A revolution that will transform how we live, work, and think, 2014.

[116] John McCarthy and Edward A Feigenbaum. In memoriam: Arthur samuel: Pioneer in machine learning. *AI Magazine*, 11(3):10, 1990.

[117] Calum McClelland. The difference between artificial intelligence, machine learning, and deep learning, Dec 2017. URL `https://medium.com/iotforall/the-difference-between-artificial-intelligence-machine-learning-and-deep-learning-3aa67bff5991`.

[118] Justin McCurry. South korean woman's hair 'eaten' by robot vacuum cleaner as she slept, Feb 2015. URL https://www.theguardian.com/world/2015/feb/09/south-korean-womans-hair-eaten-by-robot-vacuum-cleaner-as-she-slept.

[119] Michael J McGrath and Cliodhna Ní Scanaill. Regulations and standards: Considerations for sensor technologies. In *Sensor Technologies*, pages 115–135. Springer, 2013.

[120] Jenna McLaughlin. The White House asked social media companies to look for terrorists. Here's why they'd fail., Jan 2016. URL https://theintercept.com/2016/01/20/the-white-house-asked-social-media-companies-to-look-for-terrorists-heres-why-theyd-fail/.

[121] Dan McMorrow. Rare events. Technical report, MITRE Corporation, 2009.

[122] Jim McPherson. How uber's self-driving technology could have failed in the fatal tempe crash, Mar 2018. URL https://www.forbes.com/sites/jimmcpherson/2018/03/20/uber-autonomous-crash-death/2/#1e20c71e44d4.

[123] MIL-STD-882D. Standard practice for system safety. 2000.

[124] Kevin Miller. Total surveillance, big data, and predictive crime technology: Privacy's perfect storm. *J. Tech. L. & Pol'y*, 19:105, 2014.

[125] Newton N Minow. *Safeguarding privacy in the fight against terrorism report of the Technology and Privacy Advisory Committee: executive summary.* DIANE Publishing, 2004.

[126] Oliver Mitchell. Regulatory challenges holding back healthcare AI, Apr 2018. URL https://www.therobotreport.com/regulatory-challenges-holding-back-healthcare-ai/.

[127] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[128] Ian I Mitroff and Abraham Silvers. *Dirty rotten strategies: How we trick ourselves and others into solving the wrong problems precisely.* Stanford University Press, 2010.

[129] Brent Daniel Mittelstadt, Patrick Allo, Mariarosaria Taddeo, Sandra Wachter, and Luciano Floridi. The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2):2053951716679679, 2016.

[130] Niklas Möller. The concepts of risk and safety. In *Handbook of risk theory*, pages 55–85. Springer, 2012.

[131] Niklas Möller and Sven Ove Hansson. Principles of engineering safety: risk and uncertainty reduction. *Reliability Engineering & System Safety*, 93(6):798–805, 2008.

[132] Jojo Moolayil. *Smarter Decisions–The Intersection of Internet of Things and Decision Science.* Packt Publishing Ltd, 2016.

[133] Anita Nuopponen. Methods of concept analysis-towards systematic concept analysis (part 2 of 3). *LSP Journal-Language for special purposes, professional communication, knowledge management and cognition*, 1(2), 2010.

[134] Ziad Obermeyer and Ezekiel J Emanuel. Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216, 2016.

[135] Michael Oppenheimer, Brian C O'neill, Mort Webster, and Shardul Agrawala. The limits of consensus. *Science Magazine's State of the Planet 2008-2009: with a Special Section on Energy and Sustainability*, 317:1505–06, 2007.

[136] Rob Palin, David Ward, Ibrahim Habli, and Roger Rivett. ISO 26262 safety cases: Compliance and assurance. 2011.

[137] Ovidiu Păucă and Letitia Mirea. Comparative study on pitch angle control of an aircraft using neural networks. In *System Theory, Control and Computing (ICSTCC), 2017 21st International Conference on*, pages 340–345. IEEE, 2017.

[138] MP Petticrew, AJ Sowden, D Lister-Sharp, and K Wright. False-negative results in screening programmes: systematic review of impact and implications. *Health technology assessment (Winchester, England)*, 4(5):1–120, 2000.

[139] Lawrence T Pinfield. A field evaluation of perspectives on organizational decision making. *Administrative science quarterly*, pages 365–388, 1986.

[140] David Pogue. A thermostat that's clever, not clunky, Nov 2011. URL `https://www.nytimes.com/2011/12/01/technology/personaltech/nest-learning-thermostat-sets-a-standard-david-pogue.html`.

[141] W Price and II Nicholson. Black-box medicine. *Harv. JL & Tech.*, 28:419, 2014.

[142] W Nicholson Price. Artificial intelligence in health care: Applications and legal issues. 2017.

[143] Bruce Ricketts. Quebec bridge collapse, Feb 2017. URL `https://www.mysteriesofcanada.com/quebec/quebec-bridge-collapse/`.

[144] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[145] Justin Rowlatt. Kolkata flyover collapse: How a tragedy brought a city together, Apr 2016. URL `http://www.bbc.com/news/world-asia-india-35941072`.

[146] William Runciman, Peter Hibbert, Richard Thomson, Tjerk Van Der Schaaf, Heather Sherman, and Pierre Lewalle. Towards an international classification for patient safety: key concepts and terms. *International journal for quality in health care*, 21(1):18–26, 2009.

[147] John Rushby. Logic and epistemology in safety cases. In *International Conference on Computer Safety, Reliability, and Security*, pages 1–7. Springer, 2013.

[148] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[149] Arnaud Sahuguet, John Krauss, Luis Palacios, and David Sangokoya. Open civic data: Of the people, for the people, by the people. *IEEE Data Eng. Bull.*, 37(4):15–26, 2014.

[150] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki. An analysis of ISO 26262: Using machine learning safely in automotive software. *arXiv preprint arXiv:1709.02435*, 2017.

[151] Paul M Salmon, Guy H Walker, and Neville A Stanton. Pilot error versus sociotechnical systems failure: a distributed situation awareness analysis of air france 447. *Theoretical Issues in Ergonomics Science*, 17 (1):64–79, 2016.

[152] David C Sandomir. Preventing terrorism in the long term: The disutility of racial profiling in preventing crime and the counterproductive nature of ethnic and religious profiling in counterterrorism policing. Technical report, Naval Postgraduate School Monterey Canada, 2009.

[153] Ian Savage. Comparing the fatality risks in united states transportation across modes and over time. *Research in Transportation Economics*, 43(1):9–22, 2013.

[154] Stephen H Schneider. Climate change: Do we know enough for policy action? *Science and Engineering Ethics*, 12(4):607–636, 2006.

[155] Bruce Schneier. Why data mining won't stop terror. *Wired News*, (March 9), 2006.

[156] Lasse Schultebraucks. A short history of artificial intelligence, Dec 2017. URL `AShortHistoryofArtificialIntelligence`.

[157] Nicu Sebe, Ira Cohen, Ashutosh Garg, and Thomas S Huang. *Machine learning in computer vision*, volume 29. Springer Science & Business Media, 2005.

[158] Evan Selinger and Woodrow Hartzog. Future - the dangers of trusting robots, Aug 2015. URL `http://www.bbc.com/future/story/20150812-how-to-tell-a-good-robot-from-the-bad`.

[159] Sanjit A Seshia, Dorsa Sadigh, and S Shankar Sastry. Towards verified artificial intelligence. *arXiv preprint arXiv:1606.08514*, 2016.

[160] Aatash Shah. Machine learning vs statistics, 2016. URL https://www.kdnuggets.com/2016/11/machine-learning-vs-statistics.html.

[161] Carl Shan. How data science can be used for social good, Jan 2015. URL http://www.carlshan.com/2015/01/08/data-science-social-good.html.

[162] EMILY Shaw. Improving service and communication with open data: A history and how-to. *Ash Center, Harvard Kennedy School, Tech. Rep*, 2015.

[163] Lance Sherry and Robert Mauro. Controlled flight into stall (CFIS): Functional complexity failures and automation surprises. In *Integrated Communications, Navigation and Surveillance Conference (ICNS), 2014*, pages D1–1. IEEE, 2014.

[164] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[165] Tavish Shrivastava. Difference between machine learning statistical modeling, Jul 2015. URL https://www.analyticsvidhya.com/blog/2015/07/difference-machine-learning-statistical-modeling/.

[166] Martyn Shuttleworth and L.T. Wilson. Type I error and type II error, Nov 2008. URL https://explorable.com/type-i-error.

[167] Eric Siegel. *Predictive analytics: The power to predict who will click, buy, lie, or die.* John Wiley & Sons Incorporated, 2016.

[168] Alex Smola and S.V.N. Vishwanathan. *Introduction to Machine Learning*. Cambridge University Press, 2008.

[169] Susan Solomon. *Climate change 2007-the physical science basis: Working group I contribution to the fourth assessment report of the IPCC*, volume 4. Cambridge university press, 2007.

[170] Olivia Solon. Who's driving? autonomous cars may be entering the most dangerous phase, Jan 2018. URL https://www.theguardian.com/technology/2018/jan/24/self-driving-cars-dangerous-period-false-security.

[171] Niladri Syam and Arun Sharma. Waiting for a sales renaissance in the fourth industrial revolution: Machine learning and artificial intelligence in sales research and practice. *Industrial Marketing Management*, 69:135–146, 2018.

[172] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[173] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Introduction to data mining. 1st, 2005.

[174] Kevin E Trenberth. Attribution of climate variations and trends to human influences and natural variability. *Wiley Interdisciplinary Reviews: Climate Change*, 2(6):925–930, 2011.

[175] Dick Nijen Twilhaar. Linkedin page. URL https://www.linkedin.com/in/dick-nijen-twilhaar-521a672/.

[176] Steve Tyrell. Regulatory strategies for AI and emerging technologies, Aug 2017. URL https://www.mastercontrol.com/gxp-lifeline/regulatory-strategies-for-ai-and-emerging-technologies-.

[177] James Urnes, Ron Davidson, and Steve Jacobson. A damage adaptive flight control system using neural network technology. In *American Control Conference, 2001. Proceedings of the 2001*, volume 4, pages 2907–2912. IEEE, 2001.

[178] Perry Van Wesel and Alwyn E Goodloe. Challenges in the verification of reinforcement learning algorithms. 2017.

[179] Kush R Varshney. Engineering safety in machine learning. In *Information Theory and Applications Workshop (ITA), 2016*, pages 1–5. IEEE, 2016.

[180] Kush R Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *Big data*, 5(3):246–255, 2017.

[181] Kush R Varshney, Ryan J Prenger, Tracy L Marlatt, Barry Y Chen, and William G Hanley. Practical ensemble classification error bounds for different operating points. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2590–2601, 2013.

[182] Suresh Venkatasubramanian. Programming and prejudice, Aug 2015. URL `https://unews.utah.edu/programming-and-prejudice/`.

[183] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.

[184] Daisuke Wakabayashi. Uber ordered to take its self-driving cars off arizona roads, Mar 2018. URL `https://www.nytimes.com/2018/03/26/technology/arizona-uber-cars.html`.

[185] Leon Shyue-Liang Wang. *Intelligent Soft Computation and Evolving Data Mining: Integrating Advanced Technologies: Integrating Advanced Technologies*. IGI Global, 2010.

[186] Larry Wasserman. Statistics versus machine learning, Jun 2012. URL `https://normaldeviate.wordpress.com/2012/06/12/statistics-versus-machine-learning-5-2/`.

[187] Jos Wassink. Trainspotting, the digital way, Apr 2017. URL `https://www.delta.tudelft.nl/article/trainspotting-digital-way`.

[188] Max Welling. Are ML and statistics complementary? In *IMS-ISBA Meeting on 'Data Science in the Next 50 Years*, 2015.

[189] William Wiersma and Stephen G Jurs. Research methods in education: An introduction. 2005.

[190] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[191] Richard T Wood, Belle R Upadhyaya, and Dan C Floyd. An autonomous control framework for advanced reactors. *Nuclear Engineering and Technology*, 49(5):896–904, 2017.

[192] Robert K Yin. Case study research: Design and methods . thousands oaks. *Sage. Young, LC and Wilkinson, IR (1989). The role of trust and co-operation in marketing channels: a preliminary study. European Journal of Marketing*, 23(2):109–122, 2003.

[193] Robert K Yin. *Case Study Research, Design & Methods 4th ed.* 2009.

[194] Jim Young, Patrick Graham, and Richard Penny. Using bayesian networks to create synthetic data. *Journal of Official Statistics*, 25(4):549, 2009.

[195] Yuhao Zhu and Vijay Janapa Reddi. Cognitive computing safety: The new horizon for reliability. *IEEE Micro*, 37:15–21, 2017.

[196] Andrew Zola. Top 10 machine learning challenges we've yet to overcome, Oct 2017. URL `https://ukraine.intersog.com/blog/augmented-reality/top-10-machine-learning-challenges/`.